



UNIVERSIDAD NACIONAL AUTÓNOMA  
DE MÉXICO

---

---

FACULTAD DE CIENCIAS

SISTEMA DE ASESORÍAS EN LÍNEA DE LA UNAM

ALUNAM

REPORTE DE  
ACTIVIDAD DOCENTE

QUE PARA OBTENER EL TÍTULO DE:  
LICENCIADO EN CIENCIAS DE LA COMPUTACIÓN

PRESENTAN:

AGUILAR SOLÍS ARIES OLAF

SAUCEDO MEDINA RODRIGO

DIRECTORA DE TESIS:

M. EN C. M. ELENA GUADALUPE IBARGÜENGOITIA  
GONZÁLEZ

2008





Universidad Nacional  
Autónoma de México

Dirección General de Bibliotecas de la UNAM

**Biblioteca Central**



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

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

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

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

1. Datos del alumno.

Aguilar  
Solís  
Aries Olaf  
56 57 99 43  
Universidad Nacional Autónoma de México  
Facultad de Ciencias  
Ciencias de la Computación  
098015665

Saucedo  
Medina  
Rodrigo  
55 77 61 46  
Universidad Nacional Autónoma de México  
Facultad de Ciencias  
Ciencias de la Computación  
098077588

2. Datos del tutor.

M en C  
María Guadalupe Elena  
Ibargüengoitia  
González

3. Datos del sinodal 1

Dra  
Amparo  
López  
Gaona

4. Datos del sinodal 2

M en C  
Wilfrido  
Martínez  
Flores

### 5. Datos del sinodal 3

Dra  
Hanna  
Jadwiga  
Oktaba

### 6. Datos del sinodal 4

L en CC  
Karla  
Ramírez  
Pulido

### 7. Datos del trabajo escrito

Sistema de Asesorías en Línea de la UNAM ALUNAM  
153 p.  
2008

## **Agradecimientos de Olaf.**

Quiero agradecer, no solo por éste trabajo sino por el recorrido para llegar a él, a mi familia entera. A mi madre por ser quien me trajo a éste mundo y cuidarme todos los días, a mi padre por haberme mostrado mis primeros juegos de video. A mi abuelita por su cariño. A mi hermano porque sé que es una persona en la que puedo confiar. A Yakko que me esperaba cuando terminaba tareas por las noches. A la gente que ya no está conmigo y a los que aún nos acompañan

A mis compañeros de primaria, secundaria y preparatoria. A todos mis amigos por los momentos divertidos y de apoyo. A cada uno de mis compañeros de la facultad quienes me ayudaron a terminar ésta carrera. A todos los profesores que compartieron su conocimiento, muchas gracias por sus enseñanzas. A la maestra Guadalupe Ibarguengoitia por tener la paciencia para ayudarnos a terminar este reporte. A Mario Hernández y a la DGSCA por darme la oportunidad de comenzar en el ámbito profesional y laboral. A Rodrigo por ayudarme a crear este trabajo y a quien también felicito por haber terminado esta carrera. A la Universidad y a mi país por otorgarme un lugar estudio de calidad.

Y finalmente a la persona más importante para mí, a mi esposa por quien me levanto cada día. Azu, te quiero, gracias por ser el amor de mi vida.

## **Agradecimientos de Rodrigo.**

Quiero empezar diciendo que si estoy aquí, es gracias a todas las personas que de una u otra forma han entrado en mi vida y me han hecho la persona que ahora soy, me gustaría poder mencionarlos a todos pero es un poco difícil así que a todas esas personas quiero darles las gracias por los buenos y malos momentos que viví y sigo viviendo con ellas ya que si no hubiera tenido esas experiencias quien sabe que sería de mi en estos momentos.

Quiero agradecerle a la maestra Guadalupe Ibargüengoitia que gracias a su ayuda, paciencia y consejos pudimos concluir la elaboración de este trabajo, a Olaf por apoyarme y ayudarme tanto en este trabajo como en muchas otras cosas, a Mario Hernández por su apoyo y enseñanzas que me hicieron crecer mucho como programador y persona y en general a todas las personas que conocí en DGSCA por darme la oportunidad de empezar a desarrollarme en un ambiente tan increíble como lo fue SERUNAM, por supuesto no puedo dejar de mencionar a la UNAM como la mejor escuela en la que he estado, gracias por todo lo que me dejó y me seguirá dando esta maravillosa institución.

En especial quiero agradecerle a mi familia, que siempre me apoya y ha creído en mí, soportándome y animándome a seguir adelante en cualquier cosa que me proponga, en especial a concluir la carrera, a mi madre que siempre estuvo al pendiente, deseando que este día llegara y me viera titulado, a mi hermana que me ha ayudado desde que era un niño, a mi padre por toda la sabiduría que me inculco aunque a veces se me olvide un poco ponerla en práctica, gracias por ser mi familia.

# Índice

<b>Introducción.</b> .....	I
<b>Capítulo 1. Conceptos Básicos.</b>	
1.1 Conceptos de e-learning. ....	2
1.2 Learning Management System. ....	3
1.3 Learning Content Management System. ....	4
<b>Capítulo 2. El Sistema ALUNAM.</b>	
2.1 ¿Qué es?. ....	7
2.2 Historia. ....	8
2.3 Herramientas del Sistema. ....	9
2.4 Demanda del Sistema ALUNAM. ....	10
2.5 Estándares del Sistema. ....	11
<b>Capítulo 3. Vista General del Sistema.</b>	
3.1 Descripción. ....	13
3.2 Tipos de usuarios del Sistema ALUNAM. ....	16
3.3 Aportes Técnicos a la Vista General. ....	17

## **Capítulo 4. Administrador.**

4.1 Descripción. ....	19
4.1.1 Usuarios. ....	20
4.1.1.1 Asesores. ....	21
4.1.1.2 Alumnos. ....	22
4.1.1.3 Administradores. ....	23
4.1.1.4 Buscar usuario. ....	24
4.1.1.5 Agregar nuevos usuarios. ....	25
4.1.2 Organización Académica. ....	26
4.1.3 Estadísticas. ....	28
4.1.4 Notificaciones. ....	31
4.2 Aportes Técnicos a la Vista de Administrador. ....	33
4.2.1 Usuarios. ....	33
4.2.2 Organización Académica. ....	39

## **Capítulo 5 Asesor.**

5.1 Descripción. ....	43
5.1.1 Presentación. ....	45
5.1.2 Contenidos. ....	46
5.1.2.1 Notas. ....	47
5.1.2.2 Actividades. ....	48
5.1.2.2.1 Tipos de Preguntas. ....	49
5.1.2.3 Bibliografía. ....	52
5.1.3 Grupo. ....	52
5.1.4 Foro. ....	54

5.1.5	Calendario. ....	57
5.1.6	Unidades de Aprendizaje. ....	58
5.1.6.1	Vista de las Unidades de Aprendizaje. ....	63
5.2	Aportes Técnicos de la Vista del Asesor. ....	65
5.2.1	Presentación. ....	65
5.2.2	Contenidos. ....	65
5.2.2.1	Notas. ....	65
5.2.2.2	Actividades. ....	66
5.2.3	Grupo. ....	69
5.2.4	Foro. ....	70
5.2.5	Calendario. ....	72
5.2.6	Unidades de Aprendizaje. ....	74
5.2.6.1	Importación de Contenidos. ....	80
5.2.6.2	Exportado de Unidades de Aprendizaje. ....	83
5.2.6.2.1	Primera Fase. ....	84
5.2.6.2.2	Exportador. ....	85
5.2.6.2.3	Uso de Patrones de Diseño. ....	90
5.2.6.2.4	Mejoras al Exportador. ....	95
5.2.6.2.5	Exportar como SCO. ....	96

## **Capítulo 6. Alumno.**

6.1	Descripción. ....	99
6.1.1	Presentación. ....	101
6.1.2	Asesor. ....	102
6.1.3	Contenidos. ....	102

6.1.4 Foro. ....	103
6.1.5 Calendario. ....	103
6.1.6 Unidades de Aprendizaje. ....	103
6.2 Aportes Técnicos a la Vista del Alumno. ....	105

## **Capítulo 7. Mejoras Generales al Sistema.**

7.1 Correos electrónicos. ....	107
7.2 Estilos. ....	108
7.3 Validaciones HTML y CSS por la W3C. ....	109
7.4 Instalación. ....	109
7.5 Migración de datos. ....	111
7.6 Etiquetas. ....	111
7.7 Barra de seguimiento. ....	112
7.8 FCKEditor. ....	113
7.9 JSCalendar. ....	116
7.10 JSDialog. ....	117

<b>Conclusiones.</b> ....	119
---------------------------	-----

<b>Apéndice A.</b> ....	121
-------------------------	-----

<b>Apéndice B.</b> ....	129
-------------------------	-----

## **Bibliografía.**

## **Introducción.**

En la actualidad el uso de herramientas en línea es fundamental para diversas áreas. La educación y la forma de enseñanza han evolucionado para utilizar estas herramientas.

La UNAM en su constante trabajo para ofrecer educación a la vanguardia y para las necesidades de los estudiantes y profesores, requiere poder dar educación no solo presencial, sino con técnicas y métodos que aporten más y mejor contenido educativo en línea. La DGSCA comenzó con el desarrollo de una plataforma que fue creciendo para satisfacer esta necesidad. Esta plataforma, ALUNAM, no era un sistema que se haya concebido para poder dar una gran variedad de cursos en línea, pero se trabajó en ella para hacerla crecer. Sin embargo, el desarrollo de sistemas que aprovechen la tecnología para la educación no es tan simple como podría parecer.

Es por eso que nos damos a la tarea de describir los distintos retos y soluciones que tuvimos que enfrentar y de los cuales obtuvimos la experiencia como para liberar un sistema de gran apoyo para los educadores y los alumnos.

Este documento describe el Sistema de Asesorías en Línea de la UNAM, ALUNAM. Se describen las características del sistema, las modificaciones y mejoras que se le realizaron a lo largo de los ciclos de su desarrollo en los que nos vimos involucrados.

Se ha organizado, para una mejor comprensión, primeramente en una breve descripción de lo que es el sistema, cómo se utiliza y algunas de las características principales que lo componen. En los siguientes capítulos se presentan sus 3 principales vistas: el administrador, asesor, alumno. Cada capítulo está dividido en una sección para la descripción de la vista y una sección para revisar la parte técnica y las modificaciones que se le realizaron a dicha vista. En el capítulo 7 se presentan, los cambios generales a la plataforma.

Para finalizar se presentan las conclusiones que se han obtenido por la realización del sistema y de este documento.

Esperamos que este trabajo sirva como una referencia para el sistema y como un ejemplo de desarrollo de plataformas reusables, adaptables y confiables.

# Capítulo 1. Conceptos Básicos.

Conforme el tiempo avanza las tecnologías evolucionan, se generan nuevas herramientas y aprendemos de ellas para seguir desarrollando distintas aplicaciones. Pero nada de esto sería útil si no lo llevamos a las disciplinas con las que el hombre se enriquece en diversas áreas. Sin duda alguna el aprendizaje se lleva a cabo día con día y al utilizar el cómputo como medio de enseñanza entonces el concepto de *e-learning* es utilizado.

*E-learning* se puede interpretar como el *aprendizaje asistido por tecnologías de la información* y es un concepto que promueve el uso de las TIC (Tecnologías de la Información y Comunicación) para la creación, adopción y distribución de contenidos, además de la adaptación a cierto ritmo de aprendizaje con ciertas herramientas que no tengan límites geográficos o relacionados con horarios específicos.

Para el cometido de esta estrategia educacional se contemplan por una parte las herramientas para mostrar contenidos como son:

- Textos
- Animaciones
- Gráficos
- Video
- Audio

Por otro lado las que sirven para comunicarse entre alumnos y tutores:

- E-mail
- Chat
- Foros

A su vez es importante que *e-learning* contemple las respectivas consideraciones pedagógicas, ventajas, desventajas y limitaciones de otras herramientas ampliamente necesarias, como lo es el Internet, así como el comportamiento de los usuarios.

Algunos años atrás se concebía el *e-learning* como una propuesta de capacitación de tipo *Just in time* (Justo a tiempo) por la aparente ventaja de reducir costos de entrenamiento y poder realizarse a cualquier hora en cualquier sitio. Sin embargo, llevado a la práctica no ha sido tan conveniente por diversos retrasos que pueden llegar a darse como la falta de equipo, el ancho de banda necesario, etc. y que sin duda alguna han opacado la incorporación de tecnología utilizada para la educación.

Un hecho importante es que realmente no es más barato que la educación presencial, por lo menos en grupos minoritarios o de bajos recursos, ya que en cualquiera de los dos tipos de enseñanza se requiere de la producción de materiales didácticos previos que cumplan con algún programa educativo.

Actualmente, como muchos conceptos computacionales, el e-learning ha pasado a una etapa de madurez en la que el concepto de educación tecnológica como tal en realidad se empieza a conocer como “*tutorización con apoyo tecnológico*” haciendo énfasis en la gran labor pedagógica que los tutores y/o profesores realizan.

## *1.1 Conceptos de e-learning.*

Cuando se utiliza Internet como medio de enseñanza entonces estamos hablando de *e-learning* puro o en línea (online), pero si es mezclado con clases educativas presenciales bajo una estructura diseñada donde las actividades se llevan a cabo completamente en línea, entonces le llamamos *b-learning* o *blended learning* (aprendizaje mezclado).

Como es de esperarse, gracias a conceptos como el *e-learning* se han generado propuestas como:

- Entrenamiento apoyado en equipos de cómputo. (*Computer Based Training - CBT*).
- Entrenamiento basado en tecnologías Web. (*Web Based Training – WBT*).
- Educación Virtual.
- Aprendizaje colaborativo basado en la Web.
- Aprendizaje colaborativo con apoyo informático.
- Televisión Educativa.
- Educación por radio.
- Juegos interactivos y/o digitales.

Gracias al gran auge que ha tenido la educación en línea se han creado diferentes conceptos, herramientas y definiciones para esta nueva área, a continuación se definirán algunos de estos conceptos y estándares que la plataforma utiliza para una comprensión mejor de su funcionamiento.

A continuación se verá una pequeña descripción de cómo se pueden dividir las plataformas por su uso:

## *1.2 Learning Management System.*

Un *Learning Management System (LMS)* es un paquete de software que permite la administración y entrega de contenidos de aprendizaje en línea, (se puede definir un contenido como un conjunto de recursos de aprendizaje tales como notas, bibliografías, actividades etc. unidos por un contexto en común como un curso, diplomado, un tema). La mayoría están basados en web para facilitar el dicho “a cualquier hora, en cualquier lugar” poder acceder a contenido de aprendizaje y administración. Generalmente un LMS permite el

registro de usuarios, entrega de actividades de aprendizaje y evaluaciones en un entorno en línea. Frecuentemente incluye herramientas como el manejo de competencias, certificados, análisis por medio de gráficas entre otros recursos.

A diferencia de un LMS un *Learning Content Management System* (LCMS) es un sistema con la capacidad primaria para crear, administrar y transferir contenidos de aprendizaje.

### *1.3 Learning Content Management System.*

Un LCMS incluye la siguiente funcionalidad:

- a) Creación de contenidos: El sistema puede crear nuevos contenidos de aprendizaje, construcción de plantillas, importación sencilla de documentos externos y una forma sencilla de importar los contenidos de aprendizaje creados en otros sistemas.
- b) Creación de evaluaciones: Puede llevar el registro de los participantes y puede crear evaluaciones avanzadas sin la ayuda de otras herramientas.
- c) Herramientas de colaboración: Permiten la administración del trabajo, de tal forma que diferentes expertos en la materia y diseñadores puedan acceder al contenido y administren el proceso de aprobación.
- d) Administración de contenidos: Proporciona herramientas para una efectiva administración de contenidos como lo sería un fuerte esquema y metadatos, control de versión, la habilidad de meter y sacar contenidos y archivarlos.

- e) Soporte de iniciativas formales para el aprendizaje: Puede tener un portal para colocar los cursos, soportar iniciativas de aprendizaje con varios modos de hacer accesible los contenidos a terceros como lo sería soporte a PDA's<sup>1</sup>, flash<sup>2</sup>, PDF<sup>3</sup>, etc.
  
- f) Recuperación de contenidos: Permite la rápida y fácil recuperación de contenidos para su reuso, con un fuerte esquema y metadatos.
  
- g) Herramientas de Exportación: Incluyen herramientas para exportar y publicar contenidos a la mayoría de los estándares en el *e-learning* (*SCORM*, *QTI*, *IMS*, etc.).
  
- h) Apoyo al rendimiento y aprendizaje informal: Tiene herramientas de apoyo al rendimiento tales como el conocimiento compartido punto a punto (*peer-to-peer*) y herramientas de simulación. El conocimiento compartido punto a punto es una parte importante para el proceso de aprendizaje informal.

Actualmente la plataforma ALUNAM soporta los primeros tres incisos de las características de un *LCMS* en su totalidad. Los puntos restantes, los soporta medianamente a excepción del *Apoyo al rendimiento y aprendizaje informal*.

---

<sup>1</sup> *Personal Digital Assistant* (Asistente Digital Personal), computadora de mano.

<sup>2</sup> Software para contenidos multimedia.

<sup>3</sup> *Portable Document Format* (formato portátil de documento), formato de almacenamiento de documentos.

## Capítulo 2. El Sistema ALUNAM.

### 2.1 ¿Qué es?

Asesorías en línea de la UNAM o ALUNAM es un sistema de asesorías en línea integrado por un conjunto de herramientas diseñadas con el objetivo de atender las necesidades y retos que desde la educación se plantean y así robustecer la enseñanza presencial, continua y a distancia. *Fig. 2.1*

ALUNAM es un LCMS que nace, se conforma y evoluciona a partir de herramientas y bibliotecas de cómputo diseñadas para ofrecer soluciones concretas a los distintos programas y proyectos de la educación superior.

A través de ALUNAM los asesores administran en línea los contenidos didácticos de sus materias, pueden llevar un control de avance de los alumnos y establecen una comunicación personal y grupal. Los alumnos por su parte, tienen acceso a los contenidos y ejercicios que marcan los asesores, así como a notas, bibliografía y carpetas de trabajo.

La plataforma fue concebida en la Coordinación de Servicios Educativos en Red, UNAM, de la DGSCA (Dirección General de Servicios de Cómputo Académico) y se apega a los estándares internacionales para cubrir aspectos de accesibilidad y modularidad, así como también para impulsar la interoperabilidad con otros sistemas.

Desarrollado con tecnologías de software libre como Jakarta, Struts, Java y PostgreSQL, (*Ver Apéndice B*) permite la incorporación de diferentes servicios en red para el apoyo de la enseñanza y el aprendizaje.

ALUNAM maneja exportación de contenidos que cumplen con el estándar SCORM versión 2004 y permite el empaquetado de contenidos para su exportación hacia otros LCMS, así como la importación de contenidos que cumplan con el estándar IMSCP 1.1.2 y 1.1.3. (Ver Apéndice A)

Consta de un entorno de administración desde el cual se configuran los cursos, se dan de alta los alumnos, se manejan contenidos, se habilitan servicios, etc.

The screenshot shows the ALUNAM web interface. At the top, there is a purple header with the text 'ASESORÍAS EN LÍNEA UNAM'. Below this, there is a navigation menu with options like 'Plan de estudios', 'Mis Asesorías', and 'Análisis y diseño orientado a Objetos con UML (pavo abecedario)'. A table titled 'Usuarios > Asesores > A' is displayed, listing users with columns for 'Nombre', 'Clave', 'Correo electrónico', 'Accesos', 'Datos personales', and 'Aseso'. The table contains three rows of data.

Nombre	Clave	Correo electrónico	Accesos	Datos personales	Aseso
Abecedario Olafor	abc	kughiu@oiuo.com	Consultar	Editar	Editar
Abecedario Pavo	pavo	ariesolaf@gmail.com	Consultar	Editar	Editar
Águilar Aries	anular	kihalkrdi@oniunoi.com	Consultar	Editar	Editar

FIG.- 2.1 DIFERENTES PLATAFORMAS ALUNAM.

## 2.2 Historia.

El proyecto nació como parte de una ayuda a la Facultad de Estudios Superiores de Cuautitlán, los cuales necesitaban una forma de poner el nombre, temario y tiempo que iban a durar los cursos que impartían, en esa época solo se

constaba de la sección de notas la cual servía para colocar estos aspectos en línea.

En un inicio la plataforma fue diseñada e implementada por el Act. Mario García Burgos quien dadas las necesidades descritas con anterioridad y las herramientas disponibles empezó con el desarrollo del sistema.

Retomando este esfuerzo, la Coordinación de Servicios Educativos en Red, de la DGSCA empezó a crear una plataforma de educación en línea para futuros proyectos como el de Bibliotecología que ya contaba con una plataforma que constaba de una parte de registro de actividades, calendario, notas, bibliografía y foro, realización que estuvo a cargo del Jefe de Departamento Mario Hernández Mayorga.

ALUNAM surge como una necesidad natural de poder tener un sitio en el que se pudieran manejar temarios completos, esto incluye temas y subtemas, por parte de las Asesorías de Bibliotecología. Durante el periodo de Abril de 2004 a Agosto de 2005 se trabajó para poder integrar manejo de interfaces junto con el diseño gráfico, esto como parte de funcionamiento y vista, además de otras funcionalidades como son:

- Archivo de etiquetado.
- Ligas en las notas.
- Ordenamiento de archivos en carpeta
- Orden de temas y subtemas.
- Envío de correos electrónicos
- Editor en el navegador Internet Explorer

### *2.3 Herramientas del Sistema.*

ALUNAM cuenta con un conjunto de herramientas distribuidas de la siguiente forma:

- **Herramientas de distribución de contenidos:** Editor de contenidos en línea, repositorios de archivos de imágenes y de texto como biblioteca, inserción de hipervínculos, así como administración de calendario de contenidos.
- **Herramientas de comunicación y colaboración:** Foros de discusión por curso, formación de grupos de trabajo, comunicación entre tutores y miembros del curso, novedades y calendario del curso.
- **Herramientas de seguimiento y evaluación:** Estadísticas y fichas personales por alumno, seguimiento de cada actividad, sistemas de exámenes editables por el docente o tutor y reportes de actividades.
- **Herramientas de administración y asignación de permisos:** Otorgamiento de permisos y autorizaciones, asignación de permisos por perfil de usuario, administración personal de perfiles de usuario y proceso de inscripción.

El sistema permite ser configurado para ser utilizado por un profesor para impartir distintas materias ó por una institución para el manejo y seguimiento de sus distintas carreras.

## *2.4 Demanda del Sistema ALUNAM.*

Al paso del tiempo varias dependencias de la UNAM han formado una estrecha colaboración con la coordinación y el sistema ha sido adaptado para las necesidades de las mismas. Algunas de estas adaptaciones son:

- Oferta educativa DGSCA.
- Asesorías, Licenciatura en Bibliotecología y Estudios de la información.
- Maestría en Docencia para la Educación Media Superior (MADEMS).
- Colegios de Facultad de Filosofía y Letras.
- Facultad de Estudios Superiores Aragón (PAPIME, Periodismo).
- Diplomado de Multimedia XI emisión.
- CESU / Facultad de Ciencias Políticas y Sociales.

- Instituto de Geofísica.
- Centro de Enseñanza de Lenguas Extranjeras (CELE).
- Facultad de Estudios Superiores Iztacala.
- Facultad de Estudios Superiores Cuautitlán.
- Facultad de Química.
- Oferta Educativa SER-UNAM.
- Facultad de Estudios Superiores Aragón (Derecho).
- Instituto de Matemáticas.
- Facultad de Psicología.
- Consejo Académico del Bachillerato.
- Diseño Industrial.
- Facultad de Ciencias Políticas y Sociales.
- DGSCA Oferta "Diplomado en Edición Digital".

En las que varias de ellas están disponibles y algunas otras están bajo desarrollo.

Así como dependencias internas de la UNAM, existen algunas otras organizaciones que también hacen uso del sistema como son:

- Universidad del Claustro de Sor Juana
- Secretaría de Salud
- Aeropuertos y Servicios Auxiliares

## *2.5 Estándares del Sistema.*

Algunos estándares internacionales que soporta tanto completa como parcialmente la plataforma se listan a continuación:

- EL IMS-QTI (*Question and Test Interoperability Specification*) utiliza el lenguaje XML para el intercambio de preguntas y evaluaciones entre diferentes sistemas, consiste de un modelo que define la estructura de las preguntas, evaluaciones y resultados de estas. (*Ver Apéndice A*)
- El modelo de información del IMS *Content Packaging (CP)* describe las estructuras de datos que son usadas para proporcionar una interoperabilidad de contenidos basados en web con herramientas para la creación de contenidos. (*Ver Apéndice A*)
- *SCORM (Shareable Content Object Reference Model)* es una colección de estándares y especificaciones para el e-learning basado en web, define la comunicación entre el contenido del lado del cliente y un sistema servidor llamado ambiente de ejecución (*run-time environment*), *SCORM* también define como los contenidos deben ser empaquetados en archivos zip<sup>1</sup> para ser transferidos. (*Ver Apéndice A*)

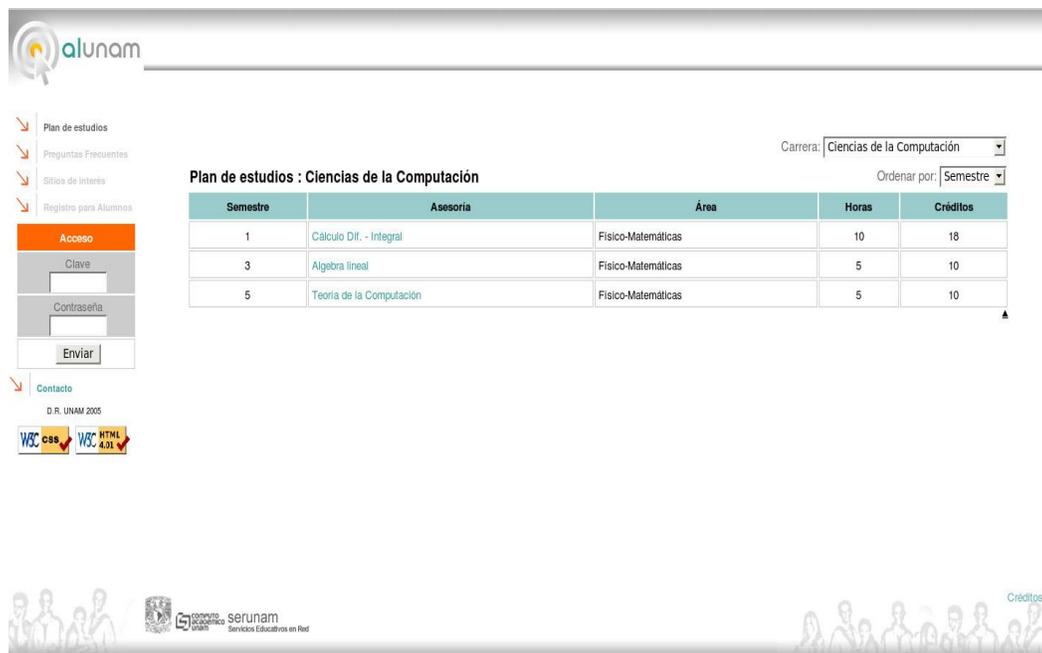
---

<sup>1</sup> Formato de almacenamiento para la compresión de datos

## Capítulo 3. Vista General del Sistema.

### 3.1 Descripción.

La primera página que se muestra al ingresar al sistema es la Vista General, la cual se puede dividir en dos partes: *Fig. 3.1*



The screenshot displays the Alunam platform interface. On the left, there is a navigation menu with links for 'Plan de estudios', 'Preguntas Frecuentes', 'Sitios de interés', and 'Registro para Alumnos'. Below this is an 'Acceso' section with input fields for 'Clave' and 'Contraseña', and an 'Enviar' button. At the bottom left, there is a 'Contacto' section with the text 'D.R. UNAM 2005' and icons for W3C, CSS, HTML, and XHTML. The main content area shows the 'Plan de estudios : Ciencias de la Computación' for the 'Carrera: Ciencias de la Computación'. A table lists the courses, and a dropdown menu allows sorting by 'Semestre'.

Semestre	Asesoría	Área	Horas	Créditos
1	Cálculo Dif. - Integral	Físico-Matemáticas	10	18
3	Álgebra lineal	Físico-Matemáticas	5	10
5	Teoría de la Computación	Físico-Matemáticas	5	10

FIG.- 3.1 VISTA GENERAL DE LA PLATAFORMA ALUNAM.

### ***Menú Lateral.***

Este menú se encuentra del lado izquierdo de la pantalla y en él se muestran opciones sobre la plataforma dependiendo de lo que se requiera por versión.

*Fig.- 3.2*

En general se tiene:

El Plan de Estudios, el cual muestra una lista de las asesorías por carrera.

Preguntas frecuentes y sitios de interés son opciones que se dejan abiertas para el uso específico que la institución que lo use le desee dar.

El registro para alumnos permite hacer un pre-registro de los alumnos para que ellos mismos se puedan dar de alta en la plataforma; posteriormente el administrador los dará de alta en las respectivas asesorías. En seguida se encuentra un cuadro de acceso donde se pide la clave y contraseña para que cualquier tipo de usuario pueda entrar al sistema.

Al final se presenta una liga de contacto para comunicarse con el administrador del sistema ya sea para una duda o aclaración, por medio de un correo electrónico.

También se muestran los logotipos de CSS y HTML 4.01, los cuales indican que la plataforma es válida por la W3G en estos rubros. (*Ver Tema 7.3*)



FIG.-3.2 MENÚ LATERAL GENERAL.

### ***Zona Central.***

En esta parte aparece la información que se despliega al elegir alguna de las opciones del menú lateral, principalmente del Plan de Estudios donde se mostrará una lista de las asesorías mostradas por carrera que actualmente están siendo impartidas por medio de la plataforma, en esta lista se muestran el semestre, nombre de la asesoría, área, entre otros datos de las asesorías impartidas, así mismo, se puede listar las diferentes carreras que existen y ordenar sus asesorías de acuerdo a ciertos criterios mostrados en la parte superior derecha. *Fig. 3.3*

Al seleccionar el nombre de la asesoría se muestra una pantalla emergente donde se encuentra más información de la respectiva asesoría.

Al escoger registro por alumno, se muestra un formulario para el llenado de los datos del alumno el cual cuenta con mensajes de alerta para no ingresar información ya existente en el sistema.

Carrera:

**Plan de estudios : Ciencias de la Computación** Ordenar por:

Semestre	Asesoría	Área	Horas	Créditos
1	Cálculo Dif. - Integral	Fisico-Matemáticas	10	18
3	Algebra lineal	Fisico-Matemáticas	5	10
5	Teoria de la Computación	Fisico-Matemáticas	5	10

FIG.- 3.3 ZONA CENTRAL.

### 3.2 Tipos de usuarios para el Sistema ALUNAM.

Los roles manejados por la plataforma son:

*Alumno:* Como su nombre lo indica es el rol al que va dirigido todos los contenidos creados en la plataforma.

*Asesor:* Es el encargado de la creación de contenidos para las asesorías que imparte.

*Administrador:* Es el que se encarga de la administración de los usuarios de la plataforma, los cursos y niveles que existan, de las notificaciones del sistema y tiene acceso a las estadísticas que se generan.

En los subsecuentes capítulos se describen las funcionalidades que el sistema proporciona a cada tipo de usuario.

### *3.3 Aportes Técnicos a la Vista General.*

Para esta vista se modificó la parte de estilos ya que carecía de ambientación gráfica. También para dar una mayor información a los usuarios se agregó la parte de mostrar la presentación de la asesoría en una ventana emergente al seleccionar el nombre de la asesoría. Las asesorías que no están siendo impartidas actualmente no mostrarán esta liga, si dicha asesoría fuera impartida por más de un tutor, primero se muestra una lista de tutores y al escogerlos se mostrará la presentación correspondiente, con la posibilidad de navegar sobre las diferentes presentaciones de las asesorías impartidas.

Otra aportación a esta vista fue en la parte del Registro para Alumnos, el cual ahora presenta tanto en la clave como en el número de cuenta un verificador que indica si el dato ya existe dentro de la base de datos del sistema e impide el envío del formulario hasta que su información sea consistente con la base de datos del sistema, esto se logra gracias a la tecnología *AJAX* (*Ver Apéndice B*), la cual se utilizó gracias a una biblioteca llamada *AJAXTags*. Esta tecnología así como su uso dentro de la plataforma será explicada en la Parte Técnica del Administrador.

## Capítulo 4. Administrador.

### 4.1- Descripción.

Este rol se encarga, como su nombre lo indica, de la administración de la plataforma en cuanto a alta, baja, consulta, cambio de usuarios y de los diferentes rubros que componen la organización académica, además de contar con una sección de estadísticas que se generan por el uso del sistema y de una parte de notificaciones para enviar a quien corresponda.

Al acceder al sistema se puede ver del lado izquierdo el menú principal, donde se encuentran las principales funcionalidades clasificadas esencialmente en:

- Usuarios
- Organización Académica
- Estadísticas
- Notificaciones

*Fig. 4.1*



FIG.- 4.1 MENÚ LATERAL DEL ROL ADMINISTRADOR.

A continuación se detalla cada división.

### 4.1.1 Usuarios.

En esta sección se tiene la posibilidad de ver a los distintos usuarios que forman parte del sistema; también se ofrecen herramientas para la creación de nuevos usuarios, borrado, modificación de sus datos personales y asignación de cursos.

#### 4.1.1.1 Asesores

Al seleccionar el rol de asesor se puede ver una lista de todos los asesores que forman parte del sistema, ya sea buscando en el índice por letra o bien una lista general ordenada alfabéticamente. Se pueden consultar diversos datos como su clave, número de accesos al sistema o incluso se les puede enviar un correo electrónico.

Así mismo también se pueden ver los datos estadísticos por sesión, donde se obtienen referencias como la fecha, la dirección IP<sup>1</sup> de la máquina desde la cual se conectó y el navegador que utilizó. Se Tiene la vista de sesiones por IP, donde se puede observar la lista de IPs y el número de veces que ha accedido este usuario al sistema desde una computadora específica. Existe la posibilidad de editar los datos del asesor además de sus asesorías. Desde esta edición de asesorías es posible constatar las asesorías que imparte el asesor y los alumnos inscritos en ésta, se les puede enviar un correo electrónico y también observar sus datos. Por último se brinda la posibilidad de copiar el temario de una asesoría previamente hecha.

En la parte de asignar asesoría a algún asesor, se muestran todas las asesorías que tiene cada carrera hasta el momento. En esta nueva lista se observa el nombre de la asesoría, el área a la que pertenece, el semestre, el número de créditos seguido por el número de horas que durará el curso. Por último el sistema solicita el tipo de asesoría, es posible seleccionar más de una a la vez.

Es posible verificar el contenido de la carpeta del usuario, si dicha carpeta no existiera, el sistema brinda da la posibilidad de crearla. Por último, se muestra una opción de borrado.

---

<sup>1</sup> Número que identifica de manera lógica y jerárquica a una máquina conectada a una red.

También se cuenta con la opción de agregar nuevos asesores ya sea manualmente o ingresando la información a través del formulario que se presenta o por medio de una hoja de Excel. (*Ver sección 4.1.1.5*)

#### *4.1.1.2 Alumnos.*

De forma similar a la de asesores, se puede ver a todos los alumnos que forman parte del sistema, ya sea buscando en el índice alfabético por letra o bien consultando todos a la vez.

En esta vista los alumnos aparecen listados por nombres, con la clave que poseen para el ingreso al sistema, su correo electrónico, el cual es una liga para mandar correos, el número de cuenta del alumno, la carrera a la que pertenece dicho alumno, una opción para consultar sus accesos al sistema, otra opción para editar sus datos personales y otra para borrarlo.

Al acceder a *Consultar* en la columna de *Acceso*, aparece una lista de todas las páginas del sistema que ha visitado este usuario. El sistema brinda la posibilidad de ordenar esta lista en forma ascendente o descendente, así mismo, se pueden consultar las vistas de Fecha-IP-Navegador por sesión, donde aparece una lista con la fecha, seguida de la IP de la computadora que se utilizó para acceder al sistema, además del navegador que se utilizó en dicha ocasión. Por último se tiene la vista de sesiones por IP, donde se puede observar la lista de IPs y el número de sesiones que ha tenido, es decir cuántas veces ha accedido este usuario al sistema desde alguna computadora específica.

Volviendo al listado principal, al elegir *Editar*, se tiene la posibilidad de actualizar los datos del alumno, excepto por la clave que es el único campo que nunca se podrá cambiar. En esta misma vista se encuentra la sección de asesorías, en la cual se muestra una lista con todas aquellas en las que está inscrito. Dicha lista se compone por el nombre de la asesoría, seguido por el

nombre del asesor que está impartiendo dicha asesoría, el área y semestre en que se imparte esta asesoría, su valor en créditos y las horas de duración.

Por último se tiene la opción de borrar al alumno de la asesoría también en el listado principal. Si no estuviera inscrito a ninguna asesoría se muestra un mensaje aclaratorio, independientemente de estar inscrito a alguna asesoría o no, en la parte superior derecha se encuentra la opción para asignar asesorías. Al utilizar esta liga se mostrará una lista de todas las posibles asesorías en las que el alumno se puede inscribir.

En la parte superior izquierda se muestra un cuadro de información donde se presenta el nombre, correo, con la posibilidad de mandar un correo electrónico, y nombre de la carrera; del lado derecho se muestra un posible ordenamiento para la lista, por área, semestre o asesoría. La lista de las asesorías disponibles se encuentra listada por el nombre de la asesoría, seguido por el nombre del asesor que está impartiendo dicha asesoría, el área y semestre en que se imparte, su valor en créditos y las horas de duración. Se puede seleccionar más de una asesoría a la vez, en dado caso que todas las asesorías de dicha carrera ya estén asignadas al alumno o no existieran asesorías para la carrera en la que está inscrito el alumno, se muestra el mensaje correspondiente.

Es posible dar de alta nuevos alumnos, esta sección se encuentra debajo de la lista de los alumnos, ya sea manualmente a través del formulario que aparece o por medio de una hoja de Excel. (*Ver sección 4.1.1.5*)

### *4.1.1.3 Administrador.*

La lista de administradores se encuentra constituida por el nombre del administrador, la clave que poseen para el ingreso al sistema, el correo electrónico, una opción para consultar sus accesos al sistema, otra para editar sus datos personales y otra para borrar al administrador del sistema

Con *Consultar* aparecen las visitas por página, la cual muestra una lista de todas las páginas del sistema que ha visitado este usuario. Así mismo se puede también ver las vistas de Fecha-IP-Navegador por sesión con una lista donde aparecerá la fecha, seguida de la IP de la computadora que se utilizó para acceder al sistema y por último el navegador que se utilizó en dicha ocasión. Finalmente la vista de sesiones por IP, donde se podrá observar la lista de IPs y el número de sesiones que ha tenido, esto es cuantas veces ha accedido este usuario al sistema desde una computadora específica.

Con *Editar* se pueden actualizar los datos del administrador elegido, excepto por la clave que es el único campo que nunca se podrá cambiar.

En la parte superior del listado se cuenta con la opción para dar de alta nuevos administradores, esta sección se encuentra debajo de la lista de los administradores, de igual manera que con los otros usuarios se pueden utilizar los dos métodos antes mencionados para la inserción de nuevos administradores al sistema. (*Ver sección 4.1.1.5*)

#### *4.1.1.4 Buscar usuario.*

Por último, se tiene la sección de búsqueda, en la cual se brinda la posibilidad de encontrar a cualquier usuario del sistema en base a ciertos parámetros, los cuales, aparecen divididos en dos grupos, el tipo de usuario que está conformado por todos, alumno, asesor y administrador y el campo que consta de nombre, apellido, correo electrónico y clave. Gracias a estos dos grupos se puede hacer una consulta con cualquier tipo de usuario y campo. El sistema, dependiendo del tipo y campo, muestra una ayuda en una lista desplegada en el campo de entrada del dato para poder realizar una búsqueda más sencilla. Al localizar a los usuarios que empatan con los parámetros, se muestra una lista de todos los posibles resultados, mostrados por nombre, clave, correo electrónico, el cual es una liga para poder mandarle un correo, el tipo de usuario del que se

trata y una liga para editar sus datos personales que es igual a la que se muestra en la parte de los roles descritos anteriormente y contiene las mismas funcionalidades. *Fig. 4.2*

**Buscar usuario**

Palabra clave	Seleccionar tipo de usuario	Seleccionar campo	
<input type="text"/>	<ul style="list-style-type: none"><li>✓ Todos</li><li>Alumno</li><li>Asesor</li><li>Administrador</li></ul>	<ul style="list-style-type: none"><li>✓ Todos</li><li>Nombre</li><li>Apellidos</li><li>Correo electrónico</li><li>Clave</li></ul>	<input type="button" value="Buscar"/>

FIG.- 4.2 BARRA DE BÚSQUEDA DE USUARIOS.

#### 4.1.1.5 Agregar nuevos usuarios.

Para dar una mayor facilidad y teniendo en cuenta los requerimientos de terceros sobre la plataforma, se decidió darle la opción al administrador de poder añadir usuarios a través de una hoja de *excel*, esta opción se agregó tanto en los listados de alumnos como los de asesores. Esto se logró gracias a un proyecto de Apache (*Apache Software Foundation*) llamado POI<sup>2</sup>, el cual consta de una bibliotecas en java para el uso de formatos pertenecientes a Microsoft, en nuestro caso se hace uso de ellas para obtener la información de hojas de Excel. *Fig. 4.3*

---

<sup>2</sup> El proyecto POI consiste en bibliotecas para manipular varios formatos de ficheros basados en el formato de Documento Compuesto OLE 2 de Microsoft, utilizando java.

alunam

Cerrar ventana

**Ejemplo de formato del archivo excel**

Nombre	Apellidos	Clave	Contraseña	Correo electrónico	Sexo	Edición de asesorías
Juan	Sánchez	mju655	a7884d	jsan@hotmail.com	masculino	no
Mario	Hernández	alihdez	a7884d	jsan@hotmail.com	masculino	si
Juan	Hernández	mju655	j1277	allli@gmail.com	femenino	si

Cerrar ventana

W3C CSS W3C HTML 4.01

Créditos

COMPUTO ACADÉMICO unam serunam Servicios Educativos en Red

FIG.- 4.3 EJEMPLO DE LA INSERCIÓN DE NUEVOS USUARIOS POR MEDIO DE UNA HOJA DE EXCEL.

#### 4.1.2 Organización Académica.

En esta sección se tiene la posibilidad de consultar, agregar y editar los diferentes rubros académicos con los que cuenta la plataforma, estos son:

##### *Áreas.*

Al ingresar a esta sección se muestra una lista de todas las áreas con las que cuenta la plataforma, consta del nombre del área seguido de una liga para su edición, al acceder a esta opción se presentará un campo con el antiguo nombre el cual puede ser cambiado en ese momento. Debajo se encuentra una lista

simple de todas las asesorías que se tienen registradas sobre esta área, en la parte superior derecha se encuentra una liga para agregar nuevas áreas que lleva a un formulario.

### ***Carreras.***

En esta sección se muestra una lista de todas las carreras existentes en el sistema. Esta lista consta del nombre de la carrera seguido de una liga para su edición, en esta opción se presenta un campo con el antiguo nombre el cual puede ser cambiado en ese momento. En esta página también se tiene la opción para agregar nuevas carreras que lleva al formulario correspondiente.

### ***Asesorías.***

Esta sección despliega una lista de las asesorías que contiene el sistema, las cuales pueden ser ordenadas por carrera, así como semestre, área y nombre de la asesoría. La lista está conformada por el semestre en el que es impartida dicha asesoría seguido del nombre de la asesoría, el área a la que pertenece, los créditos con los que cuenta, así como las horas de duración, su estado ya sea activa o inactiva y por último la opción para editar éstas.

Es posible agregar nuevas asesorías por medio de un formulario en la parte inferior de la página, que cuenta con atributos como nombre, área, carrera, semestre, tipo, carácter, cabe destacar que a estos últimos tres, se da la posibilidad de agregar nuevas opciones por medio de ligas que se encuentran en la parte superior de la página, debajo de la liga de agregar asesoría. Al utilizar alguna de ellas se muestra una lista de las respectivas opciones empezando por nombre y seguidos de una liga que permite editarlas con un formulario. La edición de estos tres campos sirve de manera similar.

Al acceder a la parte de edición se presenta un formulario que consta del nombre de la asesoría y una lista de áreas para elegir. Las carreras se presentan

listadas para poder escoger más de una carrera a la que pertenecerá esta asesoría, a continuación se presenta el semestre, tipo y carácter de la asesoría, todos estos se pueden escoger a través de una lista hecha a partir de datos del sistema, los créditos y horas y por último, el estado.

### ***Asesorías por asesor.***

Al ingresar en dicha sección se presenta una lista de todas las asesorías impartidas actualmente por algún asesor, se muestran por nombre de la asesoría, el asesor que las imparte, una liga para editar el grupo con el que cuenta dicha asesoría, ver el temario del curso y por último se brinda la posibilidad de borrar la asesoría del sistema.

Al elegir la opción de editar grupo, se puede observar del lado superior izquierdo información importante correspondiente a esta asesoría, debajo se encuentra una lista de todos los alumnos que actualmente se encuentran inscritos en la asesoría y si no hubiera ningún alumno inscrito se presenta el mensaje correspondiente. La lista cuenta con el nombre del alumno, apellido, correo electrónico, carrera y al final se presenta una liga para darlo de baja definitivamente de esta asesoría. En la parte superior derecha de esta lista se presenta la opción para agregar alumnos a esta asesoría. En dado caso que exista al menos un alumno inscrito en la asesoría es posible mandar un correo a todo el grupo. Cuando se agrega al alumno, se presenta una lista de todos aquellos que tengan asociada la misma carrera donde está siendo impartida esta asesoría y es posible agregar a varios simultáneamente.

### ***4.1.3 Estadísticas.***

En esta parte se pueden consultar las estadísticas que el sistema ha guardado del comportamiento de los usuarios, está dividida en: *Fig. 4.4*

### ***Navegadores.***

Muestra una lista con el nombre del navegador y el nombre del sistema operativo, a continuación se muestra el número de sesiones que se han creado por cada especificación navegador-sistema operativo.

### ***Meses.***

Muestra una lista con el mes y el número de sesiones que se han creado por cada mes.

### ***Días.***

Muestra una lista con el día, el número de sesiones que se han creado por cada día y al final se muestra una liga para poder ver una lista de todos los usuarios que entraron al sistema dicho día. La lista muestra la clave del usuario, su rol en el sistema y por último el número de veces que ingresó.

### ***Horas.***

Muestra una lista con el día, la hora en la que entró al sistema y por último se muestran las veces que el usuario ha entrado al sistema por día.

### ***Páginas.***

Muestra una lista con la página consultada, el número de veces que la página ha sido vista y por último se muestran una pequeña descripción que se tiene de la página en cuestión.

### ***Visitas por usuario.***

Se muestra una lista donde aparece el usuario, su rol y por último se presenta el número de visitas del usuario en cuestión.

### ***Usuario por tipo.***

Muestra una lista donde aparece el rol del usuario y el número de usuarios en cuestión que están dados de alta en el sistema.

### ***Usuarios por navegador***

Muestra una lista donde aparece el número de sesiones, el navegador y el sistema operativo que se utilizó cada vez que se inició una sesión, el nombre del usuario y por último el rol al que pertenece.

### ***Participaciones de los usuarios en el Foro.***

Muestra una lista con el número de participantes que se han tenido en el foro, el nombre del usuario, el rol al que pertenece y el nombre del foro en el que se han hecho las participaciones.

### ***Direcciones IP desde donde se conectan los usuarios.***

Muestra una lista que contiene el número de accesos desde la IP registrada, así como el lugar donde se hicieron dichos accesos, el nombre del usuario y su rol.

Todas las opciones cuentan, en la parte superior derecha con una opción para ordenar la lista de forma ascendente o descendente.

Ordenar por Número de accesos desde la IP Descendente ▾

Número de accesos desde la IP	IP	Usuario	Tipo
4	200.67.211.146	admon	Administrador
3	127.0.0.1	asesor1	Tutor
3	200.67.211.146	alumno1	Alumno
2	132.248.200.19	asesor1	Tutor
1	189.137.27.63	admon	Administrador
1	132.248.200.2	admon	Administrador
1	189.137.27.63	asesor1	Tutor
1	189.137.55.112	admon	Administrador
1	127.0.0.1	admon	Administrador
1	200.67.211.146	asesor1	Tutor
1	189.145.71.60	admon	Administrador
1	132.248.200.19	admon	Administrador

FIG.- 4.4 EJEMPLO DE LA SECCIÓN DE ESTADÍSTICAS.

#### 4.1.4 Notificaciones.

El objetivo de ésta sección es la de brindar al administrador una forma sencilla para poder enviar notificaciones a cualquier usuario o grupos de usuarios que se necesite, siempre y cuando estén dados de alta en el sistema, para facilitar el envío de estas notificaciones, se crean plantillas especiales que podrán ser utilizadas posteriormente, las cuales pueden ser llenadas con datos de los usuarios automáticamente a través de la inserción de variables como el nombre o clave del usuario al momento de crear la plantilla de notificación.

Al ingresar a esta sección se muestra una lista de todas las notificaciones que previamente se han creado en el sistema. *Fig. 4.5*

Dicha lista contiene el nombre de la notificación, una opción que permite tener una vista previa de la notificación, una liga para su edición, una de borrado y por último una liga para seleccionar los usuarios a los que se les mandará dicha notificación.

Es posible agregar nuevas notificaciones en esta vista; al acceder a esta opción se muestran las instrucciones con los campos correspondientes de título y mensaje.

Agregar Notificación

Notificación	Vista previa			
Calificaciones	Ver	Editar	Borrar	Notificar a usuarios
Inicio del curso	Ver	Editar	Borrar	Notificar a usuarios

FIG.- 4.5 EJEMPLO DE LISTA DE NOTIFICACIONES DEL SISTEMA.

### ***Cerrar sesión***

Por último se tiene la opción para salir del sistema y regresar a la página principal. También se da la posibilidad de mandar un correo a soporte técnico por si existe alguna duda o error en el sistema.

## *4.2 Aportaciones Técnicas a la vista de Administrador.*

En esta sección se habla a detalle de los aspectos técnicos, tecnologías usadas y retos que se superaron para dejar la plataforma ALUNAM en su versión 1.0.

Al igual que en la sección de administrador se dividieron los cambios conforme al menú lateral del mismo.

### *4.2.1 Usuarios.*

Se agregó la posibilidad de borrar cualquier tipo de usuario que se encuentre dado de alta en el sistema. En el caso de que sea un usuario de tipo asesor, pasan a formar parte de un usuario por omisión del sistema, todas las asesorías que tenía asignadas, esto con el motivo de no perder las asesorías previamente creadas.

En los listados se agregó en el campo de correo electrónico la opción de poder enviar correos a la dirección que aparece; como todos los formularios para correo electrónico, no se permite cambiar los campos de remitente y destinatario.

Al acceder a la parte de datos personales se omitió la posibilidad de cambiar la clave del asesor con fines de consistencia de la base de datos.

Una de las principales novedades con las que cuenta esta sección es en la parte de los formularios para ingresar usuarios de cualquier tipo. Se agregó funcionalidad a los campos de clave (para los tres tipos de usuario) y número de

cuenta (para el alumno), los cuales permiten saber al momento en el que se está ingresando una nueva clave o número de cuenta si ya existe en el sistema, si existiera, al formulario se le quita la posibilidad de ser enviado hasta que se inserten datos que no existan previamente, esto se hace con la finalidad de mantener la consistencia de la base de datos ya que estos dos campos deben ser únicos dentro de ésta. Esto se logró con una tecnología llamada *ajaxtags*<sup>3</sup>, la cual sirve para hacer uso de otra tecnología llamada *AJAX* (Ver Apéndice B) de manera más fácil, esto lo logra a través de unas etiquetas (*tags*) para *jsp*<sup>4</sup>. Existen diferentes *etiquetas* dependiendo de los tipos de información que se pida y como se quiera mostrar, en este caso utilizó la etiqueta de autocompletar `<ajax:autocomplete>`, ya que nuestra finalidad es indicarle al usuario si el tipo de dato que esta insertando ya existe. Para utilizar esta tecnología se necesita la biblioteca de *ajaxtags* *ajaxtags.jar* así como la ayuda de otras bibliotecas de *javascript* que son:

- Prototype 1.4.0: *prototype-1.4.0.js*
- Scriptaculous 1.5.1: *scriptaculous.js*, *builder.js*, *controls.js*, *dragdrop.js*, *effects.js*, *slider.js*
- OverLIBMWS (opcional, para *ajax:callout* solamente): *overlibmws.js*

Teniendo estos requerimientos, se debe de indicar en la directiva *taglib* que se va a utilizar *ajaxtags*, esta se coloca en el archivo *jsp*:

```
<%@ taglib uri="http://ajaxtags.org/tags/ajax" prefix="ajax"%>
```

Esto le indica a la página que hacer cuando se encuentre una etiqueta que empieza con *ajax* en nuestro caso:

---

<sup>3</sup> La biblioteca de *ajaxtags* es un conjunto de etiquetas para *jsps* que simplifican el uso de Ajax en dichas paginas..

<sup>4</sup> JavaServer Pages (JSP) es una tecnología Java que permite generar contenido dinámico para web, en forma de documentos HTML, XML o de otro tipo.

```
<ajax:autocomplete
  source=" "
  target=""
  baseUrl=""
  className=" "
  indicator=""
  minimumCharacters=""
  parser=""/>
```

La etiqueta para autocompletar tiene diferentes atributos, entre otros:

*source*: Id del componente que se desea verificar y autocompletar.

*target*: Id del componente al que se desea hacer alguna modificación.

*baseUrl*: Indica la ruta de la petición. En nuestro caso se utilizó el método *request.getContextPath()* que regresa la ruta de nuestro contexto y al final se le agregó una indicación para saber el *servlet*<sup>5</sup> se usará.

*className*: Función *javascript* que se encarga de enviar los parámetros al servidor.

*indicator*: Id del indicador que muestra al usuario que se está haciendo la llamada *ajax*. En nuestro caso es una imagen que sale al final del campo del formulario.

*minimumCharacters*: Número de caracteres que debe de tener el campo para que se empiece a generar la llamada *ajax*.

---

<sup>5</sup> Un servlet es un objeto en java que recibe una petición y regresa una respuesta basada en esa petición.

*parser*: Función *javascript* que analiza sintácticamente los datos que regresa el servidor. En este caso se utilizó una función que convierte del texto *XML*<sup>6</sup> que se recibe a *HTML* con algunas modificaciones propias.

Dentro del archivo *WEB-INF/web.xml* de la aplicación se puso la referencia al *servlet*, así como el mapeo correspondiente. La etiqueta *servlet* ayuda a indicar que existe un nuevo *servlet*, su nombre y que clase se encarga de ser nuestro *servlet*, en la etiqueta *servlet-mapping* se le indica que ruta debe de ser utilizada en la petición para que utilice dicho *servlet*.

En la parte para autocompletar la etiqueta en el *ajaxtags.js* se crea la llamada *ajax* que llama al *servlet* correspondiente, en este caso un archivo *JAVA* se encarga de revisar en la base de datos para verificar si dicha clave ya existe, si es así se crea un archivo *xml* que informa de la existencia de dicha clave o en otro caso si hubo error, de lo contrario se regresa un archivo *xml* vacío. En nuestro caso se crearon unos analizadores sintácticos que muestran el resultado en una tabla. Se analiza el documento *xml*, se pide la etiqueta padre, al pedir a sus hijos se verifica que éstos existan; el hijo que traería es la indicación de que esa clave ya existe, si esto llega a pasar se ejecuta el método *cambiaBoton* el cual deshabilita el botón de enviar si ya existiera la clave en la base de datos; si se escribe otra letra después o se borra una o varias letras se tiene que verificar que el mínimo número de caracteres que se indican al método de autocompletar se cumpla para hacer otra llamada, lo cual quitaría el aviso de que esa clave ya existe, así como cambia el estado del botón, si todavía hubiera problemas con la nueva clave se haría un nuevo aviso y se bloquearía de nuevo el botón de enviar. Esto sirve exactamente igual tanto para el alumno como para el asesor y el administrador. En el formulario de alumno además se hace una llamada para ver que el número de cuenta sea único, esto se hace con la misma función autocomplete, solo que se utiliza otro *servlet*, uno que solo cheque si ya existe ese número de cuenta y regrese otro *xml* indicando si existe. Se creó un analizador sintáctico diferente para recibir dicha información.

---

<sup>6</sup> Extensible Markup Language (XML) es un metalenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium (W3C) que nos ayuda a definir lenguajes.

## ***Asesores.***

En esta sección se agregó el listado de todos los asesores, anteriormente solo se podían ver los asesores listados por la inicial de su apellido, ahora se puede hacer una lista de todos los asesores del sistema, divididos en grupos conforme a su inicial.

En la parte de editar las asesorías, se agrego una pequeña tabla en la parte superior izquierda de la página que indica el nombre del asesor al que pertenecen estas asesorías así como una liga para enviar un correo electrónico con la funcionalidad descrita anteriormente.

En la carpeta del asesor se arreglo la ruta de las caretas que sale indicada en la parte de Contenido de la carpeta, así como se agrego una liga para poder regresar siempre a la raíz de la carpeta del asesor.

En la parte del formulario se agregó el campo de edición de asesorías, el cual ayuda a permitir que el asesor pueda o no editar, agregar o borrar alguna especificación dentro de las asesorías que imparte, casi convirtiéndolo en un simple alumno, esto se logro añadiendo una nueva columna *iseditor* en la tabla de *users* de la base de datos.

## ***Alumnos***

Así como en la parte de asesor se agregó el listado de todos los asesores, anteriormente solo se podían ver los asesores listados por la inicial de su apellido, ahora se puede hacer una lista de todos los asesores del sistema, divididos en grupos conforme a su inicial.

### ***Buscar Usuario.***

Esta es una nueva sección que se creó en su totalidad para hacer una búsqueda de todos los usuarios que se encuentran dados de alta en el sistema, lo primero que se nota es el campo del formulario donde se escribe la palabra a buscar y después se provee de dos selecciones para buscar, por tipo de usuario y por campo esto da una gran funcionalidad ya que se puede enlazar cualquier tipo de usuario con cualquier campo que se quiera para hacer las búsquedas, esto es se dan 20 diferentes posibilidades para buscar al usuario que se desea.

Esta parte se mejoró poniéndole la inclusión de *AJAX*, para ayudar al usuario y darle diferentes posibilidades al colocar la palabra clave, aquí de nuevo se utilizó la función *autocomplete*, solo que ahora se le agrego a la etiqueta un nuevo atributo llamado *parameters* (parámetros), donde se pasa el tipo de usuario y el campo que se tengan elegidos en ese momento, al ingresar cada letra en la barra de búsqueda un *servlet* se encarga encontrar todas las coincidencias con la palabra actual, se regresa un archivo *xml* en el cual se encuentran las coincidencias que hubo, el analizador sintáctico que se utiliza también es especial para esta búsqueda y presenta una lista de todas las coincidencias que se encontraron, permitiendo elegir alguna de ellas, al oprimir el botón de buscar se presentará una lista de todos los usuarios que hayan coincidido con la clave, conteniendo una liga para enviar un correo electrónico y otra para editar sus datos.

### ***Agregar usuarios.***

Para poder añadir ya sea alumnos o profesores desde una hoja de *Excel*<sup>7</sup> dentro del listado ya sea de profesores o alumnos, en la parte inferior de esta lista se da un campo donde se puede buscar en la máquina del usuario el archivo *excel* correspondiente para ser utilizado, en la parte superior de esta se dejó una liga para proporcionar la información del formato que debe ser utilizado para que la plataforma pueda leer correctamente los datos y poderlos almacenar.

---

<sup>7</sup> Microsoft Excel es una aplicación para manejar hojas de cálculos.

Al elegir *agregar desde hoja de excel*, se redirige a una página para tener una vista previa de los usuarios que serán insertados en la plataforma. Al llegar a esta página se comprueba que el archivo pudo ser subido correctamente a la plataforma para poder leerlo, se hace una llamada al método *writeHtmlPreviewTable* con sus parámetros correspondientes y se imprime la información que regresa este método. A la vez se elimina el *excel* que se subió al sistema para no dejar basura en este. El usuario ve si hubo errores los cuales son marcados para la mejor apreciación de estos y se deja una liga para que el usuario pueda regresar a la página anterior.

Si no hubo errores al dar aceptar se envía esta información a la página de inserción en la cual la información que se generó a través del archivo *excel* es almacenada en la base de datos, se muestra una leyenda de éxito si se insertaron uno o más registros y se muestra una liga para regresar al listado del tipo de usuario correspondiente.

## 4.2.2 Organización Académica.

### *Áreas.*

Se cambio esta sección para que se pudieran editar las áreas que se han creado, en la página de editado se muestran todos las asesorías que existen en esta área, así como el borrado de éstas, si existieran asesorías que estaban dadas de alta en un área que fue borrada, ahora estas asesorías tendrían por área una que tiene por default el sistema que se llama *Sin Área*, esta área se ha agregado directamente al sistema para evitar problemas con los borrados así como cuando no existiera ninguna otra área.

### ***Carreras.***

En esta sección se agregó la opción para editar información correspondiente a las carreras, así como del borrado de éstas, al borrar un carrera del sistema, todas las asesorías que estuvieran relacionadas con esta carrera, serán cambiadas a la carrera *No disponible*, esta carrera se ha agregado por omisión a la configuración del sistema para evitar problemas con los borrados de otras carreras, ya que se podría dar incongruencias dentro del mismo, así como cuando no existiera ninguna otra carrera dentro del sistema.

### ***Asesorías.***

En esta sección se ha agregado el borrado de asesorías, al ser eliminada una asesoría, esta acción se realiza de dos formas, la primera consiste en checar si está dada por un asesor, si así sucediera entonces en realidad no se borrará del sistema, será asignada a un tutor por omisión del sistema, al que solo la parte de administración tiene acceso, esto con la finalidad de no perder todos los contenidos que ya se habían creado en esta asesoría, si no está siendo dada por ningún asesor entonces se borrará por completo del sistema.

Así también se ha dado la posibilidad de editar o borrar las siguientes propiedades de la asesoría: semestre, tipo y carácter, los cuales antes eran estáticos y solo existían los que se dieran de alta al instalar el sistema, ahora se puede manipular libremente para su creación, edición o borrado de acuerdo de las especificaciones del lugar donde se use el sistema. El borrado funciona de la siguiente manera, todas estas propiedades poseen un valor por omisión que es: *Sin semestre*, *Sin tipo*, *Sin carácter*, si se llega a borrar una de estas propiedades y estaba como atributo de alguna asesoría, entonces la asesoría tendrá el valor por omisión de la propiedad.

También se arregló un problema que se tenían con los cuadros para elegir carrera, que tenían problemas cuando no había ninguna carrera o existía solo una.

### ***Asesorías por asesor. (Asignaturas)***

Esta sección fue creada completamente desde cero, se creó con la finalidad de poder ver de una forma más clara todas las asesorías que ya tenían asociado un asesor, se da la posibilidad de poder editar el grupo de esta asignatura. En la página de edición se creó una tabla para poner los datos del asesor y la asesoría, así como una liga para su correo electrónico, aquí se brinda la posibilidad de borrar al alumno de esta asignatura, esto es, no borrarlo del sistema sino solo borrarlo de esta asignatura en particular; el alumno todavía puede ser dado de alta en otras asignaturas, sin embargo se habrán borrado todo sus antecedentes en esta asignatura. Se puede enviar un correo electrónico a todo el grupo, esto se hace juntando todos los correos de los alumnos de la asignatura y colocarlos en el formulario del correo, como en todos los correos del sistema no se permite mover ni el destinatario ni el remitente.

Se puede dar de alta un nuevo alumno al grupo, se mostrará una lista de todos los posibles alumnos que pudieran ser dados de alta en esta asignatura, si no existiera ninguno porque ya todos fueron dados de alta previamente o porque no existe ninguno en el sistema se enviará un mensaje explicativo.

En la parte de ver temario se muestra una lista del temario que fue dado de alta para esta asignatura.

En lo que corresponde a la parte de estadísticas los cambios realizados fueron mínimos, solo corrigiendo algunos aspectos de la sentencia con que se pedía la información a la base de datos para su ordenamiento.

## Capítulo 5. Asesor.

### 5.1 Descripción.

Sin lugar a dudas el rol de asesor es de suma importancia para ALUNAM ya que es capaz de crear, editar, borrar, importar y exportar contenidos educativos. En cada rubro de la plataforma, el asesor tiene alguna función necesaria para el buen desempeño de la misma. A continuación se dará una breve descripción de lo que el usuario como asesor es capaz de realizar en el sistema.

Al iniciar como asesor en el sistema lo primero que se visualiza es el menú lateral izquierdo y una lista de las asesorías que el mismo asesor imparte. Por el lado del menú se encuentran varias ligas que redirecciona ya sea a ventanas emergentes o a la misma ventana principal. *Fig. 5.1*

Asesoría	Horas
para borrar asig	45
para borrar asig	45
Asignatura de caracter 2	56
asignatura2	56
asignatura2	56
asignatura3	56
asignatura 1	56
asignatura 1	56

FIG.- .5. 1 SESIÓN DEL ASESOR.

---

## **Mis Datos**

Este formulario permite cambiar datos personales del usuario como su nombre, apellido y contraseña. Cabe destacar que lo que no se puede modificar es el nombre de usuario (*clave*), esto para preservar la consistencia en los datos de la base.

## **Carpeta**

Este rubro es de bastante utilidad tanto para el sistema como para el usuario, en particular para el asesor, ya que él puede administrar tanto su carpeta personal como las carpetas grupales que corresponden a cada asesoría que está impartiendo.

La funcionalidad de cada carpeta es prácticamente la misma. Al ingresar a alguna de ellas se presenta un listado de los archivos y subcarpetas que ha subido y creado respectivamente el asesor, además de un formulario que permite precisamente realizar esas funcionalidades. Para cada carpeta o archivo existe la posibilidad de renombrar o borrar el elemento.

## **Mis Asesorías**

Como se comentó en un inicio, al iniciar con la sesión del asesor, el usuario notará que se listan las asesorías que éste imparte. Una vez entrando a alguna de ellas es cuando el usuario puede realmente sacar provecho del manejo de contenidos. Mediante el menú superior se puede acceder a las distintas funcionalidades principales relacionadas con los contenidos iniciando con un listado de los temas junto con los subtemas que el asesor haya creado. A continuación se describe a detalle cada elemento que esté disponible desde el menú superior. *Fig. 5.2*



Fig.- 5.2 MENÚ SUPERIOR.

### 5.1.1 *Presentación.*

En esta sección el asesor puede crear o editar la presentación de la asesoría que está impartiendo. Mediante un RTE (*Rich Text Editor*<sup>1</sup>), frecuentemente utilizado en el sistema, el asesor dispone de distintas opciones para crear y actualizar su presentación y finalmente salvarla para que este disponible para los alumnos en cuanto entren a observar el contenido de su asesoría. *Fig. 5.3*

---

<sup>1</sup> Editor de texto que permite crear textos con estilos asociados, a diferencia del texto plano que no contiene ningún tipo de estilo.

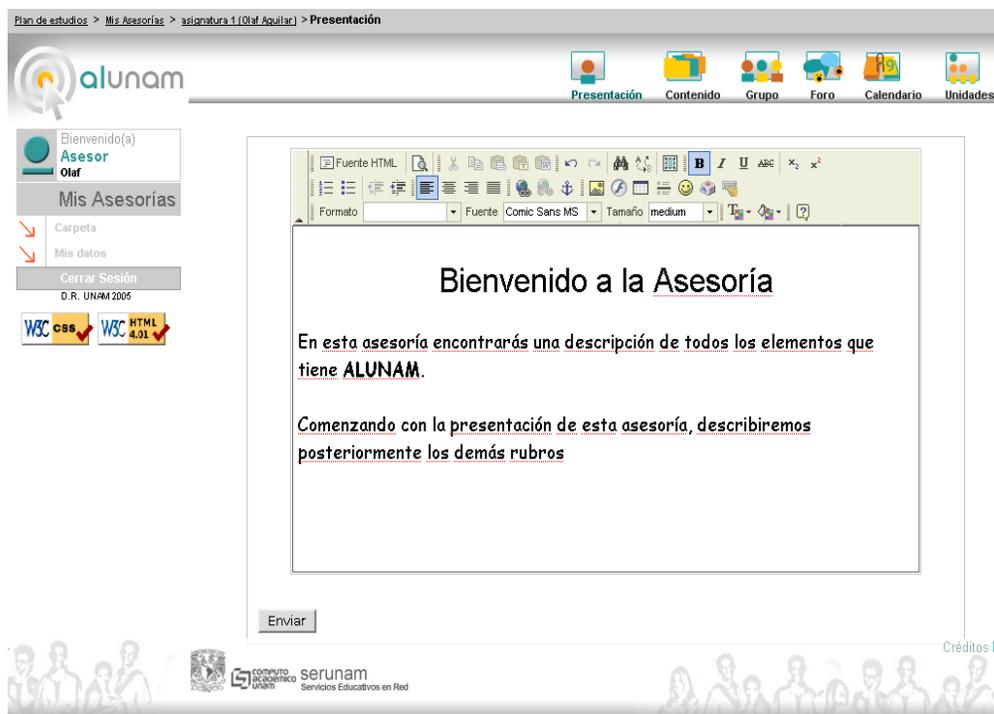


FIG.- 5.3 PRESENTACIÓN.

### 5.1.2 Contenido.

Contenido puede considerarse como la raíz del generador de contenidos. Lo primero que se observa al entrar a esta sección es el listado de los temas y subtemas de la asesoría. Desde esta página se tienen los accesos para crear, editar y borrar los temas y subtemas, además de la posibilidad de darles algún ordenamiento deseado. Los temas son un rubro que alberga tanto a ellos mismos como a los subtemas. Considerado lo anterior, es necesario decir que los subtemas son los verdaderos entes que tienen su contenido, tanto los temas como los subtemas cuentan con la opción de ordenamiento.

Al ingresar a alguno de los subtemas automáticamente se unirá un segundo menú superior particular para el subtema elegido. En este menú se puede iniciar el manejo de las notas, actividades y bibliografía del subtema. *Fig. 5.4*

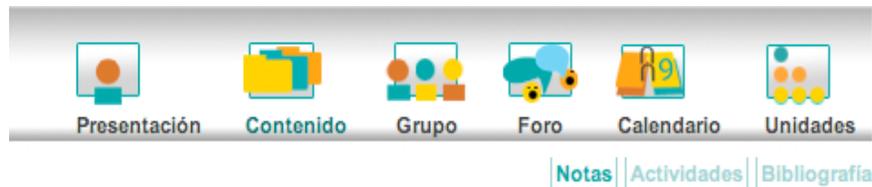


Fig.- 5.4 MENUDE LA SECCIÓN DE CONTENIDOS.

### 5.1.2.1 Notas.

Notas es la primera vista de los contenidos del subtema. Este rubro sin duda alguna ha sido muy útil y a la vez uno al que se le ha agregado bastante funcionalidad desde los inicios de ALUNAM. Las notas en un inicio solo se limitaban a ser creadas mediante un editor HTML que solo funcionaba en el navegador Internet Explorer. Si el asesor utilizaba cualquier otro navegador entonces tenía que crear su nota de manera muy simple y sin detalles. Actualmente, al crear o editar una nota se utiliza un editor de texto enriquecido que corre bajo otros navegadores, no solo Internet Explorer.

A una nota se le puede asociar un título, el texto de la nota y además varias ligas y varios documentos de la carpeta del usuario, esta última funcionalidad estaba ausente de las primeras versiones del sistema. Asimismo las ligas y documentos pueden ser editados, eliminados y ordenados según lo requiera el asesor, además, desde la misma vista de listado de las notas se presenta el acceso para agregar más de estos elementos. Incluso las mismas notas tienen su propio campo de orden el cual puede ser modificado a conveniencia. De igual forma las notas pueden ser eliminadas del subtema.

---

Las notas son parte básica de cualquier contenido educativo ya que almacenan información y conocimiento acerca de los temas a estudiar y ALUNAM permite una sencilla manipulación de ellas.

### *5.1.2.2 Actividades.*

Las actividades en ALUNAM son una parte interesante del sistema. Básicamente son cuestionarios que se crean con el fin de que el alumno los resuelva, esto es registrado en el sistema a detalle; es decir, se preservan datos muy importantes como los intentos que lleva resolviendo la actividad y por supuesto, las respuestas a las preguntas que contestó el alumno. *Fig. 5.5*

Las actividades de un subtema son listadas tomando de nueva cuenta como referencia el campo de orden que también puede ser modificado. Se tienen varias opciones que ayuda al manejo de ellas.

Cuando se elige agregar una actividad se presenta un formulario en el que se asocia los datos como el nombre, las indicaciones, la opción de mostrar refuerzos de las preguntas y el número de aplicaciones, además de que opcionalmente se puede asociar una imagen de fondo y otra de encabezado para la vista de la página que muestra la actividad.

La sección de preguntas de la actividad es la que sirve para crear ya la actividad como tal, es decir, realmente se ocupa del manejo del contenido de una actividad mediante los reactivos y su manipulación. Como este es un rubro de mucho detalle también en la plataforma, se describirá a continuación su uso.

Plan de estudios > Mis Asesorías > nueva4(Rodrigo Saucedo) > Tema: Tema 1 > Subtema: subtema 1 > Actividades

Bienvenido(a)  
**Asesor**  
**Rodrigo**

**Mis Asesorías**

Carpeta

Mis datos

Cerrar Sesión  
D. R. UNAM 2005

W3C CSS W3C HTML 4.01

Presentación Contenido Grupo Foro Calendario Unidades

Notas **Actividades** Bibliografía

Agregar Actividad

**Orden**

1	<b>Actividad1: Actv 1</b>	<b>Indicaciones: Indicación de actv1</b>	Editar	Probar actividad/resolver	Ver Resultados	Borrar	Exportar como Qti
---	-------------------------------	--	--------	------------------------------	-------------------	--------	-------------------------

Preguntas de esta actividad

2	<b>Actividad2: Actv2</b>	<b>Indicaciones: Indicaciones actv2</b>	Editar	Probar actividad/resolver	Ver Resultados	Borrar	Exportar como Qti
---	------------------------------	---	--------	------------------------------	-------------------	--------	-------------------------

Preguntas de esta actividad

Almacenar el orden y reorganizar las actividades

Créditos

serunam  
Servicios Educativos en Red

COMPUTO ACADÉMICO UNAM

FIG.- 5.5 ACTIVIDADES.

### 5.1.2.2.1 Tipos de preguntas.

Se cuenta con un listado de reactivos dependiendo de la actividad que se haya elegido. De igual manera, los reactivos tienen un campo de ordenamiento que el profesor puede cambiar. Los reactivos presentan las opciones básicas de uso como agregar, editar y borrar.

Al seleccionar la opción de agregar pregunta se tiene que llenar el formulario correspondiente. Una pregunta o reactivo consta de su texto y su tipo, además de otros dos atributos dependiendo precisamente del tipo de reactivo que se elija, los cuales son longitud de respuesta y número de respuestas para las preguntas de tipo respuesta abierta y opción múltiple respectivamente.

---

Opcionalmente se le puede asociar una imagen y la posición de la misma con respecto al reactivo.

Con respecto a los tipos, se mencionaron dos de ellos, respuesta abierta y opción múltiple, pero en realidad se cuenta con seis de ellos. Una vez que se elige de qué tipo se quiere crear el reactivo el sistema pedirá ciertas características correspondientes a ese tipo. Los tipos de reactivos son:

### ***Solo pregunta o indicación***

Estos reactivos son útiles al asesor para dar alguna indicación que la actividad requiera y que no haya sido respondida por el alumno.

### ***Respuesta abierta***

Esta es el primer tipo de reactivo en el que el alumno tendrá realmente una interacción más profunda con la actividad. El asesor podrá ingresar un refuerzo para este tipo de pregunta en la que el alumno deberá escribir lo que considera será la respuesta correcta.

### ***Opción múltiple (una sola respuesta correcta)***

Este tipo de reactivo permite mostrar al alumno varias respuestas de las cuales solo una es la correcta. El asesor propone las opciones añadiéndoles además un refuerzo que pueda guiar al alumno en cualquiera de las opciones que elija. Adicionalmente el profesor puede relacionar una puntuación numérica para cada opción.

---

### ***Opción múltiple (varias respuestas correctas)***

Muy similar al caso anterior, solo que el alumno puede elegir varias respuestas que posiblemente sean correctas, según lo establezca el profesor. De igual forma a cada opción se le asocia una respuesta de refuerzo y una puntuación.

### ***Llenar tablas***

Este tipo de reactivos está conformado por varias filas y columnas, con el fin de relacionar cada entrada de éstas (fila por columna). El alumno tendrá que llenar los campos de texto que hacen la función de tablas y sirven de relación de alguna entrada de la fila con alguna otra de la columna. El asesor, como en los casos anteriores, escribe un refuerzo en cada tabla que a su vez el alumno podrá observar una vez terminada su actividad.

### ***Completar frases***

Con este reactivo el asesor crea un texto con la particularidad de que puede encerrar entre llaves, fragmentos del texto de manera que el alumno, cuando vea esta pregunta en la actividad, dichos fragmentos no aparecerán en el texto sino que se desplegarán como un listado para que el alumno los asocie con los huecos que aparecen a lo largo del mismo texto que tiene que completar.

Adicionalmente para cada pregunta del tipo que sea disponible, es posible editar tanto las respuestas como sus refuerzos, además de contar con la opción de eliminar la pregunta deseada.

Una vez construida la actividad con sus respectivas preguntas, el alumno podrá visualizarla y resolverla. El sistema brinda al asesor la posibilidad de probar la actividad y resolverla para que pueda experimentar como el alumno la

---

verá finalmente en web, esto mediante la opción de *Probar actividad/resolver*. Al seleccionar en esta opción una ventana emergente mostrará el total de la actividad indicando el número de intentos restantes, con sus preguntas y el botón para almacenarla.

El asesor puede tener un seguimiento de lo que se ha respondido para cada pregunta de la actividad. Esto se puede observar en la funcionalidad de *Ver resultados*. Esta opción es esencial en un sistema LCMS, de tal manera que se pueda llevar un registro o un seguimiento de los alumnos relacionados con los contenidos de aprendizaje, en este caso, las actividades de los subtemas, donde se lleva un control de las respuestas que se han obtenido por parte de los alumnos.

Finalmente cada actividad puede ser eliminada según lo desee el asesor.

### *5.1.2.3 Bibliografía.*

La opción de bibliografía es utilizada por el asesor para agregar referencias bibliográficas con relación a su asesoría o tema o subtema. Estas bibliografías son también editables y también pueden ser eliminadas por el asesor así como también alterar el orden en el que son presentadas en la página.

### *5.1.3 Grupo.*

Por cada asesoría impartida por un asesor se crea un grupo en el sistema. Los alumnos inscritos en una asesoría en realidad están inscritos en el grupo de la asesoría. El asesor puede manipular dicho grupo mediante esta opción ubicada

en el menú superior. En esta página se encuentran varias opciones que permiten tener control sobre el grupo. *Fig. 5.6*

Nombre	Apellidos	Clave	Correo electrónico	Fecha de nacimiento	Número de Cuenta	Actividades	Eliminar del grupo	Activo / Suspendido	Equipo
alumnoaustra	alumnoaustra	alumnoaustra	alumnoaustra@gmail.com	8 de septiembre de 2008	989898989	Consultar	Baja	<input checked="" type="checkbox"/>	Equipo 1 ▾
alumnohikari	alumnohikari	alumnohikari	alumnohikari@ga.com	8 de septiembre de 2008	121212123	Consultar	Baja	<input checked="" type="checkbox"/>	Equipo 1 ▾
alumnoolaf	alumnoolaf	alumnoolaf	alumnoolaf@gmail.com	8 de septiembre de 2008	090909090	Consultar	Baja	<input checked="" type="checkbox"/>	Equipo 1 ▾

Enviar correo al grupo Enviar correo al equipo: Equipo 1 ▾

FIG.- 5.6 LISTADO DE GRUPO.

Se presenta un listado de los alumnos del grupo. Este listado incluye datos como son el nombre y apellido, clave del alumno, fecha de nacimiento y número de cuenta. Además de los datos anteriores también incluye el correo electrónico del alumno el cual al ser seleccionarlo despliega una ventana emergente con una forma para poder enviar un correo al alumno en cuestión.

Existen otras opciones muy importantes que son también listadas para cada alumno.

La primera de ellas es la relacionada con *consultar actividades*. A diferencia de la opción *ver resultados*, el consultar actividades de cada alumno permite ver de manera individual lo que el alumno ha respondido en cada actividad por cada subtema de manera detallada además de que desde esta pantalla también se puede contactar al alumno vía correo electrónico.

La siguiente opción es *eliminar del grupo*. Al dar de baja al alumno del grupo, éste definitivamente perderá todos los avances que haya tenido en las actividades de la asesoría y con ello el alumno será prácticamente expulsado de ella.

---

Un caso no tan drástico como el anterior es el de la siguiente opción, *Activo/Suspendido*. Cada alumno en la lista tiene una casilla que estará seleccionada para aquellos que están inscritos de manera activa en el grupo; esto quiere decir que el alumno está habilitado para poder resolver las actividades y participar en el foro de la asesoría. En caso contrario, el alumno solo podrá consultar las actividades mas no resolverlas y no tendrá acceso al foro. El asesor puede cambiar el estado de cada alumno en el momento que se requiera a través del botón correspondiente.

La siguiente columna muestra la opción de *equipo*. Cada alumno pertenece por omisión a un equipo, por ejemplo el equipo 1. El asesor tiene la posibilidad de cambiar a los alumnos de equipo. El sistema provee la posibilidad de tener hasta 5 equipos para la asesoría, situación que puede ser modificada desde archivos de configuración de la plataforma. Los equipos son actualizados cuando el asesor oprime el botón correspondiente.

Por último se tiene la opción de enviar correo a todo el grupo o algún equipo en específico, mediante el mismo formulario de correo utilizado en el sistema.

#### *5.1.4 Foro.*

Otro de los rubros con los que el sistema es LCMS es el *foro*. Los foros tienen un canal de comunicación entre los mismos alumnos y el asesor. Este último tiene mayores privilegios mediante algunas opciones, como las fechas de inicio y término, que se ven reflejadas tanto para él como para el alumno, cuando cualquiera de los dos están navegando a través de los temas que se hayan generado. Es importante también mencionar que todos los textos generados para las participaciones del foro son realizados bajo el editor de texto enriquecido que se ha mencionado con anterioridad.

---

Un foro en ALUNAM es creado a la par con cada asesoría que es impartida por algún asesor. Al entrar a la sección del foro, lo primero que se observa es una tabla descriptiva de la situación del mismo. Ésta incluye datos como son el nombre, descripción del foro, fecha de inicio, fecha de término, moderador, número de temas, número de participantes y última participación.

La misma tabla ofrece la opción de edición, en particular, el nombre, la descripción, la fecha de inicio y la fecha de término. Cabe mencionar que una vez pasado la fecha de término, absolutamente nadie puede emitir ninguna otra participación o creación de algún tema en el foro. El moderador, en este caso el asesor, tiene la posibilidad de ampliar la fecha de término del mismo y así poder seguir participando en el foro.

Continuando con la página principal de *foro*, se puede visualizar debajo de la tabla al foro como tal, con temas creados y las respuestas a ellos. Añadidos a esta vista se tiene la opción de ordenamiento, ya sea por autor, fecha o tema, y las de *imprimir foro* y *agregar tema*. Los ordenamientos muestran los temas del foro junto con sus respuestas según el campo de orden que se le indique a la opción.

*Imprimir foro* es una de las opciones bastante útiles para este rubro. Permite imprimir completamente el árbol de las participaciones del foro, mediante una vista sencilla en una ventana emergente, se muestran las participaciones junto con el texto de las mismas. En la ventana emergente se tiene a la vista el ícono u opción que sirve ya como tal para imprimir ese foro por completo. *Fig. 5.7*

Cerrar ventana



**Part 1 asesorolaf asesorolaf dom, sep 28, '08**

Testo de la participacion

**RE: Part 1 asesorolaf asesorolaf dom, sep 28, '08**

Otra respuesta a esa part.

**RE: RE: Part 1 asesorolaf asesorolaf dom, sep 28, '08**

ALUNAM

**RE: Part 1 asesorolaf asesorolaf dom, sep 28, '08**

Esta es una respuesta.

Créditos



FIG.- 5.7 IMPRIMIR FORO.

*Agregar tema*, la opción a un costado de la anterior, permite a los usuarios del foro añadir nuevas discusiones sobre el mismo foro. Al añadir un tema nuevo, el sistema solicita el título del tema o participación y el texto de la participación. Una vez agregado el tema este se mostrará de inmediato en la página principal del foro.

El listado de los temas y participaciones del foro indica tanto el título de la participación o tema, su autor y la fecha de creación.

En el nombre de la participación el sistema envía a otra página del foro en donde se puede ver una tabla con los mismos datos anteriores junto con el texto de la participación. El asesor es capaz de editar la participación con un formulario similar al que sirve para la creación de un tema o participación. El usuario, tanto asesor como alumno, también tendrá la posibilidad de responderla de igual forma con el mismo formulario (anteriormente mencionado). Justo debajo de estas opciones se encuentra la posibilidad de

---

regresar al foro principal, la lista de participaciones, si esta participación sirve como respuesta de alguna superior, se encuentra también la posibilidad de trasladarse a esa participación superior con el mismo tipo de vista y opciones.

Si la participación en la que se encuentra ha tenido respuesta o respuestas, se tiene el caso contrario y se puede observar a las participaciones hijas de la que está siendo visualizada. Estas mismas tienen su opción de poder ser vistas con respecto al campo de ordenamiento que se le indique.

Volviendo a la lista principal de participaciones del foro, para cada tema principal el asesor cuenta con la opción de *borrar tema e imprimir tema*.

Al borrar el tema se elimina la participación de éste así como de las participaciones que lo usaron (participaciones hijas del tema).

El *imprimir tema* funciona de manera muy similar a la opción de *imprimir foro* de la lista principal de participaciones, solo que ésta se aboca solamente al tema al que la opción está siendo referida, es decir, se podrán imprimir el tema y las participaciones que son respuesta al tema del que se haya elegido la opción.

### *5.1.5 Calendario.*

ALUNAM maneja un calendario para el manejo de eventos. Un evento es un aviso en el calendario y tiene asociado una fecha.

Al entrar al calendario se observa el mes actual con sus días y ligas a los meses anteriores o siguientes. Si el usuario se ubica en algún mes que no sea el actual, una liga al presente mes aparecerá. Para cada día de cada mes es posible

---

agregar un evento e inclusive en la pantalla del formulario de creación del evento es posible cambiar la fecha del mismo. Una vez creado el evento éste puede ser editado o eliminado.

Además de los eventos existe otro rubro importante que está relacionado con el calendario: las *unidades de aprendizaje*. De ellas se hablará más adelante, por el momento basta con señalar que cuando las unidades son creadas éstas tienen asociadas una fecha de inicio y una fecha de término. En el calendario de ALUNAM, si existe alguna unidad de aprendizaje que este dentro del periodo que muestra el calendario, entonces se podrá tener acceso a esa unidad en todos los días que se muestren y que se encuentren en dicho periodo.

Si alguna fecha tiene algún evento o alguna unidad de aprendizaje, el número que indica el día de la fecha es habilitado de tal manera que se pueda ver en una ventana emergente todos los eventos de ese día. También es posible ver un evento de manera individual en una ventana emergente al seleccionar directamente dicho evento.

### *5.1.6 Unidades de Aprendizaje.*

Las unidades de aprendizaje son una forma de poder llevar a cabo un seguimiento y estrategia educacional, de tal manera que el alumno pueda tomar esa ruta con los elementos que se han creado en las distintas secciones, ya sea una nota, una actividad, una participación del foro, etc. en el orden que el asesor lo crea conveniente. A cada uno de estos elementos se le conocen como recursos. Para la plataforma los recursos son los distintos contenidos educativos que se han creado o que incluso pueden ser archivos de la carpeta del asesor. Los recursos se insertan a las unidades mediante organizaciones u otros recursos. Las organizaciones son secciones dentro de las unidades de aprendizaje que mantienen parte fundamental del seguimiento que se le desea dar a la unidad además de darle un orden. **Fig. 5.8**

## Recursos

Ver el contenido de esta unidad (Unidad)

- ◊ Unidad 1 | Editar | Agregar Recurso | Borrar | ▼
  -  Part. de Foro | Editar | Agregar Recurso | Borrar | ▼ | Mover a
    -  Nota 1 de subtema 1 como recurso. | Editar | Agregar Recurso | Borrar | Mover a
  - Recurso Ninguno | Editar | Agregar Recurso | Borrar | ▲ | Mover a
    -  Archivo de carpeta | Editar | Agregar Recurso | Borrar | Mover a
- ◊ Org. 2 | Editar | Agregar Recurso | Borrar | ▲ | ▼
  -  Liga UNAM | Editar | Agregar Recurso | Borrar | Mover a
- ◊ Org 3 | Editar | Agregar Recurso | Borrar | ▲
  -  Otro archivo de carpeta | Editar | Agregar Recurso | Borrar | ▼ | Mover a
  -  Part. de Foro 2 | Editar | Agregar Recurso | Borrar | ▲ | Mover a

FIG.- 5. 8 UNIDADES DE APRENDIZAJE.

En esta sección se cuenta con las opciones de ordenamiento del listado principal, el agregar, eliminar y editar de las unidades, además de poder ver ciertas características de cada unidad como su fecha de inicio y término, así como también su descripción. También se encuentran disponibles las opciones para exportar contenidos, exportar contenidos como objetos con contenido compartido (*SCO*) (*Ver Apéndice A*) y editar la publicación de recursos de la unidad. Finalmente, en la parte inferior del listado se encuentra la opción de importado de contenidos para poder crear una unidad que ya haya sido elaborada con anterioridad mediante el estándar *imscp* 1.1.2 o 1.1.3 (*Ver Apéndice A*).

Lo primero que se debe hacer para crear una unidad de aprendizaje es seleccionar la opción de agregar. En esta parte solo se le da un nombre, una descripción, una fecha de inicio y una fecha de término. Una vez creada la unidad lo siguiente que hay que hacer es añadirle recursos mediante la opción de editar recursos. Como se indicó anteriormente las unidades constan de organizaciones. Al crear una unidad de aprendizaje en

ALUNAM se crea por omisión la primera organización con el mismo nombre de la unidad. Esto se puede constatar en la página a la que envía la opción comentada. Desde aquí se tiene acceso a las distintas opciones como son agregar recurso, editar y borrar la organización, si existen otras organizaciones en la unidad, la posibilidad de subir o bajar de nivel la organización completa. Para los recursos se tiene opciones similares las cuales son agregar recurso, editar o borrar el recurso y la de mover el recurso a otro sitio de la unidad. De esta manera se tiene un desplegado similar a un árbol para la unidad en donde hay distintas ramas (organizaciones) y en cada una de ellas se tienen recursos que a su vez albergan otros recursos. En la misma página se cuenta con la opción de poder ver la unidad de aprendizaje tal y cual es, es decir, con los verdaderos contenidos y no referencias a ellos mediante los recursos.

Para crear una organización se toma la opción correspondiente. Mediante ella se le asocia un nombre y una descripción a la nueva organización. La edición funciona de manera análoga para una organización ya existente. **Fig. 5.9**

Cuando una organización es borrada, también borra todos sus recursos hijos.

### Agregar Organización

<b>Título:</b>	<input type="text" value="Org 1"/>
<b>Descripción:</b>	<input type="text" value="Uso del contenido original"/>
<input type="button" value="Enviar"/>	

FIG.- 5.9 AGREGAR/EDITAR ORGANIZACIÓN.

---

Las flechas al costado de las opciones de una organización sirven para poder manipular el orden de las organizaciones. Mediante ellas se puede subir o bajar de nivel a la organización dentro de la misma unidad.

La opción de agregar recurso, permite añadir recursos al mismo nivel dentro de la organización. Al tomar esta opción se presenta un formulario para la creación del nuevo recurso. Aquí se escribe el título del recurso, su descripción y el tipo de recurso que es los cuales pueden ser:

- Ninguno
- Nota
- Tema de discusión (participación del foro)
- Actividad
- Bibliografía
- Liga
- Archivo en la carpeta
- Notas de un subtema

Para cada tipo de recurso se tiene asociada una imagen de referencia que se despliega junto a cada recurso en la vista de las unidades o en la del árbol de la unidad.

La última opción es para importar el contenido (que prácticamente es la misma que se había señalado anteriormente) en el cual el árbol de contenidos se inserta en el nivel en que se está agregando el nuevo recurso.

Una vez agregado un recurso, es posible agregarle varios recursos hijos con el mismo formulario que se detalló anteriormente, además de contar con las opciones de editar y borrar recurso. Otra opción muy interesante es la de *mover a*, que permite mover un recurso de un lugar a otro, es decir, no necesariamente un recurso tiene que estar a cierto nivel de alguna organización, éste puede ser llevado a otro nivel o a otra organización. Esto es indicado por unas pequeñas imágenes intermitentes en el árbol de la unidad.

---

Regresando al listado principal de las unidades de aprendizaje también se tienen las opciones de exportación de contenidos. La primera de ellas, exportar contenidos, permite precisamente exportar la unidad de aprendizaje, es decir, dado el estándar *imscp* 1.1.3 (*Ver Apéndice A*), se toman los recursos y organizaciones de la unidad de tal manera que se empaquetan en un archivo de extensión *zip* que se guarda en la carpeta del asesor. En este archivo empaquetado se encuentran a su vez los archivos de configuración necesarios señalados por el estándar, los cuales permiten el manejo de los contenidos que se encuentran dentro de éste. Esto quiere decir que ALUNAM prepara este paquete para que pueda ser descomprimido y utilizado en alguna otra plataforma que soporte el estándar mencionado y así poder administrar y desplegar dichos contenidos exportados en ese sistema.

Además del exportado de contenidos común, ALUNAM también exporta las unidades de aprendizaje en forma de objetos con contenido compartido (*SCO*). Los *SCO* son elementos que sirven para dar un seguimiento y conexión con algún *LCMS* que soportan el estándar *SCORM* (*Ver Apéndice A*), en este caso en su versión 2004. El exportado de objetos con contenido compartido es muy similar al exportado de contenidos común, de hecho también cumple con el estándar de contenidos empaquetados, solo que le son agregados archivos adicionales que son necesarios para la comunicación del *SCO* con el *LCMS* además de unos cuantos datos más en los archivos de configuración. De igual forma los contenidos van dentro de un archivo extensión *zip* que es almacenado en la carpeta del asesor.

Otra de las opciones que ofrece ALUNAM es la de importar contenidos. Esta se realiza desde el listado principal de las unidades así como también en el momento de agregar recurso. La diferencia radica en que al importarlos desde la página principal, se creará directamente una unidad de aprendizaje nueva que se listará junto con las demás, compartiendo las mismas opciones y comportamiento. En el caso de agregar recurso, el árbol de contenidos se incrustará directamente al nivel en el que está el recurso dentro de la unidad tomando a las organizaciones que vengan en el paquete como recursos padres como tipo ninguno de los verdaderos recursos hijos de esas organizaciones.

Cabe destacar que los contenidos agregados de esta forma se almacenarán como archivos en la carpeta del asesor y serán precisamente de este tipo. Algunos recursos que no tengan realmente contenido son tomados como de tipo ninguno.

---

### 5.1.6.1 Vista de las Unidades de Aprendizaje.

Lo que se describió anteriormente fue prácticamente la construcción y manipulación de los recursos o contenidos de las unidades de aprendizaje. Pero para poder ver el producto terminado existen dos formas. Una es por medio de la *edición de recursos*; desde ahí se tiene la opción de *Ver el contenido de esta unidad*. Esta opción envía una ventana emergente con la unidad desplegando sus organizaciones además de poder acceder a las demás unidades. La otra forma es desde luego, mediante el listado principal de las unidades. Desde aquí seleccionando a la unidad, el sistema lleva a esa misma ventana emergente.

La vista de las unidades desde este punto permite mostrar los datos de la unidad (asesoría, asesor, nombre de la unidad de aprendizaje, fecha de inicio y fecha de término). Se cuenta con un menú lateral que, como se mencionó anteriormente despliega el resto de las unidades de la asesoría, dejando activa la unidad que se haya seleccionado o se haya tomado con la opción de *Ver contenido de esta unidad* además de poder escoger alguna de las otras unidades. Para dicha unidad activa están disponibles sus organizaciones las cuales se pueden desplegar para poder tener acceso a todos los recursos. En la parte central de la ventana emergente se visualiza el recurso activo, ya sea la nota con sus ligas y documentos, la actividad, bibliografía o la participación del foro, además de abrir ahí mismo en una ventana auxiliar los recursos de tipo archivo y en el caso de liga pues solamente se abre otro explorador con esa referencia.

**Fig. 5.10**

Para el caso de notas de un subtema se puede checar cada una de ellas también con sus respectivas ligas y documentos.

Cerrar ventana

Asesoría:	Historia de México	Nombre:	Foro de la asignatura	Moderado por:	asesor1 asesor1.
Asesor:	asesor1 asesor1	Descripción del foro:	--	Número de Temas:	1
Unidad de aprendizaje:	Unidad 1			Participaciones:	3
Inicio:	3 de junio de 2008			Última participación:	3 de Junio de 2008 11:46:46
Fin:	4 de junio de 2010				

Imprimir Participación

Unidades de aprendizaje	
Unidad 1	
Org 1	
Part. de Foro	
Nota 1 de subtema 1 como recurso.	
Recurso Ninguno	
Archivo de carpeta	
Org. 2	
Liga UNAM	
Org 3	
Otro archivo de carpeta	
<b>Part. de Foro 2</b>	

Participación inicial	
Enviado por:	asesor1 asesor1
Fecha de envío:	3 de Junio de 2008 11:45:45
Asunto:	Primer tema Historia de México
Participación:	Con esto se da inicio al foro de Historia de México
<input type="text" value="Responder a esta participación."/>	

### Seguimientos

Ordenar por : Autor | Fecha | Tema

RE: Primer tema Historia de México asesor1 asesor1 mar, jun 3, '08

RE: RE: Primer tema Historia de México asesor1 asesor1 mar, jun 3, '08

Cerrar ventana

FIG.- 5. 10 UNIDADES DE APRENDIZJE EN USO.

---

## *5.2 Aportes Técnicos a la Vista de Asesor.*

### *5.2.1 Presentación.*

Este rubro fue agregado en su totalidad. El requerimiento pedía que cada asesoría impartida por el asesor tuviera una presentación elaborada para los alumnos. Esto se consiguió agregando esta opción en el menú superior y creando una tabla en la base de datos que relacionara el contenido, el asesor y la asesoría correspondiente.

### *5.2.2 Contenidos.*

En esta sección fue integrada la parte de ordenamientos de los temas y subtemas, así como los borrados de éstos.

#### *5.2.2.1 Notas.*

Las notas en ALUNAM han sido fundamentales para la generación de contenidos. Anteriormente ALUNAM soportaba notas muy sencillas y no era fácil incluir o relacionar algunos archivos a ellas. Además se deseaba tener la posibilidad de agregar referencias a ligas en Internet. En un principio esto era sencillo, ya que fundamentalmente era agregar los campos necesarios a la tabla de notas en la base de datos. Con esto se insertaba en los campos correspondientes una liga asociada a la nota y una dirección asociada al servidor que relacionaba un archivo de la carpeta del asesor con esa nota.

---

Sin embargo esto no era suficiente para fines de docencia donde la posibilidad de tener notas relacionadas con más ligas y más archivos era necesaria. Para solucionar esto primero se modificó la base de datos a fin de tener dos tablas. Estas tablas son un ejemplo sencillo de cómo hacer instrucciones de cruzados naturales para hacer consultas mediante el lenguaje de consultas estructurado *SQL*<sup>2</sup>. Las dos tablas son similares con diferencia en un campo que sirve para asociar un nombre a la liga que se crea para la nota y lo que tienen en común es el campo que las asocia con alguna nota en particular.

### 5.2.2.2 *Actividades.*

Las actividades en ALUNAM conllevan una gran cantidad de código, tanto en los que son código fuente escrito en JAVA, como en las mismas páginas de extensión *jsp*. Además varias modificaciones fueron hechas para el correcto funcionamiento de la sección, ya que carecía de revisiones amplias y contaba con varios errores.

Entre los detalles que se le agregaron a la sección de actividades se encuentra el hecho de poder limitar a algún alumno el número de veces que puede contestar una actividad. Este campo le fue agregado al formulario así como también a la tabla correspondiente en la base de datos con un valor por omisión de tres intentos.

El darle mantenimiento a las preguntas fue una tarea ardua dado los distintos tipos de preguntas que existen y los diferentes formularios por los que hay que pasar para completarlos. **Fig. 5.11**

Un error que se cometía de inicio era insertar la tabla de *preguntas* una pregunta en un punto en el cual aún no se le era asociado un refuerzo o se llenaban los formularios siguientes, dependiendo del tipo de pregunta.

---

<sup>2</sup> Es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones sobre las mismas.

---

El hecho de que ya había preguntas en la actividad que quizá no tenían los datos o respuestas que la complementaran y que al entrar a una vista que corresponde a cierto tipo de pregunta pues no existían esos datos y la aplicación enviaba errores de campos nulos o inalcanzables.

Lo anterior ocurría con frecuencia, ya que en cualquier momento el usuario podía llenar el formulario de pregunta, la agregaba y quizá con su navegador o con la barra de seguimiento se trasladaba a otra página que no continuaba con el proceso normal de agregar pregunta de cierto tipo.

Dado que existía la posibilidad de que con el paso del tiempo pudieran existir otros tipos de preguntas, se optó por agregar la pregunta a la base hasta el final del proceso que el usuario ejerce para insertar todo lo relacionado con ella y así asegurar que solo insertarían preguntas que contuvieran los datos necesarios.

Una desventaja de esta solución dados los tiempos de entrega, es que había que pedir estos datos por cada llamada al archivo de extensión *jsp* que los requiriera para al final pasárselos al otro archivo *jsp* que tiene la instrucción de insertarlo en la base de datos.

Aún así se hacen verificaciones pertinentes para que los objetos o campos que se necesiten en alguna vista no sean nulos y poder manejarlos sin que la aplicación interrumpa de manera abrupta su ejecución.

A pesar del mantenimiento que se le daba a las preguntas en casos como el anterior, el probarlas detalladamente arrojaba otros errores que de alguna u otra forma se escapaban. Un ejemplo claro de esto es cuando se inserta una pregunta de tipo relacionar tablas. Una vez creadas las respuestas, como en cualquier otro tipo de pregunta, se pueden editar. La manipulación de las tablas permite borrar en algún caso ya sea una u otra tabla de la relación. En ese caso podía hasta llegar a tener solo columnas o solo renglones, es decir, el sistema se quedaba sin poder relacionar alguna respuesta de algún renglón con alguna columna o viceversa lo que ocasionaba confusión y errores en el sistema.

Para poder resolver este problema primero se hizo un análisis de cómo se guardan los datos de una pregunta de tipo relacionar tablas. Como con varias

respuestas de varias preguntas, se cuenta con una tabla por cada tipo de respuesta que correspondía a cierto tipo de pregunta, en este caso es la tabla *respuestas3*. Según la descripción de esta tabla, cada respuesta de este tipo tiene asociado un eje, además de la relación con la pregunta que le corresponde.

**Texto para las columnas**

**Texto:** 1FIFA

Borrar Respuesta

**Texto:** 2NFL

Borrar Respuesta

**Texto para los renglones**

**Texto:** 1Soccer

Borrar Respuesta

**Texto:** 2Americano

Borrar Respuesta

**Texto:** 3Automovilismo

Borrar Respuesta

Enviar

FIG.- 5.11 INSERCIÓN DE PREGUNTA.

Este eje es un campo en la tabla que tiene por omisión el valor de “x” que precisamente quiere decir que cierta respuesta es un renglón. Cuando correspondía a una columna en la base de datos de inserta el valor de “y”.

---

La forma de resolver este problema es verificar antes de borrar dicho elemento, según sea el caso, después de borrar ese elemento, esto ya sea haciendo una instrucción de conteo (*count*) de *SQL* considerando ya sean *x* o *y*.

Si no pasaban dicha verificación entonces no se eliminaba esa respuesta y se envía el mensaje pertinente, el cual debe de indicar que se necesitan por lo menos un renglón o una columna para el uso correcto de este tipo de pregunta.

Otro de los aspectos importantes son los refuerzos de las respuestas a las preguntas. En un principio las preguntas de opción múltiple eran las únicas que manejaban refuerzo para las respuestas. Se sugirió que todo tipo de preguntas también lo manejaran. Este manejo era sencillo para la mayor parte de los diferentes tipos de preguntas, sin embargo, se enfrentó el problema de cómo guardar los refuerzos de las respuestas de las preguntas tipo relacionar tablas. El planteamiento en realidad era muy similar a la forma de almacenar las respuestas de ALUNAM, esto implicó la creación de una tabla nueva, se tuvo que adaptar esa idea para la funcionalidad de refuerzos en este tipo de pregunta. Básicamente se guarda el refuerzo asociado con el campo *id* de la pregunta junto con las coordenadas *x* y *y* que son las que ubican el refuerzo.

### 5.2.3 Grupo.

Otra área a la que se le aplicaron mejoras fue a la de grupo. Se hicieron cosas muy sencillas como manipular el número de equipos que manejaría, hasta cosas un poco más elaboradas como consultas que involucran más campos o valores como los correos de los alumnos de un equipo.

Un requerimiento que hacía falta a la plataforma era el poder enviar correos a los equipos de un grupo. Para esta parte se creó un archivo de extensión *jsp* que llevará toda la lógica de recuperación de los correos que correspondieran a cierto equipo y luego se incrusta el código que corresponde a este archivo de correo a quien se le pasan los parámetros de *correoDe* y *correoPara*, donde *correoDe* es el correo desde el cual se envían un mensaje y *correoPara* son los correos destino. **Fig. 5.12**

---

Al igual que al equipo, es factible poder enviar correos al grupo o algún alumno en particular.

### Enviar correo al grupo

Para:	alumno1@alumno1correo.com,alumno2@alumno1correo
De:	asesor1@asesor1correo.com
Asunto:	
Mensaje:	

FIG. - 5. 12 ENVIAR CORREO.

### 5.2.4 Foro.

El foro ha tenido varios cambios y sirvió para varias pruebas de tecnología que actualmente son más comunes, por ejemplo, las herramientas para formularios del marco de desarrollo *Struts*. Además de eso, se agregaron funcionalidades nuevas que lo hacen más versátil.

1. Se mejoró la vista de las ramificaciones del foro para hacerlo mucho más amigable con el usuario aunque esto significó dejar un poco de lado que desde la vista principal de foro se pudiera leer el contenido de todas las participaciones.
2. Se añadió la opción de imprimir foro, temas y participación, en esta ocasión, dejando visible los contenidos de las participaciones.
3. Se incluyó el calendario gráfico *jscalendar* el cual se describe en la respectiva sección de este documento. **Fig. 5.13**

**Editar Unidades de aprendizaje**

Título:	Unidad 1		
Descripción:	Uso del contenido original		
Fecha de inicio:	3	Junio	2008
Fecha de término:	4	Junio	2010
<input type="button" value="Enviar"/>			

? Junio, 2008 x

« « Hoy » »

Lun	Mar	Mié	Jue	Vie	Sáb	Dom
26	27	28	29	30	31	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	1	2	3	4	5	6

Seleccionar fecha

FIG. - 5. 13 JSCALENDAR.

### 5.2.5 Calendario.

Calendario tuvo un trabajo tanto gráfico como de funcionamiento. El calendario es generado por el archivo *Calendario4.java* que se encarga de generar código JSP con el contenido y después es incrustado en el archivo de extensión jsp que contiene al calendario. Este archivo tiene métodos que regresan el mes del calendario ya sea con estilos y con ligas para agregar eventos para el profesor. **Fig. 5.14**

Los estilos del calendario no eran fijos y no funcionaban del todo correctamente. Así que se trabajó analizando el código y los diferentes métodos que generan el calendario en el archivo JAVA, con el cual se puede establecer de manera congruente los estilos para los días entre semana, fin de semana y los días que no entraban en el mes que el calendario presenta.

Anterior		Junio de 2008					Siguiente
Domingo	Lunes	Martes	Miércoles	Jueves	Viernes	Sábado	
1 Agregar	2 Agregar	3 Unidad 1 Agregar	4 Unidad 1 Agregar	5 Unidad 1 Agregar	6 Unidad 1 Agregar	7 Unidad 1 Agregar	
8 Unidad 1 Agregar	9 Unidad 1 Agregar	10 Unidad 1 Agregar	11 Unidad 1 Agregar	12 Unidad 1 Agregar	13 Unidad 1 Agregar	14 Unidad 1 Agregar	
15 Unidad 1 Agregar	16 Unidad 1 Agregar	17 Unidad 1 Agregar	18 Unidad 1 Agregar	19 Unidad 1 Agregar	20 Unidad 1 Agregar	21 Unidad 1 Agregar	
22 Unidad 1 Agregar	23 Unidad 1 Agregar	24 Unidad 1 Agregar	25 Unidad 1 Agregar	26 Unidad 1 Agregar	27 Unidad 1 Agregar	28 Unidad 1 Agregar	
29 Unidad 1	30 Unidad 1						

FIG: - 5.14 CALENDARIO.

Como es de suponerse el calendario toma el tiempo del sistema en el que se está ejecutando, es decir para el calendario, la fecha actual será la que presente el servidor en ese momento. Si es, por ejemplo, 30 de marzo de 2007 entonces el calendario se ubicará el mes de marzo teniendo como día actual el 30. Si se deseaba trasladarse al mes de abril entonces la fecha actual para ese calendario era la de ese día mas un mes. Pero si en lugar de trasladarse al mes de abril o cualquier otro mes el usuario lo hacía particularmente a febrero, éste contenía 30 días, dada la codificación original, y si estuviéramos en la fecha de ejemplo y se oprimía la opción a anterior efectivamente trasladaba al mes de febrero con 30 días. Solo se pudo notar este error justo en un día 29, 30 o 31. Para solucionar se investigó como es que se manipulan los objetos relacionados con fechas y se encontró que es posible validar si estas tienen alguna relación con el mes de Febrero y se modificó el código de *Calendario4.java* para que se haga en automático el cambio a la cantidad correcta de número de días que el mes tiene.

**Fig. 5.15**

Anterior		Febrero de 2008 Ir al mes actual				Siguiete	
Domingo	Lunes	Martes	Miércoles	Jueves	Viernes	Sábado	
					1	2	
					Agrega	Agrega	
3	4	5	6	7	8	9	
Agrega	Agrega	Agrega	Agrega	Agrega	Agrega	Agrega	
10	11	12	13	14	15	16	
Agrega	Agrega	Agrega	Agrega	Agrega	Agrega	Agrega	
17	18	19	20	21	22	23	
Agrega	Agrega	Agrega	Agrega	Agrega	Agrega	Agrega	
24	25	26	27	28	29		
Agrega	Agrega	Agrega	Agrega	Agrega	Agrega		

FIG. – 5.15 CALENDARIO EN FEBRERO.

## 5.2.6 Unidades de Aprendizaje.

Las unidades de aprendizaje conllevaron un gran esfuerzo para su funcionamiento en ALUNAM, como se explica a continuación, en la parte administrativa de la unidad y la parte de la vista de la unidad. En principio se hará énfasis en la parte administrativa y su aspecto técnico.

Las unidades de aprendizaje, a nivel del modelo de la base de datos, están organizadas con las siguientes tablas:

**unas.** Fig.5.16

Columna	Tipo	Tabla «public.unas» Modificadores
id	integer	not null default nextval('unas_id_seq'::regclass)
titulo	text	
descripcion	text	
fechai	date	default now()
fechaf	date	default now()
taid	integer	not null
orden	integer	default 1

Índices:  
«unas pkey» PRIMARY KEY, btree (id)

FIG.- 5.16 TABLA UNAS.

La tabla *unas* representa a las unidades de aprendizaje con los siguientes campos:

- *id*: El identificador de la unidad de aprendizaje.
- *titulo*: El nombre de la unidad de aprendizaje.

- *descripcion*: Un texto descriptivo para la unidad de aprendizaje.
- *fechai*: Fecha de inicio.
- *fechaf*: Fecha de término.
- *taid*: El identificador de la asesoría a la que corresponde esta unidad.
- *orden*: Un campo de ordenamiento con respecto a las otras unidades de aprendizaje de esta asesoría.

**organizacion. Fig. 5.17**

Columna	Tipo	Tabla «public.organizacion» Modificadores
id	integer	not null default nextval('organizacion_id_seq'::regclass)
titulo	text	
descripcion	text	
unasid	integer	not null
orden	integer	default nextval(('organizacion_orden_seq'::text)::regclass)

Índices:  
«organizacion\_pkey» PRIMARY KEY, btree (id)

FIG.- 5.17 TABLA ORGANIZACIÓN.

La tabla *organización* representa a las organizaciones que están ligadas a alguna unidad y está compuesta por:

- *id*: El identificador de la organización.
- *titulo*: El nombre de la organización.
- *descripcion*: Texto descriptivo para la organización.
- *unasid*: El identificador de la unidad de aprendizaje a la que esta organización pertenece.
- *orden*: Un campo de ordenamiento con respecto a las otras

organizaciones de la unidad de aprendizaje referida.

*items*. Fig. 5.18

Columna	Tipo	Tabla «public.items» Modificadores
id	integer	not null default nextval('items_id_seq'::regclass)
titulo	text	
descripcion	text	
tiposerv	integer	
hreftext	text	
visible	integer	default 1
fechai	date	default now()
fechaf	date	default now()
parentid	integer	default 0
hijos	integer	default 0
orgid	integer	not null default -1
orden	integer	default nextval(('ordenItems_seq'::text)::regclass)

Índices:  
«items pkey» PRIMARY KEY, btree (id)

FIG.- 5.18 TABLA ITEMS.

Análogamente para los recursos o *items* y su tabla *items* donde se puede obtener la información acerca de qué organización los posee así como también si son hijo de algún *item* en especial:

- *id*: Identificador del recurso.
- *titulo*: Nombre del recurso.
- *descripcion*: Texto descriptivo del recurso.
- *tiposerv*: Tipo del recurso.
- *hreftext*: Ubicación del recurso.

- 
- *visible*: Bandera booleana para saber si es visible o no el recurso.
  - *fechai*: Fecha de inicio.
  - *fechaf*: Fecha de término.
  - *parentid*: Identificador del recurso padre.
  - *hijos*: Bandera booleana que indica si el recurso tiene como hijos a otros recursos.
  - *orgid*: Identificador de la organización a la que pertenece este recurso.
  - *orden*: Campo para ordenar este recurso con respecto a los demás de la organización a la que pertenece.

El listado principal de las unidades carecía de las opciones que se le agregaron con lo cual se le daría una funcionalidad mayor. En esta parte como en muchas otras de ALUNAM se le dio mantenimiento a lo que es el ordenamiento de las unidades, mediante una campo de orden en la tabla de unidades en las base de datos.

Con respecto al formulario para agregar una unidad, éste no lleva más que el nombre, la descripción y las fechas de inicio y término. Se podía elegir una fecha final previa a la fecha de inicio así que se desarrollaron validaciones en los calendarios para que esto no sucediera.

En el aspecto de la organización de la unidad, se encuentra la opción de *Editar la publicación de recursos*. Precisamente en esta vista se contempla el seguimiento que se le da a la unidad y se tiene la opción de poder mover algún elemento (*ítem*) a cualquier posición dentro de la unidad con *Mover a*. Esta parte presentaba pequeños errores, desde que el elemento a mover no se posicionaba en donde se deseaba o se obtenía la opción de poder trasladarlo hacia su mismo lugar, esta lógica de presentar los posibles lugares en los que un elemento puede ser movido se lleva a cabo en una función en el lenguaje *javascript mueveItem*, la cual recibe el identificador del elemento a mover, los

identificadores de los elementos padres, si es que lo tiene, y un conjunto de identificadores de elementos que representa dentro de sus padres adonde puede tener una nueva posición. Esta función muestra una pequeña imagen que indica donde puede ser llevado el elemento y se determina mediante una validación recorriendo el arreglo de lugares que hay en la estructura de la unidad. Después se envía la petición del cambio y el servidor devuelve la misma página con la nueva estructura arbórea que se solicitó. **Fig. 5.19**

### Recursos

- ◊ Org 1 | Editar | Agregar Recurso | Borrar | ▼
  -  Part. de Foro | Editar | Agregar Recurso | Borrar | ▼ | Mover a
    - 
    -  Nota 1 de subtema 1 como recurso. | Editar | Agregar Recurso | Borrar | Mover a
      -
    - Recurso Ninguno | Editar | Agregar Recurso | Borrar | ▲ | Mover a
      - 
      -  Archivo de carpeta | Editar | Agregar Recurso | Borrar | Mover a
        -
- ◊ Org. 2 | Editar | Agregar Recurso | Borrar | ▲ | ▼
  - 
  -  Liga UNAM | Editar | Agregar Recurso | Borrar | Mover a
- ◊ Org 3 | Editar | Agregar Recurso | Borrar | ▲
  - 
  -  Otro archivo de carpeta | Editar | Agregar Recurso | Borrar | ▼ | Mover a
    -
  -  Part. de Foro 2 | Editar | Agregar Recurso | Borrar | ▲ | Mover a
    -

FIG. - 5. 19 VISTA PARA MOVER RECURSOS.

Además de mover los elementos mediante la opción *Mover a* también se puede utilizar las flechas que sirven para subir o bajar de nivel el elemento o toda una organización. A esta parte también se le dio mantenimiento para que las flechas salieran de forma adecuada. El código HTML generado para presentar los elementos de alguna organización se recibe en una función de la clase

*unas.Unas.java*, este código incluye esta parte de incrustar las flechas correspondientes para los elementos. Pero se requería también para las

---

organizaciones. Como el código generado para *imprimir* las organizaciones en el archivo *jsp* están dentro del mismo, se decidió que la parte que añade las flechas para poder subir o bajar de nivel las organizaciones se incluyera también en el mismo archivo *jsp*. Esto le da más flexibilidad al manejo de elementos en algún momento ya que se puede llamar a una u otra función de la clase *JAVA* para que varíe ya sea en los parámetros o en los resultados deseados.

Otra petición más que se hizo fue el hecho de crear en automático una organización al crear una unidad de aprendizaje. Esto se realizó con el simple hecho de ejecutar la instrucción *SQL* para insertar en la tabla de *organizacion* una vez que se ha insertado la unidad de aprendizaje en su tabla. Los datos que se guardan para esta organización son prácticamente los mismos que se guardan para la unidad.

Dentro de las opciones *agregar recurso* o *editar recurso* de las organizaciones y otros recursos, uno elige el tipo de recurso que desea agregar o actualizar. Se solicitó la posibilidad de poder agregar como recurso todas las notas de algún subtema y no solamente una. Esto involucró manejar este tipo nuevo de recurso. Tomando como ejemplo los demás casos se decidió relacionarlo directamente con el identificador correspondiente al subtema agregando un caso más de tipo de recurso, es decir, si el recurso era de tipo *notas de un subtema* entonces en el campo *hreftext* de la tabla *ítems* de la base de datos se escribe y se toma como dato el identificador del subtema.

### *5.2.6.1 Importación de Contenidos.*

---

Gracias al desarrollo de nuevas tecnologías para la educación en línea, ahora se cuenta con algunos estándares para poder compartir los contenidos de un curso, la plataforma ALUNAM a través del estándar IMS Content Packaging (*Ver Apéndice A*) ofrece el intercambio de contenidos de la plataforma, así como el uso de otros contenidos creados en otras plataformas dentro de ALUNAM.

A continuación se explica cómo se logra la importación de contenidos dentro de la plataforma.

La plataforma ALUNAM ya contaba con la parte de unidades de aprendizaje, estas existían de una forma muy simple, la cual solo estaba constituida de una forma para guardar los contenidos existentes en dicha versión de la plataforma de una manera arbórea; es decir, existía un nodo raíz, que podía ser de algún tipo de contenido y este podía tener cuantos hijos se quisiera y subhijos, todo esto se hizo pensando en la posibilidad a futuro de poder exportar e importar contenidos a través del estándar.

Para la importación de contenidos en un principio se pensó en la posibilidad de importar contenidos de la versión 1.1.2 (*Ver Apéndice A*), sin embargo no existen mucha diferencia entre la versión 1.1.2 y 1.1.3, sino que las versión 1.1.2 estaba contenida dentro de la versión 1.1.3, así se llegó al acuerdo de que en el momento de importar un contenido y ver que el archivo *manifest.xml* fuera válido en alguna versión del estándar sería sobre la 1.1.3. Al llegar a este acuerdo se pudo hacer que la plataforma importara contenidos de las versiones 1.1.2 y 1.1.3.

Para importar un contenido en la plataforma, lo que primero se verifica es que el archivo con extensión *zip* que se esté pasando sea un archivo válido esto es que no tenga acentos, espacios en blanco o caracteres extraños, si se está en la posibilidad de descomprimir este archivo entonces se crea un directorio en la carpeta del asesor del mismo nombre que el archivo y se descomprime su contenido dentro de este directorio, esto se hace de esta manera porque el archivo *zip* con el que se exporta en *content package* siempre lleva la misma forma, esto es un archivo *manifest.xml* donde se describe la organización del contenido y sus archivos auxiliares, si no contiene nada el archivo *zip* se enviará

---

a una página que así lo informa, de la misma manera si no se pudo analizar sintácticamente el archivo porque tenía un error o alguna otra causa.

En general se tienen dos formas de importar contenidos en este formato, una es importarlos directamente en una nueva unidad de aprendizaje y la otra es importarlos dentro una unidad de aprendizaje existente, los métodos utilizados en ambas formas son similares, solo que al ser agregados a una unidad que ya existía se tiene que pasar el identificador de la unidad para saber que se debe poner como hijo de ésta en lugar de crear una nueva unidad.

Después de ser descomprimido el archivo, se usarán los métodos de la clase que fue diseñada para analizar sintácticamente el archivo *manifest.xml* la cual es *ManifestParser.java*, dentro de esta clase se definieron todo los métodos necesarios, así como variables estáticas que se usan para las validaciones por ejemplo *IMS\_VAL\_MAL* e *IMS\_VAL\_BIEN* para saber si pasa o no la validación el archivo o *IMS\_PARS\_MAL* e *IMS\_PARS\_BIEN* para saber si hubo algún error o no al analizarlo sintácticamente.

El primer método que se ejecuta es el método *parseDocument* con sus parámetros correspondientes, en caso de insertar una nueva unidad. Lo que se hace es verificar que el archivo *imsmanifest.xml* exista dentro de la carpeta donde fue descomprimida la información, si el archivo existiera se utiliza una clase auxiliar *XmlOptions* que permite cambiar el nombre de espacio del archivo de la versión 1.1.2 a la 1.1.3 esto con el fin de poder validar el archivo *XML* con la versión 1.1.3 por las razones explicadas anteriormente, si el archivo no fuera valido se enviarán los errores a la salida estándar y se regresará el código de error en el archivo. Si no hubo errores se pedirá a la base de datos la información con el último identificador de la tabla de unidades así como el ultimo numero de la columna de orden; a continuación se solicitan todos los recursos que contenga el archivo, los cuales serán guardados en una estructura de mapeo (*hashtable*) que contendrá como llave el identificador y como valor el recurso asociado.

Se insertará en la base de datos la nueva unidad con los datos obtenidos previamente así como del archivo *imsmanifest.xml* y a continuación se llamará al método *insertsOrgs* con sus respectivos parámetros. En el método *insertsOrgs* se solicitan las organizaciones y se guardan en un arreglo, se itera sobre este arreglo pidiendo el siguiente identificador de la tabla de organizaciones para no perder el orden que se tenía previamente en la tabla, a su vez se insertará la nueva organización con los datos previamente obtenidos y los de la nueva

---

organización, antes de terminar la iteración se le pedirá a la organización sus elementos, se itera sobre ellos llamando al método *insertItems* con sus respectivos parámetros. En éste se pedirá a la base de datos el siguiente identificador de la tabla de elementos. Cabe mencionar que se insertan todos los elementos como si fueran de tipo archivo, esto con la finalidad de poder visualizar y acceder a todos ellos no importando su tipo, después se verifica si este elemento tiene un recurso asociado, esto se consigue gracias a la estructura de mapeo que se había llenado al principio con todos los recursos, si tiene recurso se trae. Se verifica si el recurso que se tiene no es nulo, si no lo es se checa de que tipo es, puede ser que no tenga nada asociado, que sea una liga en cuyo caso el campo de la tabla *hrefText* será la liga o que se quede como tipo archivo en cuyo caso *hrefText* será una cadena la cual indica dónde se encuentra ese archivo dentro de la carpeta del usuario, finalmente se inserta al elemento con los datos obtenidos de la base y de éste, se verifica si su identificador padre (*pid*) es 0, entonces es el primer elemento y no tiene padre, de lo contrario se actualiza a su padre y se le asigna un 1 en el campo hijos, así se indica que tiene un hijo.

Por último se solicitan todos los elementos de éste, y se guardan en un arreglo, si este arreglo no es vacío y tiene un tamaño mayor a 0 entonces se itera sobre el arreglo y se llama al método *insertItems* recursivamente<sup>3</sup> para todos los hijos.

En el caso de importar dentro de una Unidad de Aprendizaje que ya existía se llamará al método *parseItemDocument* el cual verifica si existe el archivo *imsmanifest.xml* dentro de la carpeta descomprimida, se cambia el espacio de nombres y se solicitan todos los recursos para que sean insertados en una estructura de mapeo. Se inserta una nueva organización que esté asociada a la presente unidad en el sistema y se llama al método *insertOrgs* para registrar las organizaciones especificadas en el archivo *imsmanifest.xml*. Posteriormente se llama a *insertItems* para insertar los recursos como elementos.

Para cualquier caso el resultado puede ser que el contenido haya sido insertado exitosamente en ALUNAM, si hubo un error al momento de analizar

---

<sup>3</sup> Es la forma en la cual se especifica un proceso basado en su propia definición

---

sintácticamente el archivo *imsmanifest.xml* o bien si hubo un error al momento de validarlo, el sistema enviará dicho error a la página correspondiente.

### *5.2.6.2 Exportado de Unidades de Aprendizaje.*

El exportado de unidades de aprendizaje es quizá uno de los elementos más atractivos de ALUNAM y que lo vuelve un sistema a la vanguardia en lo que educación en línea se refiere ya que muchas plataformas actuales proveen portabilidad de contenidos y mediante ésta opción es posible.

Prácticamente hubo dos procesos o fases para el desarrollo de esta parte. A continuación se describe el desarrollo de la primera.

#### *5.2.6.2.1 Primera Fase.*

Antes que nada comparte con el importador de contenidos el hecho de que se realizó un análisis e investigación exhaustiva acerca del concepto del *IMSGGCCP* (*Ver Apéndice A*) el cual se describe previamente en este documento. Una vez comprendida la idea era momento de utilizarla en la plataforma. Las unidades de aprendizaje eran el rubro ideal para poder llevar a cabo el

---

intercambio de paquetes ya que la estructura organizacional de ellas es bastante similar al modelo de contenido empaquetado (*content packaging*).

Fue entonces que se decidió primero que nada crear el cascaron a nivel interfaz de usuario que permitiera esta acción.

Se agregaron ligas correspondientes a cada unidad de aprendizaje para que se enviara a un *jsp* con la petición de crear el archivo de intercambio de paquetes. Este nuevo *jsp* contiene la lógica de recuperación de parámetros para la creación del paquete como son el identificador de la unidad de aprendizaje, su nombre, el identificador de la asesoría impartida por el tutor, así como también parámetros globales en la aplicación que sirven para el procesamiento de la información que se genera en el momento de la construcción del paquete como son rutas globales del sitio, la ruta de la carpeta del tutor y la ruta de los archivos auxiliares que necesita el archivo *imsmanifest.xml*.

Después se crea un objeto de la clase *unam.ims.cp.XmlManifestExport* que es el que verdaderamente se encarga de todo el proceso de la creación del archivo de intercambio de paquetes.

Posteriormente se verifica si el archivo fue creado y se envía al usuario a la página de éxito correspondiente de donde puede descargar el archivo en formato *zip* o en su defecto a la página con el error de no haberse podido crear el archivo.

La clase que se encarga del trabajo pesado es *unam.ims.cp.XmlManifestExport* es por eso que se describirá a detalle que es lo que exactamente hace.

#### 5.2.6.2.2 *Exportador.*

---

La clase *unam.ims.cp.XmlManifestExport* contiene un conjunto de métodos y variables globales dedicado a la exportación de contenidos de ALUNAM. Cuando se comenzó a diseñarla se notó la necesidad de varios valores que podrían cambiar con el traslado del contexto de un sitio a otro o simplemente con usuarios diferentes. Algunas de estas variables son la ruta a la carpeta del usuario, el identificador de la unidad de aprendizaje, la ruta a donde exportaría el archivo de intercambio de paquetes o la ruta en donde se encuentran algunos archivos auxiliares.

Otro de los factores a destacar es el uso de otras herramientas como *xmlbeans* (Ver Apéndice B) el cual permite un manejo transparente para ver la creación de un archivo *imsmanifest.xml* (Ver Apéndice A) como un objeto. De esta forma se comenzó por crear un objeto de tipo *manifest* al cual se le asigna un encabezado y los datos generales necesarios, como la versión, algunas direcciones de espacios de nombres y su identificador, generado mediante el título de la unidad de aprendizaje. La creación del *manifest* como objeto (de este momento en adelante se le llamará *manifest* al objeto en manipulación) incluye la creación de organizaciones mediante otro método, en este método se agrega al objeto *manifest* tanto el elemento *organizations* como el elemento *resources* los cuales son los mismos vistos como elementos dentro del archivo *imsmanifest.xml*, que son necesarios para la validez del archivo.

Los *organizations* contienen a su vez objetos de tipo *organization* los cuales se mapean a una organización de la unidad de aprendizaje, manteniendo la estructura jerárquica de las organizaciones. Por cada creación de *organization*

que es agregada al conjunto *organizations*, se crean los elementos (*items*) de los *organization* los cuales son los recursos de ALUNAM.

En esta parte de creación de elementos es en donde se hicieron cambios para poder manejar los recursos de una manera más transparente. En un principio lo que se hacía era poder tener la certeza de llevar los recursos, así que la creación

---

de elementos consiste de dos partes principales: la que añade el elemento (*item*) a su correspondiente padre (*organization* u otro elemento) y la parte que copia el recurso físicamente como tal al archivo de intercambio de paquete.

La creación del elemento sobre su padre incluía la localización exacta del recurso y saber en dónde debía ser expuesto dentro del directorio, (que se comprimiría para ser transformado en el archivo de intercambio de paquete) así que, recordando los recursos que se manejan en ALUNAM se especificó que lo que se llevaría serían los recursos de tipo liga, archivo, nota, notas de un subtema y bibliografía. En cuanto a los de tipo ninguno, tema de discusión (participación en un foro) y actividad por el momento no se tomarían en cuenta para exportarlos, solo serían asociados como elementos dentro del *imsmanifest.xml* para mantener la jerarquía.

Conociendo los tipos de recursos que se exportarían, se implementó la lógica para agregar los archivos relacionados con los recursos, tanto copiarlos físicamente como establecer y escribir las rutas correctas dentro de los elementos *resource* del elemento *resources* dentro del *imsmanifest.xml*.

Los tipos *resource* a los que los elementos apuntaban se asociaba el atributo *identifierref* para cada *item* (*identifierref* es el identificador de algún elemento tipo *resource* dentro del elemento *resources*). Para el tipo de recurso liga no había problema alguno ya que solo se agrega a ese *resource* el atributo *href* correspondiente (*href* es una dirección en Internet o una ruta a algún archivo dentro del directorio que representa el archivo de intercambio de paquetes). Para el caso de archivo, nota y bibliografía se crearon métodos auxiliares que se encargarían de particularizar la lógica para cada recurso. Con esto se quiere decir que en caso de archivo se busca directamente en la carpeta del usuario dada la dirección que se encuentra en la base de datos, en el caso de nota y bibliografía

se obtenía el código HTML que está almacenado en la base de datos para crear un archivo físico que representara la nota o la bibliografía. Para el caso de notas de un subtema era llamar tantas veces como fuera necesario al caso de nota, solo que viendo al elemento notas de un subtema como un solo elemento.

---

Aunque se mencionó que se particularizaba la lógica de obtención de datos, en general se tenía la misma forma: se pregunta qué tipo de recurso es, se obtiene un archivo, se añade dicho archivo donde sea necesario (en el directorio que se convertirá en el archivo de intercambio de paquetes), se obtienen sus archivos auxiliares y todo esto se tiene que ver reflejado en el archivo *imsmanifest.xml* con sus elementos *item* y *resource*. Esto dió pauta a un mejoramiento en cuanto diseño posteriormente.

Los problemas y retos a los que se enfrentó durante este proceso fueron en verdad grandes ya que aunque las rutas, generadas al crear los archivos en directorios temporales para posteriormente ser copiados al directorio final, apuntarían a un mismo elemento, tienen que respetar la jerarquía de subdirectorios que representan; no es lo mismo obtener una ruta que se utiliza para copiar un archivo origen a la ruta que se utilizaría para el destino o a la ruta que se tendría que especificar en los elementos de tipo *file* de los *resource* (los elementos de tipo *file*, dentro de *resource* son los archivos auxiliares que necesita el archivo principal indicado por el mismo recurso (*resource*)).

En cada caso en particular se tuvo que verificar cada una de las rutas que se obtendrían. El simple hecho de que no existiera un archivo físico de la nota o de la bibliografía y que se tuviera que generar, implicó un gran esfuerzo ya que se tendrían que copiar con una ruta exacta y localizar todos los elementos auxiliares que se necesitarían y verificar las mismas situaciones para cada uno de ellos. Por ejemplo, la nota o la bibliografía o incluso el archivo (en caso de que se tratara de un archivo *HTML*) tuvieran archivos auxiliares, se necesitaba de una forma de leer ese recurso (analizarlo sintácticamente), encontrar todos los elementos que apuntarían a algún archivo auxiliar y entonces copiarlo decidiendo si se respetaría o no la jerarquía de rutas de dichos archivos auxiliares o solo se pondría a nivel del archivo principal para que éste los localizara de inmediato, y para hacer uso de ellos; además de establecer esa misma ruta para los archivos auxiliares en el atributo *href* de los elementos tipo *file*.

Se hizo uso de una clase JAVA cuya principal función es leer un archivo HTML y devolver un objeto iterador que contiene una lista de los archivos auxiliares del archivo principal. Esta clase llamada *unam.ims.cp.ParserCP.java*, recorre cada etiqueta (*tag*) del archivo HTML y por cada uno de ellas busca que atributo de esa etiqueta hace referencia a algún archivo auxiliar. La ruta a la que se hace referencia se agrega a un lista de la cual se obtiene el mencionado iterador y con el se trabajaba para obtener las rutas, manipularlas para copiar los

---

archivos auxiliares y establecer las mismas en el archivo de intercambio de paquetes, además de escribirlas en el archivo de configuración. Cabe destacar que la creación de esta clase JAVA fue en base a unas bibliotecas de un proyecto llamado *HTML Parser* el cual es de licencia libre (*Ver Apéndice B*).

Con el archivo creado y la lista de recursos auxiliares para el archivo principal se pudo entonces programar la forma en la que se añadirían los recursos.

Inicialmente se había pensado en algunos métodos que se encargasen de todo el trabajo que esto implicaba. Se obtenía la ruta original del archivo auxiliar y se comprobaba la ruta que se escribiría en el archivo de intercambio de paquetes y a su vez en el archivo de configuración. Sin embargo en las primeras versiones de la clase se tenía que hacer una distinción entre saber si era un recurso de tipo archivo; es decir, que éste ya existiera en el servidor, o si era un recurso que se había creado como archivo físico (notas o bibliografía). Las principales diferencias son:

- El usuario es quien generaba su propio archivo y si éste era un archivo HTML entonces dependía de él hacer que el archivo tuviera las rutas, ya sean relativas o absolutas, de manera correcta con respecto a su propia carpeta.
- El usuario puede agregar texto, manipularlo, editarlo y hacer referencia ya sea a imágenes, archivos en Flash o a otros elementos, sin embargo el archivo físico nunca se crea, se guarda el contenido del texto en HTML tal cual en la base de datos y el editor se encarga de darle una ruta absoluta (es decir, con respecto al servidor y con el nombre del contexto) a todos los elementos auxiliares que se usan en el texto HTML original e incluso se

pueden hacer referencias a otros archivos propios del editor, como hojas de estilo para darle cierto formato a la vista del texto y del editor a la vez.

---

Se genera el archivo de nota o notas y bibliografía según el recurso y se hace la distinción dentro de un método que recibía como parámetro el archivo del servidor. Esto funcionó inicialmente pero una vez con más tiempo, se planeó una nueva forma de que los datos en la base de datos no dependan del servidor

Una vez con las rutas correctas se procedía a agregar el elemento de tipo *resource* que es referido por el elemento *item*, al elemento *resources* del objeto *manifest*.

Algo que es válido que ocurra es que varios ítems pueden hacer referencia al mismo *resource* así que esa fue una de las primeras verificaciones que se hicieron para no repetir el elemento en el objeto *manifest*. De no ser el mismo entonces se crea un elemento tipo *fileType* los cuales son elementos hijos de un *resource* y funcionan como los archivos auxiliares. En seguida, todos los elementos tipo *resource* que no hayan sido una dirección web son necesarios como primer *fileType*. El elemento *file* del archivo de intercambio de paquetes, como se habrá leído con anterioridad, cuenta con el atributo *href* en el cual se asigna el valor de la ruta, en donde se encuentra el archivo auxiliar, ruta relativa al archivo de configuración de intercambio de paquetes, así que como valor del *href* del primer *fileType* se tiene el mismo que el de *href* de *resource*. En el archivo creado ya se tenían las rutas de los archivos auxiliares, entonces solo faltaba buscar esos archivos en su ruta original con respecto a la carpeta del usuario y añadirlos a su nuevo destino. Lo que se hizo fue pasarle un parámetro más al método; en este caso, el *resourceType*, para que conforme se generarán las rutas correctas también se añadieran estas mismas en un nuevo *fileType* que se agregaría a ese mismo *resourceType*.

Finalmente el objeto *manifest* se almacenaba y posteriormente se creaba el archivo de intercambio de paquetes, listo para ser usado por el usuario bajo cualquier plataforma que él gustara.

### 5.2.6.2.3 *Uso de patrones de diseño.*

---

La clase JAVA que se describió con anterioridad *unam.ims.cp.XmlManifestExport* era funcional, sin embargo era bastante dependiente de diversos datos que se conocen como *duros* e incluso con código redundante en ciertos momentos. Así entre algunos integrantes del equipo se decidió hacer uso de *patrones de diseño* (Ver Apéndice A).

El patrón de diseño que se usa en ALUNAM es conocido como inyección de dependencia (*dependency injection*). Inyección de dependencia es un patrón de diseño de tipo de comportamiento y funciona de la siguiente forma:

Existe una interfaz o clase en sí que abstrae todo el comportamiento de varias clases que implementan la interfaz o el código deseado. El código que es requerido en alguna situación le es solicitado a esta interfaz o clase, y en realidad son varias subclases las que se responsabilizan del comportamiento de dicho código, en particular, solo la clase que es inyectada es la que se encarga del comportamiento el cual puede variar dependiendo precisamente de la clase que se haya inyectado al momento de implementar la interfaz o la clase abstracta.

**Fig. 5.20**

En esta nueva etapa se solicitó que también se exportaran los recursos de tipo vacío (los que son de tipo *ninguno*) y de tipo actividad. Fue entonces cuando se decidió probar el patrón de inyección de dependencia para diseñar esta parte:

- Se manejarían a los recursos o elementos (*items*) como tales.
- Se regresarían los atributos en que se exportarían, así como su contenido.
- Cada recurso sería responsable de construirse y obtener la información requerida.

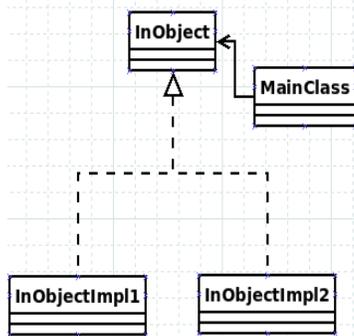


FIG. - 5.20 DEPENDENCY INJECTION. "MAINCLASS" UTILIZA "INOBJECT" CUYA IMPLEMENTACIÓN ES "INYECTADA" SIN SABER COMO O QUE IMPLEMENTACIÓN TIENE.

De esta forma se comenzó por crear una interfaz *resource* llamada *unam.ims.cp.InResource* descrita a continuación. **Fig. 5.21**

```

/*
 * ResourcesCP.java
 *
 * Created on September 13, 2006, 10:21 AM
 *
 * To change this template, choose Tools | Template Manager
 * and open the template in the editor.
 */

package unam.ims.cp;

/**
 *
 * @author rodrigo
 */
public interface InResource {

    String getHtmlAsAsset();

    String getHtmlAsSco();

    String getHtmlTitle();

    String getRouteExp();

    String getResId();

    void setHtmlContent(String htmlContent);

}

```

FIG. - . 5.21 unam.ims.cp.InResource

---

La interfaz *unam.ims.cp.InItem* se crea en tiempo de ejecución del programa, la única clase que la implementa es *unam.ims.cp.resourcesimpl.ItemImpl*, sin embargo cuando se requiera recuperar un *item* de otra forma, solo se desarrolla la implementación de la forma en la que se requiera y el objeto *unam.ims.cp.InItem* se crea como un objeto de la clase inyectada, así se obtendrán los datos deseados de dicho objeto. *Fig. 5.22*

```
/*
 * ItemCP.java
 *
 * Created on September 14, 2006, 11:25 AM
 *
 * To change this template, choose Tools | Template Manager
 * and open the template in the editor.
 */

package unam.ims.cp;

/**
 *
 * @author rodrigo
 */
public interface InItem {

    String getTitle();

    String getSType();

    String getFather();

    String getChild();

    String getHreftext();

}
```

FIG - 5. 22 unam.ims.cp.InItem

La interfaz *unam.ims.cp.InResource* es inicializada dependiendo del tipo de elemento que es enviado (*nota, actividad, bibliografía, etc.*).

---

Lo anterior reduce considerablemente el código de la clase principal ya que éste se traslada a las clases que implementan la interfaz. Se crearon tantas clases que implementan la interfaz como tipos de recurso para los *items*:

- unam.ims.cp.resourcesImpl.ActividadImpl
- unam.ims.cp.resourcesImpl.ArchivoImpl
- unam.ims.cp.resourcesImpl.BibliografíaImpl
- unam.ims.cp.resourcesImpl.LigaImpl
- unam.ims.cp.resourcesImpl.NoneImpl
- unam.ims.cp.resourcesImpl.NotaImpl
- unam.ims.cp.resourcesImpl.Notas\_SubImpl

Cada implementación es responsable de proveer la información que es requerida de la interfaz. En este caso, cada una de ellas busca la información en la base de datos de la forma en que es especificada en estas clases. Esto además de hacer más limpia la implementación de esta parte de ALUNAM la deja más clara.

Lo que ya se tenía era prácticamente la implementación de archivo, bibliografía, liga y nota, así que se tuvo que modificar un poco la implementación de *notas\_subtema* para adaptarla a este nuevo esquema. Se añadió la de tipo *ninguno* como la generación de un simple archivo que contuviera el título de este tipo de recursos. El reto fue implementar la interfaz para cada tipo de actividad.

La clase *unam.ims.cp.resourcesimpl.ActividadImpl* fue diseñada de tal forma que se tuvieran los valores de los datos principales que se obtienen del objeto de la interfaz *unam.ims.cp.InResource*, así se comienza por tener métodos propios de la clase que inicializaran estos valores. En este caso se tiene lo que se le conoce como método constructor, el cual se encarga de obtener datos que le son suministrados y que ayudan a poder recopilar la información de una actividad. Cuando al objeto interfaz se le solicita el valor de algún método en especial, entonces, si la implementación de ese objeto interfaz es de tipo *actividad*, devuelve el valor como *ActividadImpl* lo hace. Por ejemplo, durante la creación de los recursos se necesita conocer el identificador a asignar al objeto *resource* dentro del objeto *resources* del objeto *manifest*, la interfaz cuenta con el método

---

*getResId* el cual regresa justo el dato necesitado; en el caso de la implementación con *ActividadImpl*, se devuelve una cadena que hace referencia precisamente a que es una actividad. Esto quiere decir que cada implementación es responsable de proveer esa información.

Continuando con el desafío que es obtener la información de una actividad, un factor importante fue el saber cómo presentar la información, ya que lo que se ve en ALUNAM como una actividad es una recopilación de todas las preguntas. Así que se creó un archivo extensión *jsp* muy similar al original, que muestra dicha actividad, pero con ciertas diferencias que ayudan a generar de una forma en particular la página, dependiendo de algunos parámetros. Mediante un método auxiliar dentro de la implementación de la actividad, se abre una conexión a esta página, pasándole los datos de inicio para saber qué actividad es la que tiene que mostrar, entonces se recupera todo el contenido de dicha página en forma de una simple cadena de caracteres. Esta cadena es procesada y se usa para generar el archivo físico que sirve para poder exportar la actividad.

Así como la implementación de actividad cuenta con cierto comportamiento para obtener los datos necesarios, todas las implementaciones también lo tiene por lo que solo es cuestión de cambiar o aumentar una línea de código a la clase principal, en este caso *unam.ims.cp.XmlManifestExport*, para manejar más implementaciones o bien para cambiar la forma en que se instancia alguna de ellas.

#### 5.2.6.2.4 *Mejoras al Exportador.*

Gracias a cambios en el diseño como el mencionado con inyección de dependencia, algunas partes de la clase *unam.ims.cp.XmlManifestExport* cambiaron. En general se eliminaron algunos métodos que hacían tarea extra innecesaria. Surgieron algunos otros que al contrario, facilitaban mucho el manejo de los recursos y además se contemplaron los factores para hacer posible que el contenido empaquetado se exportara como *SCO*.

---

En particular se utilizó una clase extra que contiene métodos auxiliares para diversos propósitos los cuales son necesarios para exportar los contenidos y mantener la persistencia de los datos.

Alguno de estos métodos fueron mejorados y otros agregados de otras utilerías. Por ejemplo, contemplando el problema que se tenía con el editor *FCKeditor* al almacenar el contenido, se planeó programar un método que alterará la presentación de ese contenido de forma tal que no dependiese del contexto en el servidor y a su vez otro método que alterará la forma en la que se almacena dicho contenido, esto pensando en que ya se tenían muchos datos con el contenido guardado de la forma original en la que lo hace *FCKeditor* y a su vez guardarlo de la manera en que fuera conveniente.

Con el método que altera la presentación del contenido se pudo tratar a los distintos tipos de recursos como uno solo para poder recuperar los archivos auxiliares. De esta forma ya no se tuvo el problema para hacer la distinción del tipo de recurso, entonces recupera los archivos auxiliares de cierta forma y si era de tipo *nota*, *bibliografía* o algún otro del que no existiera un archivo físico se crean las referencias para que sean fáciles de manipular.

Con ese ahorro se eliminaron métodos que hacían ese tipo de tareas de distinción e incluso ya no fue necesaria una clase auxiliar que se dedicaba a almacenar en un objeto algunas de las rutas, como el lugar donde se escribiría el recurso, el lugar en donde se leería y otros datos que ya son proporcionados por las implementaciones o que son mucho más fáciles de generar gracias a la generalización del manejo de recursos.

Finalmente se agregó al paquete algunos archivos en lenguaje *javascript* que le aplican algunos efectos a la vista de diversos archivos que se exportan, en particular para las actividades. La mayoría de estos archivos provienen del proyecto *scriptaculous* el cual es un conjunto de bibliotecas en *javascript* dedicado a este tipo de situaciones y son de uso libre.

---

### 5.2.6.2.5 *Exportar como SCO.*

El modelo *SCORM*, el cual es utilizado en distintas áreas del aprendizaje en línea, no es soportado en la versión actual de ALUNAM. Sin embargo eso no quiere decir que el usuario no pueda tener sus contenidos en forma de *SCOs*.

La exportación de las unidades de aprendizaje con los contenidos *SCOs* fue una labor sencilla ya que solo se agregaron los atributos y archivos necesarios con el fin de que todo el contenido pudiera generarse de esta manera.

Lo primero que se hizo fue declarar una bandera booleana. Si el usuario exporta la unidad de aprendizaje como *SCO* entonces se evalúa a verdadera y de otra forma falso.

Así como con la creación de un archivo de intercambio debe de cumplir con el estándar, también se debe asegurar que el archivo cumpla con las normas de *SCORM* y para la modificación del objeto *manifest* se debe incluir la configuración necesaria para que sea reconocido como un paquete con contenido en *SCORM*, se copian los archivos en formato JAVASCRIPT que contienen las funciones relacionadas con *SCORM* y se comienza a crear el contenido del recurso como *SCO*.

Para generar el contenido como *SCO* la interfaz *unam.ims.cp.InResource* cuenta con un método en particular que se encarga de regresar el contenido HTML con dicha especialización. La mayoría de las implementaciones pasan por otra clase que utiliza unas herramientas del proyecto ECS de Jakarta. Estas herramientas son de licencia libre y se usan principalmente para crear y alterar el contenido de un texto en HTML. El hecho de modificar el contenido vía la clase que lo envuelve y lo entrega alterado hace ver que el uso de otro patrón de diseño llamado *wrapper*. En si un wrapper convierte la interfaz de una clase en otra interfaz. En este caso se utilizaron los métodos de una interfaz como tal, sin

embargo en la implementación se usa el método de otra que es la que la convierte tal cual en un contenido en forma de SCO. Fig. 5.23, 5.24

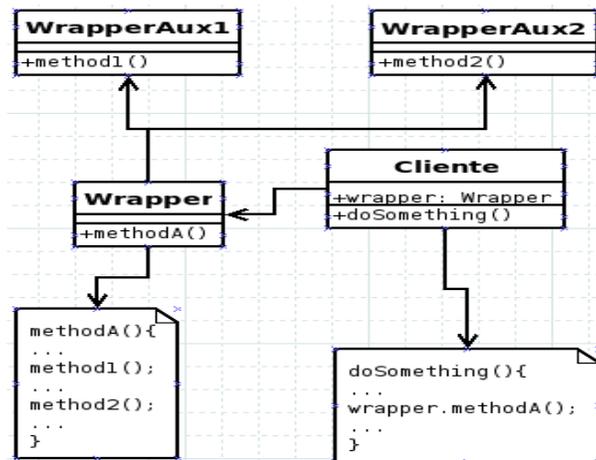


FIG.- 5.23 PATRÓN WRAPPER.

---

```

public static String scoWrapper(String body, boolean sco){
    String content = "";

    Html html = new Html();

    if (sco){
        content = makeSco(html, body);
    }else{
        content = makeHtml(html, body);
    }
    return content;
}

private static String makeSco(Html html, String content) {
    Head head = new Head();
    Body body = new Body();
    body.addElement(content);

    head.addElement(new Script("", "../scormjs/APIWrapper.js", "text/javascript"));
    head.addElement(new Script("", "../scormjs/SCOFunctions.js", "text/javascript"));

    body.setOnLoad("loadPage()");
    body.addAttribute("onbeforeunload", "unloadPage()");
    body.setOnUnload("unloadPage()");

    html.addElement(head);
    html.addElement(body);

    return html.toString();
}

private static String makeHtml(Html html, String content){
    Head head = new Head();
    Body body = new Body();
    body.addElement(content);
    html.addElement(head);
    html.addElement(body);

    return html.toString();
}

```

---

FIG.- 5.24 MÉTODO SCOWRAPPER. DEPENDIENDO DEL RESULTADO DESEADO EL CLIENTE SOLO LLAMA A ESTE MÉTODO CUYA IMPLEMENTACIÓN ES AFECTADA POR OTROS MÉTODOS AUXILIARES.

Finalmente para cada recurso se agrega un objeto *fileType* cuya referencia son los archivos JAVASCRIPT con la funcionalidad de SCORM y se altera al elemento resource para aumentarle el atributo que indica que el elemento principal que se hace referencia es de tipo *SCO*.

## **Capítulo 6. Alumno.**

### *6.1 Descripción.*

El alumno será el usuario que utilizará todos los materiales creados por el asesor con el fin de sacarles el mayor provecho.

Al iniciar una sesión como alumno, lo primero que se puede ver es una página donde se muestran las asesorías en las que se está inscrito, del lado izquierdo se puede observar un menú lateral, el cual se encuentra siempre presente dentro de la sesión. Las opciones que se pueden encontrar son:

#### **Mis asesorías**

Esta liga tiene la finalidad de regresar a la página de listado de asesorías desde cualquier otro punto en el que se este.

#### **Mis Datos**

Este formulario permite cambiar datos personales del usuario como su nombre, apellido y contraseña. Cabe mencionar que lo que no se puede modificar es el *login* o *clave* del usuario, esto para preservar la consistencia en los datos de la base.

## Carpeta

Este rubro es de mucha utilidad tanto para el sistema como para el usuario, ya que aquí se puede subir todo tipo de archivos para su posterior uso. Aquí el alumno puede administrar su carpeta personal, así como su carpeta grupal.

Fig. 6.1

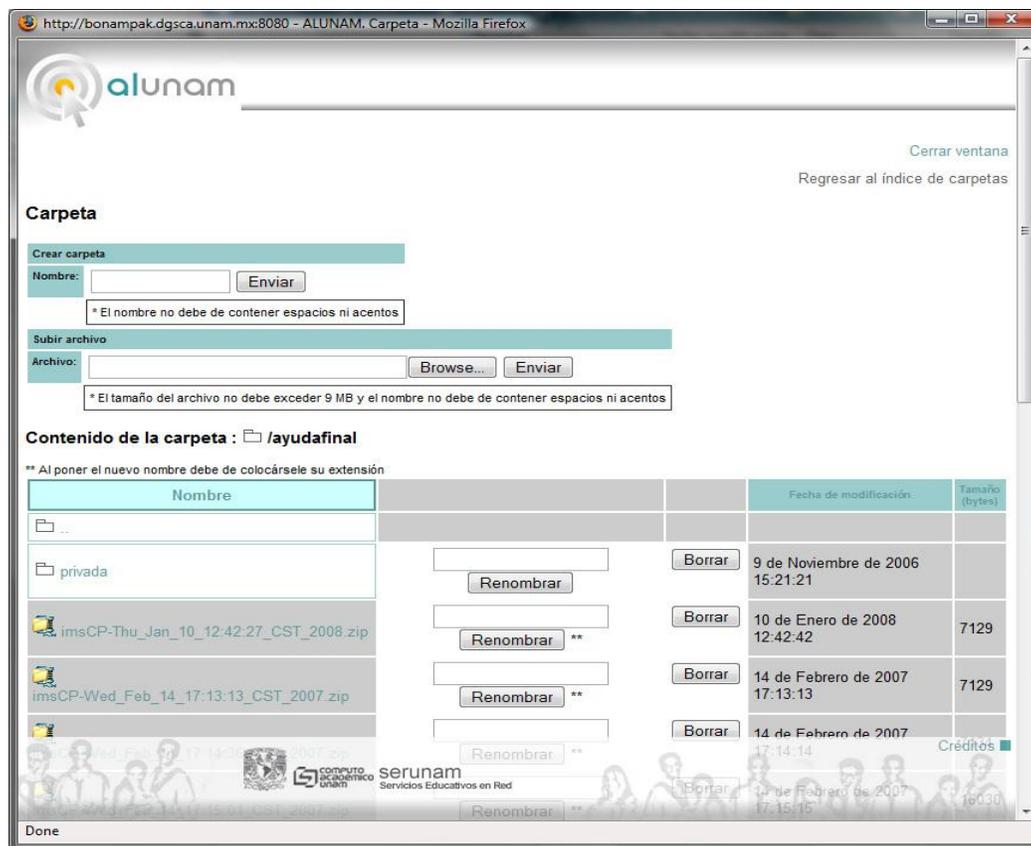


FIG.- 6.1 CARPETA DEL ALUMNO.

La funcionalidad de cada carpeta es prácticamente la misma. Al ingresar a alguna de ellas se obtiene un listado de los archivos que han subido y

subcarpetas que se hayan creado; además de un formulario que permite precisamente realizar esas funcionalidades. Para cada carpeta o archivo existe la posibilidad de renombrar o borrar el elemento.

Al comenzar con la sesión del alumno, el usuario puede notar que se listan las asesorías en las que está inscrito. Mediante el menú superior se puede acceder a las funcionalidades principales relacionadas con los contenidos, donde lo primero que ve el alumno al entrar a una asesoría es la presentación de dicha asesoría.

A continuación se describirá a detalle cada elemento que está disponible desde el menú superior. **Fig 6.2**



FIG.- 6.2 MENÚ DEL ALUMNO.

### *6.1.1 Presentación.*

Esta es la primera página que se muestra al iniciar una asesoría. La presentación como su nombre lo indica permite al alumno conocer el enfoque del curso, así como la idea que tiene el asesor para dar el mismo.

### *6.1.2 Asesor.*

Esta funcionalidad permite abrir una ventana emergente en la cual se puede llenar un formulario para enviar correos directamente al asesor.

### *6.1.3 Contenidos.*

Esta es una de las funcionalidades principales para el alumno. Aquí se pueden visualizar todos los temas y subtemas en los que se encuentra dividida la materia. Al ingresar a alguno de los subtemas automáticamente se agrega un segundo menú superior particular para el subtema elegido<sup>1</sup>. Gracias a este menú se puede acceder a las funcionalidades de notas, actividades y bibliografía del cada subtema. Se pueden listar las notas con las que cuenta dicho subtema, así como los recursos que contengan, ya sean éstos ligas o imágenes que se le definieron.

En actividades, se puede ver una lista de las actividades que se han creado para este subtema. Al escoger una actividad se abre una ventana emergente que muestra el cuestionario de la actividad. Antes de que la actividad sea enviada para que se califique, se le pregunta al alumno si está seguro de querer enviarlo. Cabe destacar que solo se da la posibilidad de contestarlo si el alumno tiene aún oportunidad para responder la actividad y no está dado de baja temporal o definitiva del curso.

En la sección de bibliografía el alumno puede consultar la bibliografía asociada al curso.

---

<sup>1</sup> El mismo submenú que aparece en el rol de Asesor.

#### *6.1.4 Foro.*

Al ingresar a la sección de foro, lo primero que se observa es una tabla descriptiva de la situación del mismo. Esta incluye datos como son el nombre, descripción del foro, fecha de inicio, fecha de término, moderador, número de temas, número de participantes y última participación; una vez expirada la fecha de término, no se podrá realizar ninguna otra participación o creación de algún tema en el foro.

#### *6.1.5 Calendario.*

El calendario sirve para que el asesor le pueda recordar a los alumnos actividades importantes que se deben realizar en cierto día, como un simple recordatorio, o hasta la duración de las unidades de aprendizaje, de esta manera el alumno siempre tendrá presente una lista de las actividades a realizar.

#### *6.1.6 Unidades.*

Al entrar en esta sección se obtiene un listado de todas las unidades de aprendizaje que se han creado para esta asesoría. Al ingresar a alguna de ellas se despliega una ventana emergente en la cual, en la parte izquierda, se muestran datos tanto de la asesoría, asesor y de la unidad de aprendizaje. En la parte inferior se muestra un menú desplegable, que muestra a su vez lo que contiene la unidad elegida, cabe mencionar que esta también permite navegar a través de las otras unidades. Al escoger cada recurso de alguna unidad, se despliega en la parte central derecha su contenido.

La funcionalidad del recurso mostrado es exactamente la misma que si se accederá a él a través del menú superior, por ejemplo, si es el caso de una actividad se observa el número de intentos restantes para resolverla junto con la posibilidad de enviarla si el alumno esta en posibilidad de hacerlo.

## 6.2 *Aportes Técnicos a la Vista del Alumno.*

En esta sección se hablará a detalle de los aspectos técnicos, tecnologías usadas y retos que se sobrepasaron para dejar la sección del alumno dentro de la plataforma ALUNAM para la liberación de la versión 1.0.

En la parte del alumno los cambios que se han realizado van de la mano con los cambios de la parte del asesor. Lo primero que se nota es el cambio en el menú superior de esta vista, el cual a primera instancia consta de la parte de la presentación, esta es una nueva sección que fue implementada para que el alumno pudiera ver cuáles son las características más importantes del curso según el asesor, por supuesto a diferencia del asesor el alumno no puede modificar esta sección.

En la parte que dice asesor y que despliega una ventana emergente para enviar correos al asesor, se quitó la capacidad de poder modificar tanto el destinatario como el remitente, esto fue diseñado así con el fin de contactar al asesor.

En la sección de contenidos, específicamente dentro de la sección de subtemas, se han modificado las presentaciones de las notas con el propósito de mostrar las ligas y documentos adjuntos en una nota. En la parte de actividades, se muestra una lista de todas las actividades que se han creado para esta asesoría, al final de cada actividad se mostrará una liga que permitirá tener acceso a la actividad, esta se modificará de acuerdo a los intentos que tiene el alumno para poder realizarla, si todavía le quedan intentos para realizar la actividad esta liga se llamará *Resolver*, permitirá abrir una ventana emergente que contiene dicha actividad. Como se ha mencionado en la parte de asesor, estas actividades ahora cuentan con un número máximo de intentos para su resolución, por tanto si el alumno ha realizado dicha actividad el total de veces permitidas, ya no se presentará la liga de resolver, sino que ahora aparecerá una liga llamada

*Consultar*, de la misma manera sucederá en el caso de que el alumno sea dado de baja temporalmente de la materia<sup>2</sup>.

La diferencia fundamental entre poder resolver y consultar la actividad, radica visualmente en la existencia de un botón que aparecerá al final de la actividad para poderla guardar.

Cuando se ha seleccionado una actividad, la página que muestra dicha actividad, mostrará en la parte superior derecha una leyenda donde se indica el número de oportunidades que le quedan al alumno para resolver esta actividad, toda esta información es solicitada a la base de datos. En dado caso que ya se hayan ocupado todas sus oportunidades, se mostrará la leyenda que dice “*Ya no le quedan más intentos*” y como se mencionó anteriormente, no se le dará la posibilidad al alumno de enviar dicha actividad, en caso contrario se presentará en la parte inferior de la actividad un botón que dice *Guardar*, este botón solo se presentará en este caso y su funcionalidad llamará al *JSDialog* (ver sección 7.10) para que muestre una ventana en Firefox u otros<sup>3</sup> o un *ALERT* (Internet Explorer) que pregunte si se está seguro de enviar la actividad.

En la sección de foro al igual que en el asesor se ha agregado un ordenamiento con base al autor, fecha, y tema, así como la posibilidad de imprimir el foro en su totalidad o la posibilidad de solo imprimir un tema de discusión, o bien imprimir tema.

En la parte de calendario se corrigieron algunos errores referentes al mes de febrero, por ejemplo para que no muestre más que 28 o 29 días si es año bisiesto.

---

<sup>2</sup> En dicho caso las actividades solo podrán ser consultadas, lo cual se logra consultando en la base de datos el número de intentos totales de la actividad y el número intentos se han realizado.

<sup>3</sup> Konqueror, Safari, Opera, etc.

## Capítulo 7. Mejoras Generales al Sistema.

### 7.1 Correos electrónicos.

La utilización de los correos electrónicos es una de las opciones que más se utiliza en todas las vistas de los roles, por este motivo se tuvo que mejorar esta sección.

Para cambiar la funcionalidad de los correos se tuvo que modificar a los archivos *correo.jsp* que despliega la forma a ser llenada, después pasa por otro programa, *enviarCorreo.jsp* que es el cual se encarga de enviar el correo.

Entre las cosas que se le modificaron al correo se encuentran:

- Que el usuario no puede mover los campos *para:* y *de:*. A menos que así se necesitase.
- Se agregó la opción de enviar un correo oculto para poder informar a una tercera persona (que no sea el administrador), los correos que son enviados desde el sistema, lo cual sirve para llevar un control de los movimientos realizados a través del correo electrónico.

Se optó por hacer que el sistema automáticamente reconociera a quién iba dirigido el mensaje, esto para poder poner el título correcto en la ventana del correo, teniendo que mandar un parámetro extra para este fin, checando dentro de *correo.jsp* a quien iba dirigido.

Al modificar la parte de *Grupo*, se tuvo la necesidad de enviar notificaciones ya sea a un equipo de alumnos o a todo el grupo en general, por tal motivo se decidió darle la capacidad al correo de poder enviar correos electrónicos ya sea a una sola persona o a un grupo de personas; para poder enviar un correo al grupo, en la implementación, se le pasa como parámetro a el archivo *correo.jsp* una lista de todos los correos electrónicos de los alumnos del grupo, luego se analiza sintácticamente dentro del archivo *enviarcorreo.jsp* para crear una dirección en internet de cada uno, para el caso de equipo se utilizó un archivo auxiliar con el fin de poder obtener los correos de los integrantes del equipo esto se hace en el archivo *obtenCorreosEquipo.jsp*, en el cual se incrusta el archivo *correo.jsp* con los parámetros correspondientes del equipo.

## 7.2 Estilos.

Al empezar a trabajar sobre la plataforma ALUNAM, no se contaba con un estándar en la parte de programación. Esto tuvo que cambiar en el momento que surgió la necesidad de hacer diferentes versiones para las necesidades que cada institución tenía, así nació la idea de crear un estándar para la migración a sus respectivas versiones.

Así se creó para los estilos de la plataforma. Con esto cambiaron todos los estilos que tenía la plataforma, además de que se tuvieron que agregar estilos donde no los había. Todo lo anterior con base a un acuerdo<sup>1</sup> al que se llegó donde se tomaría un estilo por cada rubro del sistema, así como de cada tipo de rol, por tipo de tabla, tipo de columna, etc.

---

<sup>1</sup> Este acuerdo se llegó con gracias a la experiencia de nuestro jefe en DGSCA Act. Mario Hernández

El resultado final fue una plataforma totalmente integrada con hojas de estilo, gracias a esto ahora se puede dar una vista completamente diferente a la plataforma solo moviendo las hojas de estilo y no el código dentro de la plataforma, esto asegura una integración más rápida y fácil del sistema con el tipo de interfaz gráfica que la institución requiera.

### *7.3 Validaciones HTML y CSS por la W3C.*

Como se mencionó en el inciso anterior, se hizo un estándar para los estilos del sistema, tomando en cuenta que al finalizar la plataforma se pudiera tener una validación CSS<sup>2</sup> de la W3C<sup>3</sup> lo cual se logró haciendo los cambios pertinentes en base a los requerimientos de la W3C. Así mismo se hicieron cambios para el código HTML, se cambiaron, agregaron y corrigieron todas las etiquetas en HTML de la plataforma, gracias a esto el sistema ahora también es válido en la versión HTML 4.01 Transaccional según la W3C.

### *7.4 Instalación.*

Al principio la parte de la instalación de la plataforma era nula, lo único que existía eran archivos de procesamiento de lotes para la base de datos, los cuales tenían que ser pasados directamente al manejador de base de datos y el código compilado se tenía que colocar dentro de un servidor de aplicaciones que

---

<sup>2</sup> Las hojas de estilo en cascada (*Cascading Style Sheets*) son un lenguaje formal usado para definir la presentación de un documento estructurado escrito en HTML o XML

<sup>3</sup> El World Wide Web Consortium, abreviado W3C, es un consorcio internacional que produce estándares para la World Wide Web

implementara las especificaciones *Java Servlet*<sup>4</sup> y *Java Server Pages*<sup>5</sup> de Sun *Microsystems*, en nuestro caso el desarrollo siempre se encaminó a usar la base de datos *PostgreSQL* y como contenedor de archivos *servlets* se usa *Apache Tomcat*<sup>6</sup>.

Al utilizarse la plataforma, se vio la necesidad de crear una nueva forma de instalación de la plataforma, se optó por utilizar la herramienta *ANT*<sup>7</sup>, con la cual se pudo crear la base de datos preguntando a la persona que lo vaya a instalar solo los datos necesarios para la conexión con esta), lo anterior, lo anterior se consiguió con unas tareas especializadas de ANT, se creó un archivo tipo *war*<sup>8</sup> del proyecto, el cual es descomprimido automáticamente por el servidor de aplicaciones.

Se creó un archivo *LEEME* donde se indican los pasos a seguir para la instalación del sistema.

---

<sup>4</sup> Son archivos Java que permiten añadir contenido dinámico a un servidor web.

<sup>5</sup> Es una tecnología Java que permite generar contenido dinámico en base a una solicitud web.

<sup>6</sup> Contenedor de servlets y servidor web.

<sup>7</sup> Herramienta para el tener un proceso automático de construcción de software

<sup>8</sup> Web Application Archive es un archivo que se utiliza para distribuir una colección de JavaServer Pages, servlets, clases Java, archivos XML, bibliotecas de etiquetas y páginas HTML.

## 7.5 Migración de datos.

Otro problema que surgió, fue la necesidad de cambiar un sistema que ya estaba siendo utilizado en algún servidor a otro lugar, esto provoca que la información que se había almacenado fuera difícil de trasladar a la nueva localización del sistema o en ocasiones solo se necesitaba cierta información de la base. Viendo esto para realizar las migraciones más fácilmente o restaurar información que se necesitara se crearon programas especiales para poder migrar los datos a otra base de datos solo corriendo un programa llamado `ClonALunamDBO.java` el cual se encargaba de copiar todos los datos de una base a otra, sin embargo con el aumento de capacidades del sistema se fue agrandado o cambiando la base de datos, por este motivo se dio constante mantenimiento a este programa para que se mantuviera funcionando tanto con base de datos con vieja estructura como con las que tienen los nuevos cambios.

## 7.6 Etiquetas.

Cada página de ALUNAM en un inicio contaba con los textos de manera fija; es decir, si se necesitaba cambiar algún texto que se muestra en pantalla se tenía que hacer directamente en la página. Este tipo de mantenimiento es muy costoso tanto en tiempo como en esfuerzo.

Gracias a las herramientas que provee *Struts* es posible aligerar ese problema y esta fue una de las primeras tareas encomendadas. Se identificaron todos los lugares en las páginas *jsp* en las que se despliega texto fijo (nada de información que se genere dinámicamente). Una vez hecho esto se utilizó la etiqueta *bean* de *Struts* con el atributo *key* (*llave*), con el valor de *key* se encuentra en el archivo de configuración con extensión *.properties* el cual está indicado en la configuración de *Struts*. La finalidad principal de esta técnica es poder utilizar esa misma llave en cualquier otra página que requiera la misma palabra.

Por ejemplo, en la sección de *notas* las palabras *nota* o *notas* son usadas en varias páginas y cambiarlas en cada una de ellas, si es requerido, así sería más sencillo si solo se modifica el valor de la *llave: nota.label* en el archivo de configuración y de forma automática se viera reflejado.

Otra de las ventajas que tiene esta herramienta es la posibilidad de internalización del sitio ya que resulta mucho más sencillo solo traducir los valores de un archivo de configuración, agregarlos a otro y *Struts* por medio de la configuración regional del navegador se mostrarán los cambios de manera automática.

Una vez que se había modificado en todas las páginas la forma de llamar a los textos fijos, las versiones de ALUNAM siguientes requirieron de otro aspecto. El manejar el archivo de configuración era correcto; sin embargo se optó por clasificar las etiquetas en secciones. En un inicio la creación de etiquetas estaba determinada de la siguiente forma: si ya existía una palabra etiquetada entonces no se volvía a etiquetar. Sin embargo con la clasificación de las etiquetas por secciones se logró una mejor organización de todos los textos en el sistema. Esto permitió que por ejemplo la palabra “asesoría” fuera utilizada en varias secciones sin embargo, el desarrollador o diseñador no espera la palabra “asesoría” en el archivo de configuración sino que esta seccionado tanto por asignaturas como por asesorías (sección de etiquetas para asignaturas y sección de etiquetas para asesorías), dos rubros distintos pero relacionados cuyos valores actuales en ambas secciones tienen la palabra *asesoría* pero que si en algún momento se desea hacer la distinción entre los dos rubros solo es cuestión de cambiar el valor en la sección de etiquetado del que se requiera editar.

## *7.7 Barra de seguimiento.*

Ubicada en la parte superior izquierda de las páginas, debajo del encabezado, la barra de seguimiento es una barra de navegación que facilita y brinda mucho

mayor agilidad en el proceso de acceso a las secciones en ALUNAM que el usuario haya visitado. Por ejemplo, si el usuario se encuentra leyendo alguna nota de algún subtema notará que en la barra de seguimiento se puede elegir alguna de las secciones previas al acceso de esa nota, como es el subtema, tema, la asesoría, etc.

La barra de seguimiento es manejada por un archivo de extensión *jsp* que es insertado en la página por medio de las herramientas *tiles*<sup>9</sup> de *Struts*, tal como ocurre con la organización del resto de los elementos principales de las páginas en ALUNAM.

Cabe señalar que el trabajo que se realizó aquí fue de mantenimiento, es decir, en un inicio la barra no tenía las ligas a los sitios previos totalmente bien referidas, no tenía absolutamente ninguna etiqueta y carecía de algunos estilos para que pudiera resaltar lo suficiente como para que el usuario notara su presencia. Lo anterior fue modificado de manera tal que estos detalles fueran mejorados y útiles, cambiando cada nombre de nivel con etiquetas y asignando una clase de estilos único.

## 7.8 FCKeditor.

Como se comentó en la parte técnica de ALUNAM, la plataforma utiliza *JAVA* para varias funcionalidades del lado del servidor. Las páginas de ALUNAM al ser archivos de extensión *JSP* tienen ciertas propiedades como cualquier otro tipo de página que se ejecuta en la web.

Para este caso, se deseaba que el asesor en ALUNAM utilizara un editor *HTML* que se ejecutara en varios navegadores ya que en las primeras versiones del sistema se ejecutaba con algunas limitaciones, en Internet Explorer.

---

<sup>9</sup> Es un sistema de plantillas el cual permite crear vistas comunes en aplicaciones web.

FCKeditor es el nombre del editor HTML que se probó para ser incluido en la plataforma. Este software es flexible para poder ser añadido a algún proyecto que reúna ciertas características. El editor ofrece varias de las funcionalidades de editores de texto de escritorio pero a través de la web. No es necesario instalar nada del lado del cliente<sup>10</sup> para que el editor funcione correctamente. Las principales características del editor son:

- Compatibilidad con varios navegadores.
- Posibilidad de generar el código en *XHTML*
- Soporte de archivos de estilo.
- Funcionalidades de editores de escritorio (formato de texto y fuente, inserción de archivos, tablas, cortar y pegar texto, pegar texto desde *Word*, etc.).
- Soporte para *skins* (para dar otros aspectos al editor).
- Multilinguaje.
- Edición del código generado.
- Rápido y ligero.

En principio al descargar la versión compatible con archivos *jsp*, se contaba con los archivos fuente. Esto es útil dado que más adelante se detallará que se modificaron un par de detalles para un uso acorde al de nuestra plataforma. La principal diferencia para poder hacer uso del editor en alguna página *jsp* es el uso de una etiqueta especial. Dicha etiqueta habilita a los archivos de configuración, uno de ellos con extensión *.tld*<sup>11</sup> que debe ser incluido dentro de la carpeta *WEB-INF* del contexto. Dados los parámetros indicados en el archivo *.tld* a los que se le ajustan valores, el editor es hasta cierto punto fácil de manipular y configurar para afectar campos de texto o rutas en el servidor.

Con *FCKeditor* se realizó un arduo trabajo de investigación sobre el código del editor para alterar su funcionamiento. Principalmente, se cuenta con un archivo *javascript* de configuración (*fckconfig.js*) en el que se pueden cambiar

---

<sup>10</sup> Código que se ejecuta en la máquina del usuario en lugar de la máquina servidor.

<sup>11</sup> Documento XML que mapea una etiqueta JSP a su archivo asociado o manejador.

varias propiedades generales del editor como el lenguaje o los elementos de la barra de herramientas, el programador puede elegir alguna de estas versiones o crear su propia versión de elementos que desee que se encuentren en la barra.

Además de los archivos *javascripts* se investigó sobre las opciones que ofrece el *.tld* para el manejo del *tag*. En particular se deseaba que el asesor pudiera relacionar el texto que estaba editando con algún archivo de su carpeta. *FCKeditor* relacionaba la opción de navegar por el servidor de manera que el usuario podía tener vista y acceso a archivos que no le correspondían. Al revisar detalladamente el código de distintos archivos se identificó que era lo que se debía modificar y además indicarle a la etiqueta del *FCKeditor* lo que se le llama el *conector*. Este conector sirve para relacionar desde dónde *FCKeditor* puede navegar archivos del servidor. Esto claramente sirvió para que el asesor solo pudiera relacionar contenidos con archivos que tenga en su carpeta y no buscar en ningún otro lugar.

Precisamente al comentar sobre la relación de archivos del asesor con el contenido que se genera, *FCKeditor* provocó el inicio de otro pequeño problema que sale a relucir junto con las unidades de aprendizaje, El cual tiene que ver con la manera en que *el editor* genera el código *HTML* cuando se relacionan los contenidos de la carpeta con el texto generado.

El conector como otros archivos fueron alterados a manera de tener mayor control y seguridad para la plataforma, esto incluye desde vistas de las ventanas que surgían al tomar una opción del editor hasta el cambio de funcionalidad completa de este, como subir archivos desde la computadora del cliente al servidor. Este tipo de decisiones se tomaron por razones de estabilidad y coherencia en la plataforma al hacer ciertas validaciones.

*FCKeditor* demostró ser una herramienta bastante útil para la plataforma. La plataforma puede generar contenidos con mayor vista y más completos gracias a *FCKeditor* el cual es considerado como uno de los editores *HTML* más versátiles de la web.

## 7.9 *JSCalendar*.

Este calendario tiene la finalidad de otorgar una interfaz amigable al usuario para seleccionar fechas. Utilizando *DHTML*<sup>12</sup> se modifica dinámicamente la página HTML a través de un programa y este calendario es insertado. Añadir este calendario a la página es muy sencillo usando *Javascript*. Solo se inserta un fragmento de código en este lenguaje y el calendario es presentado.

En ALUNAM se identificaron las distintas situaciones en las que se manejan las fechas de manera que se pudiera insertar un calendario adecuado a las necesidades del usuario. Esto puede ser el simple hecho de ayudar con el calendario para escoger una fecha o el coordinar dos fechas simultáneas con un elemento *JSCalendar* para cada una. Generalmente, en este último caso, una de las fechas es la fecha de inicio de algún rubro en especial y la otra es la fecha de término del mismo, las fechas de inicio y término de las Unidades de Aprendizaje son un ejemplo de esto.

Después de identificar el uso de *JSCalendar*, se configuró para que tuviera la funcionalidad correcta y para que estuviera ligado con algún otro elemento con el que también se pudiera elegir la fecha, en el caso de ALUNAM se usaron cajas de selección de día, mes y año. Como se mencionó, el añadirlo es sencillo pero en el caso de coordinar las dos fechas de inicio y término, de manera que una no sobrepase a la otra, esto implica una lógica más elaborada dado que las fechas elegidas en el *JSCalendar* afectan directamente a elementos en la página con los cuales también es posible editarlas y viceversa. Finalmente se utiliza herramientas de validación para que el usuario este enterado de que la fecha de inicio y a su vez que no puede sobrepasar la fecha de término, caso que es evitado utilizando *JSCalendar* ya que las fechas que no deben ser elegibles se bloquean, dependiendo del valor que se tenga escogido en uno u otro *JSCalendar*.

---

<sup>12</sup> Combinación entre javascript, CSS, HTML y DOM

## 7.10 *JSDialog*.

Otro nuevo aporte que se le agregó a la plataforma fue el uso de una nueva forma de alertas llamada *JSDialog*, se optó por usar unas herramientas en *javascript* llamadas *DOJO*, sin embargo esto generó problemas de compatibilidad con las funciones ya hechas con *AjaxTags*, debido a unas bibliotecas que ambos compartían y sobrescribían variables; se optó entonces por utilizar una nueva herramienta llamada *JSDialog* la cual a través de unas bibliotecas en *javascript* daba la capacidad de crear un cuadro de texto con opciones, bloqueando la pantalla principal, esto es como una alerta común de *javascript* pero con más estilo.

Al incorporar esta nueva herramienta se tuvieron que crear muchas nuevas funciones para que pudieran interactuar de manera correcta con las opciones necesarias, lo cual creo nuevas funciones para el llamado de los mensajes, el chequeo de tipo de orden (actividad, item, eliminación, ordenación, etc.), ahora todas las anteriores necesitaban validarse, lo cual se hacían llamando al *JSDialog* con el respectivo mensaje sacado de un archivo de propiedades (*Ver sección 7.6*) y el tipo del que se trataba.

También se modificó el código para eliminar el uso de ciertas teclas, ya que con ellas se podía acceder a la página principal sin necesidad de quitar el mensaje del *JSDialog*.

Un grave problema fue la compatibilidad con el navegador Internet Explorer, ya que no reconocía las funciones del *JSDialog*, por tal motivo se modificó el código para preguntar primero al servidor que tipo de navegador se estaba utilizando, si era Internet Explorer entonces se utilizaba las alertas normales de *javascript*, en otro caso el *JSDialog*.

## **Conclusiones.**

Los sistemas *LMS* son herramientas que muestran su flexibilidad en cuanto a la formación docente y presentan la gran ventaja de eliminar las distancias geográficas. El ritmo y el tiempo de aprendizaje van de acuerdo tanto con las necesidades del profesor como del alumno, además de ofrecer retroalimentación entre ellos de manera constante.

Al constatar la funcionalidad que ha dado esta plataforma a los diferentes sectores que la utilizan, nos llena de orgullo el saber que algo en lo que trabajamos sea aprovechado por la UNAM y en diversas de sus dependencias, como la Dirección General de Servicios de Cómputo Académico, la Facultad de Filosofía y Letras, la Maestría en Docencia para la Educación Media Superior o en lugares como la Universidad del Claustro de Sor Juana, Aeropuertos y Servicios Auxiliares y la Academia Mexicana de Derechos Humanos.

Con ALUNAM aprendimos muchas cosas que se deben hacer en un proyecto y otras que no son convenientes. Por ejemplo, se debe evitar desarrollar sin poseer un diseño previo, que contemplen estándares de documentación y de codificación para facilitar su comprensión y respetar esos diseños para facilitar el desarrollo y mantenimiento. Que la planeación de las pruebas ayude a demostrar que la funcionalidad sigue íntegra a través de los cambios. Conforme se avanzó en el desarrollo del sistema se cambió la dinámica de trabajo a realizar este tipo de prácticas. Esta experiencia nos está ayudando en el desarrollo de nuevos proyectos en nuestra vida profesional.

Al realizar este documento, nos dimos cuenta de los cambios que se hicieron a ALUNAM en los que tuvimos participación y ayudaron a hacer el sistema más útil y funcional. La recopilación y explicación de los aportes técnicos al sistema enriquecieron nuestras habilidades como programadores y muestran la funcionalidad y el alcance que tiene la plataforma. Sin embargo, lo más importante es la enseñanza que nos dejó como desarrolladores

Esperamos que éste trabajo contribuya al desarrollo de futuros proyectos de educación en línea y al mantenimiento del mismo sistema para que otras personas puedan actualizarlo y mejorarlo.

## Apéndice A.

### *Especificación de contenido empaquetado del IMSGLC (IMSGLC Content Packaging Specification)*

Dentro del mundo de la tecnología de aprendizaje existe un consorcio internacional que se dedica a desarrollar y promover la adopción de especificaciones técnicas abiertas para lograr la interoperabilidad entre las mencionadas tecnologías. Este consorcio es el *IMS Global Learning Consortium* del que muchas de sus especificaciones se han vuelto estándares para el manejo y la entrega de productos y servicios de aprendizaje.

De las especificaciones dadas por esta asociación ALUNAM utiliza la de contenido empaquetado (*Content Packaging*).

Lo que se define y describe son las estructuras de datos que nos ayudan a proveer la interoperabilidad entre el contenido basado en Internet con las herramientas de creación de contenido, *LMS* y herramientas de ejecución. Este conjunto de estructuras son las que nos ayudan a intercambiar contenido y permiten estandarizar los datos que sirven como liga a otros, lo cual también permite que los desarrolladores de software y la gente que lo implementa cree distintos materiales que interoperan con otras herramientas o *LMS's* o ambientes de ejecución desarrollados de forma independiente.

El paquete consiste principalmente de dos elementos: un archivo en formato *XML* que describe la organización y recursos del contenido, y precisamente los archivos físicos descritos por el *XML*. El archivo *XML* es conocido como el archivo manifiesto *IMS* (*imsmanifest.xml*). Una vez que el paquete se comprime, en formato *zip* por ejemplo, para tenerlo en un solo archivo que puede ser transportado se le conoce como el archivo de intercambio del paquete.

Este archivo entonces contiene a nivel raíz el archivo `imsmanifest.xml` junto con los archivos físicos dada la descripción que se encuentra en el archivo `xml`. Además contiene otros documentos de control *XML* que son referenciados por `imsmanifest.xml`, ya sean *DTD* o *XSD* (archivos que definen la estructura de un *XML* para que sea válido).

El paquete representa una unidad con contenido utilizable. Esta unidad puede ser un curso o parte de él. El contenido consta de recursos los cuales son los archivos físicos o referencias a ciertos URL's.

Estos recursos son distribuidos de tal forma que la combinación de los archivos físicos es realmente el contenido y lo que viene siendo la parte atómica de estas combinaciones son precisamente los archivos físicos.

La especificación del modelo es muy precisa al mencionar que en la raíz del paquete debe estar incluido el archivo “`imsmanifest.xml`” junto con su archivos de apoyo (*DTD's* o *XSD's*). Si este archivo no se encuentra en este lugar con ese nombre entonces el paquete no puede ser procesado.

El archivo `imsmanifest.xml` de la versión del modelo 1.1.3 organiza el contenido de la siguiente manera.

*Meta-Data*: En los casos, *manifest*, *organization* o *item*, el meta-data sirve como descripción a su elemento padre.

*Manifest*: Unidad de instrucción reusable. Encapsula *meta-data*, *organizations* y *resources*.

*Organizations*: Describe una o más estructuras dentro del paquete.

*Organization*: Describe una organización jerárquica en particular.

*Item*: Un nodo que va dando forma a la organización.

*Resources*: Una colección de recursos.

*Resource*: Referencia a algún recurso.

*File*: Una lista de archivos de los cuales este recurso (*resource*) depende.

*Dependency*: Identifica un recurso (*resource*) de cuyos archivos (*files*) este recurso depende.

Se cuenta con un nodo padre que es *manifest* el cual engloba todos los elementos del archivo. Los elementos a los que envuelve este padre son los *organizations* y los *resources*. Los *organizations*, almacenan elementos del tipo *organization*. Los *organization* contienen una estructura jerárquica de elementos *item*. Estos *item* entonces también pueden o no contener elementos *item*. Los *item* tienen un atributo (*identifierref*) que hace referencia a uno de los elementos *resource* que están almacenados en el elemento *resources*. Cada *resource* contiene a su vez elementos *file* y *dependency*. Estos *file* representan la ruta dentro del paquete a algún archivo físico que el archivo principal del *resource* necesita como auxiliar y *dependency* hace referencia a algún otro *resource* de cuyos archivos este *resource* depende.

Cabe destacar que cada elemento *organization* e *item* cuentan con su propio identificador, además de que cada elemento cuenta con atributos propios que para el alcance de este documento no es necesario cubrir. *Meta-data* como se había mencionado sirve para darle descripción al elemento padre y la definición de estos es también proporcionada por el modelo. Esta descripción puede llegar a ser utilizada para efectos de extensibilidad del modelo.

A continuación se presenta un ejemplo mucho más claro y concreto de un archivo *imsmanifest.xml* que sirve para representar el modelo. **Fig. A.1**

En este ejemplo sencillo se cuenta con los elementos más básicos pero también los más útiles dentro del `imsmanifest.xml`. Se cuenta con tres *organization* dentro de *organizations*. Cada *organization* cuenta con su identificador (*identifier*) y su título (*title*). Como se había mencionado cada *organization* representa una estructura jerárquica de elementos tipo *items*. En el primer *organization* se cuenta con dos elementos *item*, a diferencia de los otros dos *organization* los cuales a pesar de también tener dos elementos de tipo *item* tienen jerarquía diferente ya que un *item* está contenido dentro de otro.

Cada tipo *item* cuenta con identificador y título además del atributo *identifierref* el cual hace referencia al identificador de algún *resource* dentro de los *recursos*. Por ejemplo, el *item* con identificador “ITEM-153” de la segunda *organization* hace referencia al tipo *resource* con identificador 153, este *resource* nos dice que el elemento al que el hace referencia es una liga a “<http://www.alunam.unam.mx>”, entonces ese *item* en realidad por debajo representa esa liga. Otro caso distinto es el del *item* con identificador “ITEM-154” del tercer *organization* ya que este hace referencia al *resource* con identificador 154, el cual a su vez hace referencia al archivo con ruta dentro del paquete `recursos_vacios/recurso_vacio_1.html` y cuyos archivos que necesita son el mismo archivo, estos es prácticamente para todos estos casos, además de otro en `notas/js/funciones.js` cada uno de ellos especificados por elementos de tipo *file*.

### *Patrones de diseño.*

Los patrones de diseño son modelos tomados de la experiencia cuya función principal es proveer una solución efectiva y reusable a los distintos problemas que se enfrentan durante el diseño y desarrollo de software.

Existen diversas fuentes que hablan de este tema; sin duda alguna la de mayor popularidad es el libro *Design Patterns* escrito por el cuarteto conformado por Erich Gamma, Richard Helm, Ralph Johnson y John Vlissides, mejor conocido como el Gang of four. De este libro y otras referencias se basa la búsqueda y la aprobación para encontrar patrones de diseño por distintos estudiosos del tema.

Los patrones de diseño son clasificados generalmente de acuerdo al tipo de problemas que resuelven. Dicha clasificación también surge como base a las que propone el *Gang of four* y son:

***De construcción:*** Dedicados a resolver problemas de creación y construcción de objetos.

***Estructurales:*** Patrones que de alguna u otra forma enmascaran el hecho de utilizar otras clases por debajo de otras que funcionan como las principales y que por consecuencia crean estructuras más complejas mediante su relación

***De comportamiento:*** Encapsulan y responsabilizan la funcionalidad a los objetos de forma tal que pueden delegar funcionamiento específico.

## *SCORM*

SCORM es una colección de estándares y especificaciones adaptadas de múltiples fuentes para proveer un espacio de capacidades en e-learning que permiten la interoperabilidad, accesibilidad y reusabilidad de contenido de aprendizaje basado en la Web.

Mientras existan sistemas que soporten este estándar es posible intercambiar contenidos que hayan sido creados en otros.

Son varias las especificaciones de las organizaciones que han sido integradas a este modelo o norma, entre las que se encuentra la del *IMS Global Learning Consortium, Inc (IMS Global)*, la cual detalla la forma de intercambio de paquetes.

*SCORM* abarca muchos detalles que van más allá del alcance de este documento. Sin embargo es importante recalcar el hecho de que esta otra especificación permite el intercambio de paquetes.

### *SCO (Sharable Content Object)*

Es un objeto de aprendizaje reusable que sigue el estándar *SCORM*. Un *LMS* que soporte *SCORM* puede comunicarse con este tipo de objetos y es capaz de interpretar instrucciones que le digan cual es el siguiente *SCO* a utilizar. El modelo *SCORM* incluye componentes adicionales que le permiten crear *SCOs* y ensamblarlos en unidades de aprendizaje más grandes.

```

<manifest xsi:schemaLocation="http://www.imsglobal.org/xsd/imsdp_v1p1
imsdp_v1p1.xsd http://www.imsglobal.org/xsd/imsmd_v1p2 imsmd_v1p2.xsd"
identifier="MANIFEST-con_lo_de_exp_sco">

  <organizations>
    <organization identifier="ORG-83">
      <title>con_lo_de_exp_sco</title>
      <item identifier="ITEM-66" identifierref="RES-AC-3">
        <title>act</title>
      </item>

      <item identifier="ITEM-67" identifierref="RES-N-3">
        <title>nota</title>
      </item>
    </organization>

    <organization identifier="ORG-84">
      <title>org_2</title>
      <item identifier="ITEM-68" identifierref="RES-NS-4">
        <title>notas</title>

        <item identifier="ITEM-153" identifierref="153">
          <title>liga</title>
        </item>
      </item>
    </organization>

    <organization identifier="ORG-108">
      <title>org3</title>

      <item identifier="ITEM-154" identifierref="154">
        <title>ninguno_utem</title>

        <item identifier="ITEM-155">
          <title>foro_item</title>
        </item>
      </item>
    </organization>
  </organizations>

  <resources>

    <resource identifier="RES-AC-3" type="webcontent" href="actividades/actividad_1.html">
      <file href="actividades/actividad_1.html"/>
    </resource>

    <resource identifier="RES-N-3" type="webcontent" href="notas/nota_1.html">
      <file href="notas/nota_1.html"/>
    </resource>

    <resource identifier="RES-NS-4" type="webcontent" href="subtemas/subtema_1.html">
      <file href="subtemas/subtema_1.html"/>
    </resource>

    <resource identifier="153" type="webcontent" href="http://www.alunam.unam.mx">
    </resource>

    <resource identifier="154" type="webcontent" href="recursos_vacios/recurso_vacio_1.html">
      <file href="recursos_vacios/recurso_vacio_1.html"/>
      <file href="notas/js/funciones.js"/>
    </resource>
  </resources>
</manifest>

```

FIG.- A.1 imsmanifest.xml



## Apéndice B.

### *Ajax.*

Acónimo de *Asynchronous JavaScript And XML* (*JavaScript* y *XML* asíncronos), es una técnica de desarrollo web para crear aplicaciones interactivas. Éstas se ejecutan en el cliente, es decir, en el navegador del usuario, y mantiene comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre la misma página sin necesidad de recargarla. Esto significa aumentar la interactividad, velocidad y usabilidad en la misma. **Fig. B.1**

*AJAX* es una combinación de tres tecnologías ya existentes:

*XHTML* (o *HTML*) y hojas de estilos en cascada (*CSS*) para el diseño que acompaña a la información.

*Document Object Model (DOM)* es un lenguaje de procesamiento por lotes por parte del usuario, especialmente implementaciones *ECMAScript* como *JavaScript* y *JScript*, para mostrar e interactuar dinámicamente con la información presentada.

El objeto *XMLHttpRequest* para intercambiar datos asincrónicamente con el servidor web. En algunos marcos de desarrollo y en algunas situaciones concretas, se usa un objeto *iframe* en lugar del *XMLHttpRequest* para realizar dichos intercambios.

*XML* es el formato usado comúnmente para la transferencia de vuelta al servidor, aunque cualquier formato puede funcionar, incluyendo *HTML* preformateado, texto plano, *JSON* y hasta *EBML*.

### *Jakarta.*

Es un el nombre de un proyecto que mantiene código abierto para la plataforma JAVA, opera bajo los auspicios de la Fundación para Software Apache.

### *PostgreSQL.*

Es un sistema manejador de bases de datos relacionales y es libre.

### *Struts.*

Es un entorno de desarrollo de código abierto para aplicaciones web, usa y extiende la interface de programación de los *servlets* de *JAVA* y está hecho para que los desarrolladores adopten una arquitectura *MVC*, modelo-vista-controlador.

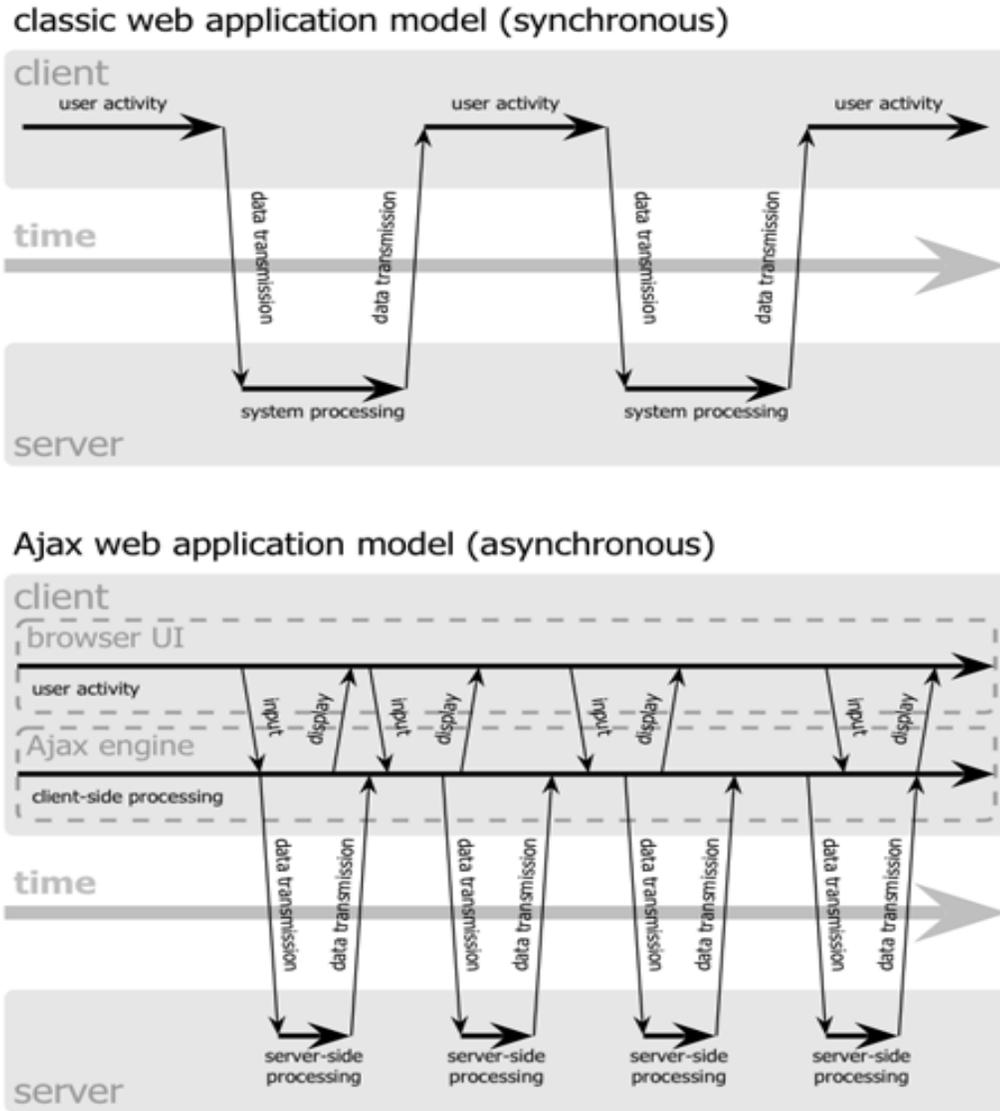


FIG.- B.1. 1 CONEXIÓN CON AJAX.

### *XMLBeans.*

*XMLBeans* es una tecnología para acceder al contenido de un archivo *XML* utilizando objetos Java que son creados a través de los metadatos del archivo *XML*.

Con un esquema de un archivo *XML* que ha sido compilado para generar objetos Java, ahora se puede acceder a ellos como se haría con un *JavaBean*, con métodos del tipo *get* y *set*, tiene un modelo de cursores con los cuales se puede obtener todo el modelo de datos del archivo *XML* y cuenta además con soporte para *XML DOM (Document Object Model)*, define una forma estándar para acceder y manipular archivos de extensión *xml*).

## **Bibliografía.**

### ***Dirección General de Servicios de Cómputo Académico.***

Sistema Alunam, Consulta: febrero, 2008

<http://www.alunam.unam.mx/>

### ***Wikimedia Foundation.***

Wikipedia, Consulta: julio, 2007

<http://wikipedia.org>

### ***Advanced Distributed Learning.***

Scorm, Consulta: febrero, 2008

<http://www.adlnet.gov/scorm/>

### ***Apache Software Foundation.***

Consulta: julio, 2007

<http://www.apache.org/>

### ***World Wide Web Consortium.***

Consulta: julio, 2007

<http://www.w3.org/>

FCKEditor, Consulta: marzo, 2008

<http://www.fckeditor.net/>

Dojo, Consulta: abril, 2008

<http://dojotoolkit.org/>

***Dynarch.***

JSCalendar, Consulta: marzo, 2008

<http://www.dynarch.com/projects/calendar/>

***IMS Global Consortium.***

Content Packaging, Consulta marzo, 2008

<http://www.imsglobal.org>

<http://www.imsglobal.org/content/packaging/index.html>

Moodle, Consulta: marzo, 2008

<http://moodle.org>

PostgreSQL, Consulta: febrero, 2008

<http://www.postgresql.org/>

***Sun Microsystems.***

Java, Consultado: julio, 2007

<http://java.sun.com/>

Scorm concepts, Consulta: febrero 2008

<http://www.eduworks.com/LOTT/Tutorial/scormconcepts.html>

Stephen Stelting, Olav Maassen, *Applied Java Patterns*, Prentice Hall Professional Technical Reference, Estados Unidos, 2001

Gamma Erich, Helm Richard, Johnson Ralph, Vlissides John, *Design Patterns: Elements of Reusable Object-Oriented Software*, Addison-Wesley Professional Computing Series, Estados Unidos, 1995