

**PROTOTIPO DEL SISTEMA DE MENTORÍA VIRTUAL PARA
DOCENTES DEL CENTRO DE IDIOMAS DE LA UNIVERSIDAD
EAFIT**

ANA SOFÍA GALLO VARGAS
LUISA FERNANDA HURTADO JARAMILLO

UNIVERSIDAD EAFIT
ESCUELA DE INGENIERÍAS
DEPARTAMENTO DE INFORMÁTICA Y SISTEMAS
MEDELLÍN
2008

**PROTOTIPO DEL SISTEMA DE MENTORÍA VIRTUAL PARA
DOCENTES DEL CENTRO DE IDIOMAS DE LA UNIVERSIDAD
EAFIT**

ANA SOFÍA GALLO VARGAS
LUISA FERNANDA HURTADO JARAMILLO

Proyecto de grado

Asesores

John Antonio Trujillo Vargas
Carolina Pabón Ramírez

UNIVERSIDAD EAFIT
ESCUELA DE INGENIERÍAS
DEPARTAMENTO DE INFORMÁTICA Y SISTEMAS
MEDELLÍN
2008

Nota de aceptación:

Firma del presidente del jurado

Firma del jurado

Firma del jurado

Medellín, ___ de _____ de ____

TABLA DE CONTENIDO

INTRODUCCIÓN.....	17
1. DEFINICIÓN DEL PROBLEMA	19
2. OBJETIVOS DEL PROYECTO.....	21
2.1. OBJETIVO GENERAL.....	21
2.2. OBJETIVOS ESPECÍFICOS.....	21
3. ALCANCE Y PRODUCTOS ESPERADOS.....	23
4. JUSTIFICACIÓN	24
4.1. CENTRO DE IDIOMAS	24
4.2. SOCIAL	24
4.3. PERSONAL.....	25
5. MARCO DE REFERENCIA	27
5.1. ESTÁNDAR SCORM.....	27
5.1.1. Contexto e inicios.....	27
5.1.2. ¿Qué es SCORM?	29
5.1.3. Características	30
5.2. E-LEARNING	36
6. SOLUCIÓN.....	41
6.1. METODOLOGÍA A SEGUIR	41
6.2. ¿POR QUÉ SCORM?.....	43

6.3.	¿POR QUÉ EAFIT INTERACTIVA?	44
6.4.	DISEÑO DEL CURSO	45
7.	DESARROLLO DEL PROTOTIPO	63
7.1.	INFORMACIÓN TÉCNICA DEL PROTOTIPO	63
7.2.	FASE DE INICIO	63
7.2.1.	Captura de requisitos	63
7.3.	FASE DE ELABORACIÓN	70
7.3.1.	Modelo de casos de uso	70
7.4.	FASE DE CONSTRUCCIÓN	129
7.4.1.	Diagrama de paquetes	129
7.4.2.	Diagrama de clases	129
7.4.3.	Diagrama de comunicación	142
7.4.4.	Diagrama de base de datos	144
7.5.	FASE DE TRANSICIÓN	146
7.5.1.	Manual de usuario.....	146
8.	RECOMENDACIONES FUTURAS	184
	CONCLUSIONES.....	187
	BIBLIOGRAFÍA.....	190
	ANEXOS	194

LISTA DE FIGURAS

Figura 1. Comportamiento típico del proceso de comunicación de un SCO con un LMS.....	32
Figura 2. Relación entre el contenido y el árbol de actividades	34
Figura 3. Estructura del entorno de ejecución del SCORM	36
Figura 4. Página de ingreso al curso	59
Figura 5. Página para seleccionar los ítems	60
Figura 6. Ejemplo de comic	60
Figura 7. Ejemplo de comic	61
Figura 8. Ejemplo de la actividad evaluativa	62
Figura 9. Casos de uso generales del sistema	71
Figura 10. Diagrama del caso de uso ingresar al sistema	72
Figura 11. Diagrama del caso de uso seleccionar curso	75
Figura 12. Diagrama del caso de uso gestionar curso.....	77
Figura 13. Diagrama del caso de uso gestionar usuarios	80
Figura 14. Diagrama del caso de uso gestionar recursos.....	82
Figura 15. Diagrama del caso de uso gestionar foro	84
Figura 16. Diagrama del caso de uso crear recepción de trabajos.....	87
Figura 17. Diagrama del caso de uso iniciar conversación en chat	88
Figura 18. Diagrama del caso de uso gestionar wiki	90
Figura 19. Diagrama del caso de uso gestionar agenda.....	91
Figura 20. Diagrama del caso de uso revisar frecuencia de uso	93
Figura 21. Diagrama del caso de uso gestionar bibliografía	95

Figura 22. Diagrama del caso de uso gestionar glosario	98
Figura 23. Diagrama del caso de uso gestionar FAQ	100
Figura 24. Diagrama del caso de uso gestionar Enlaces.....	103
Figura 25. Diagrama del caso de uso gestionar contenido.....	105
Figura 26. Diagrama del caso de uso desplegar contenido SCORM.....	108
Figura 27. Diagrama del caso de uso gestionar contenido SCORM.....	110
Figura 28. Diagrama del caso de uso visualizar SCO	115
Figura 29. Diagrama del caso de uso hacer seguimiento al usuario	116
Figura 30. Diagrama del caso de uso realizar evaluación	118
Figura 31. Diagrama del caso de uso enviar notificación al docente	120
Figura 32. Diagrama del caso de uso error en procesamiento	121
Figura 33. Diagrama del caso de uso cancelar recurso.....	123
Figura 34. Diagrama del caso de uso ingresar archivo adjunto	124
Figura 35. Diagrama del caso de uso notificar fecha inválida.....	126
Figura 36. Diagrama del caso de uso notificar datos erróneos.....	127
Figura 37. Diagrama de paquetes	129
Figura 38. Clases del paquete control	130
Figura 39. Clases del paquete modelo	132
Figura 40. Clases del paquete vo	134
Figura 41. Clases del paquete SCORM.....	136
Figura 42. Diagrama de comunicación para el caso de uso “Gestionar contenido SCORM”	142
Figura 43. Diagrama de comunicación para el caso de uso “Visualizar SCO”.....	143
Figura 44. Diagrama de base de datos.....	144
Figura 45. Página de inicio al sistema	147
Figura 46. Página de inicio	148
Figura 47. Página de opciones generales del sistema	149
Figura 48. Página de contenidos de una materia	150
Figura 49. Página para editar un contenido.....	151
Figura 50. Página para importar un archivo SCORM	152

Figura 51. Página de importar archivo SCORM con los datos.....	153
Figura 52. Página de información del resultado del análisis del curso	154
Figura 53. Página de inicio del curso	155
Figura 54. Página que muestra un ítem del curso	155
Figura 55. Página de finalización del curso	156
Figura 56. Organización de la carpeta "Template"	157
Figura 57. Cambio del nombre de la carpeta.....	158
Figura 58. Código de ejemplo de la página HTML.....	159
Figura 59. Código de ejemplo para crear los botones de navegación	159
Figura 60. Código de ejemplo para incluir un archivo flash	159
Figura 61. Código de ejemplo para crear una sección	160
Figura 62. Organización de la carpeta Curso1	161
Figura 63. Páginas que hacen parte del curso	162
Figura 64. Creación paquete SCORM 2004 en RELOAD.....	163
Figura 65. Ventana para seleccionar la carpeta donde se va a almacenar el paquete	164
Figura 66. Secciones del Reload	165
Figura 67. Cambio del perfil por defecto	166
Figura 68. Ítems que hacen parte del manifest.....	167
Figura 69. Cambio de la versión del esquema.....	168
Figura 70. Ítems que deben ser arrastrados a recursos	170
Figura 71. Recursos creados en el manifest.....	171
Figura 72. Cambio del tipo de recurso	173
Figura 73. Creando el apartado 1	175
Figura 74. Pasos para crear un nuevo ítem.....	176
Figura 75. Vista de la tabla de contenidos	177
Figura 76. Pagina HTML que se ejecutara para el ítem introducción	178
Figura 77. Pasos para guardar un paquete	179
Figura 78. Pasos para ver el diseño final del curso	180
Figura 79. Pre-visualización del curso	181

Figura 80. Pasos para empaquetar el curso	182
Figura 81. Ventana para guardar el paquete	183

ANEXOS

Anexo 1: Código de los Java Script utilizados en el template del curso.	194
Anexo 2: Código Java Script que se posee la implementación de la comunicación con el API de SCORM.....	222

GLOSARIO

ADL (Advanced Distributed Learning): Iniciativa propuesta por el departamento de defensa y la oficina de políticas de ciencia de la casa blanca de los Estados Unidos para desarrollar tecnologías de aprendizaje. Es la propulsora del SCORM.

AICC (Aviation Industry Computer-Based Training Committee): Asociación que se encarga de la capacitación de profesionales apoyado en la tecnología.

API (Application Program Interface): Es un conjunto de métodos y funciones de programación que permite abstraer e invocar servicios, evitando la tarea de programar todo desde el principio.

Applet: Un applet es un programa escrito en el lenguaje de programación Java que puede ser incluido en una página HTML, en la misma forma en que una imagen es incluida en una página. Cuando utiliza un navegador que tiene habilitada la tecnología Java para ver una página que contiene un applet, el código del applet es transferido a su sistema y ejecutado por la máquina virtual de Java (JVM)⁽¹⁾.

Árbol DOM: Es una representación de un documento XML como una estructura de árbol, con elementos, atributos, y texto como nodos⁽²⁾.

Asset: Recurso que hace parte de un objeto de aprendizaje. Puede ser una imagen, una página web, un video, etc.

CI (Centro de Idiomas): Dependencia de la Universidad EAFT encargada de la enseñanza de lenguas extranjeras a estudiantes tanto externos como vinculados a dicha institución.

CSS (Cascading Style Sheets): es un mecanismo simple que describe cómo se va a mostrar un documento en la pantalla, o cómo se va a imprimir, o incluso cómo va a ser pronunciada la información presente en ese documento a través de un dispositivo de lectura. Esta forma de descripción de estilos ofrece a los desarrolladores el control total sobre estilo y formato de sus documentos⁽³⁾.

DOM (Document Object Model): Es un API para documentos HTML y XML. Provee una representación estructural del documento, lo que permite modificar su contenido y presentación visual. Esencialmente, conecta páginas web a scripts o lenguajes de programación⁽⁴⁾.

E-learning: Es una manera flexible y poderosa mediante la cual individuos y grupos apropián nuevos conocimientos y destrezas con apoyo de tecnología de redes de computadores. Esta permite diseminar y tener acceso a información multimedia, hacer uso de simuladores, al tiempo que permite interacción y colaboración con aprendices que pueden estar dispersos alrededor del mundo⁽⁵⁾.

Frame (Marco): Los marcos HTML permiten a los autores presentar documentos con vistas múltiples, que pueden ser ventanas o subventanas independientes. Las vistas múltiples ofrecen a los diseñadores una manera de mantener cierta información visible mientras otras vistas se desplazan o se sustituyen⁽⁶⁾.

Frameset: Es un documento HTML que describe un diseño de marco (llamado un documento frameset) el cual tiene una organización diferente a un documento HTML sin marcos. Un documento estándar tiene una sección head y una body. Un

documento frameset tiene una sección head, y una frameset en lugar de un body⁽⁶⁾.

IEEE (Institute of electrical and electronic engineers): Es una asociación internacional sin ánimo de lucro con sede principal en la ciudad de Piscataway en los Estados Unidos y subsedes en más de 150 países del mundo, con alrededor de 360.000 miembros, entre profesionales y estudiantes de ingeniería, diseño, derecho, administración, medicina, biología y ciencias afines⁽⁷⁾.

IEEE LTSC (Learning Technologies Standardization Committee): Comité que desarrolla normas técnicas y prácticas recomendadas para el mantenimiento y el desarrollo de software educativo.

IMS Global Learning Consortium: Consorcio norteamericano que se encarga de definir especificaciones para la aplicación de servicios educativos distribuidos.

LMS (Learning management System): Software que automatiza la administración del aprendizaje. El LMS registra a los usuarios, hace seguimiento de los cursos en un catalogo, almacena la información de los estudiantes; y proporciona informes de gestión. Un LMS está diseñado típicamente para manejar cursos de múltiples editores y proveedores. Por lo general, no incluye sus propias capacidades de autoría; al contrario, se centra en la gestión de los cursos creados por una variedad de otras fuentes⁽⁸⁾.

Metadatos: Información acerca de un contenido que lo habilita para ser almacenado y obtenido de una base de datos⁽⁸⁾.

Objeto de Aprendizaje: Elemento susceptible a ser desplegado en un navegador Web que potencia la adquisición de un conocimiento con un único objetivo. Debe

cumplir con ciertas reglas como: La interoperabilidad, la independencia del contexto bajo el cual está incluido.

Debe tener un único objetivo de aprendizaje el cual potenciar en el aprendiz. Se compone de recursos digitales que pueden ser desplegados en un navegador Web⁽⁹⁾.

SCO (Sharable Content Object): Es un recurso de aprendizaje que puede ser corrido en un entorno SCORM. Un SCO está compuesto por varios assets.

SCORM (Sharable Content Object Reference Model): Es un conjunto de especificaciones técnicas basadas en la labor de la AICC, IEEE e IMS para crear un modelo unificado de contenido. Estas especificaciones permiten la reutilización de contenido de aprendizaje basado en web a través de múltiples ambientes. SCORM dicta cómo un LMS debe hacer la API disponible para el contenido, para que desarrolladores de contenidos sepan exactamente cómo escribir el código JavaScript para localizar y llamar a la API⁽⁸⁾.

Storyboard: Es la descripción provisional previa a la elaboración definitiva de un curso.

TIC (Tecnologías de la Información y la Comunicación): Son un conjunto de servicios, redes, software y aparatos que tienen como fin la mejora de la calidad de vida de las personas dentro de un entorno, y que se integran a un sistema de información interconectado y complementario⁽¹⁰⁾.

TOC (Table Of Contents): Tabla de contenidos de un curso desarrollado en SCORM.

Token: También llamado componente léxico es una cadena de caracteres que tiene un significado coherente en cierto lenguaje de programación. Ejemplos de

tokens, podrían ser palabras clave (if, while, int), identificadores, números, signos, o un operador de varios caracteres⁽¹¹⁾.

XML (Extensible Markup Language): La próxima generación de lenguaje de codificación para páginas web que permite a los diseñadores web programar sus propios comandos markup, los cuales pueden ser utilizados como si fueran comandos HTML estándar⁽⁸⁾.

INTRODUCCIÓN

La enseñanza de lenguas extranjeras es una práctica que ha venido tomando fuerza desde hace algunos años en nuestro país. La Universidad EAFIT con su Centro de Idiomas busca prestar este servicio de la forma más competente y optima posible, para de esta forma consolidarse como una institución de calidad a nivel internacional.

Para lograr lo anterior, el Centro de Idiomas se ha preocupado por establecer unos estándares metodológicos que ayuden a los docentes vinculados al centro a lograr un mejor desempeño que se vea reflejado en el aprendizaje de los alumnos.

Basados en lo anterior, y viendo el esfuerzo que ha venido ejerciendo la Universidad EAFIT por incorporar la tecnología para apoyar sus procesos pedagógicos, surgió la iniciativa de investigar una solución desde el área de la ingeniería de sistemas que facilite el proceso de capacitación de los docentes y que posibilite la comunicación entre ellos sin barreras de tiempo y espacio al mismo tiempo que propicia un aprendizaje colaborativo.

Para propiciar la comunicación entre los docentes nos valdremos de una plataforma ya existente en la Universidad EAFIT. De esta forma, utilizaremos las funcionalidades existentes en EAFIT Interactiva tales como foro, wiki, correo, agenda y contenidos entre otros para que los docentes intercambien información y permanezcan en contacto. Igualmente, se integrará a dicha plataforma una opción

para incluir contenidos virtuales que cumplan el estándar SCORM a una asignatura.

Adicionalmente, y conscientes de las dificultades que pueden presentarse para una persona no especializada en el tema la construcción de un curso virtual que cumpla con el estándar SCORM, se entrega un curso base que explica los pasos que se deben seguir para la construcción de éste.

Finalmente, y con el objetivo de dar una base a trabajos futuros, se desarrolló un prototipo de un modulo de un curso virtual enfocado a la capacitación de docentes del centro de idiomas de la Universidad EAFIT.

La propuesta acá descrita es presentada como proyecto de grado para optar por el título de Ingeniería de Sistemas de la Universidad EAFIT.

1. DEFINICIÓN DEL PROBLEMA

Para el Centro de Idiomas de la universidad EAFIT es importante que la metodología de enseñanza aplicada a todos los programas de lenguas sea homogénea y estandarizada pues de esta forma se asegura el cumplimiento de los objetivos propuestos por dicho centro. Lo anterior se logra mediante capacitaciones y mentorías a los docentes nuevos dirigidas por un asesor de idioma que inducen, no solo en los modelos pedagógicos seguidos por el Centro de Idiomas, sino en los procesos y formatos administrativos que los docentes deben seguir mientras permanezcan vinculados al Centro como empleados.

Actualmente para los docentes de ciertos programas asistir a dichas capacitaciones e inducciones se dificulta debido a los lapsos de tiempo que permanecen en el Centro de Idiomas, por ser docentes de cátedra que sólo asisten al Centro de Idiomas para dar las clases que les son asignadas.

Es por esto que no siempre el proceso de mentoría se puede realizar, trayendo como consecuencia metodologías y seguimiento de actividades subjetivas y determinadas, no por el Centro de Idiomas, sino por el docente mismo.

En otros casos, a medida que transcurre el tiempo los docentes no tienen la posibilidad de realizar una retroalimentación o adquisición y actualización del conocimiento, técnicas y metodologías.

Por esta razón se hace necesario presentar una solución tecnológica, que pueda ser accedida a cualquier hora y desde cualquier lugar, que apoye de manera virtual no sólo el proceso de mentoría inicial sino que provea una herramienta que permita a los docentes ser agentes activos de su propio proceso y de esta manera actualizarse, construir conocimiento de manera colectiva y mejorar en cuanto al manejo de metodologías, requisitos administrativos y políticas del centro de idiomas, estando así acorde con los lineamientos pedagógicos de la institución.

2. OBJETIVOS DEL PROYECTO

2.1. OBJETIVO GENERAL

Construir el prototipo de un sistema que apoye un proceso de mentoría de manera virtual, donde los lineamientos establecidos en cuanto a metodología, estrategias, requisitos administrativos, procesos académicos y gestión del conocimiento estén estandarizados para todos los programas, homogeneizando la calidad académica que percibe el estudiante.

2.2. OBJETIVOS ESPECÍFICOS

- ⌚ Reflejar conceptos de teorías de aprendizaje y didáctica en los diferentes módulos del prototipo del sistema que permitan a los aspirantes a docentes asumir sus procesos de formación de una manera estándar y acordes con los objetivos del centro de idiomas.
- ⌚ Proporcionar una herramienta que no tenga barreras de tiempo, espacio e idioma, donde los aspirantes a docentes puedan acceder cuando lo requieran sin importar sus particulares limitaciones.
- ⌚ Adaptar el estándar SCORM a la plataforma EAFIT interactiva para que los diferentes cursos virtuales previamente diseñados puedan ser visualizados, adicionados y eliminados en esta plataforma.

- ⌚ Desarrollar un prototipo de curso de mentoría virtual basado en el estándar SCORM que contenga los lineamientos básicos metodológicos y administrativos del Centro de Idiomas para que los aspirantes a docentes, coordinadores y mentores puedan navegar por éste a través de EAFIT Interactiva.

- ⌚ Filtrar la información desplegada de acuerdo al tipo de usuario que ingresa al sistema y tener en cuenta los permisos de cada usuario para permitirle realizar las diferentes funciones.

- ⌚ Implementar sesiones con audio, video e imágenes que muestren fragmentos de clases ideales a partir de los cuales los docentes puedan realizar actividades de análisis y reformulación de metodologías en su proceso de capacitación.

3. ALCANCE Y PRODUCTOS ESPERADOS

El presente proyecto dará como resultado un prototipo de software de uso exclusivo para el Centro de Idiomas de la Universidad EAFIT, el cual será desarrollado para la plataforma EAFIT Interactiva, realizando las adecuaciones necesarias a la plataforma entre las que se incluye la adaptación de estándar SCORM.

El prototipo del sistema contendrá un curso virtual inicial apoyado en las necesidades más inmediatas del Centro de Idiomas. Del mismo modo, el sistema quedará abierto a modificaciones y posibles mejoras que vayan surgiendo de acuerdo a las necesidades específicas del Centro de Idiomas, los clientes y usuarios del sistema.

Se presentará un documento escrito que servirá de fuente y consulta sobre los principios metodológicos, didácticos y tecnológicos utilizados para la implementación del sistema. Adicionalmente se entregará un manual escrito de usuario desde el cual los docentes, coordinadores y asesores puedan resolver las dudas particulares sobre el funcionamiento de éste.

Como subproducto, esta implementación será una fuente de información integral en el tema del software educativo, y servirá a su vez de motivación para la implementación y la investigación de este tipo de sistemas.

4. JUSTIFICACIÓN

4.1. CENTRO DE IDIOMAS

El uso de las tecnologías de información y la comunicación (TIC) como apoyo a los procesos de aprendizaje de los diferentes entornos educativos se ha convertido en un tema que día a día ha tomado auge en nuestra sociedad.

Al apoyar el proceso de enseñanza - aprendizaje del Centro de Idiomas de la Universidad EAFIT con la tecnología, no sólo busca estar a la vanguardia y entrar en el auge antes mencionado; el planteamiento de la solución acá expuesta se basa en la necesidad del Centro de Idiomas por mantener procesos de capacitaciones sin limitaciones de tiempo y espacio y qué mejor forma de romper estas barreras que con la ayuda de la tecnología y más específicamente de la Web 2.0 cuyas características de comunicación hacen más fácil establecer ambientes de aprendizaje virtuales y colaborativos. La solución tecnológica aquí presentada es útil para el Centro de Idiomas porque no solo permitirá un intercambio de información más dinámica e interactiva, si no porque promoverá una participación más activa del proceso de aprendizaje por parte de los docentes vinculados a dicho centro. De esta forma los docentes podrán tener capacitaciones e información actualizada y pertinente que se verán reflejados en la calidad de la enseñanza impartida en sus cursos.

4.2. SOCIAL

El impacto que tiene una mentoría virtual y la plataforma EAFIT Interactiva sobre los docentes de lenguas extranjeras vinculados al Centro de Idiomas es muy alto porque contribuye a que el desempeño del docente sea mejor en las aulas. Las herramientas colaborativas e interactivas que se están ofreciendo, posibilita que el intercambio de información sea más fluido y se pueda hacer una retroalimentación de lo aprendido entre los mismos pares. De esta forma, la trascendencia de este proyecto en lo social - aunque no se evidencia de forma directa - radica en el aporte educativo que pueden traer estos docentes a los alumnos que están interesados en aprender una segunda lengua no solo en el ámbito del Centro de Idiomas de la Universidad EAFIT si no en otros entornos educativos. Los docentes serán personas mejor preparadas en los aspectos metodológicos y pedagógicos concernientes con sus materias, reflejándose en el apoyo, aprendizaje y estimulación de los alumnos pertenecientes a los diferentes espacios educativos de la sociedad.

4.3. PERSONAL

A nivel personal, la realización de este trabajo tiene gran importancia, porque por medio de él podemos enfrentar situaciones reales que nos permitan aplicar los conocimientos aprendidos durante la carrera de ingeniería de sistemas. Este trabajo de grado nos dará una base para comprender la forma en que deben ser manejadas las aplicaciones Web en el contexto de ambientes de aprendizaje y de esta forma adquirir experiencia y adoptar mejores prácticas en desarrollos posteriores que tengan características similares.

Igualmente gracias al trabajo aquí presentado podemos entrenar y mejorar nuestras habilidades investigativas y de trabajo en equipo, hecho que es altamente importante para nuestro desempeño como profesionales.

5. MARCO DE REFERENCIA

5.1. ESTÁNDAR SCORM

5.1.1. Contexto e inicios

En un principio, cuando se empezó a desarrollar contenidos educativos, no existían estándares que aseguraran la calidad de las aplicaciones, de esta forma, cuando se implementaba una solución, esta podía ser ejecutada solamente en el entorno específico para la cual había sido creada. Por lo tanto la mantenibilidad, actualización y reutilización de los sistemas era una tarea difícil. Con lo anterior se hizo necesario definir una solución que resolviera la situación y permitiera mayor facilidad en los desarrollos. Algunas entidades públicas y privadas dedicadas al e-learning (Se cuentan como las más importantes IEEE LTSC (Learning Technologies Standardization Committee), IMS Global Learning Consortium, AICC (Aviation Industry Computer-Based Training Committee), y ADL (Advanced Distributed Learning)) crearon algunas especificaciones buscando mejorar aspectos como calidad, costo, facilidad de aplicación, etc ⁽¹²⁾. A continuación se describe brevemente algunos de los aspectos que se buscaban normalizar ⁽¹³⁾:

Metadatos

Los metadatos es la información relacionada con los tópicos educativos que se desean implementar. Es un conjunto de elementos que describe la estructura del recurso y de esta forma se permite la gestión y recuperación de cada uno de los recursos educativos.

Organización de contenido

Está relacionado con la necesidad de compartir recursos educativos entre diferentes aplicaciones, por esto se hace necesario estandarizar la forma en que dichos recursos son encapsulados y la estructura que estos deben mantener.

🕒 **Entorno de ejecución**

Lo que se busca con la normalización de los entornos de ejecución es implementar prácticas estandarizadas para la interacción entre el contenido educativo y los LMS y la forma en que este contenido va a ser desplegado dependiendo de las acciones que tomen los actores.

🕒 **Lenguajes para el modelado de unidades de aprendizaje**

El propósito con los lenguajes para el modelado es crear materiales altamente estructurados que describa no solo una unidad de estudio si no también los roles, las relaciones y las interacciones que se presentan dentro de un curso. El lenguaje debe estar definido de tal manera que sea compatible con muchas pedagogías diferentes.

🕒 **Información del alumno**

Con la información del alumno se espera crear modelos de datos estándares que permitan describir al alumno que interactúa. De esta forma, el alumno puede intercambiar información entre diferentes sistemas conservando un perfil y manteniendo la seguridad y privacidad necesarias.

🕒 **Evaluación**

Se busca definir formatos y procedimientos comunes para la forma en que son manejadas las evaluaciones en el e-learning. Dentro de esta área cabe también el proceso de intercambio de evaluaciones entre diferentes recursos.

5.1.2. ¿Qué es SCORM?

SCORM (Sharable Content Object Reference Model) cuya traducción al español significa modelo de referencia para objetos de contenido compatible es un estándar para el manejo de contenido e-learning propuesto por el Departamento de Defensa y la oficina de políticas de ciencia de la casa blanca de los Estados Unidos cuyo objetivo es proveer una metodología común para desarrollar contenidos de aprendizaje de tipo Web de forma que sean compatibles y reutilizables. De esta forma, los objetos de aprendizaje pueden ser empaquetados y compartidos utilizando un LMS (Sistema de gestión de aprendizaje).

Las principales características de SCORM son⁽¹⁴⁾:

Accesibilidad: es la habilidad de localizar y acceder componentes desde una locación remota y entregarlo a otras locaciones.

Adaptabilidad: es la habilidad de transformar las instrucciones para que se adapten a las necesidades individuales y organizacionales.

Viabilidad: La habilidad de incrementar la eficiencia y la productividad reduciendo tiempo y costos involucrados en la entrega de una instrucción.

Durabilidad: La habilidad de soportar la evolución y los cambios de la tecnología sin rediseñar costos, reconfigurar o re-codificar.

Interoperabilidad: La habilidad de tomar componentes desarrollados con ciertas herramientas o plataforma y usarlos en otra locación que tiene un conjunto diferente de herramientas o plataforma.

Reusabilidad: La flexibilidad al incorporar componentes en múltiples aplicaciones y contextos.

La norma SCORM está dividida en tres grandes aspectos. Para entender mejor el funcionamiento del SCORM se hará una descripción breve de cada una de ellos y de los elementos que la componen. Los 3 aspectos son⁽¹³⁾:

- ⌚ El modelo de agregación de contenido - **CAM (Content Aggregation model)** encargado del ensamblado, etiquetado y empaquetado de los contenidos educativos.
- ⌚ Secuenciado y navegación- **SN (Sequencing and Navigation)** encargado de la descripción de cómo los diferentes contenidos educativos que conforman un curso pueden ser mostrados al estudiante.
- ⌚ Entorno de ejecución- **RTE (Run-time Environment)**, cuyo propósito es garantizar la interoperabilidad entre los diferentes elementos educativos o SCOs.

5.1.3. Características

5.1.3.1. Modelo de agregación de contenidos (CAM)⁽¹⁵⁾

El SCORM CAM describe los componentes necesarios para construir una experiencia de aprendizaje, como empaquetarlos y como intercambiarlos de un sistema a otro. A continuación se detallan algunos de los componentes que hacen parte del estándar:

Asset

Un Asset es el bloque básico de un recurso de aprendizaje. Son representaciones electrónicas como texto, imágenes, sonido o cualquier otro tipo de recurso que pueda ser desplegado en un cliente Web con el objetivo de presentárselo a un alumno.

SCO (Sharable content object)

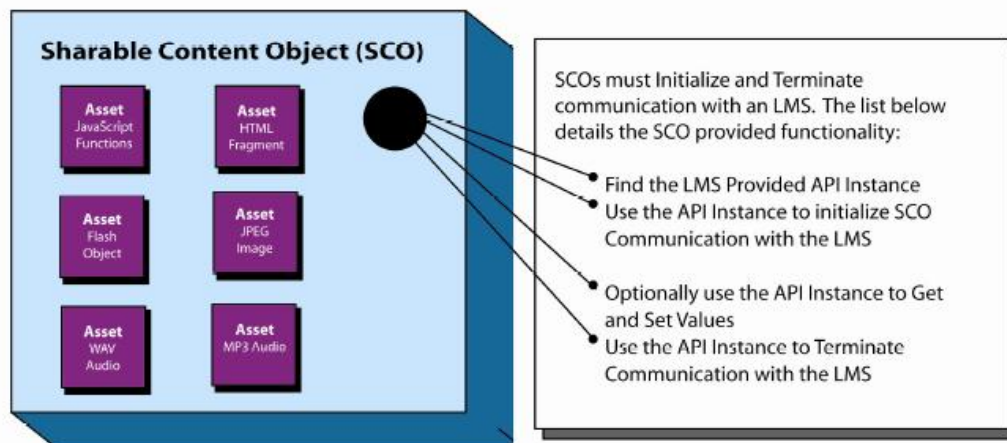
Un SCO constituye la más mínima unidad que puede ser gestionada por un LMS. Es una colección de uno o más Assets que representan un único recurso de aprendizaje, siendo lanzada en el entorno de ejecución del SCORM (RTE).

Los SCO son descritos por medio de metadatos para facilitar su búsqueda y mantenimiento.

En nuestro caso, el SCO contiene un API (Código java script) que se encarga de establecer un puente de comunicación entre el LMS y el objeto de aprendizaje.

La imagen a continuación muestra el comportamiento típico del proceso de comunicación de un SCO con un LMS: el SCO encuentra el API, usa la instancia API para inicializar la comunicación del SCO con el LMS y finalmente vuelve a utilizar el API para terminar la comunicación. Opcionalmente el API puede ser utilizado para obtener y dar valores.

Figura 1. Comportamiento típico del proceso de comunicación de un SCO con un LMS



Fuente: Documentación de SCORM⁽¹⁵⁾

🕒 Actividades

Una actividad de aprendizaje puede ser descrita como una unidad de instrucción significativa, es algo que el aprendiz realiza mientras avanza a través del proceso de aprendizaje. Puede proveer al alumno un SCO o un Asset o puede estar compuesta de varias sub-actividades.

🕒 Organización de Contenido

Es un mapa que define la forma como se usara el contenido a través de las actividades. El mapa muestra como están relacionadas las actividades de forma jerárquica para poder definir de esta forma el secuenciamiento. Así, el LMS interpreta la información de secuenciamiento descrita en la organización de contenido (Content organization) y controla dicho proceso en tiempo de ejecución.

🕒 Agregación de contenido

La agregación de contenido puede ser utilizada para describir el proceso de organizar un conjunto de objetos relacionados funcionalmente y de esta forma aplicarlos en una experiencia de aprendizaje.

Manifiesto (Manifest)

En un archivo XML que contiene un inventario estructurado del contenido de un paquete. Algunas veces puede también contener información de cómo está organizado el contenido.

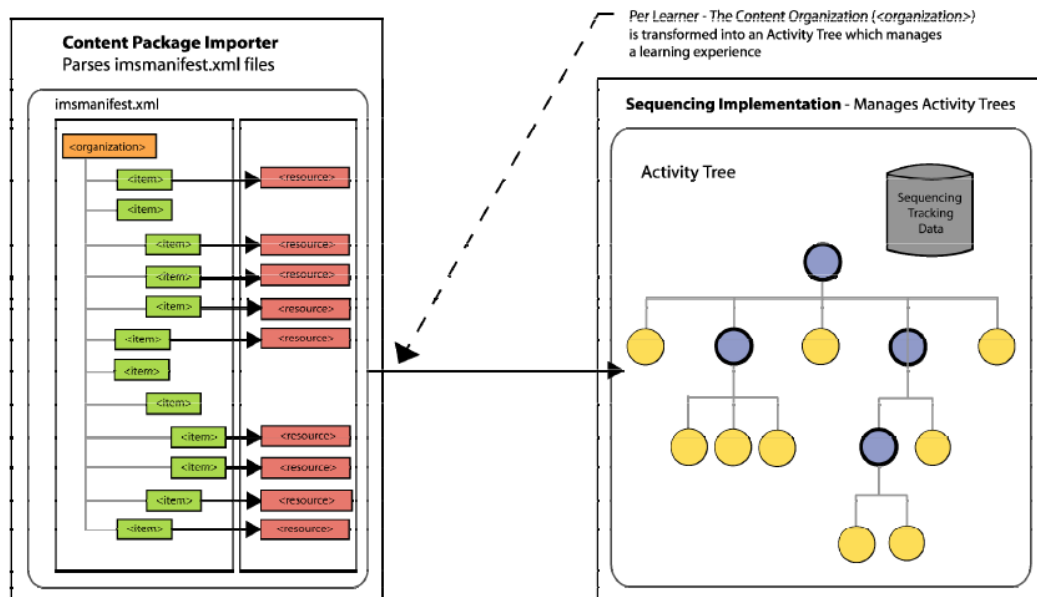
5.1.3.2. Secuenciado y navegación (SN) ⁽¹⁶⁾

Secuenciado

En esta parte se describe cómo el contenido es desplegado al estudiante de forma secuencial a través de eventos de navegación iniciados por el alumno y manejados a través de un LMS. El secuenciamiento en SCORM es definido en términos de actividades estructuradas de aprendizaje, en el “manifest” se encuentra estructurado el contenido y a partir de este se crea un árbol de actividades que muestra la jerarquía y las relaciones que tienen los elementos. Adicionalmente se define una estrategia de secuenciamiento.

La grafica siguiente muestra la relación existente entre el contenido y el árbol de actividades.

Figura 2. Relación entre el contenido y el árbol de actividades



Fuente: Documentación de SCORM⁽¹⁶⁾

El árbol de actividades representa la estructura conceptual del contenido. Para diseñar este árbol y de esta forma conocer el orden lógico del contenido es necesario que un LMS traduzca lo existente en el paquete de contenido (“content package”) y lo organice jerárquicamente (incluyendo el estado de rastreo de cada actividad) en un árbol.

Cuando un usuario desea interactuar con el contenido representado en el árbol de actividades, el LMS evalúa el árbol y el rastreo para determinar la secuencia de las actividades de aprendizaje.

🕒 Navegación

La navegación asume la existencia de mecanismos para lanzar eventos de navegación en las interfaces de los usuarios. Cuando el usuario lanza un evento,

el LMS traduce el evento a su correspondiente petición de navegación, procesa la petición y finalmente indica la próxima actividad de aprendizaje

5.1.3.3. Entorno de ejecución (RTE)⁽¹⁷⁾

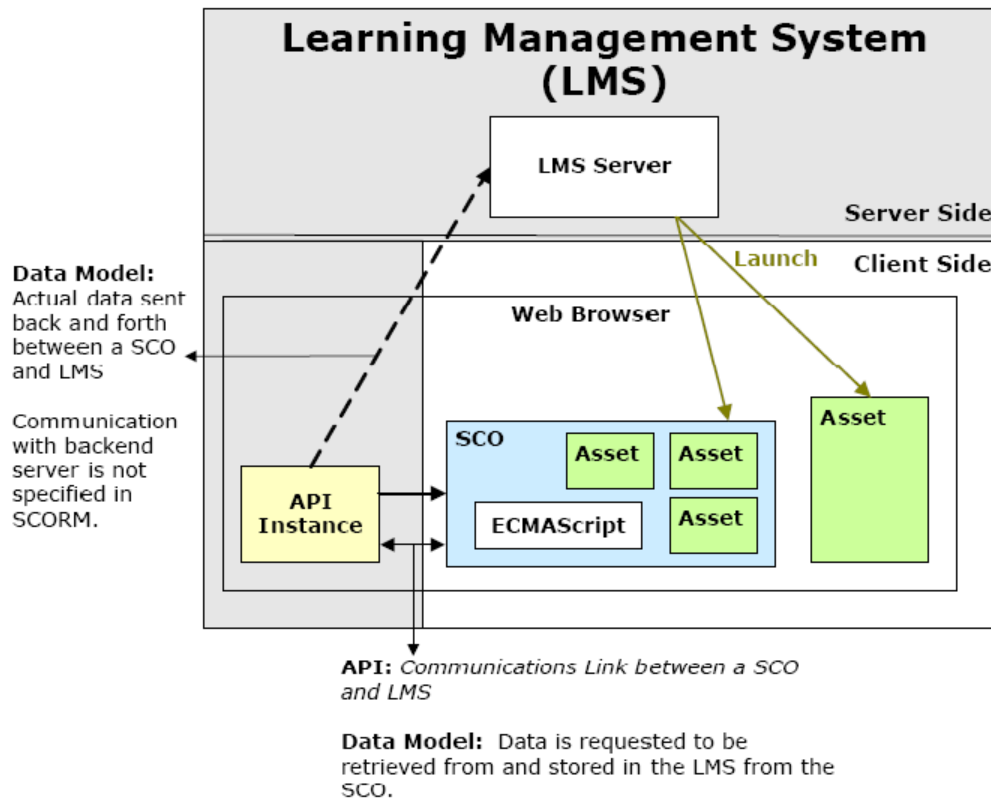
El modulo del entorno de ejecución del SCORM 2004 detalla los requisitos para ejecutar y lanzar objetos de contenido (objetos de aprendizaje o Assets), estableciendo una comunicación y manejando el rastreo entre SCO's y un LMS.

Específicamente el RTE de SCORM maneja los siguientes aspectos para lograr que los contenidos sean reutilizables e interoperables entre varios LMS's:

- ⌚ El proceso de la entrega de un objeto de contenido a un navegador web.
- ⌚ La forma en que un objeto de contenido se comunica con un LMS. El mecanismo de comunicación es estandarizado con un API común que entrega y almacena información entre el SCORM y el LMS.
- ⌚ La información que es rastreada de un objeto de contenido y como el LMS maneja esa información. Para lograr esto se utiliza un lenguaje predefinido o vocabulario por medio de un modelo de datos. El modelo de datos define de esta forma elementos que tanto el SCO como el LMS están esperando recibir y conocer. El LMS debe mantener el estado del modelo de datos del SCO durante todo el trascurso de la sesión del aprendiz.

La figura 3 expone de forma general la estructura del entorno de ejecución del SCORM.

Figura 3. Estructura del entorno de ejecución del SCORM



Fuente: Documentación SCORM⁽¹⁷⁾

Se puede observar que existe un LMS del lado del servidor que esta comunicándose que un objeto de contenido (bien sea un Asset o un SCO) que está localizado en la parte del cliente. Este puente de comunicación es un API que entrega en ambos sentidos un modelo de datos.

5.2. E-LEARNING

E-Learning se puede traducir al español como aprendizaje electrónico, por lo tanto comprende cualquier actividad que involucre la utilización de medios electrónicos y tecnológicos para apoyar toso o parte de un proceso de aprendizaje.

Una de las tantas definiciones que podemos encontrar de e-learning en las fuentes especializadas en el tema es la siguiente⁽¹⁸⁾: “el uso de tecnologías Internet para la entrega de un amplio rango de soluciones que mejoran el conocimiento y el rendimiento. Está basado en tres criterios fundamentales: 1. El *e-learning* trabaja en red, lo que lo hace capaz de ser instantáneamente actualizado, almacenado, recuperado, distribuido y permite compartir instrucción o información. 2. Es entregado al usuario final a través del uso de ordenadores utilizando tecnología estándar de Internet. 3. Se enfoca en la visión más amplia del aprendizaje que van más allá de los paradigmas tradicionales de capacitación”

El E-learning cuenta con tres elementos básicamente⁽¹⁹⁾: La tecnología, los contenidos y los servicios; incluyendo en este último las evaluaciones y los elementos de gestión y comunicación.

La plataforma utilizada para apoyar un proceso de aprendizaje en e-learning es el LMS (Learning Management System), Sistemas de manejo del aprendizaje, que son aplicaciones basadas en un servidor web que por medio de una interfaz gráfica permite que los alumnos interactúen con los contenidos del curso. Entre las funcionalidades más comunes que contienen las diferentes plataformas LMS son: visualización de contenidos del curso, posibilidad de realizar actividades programadas, comunicación con el docente y demás alumnos, seguimiento de actividades, calificaciones y datos estadísticos de progreso.

Entre las plataformas comerciales más comunes se encuentran: Blackboard (<http://www.blackboard.com>), WebCT (<http://www.webct.com>), Moodle (<http://moodle.org>) y Claroline (<http://www.claroline.net>).

Ventajas y Desventajas

	Ventajas	Desventajas
Comunicación e Interacción	El correo electrónico y los grupos de discusión crean y motivan las oportunidades de contactar e interactuar con el instructor u otros estudiantes fuera del aula de clase. Las interacciones asincrónicas en línea fomentan la reflexión y la participación del estudiante.	Dependiendo del contexto de aprendizaje o actividad puede darse que los intercambios en línea sean superficiales e impersonales y por lo tanto sean menos provechosos que las interacciones cara a cara.
Trabajo en grupo y colaboración	La enseñanza y el aprendizaje mediados por computadoras aumentan la participación del estudiante ya sea en las consultas o discusiones. Algunas herramientas o tecnologías de comunicación basadas en Web pueden ser usadas para dar soporte y alentar el aprendizaje colaborativo, el trabajo en grupo y la discusión. Por ejemplo, el trabajo en grupo y la colaboración pueden ser facilitados a través de	Los estudiantes e instructores pueden considerar que estas actividades en línea son un desperdicio de tiempo. El acceso a Internet o a la tecnología puede ser muy costoso como es el caso de nuestro medio.

	intercambio asincrónico, como el correo electrónico, o sincrónico como la videoconferencia.	
Flexibilidad, conveniencia y acceso a recursos en cualquier momento desde cualquier lugar	Fomenta el aprendizaje centrado en el estudiante y ofrece acceso al material de manera flexible, inmediata y conveniente. Un VLE basado en Web permite al estudiante escoger su forma de aprender y además facilita un intercambio abierto de ideas y opiniones.	La Web es un ambiente fragmentado y no lineal que puede resultar desorientador. Los estudiantes pueden al principio presentar problemas, lo cual significa trabajo extra.
Facilita la investigación y compartir recursos	La Internet es una fuente gigante de información. Fácilmente se puede hacer búsquedas y encontrar enlaces a información sobre temas y recursos relacionados con el curso.	Algunos recursos en línea pueden ser falsos o menos confiables que los que se encuentran en libros, y esto requerirá una evaluación de calidad y relevancia. A veces puede resultar difícil encontrar los recursos en Internet.
Pruebas y evaluación de desempeño/ recepción y almacenamiento	Las simulaciones, correos y herramientas de evaluación proporcionan oportunidades para auto-evaluar el desempeño y obtener	Puede ser una barrera que no satisface las necesidades de los estudiantes.

de retroalimentación	retroalimentación	
Presentación del material del curso	Se pueden usar sonidos, animaciones, vídeos y otros multimedios que ayuden al estudiante a comprender los contenidos del curso.	Los estudiantes pueden distraerse con los multimedios, porque no son una forma común de hacer las cosas.

Fuente: <http://www.cti.espol.edu.ec/docs/documento15.doc>

6. SOLUCIÓN

6.1. METODOLOGÍA A SEGUIR

Para desarrollar el prototipo de sistema de mentoría para docentes del Centro de Idiomas de la Universidad EAFIT, tomamos como base los principios del RUP (Rational Unified Process) por ser esta una de las metodologías más utilizadas en los procesos software, y por estar dirigido por los Casos de Uso, estar centrado en la arquitectura y ser iterativo e incremental; características que nos traían beneficios para la solución a nuestro problema específico.

De forma general explicaremos la metodología utilizada para la construcción del proyecto. Estos son los principales pasos y actividades:

Definición del problema

El centro de idiomas inicialmente expresó su necesidad de utilizar una herramienta virtual que apoyara el proceso de capacitación y mentoría a los docentes nuevos, sin embargo el problema no se quedó ahí. Al analizar más profundamente se llegó a la conclusión, que tener una herramienta virtual para la mentoría era importante, pero era también necesario proveer un medio donde los docentes, una vez completada la mentoría, pudieran seguir intercambiando información y tuvieran un canal de comunicación donde estos construyeran su aprendizaje y pudieran mejorar sus metodologías de manera conjunta y continua.

Búsqueda de soluciones

A partir del problema planteado por el centro de idiomas se empezó a pensar en soluciones que apoyadas en la tecnología, que facilitara el proceso de enseñanza.

Se comenzó a concebir una solución tecnológica que además de entregar un curso virtual, tuviera herramientas comunicativas como chat, foros, wikis, intercambio de contenido, entre otros. Adicionalmente, era necesario que esta solución cumpliera con un estándar que le permitiera ser manipulada desde cualquier entorno. A partir de esto se pensó en SCORM como estándar para el manejo de contenidos.

En esta parte del proceso nos percatamos que desarrollar desde cero un aplicativo que tuviera estas características era una tarea innecesaria, pues existían varias plataformas en el mercado que ofrecían de manera gratuita los atributos comunicativos e interactivos solicitados por el centro de idiomas; utilizar uno de ellos y adicionarle el curso de mentoría virtual era la mejor alternativa. Al analizar las diferentes plataformas, concluimos que las que mejor se adaptaban eran EAFIT Interactiva y Moodle. Finalmente fue seleccionada EAFIT Interactiva por ser una herramienta institucional y por el soporte que se podía ofrecer al utilizar la misma. Después de definir claramente la solución, se creó una planificación que detallaba las actividades que debían cumplirse y los tiempos para cada una de ellas. Se construyeron los documentos de requisitos y casos de uso y se modeló formalmente el problema y la solución.

Implementación

En esta etapa del proceso se comenzó con la construcción en sí del sistema. Se desarrollaron cada una de las actividades planificadas, asumiendo los roles definidos.

🕒 **Etapas finales**

Se realizaron pruebas unitarias de la aplicación y se prosiguió a documentar el sistema, construyendo el presente documento que explica detalladamente los elementos presentes en el software.

6.2. ¿POR QUÉ SCORM?

La principal razón para elegir SCORM como estándar de manejo de contenido e-learning en nuestro proyecto de grado es que SCORM agrupa los principales estándares y especificaciones relacionadas con contenido e-learning propuestas anteriormente por AICC, IEEE e IMS por lo tanto es altamente completo y permite más funcionalidades. Al mismo tiempo, este estándar está muy bien documentado y por lo tanto se hace más fácil su investigación. Para tomar la decisión de utilizarlo, investigamos algunas razones por las cuales SCORM es elegido como estándar y encontramos las siguientes ventajas:

“Agrupa un conjunto de estándares y especificaciones adaptadas de múltiples fuentes para lograr una descripción exhaustiva de características del e-learning referidas a contenidos.

- 🕒 Es el más utilizado en el e-learning.
- 🕒 Su filosofía es crear objetos de aprendizaje totalmente independientes:
 - 📁 De la plataforma e-learning, es decir, interoperables.
 - 📁 De su almacenamiento, es decir, accesibles.
 - 📁 En cuanto a su contenido, es decir, reutilizables”⁽¹⁵⁾

- ⌚ “Se puede mezclar y unir contenido de diferentes fuentes sin preocuparse por incompatibilidades técnicas.
- ⌚ El contenido obsoleto puede ser archivado en un documento estándar bien formado.
- ⌚ Disminuye costo y tiempo para desplegar una solución basada en e-learning.
- ⌚ SCORM define un conjunto mínimo de metadatos, haciendo más práctico construir catálogos de contenido independientemente de donde provengan estos”⁽¹⁶⁾

Apoyados en las anteriores ventajas consideramos que el estándar SCORM es una buena alternativa para el manejo del contenido e-learning en la universidad EAFIT y el centro de idiomas. Adicionalmente, consideramos que tiene una gran ventaja al adaptarse a EAFIT Interactiva porque tanto dicha plataforma como las soluciones de implementación de códigos abiertos que sirven como base y ejemplo están desarrollados en el mismo lenguaje de programación (Java), haciendo el proceso de integración mucho más fácil.

6.3. ¿POR QUÉ EAFIT INTERACTIVA?

Cuando el Centro de Idiomas expresó su necesidad de utilizar un sistema que permitiera a los docentes recibir una capacitación de forma virtual, al mismo tiempo que se posibilitaba una interacción continua, donde el intercambio de información y la comunicación fuera permanente, inmediatamente se pensó en la utilidad de herramientas tales como chat, foro, intercambio de archivos, entre otras. Para el centro de idiomas era de vital importancia que todo lo anterior estuviera integrado y de esta forma la aplicación podría traer beneficios tanto para

docentes nuevos que no habían recibido capacitaciones como para los docentes antiguos que necesitaban seguir recibiendo información y retroalimentaciones.

Desarrollar un sistema desde cero que incorporará las herramientas de comunicación antes mencionadas era una tarea además de tediosa, innecesaria, pues son muchas las plataformas que ofrecen este tipo de servicios de forma gratuita y pueden ser modificadas según las necesidades específicas.

Al investigar sobre las plataformas que se podían acomodar mejor a los requisitos del centro de idiomas, llegamos a la conclusión que la mejor solución era la plataforma utilizada dentro de la universidad para el apoyo virtual a la docencia. EAFIT Interactiva contiene todos los módulos necesarios para mantener un ambiente virtual óptimo de aprendizaje (ofrece funcionalidades en: información, comunicación, propuesta docente y herramientas del profesor) y al ser una aplicación institucional, el centro de idiomas tendría soporte por parte del centro de informática cuando se presentara algún problema y las modificaciones futuras serían más fáciles de implementar. Por lo anterior optamos por incorporar una opción para cargar cursos virtuales de tipo SCORM y de esta forma todo el proceso de capacitación virtual estaría integrado.

En la actualidad EAFIT Interactiva no hace parte de los cursos impartidos por el centro de idiomas, sin embargo esta empezará a funcionar en un futuro cercano.

6.4. DISEÑO DEL CURSO

Para el diseño utilizamos una tabla basada en el lesson plan de los cursos del Centro de Idiomas y así tener un guión prediseñado. Esto permite desarrollar de una forma más sencilla la parte gráfica del curso, para nuestro caso sólo desarrollamos el primer ítem que corresponde al primer día de clases.

Ítem	Objeto de aprendizaje	Assets	Storyboard
<p>Primer día de clase</p>	<p>Protocolo</p> <p>Objetivo: familiarizar al docente con la información que debe suministrar a los estudiantes el primer día de clase.</p>	<ol style="list-style-type: none"> 1. Guía del primer día de clase 2. Lista de clase 3. Autorizaciones y permisos temporales 4. Logros del curso 5. Compromisos del estudiante 6. Manual de convivencia (para el programa de niños y adolescentes) 7. Calendario de trabajo (fechas de parciales y finales) 	<p>Ambientación:</p> <p>Survey: una lista de actividades a hacer el primer día de clase. El docente debe chequear las que hace, basándose en sus creencias.</p> <p>Presentación:</p> <p>Comics: 8 historietas con situaciones típicas correctas e incorrectas de primer día de clase (profesor que inicia clase sin hacer presentación del curso, profe que no lleva los documentos necesarios, profe que no chequea que los estudiantes estén en la lista, etc). El docente lee cada comic y lo clasifica en un cuadro de “yo haría...” o “yo no haría...” al terminar de clasificar todos los comics hace clic en “chequear” y los comics correctamente clasificados se ponen en sombra, mientras que los que no, se iluminan y sale un pop-up</p>

			<p>aclarando por qué no está bien hacer o no la actividad seleccionada.</p> <p>Actividad evaluativa: Cuestionario para ser enviado on-line al mentor virtual.</p>
<p>Planeación de clases</p>	<p>Plan de clase</p> <p>Objetivo: familiarizar al docente con los puntos a tener en cuenta al planear su clase</p>	<ol style="list-style-type: none"> 1. Formato de planeación de clase 2. Ejemplo de formato diligenciado 	<p>Ambientación: Survey: creencias sobre plan de clase: importante? Lo uso? Cómo planeo? Reviso syllabus? Me baso en act. Del libro?</p> <p>Presentación: Análisis de lp, identificación de cada sección: nombrar cada una. Pop-ups explicativos.</p> <p>Actividad guiada: Clasificar varias entries, según sean indicadores de logro, objetivos de clase, ambientación, input, actividad guiada y actividad de closure. Check & feedback.</p>

			<p>Actividad libre: Crear un plan de clase utilizando varias opciones dadas. Check.</p> <p>Actividad evaluativa: Enviar un plan de clase al mentor virtual, quien dará feedback.</p>
	<p>Uso del tablero</p> <p>Objetivo: familiarizar al docente con las estrategias a tener en cuenta con el uso efectivo del tablero en la enseñanza de idiomas</p>	<ol style="list-style-type: none"> 1. Dibujo de uso indebido del tablero 2. Dibujo de uso efectivo del tablero 3. Dibujo con posición del docente frente al tablero. 	<p>Ambientación: Comparación de dos tableros. Cuál te parece más efectivo? Por qué?</p> <p>Presentación: Técnicas de uso efectivo</p> <p>Actividad guiada: Encuentra los problemas en el uso de este tablero y cómo los solucionarías.</p> <p>Actividad libre:</p>

Manejo de clase			<p>Action research: autoevaluación del uso del tablero. Enviar comentarios al mentor virtual.</p> <p>Actividad evaluativa: Observación de clase por parte del mentor.</p>
	<p>Asignación de turnos</p> <p>Objetivo: familiarizar al docente con los aspectos a tener en cuenta al asignar turnos en la participación de sus estudiantes.</p>	<ol style="list-style-type: none"> 1. Video de elicitación no adecuada 2. Video de elicitación adecuada 	<p>Ambientación: Video de elicitación no adecuada. Por qué?</p> <p>Presentación: Técnicas de elicitación</p> <p>Actividad guiada: Análisis de los videos iniciales y proponer maneras de mejorar la elicitación.</p> <p>Actividad libre: Action research: durante una semana, ser conciente de la manera en que se dan turnos a los estudiantes y hacer una autoevaluación con base en esto. Hacer una lista de los</p>

			<p>aspectos a mejorar y trazarse un plan de trabajo con este objetivo. Enviar al mentor virtual.</p> <p>Actividad evaluativa: Observación de clase por parte del mentor.</p>
	<p>Técnicas de agrupación</p> <p>Objetivo: familiarizar al docente con las estrategias para agrupar a los estudiantes de acuerdo al tipo de actividad a realizar y al objetivo que se persigue.</p>	<ol style="list-style-type: none"> 1. Diagrama de trabajo individual 2. Diagrama de parejas 3. Diagrama de trabajo en pequeños grupos 4. Diagrama de dos grandes grupos 5. Diagrama de grupo completo 	<p>Ambientación: Survey: cómo agrupas a tus estudiantes? Check distintas opciones.</p> <p>Presentación: Análisis de técnicas según tipo de actividad</p> <p>Actividad guiada: Matching: agrupar distintas actividades en el cuadro que representa la agrupación más adecuada.</p> <p>Actividad libre: Dar ejemplos de actividades apropiadas para</p>

			<p>cada tipo de agrupación. Enviar al mentor virtual.</p> <p>Actividad evaluativa: Multiple choice: distintas actividades con múltiple escogencia de agrupaciones. Evaluación inmediata.</p>
	<p>Lenguaje corporal</p> <p>Objetivo: familiarizar al docente con el impacto que tiene su lenguaje corporal en el aprendizaje de los estudiantes.</p> <p>Competencia: actitudinal</p>	<p>Video-clips</p>	<p>Ambientación: Video-clips. Cuál se parece más a ti?</p> <p>Presentación: Tips y técnicas de cómo usar el lenguaje corporal. Incluye videos, comics y/o dibujos.</p> <p>Actividad guiada: What's wrong? Cómo lo mejoraría?</p> <p>Actividad libre: Autoevaluación: qué debo mejorar? A qué le</p>

			<p>debo prestar especial atención? Para enviar al mentor virtual.</p> <p>Actividad evaluativa: Observación de clase por parte del mentor.</p>
	<p>Manejo del tiempo</p> <p>Objetivo: familiarizar al docente con técnicas que le permitan optimizar el tiempo en clase.</p>	<p>Test de manejo de tiempo</p>	<p>Ambientación: Survey: cuánto tiempo utilizo en?</p> <p>Presentación: Tips y técnicas para optimizar el uso del tiempo.</p> <p>Actividad guiada: Analizar un lesson plan y asignarle el tiempo que le dedicaría. Check & feedback inmediato.</p> <p>Actividad libre: Planear una clase y asignar el tiempo a cada actividad. Check al terminar si se hizo de</p>

			<p>acuerdo al plan o fue irrealista. Razones para no cumplir con el tiempo planeado.</p> <p>Actividad evaluativa: Observación de clase por parte del mentor. Se debe presentar el plan de clase con los tiempos.</p>
	<p>Estilos de aprendizaje</p> <p>Objetivo: comprender los distintos estilos de aprendizaje y su impacto en el aula.</p>	<ol style="list-style-type: none"> 1. Presentación power point 2. Test de estilos de aprendizaje 3. Listado de actividades asociadas al estilo visual 4. Listado de actividades asociadas al estilo auditivo 5. Listado de actividades asociadas al estilo kinestésico 	<p>Ambientación: Autotest: como aprendo yo?</p> <p>Presentación: Presentación power point con las principales características de los tipos de aprendizaje en sus tres diferentes clasificaciones: - visual, auditiva, kinestésica - global, analítica -</p> <p>Actividad guiada: Matching: diferentes actividades para el estilo</p>

			de aprendizaje correspondiente. Actividad evaluativa: Escoger un tema para uno de sus cursos y diseñar una actividad que sea apropiada para cada estilo de aprendizaje. Enviar al mentor virtual.
	<p>Evaluación escrita</p> <p>Objetivo: comprender el sistema evaluativo del componente escrito del ci.</p>	<ol style="list-style-type: none"> 1. Rúbrica evaluación escrita cursos básicos 2. Rúbrica evaluación escrita cursos intermedios. 3. Ejemplo evaluación escrita parcial (en blanco) 4. Ejemplo evaluación escrita final (en blanco) 5. Convenciones para 	<p>Ambientación:</p> <p>Survey: creencias de la evaluación escrita. Qué incluyo? Como la diseño? Cuantas actividades evaluativas hago para el seguimiento? Como las evalúo? Que tengo en cuenta? (ortografía, gramática, uso de vocabulario, redacción, etc)</p> <p>Puzzle: rúbrica.</p> <p>Presentación:</p> <p>Muestra de composición en el idioma que se enseña y cómo se utilizan las convenciones y la rúbrica.</p>

<p>Evaluación</p>		<p>inglés</p> <p>6. Convenciones para alemán</p> <p>7. Convenciones para italiano</p> <p>8. Convenciones para portugués</p> <p>9. Convenciones para español</p> <p>10. Ejemplo de cómo aplicar las convenciones</p>	<p>Actividad guiada:</p> <p>Otra muestra de composición para ser evaluada por el docente, y comparada con lo propuesto.</p> <p>Actividad libre:</p> <p>Docente envía una composición evaluada de alguno de sus estudiantes al mentor. Feedback.</p>
	<p>Evaluación oral</p> <p>Objetivo: comprender el sistema de evaluación de la habilidad oral en el ci.</p>	<p>1. Rúbrica evaluación oral cursos básicos.</p> <p>2. Rúbrica evaluación oral cursos intermedios.</p> <p>3. Video evaluación oral en inglés</p> <p>4. Video evaluación oral en alemán</p>	<p>Ambientación:</p> <p>Survey: creencias de la evaluación oral. Qué incluyo? Como la diseño? Cuantas actividades evaluativas hago para el seguimiento? Como las evalúo? Que tengo en cuenta? (pronunciación, claridad, fluidez, corrección gramatical, etc)</p> <p>Puzzle: rúbrica evaluación oral</p>

		<p>5. Video evaluación oral en italiano</p> <p>6. Video evaluación oral en portugués</p> <p>7. Video evaluación oral en español</p>	<p>Presentación: Video de evaluación oral y como es evaluado.</p> <p>Actividad guiada: Otro video de evaluación oral para ser evaluada por el docente, y comparada con lo propuesto.</p> <p>Actividad libre: El docente verá un video y evaluará al estudiante. Enviará el resultado al mentor virtual, quien le dará retroalimentación.</p>
	<p>Retroalimentación</p> <p>Objetivo: comprender los aspectos a tener en cuenta durante el proceso de retroalimentación.</p>	<p>1. Video: ejemplo de retroalimentación no efectiva</p> <p>2. Video: ejemplo de retroalimentación efectiva</p>	<p>Ambientación: Comics: historietas de distintas situaciones (correctas e incorrectas) para ser clasificadas por el docente.</p> <p>Presentación: Guía para dar retroalimentación.</p>

			<p>Actividad guiada: Según los videos evaluados en la parte de evaluación oral, matching de cómo daría retroalimentación.</p> <p>Actividad libre: Como daría retroalimentación al estudiante del video de la actividad libre. Para enviar al mentor virtual.</p>
	<p>Reporte de notas Objetivo: familiarizarse con el procedimiento de reporte de notas al finalizar un curso.</p>	<ol style="list-style-type: none"> 1. Planilla de reporte de notas física 2. Informe de notas para estudiante 3. Presentación power point con sonido explicando los pasos para el reporte de notas en el sistema. 	<p>Ambientación: Facturas de restaurantes incompletas, facturas de tarjetas de crédito sin discriminación por compra, etc. Planillas de notas incorrectamente diligenciadas. Qué falta?</p> <p>Presentación: Planilla de notas correctamente diligenciada y reporte de acadi correcto.</p>

			<p>Actividad guiada: Paso a paso en power point de cómo ingresar las notas en el sistema.</p> <p>Actividad libre: Práctica en el sistema y en formato de planilla de notas.</p> <p>Actividad evaluativa: Informe del coordinador de cumplimiento de los reportes durante el semestre.</p>
--	--	--	--

Para el diseño del curso tuvimos la ayuda de un diseñador gráfico lo que nos permitió lograr que la interfaz gráfica fuera más atractiva para los docentes del Centro de Idiomas. A continuación se presenta el diseño de cómo va a quedar el curso para el primer ítem de la tabla anterior,

Ambientación

Esta pantalla es la página de inicio al sistema, muestra una puerta para entrar al nuevo curso.



Figura 4. Página de ingreso al curso

En esta pantalla el docente debe seleccionar de una lista de ítems predefinidos las situaciones que considera deben ser las adecuadas en el primer día de clase.



Figura 5. Página para seleccionar los ítems

🕒 Presentación

Esta pantalla muestra unos comics con las situaciones típicas que se presentan el primer día de clase, el estudiante debe clasificar estas historietas como correctas o incorrectas de acuerdo a la teoría mostrada en la ambientación de este ítem.



Figura 6. Ejemplo de comic



Figura 7. Ejemplo de comic

🕒 Actividad evaluativa

Esta pantalla se encarga de realizar la evaluación al estudiante que se encuentra realizando el curso, en este caso debe seleccionar la respuesta que corresponde a la pregunta que se ubica en la casilla grande, luego debe ir a la pregunta siguiente y continuar con el mismo procedimiento.

PAREJAS
MARCAR PARA BORRAR ✖ EDITAR ESTA TARJETA ✎

Visualiza el ítem en la casilla grande y selecciona la respuesta que concuerda, desde las casillas más pequeñas. Pulsa 'Siguiente Pregunta' para continuar

Es la tercera clase y todavía algunos de los estudiantes no aparecen en la lista

Correcto: 0 Error: 1
2 de 11

Rubricas, silabo del curso, convenciones escritas y plan de trabajo	Reportar para que se divida el curso	Le presto mas atención para verificar que se encuentre en el nivel adecuado
Rubricas de evaluación oral y escrita, silabo del curso, convenciones escritas	Voy al centro de recursos	Plan de trabajo
Los mando a la oficina 201	Sigo el protocolo normal y pongo al día a los estudiantes que no asistieron	Rubricas de evaluación oral y escrita

[Siguiente Pregunta](#)
[Mostrar Respuesta](#)
[Reiniciar](#)
[Ayuda](#)




Figura 8. Ejemplo de la actividad evaluativa

7. DESARROLLO DEL PROTOTIPO

7.1. INFORMACIÓN TÉCNICA DEL PROTOTIPO

Información técnica del prototipo	
Tipo de prototipo	Web
Lenguaje de programación	Java
Base de datos	Oracle 9i
Servidor web	Caucho Resin
Servidor de aplicaciones	Caucho Resin
Sistema operativo	Solaris 10

7.2. FASE DE INICIO

7.2.1. Captura de requisitos

7.2.1.1. Requisitos de almacenamiento de información

IRQ-001	Almacenamiento histórico de sesión
Versión	1.0
Autores	<ul style="list-style-type: none"> ○ Ana Sofía Gallo Vargas ○ Luisa Fernanda Hurtado Jaramillo

Fuentes	Ana María Botero Córdoba
Dependencias	Ninguno
Descripción	El sistema deberá <i>almacenar la información relacionada con las actividades y módulos a los que el docente accedió</i>
Importancia	Vital
Urgencia	puede esperar
Estado	pendiente de verificación
Estabilidad	Baja
Comentarios	Ninguno

7.2.1.2. Requisitos funcionales

FRQ-0001	Interacción con Herramientas Multimedia
Versión	1.0
Autores	Ana Sofía Gallo Vargas Luisa Fernanda Hurtado Jaramillo
Fuentes	Ana María Botero Córdoba
Dependencias	Ninguno
Descripción	El sistema deberá <i>permitir que los docentes interactúen con herramientas multimedia tales como video y audio.</i>
Importancia	Importante
Urgencia	inmediatamente
Estado	Validado
Estabilidad	Alta

Comentarios	Ninguno
--------------------	---------

FRQ-0002	Consultas
Versión	1.0
Autores	<ul style="list-style-type: none"> ○ Ana Sofía Gallo Vargas ○ Luisa Fernanda Hurtado Jaramillo
Fuentes	<ul style="list-style-type: none"> ○ Ana María Botero Córdoba
Dependencias	Ninguno
Descripción	El sistema deberá <i>permitir hacer consultas por nombre y por programa sobre el historial de los accesos al sistema.</i>
Importancia	Importante
Urgencia	inmediato
Estado	pendiente de verificación
Estabilidad	Media
Comentarios	Al hacer la búsqueda por el programa, se lista la frecuencia de acceso de todos los docentes pertenecientes al programa buscado.

FRQ-0003	Consultas de docentes
Versión	1.0
Autores	<ul style="list-style-type: none"> ○ Ana Sofía Gallo Vargas ○ Luisa Fernanda Hurtado Jaramillo
Fuentes	<ul style="list-style-type: none"> ○ Ana María Botero Córdoba
Dependencias	Ninguno

Descripción	El sistema deberá <i>permitir consultar los docentes activos por medio de su nombre o programa</i>
Importancia	importante
Urgencia	inmediatamente
Estado	pendiente de verificación
Estabilidad	Alta
Comentarios	Este requisito permite la consulta de los datos personales de los docentes que se encuentran actualmente laborando en el Centro de Idiomas Universidad EAFIT

FRQ-0004	Filtro de información
Versión	1.0
Autores	<ul style="list-style-type: none"> ○ Ana Sofía Gallo Vargas ○ Luisa Fernanda Hurtado Jaramillo
Fuentes	○ Ana María Botero Córdoba
Dependencias	Ninguno
Descripción	El sistema deberá <i>realizar un filtro de información de acuerdo a los permisos que tenga el usuario</i>
Importancia	importante
Urgencia	inmediatamente
Estado	pendiente de verificación
Estabilidad	Alta
Comentarios	Ninguno

FRQ-0005	Autenticación de usuario
Versión	1.0
Autores	<ul style="list-style-type: none"> ○ Ana Sofía Gallo Vargas ○ Luisa Fernanda Hurtado Jaramillo
Fuentes	○ Ana María Botero Córdoba
Dependencias	Ninguno
Descripción	El sistema deberá <i>hacer autenticación de usuario utilizando el login y clave del correo de la universidad EAFIT</i>
Importancia	Vital
Urgencia	inmediatamente
Estado	pendiente de verificación
Estabilidad	Alta
Comentarios	Ninguno

FRQ-0006	Manejar sesiones concurrentes
Versión	1.0
Autores	<ul style="list-style-type: none"> ○ Ana Sofía Gallo Vargas ○ Luisa Fernanda Hurtado Jaramillo
Fuentes	○ Ana María Botero Córdoba
Dependencias	Ninguno
Descripción	El sistema deberá <i>soportar diferentes sesiones para que varios usuarios puedan ingresar al sistema al mismo tiempo</i>
Importancia	Vital
Urgencia	Inmediato

Estado	pendiente de verificación
Estabilidad	Baja
Comentarios	Ninguno

FRQ-0007	Reinicio de sesión
Versión	1.0
Autores	<ul style="list-style-type: none"> ○ Ana Sofía Gallo Vargas ○ Luisa Fernanda Hurtado Jaramillo
Fuentes	○ Ana María Botero Córdoba
Dependencias	Ninguno
Descripción	El sistema deberá <i>permitir que el docente continúe las actividades sin tener que repetir las que ya realizó</i>
Importancia	Vital
Urgencia	Inmediato
Estado	pendiente de verificación
Estabilidad	Baja
Comentarios	Ninguno

FRQ-0008	Comunicación entre docentes
Versión	1.0
Autores	<ul style="list-style-type: none"> ○ Ana Sofía Gallo Vargas ○ Luisa Fernanda Hurtado Jaramillo
Fuentes	○ Ana María Botero Córdoba
Dependencias	Ninguno

Descripción	El sistema deberá <i>soportar chats, foros, wikis para que los docentes compartan la información pertinente</i>
Importancia	Vital
Urgencia	Inmediato
Estado	pendiente de verificación
Estabilidad	Baja
Comentarios	Ninguno

7.2.1.3. Requisitos no funcionales

NFR-0001	Privilegios de acceso
Versión	1.0
Autores	<ul style="list-style-type: none"> ○ Ana Sofía Gallo Vargas ○ Luisa Fernanda Hurtado Jaramillo
Fuentes	○ Ana María Botero Córdoba
Dependencias	Ninguno
Descripción	El sistema deberá <i>ser accesible para los docentes, el coordinador y los asesores de idioma</i>
Importancia	vital
Urgencia	inmediatamente
Estado	pendiente de verificación
Estabilidad	alta
Comentarios	Ninguno

7.2.1.4. Requisitos de restricción

CRQ-0001	Integración con otros sistemas
Versión	1.0
Autores	<ul style="list-style-type: none"> ○ Ana Sofía Gallo Vargas ○ Luisa Fernanda Hurtado Jaramillo
Fuentes	<ul style="list-style-type: none"> ○ Ana María Botero Córdoba
Dependencias	Ninguno
Descripción	La información almacenada por el sistema deberá satisfacer la siguiente restricción: <i>integrarse con el sistema EAFIT Interactiva de la Universidad EAFIT</i>
Importancia	vital
Urgencia	inmediatamente
Estado	pendiente de validación
Estabilidad	alta
Comentarios	Ninguno

7.3. FASE DE ELABORACIÓN

7.3.1. Modelo de casos de uso

A continuación se muestran los casos de uso que hacen parte del sistema y aquellos con los que tiene interacción de la plataforma EAFIT Interactiva.

7.3.1.1. Especificación del caso de uso: Ingresar al sistema

🕒 Descripción

Este caso de uso es incluido por otros casos de uso cuando se desea realizar el proceso de validación e ingreso de usuarios a EAFIT interactiva a través de su correo electrónico y contraseña de la Universidad EAFIT.

🕒 Diagrama del caso de uso

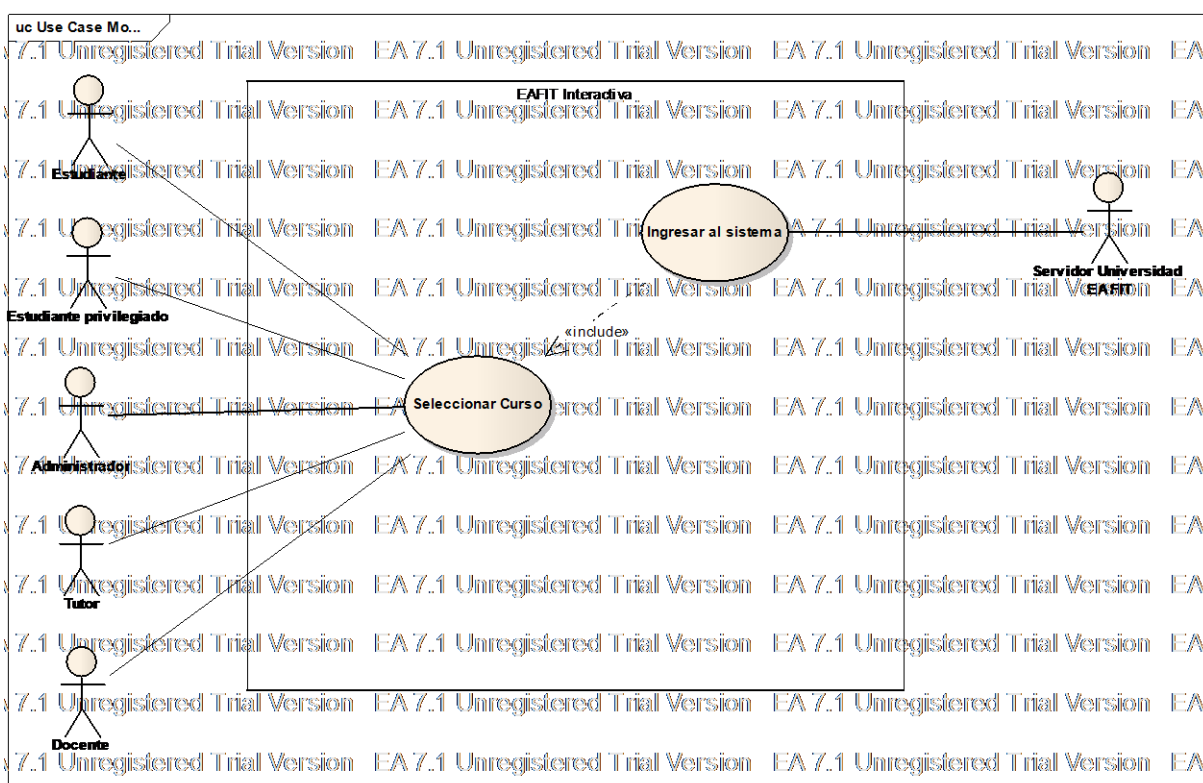


Figura 10. Diagrama del caso de uso ingresar al sistema

🕒 Precondiciones

El usuario debe estar vinculado a la Universidad EAFIT.

🕒 Flujo básico

1. El caso de uso comienza cuando el usuario abre la ventana de inicio del sistema.
2. El sistema solicita el usuario y la clave.
{Ingresar datos}
3. El usuario ingresa el usuario y la clave.
4. El sistema envía la información al servidor de la Universidad EAFIT.
5. El servidor valida si el usuario y la clave son correctas e informa al sistema.
{Usuario incorrecto}
6. Incluir subflujo 6.1. Iniciar sesión.
7. El caso de uso termina.

Flujos alternativos

Ingreso inválido

En {Usuario incorrecto} el sistema notifica al usuario que el usuario y/o contraseña ingresada no es válida.

1. El sistema muestra un mensaje notificando al usuario que los datos de ingreso no son validos.
2. El usuario selecciona la opción de regresar.
3. Continuar en {Ingresar datos}

Subflujos

Iniciar sesión

1. El sistema determina el tipo de usuario que ingresa al sistema.
2. El sistema muestra la información correspondiente a cada usuario de acuerdo a su perfil.

🕒 **Postcondiciones**

El usuario permanece en el sistema.

7.3.1.2. Especificación del caso de uso: Seleccionar curso

🕒 **Descripción**

Este caso de uso describe como EAFIT Interactiva despliega a los usuarios docente, tutor, estudiante privilegiado y estudiante los diferentes cursos que tienen asignados.

🕒 **Diagrama del caso de uso**

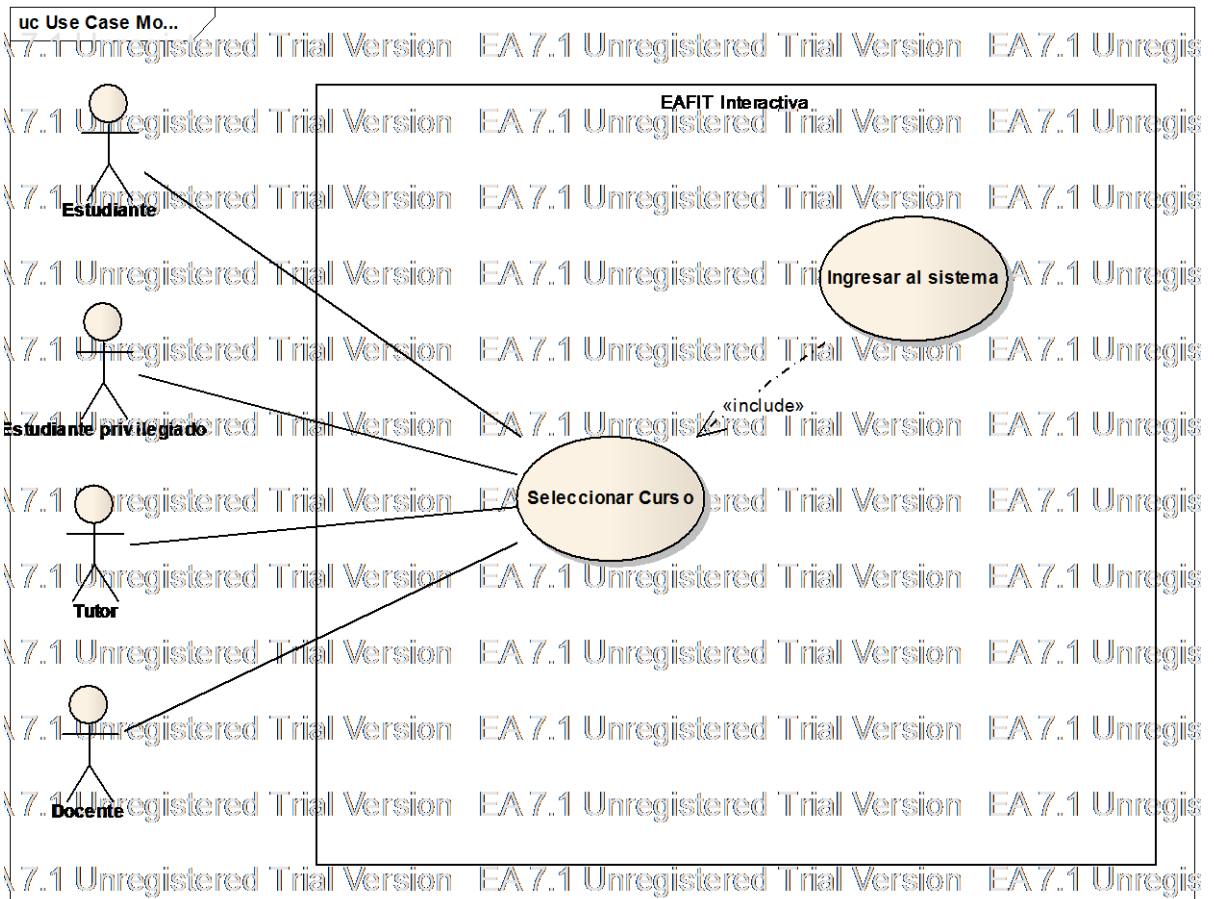


Figura 11. Diagrama del caso de uso seleccionar curso

🕒 Precondiciones

El usuario debe haber ingresado al sistema.

🕒 Flujo básico

1. El caso de uso comienza cuando el usuario se autentica en el sistema.
2. El sistema busca los cursos que están asociados al usuario que ingresó al sistema.
{Usuario sin cursos}
3. El sistema muestra un enlace para cada curso asociado.
4. El caso de uso termina.

🕒 **Flujos alternativos**

Usuario sin cursos

En {Usuario sin cursos} el sistema notifica al usuario que no se tiene cursos/materias disponibles

1. El sistema muestra un mensaje notificando al usuario que no hay cursos/materias disponibles.
2. El sistema reenvía al usuario a la página de inicio del sistema.
3. El caso de uso aborta

🕒 **Postcondiciones**

El usuario permanece en el sistema.

7.3.1.3. Especificación del caso de uso: Gestionar curso

🕒 **Descripción**

Este caso de uso describe como el usuario administrador crea, modifica, inactiva y elimina los diferentes cursos (asignaturas) que pueden ser asignados a los docentes.

🕒 **Diagrama del caso de uso**

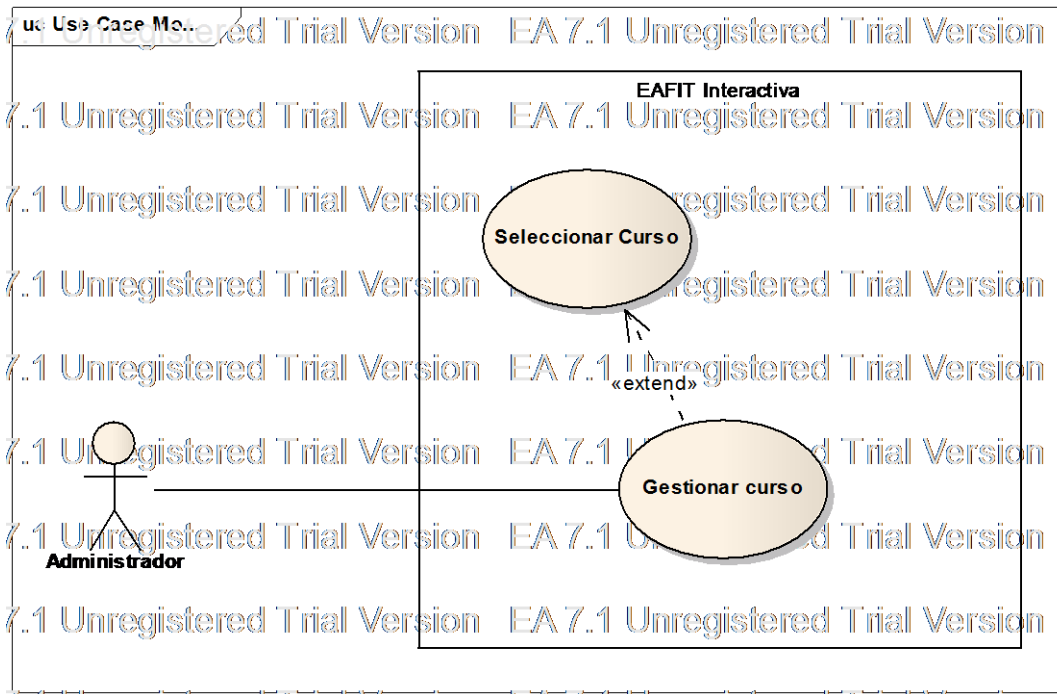


Figura 12. Diagrama del caso de uso gestionar curso

🕒 Precondiciones

El usuario debe haberse autenticado en EAFIT interactiva y debe tener permisos de administrador.

🕒 Flujo básico

1. El caso de uso comienza cuando el administrador de EAFIT interactiva desea crear, modificar o eliminar un curso.

{Crear Curso}

{Modificar Curso}

{Inactivar Curso}

{Eliminar Curso}

{Procesar cambios}

{Cancelar recurso}

2. El usuario confirma la operación
3. El sistema guarda los cambios
{Error en Procesamiento}
4. El sistema informa al usuario que la acción fue exitosa
5. El caso de uso termina.

Flujos alternativos

Crear Curso

En {Crear curso} el administrador genera un nuevo curso para ser manipulado por los demás usuarios.

1. El sistema pide el nombre del curso, tipo, usuario encargado, fechas de inicio y finalización, si es curso presencial o virtual y código.
2. El usuario ingresa los datos.
3. El sistema verifica que los datos hayan sido ingresados con éxito.
4. Si hay algún campo inválido el sistema muestra el lugar en donde se encuentra el error, solicita al usuario ingresar los datos y el usuario los ingresa.
5. Continuar en {Procesar Cambios}

Modificar Curso

En {Modificar curso} el administrador realiza cambios en los cursos previamente creados por él.

1. El sistema muestra los cursos (materias) previamente creadas.
2. El usuario selecciona el curso al que quiere aplicarle los cambios.
3. El usuario modifica los campos del curso nombre, fechas de inicio y de finalización, si es presencial o virtual.
4. El sistema verifica que los datos hayan sido ingresados con éxito.
5. Continuar en {Procesar Cambios}

Inactivar Curso

En {Inactivar curso} el administrador desactiva alguno de los cursos previamente creados por él.

1. El sistema muestra los cursos (materias) previamente creadas.
2. El usuario selecciona el curso que desea inactivar.
3. Continuar en {Procesar Cambios}

Eliminar Curso

En {Eliminar curso} el administrador borra alguno de los cursos previamente creados por él.

1. El sistema muestra los cursos (materias) previamente creadas.
2. El usuario selecciona el curso que desea eliminar.
3. El sistema verifica que el curso se encuentre vacío.
4. Continuar en {Procesar Cambios}

Postcondiciones

El usuario permanece en el sistema.

7.3.1.4. Especificación del caso de uso: Gestionar usuarios

Descripción

Este caso de uso describe como el usuario administrador crea, modifica y elimina los diferentes usuarios que va a tener el sistema.

Diagrama del caso de uso

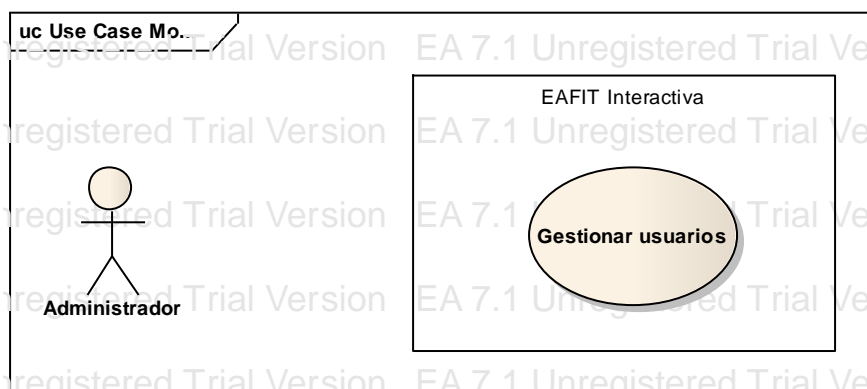


Figura 13. Diagrama del caso de uso gestionar usuarios

🕒 Precondiciones

El usuario debe haberse autenticado en EAFIT interactiva y debe tener permisos de administrador.

🕒 Flujo básico

1. El caso de uso comienza cuando el administrador desea crear, modificar o eliminar un usuario
{Crear Usuario}
{Modificar Usuario}
{Eliminar Usuario}
{Procesar cambios}
{Cancelar recurso}
2. El usuario confirma la operación
3. El sistema guarda los cambios
{Error en Procesamiento}
4. El sistema informa al usuario que la acción fue exitosa
5. El caso de uso termina.

🕒 Flujos alternativos

Crear Usuario

En {Crear usuario} el administrador genera un nuevo usuario para el sistema.

1. El sistema solicita la siguiente información del usuario que desea crearse: tipo de documento, número documento, nombre, apellidos y correo electrónico.
2. El administrador ingresa la información del usuario.
3. El sistema verifica que los datos hayan sido ingresados con éxito.
4. Continuar en {Procesar cambios}

Modificar Usuario

En {Modificar usuario} el administrador realiza cambios en alguno de los usuarios previamente creados por él.

1. El sistema muestra los usuarios previamente creados.
2. El administrador selecciona el usuario al que desea realizarle los cambios.
3. El administrador modifica los campos del usuario.
4. El sistema verifica que los datos hayan sido ingresados con éxito.
5. Continuar en {Procesar Cambios}

Eliminar Usuario

En {Eliminar usuario} el administrador borra del sistema alguno de los usuarios previamente creados por él.

1. El sistema muestra los usuarios previamente creados.
2. El usuario selecciona el usuario que desea eliminar.
3. Continuar en {Procesar Cambios}

Postcondiciones

El usuario permanece en el sistema.

7.3.1.5. Especificación del caso de uso: Gestionar recursos

🕒 Descripción

Este caso de uso describe como los usuarios que tienen permisos crean, modifican y eliminan foros, wiki, agenda, bibliografía, glosario, FAQ y enlaces dentro de EAFIT Interactiva. Si desean iniciar una conversación por medio de la herramienta chat, recibir trabajos o realizar un seguimiento de las actividades de los usuarios dentro de EAFIT Interactiva.

🕒 Diagrama del caso de uso

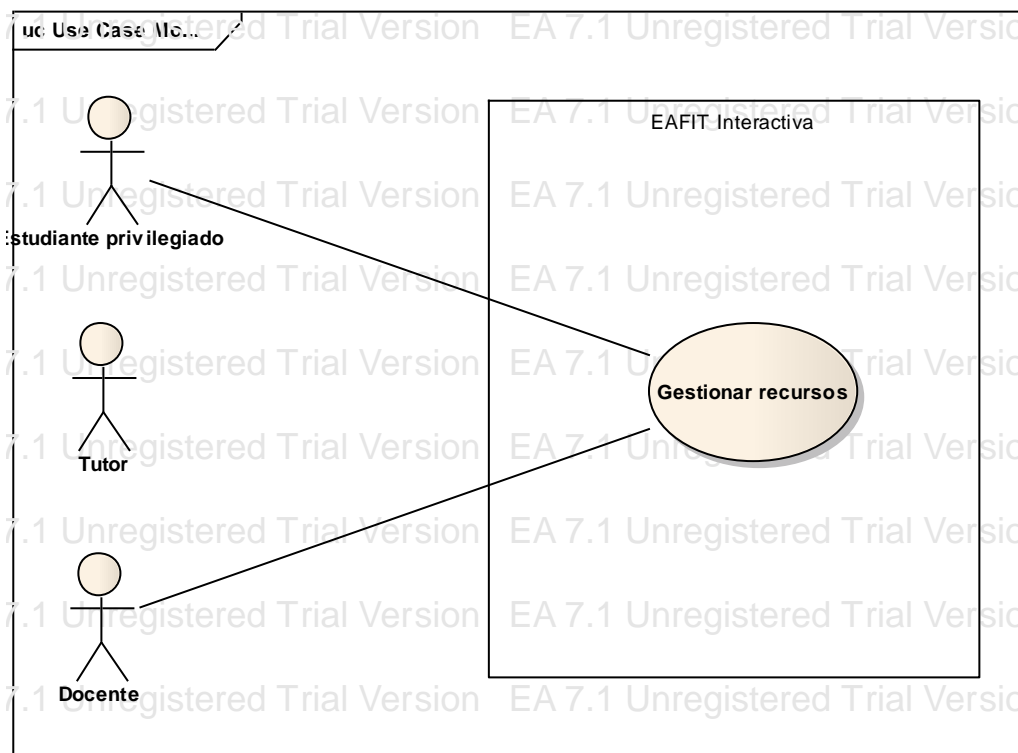


Figura 14. Diagrama del caso de uso gestionar recursos

🕒 Precondiciones

El usuario debe haber ingresado al sistema.

🕒 Flujo básico

1. El caso de uso comienza cuando el usuario desea establecer algún tipo de interacción con los recursos disponibles en la herramienta EAFIT Interactiva.
2. El sistema solicita la información necesaria de acuerdo al tipo de recurso.
 - {Enviar mensaje por mensajería instantánea}
 - {Iniciar foro}
 - {Agregar wiki}
 - {Gestionar agenda}
 - {Crear recepción trabajo}
 - {Gestionar bibliografía}
 - {Gestionar glosario}
 - {Gestionar FAQ}
 - {Gestionar enlaces}
 - {Frecuencia de uso}
 - {Recibir los datos}
 - {Recurso cancelado}
3. El usuario confirma los datos.
4. El caso de uso termina.

🕒 Postcondiciones

El usuario permanece en el sistema.

7.3.1.6. Especificación del caso de uso: Gestionar foro

🕒 Descripción

Este caso de uso describe como los usuarios que tienen permisos crean, modifican y eliminan foros dentro de EAFIT Interactiva.

🕒 Diagrama del caso de uso

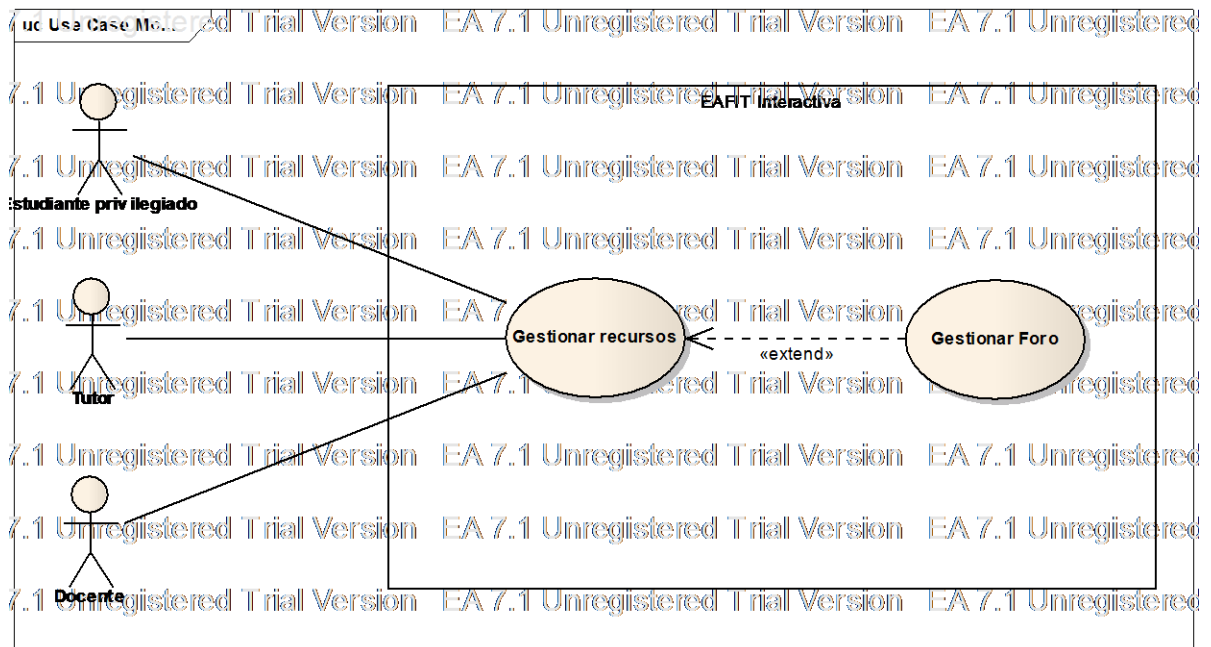


Figura 15. Diagrama del caso de uso gestionar foro

🕒 Precondiciones

El usuario debe haber ingresado al sistema.

🕒 Flujo básico

Este caso de uso extiende “Gestionar recursos” en {Iniciar foro} cuando el sistema detecta que el usuario desea crear, eliminar o modificar un foro.

{Pedir información}

1. El sistema solicita la información necesaria de acuerdo a la acción que desea realizar.

{Crear foro}

{Modificar foro}

- {Eliminar foro}
 - {Dar aporte}
 - {Archivo adjunto}
 - {Fecha inválida}
2. Continuar en {Recibir los datos}

Flujos alternativos

Crear foro

En {Crear foro} el usuario selecciona la opción de crear un nuevo foro.

1. El usuario selecciona la opción de crear nuevo foro.
2. El sistema solicita la siguiente información del foro: título, descripción, si es por equipos, si se encuentra activo, fechas de inicio y finalización.
3. El usuario ingresa la información del foro.
4. El sistema verifica que los datos hayan sido ingresados con éxito
5. Continuar en {Crear foro}.

Modificar foro

En {Modificar foro} el usuario selecciona la opción de modificar un foro ya existente.

1. El sistema muestra los foros previamente creados.
2. El usuario selecciona el foro al cual desea realizarle algún cambio.
3. El usuario modifica la información del foro.
4. El sistema verifica que los datos hayan sido ingresados con éxito
5. Continuar en {Modificar foro}.

Eliminar foro

En {Eliminar foro} el usuario selecciona la opción de eliminar un foro ya existente.

1. El sistema muestra los foros previamente creados.

2. El usuario selecciona el foro que desea eliminar.
3. Continuar en {Eliminar foro}

Dar aporte

En {Dar aporte} el usuario selecciona la opción de dar un nuevo aporte en un foro ya creado.

1. El sistema muestra los temas de foro activos.
2. El usuario selecciona el tema sobre el cual desea aportar.
3. El sistema muestra los aportes realizados por los demás usuarios.
4. El usuario selecciona la opción de ingresar nuevo aporte.
5. El sistema muestra una ventana de texto.
6. El usuario ingresa el texto.
7. Continuar en {Dar aporte}

Postcondiciones

El usuario permanece en el sistema.

7.3.1.7. Especificación del caso de uso: Crear recepción de trabajos

Descripción

Este caso de uso describe como los usuarios que tienen permisos crean un espacio dentro de EAFIT interactiva para que sus estudiantes carguen los trabajos asignados en el aula de clase.

Diagrama del caso de uso

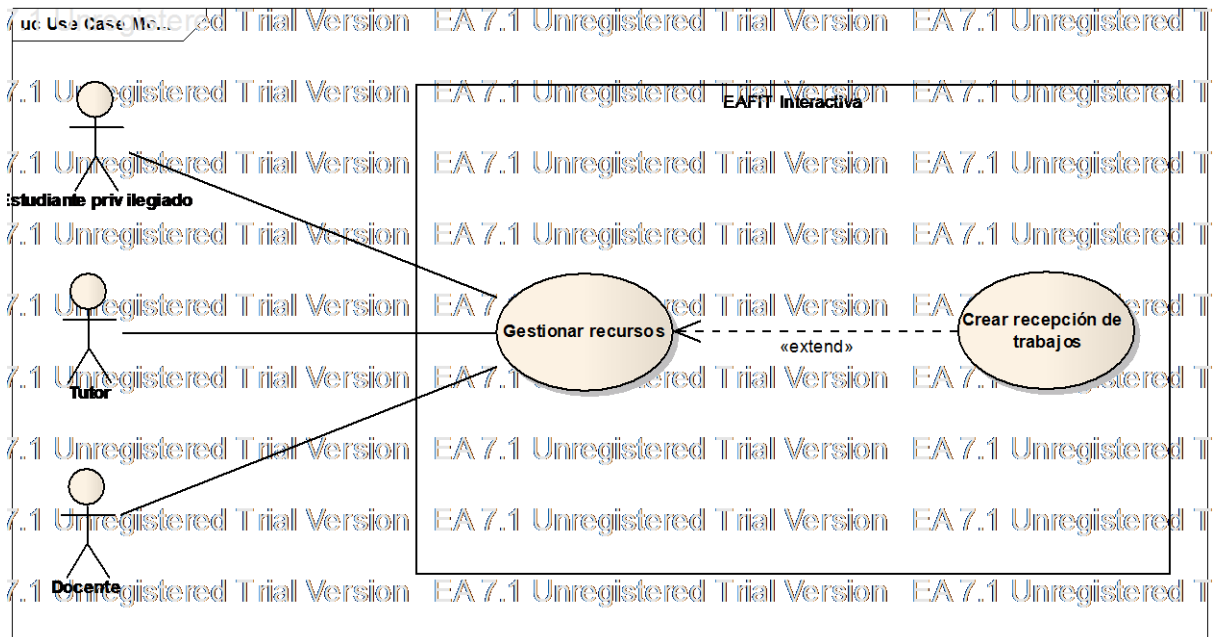


Figura 16. Diagrama del caso de uso crear recepción de trabajos

🕒 Precondiciones

El usuario debe haber ingresado al sistema.

🕒 Flujo básico

Este caso de uso extiende “Gestionar recursos” en {Crear recepción trabajo} cuando el sistema detecta que el usuario desea gestionar una entrega.

1. El sistema solicita el título del trabajo, su descripción y las fechas de inicio y finalización.
{Pedir información}
2. El usuario ingresa los datos.
{Archivo adjunto}
{Fecha inválida}
3. El usuario confirma la operación.
4. El sistema informa que la operación ha sido realizada con éxito.
5. El sistema muestra los datos de la entrega creada.

6. El sistema envía una notificación al usuario encargado de la recepción de trabajos cada vez que llegue un nuevo trabajo.
7. Continuar en {Recibir los datos}

⌚ Postcondiciones

El usuario permanece en el sistema.

7.3.1.8. Especificación del caso de uso: Iniciar conversación en chat

⌚ Descripción

Este caso de uso describe como los usuarios que tienen permisos crean una interacción de tipo chat en la plataforma EAFIT interactiva.

⌚ Diagrama del caso de uso

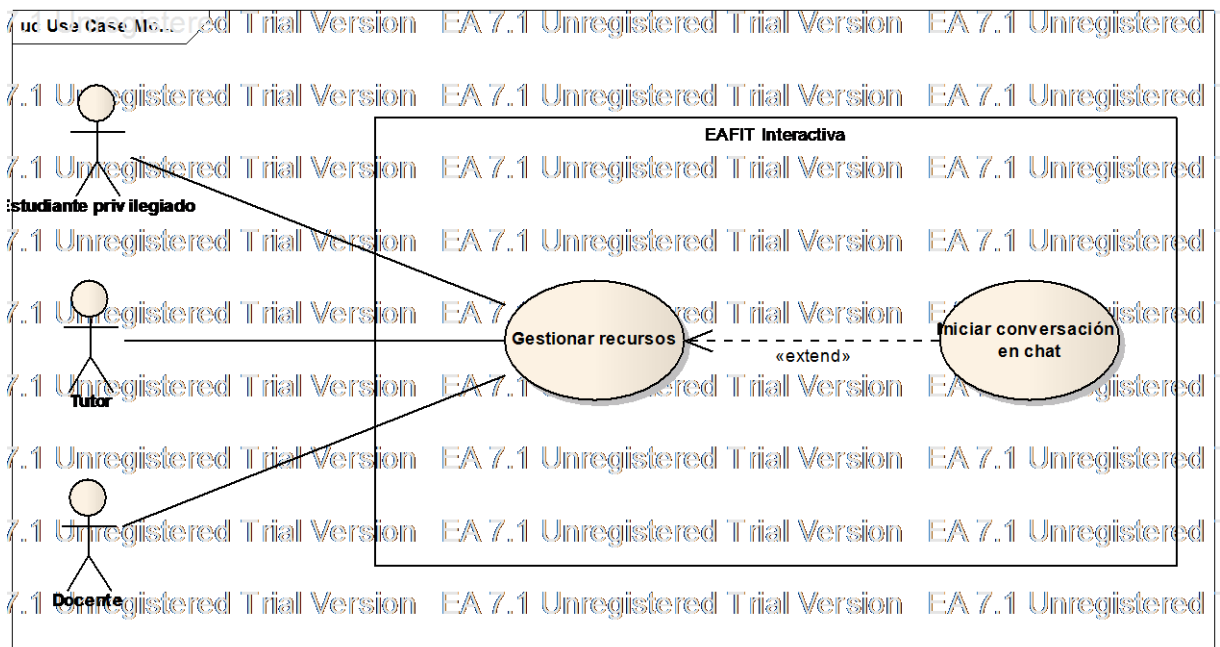


Figura 17. Diagrama del caso de uso iniciar conversación en chat

🕒 **Precondiciones**

El usuario debe haber ingresado al sistema.

🕒 **Flujo básico**

Este caso de uso extiende “Gestionar recursos” en {Enviar mensaje por mensajería instantánea} cuando el sistema detecta que el docente o el tutor desea crear un espacio para el intercambio de mensajes instantáneos.

1. El sistema pide el nombre de la interacción, su descripción, el número máximo de usuarios y las fechas de inicio y finalización.
{Pedir información}
2. El usuario ingresa los datos
3. El usuario confirma la operación.
{Fecha invalida}
{Descripción requerida}
4. El sistema muestra los datos de la entrega creada.
5. Continuar en {Recibir los datos}

🕒 **Postcondiciones**

El usuario permanece en el sistema.

7.3.1.9. Especificación del caso de uso: Gestionar wiki

🕒 **Descripción**

Este caso de uso describe como los usuarios que tienen permisos crean un nuevo elemento de tipo wiki para un curso.

🕒 **Diagrama del caso de uso**

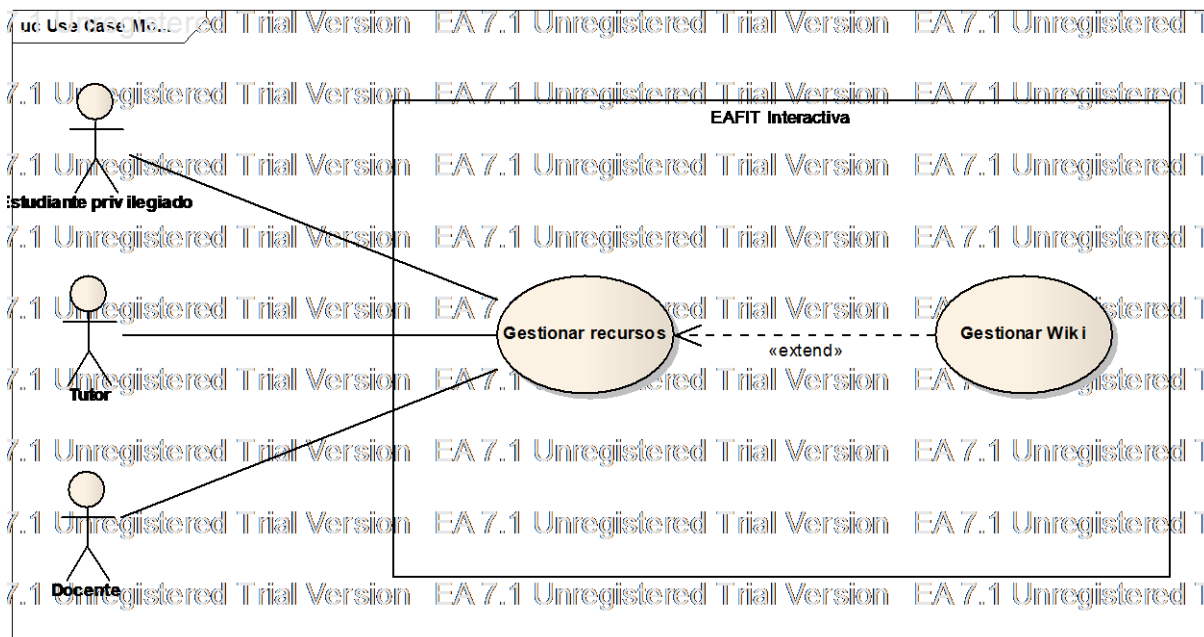


Figura 18. Diagrama del caso de uso gestionar wiki

🕒 Información adicional

Este caso de uso se encuentra implementado en (22).

7.3.1.10. Especificación del caso de uso: Gestionar agenda

🕒 Descripción

Este caso de uso describe como los usuarios que tienen permisos crean, modifican y eliminan actividades en el calendario de un curso específico.

🕒 Diagrama del caso de uso

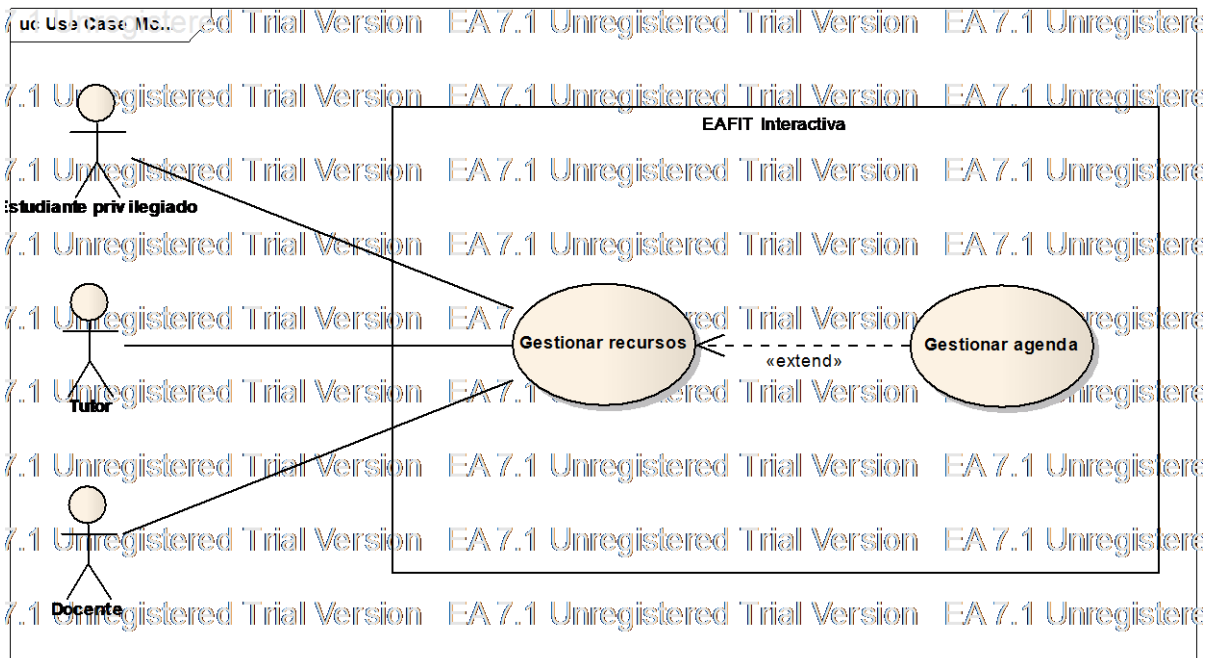


Figura 19. Diagrama del caso de uso gestionar agenda

🕒 **Precondiciones**

El usuario debe haber ingresado al sistema.

🕒 **Flujo básico**

Este caso de uso extiende “Gestionar recursos” en {Gestionar Agenda} cuando el sistema detecta que los usuarios que tienen permisos desean crear una nueva actividad para los demás usuarios que hacen parte del curso.

1. El sistema pide el título de la actividad, su descripción, la fecha de inicio y fin y la notificación de vencimiento.
{Pedir información}
2. El usuario ingresa los datos
{Archivo adjunto}
3. El usuario confirma la operación.
{Descripción requerida}
{Fecha inválida}

4. El sistema muestra un calendario mensual con todas las actividades creadas en cada día.
{Modificar actividad}
{Eliminar actividad}
5. Continuar en {Recibir los datos}

Flujos alternativos

Modificar Actividad

En {Modificar actividad} el usuario elige editar una actividad previamente creada.

1. El sistema muestra los datos actuales de la actividad.
2. El usuario realiza las modificaciones y acepta los cambios.
3. El sistema muestra los cambios.
4. El caso de uso termina.

Eliminar Actividad

En {Eliminar actividad} el usuario elige eliminar una actividad previamente creada.

1. El sistema muestra las actividades existentes.
2. El usuario selecciona la actividad que desea eliminar.
3. El sistema pide confirmar la actividad de borrado.
4. El usuario confirma que desea eliminar la actividad.
5. El sistema elimina la actividad.
6. El sistema muestra el calendario actualizado.
7. El caso de uso termina.

Postcondiciones

El usuario permanece en el sistema.

7.3.1.11. Especificación del caso de uso: Revisar frecuencia de uso

🕒 Descripción

Este caso de uso describe como el usuario docente consulta datos sobre la frecuencia de acceso y uso por parte de los demás usuarios del curso a los diferentes elementos de éste.

🕒 Diagrama del caso de uso

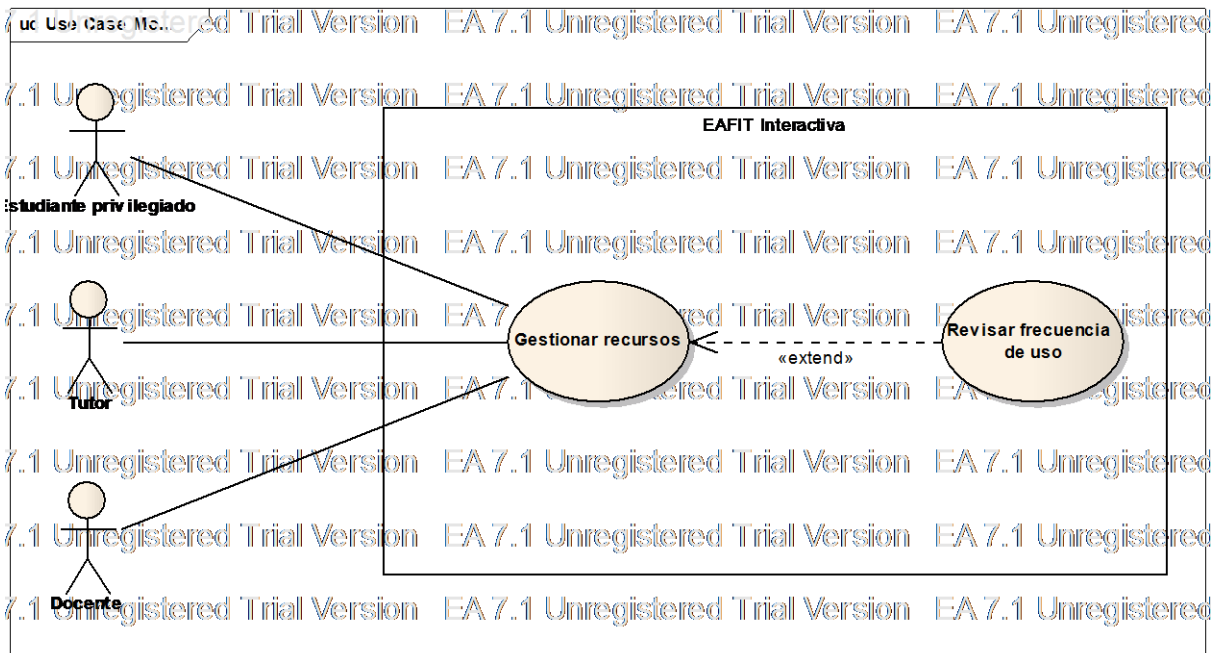


Figura 20. Diagrama del caso de uso revisar frecuencia de uso

🕒 Precondiciones

El usuario debe haber ingresado al sistema.

🕒 Flujo básico

Este caso de uso extiende “Gestionar recursos” en {Frecuencia uso} cuando el sistema detecta que el usuario desea conocer el historial de accesos de los demás usuarios del curso.

1. El sistema muestra un listado con los nombres de usuario, el número de acceso totales durante el semestre y la fecha del último acceso.
{Detalle ingresos}
{Detalle ultimas 10 sesiones}
2. Continuar en {Recibir los datos}

Flujos alternativos

Detalle ingresos

En {Detalles ingresos} el sistema muestra un listado con los ingresos clasificados por funcionalidad.

1. El docente selecciona el usuario sobre el cual desea conocer los detalles.
2. El sistema muestra el número de veces que el usuario ha ingresado a cada funcionalidad del curso y la fecha del último ingreso.
3. Continuar en {Detalles ingresos}

Detalle ultimas 10 sesiones

En {Detalle ultimas 10 sesiones} el sistema muestra un listado con los detalles de acceso de las últimas diez sesiones del usuario.

1. El docente selecciona el usuario sobre el cual desea conocer los detalles.
2. El sistema muestra un listado con la siguiente información: fecha de ingreso, dirección IP, Navegador web e idioma.
3. Si el usuario desea tener más información selecciona la fecha de ingreso de la sesión.
4. El sistema muestra fecha, funcionalidad, acción y elemento sobre el que se realizó la acción.
5. Continuar en {Detalle ultimas 10 sesiones}

🕒 **Postcondiciones**

El usuario permanece en el sistema.

7.3.1.12. Especificación del caso de uso: Gestionar bibliografía

🕒 **Descripción**

Este caso de uso describe como los usuarios que tienen permisos crean, modifican y eliminan la bibliografía de un curso dentro de EAFIT Interactiva.

🕒 **Diagrama del caso de uso**

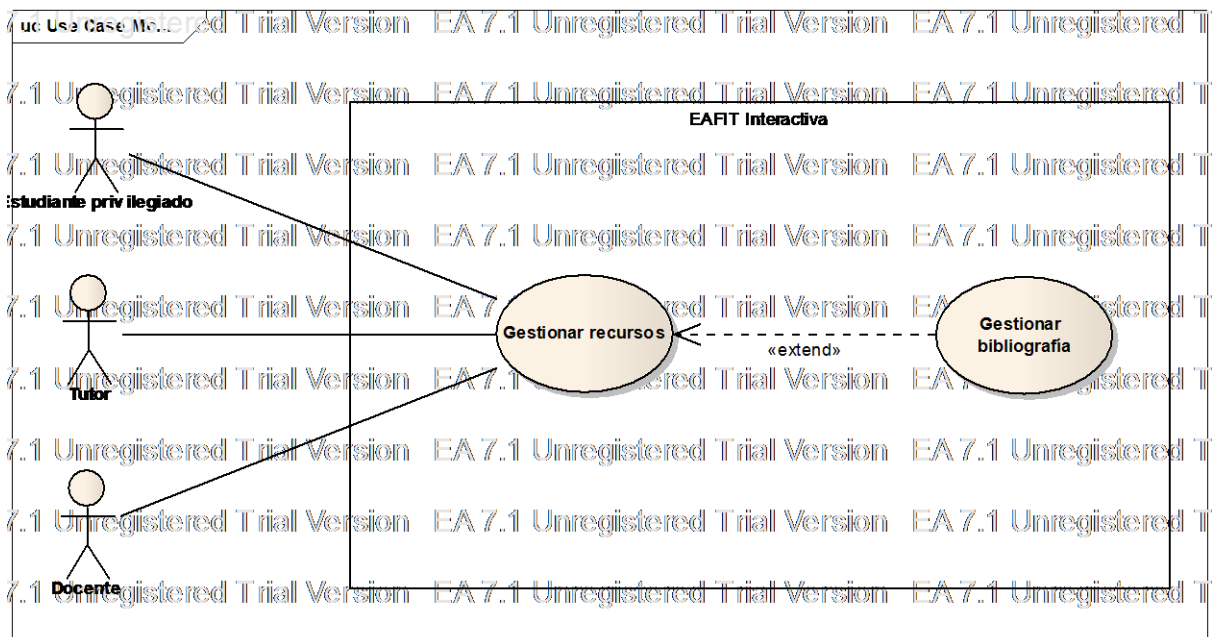


Figura 21. Diagrama del caso de uso gestionar bibliografía

🕒 **Precondiciones**

El usuario debe haber ingresado al sistema.

🕒 Flujo básico

Este caso de uso extiende “Gestionar recursos” en {Gestionar bibliografía} cuando el sistema detecta que el usuario desea crear, eliminar o modificar una bibliografía.

{Pedir información}

1. El sistema solicita la información necesaria de acuerdo a la acción que desea realizar.

{Crear bibliografía}

{Modificar bibliografía}

{Eliminar bibliografía}

{Fecha inválida}

2. Continuar en {Recibir los datos}

🕒 Flujos alternativos

Crear bibliografía

En {Crear bibliografía} el usuario selecciona la opción de crear una nueva bibliografía.

1. El usuario selecciona la opción de crear nueva bibliografía.
2. El sistema solicita la siguiente información de la bibliografía: autor, año, artículo/capítulo, revista/libro, en, editorial, datos editoriales, comentarios, activo, fechas de inicio y finalización.
3. El usuario ingresa la información de la bibliografía.
4. El sistema verifica que los datos hayan sido ingresados con éxito
5. Continuar en {Crear bibliografía}.

Modificar bibliografía

En {Modificar bibliografía} el usuario selecciona la opción de modificar un foro ya existente.

1. El sistema muestra las bibliografías previamente creadas.
2. El usuario selecciona la bibliografía a la cual desea realizarle algún cambio.
3. El usuario modifica la información de la bibliografía.
4. El sistema verifica que los datos hayan sido ingresados con éxito
5. Continuar en {Modificar bibliografía}.

Eliminar bibliografía

En {Eliminar bibliografía} el usuario selecciona la opción de eliminar una bibliografía ya existente.

1. El sistema muestra las bibliografías previamente creadas.
2. El usuario selecciona la bibliografía que desea eliminar.
3. Continuar en {Eliminar bibliografía}

Postcondiciones

El usuario permanece en el sistema.

7.3.1.13. Especificación del caso de uso: Gestionar glosario

Descripción

Este caso de uso describe como los usuarios que tienen permisos crean, modifican y eliminan un término del glosario de un curso dentro de EAFIT Interactiva.

Diagrama del caso de uso

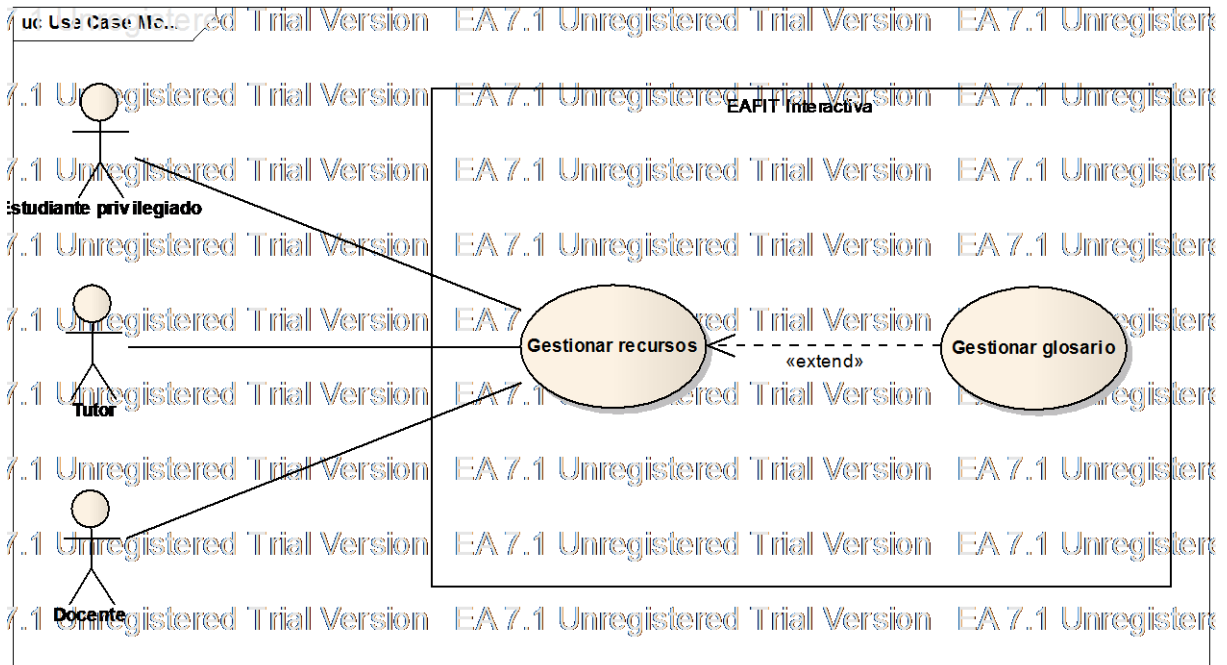


Figura 22. Diagrama del caso de uso gestionar glosario

🕒 Precondiciones

El usuario debe haber ingresado al sistema.

🕒 Flujo básico

Este caso de uso extiende “Gestionar recursos” en {Gestionar glosario} cuando el sistema detecta que el usuario desea crear, eliminar o modificar una bibliografía.

{Pedir información}

1. El sistema solicita la información necesaria de acuerdo a la acción que desea realizar.

{Crear glosario}

{Modificar glosario}

{Eliminar glosario}

{Fecha inválida}

2. Continuar en {Recibir los datos}

🕒 Flujos alternativos

Crear glosario

En {Crear glosario} el usuario selecciona la opción de crear un nuevo término.

1. El usuario selecciona la opción de crear un nuevo término.
2. El sistema solicita la siguiente información del término: término, definición, activo, fechas de inicio y finalización.
3. El usuario ingresa la información del término.
4. El sistema verifica que los datos hayan sido ingresados con éxito
5. Continuar en {Crear glosario}.

Modificar glosario

En {Modificar glosario} el usuario selecciona la opción de modificar término ya existente.

1. El sistema muestra los términos previamente creados.
2. El usuario selecciona el término al cual desea realizarle algún cambio.
3. El usuario modifica la información del término.
4. El sistema verifica que los datos hayan sido ingresados con éxito
5. Continuar en {Modificar glosario}.

Eliminar glosario

En {Eliminar glosario} el usuario selecciona la opción de eliminar un término ya existente.

1. El sistema muestra los términos previamente creados.
2. El usuario selecciona el término que desea eliminar.
3. Continuar en {Eliminar glosario}

🕒 Postcondiciones

El usuario permanece en el sistema.

7.3.1.14. Especificación del caso de uso: Gestionar FAQ

🕒 Descripción

Este caso de uso describe como los usuarios que tienen permisos crean, modifican y eliminan una pregunta frecuente de un curso dentro de EAFIT Interactiva.

🕒 Diagrama del caso de uso

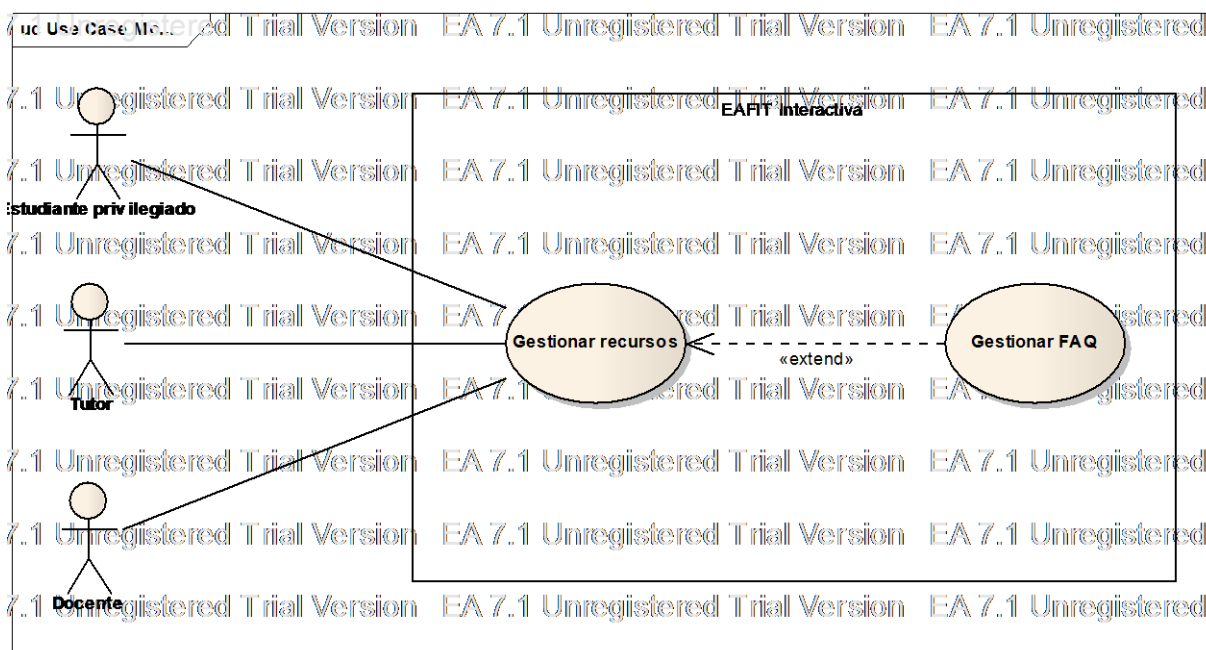


Figura 23. Diagrama del caso de uso gestionar FAQ

🕒 Precondiciones

El usuario debe haber ingresado al sistema.

🕒 Flujo básico

Este caso de uso extiende “Gestionar recursos” en {Gestionar FAQ} cuando el sistema detecta que el usuario desea crear, eliminar o modificar una pregunta frecuente.

{Pedir información}

1. El sistema solicita la información necesaria de acuerdo a la acción que desea realizar.

{Crear FAQ}

{Modificar FAQ}

{Eliminar FAQ}

{Fecha inválida}

2. Continuar en {Recibir los datos}

Flujos alternativos

Crear FAQ

En {Crear FAQ} el usuario selecciona la opción de crear una nueva pregunta frecuente.

1. El usuario selecciona la opción de crear una nueva pregunta frecuente.
2. El sistema solicita la siguiente información de la pregunta frecuente: pregunta, respuesta, activo, fechas de inicio y finalización.
3. El usuario ingresa la información de la pregunta frecuente.
4. El sistema verifica que los datos hayan sido ingresados con éxito
5. Continuar en {Crear FAQ}.

Modificar FAQ

En {Modificar FAQ} el usuario selecciona la opción de modificar una pregunta ya existente.

1. El sistema muestra las preguntas previamente creadas.

2. El usuario selecciona la pregunta a la cual desea realizarle algún cambio.
3. El usuario modifica la información de la pregunta.
4. El sistema verifica que los datos hayan sido ingresados con éxito
5. Continuar en {Modificar FAQ}.

Eliminar FAQ

En {Eliminar FAQ} el usuario selecciona la opción de eliminar una pregunta ya existente.

1. El sistema muestra las preguntas previamente creadas.
2. El usuario selecciona la pregunta que desea eliminar.
3. Continuar en {Eliminar FAQ}

Postcondiciones

El usuario permanece en el sistema.

7.3.1.15. Especificación del caso de uso: Gestionar Enlaces

Descripción

Este caso de uso describe como los usuarios que tienen permisos crean, modifican y eliminan los enlaces de un curso dentro de EAFIT Interactiva.

Diagrama del caso de uso

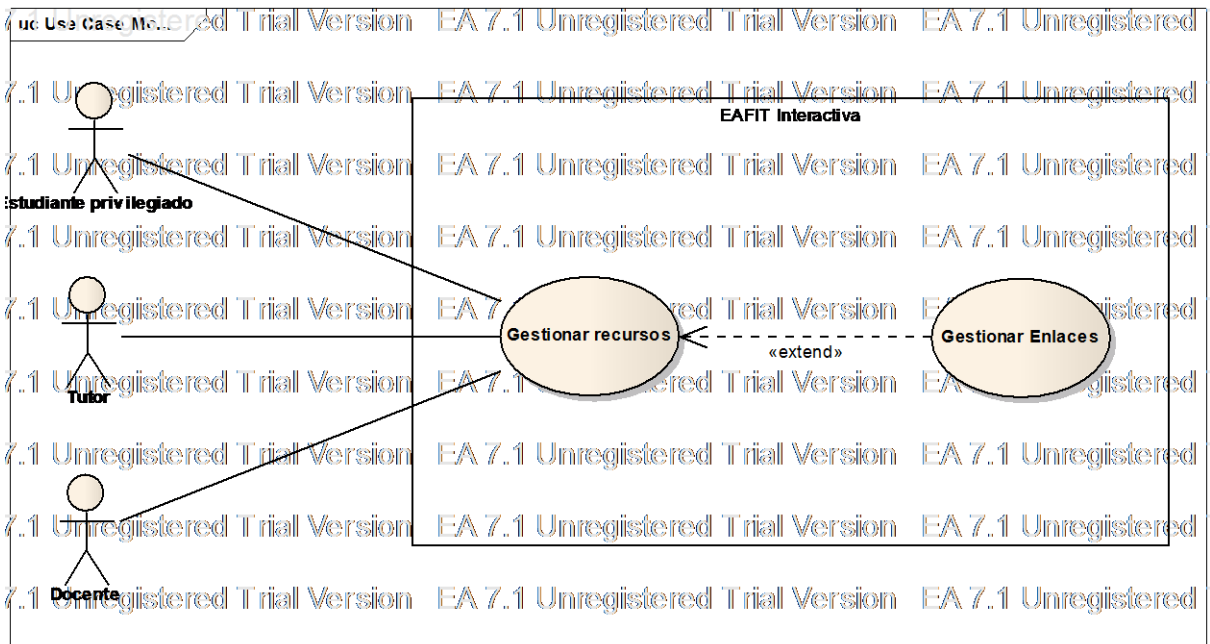


Figura 24. Diagrama del caso de uso gestionar Enlaces

🕒 Precondiciones

El usuario debe haber ingresado al sistema.

🕒 Flujo básico

Este caso de uso extiende “Gestionar recursos” en {Gestionar enlaces} cuando el sistema detecta que el usuario desea crear, eliminar o modificar un enlace.

{Pedir información}

1. El sistema solicita la información necesaria de acuerdo a la acción que desea realizar.

{Crear enlace}

{Modificar enlace}

{Eliminar enlace}

{Fecha inválida}

2. Continuar en {Recibir los datos}

🕒 Flujos alternativos

Crear enlace

En {Crear enlace} el usuario selecciona la opción de crear un nuevo enlace.

1. El usuario selecciona la opción de crear un nuevo enlace.
2. El sistema solicita la siguiente información del enlace: URL, comentarios, activo, fechas de inicio y finalización.
3. El usuario ingresa la información del enlace.
4. El sistema verifica que los datos hayan sido ingresados con éxito
5. Continuar en {Crear enlace}.

Modificar enlace

En {Modificar enlace} el usuario selecciona la opción de modificar un enlace ya existente.

1. El sistema muestra los enlaces previamente creados.
2. El usuario selecciona el enlace al cual desea realizarle algún cambio.
3. El usuario modifica la información del enlace.
4. El sistema verifica que los datos hayan sido ingresados con éxito
5. Continuar en {Modificar enlace}.

Eliminar enlace

En {Eliminar enlace} el usuario selecciona la opción de eliminar un enlace ya existente.

1. El sistema muestra los enlaces previamente creados.
2. El usuario selecciona el enlace que desea eliminar.
3. Continuar en {Eliminar enlace}

🕒 Postcondiciones

El usuario permanece en el sistema.

7.3.1.16. Especificación del caso de uso: Gestionar contenido

🕒 Descripción

Este Caso de Uso describe como los usuarios que tienen permisos cargan, modifican y eliminan diferentes contenidos de EAFIT interactiva.

🕒 Diagrama del caso de uso

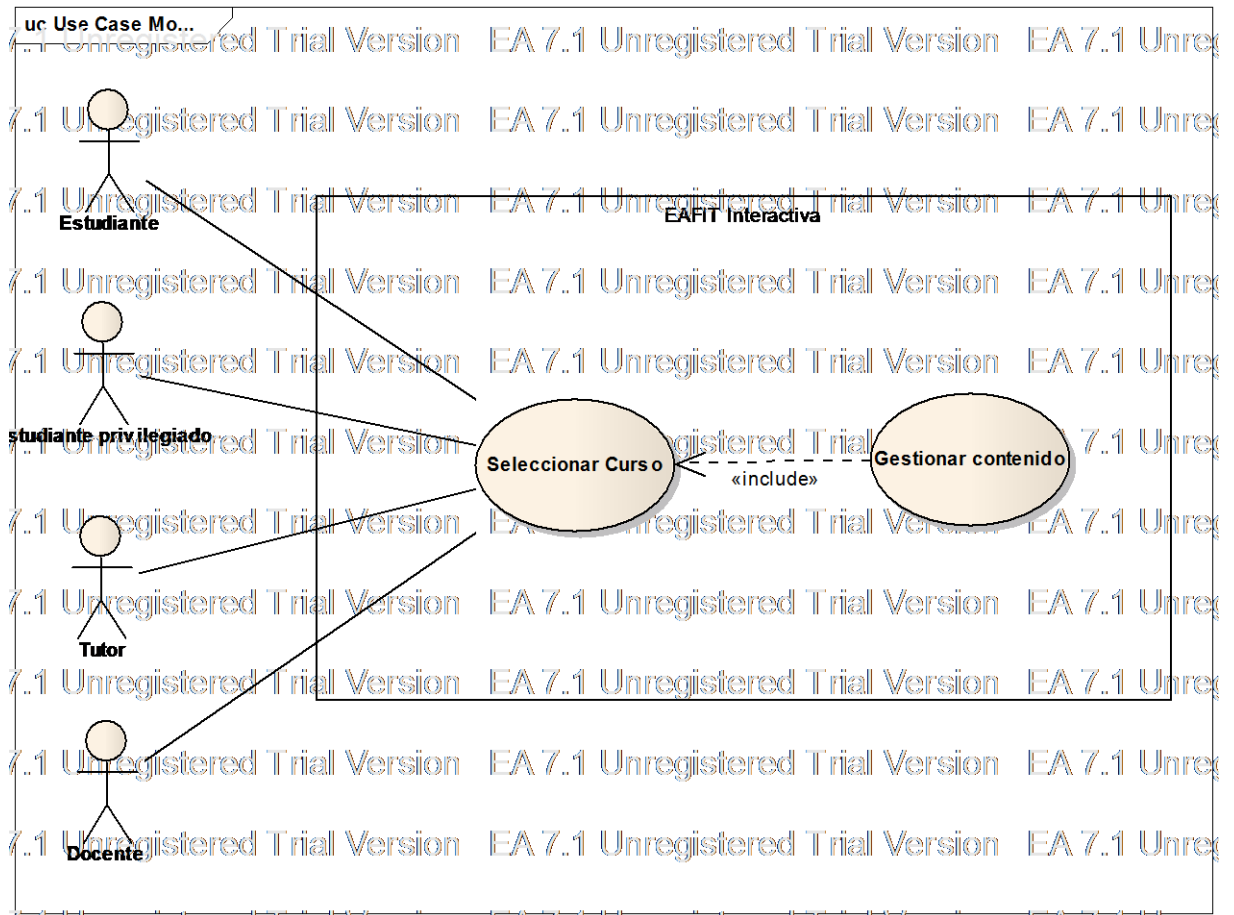


Figura 25. Diagrama del caso de uso gestionar contenido

🕒 Precondiciones

El usuario debe haber ingresado al sistema.

🕒 Flujo básico

{Nuevo contenido}

1. El caso de uso comienza cuando el sistema detecta que el usuario desea agregar un nuevo contenido al curso

{Agregar carpeta}

{Agregar elemento}

{Agregar enlace}

{Agregar podcast}

{Gestionar SCORM}

{Confirmar proceso}

2. El usuario confirma la operación.

{Archivo invalido}

3. El sistema muestra el contenido creado y almacena los datos.
4. El caso de uso termina.

🕒 Flujos alternativos

Agregar carpeta

En {Agregar carpeta} el usuario crea una nueva carpeta para el curso que podrá contener o no elementos.

1. El usuario elige crear una nueva carpeta.
2. El sistema pide el título, la descripción y la fecha de inicio y fin de la carpeta.
3. El usuario ingresa los datos.
4. Continuar en {Confirmar proceso}

Agregar elemento

En {Agregar elemento} el usuario sube un nuevo archivo al curso.

1. El usuario elige crear un nuevo elemento.
2. El sistema pide el título, la descripción y la fecha de inicio y fin del elemento.
3. el sistema pide la ruta del archivo que debe adjuntarse.
4. El usuario ingresa los datos.
5. Continuar en {Confirmar proceso}

Agregar enlace

En {Agregar enlace} el usuario adiciona un nuevo enlace web al curso.

1. El usuario elige crear un nuevo enlace.
2. El sistema pide el título, la descripción y la URL.
3. El usuario ingresa los datos.
4. Continuar en {Confirmar proceso}

Agregar podcast

En {Agregar podcast} el usuario adiciona un nuevo podcast al curso.

1. El usuario elige crear un nuevo podcast.
2. El sistema pide el título, la descripción, los permisos para los demás usuarios y la fecha de inicio y fin.
3. El docente ingresa los datos.
4. Continuar en {Confirmar proceso}

Archivo invalido

En {Archivo invalido} el sistema detiene el proceso porque los archivos no pueden adjuntarse.

1. El sistema muestra un mensaje informando que los archivos no pueden adjuntarse.
2. Continuar en {Nuevo contenido}

🕒 Postcondiciones

El usuario permanece en el sistema.

7.3.1.17. Especificación del caso de uso: Desplegar contenido SCORM

🕒 Descripción

Este caso de uso es incluido por otros casos de uso cuando se despliega a los usuarios docente, tutor, estudiante privilegiado y estudiante los diferentes contenidos que hacen parte de un curso SCORM determinado.

🕒 Diagrama del caso de uso



Figura 26. Diagrama del caso de uso desplegar contenido SCORM

🕒 Precondiciones

El usuario debe haber ingresado al sistema.

🕒 Flujo básico

1. El caso de uso comienza cuando el usuario selecciona la opción de ver contenido de un curso desarrollado en el estándar SCORM.
2. El sistema muestra la tabla de contenidos del curso.
3. El usuario selecciona la actividad que desea realizar.
4. El sistema muestra la actividad seleccionada.
{Terminar curso}
{Suspende curso}
5. El caso de uso termina.

🕒 Flujos alternativos

Terminar curso

En {Terminar curso} el usuario selecciona la opción de terminar un curso SCORM.

1. El usuario selecciona la opción de terminar curso.
2. El sistema redirecciona al usuario a la página de desplegar contenido SCORM y no almacena las actividades del usuario.
3. El caso de uso aborta.

Suspende curso

En {Suspende curso} el usuario selecciona la opción de suspender un curso SCORM.

1. El usuario selecciona la opción de suspender curso.
2. El sistema redirecciona al usuario a la página de desplegar contenido SCORM y almacena las actividades del usuario.
3. Continuar en {Suspende curso}

🕒 Postcondiciones

El usuario permanece en el sistema.

7.3.1.18. Especificación del caso de uso: Gestionar contenido SCORM

🕒 Descripción

Este caso de uso describe como el docente crea (carga en EAFIT interactiva) y elimina los diferentes cursos virtuales y la fecha máxima en la que puede finalizar un curso que hace parte del SIMEVI.

🕒 Diagrama del caso de uso

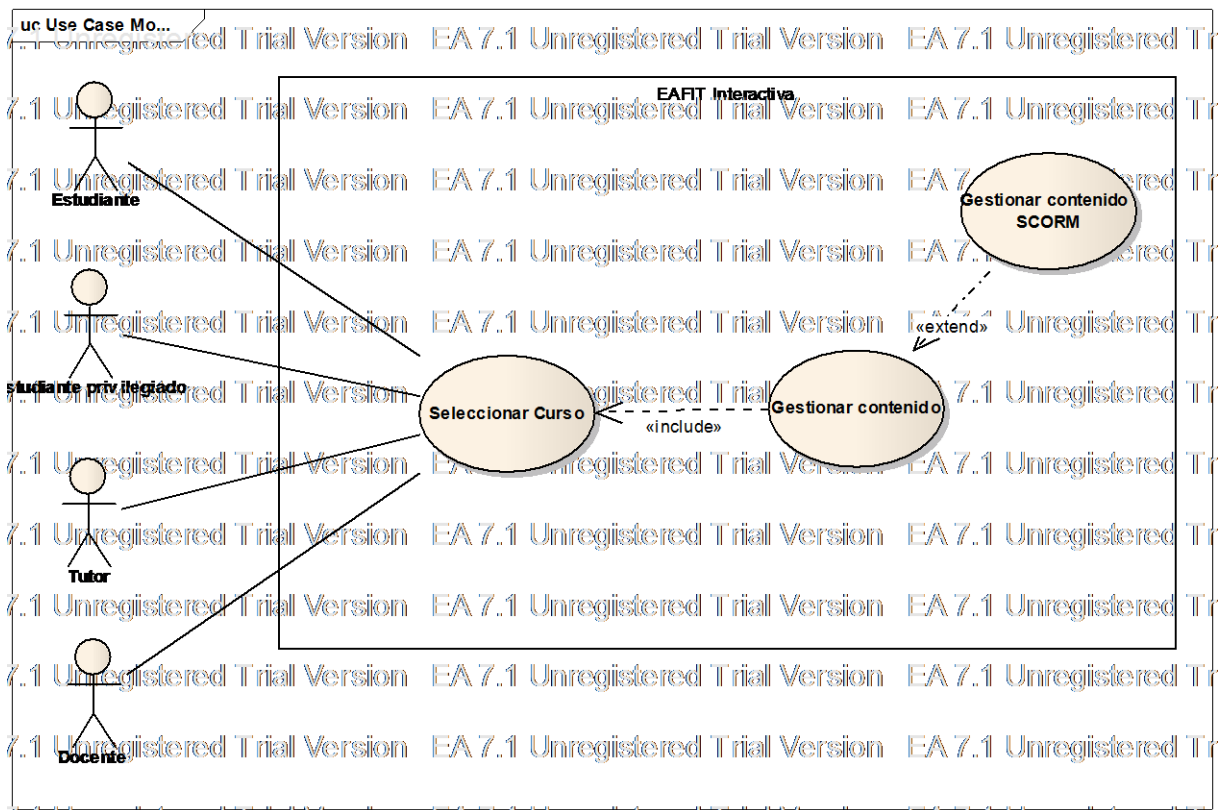


Figura 27. Diagrama del caso de uso gestionar contenido SCORM

🕒 Información Adicional

Este caso de uso lo soluciona la plataforma de EAFIT Interactiva, sin embargo se hace necesario detallar el flujo básico cuando el contenido es un curso que utiliza el estándar SCORM.

Precondiciones

El usuario debe seleccionar un curso.

Flujo básico

Este caso de uso extiende “Gestionar contenido” en {Gestionar SCORM} cuando el sistema detecta que el usuario desea ingresar un nuevo curso SCORM.

1. El caso de uso comienza cuando el usuario selecciona la opción de agregar un nuevo curso de SCORM.
2. El sistema solicita la ruta del contenido previamente creado.
3. El usuario ingresa la ruta del contenido.
{Extensión inválida}
4. El sistema descomprime el archivo.
5. El sistema valida el contenido al estándar SCORM.
{Manifest no encontrado}
{Manifest mal formado}
{Namespace inválido}
{Documentos de control no encontrados}
{Esquema inválido}
{Requisitos SCORM}
6. El sistema guarda el contenido y muestra un mensaje de que el contenido ha sido cargado con éxito.
7. El caso de uso termina.

Flujos alternativos

Extensión inválida

En {Extensión inválida} el sistema comprueba que el archivo no tiene una extensión .zip.

1. El sistema muestra un mensaje donde se notifica al usuario que el archivo no es válido porque no tiene la extensión correcta.
2. El usuario selecciona opción de regresar.
3. El sistema redirecciona al usuario a la página de ingreso de contenido.
4. El caso de uso aborta.

Manifest no encontrado

En {Manifest no encontrado} el sistema verifica la ubicación del archivo "imsmanifest.xml" y no lo encuentra en la raíz del paquete.

1. El sistema muestra un mensaje notificando al usuario que el archivo no se encuentra en la raíz del paquete.
2. El usuario selecciona opción de regresar.
3. El sistema redirecciona al usuario a la página de ingreso de contenido.
4. El caso de uso aborta.

Manifest mal formado

En {Manifest mal formado} el sistema valida las etiquetas del archivo y encuentra que alguno de los campos no se encuentra especificado de acuerdo al estándar.

1. El sistema muestra un mensaje notificando al usuario que el archivo no se encuentra bien formado.
2. El usuario selecciona opción de regresar.
3. El sistema redirecciona al usuario a la página de ingreso de contenido.
4. El caso de uso aborta.

Namespace inválido

En {Namespace inválido} el sistema valida las etiquetas del archivo y encuentra que el campo de namespace no se encuentra especificado de acuerdo al estándar.

1. El sistema muestra un mensaje notificando al usuario que el archivo no pertenece al namespace esperado.
2. El usuario selecciona opción de regresar.
3. El sistema redirecciona al usuario a la página de ingreso de contenido.
4. El caso de uso aborta.

Documentos de control no encontrados

En {Documentos de control no encontrados} el sistema valida el archivo y encuentra que los documentos de control no se encuentran en la raíz del paquete.

1. El sistema muestra un mensaje notificando al usuario que los documentos de control no se encuentran localizados en el paquete.
2. El usuario selecciona opción de regresar.
3. El sistema redirecciona al usuario a la página de ingreso de contenido.
4. El caso de uso aborta.

Esquema inválido

En {Esquema inválido} el sistema valida el archivo y encuentra que no es válido contra los esquemas.

1. El sistema muestra un mensaje notificando al usuario que el archivo "imsmanifest.xml" no es válido contra los esquemas.
2. El usuario selecciona opción de regresar.
3. El sistema redirecciona al usuario a la página de ingreso de contenido.
4. El caso de uso aborta.

Requisitos SCORM

En {Esquema inválido} el sistema valida el archivo y encuentra que no es válido contra los requisitos definidos en el estándar SCORM.

1. El sistema muestra un mensaje notificando al usuario que el archivo "imsmanifest.xml" no es válido con los requisitos definidos en SCORM 2004 3rd Edition.
2. El usuario selecciona opción de regresar.
3. El sistema redirecciona al usuario a la página de ingreso de contenido.
4. El caso de uso aborta.

Postcondiciones

El contenido queda cargado en la plataforma de EAFIT Interactiva.

7.3.1.19. Especificación del caso de uso: Visualizar SCO

Descripción

Este caso de uso es incluido por otros casos de uso cuando los usuarios, docente y estudiante, desean visualizar los “assets” de un curso creado en el estándar SCORM.

Diagrama del caso de uso

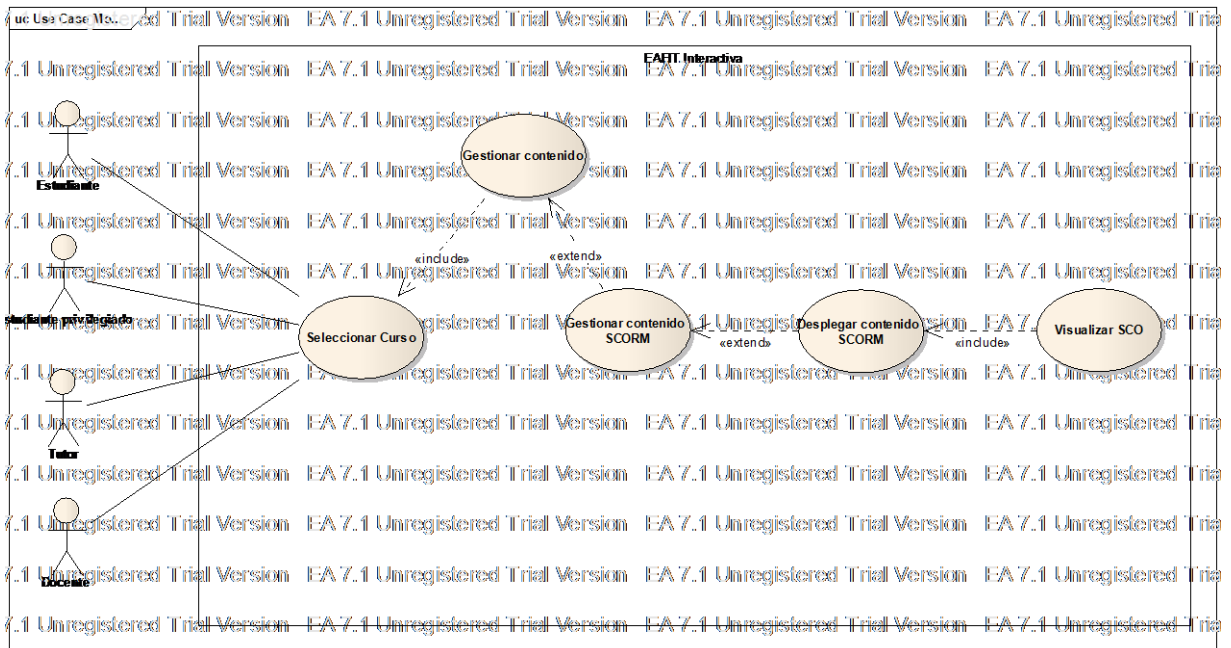


Figura 28. Diagrama del caso de uso visualizar SCO

🕒 Precondiciones

El usuario debe seleccionar un curso que haya sido implementado utilizando el estándar SCORM.

🕒 Flujo básico

1. El caso de uso comienza cuando el usuario ingresa a un contenido.
2. El sistema muestra el SCO al estudiante.
3. El usuario acepta el contenido.
4. El caso de uso termina.

🕒 Postcondiciones

El usuario permanece en el sistema para continuar otras actividades del curso.

7.3.1.20. Especificación del caso de uso: Hacer seguimiento al usuario

🕒 Descripción

Este caso de uso es incluido por otros casos de uso cuando se desea conocer cuales actividades un usuario está realizando y cuales ha terminado en un curso desarrollado en el estándar SCORM.

🕒 Diagrama del caso de uso



Figura 29. Diagrama del caso de uso hacer seguimiento al usuario

🕒 Precondiciones

El usuario debe haber iniciado un curso desarrollado en el estándar SCORM.

🕒 Flujo básico

1. El caso de uso comienza cuando el usuario da inicio a un curso desarrollado en el estándar SCORM.
2. Si es la primera vez que el usuario da inicio al curso el sistema toma la fecha y hora de ingreso al curso.
{Continuar actividades}

3. El sistema almacena la información de cada una de las actividades que realiza el usuario.
{Tiempo finalizado}
4. El usuario finaliza las actividades que hacen parte del curso.
{Enviar evaluación}
{Notificar docente}
5. El sistema calcula el tiempo que tomó al usuario terminar el curso, lo almacena y notifica al usuario de que ha terminado el curso.
6. El usuario selecciona la opción de regresar al inicio del curso.
7. El sistema envía al usuario al inicio del curso.
8. El caso de uso termina.

Flujos alternativos

Tiempo finalizado

En {Tiempo finalizado} el sistema comprueba que el tiempo máximo dado para finalizar el curso se ha terminado.

1. El sistema notifica al usuario de que el tiempo dado para finalizar el curso se ha terminado, cuál ha sido su nota y que no puede continuar.
2. El usuario selecciona la opción de regresar al inicio.
3. El sistema envía al usuario al inicio.
4. El caso de uso aborta.

Continuar actividades

En {Continuar actividades} el sistema comprueba que el usuario ha ingresado anteriormente al curso y debe continuar con las actividades que dejó sin finalizar desde el último ingreso.

1. El sistema busca la actividad que el usuario tiene sin concluir.

2. El sistema envía al usuario a la actividad que no ha finalizado y notifica al usuario.
3. El usuario selecciona la actividad que desea continuar.
4. Continuar en {Continuar actividades}.

⌚ Postcondiciones

El usuario permanece en el sistema para continuar otras actividades del curso.

7.3.1.21. Especificación del caso de uso: Realizar evaluación

⌚ Descripción

Este caso de uso describe como el usuario recibe una retroalimentación luego de realizar las diferentes actividades del sistema.

⌚ Diagrama del caso de uso

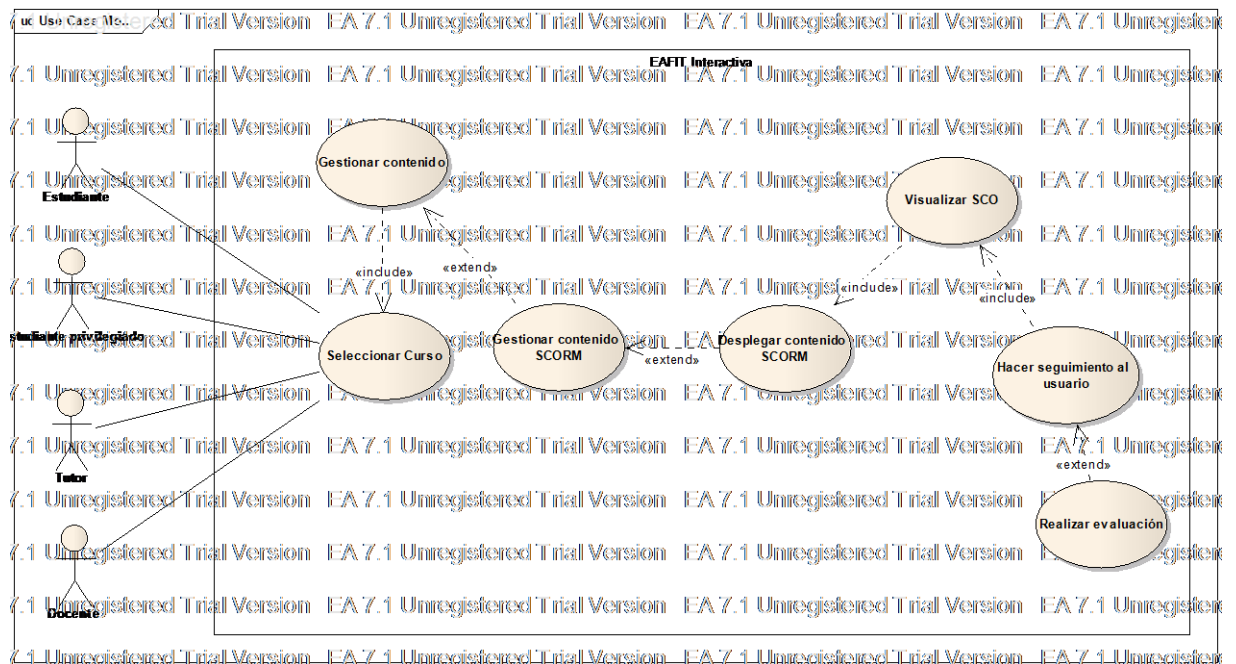


Figura 30. Diagrama del caso de uso realizar evaluación

Precondiciones

El usuario debe haber terminado un curso desarrollado en el estándar SCORM.

Flujo básico

Este caso de uso extiende “Hacer seguimiento al usuario” en {Enviar evaluación} cuando el sistema detecta que el usuario ha terminado un módulo que contiene actividades evaluativas.

1. El caso de uso comienza cuando el usuario finaliza alguna de las actividades evaluativas de los diferentes módulos.
2. El sistema analiza los datos generados por el estudiante luego de realizar la actividad.
3. El sistema busca el mensaje de retroalimentación asociado con la calificación de la actividad.
4. El sistema almacena la información y muestra mensaje.
5. El usuario confirma los datos.
6. El caso de uso termina.

Postcondiciones

El usuario permanece en el sistema para continuar otras actividades del curso. La actividad queda almacenada en el repositorio.

7.3.1.22. Especificación del caso de uso: Enviar notificación al docente

Descripción

Este caso de uso describe como el sistema envía por el correo de EAFIT Interactiva una actividad finalizada por un estudiante a su respectivo docente.

🕒 Diagrama del caso de uso

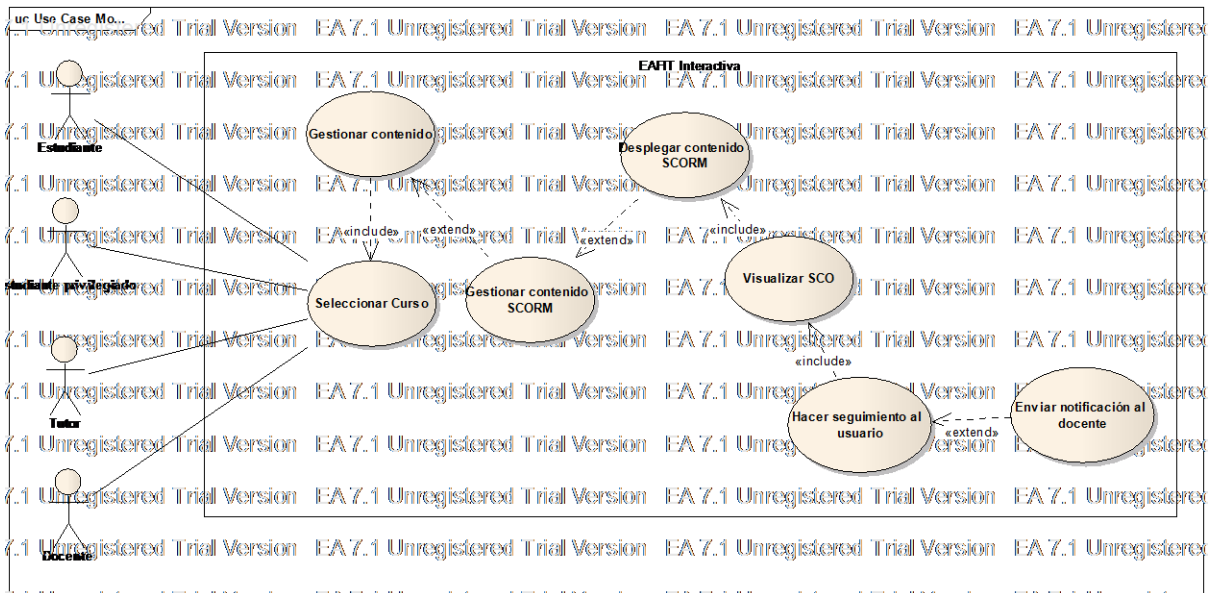


Figura 31. Diagrama del caso de uso enviar notificación al docente

🕒 Precondiciones

El usuario debe haber terminado una actividad.

🕒 Flujo básico

Este caso de uso extiende “Hacer seguimiento al usuario” en {Notificar docente} cuando el sistema detecta que el usuario ha terminado una actividad.

1. El caso de uso comienza cuando el estudiante finaliza una actividad.
2. El sistema guarda la información de la actividad en el repositorio de actividades realizadas.
3. El sistema notifica al estudiante que la actividad ha sido enviada al docente.
4. El sistema notifica por correo de EAFIT Interactiva al docente que hay actividades finalizadas.
5. El usuario confirma los datos.
6. El caso de uso termina.

🕒 Postcondiciones

El usuario permanece en el sistema para continuar otras actividades del curso.

7.3.1.23. Especificación del caso de uso: Error en procesamiento

🕒 Descripción

Este caso de uso describe como el sistema notifica al usuario cuando algún cambio en un recurso no puede realizarse.

🕒 Diagrama del caso de uso

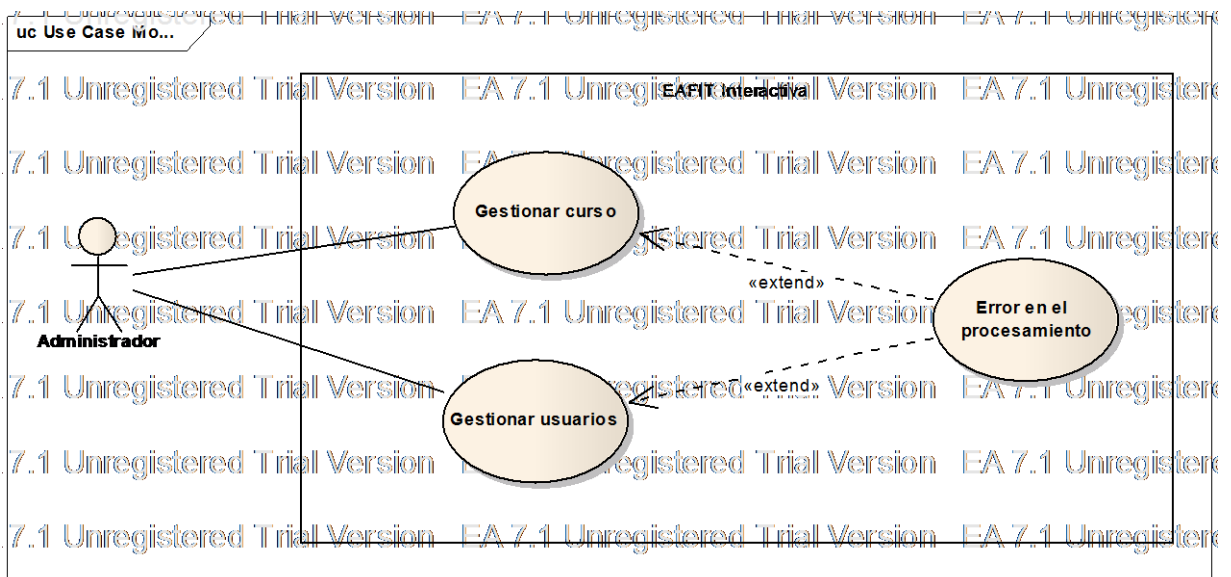


Figura 32. Diagrama del caso de uso error en procesamiento

🕒 Precondiciones

El usuario debe haber ingresado al sistema.

🕒 Flujo básico

Este caso de uso extiende “Gestionar curso” y “Gestionar usuarios” en {Error en Procesamiento} cuando el sistema detecta que ha ocurrido un error en el procesamiento de algún cambio de un recurso.

1. El sistema muestra un mensaje informando que ocurrió un error y la acción no se completó.
2. El usuario selecciona aceptar.
3. El sistema reenvía al usuario a la página principal del sistema.
4. El caso de uso termina.

Postcondiciones

El usuario permanece en el sistema.

7.3.1.24. Especificación del caso de uso: Cancelar recurso

Descripción

Este caso de uso describe como el sistema detecta que el usuario selecciona la opción de cancelar algún cambio que esté realizando en un recurso.

Diagrama del caso de uso

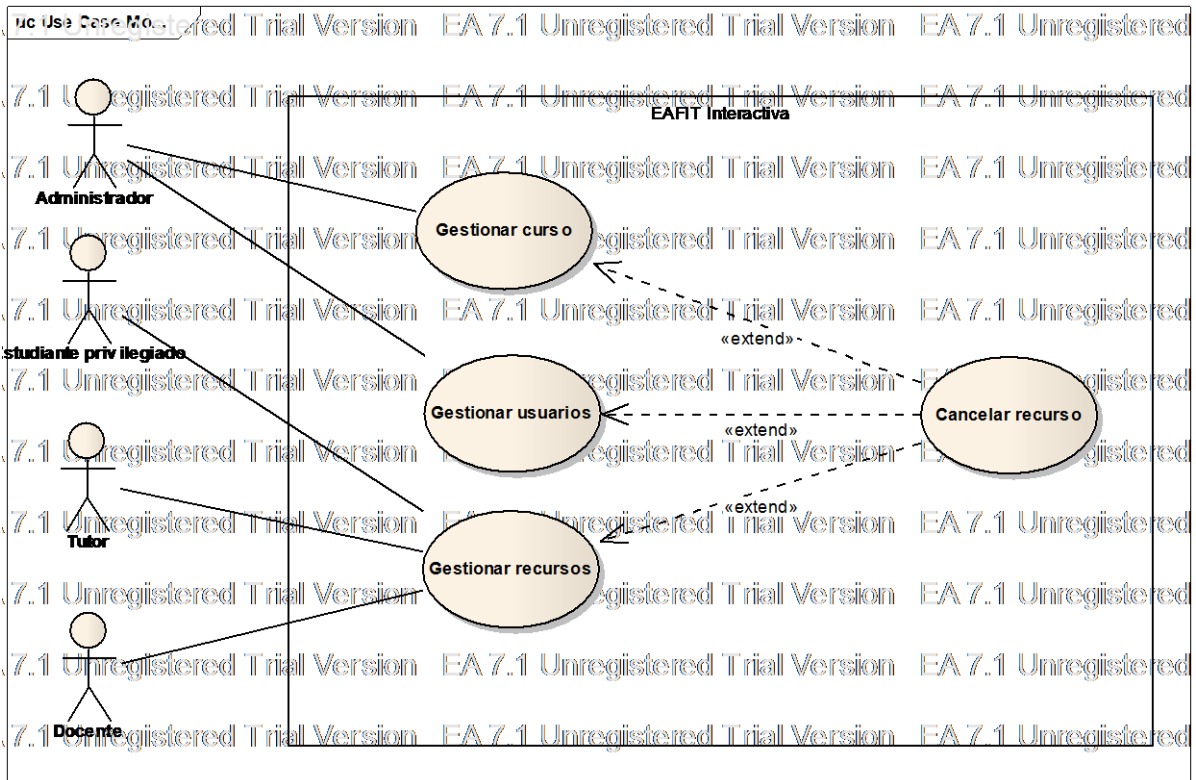


Figura 33. Diagrama del caso de uso cancelar recurso

🕒 Precondiciones

El usuario debe haber ingresado al sistema.

🕒 Flujo básico

Este caso de uso extiende “Gestionar curso”, “Gestionar usuarios” y “Gestionar recursos” en {Cancelar recurso} cuando el sistema detecta que el usuario desea cancelar algún cambio de un recurso.

1. El usuario selecciona la opción de cancelar.
2. El sistema redirecciona al usuario a la página de la actividad que está realizando.
3. El caso de uso termina.

🕒 Postcondiciones

El usuario permanece en el sistema.

7.3.1.25. Especificación del caso de uso: Ingresar archivo adjunto

🕒 Descripción

Este caso de uso describe como el sistema detecta que el usuario selecciona la opción de adjuntar un archivo a algún recurso del sistema que desea crear.

🕒 Diagrama del caso de uso

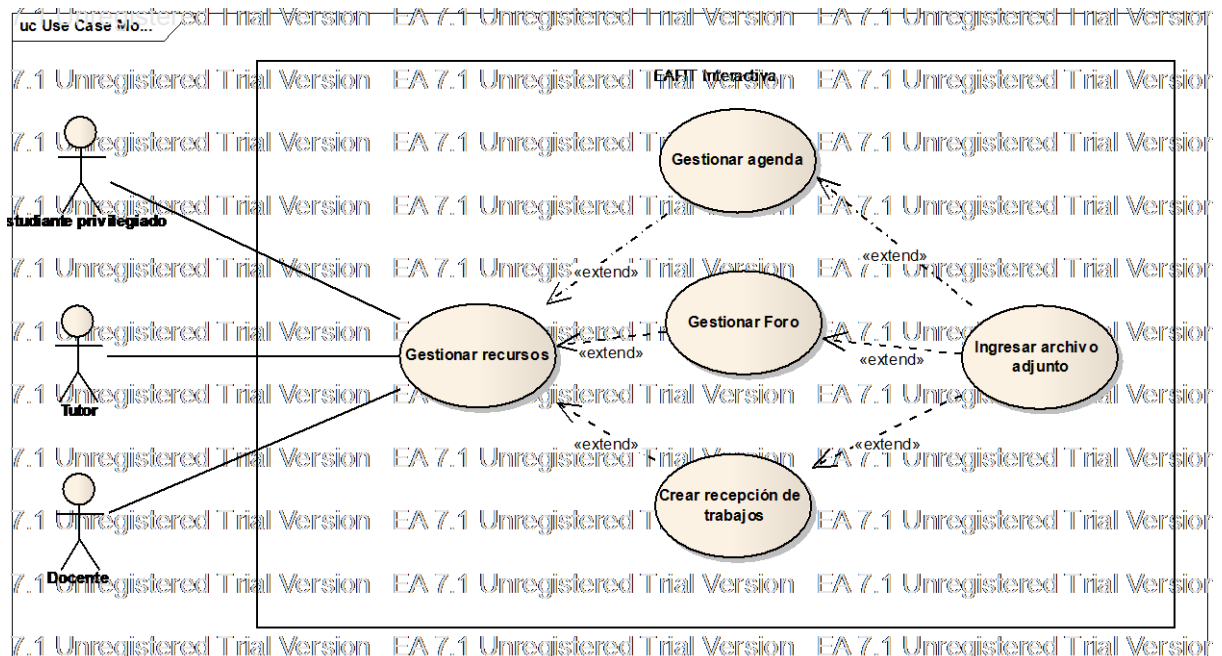


Figura 34. Diagrama del caso de uso ingresar archivo adjunto

🕒 Precondiciones

El usuario debe haber ingresado al sistema.

🕒 Flujo básico

Este caso de uso extiende “Gestionar foro”, “Crear recepción de trabajos” y “Gestionar agenda” en {Archivo adjunto} cuando el sistema detecta que el usuario crear un nuevo archivo adjunto.

1. El usuario selecciona la opción de adjuntar.
2. El sistema pregunta al usuario por la ruta del archivo que desea adjuntar.
3. El usuario ingresa la ruta del archivo.
4. El usuario confirma la operación.
5. Continuar en {Archivo adjunto}

Postcondiciones

El usuario permanece en el sistema.

7.3.1.26. Especificación del caso de uso: Notificar fecha inválida

Descripción

Este caso de uso describe como el sistema detecta que el usuario no ingresa la fecha de algún recurso de forma adecuada.

Diagrama del caso de uso



Figura 35. Diagrama del caso de uso notificar fecha inválida

⌚ Precondiciones

El usuario debe haber ingresado al sistema.

⌚ Flujo básico

Este caso de uso extiende “Gestionar foro”, “Crear recepción de trabajos”, “Iniciar conversación en chat”, “Gestionar agenda”, “Gestionar bibliografía”, “Gestionar glosario”, “Gestionar enlaces” y “Gestionar FAQ” en {Fecha inválida} cuando el sistema detecta que el usuario desea cancelar algún cambio de un recurso.

1. El sistema muestra mensaje indicando que la fecha ingresada no es válida y la razón por la cual no es válida.
2. El sistema no crea el recurso.

3. Continuar en {Pedir Información}.

Postcondiciones

El usuario permanece en el sistema.

7.3.1.27. Especificación del caso de uso: Notificar datos erróneos

Descripción

Este caso de uso describe como el sistema detecta que los datos ingresados en un campo numérico de algún recurso del sistema no son numéricos.

Diagrama del caso de uso

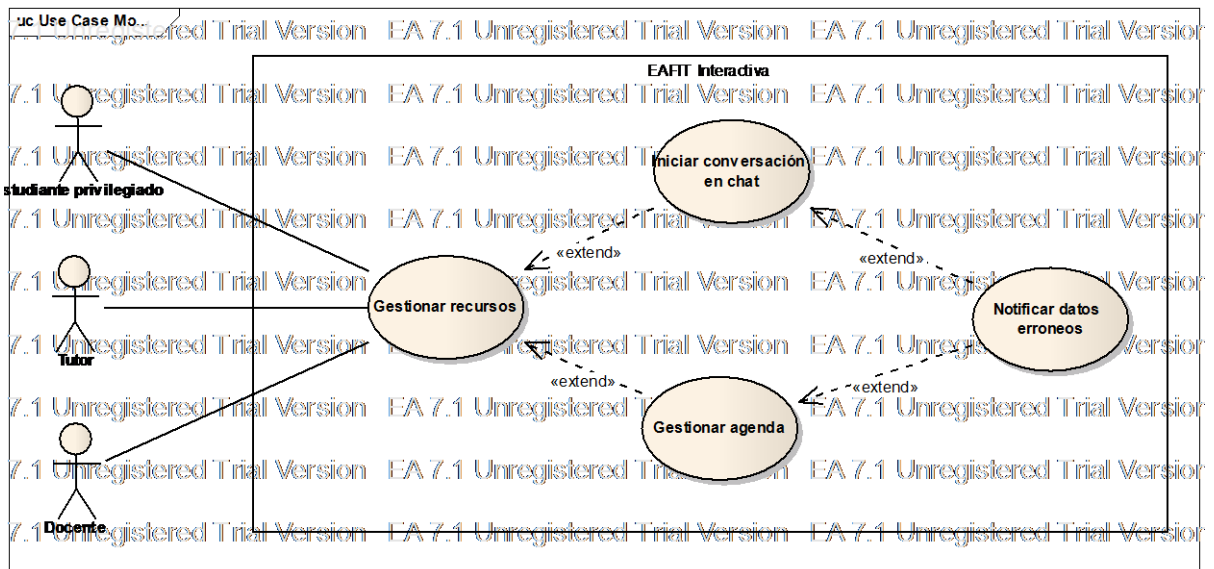


Figura 36. Diagrama del caso de uso notificar datos erróneos

Precondiciones

El usuario debe haber ingresado al sistema.

Flujo básico

Este caso de uso extiende “Iniciar conversación en chat” y “Gestionar agenda” en {Descripción requerida} cuando el sistema detecta que el usuario ha ingresado un valor que no es numérico en un campo numérico.

1. El sistema muestra mensaje indicando que el campo es requerido y no puede estar en blanco.
2. El sistema no crea el recurso.
3. Continuar en {Pedir Información}.

Postcondiciones

El usuario permanece en el sistema.

7.4. FASE DE CONSTRUCCIÓN

7.4.1. Diagrama de paquetes

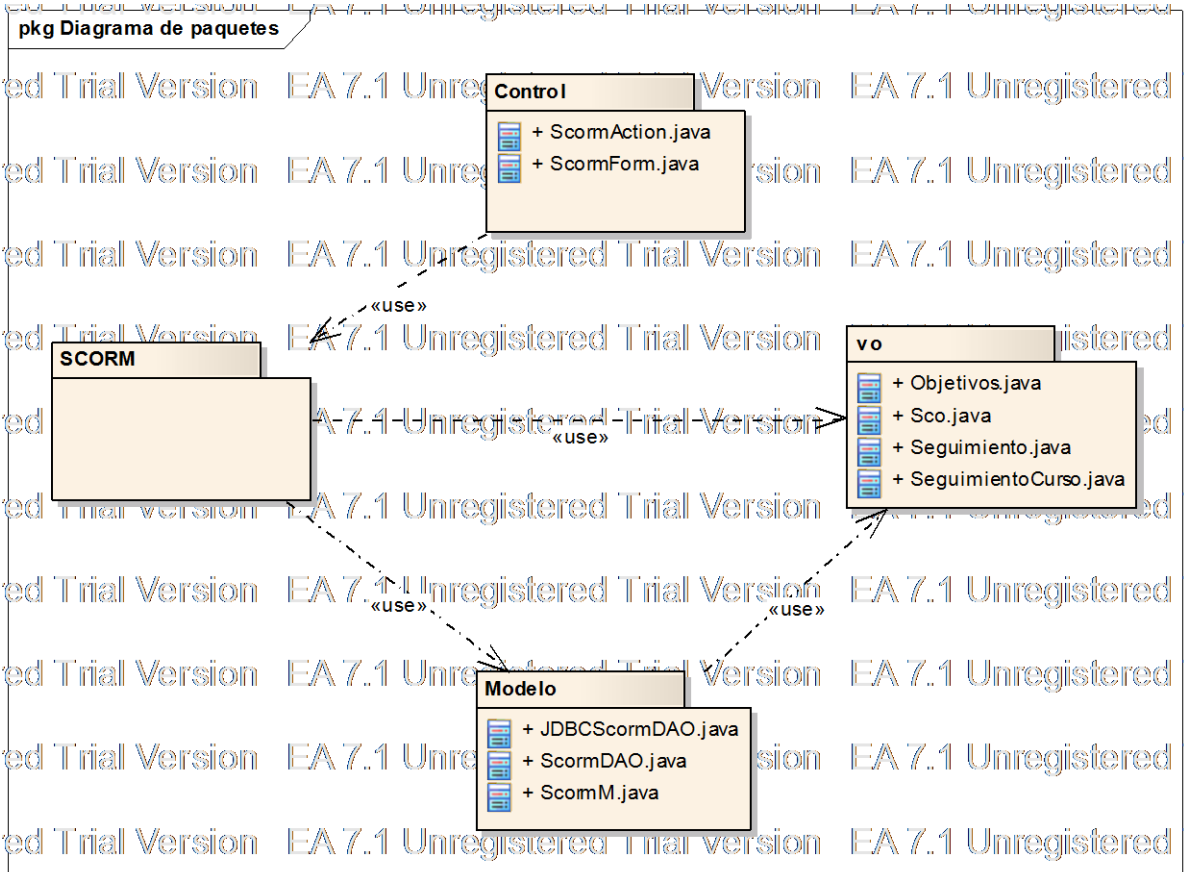


Figura 37. Diagrama de paquetes

7.4.2. Diagrama de clases

7.4.2.1. Clases del paquete control

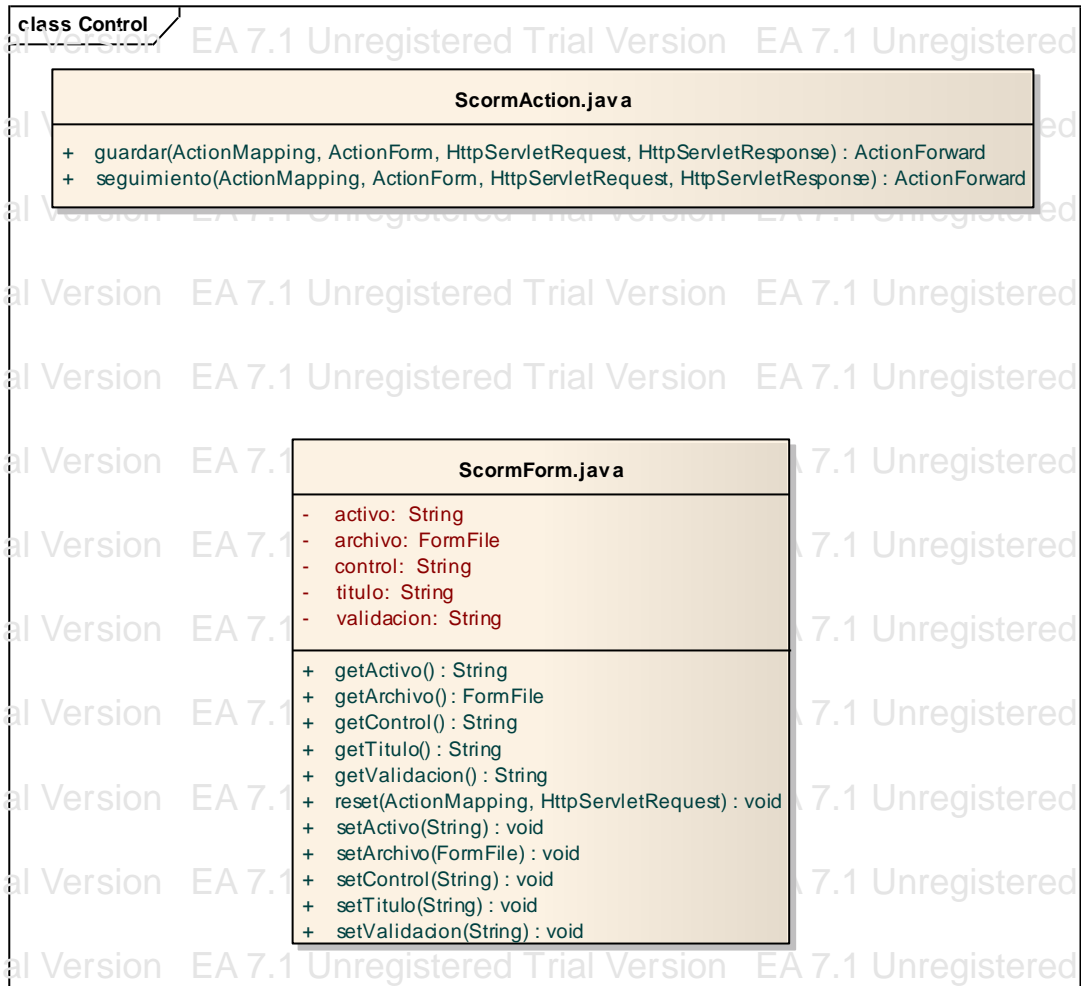


Figura 38. Clases del paquete control

Para la descripción de las clases nos apoyaremos en (23).

ScormAction.java

Tiene la función de controlador, es el encargado de manejar la lógica del negocio para operar las solicitudes relacionadas con los cursos desarrollados en el estándar SCORM.

Si la opción seleccionada es la de guardar un curso, envía al usuario a la página confirmacion.jsp que es la encargada de entregar el resultado de las validaciones realizadas al curso.

Si la opción seleccionada es la de visualizar el seguimiento de un estudiante, envía al usuario a la página visualizar.jsp que es la encargada de mostrar el estado en que un estudiante se encuentra para cada uno de los cursos que tiene activos.

ScormForm.java

Esta clase representa un formulario HTML en el cual los usuarios ingresan los datos pertenecientes a un curso desarrollado en SCORM. Implementa métodos get y set para acceder a cada una de las propiedades y posee un método reset para inicializar el formulario antes de ser usado. Además, posee un método de validación, donde verifica unas propiedades básicas sobre el archivo antes de iniciar las validaciones correspondientes al estándar SCORM.

7.4.2.2. Clases del paquete modelo

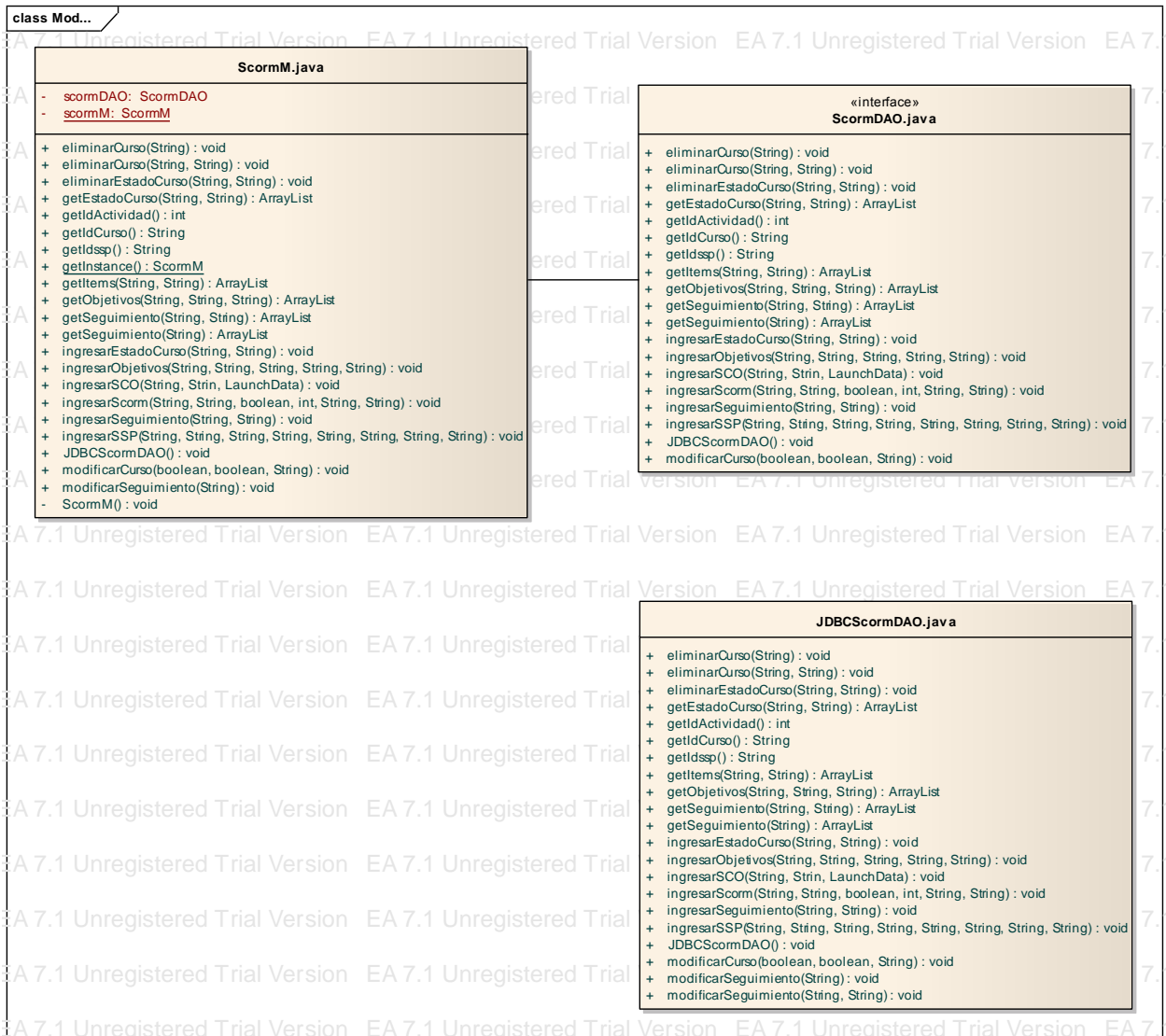


Figura 39. Clases del paquete modelo

Para la descripción de las clases nos apoyaremos en (23).

ScormM.java

Esta clase permite realizar un enlace entre todas las clases que necesiten realizar alguna operación, en la base de datos del sistema, relacionada con el estándar SCORM y la clase JDBCScormDAO.java

ScormDAO.java

Esta interfaz es utilizada para encapsular todo el acceso a la fuente de datos. Esta encargada de manejar la conexión para realizar el manejo de los datos y establece los métodos necesarios para realizar un correcto manejo del estándar SCORM en medios de persistencia.

JDBCScormDAO.java

Esta clase implementa los métodos establecidos por la interface ScormDAO.java, se encarga además de realizar las operaciones concernientes al estándar SCORM en una base de datos relacional mediante una conexión JDBC.

7.4.2.3. Clases del paquete vo



Figura 40. Clases del paquete vo

Para la descripción de las clases nos apoyaremos en (23).

Sco.java

Esta clase representa y almacena los valores de los SCO en el sistema por medio de estados que son accesibles a través de métodos set y get.

Objetivos.java

Esta clase representa y almacena los valores de los objetivos en el sistema por medio de estados que son accesibles a través de métodos set y get.

SeguimientoCurso.java

Esta clase representa y almacena los valores del seguimiento de un curso en el sistema por medio de estados que son accesibles a través de métodos set y get.

Seguimiento.java

Esta clase representa y almacena los valores del estado de un curso para un usuario en el sistema por medio de estados que son accesibles a través de métodos set y get.

7.4.2.4. Clases del paquete SCORM

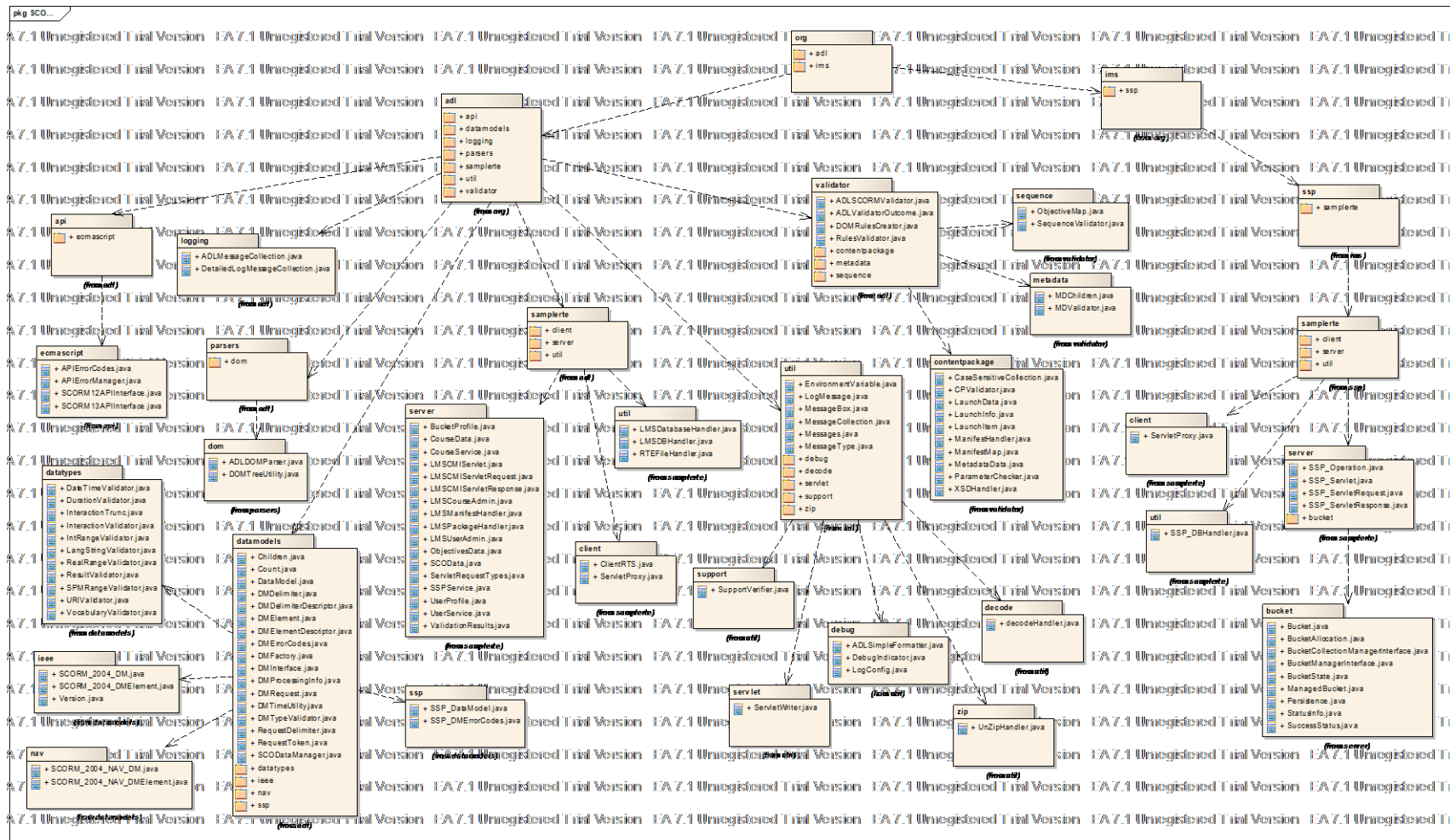


Figura 41. Clases del paquete SCORM

A continuación se presenta una descripción de las clases más importantes del estándar SCORM y con las cuales tuvimos un mayor acercamiento para adaptar este estándar a la plataforma EAFIT Interactiva⁽²⁴⁾.

CourseService.java

Esta clase maneja el acceso a la información del curso en la base de datos. Adicionalmente, maneja los inserts y updates de la nueva información.

LMSManifestHandler.java

Esta clase contiene los métodos utilizados por el ambiente de ejecución para analizar un archivo imsmanifest.xml y para procesar un paquete de contenido utilizando el ADL SCORM Validator. También contiene métodos utilizados por el ambiente de ejecución y el ADL Sequencer para obtener acceso a la información en el imsmanifest.xml.

ManifestHandler.java

Esta clase hace seguimiento, almacena y obtiene la información para lanzar los SCOs y la información de metadata, todo esto es encontrado o referenciado desde el paquete de contenido de prueba.

DOMTreeUtility.java

Esta clase provee la habilidad de acceder los subgrupos de un árbol DOM. Esta clase sirve como una clase de utilidad para la manipulación de árboles DOM.

CPValidator.java

Esta clase determina si el paquete de contenido de prueba está conforme con los perfiles de aplicación de los paquetes de contenido, como están definidos en el Modelo de Agregación de Contenidos de SCORM. Esta clase realiza las siguientes verificaciones:

- a) Determina si los archivos requeridos existen en la raíz del paquete.
- b) Analiza si el manifest está bien formado.
- c) Analiza el manifest para validarlo con los documentos de control.
- d) Determina si se utilizaron o no extensiones en el manifest.
- e) Determina si el manifest cumple con las reglas definidas en las reglas del perfil de aplicación de los paquetes de contenido.
- f) Valida metadata y scos externos si el sistema decide hacerlo.

La clase CPValidator hereda del ADLSCORMValidator para determinar si el manifest está bien formado y es válido contra los xsd(s). Esta clase valida el paquete de contenido de prueba frente a las reglas y los requerimientos necesarios para cumplir con cada perfil de aplicación de los paquetes de contenido.

UnZipHandler.java

Esta clase provee la habilidad de extraer los contenidos de un archivo comprimido (.zip) a un directorio dado.

ManifestMap.java

Esta clase es una estructura de datos utilizada para almacenar la información del manifest que es necesaria para la validación y procesamiento de los (sub)manifest.

ADLScormValidator.java

Esta clase sirve como la interfaz principal para obtener un validador de Metadata o de paquete de contenido. Este objeto contiene las funcionalidades comunes a los dos tipos de validadores (Metadata/CP) – sirviendo como padre para la herencia.

ADLValidatorOutcome.java

Un objeto ADLValidatorOutcome es retornado sobre la solicitud de un usuario a través de la utilización del método público disponible por el objeto ADLSCORMValidator. El objeto ADLValidatorOutcome sirve como el almacenamiento central del estado de las verificaciones durante las actividades de validación, incluyendo el DOM almacenado. Este objeto sirve como una forma eficiente de pasar el resultado de las actividades de validación a través de todo el sistema.

ADLSequencer.java

Esta clase encapsula de todos los cuatro procesos conceptuales requeridos por el secuenciamiento: interpretador de la navegación, secuenciamiento, enrollamiento y entrega.

El acercamiento tomado con esta implementación es el de proveer interfaces públicas para el RTE, habilitar la navegación/entrega, reporte de estado, administración simple de la sesión, y la información de la TOC.

Internamente, el ADLSequencer actúa en la un objeto SeqActivityTree, el cual provee una administración del estado del árbol de actividades y el acceso a las estructuras internas de sus actividades.

ADLDOMParser.java

El objeto ADLDOMParser hace interfaz con la clase open-source org.apache.xerces.dom.DomParser para encapsular y proveer actividades de análisis – incluyendo verificaciones de bien formado y validaciones contra los esquemas. Este objeto crea un DOM en memoria si se encuentra que está bien formado. Un objeto DOM también es creado en memoria después de la validación contra los esquemas.

DOMRulesCreator.java

Esta clase creará un DOM de las reglas XML que son necesarias para cada Perfil de Aplicación (p.ej. Validador del paquete de contenido – recursos y agregación de contenido / Validador Metadata – adlreg) / Validador de secuencia – secuencia.

RulesValidator.java

Provee soporte para los tokens de vocabulario válidos y si la entrada es válida basada en una lista, como se encuentra definido en el libro SCORM 2004 RTE.

ClientRTS.java

Esta clase implementa el SCORM versión 2004 SCO para el API de comunicación LMS definido por el IEEE.

Esta clase es implementada como un applet corriendo en un servidor LMS web. El applet corre en un contexto del cliente. Fue probado y desarrollado utilizando IE5 o IE6 y el Sun Java Runtime Environment Standard Edition Version 1.4.

Este applet interactúa con el componente del lado del servidor. El componente del servidor es esta implementado como un Java Servlet y maneja la persistencia del modelo de datos.

sequencingEngine.jsp

Este archivo determina cual ítem debe ser lanzado en el curso seleccionado. Responde a los siguientes eventos: Next – Lanza el siguiente sco o asset, Previous – Lanza el anterior sco o asset, Menu – Lanza el ítem seleccionad.

LMSMain.jsp

Esta es la página principal que contiene un frameset el cual contiene tres frames. El frame superior contiene los botones salir, anterior y siguiente. El frame inferior izquierdo contiene un menú construido de los ítems del curso seleccionado.

APIWrapper.js

Este archivo es utilizado por el entorno de ejecución para mantener la comunicación interna dentro de los frames del entorno.

7.4.3. Diagrama de comunicación

7.4.3.1. Diagrama de comunicación para el caso de uso “Gestionar contenido SCORM”

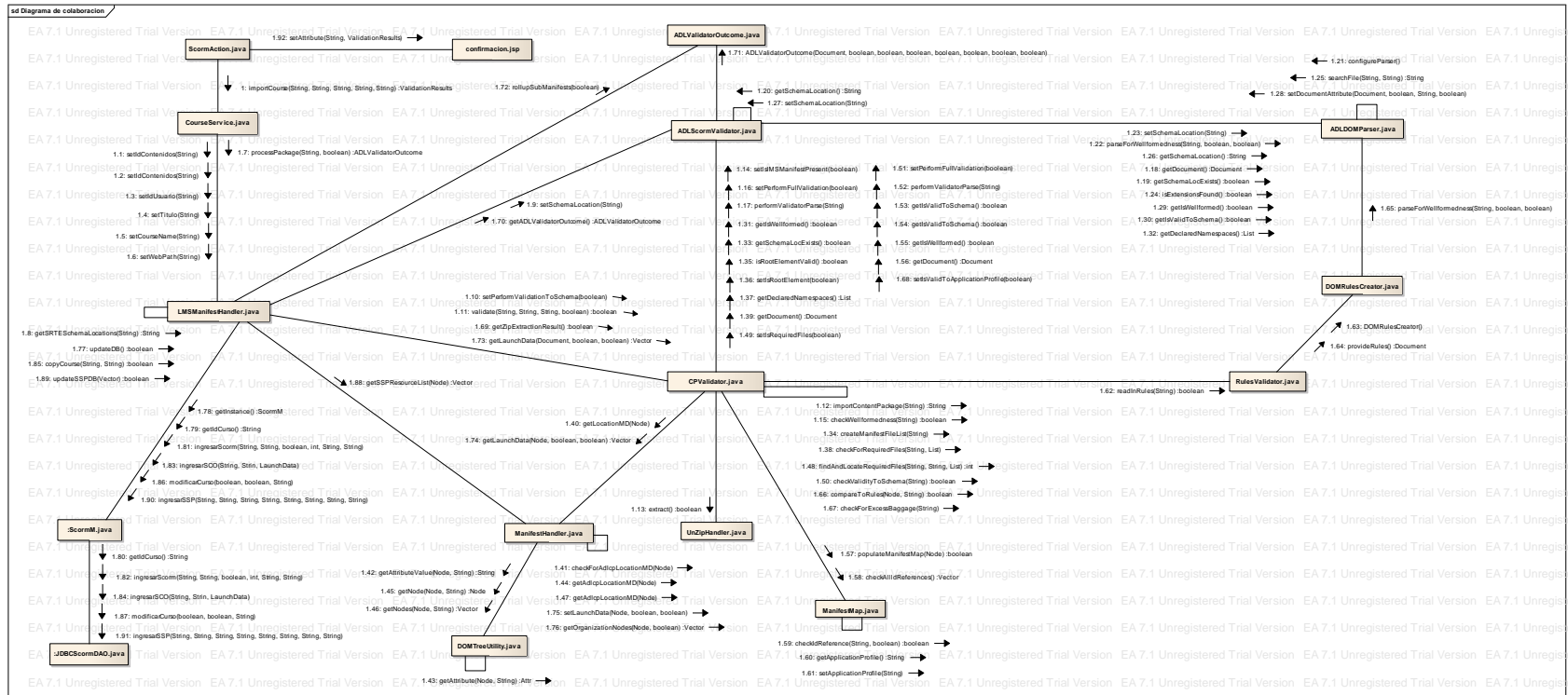


Figura 42. Diagrama de comunicación para el caso de uso “Gestionar contenido SCORM”

7.4.3.2. Diagrama de comunicación para el caso de uso “Visualizar SCO”

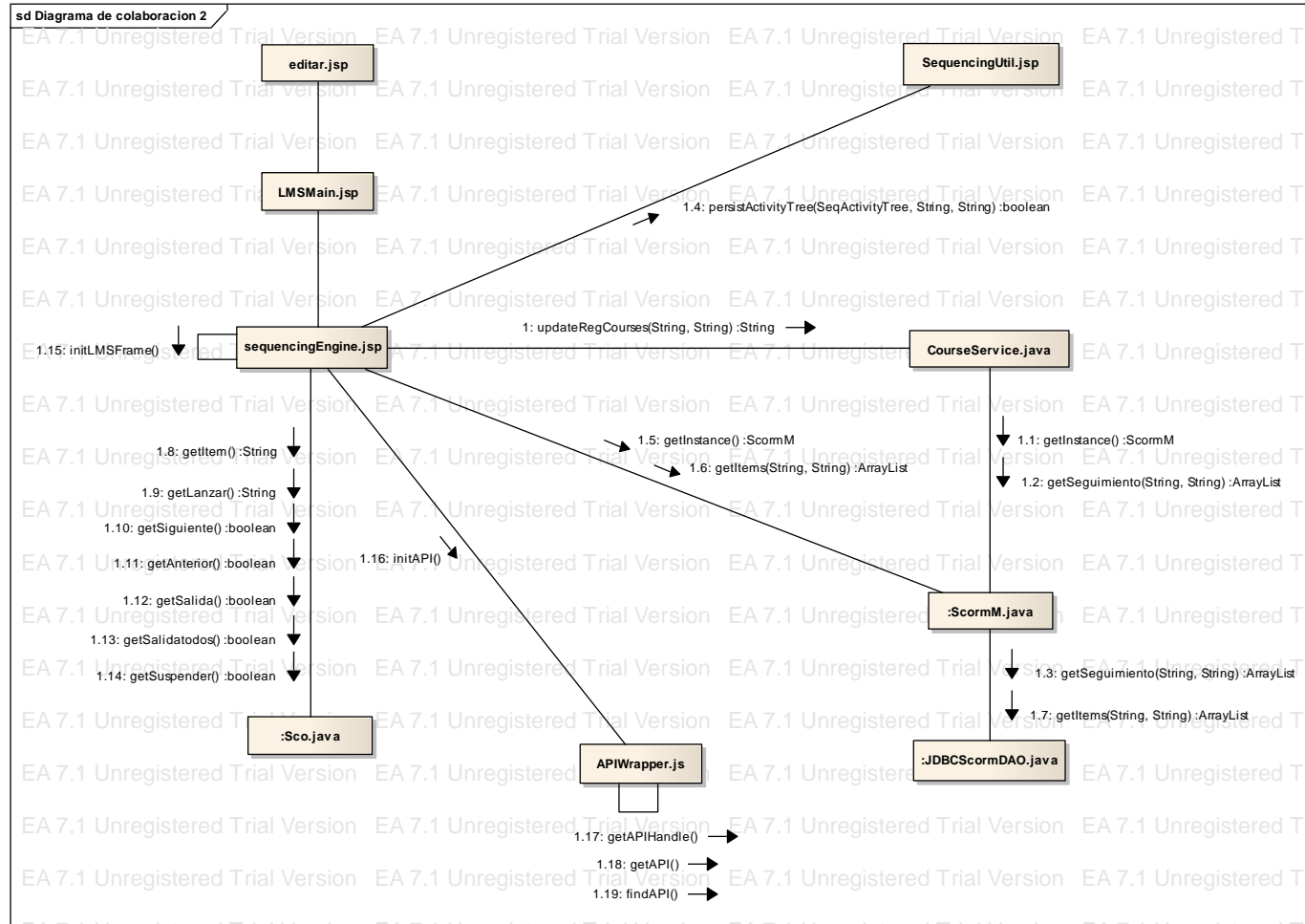


Figura 43. Diagrama de comunicación para el caso de uso “Visualizar SCO”

directamente, si posee tabla de contenidos y un ítem para relacionarlo con los contenidos de EAFIT Interactiva.

EIT_SCORM_SEGUIMIENTO

Esta tabla almacena el estado en que se encuentra un curso antes de ser completado para poder continuar la próxima vez que se inicie sesión.

EIT_SCORM_ITEM

Esta tabla almacena la información correspondiente a los SCO que hacen parte de un curso.

EIT_SCORM_COMENTARIOS

Esta tabla almacena los comentarios de una actividad, siendo de gran utilidad para desarrollos futuros y mejoras que se quieran hacer al sistema.

EIT_SCORM_OBJETIVOS

Esta tabla almacena los objetivos de un estudiante determinado, es de gran utilidad para desarrollos futuros y mejoras que se quieran hacer al sistema.

EIT_SCORM_ESTADOCURSO

Esta tabla almacena el estado del desarrollo de un curso, permite conocer si se ha completado el curso, el resultado de la evaluación de un estudiante.

EIT_SCORM_SSP

Esta tabla se deja para desarrollos futuros en este estándar cuando se desee compartir datos entre SCO.

7.5. FASE DE TRANSICIÓN

7.5.1. Manual de usuario

7.5.1.1. Manejo de estándar SCORM en EAFIT Interactiva

Este manual da una visión general de las principales características del estándar SCORM en EAFIT Interactiva. Permite a los usuarios entender la estructura del sistema y el flujo de trabajo durante el uso de la aplicación.

Utilizando el sistema

Para acceder al sistema vaya a la URL <http://interactiva.eafit.edu.co>. En la página principal debe ingresar el usuario y la clave asignados por la Universidad EAFIT (mismo usuario y clave del correo electrónico).



Figura 45. Página de inicio al sistema

Luego el sistema muestra la página de inicio con la lista de todas las materias asignadas para el semestre. Seleccione alguna de las materias que tiene activas como docente y en la cual desea ingresar un nuevo curso desarrollado en SCORM. Para este ejemplo seleccionamos la materia PROYECTO DE GRADO – SIMEVI.



The screenshot shows the EAFIT Interactiva user interface. At the top, there is a navigation bar with links for Inicio, Tutoriales, Contacto, and Cerrar Sesión. The user is logged in as Ana Sofía Gallo Vargas. The main content area is divided into three sections: a left sidebar with navigation options (Personal, Administración, etc.), a central profile section for the user, and a right sidebar with news and announcements.

Personal

- Editar Mi Perfil >
- Administración
- Dar Permisos >
- Eliminar Permisos >
- Disponibilidad Horaria >

Bienvenido(a)
ANA SOFIA GALLO VARGAS

Materias activas como estudiante

- GEMIS - OE
- ST0078 - GERENCIA DE SISTEMAS (033)
- PY0021 - GESTION DE PROYECTOS INFORMATI (032)
- ST0765 - MERCADEO (031)
- ORGANIZACIÓN ESTUDIANTIL
- ST0061 - PROYECTO (031)
- REPRESENTANTES ESTUDIANTILES

Materias activas como docente

- PROYECTO DE GRADO - SIMEVI

Noticias EAFIT Interactiva

CURSO DE EXCEL PARA ESTUDIANTES
MICROSOFT EXCEL AVANZADO PARA ESTUDIANTES:
Fechas: 10 de Octubre al 07 de Noviembre, Viernes, 8:00 a 12:00 m
Intensidad: 20 Horas.

MICROSOFT EXCEL AVANZADO PARA ESTUDIANTES:
Fechas: 25 de Octubre al 22 de Noviembre, Sábados, 8:00 a 12:00 m
Intensidad: 20 Horas

MICROSOFT EXCEL COMO HERRAMIENTA FINANCIERA PARA ESTUDIANTES:
Fechas: 18 de Octubre al 15 de Noviembre, Sábados, 8:00 a 12:00 m
Intensidad: 20 Horas

TODOS LOS CURSOS TIENEN UNA INVERSION DE \$150.000

Evaluación a la Docencia
Señor Estudiante lo invitamos a realizar la Evaluación a la Docencia la cual esta activa desde el 09 de Septiembre hasta el 22 de Noviembre, último día de finales.

Figura 46. Página de inicio

Después de seleccionar la materia el sistema muestra las opciones generales. Haga click en el enlace de contenidos que se encuentra al lado izquierdo.



The screenshot displays the EAFIT Interactiva web interface. At the top, there is a navigation bar with links for 'Inicio', 'Tutoriales', 'Contacto', and 'Cerrar Sesión'. Below this, the 'EAFIT Interactiva' logo is visible on the left, and a search bar with the text 'Ir a la materia' is on the right. The main content area is titled 'PROYECTO DE GRADO - SIMEVI'. On the left side, there is a vertical navigation menu with categories: 'El Curso' (containing 'Contenidos', 'Anuncios', 'Bibliografía', 'Enlaces', 'FAQ', 'Glosario', 'Wiki'), 'Interacción' (containing 'Correo', 'Foro', 'Chat'), 'Evaluación' (containing 'Recepción Trabajos', 'Exámenes en Línea'), and 'Herramientas' (containing 'Agenda', 'Mosaico', 'Frecuencia de Uso'). The 'Contenidos' section is expanded to show 'Información Grupos'. On the right side, there are two boxes: 'Usuarios en Línea' showing a user profile for Ana Sofía Gallo Vargas, and 'Noticias' showing two news items with dates and authors.

Figura 47. Página de opciones generales del sistema

El sistema muestra todos los contenidos que tiene la materia en la que se encuentra.



The screenshot shows the EAFIT Interactiva website interface. At the top, there is a navigation bar with links for 'Inicio', 'Tutoriales', 'Contacto', and 'Cerrar Sesión'. The user is logged in as 'Ana Sofía Gallo Vargas'. The main header features the 'EAFIT Interactiva' logo and a search bar. The page title is 'PROYECTO DE GRADO - SIMEVI'. A left sidebar menu includes sections for 'El Curso' (with sub-items like 'Contenidos', 'Anuncios', 'Bibliografía', 'Enlaces', 'FAQ', 'Glosario', 'Wiki'), 'Interacción' (with 'Correo', 'Foro', 'Chat'), 'Evaluación' (with 'Recepción Trabajos', 'Exámenes en Línea'), and 'Herramientas' (with 'Agenda', 'Mosaico', 'Frecuencia de Uso'). The main content area is titled 'Contenidos' and lists three items: 'Anteproyecto' (modified 2008/04/10 10:01 P.M.), 'SCORM' (modified 2008/04/19 07:28 A.M.), and 'Curso 1' (modified 2008/10/05 02:20 P.M.). On the right side, there are two boxes: 'Herramientas Edición' with an 'Editar Vista' button, and 'Usuarios en Línea' showing the current user's profile.

Figura 48. Página de contenidos de una materia

Guardar un curso SCORM en EAFIT Interactiva

Seleccione la opción de editar vista que se encuentra al lado derecho. A continuación el sistema muestra varias opciones para editar un contenido, seleccione la opción de SCORM, la cual sirve para cargar un nuevo curso en EAFIT Interactiva.



The screenshot shows the EAFIT Interactiva interface. At the top, there is a navigation bar with links for Inicio, Tutoriales, Contacto, and Cerrar Sesión. Below this is a header area with the EAFIT logo and the text 'EAFIT Interactiva'. A dropdown menu for 'Ir a la materia' is visible. The main content area is titled 'SISTEMA DE MENTORÍA VIRTUAL - CENTRO DE IDIOMAS' and 'Contenidos'. On the left, there is a sidebar menu with categories like 'El Curso', 'Interacción', and 'Evaluación'. The main area displays a list of three content items, each with a dropdown menu, a title (e.g., 'prueba'), a 'Ver Contenido' link, and a status indicator 'Activo: Si'. A right-hand menu offers options for creating new content: Elemento, Carpeta, Enlace Externo, Enlace Interno, Podcast, SCORM, and Terminar Edición.

Figura 49. Página para editar un contenido

Luego el sistema solicita el título y la ruta del curso, recuerde que el archivo debe estar comprimido con extensión .zip. Después de ingresar estos datos presione la opción de guardar.



The screenshot shows the EAFIT Interactiva web interface. At the top, there is a navigation bar with links for Inicio, Tutoriales, Contacto, and Cerrar Sesión. The main header features the EAFIT logo and the text 'EAFIT Interactiva'. Below the header, a blue bar indicates the user is logged in as 'Profesor Idiomas' and provides a dropdown menu to 'Ir a la materia'. The main content area is titled 'SISTEMA DE MENTORÍA VIRTUAL - CENTRO DE IDIOMAS' and 'Contenidos'. The primary task is 'Importar Archivo SCORM', with a note that asterisks indicate required fields. The form includes: a 'Título *' field; an 'Archivo *' field with a 'Browse...' button; and a 'Validación' section with radio buttons for 'Si' (selected) and 'No'. A green tooltip with the text '¿qué es esto?' is visible near the title field. At the bottom of the form are 'GUARDAR' and 'REGRESAR' buttons. A sidebar on the left contains navigation links for 'El Curso', 'Interacción', and 'Evaluación'. A 'Usuarios en Línea' box in the top right shows the current user's session details.

Figura 50. Página para importar un archivo SCORM

A continuación se muestra la página de importar archivo SCORM con los datos solicitados ingresados.



The screenshot shows the EAFIT Interactiva website interface. At the top, there is a navigation bar with links for 'Inicio', 'Tutoriales', 'Contacto', and 'Cerrar Sesión'. The main header features the EAFIT logo and the text 'EAFIT Interactiva'. Below the header, a blue banner reads 'SISTEMA DE MENTORÍA VIRTUAL - CENTRO DE IDIOMAS'. On the left side, there is a vertical menu with categories like 'El Curso', 'Interacción', and 'Evaluación'. The main content area is titled 'Contenidos' and contains a section for 'Importar Archivo SCORM'. This section includes a warning: 'Los campos marcados con asterisco (*) son obligatorios.' The form has three main fields: 'Titulo *' with the value 'Curso 1', 'Archivo *' with a file path 'C:\Users\Luisal\Documen' and a 'Browse...' button, and 'Validación' with radio buttons for 'SI' (selected) and 'No'. At the bottom of the form are 'GUARDAR' and 'REGRESAR' buttons. A 'Usuarios en Línea' box in the top right corner shows 'Profesor Idiomas' and the timestamp '(2008/10/05 02:03 P.M.)'.

Figura 51. Página de importar archivo SCORM con los datos

Si el curso cumple con los requisitos determinados por el estándar SCORM el sistema muestra una pantalla informando que el curso ha sido cargado con éxito.



The screenshot shows the EAFIT Interactiva interface. At the top, there is a navigation bar with links for Inicio, Tutoriales, Contacto, and Cerrar Sesión. Below this is the EAFIT logo and the text 'EAFIT Interactiva'. A dropdown menu for 'Ir a la materia' is visible. The main content area is titled 'SISTEMA DE MENTORÍA VIRTUAL - CENTRO DE IDIOMAS' and 'Contenidos'. A message states 'El curso ha sido importado' with a 'REGRESAR' button. A sidebar on the left lists various course sections like 'El Curso', 'Interacción', and 'Evaluación'. A 'Usuarios en Línea' box shows one active user: 'Profesor Idiomas' at '2008/10/05 02:03 P.M.'.

Figura 52. Página de información del resultado del análisis del curso

Visualizar un curso SCORM en EAFIT Interactiva

En la página que muestra los contenidos de una materia, seleccionar la opción de ver contenido, que se encuentra debajo del título del curso desarrollado en SCORM, en este caso Curso 1.

El sistema abre una nueva ventana y por defecto muestra la tabla de contenidos del curso para que el usuario seleccione el ítem que desea ver.

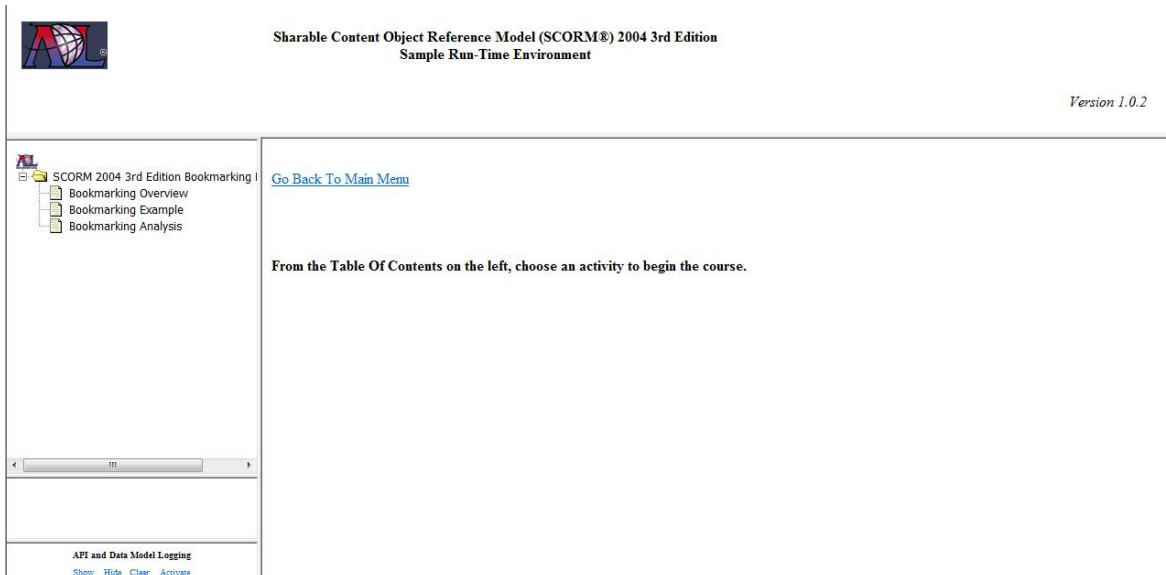


Figura 53. Página de inicio del curso

Después de seleccionar el ítem del lado izquierdo que se desea visualizar el sistema muestra la página del curso que corresponde.

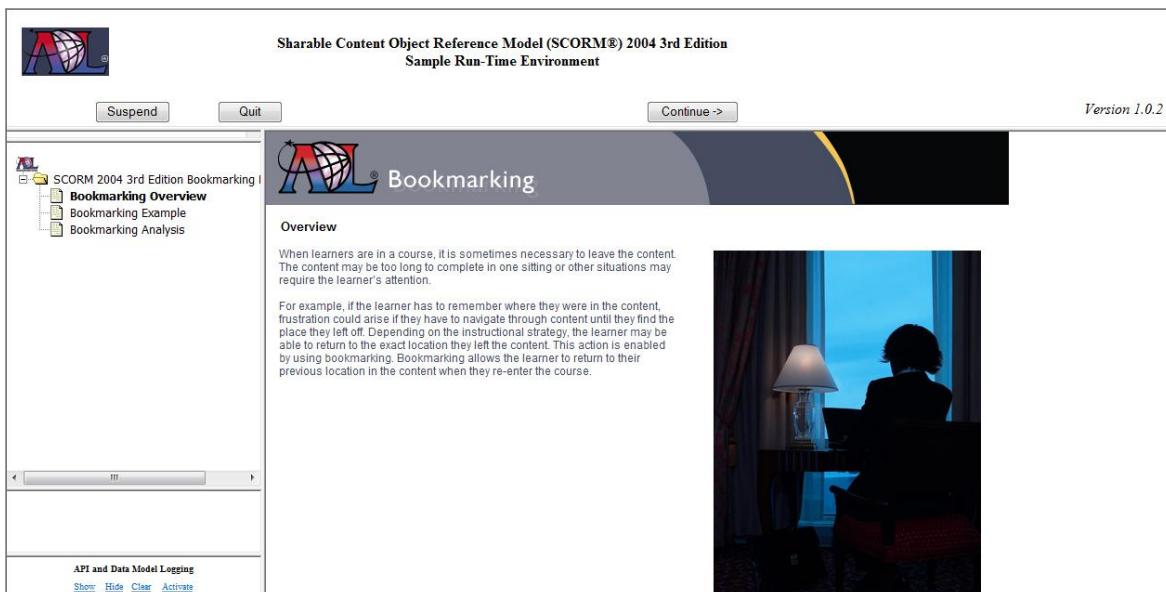


Figura 54. Página que muestra un ítem del curso

Si desea salir del curso tiene dos opciones:

- ☞ Suspenderlo que le permite la próxima vez que ingrese al curso continuar en el lugar donde estaba anteriormente.
- ☞ Salir que no almacena el ítem en el que se encontraba y la próxima vez que abra el curso muestra la página de inicio.

Después de seleccionar alguna de estas dos opciones el sistema muestra una página de notificación de que la sesión de que la sesión del curso ha terminado.

Presione la opción de regresar al menú principal para regresar a EAFIT Interactiva.

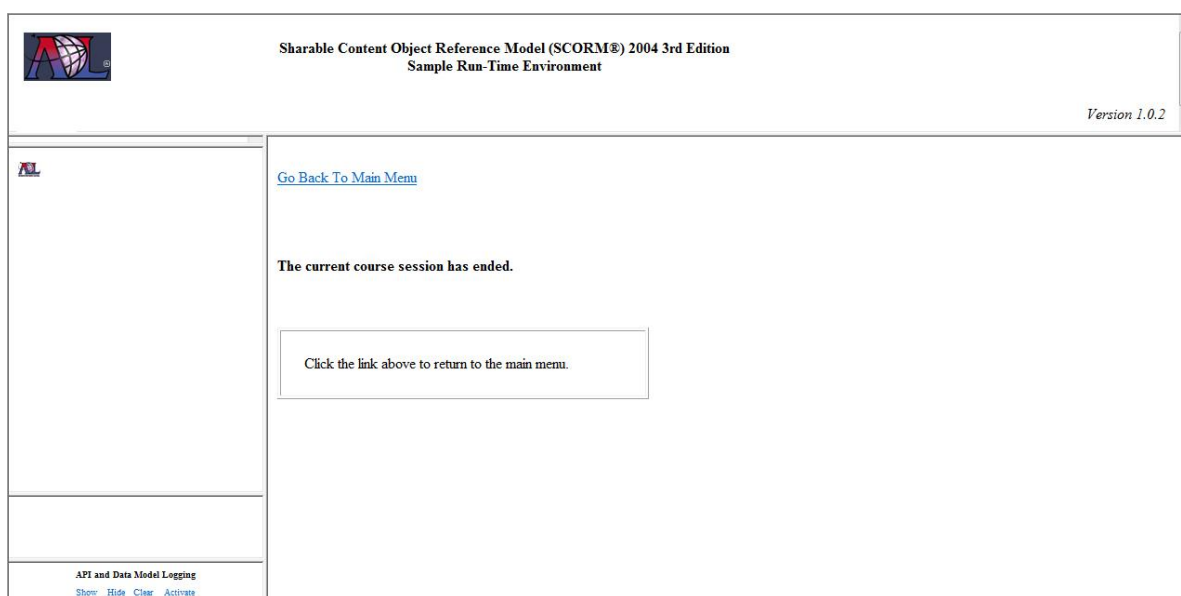


Figura 55. Página de finalización del curso

7.5.1.2. Manejo de las plantillas para desarrollar un curso en SCORM

Este manual da una visión general de cómo se debe crear un curso para que cumpla con los requisitos del estándar SCORM.

Para crear este curso se utiliza como base una carpeta con el nombre “Template”, la cual permite tener una organización básica de los elementos que hacen parte del curso.

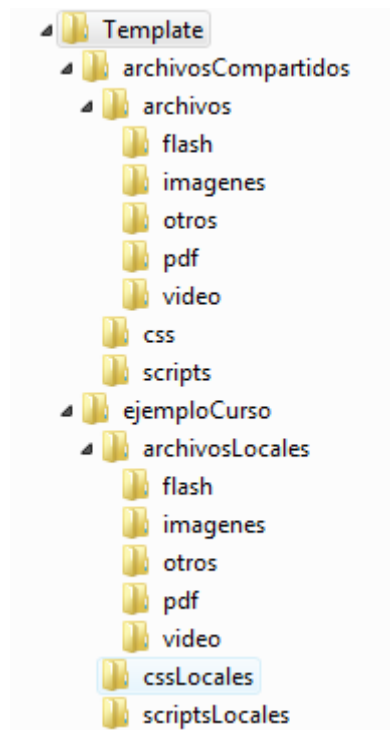


Figura 56. Organización de la carpeta "Template"

En la carpeta archivosCompartidos se pueden agregar todos los archivos que van a utilizar la mayoría de las páginas HTML del curso.

En la carpeta archivos se pueden agregar todos los videos, imágenes, flash, pdf y archivos de otro tipo que van a utilizar la mayoría de las páginas HTML.

En la carpeta css se pueden agregar todas las hojas de estilo que van a ser utilizadas por la mayoría de las paginas HTML para dar el diseño del contenido (tamaño de letra, color, etc).

En la carpeta scripts se encuentran los archivos en javascript necesarios para lograr la comunicación con el API de SCORM, estos archivos se encuentran incluidos en la carpeta.

La carpeta ejemploCurso muestra cómo debe ser la organización de un curso, en la carpeta archivos locales irían los diferentes archivos que solamente utilizaría la pagina HTML que se encuentra en esa carpeta, ocurre de igual manera para cssLocales y scriptsLocales.

Para este ejemplo vamos a hacer una copia de la carpeta ejemploCurso y le vamos a cambiar el nombre por introducción, esta carpeta debe contener las mismas carpetas que la carpeta ejemploCurso. Recuerde que por cada página HTML debe crear una carpeta con el nombre de cada uno de los ítems que hacen parte del curso.

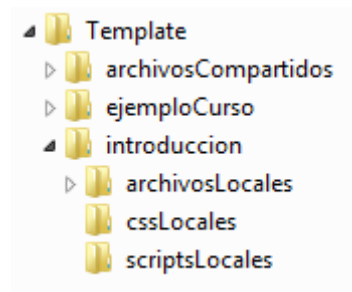


Figura 57. Cambio del nombre de la carpeta

En esta carpeta cree un archivo HTML en blanco con el nombre que usted desee, con cualquier editor de páginas de HTML, en este caso le vamos a poner el nombre de introduccion.html. Luego, ingrese el código que se muestra a continuación, el cual sirve para tener una comunicación con el API. Igualmente en la carpeta ejemploCurso se encuentra un archivo ejemplo.html que contiene este código.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html>
<head>
<title>Aquí va el título del ítem</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1"/>
<script type="text/javascript" src="../archivosCompartidos/scripts/APIWrapper.js"></script>
<script type="text/javascript" src="../archivosCompartidos/scripts/courseFunctions.js"></script>
<link rel="stylesheet" type="text/css" href="../archivosCompartidos/css/style.css" />
<!--Aquí se incluye el archivo de estilos si lo tiene-->
</head>

<body onload="javascript:Initialize();" onunload="javascript:Terminate();"><!--Estos metodos son la llamada para iniciar la
comunicacion con el API se encuentran especificados en el archivo APIWrapper.js-->
</body>
```

Figura 58. Código de ejemplo de la página HTML

Cambiamos el título de la página HTML por Introducción, por lo tanto, el código de la página quedaría de esta manera: <title>Introducción</title>.

Para crear en la página HTML los botones para navegar, se ingresa el siguiente código (este código debe ir después de la etiqueta <body>):

```
<div id="previousBtn">
  <a href="javascript:PreviousPage();"></a><!--Funcion del script para llamar a la pagina anterior-->
</div>
<div id="nextBtn">
  <a href="javascript:NextPage();"></a><!--Funcion del script para llamar a la pagina siguiente-->
</div>
```

Figura 59. Código de ejemplo para crear los botones de navegación

Si desea ingresar un archivo flash a la página HTML puede utilizar el siguiente código, cambiando los valores de acuerdo a sus necesidades:

```
<div style="margin:15px auto; width:762px">
<!--Este es un ejemplo de codigo para adjuntar un archivo flash a la pagina-->
  <object id="swf2" type="application/x-shockwave-flash" data="archivosLocales/flash/Intro_bkmg.swf"
width="762" height="440">
    <param name="movie" value="images/Intro_bkmg.swf" />
    <param name="wmode" value="transparent" />
  </object>
</div>
```

Figura 60. Código de ejemplo para incluir un archivo flash

Usted tiene la posibilidad de crear dentro de una misma página HTML, varias secciones que se muestran una después de la otra, para hacerlo puede utilizar el siguiente código:

```
<div id="p1">  
  <h2><!--Aquí va el título de la página--></h2>  
  <p><!--Aquí va el contenido de la primera página--></p>  
</div>
```

Figura 61. Código de ejemplo para crear una sección

Para crear una nueva sección simplemente copie el código y cambie el nombre de id por "p2" o el que desee, para crear una nueva sección simplemente realice el mismo procedimiento y cambie el nombre del id por p3, y así sucesivamente por el número de secciones que tenga. El id es un identificador único para cada sección, por lo tanto, no puede repetir el nombre.

Cuando termine de crear el curso, debe utilizar un programa para la creación del archivo manifest, existen varias opciones en la web nuestra recomendación es el RELOAD Editor. Después de tener el archivo manifest, debe comprimir el archivo con la extensión .zip y subirlo a la plataforma de EAFIT Interactiva.

7.5.1.3. Manejo de Reload Editor para crear el archivo manifest

Para crear un paquete que sea compatible con el estándar SCORM 2004 3rd Edition vamos a utilizar el software Reload Editor versión 2.5.5. Este software se puede descargar de la página: <http://www.reload.ac.uk/editor.html>

Paso 1: Copiar los archivos

Vamos a utilizar como base para este ejemplo la carpeta template, la cual tiene la información básica para crear un curso. Después de tener las diferentes paginas que hacen parte del curso copiar la carpeta template a una nueva dirección sin la carpeta ejemploCurso.

Igualmente puede cambiar el nombre de la carpeta template por el nombre que usted desee, para este ejemplo vamos a cambiar el nombre por Curso1.

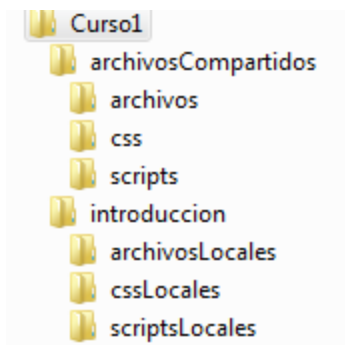


Figura 62. Organización de la carpeta Curso1

Para este ejemplo utilizaremos dos páginas HTML y dos imágenes, recuerde que es importante tener determinado como está organizado el curso y el nombre de los archivos que hacen parte del mismo.

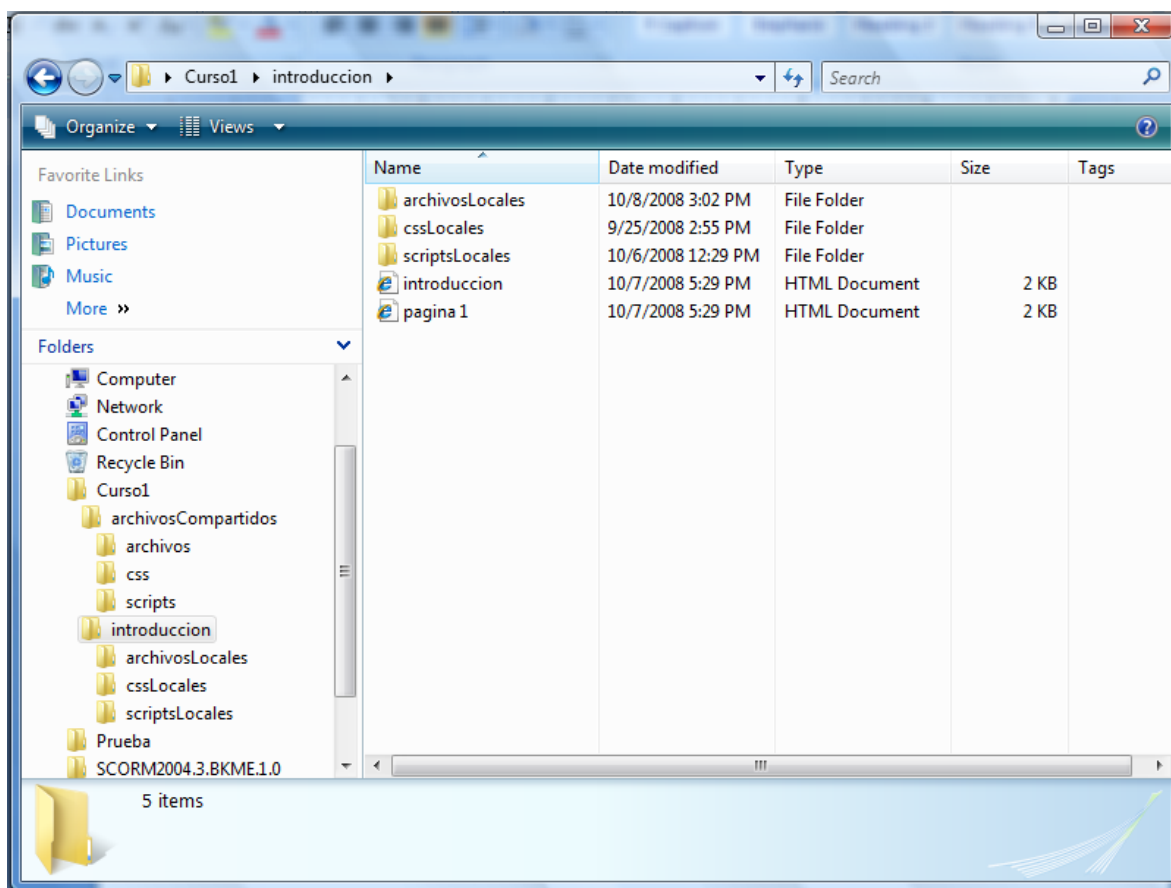




Figura 63. Páginas que hacen parte del curso

Paso 2: Crear un nuevo paquete ADL SCORM 2004

Después de abrir el Reload Editor, vamos a crear un paquete SCORM 2004 en blanco (esta herramienta crear unos archivos por defecto que son necesarios para que el paquete cumpla con las especificaciones del estándar).

Las acciones de este paso son:

-  Seleccionar el icono “New”, el cual se encuentra en la parte superior izquierda.
-  Seleccionar la opción “ADL SCORM 2004 package”.

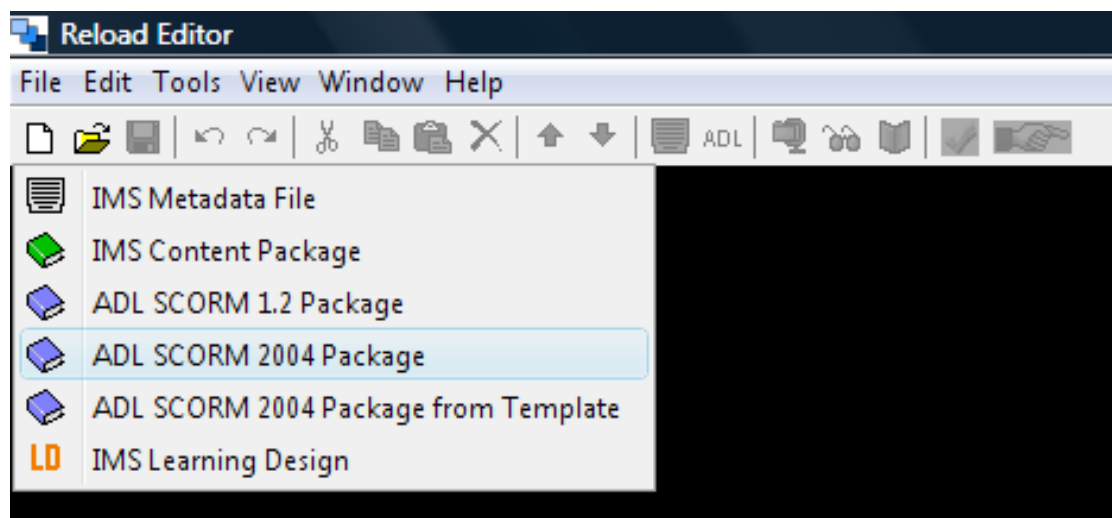


Figura 64. Creación paquete SCORM 2004 en RELOAD

- ☞ En el cuadro "Select Folder for New SCORM 2004 Package" seleccionar la carpeta donde desea crear el paquete y presione el botón "Select". Para este ejemplo seleccionamos la carpeta Curso1.

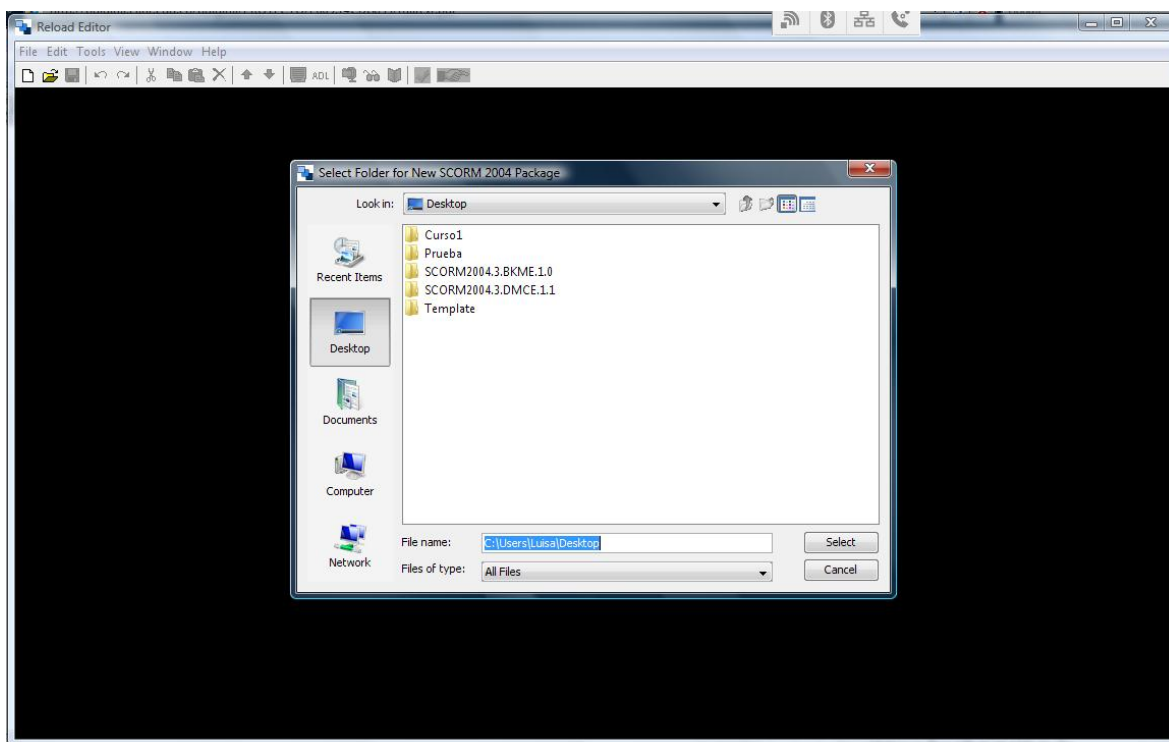


Figura 65. Ventana para seleccionar la carpeta donde se va a almacenar el paquete

La ventana principal del Reload Editor tiene tres secciones: el panel de recursos en la izquierda, el panel del manifest a la derecha, y el panel de atributos en la parte inferior.

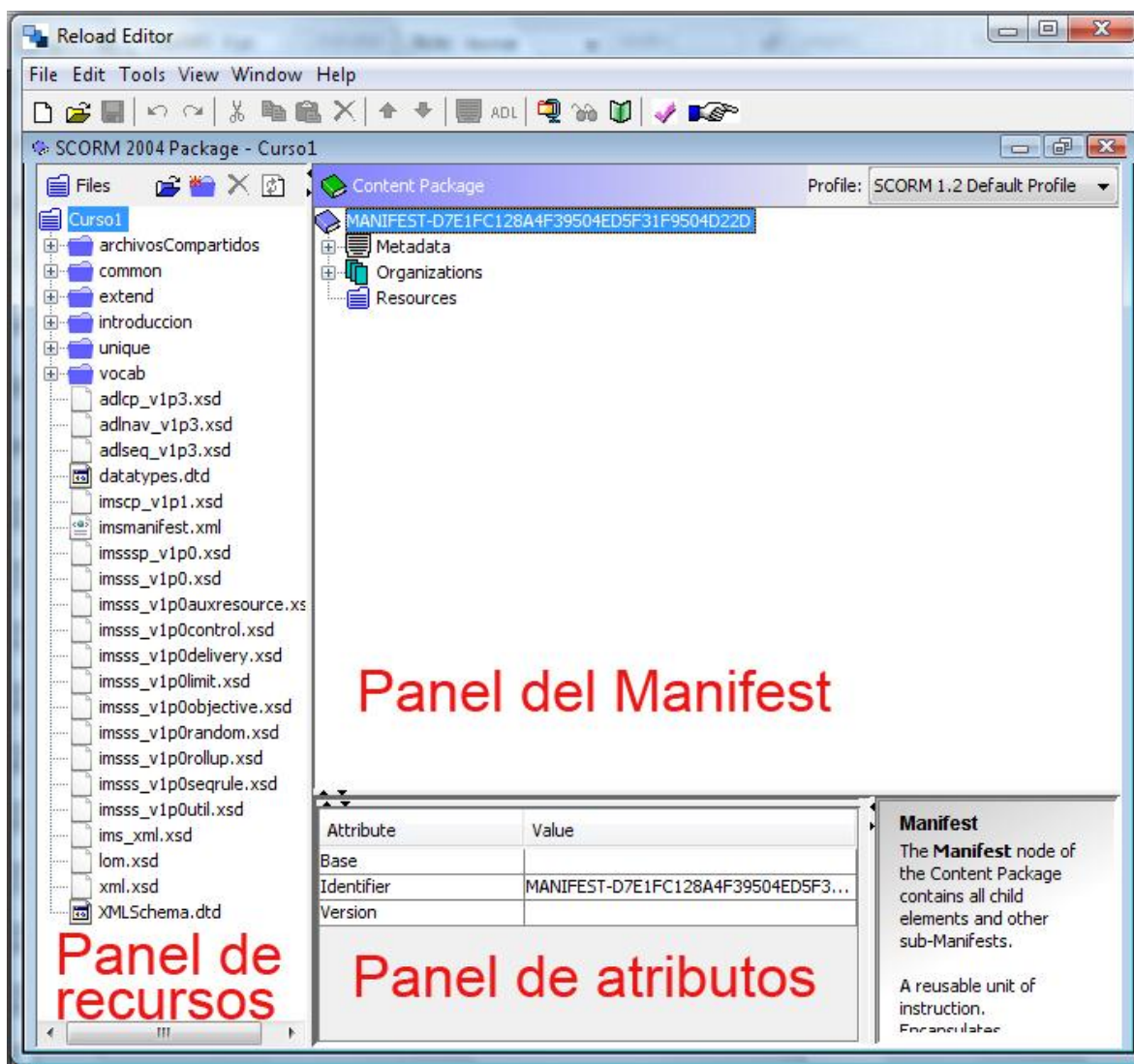


Figura 66. Secciones del Reload

Paso 3: Configurar el espacio de trabajo

Antes de copiar todos los recursos que hacen parte del curso debe cambiar el perfil que aparece por defecto al momento de seleccionar la carpeta, para hacer esto debe seleccionar la pestaña que se encuentra ubicada en la parte superior derecha con el nombre de "Profile" (el cual tiene por defecto "SCORM 1.2 Default Profile") y cambiarlo por "SCORM 1.3 Default Profile".

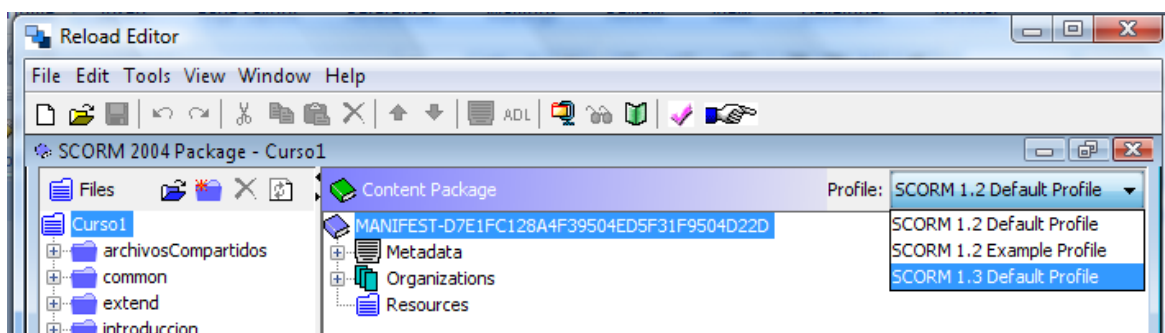


Figura 67. Cambio del perfil por defecto

Luego seleccione el icono de Metadata, expándalo presionando el signo mas que se encuentra a la izquierda de él y seleccione el icono que aparece como 1.2.

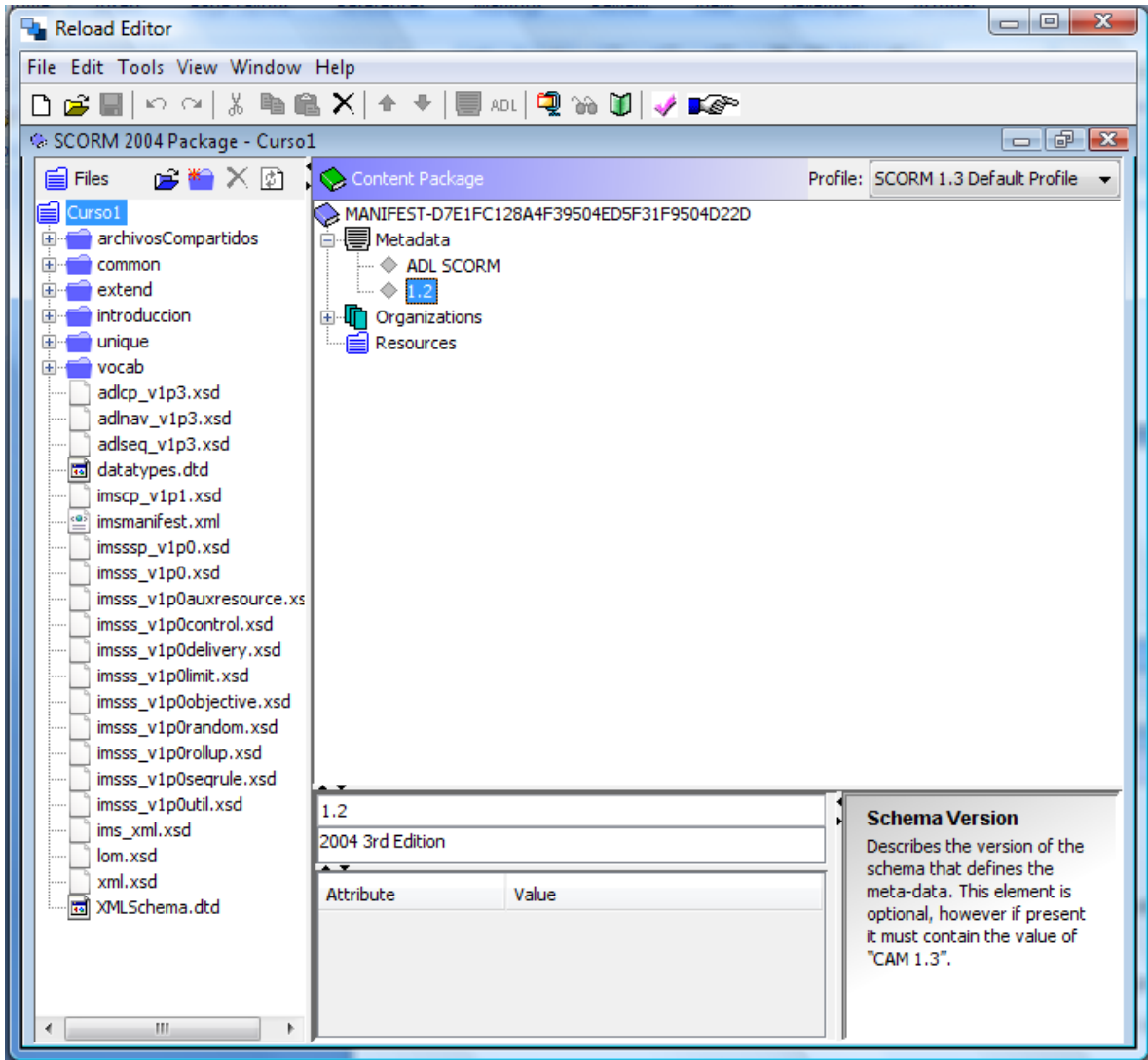


Figura 68. Ítems que hacen parte del manifest

Luego seleccione la opción que aparece en el panel de atributos como 2004 3rd Edition, esto se hace para que el paquete que se va a crear cumpla con los estándares de la versión SCORM 2004 3rd Edition.

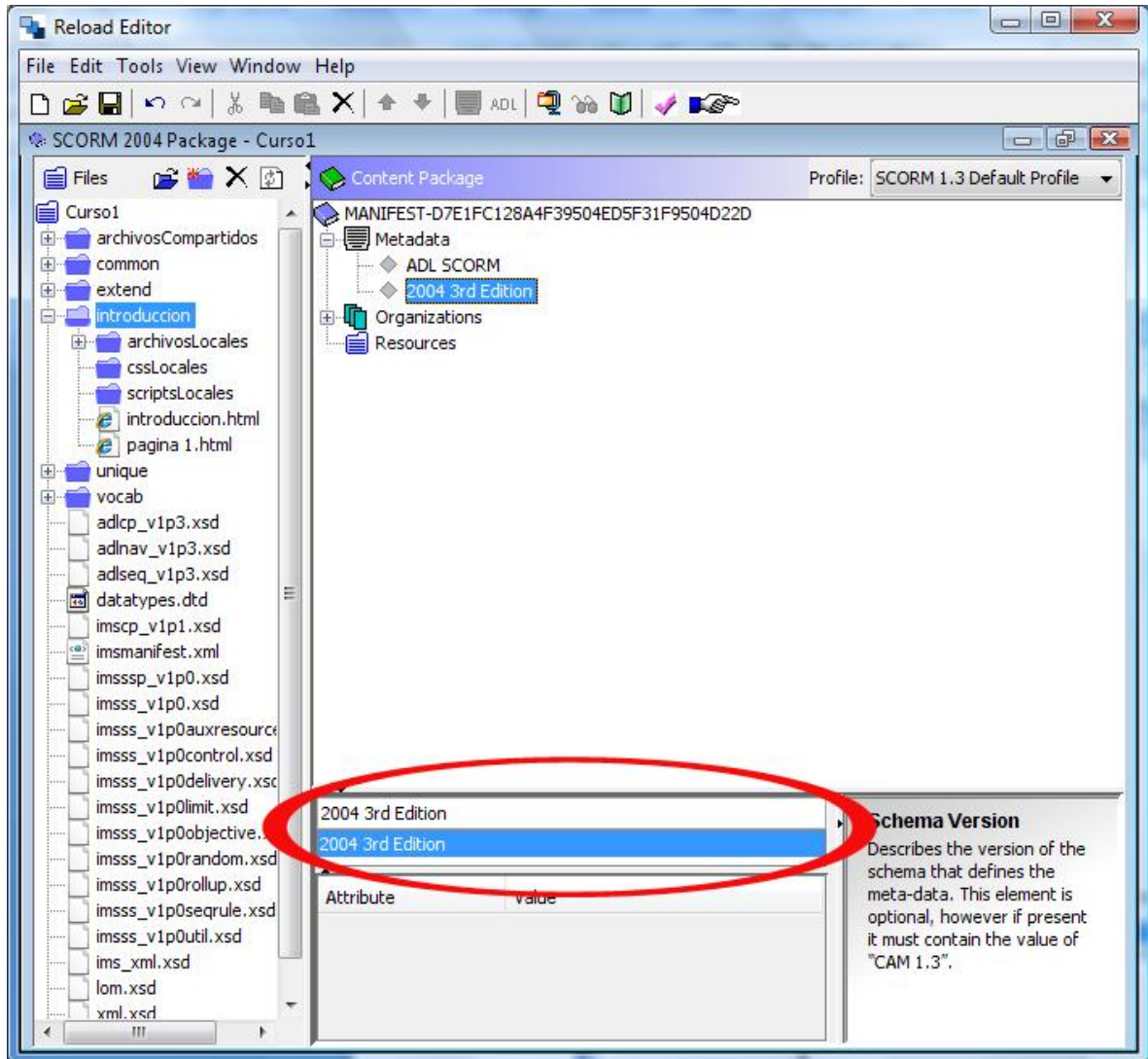




Figura 69. Cambio de la versión del esquema

Paso 4: Crear los recursos

En este paso se van a añadir al paquete las paginas HTML que permiten ejecutar cada una de las unidades (para este ejemplo introduccion.html y pagina 1.html).

En el panel de recursos en la carpeta introducción se encuentran todos los archivos que hacen parte del curso.

Las acciones de este paso son:

-  Hacer clic en el ítem “introduccion.html” y arrastrarlo hasta el ítem “Resources” en el panel del manifest.
-  Realizar esta misma acción para el ítem “pagina 1.html”.

Al arrastrar estas páginas el Reload se encarga de incluir todos los archivos que están relacionados con cada una de ellas.

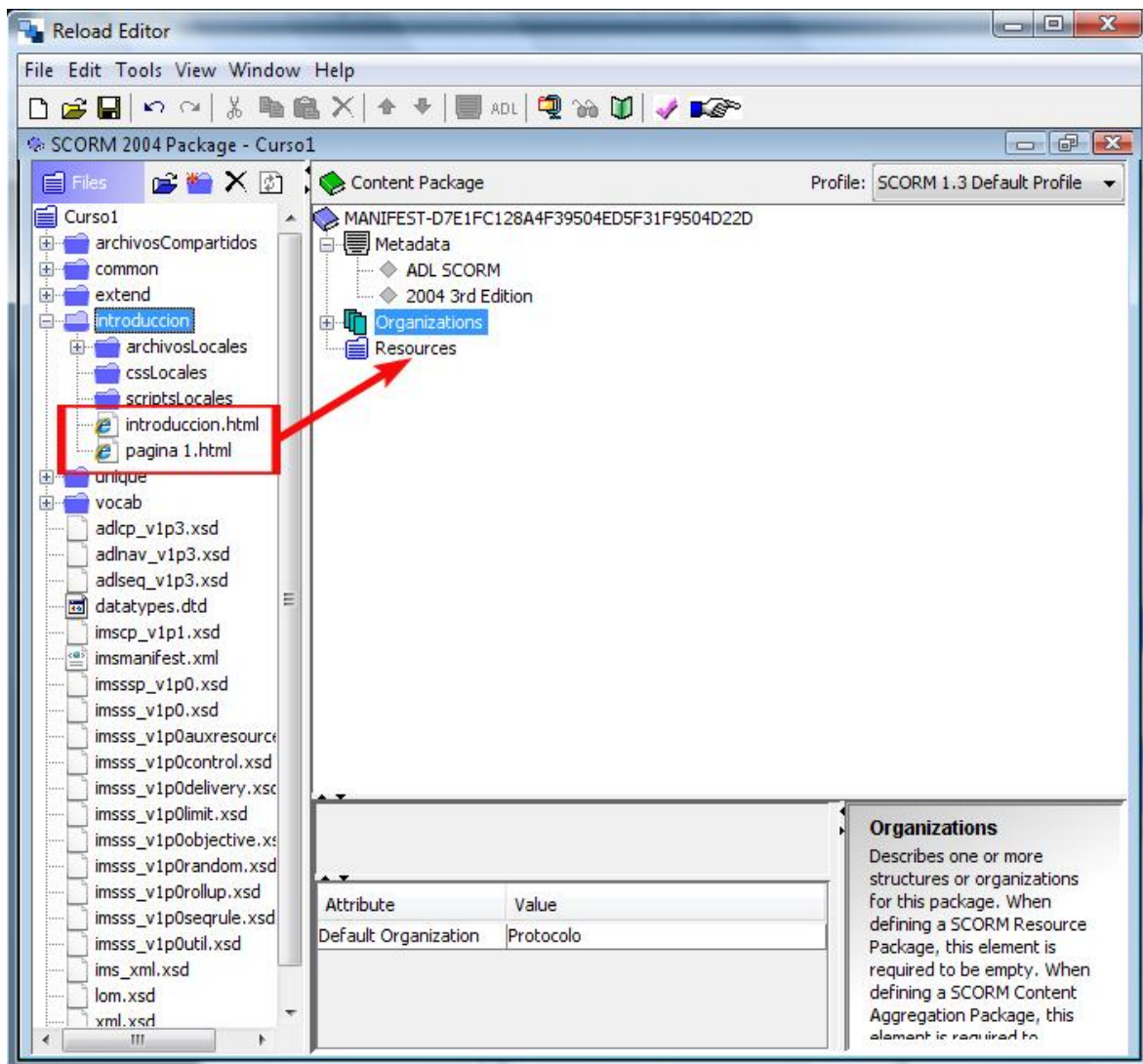


Figura 70. Ítems que deben ser arrastrados a recursos

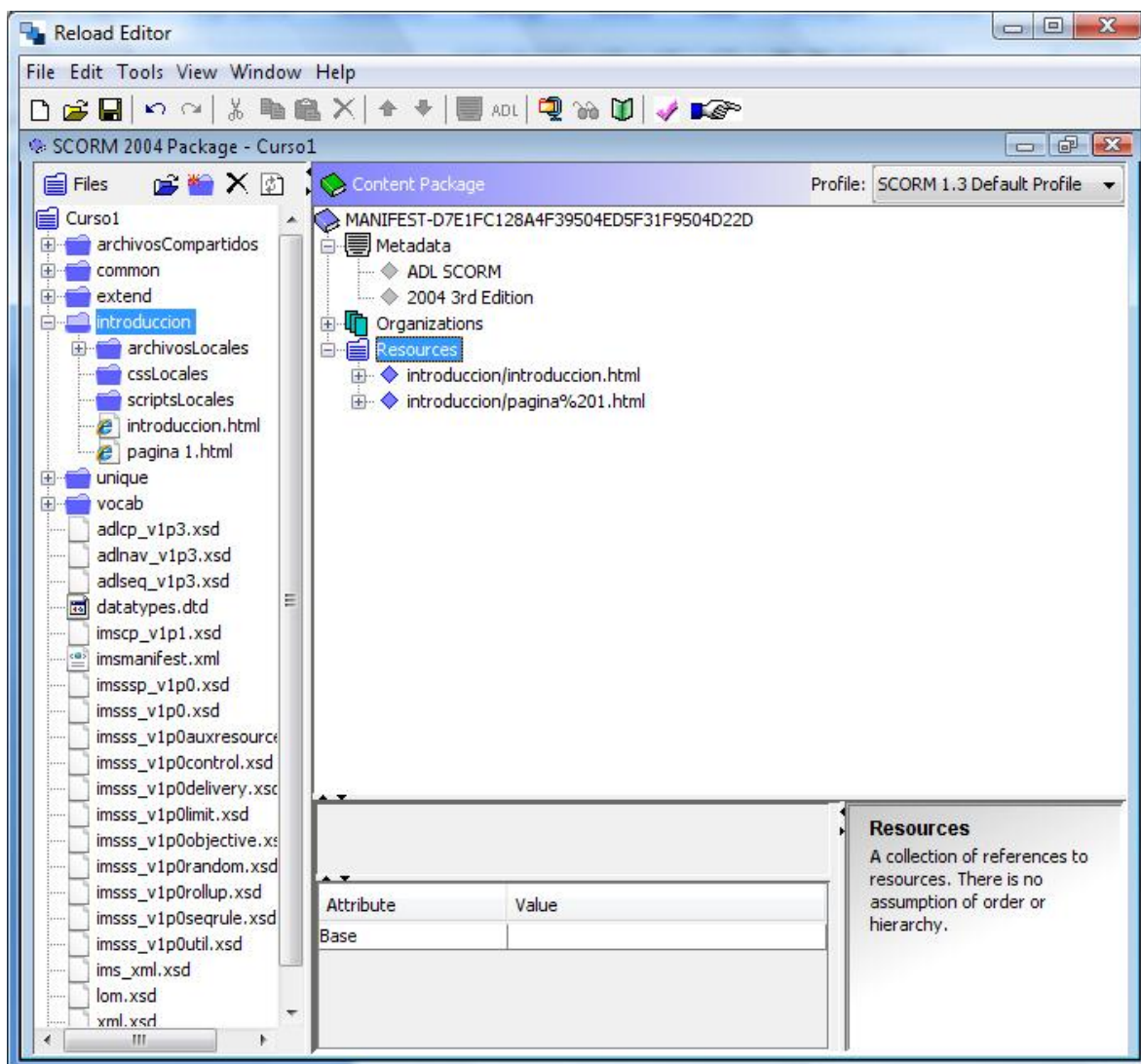


Figura 71. Recursos creados en el manifest

Paso 5: Cambiar el tipo de los recursos

Después de incluir los recursos en el panel de manifest es necesario determinar de qué tipo son estos recursos.

El estándar SCORM define dos tipos de contenidos:

- 📁 Assets: Son aquellos recursos a los cuales no se tendrá acceso desde la tabla de contenido (imágenes, flash, videos, audio, etc).
- 📁 SCO: Son aquellos recursos que poseen diferentes assets y a los cuales se puede acceder (página HTML).

Para nuestro ejemplo las dos páginas HTML corresponden a al tipo SCO.

Las acciones para este paso son:

- 📁 Seleccione el recurso
- 📁 En el panel de recursos seleccionar la fila con nombre “scormType” y cambiar la opción de “asset” por “sco”.

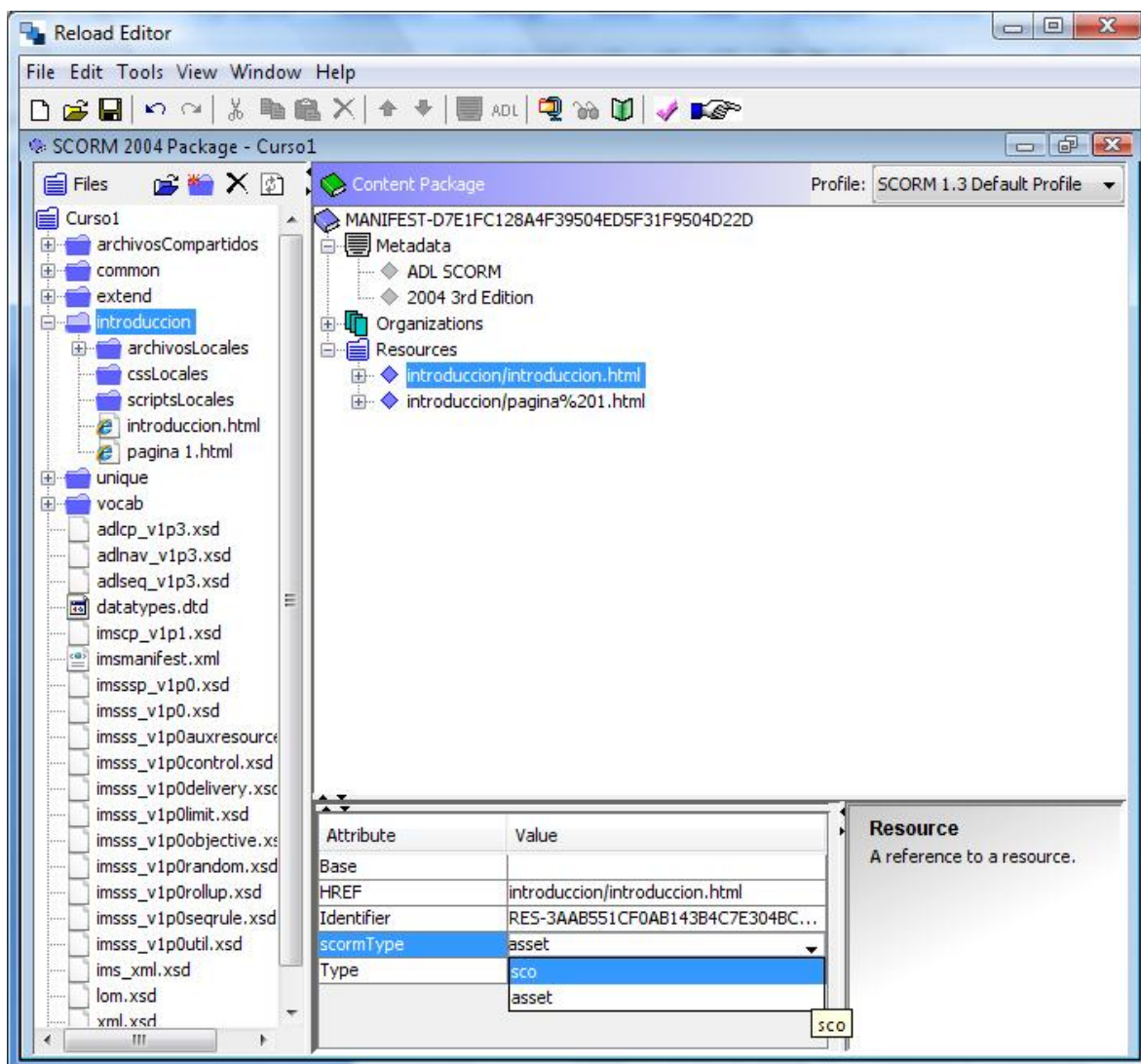


Figura 72. Cambio del tipo de recurso

Paso 6: Crear la tabla de contenidos




Para este ejemplo crearemos una tabla de contenidos, la cual va a estar organizada de la siguiente forma:

Unidad 1: Protocolo

Apartado 1: Introducción

Apartado 2: Ambientación

Las acciones para este paso son:

-  Expanda el icono de “Organizations” de la misma forma que se hizo con el metadata.
-  Expanda nuevamente el icono de “Organization” siguiendo los mismos pasos.
-  Seleccione el icono de ítem y cambie el nombre desde el panel de atributos por “Introducción”.

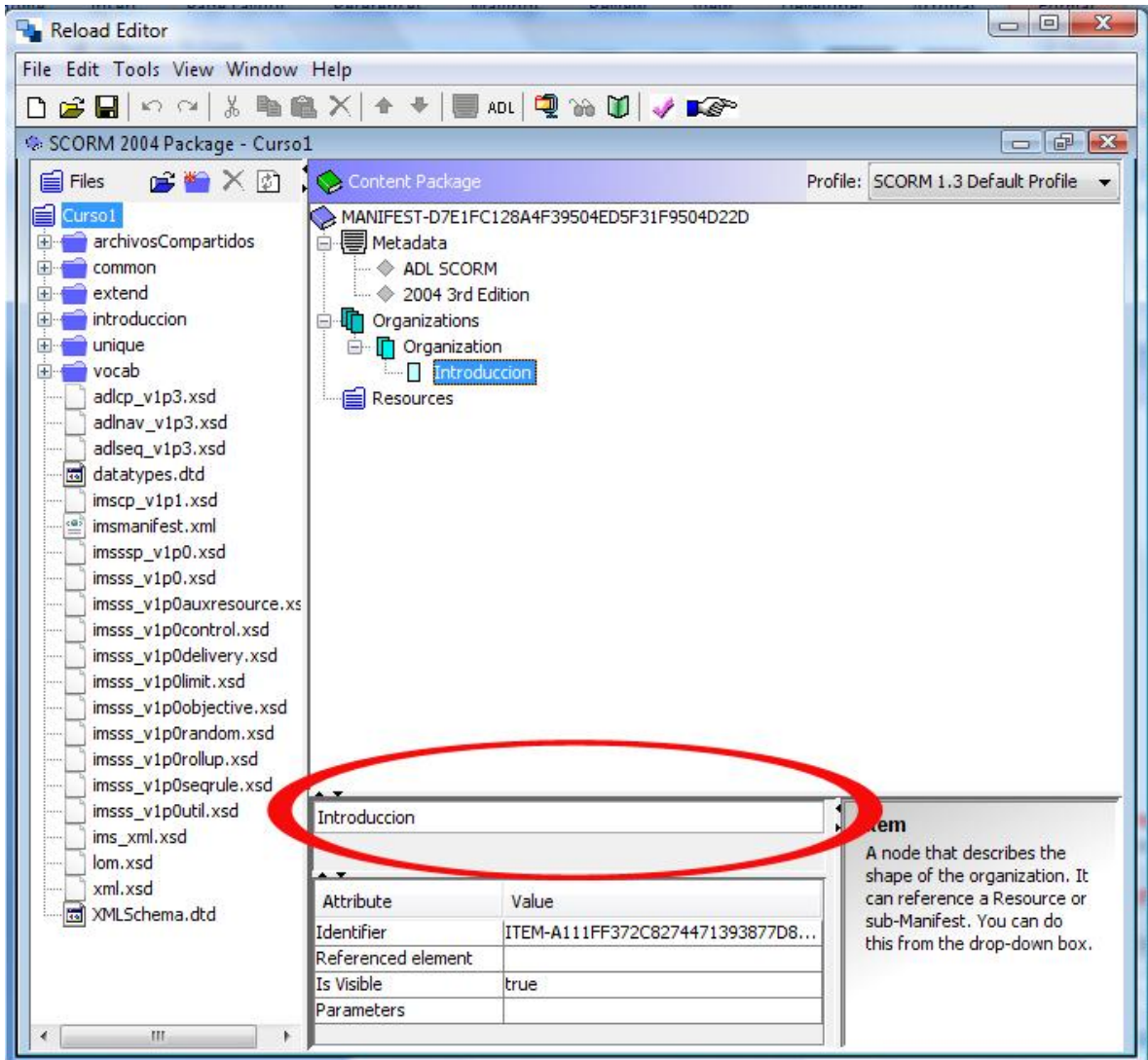


Figura 73. Creando el apartado 1

Haga clic en el botón derecho del mouse en el ítem de Organization y seleccione la opción de “Add Item”.

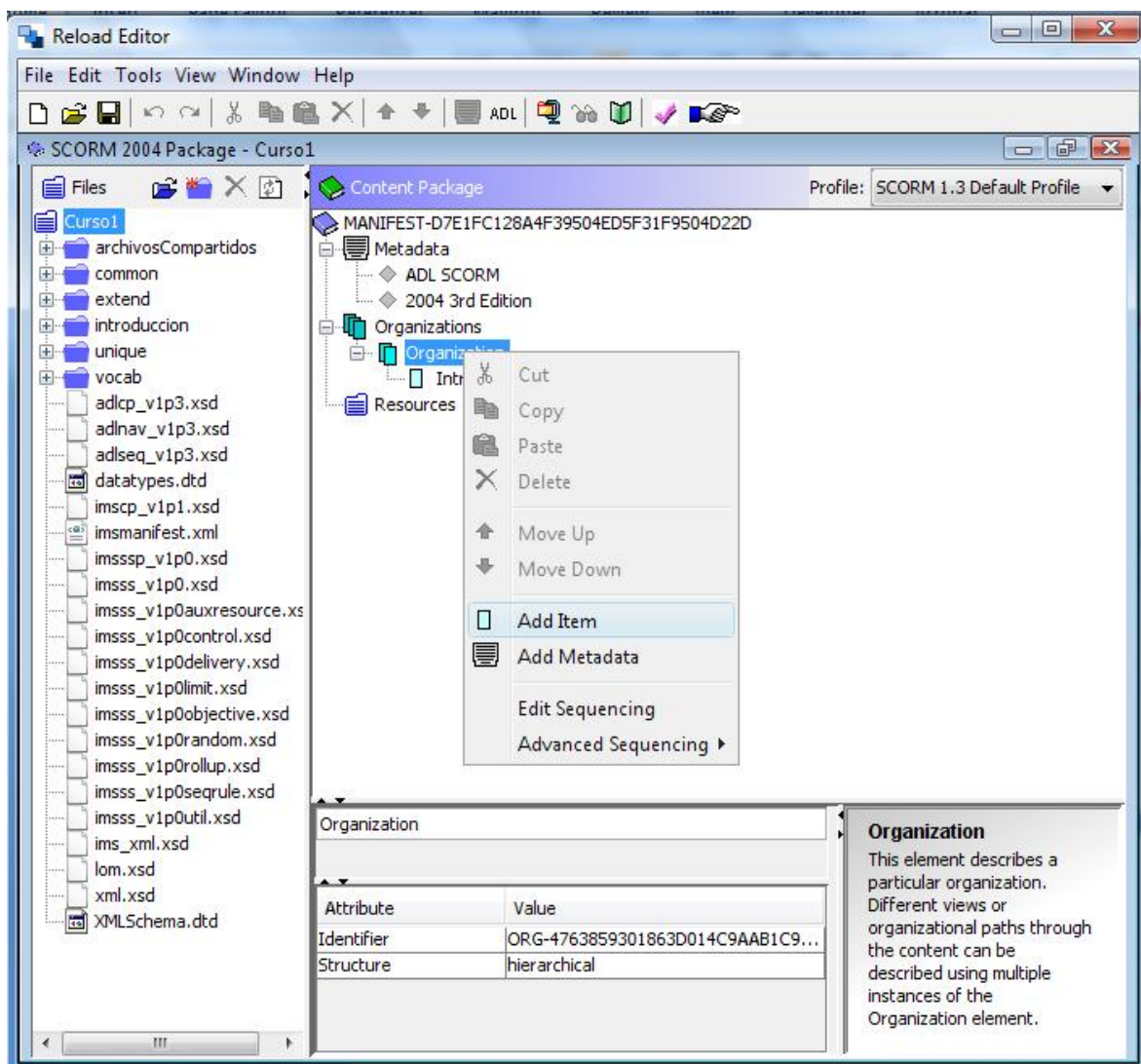


Figura 74. Pasos para crear un nuevo ítem

- ☞ Seleccione el icono ítem y cambie el nombre por el de “Ambientación”.
- ☞ Por último, seleccione el icono de “Organization” y cambie el nombre por el de “Protocolo”.

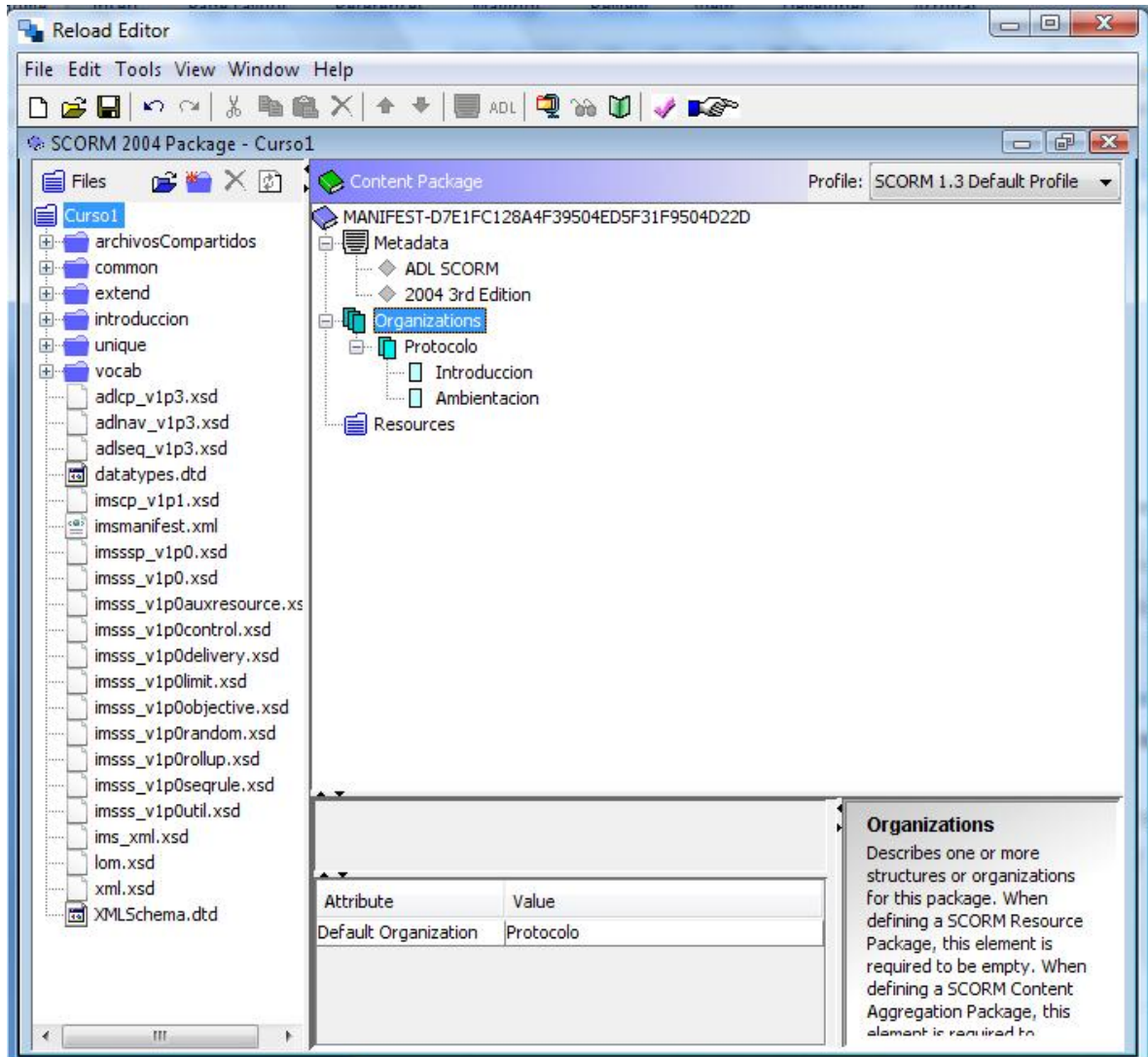


Figura 75. Vista de la tabla de contenidos

Estas son las acciones que se deben seguir para determinar que pagina HTML se ejecuta en cada ítem:

- ☞ Seleccione el ítem de “Introducción”
- ☞ En el panel de atributos en la fila “Referenced Element” seleccione el recurso correspondiente (para este ejemplo el recurso es “introduccion.html”).
- ☞ Realice estas mismas acciones para los demás ítems.

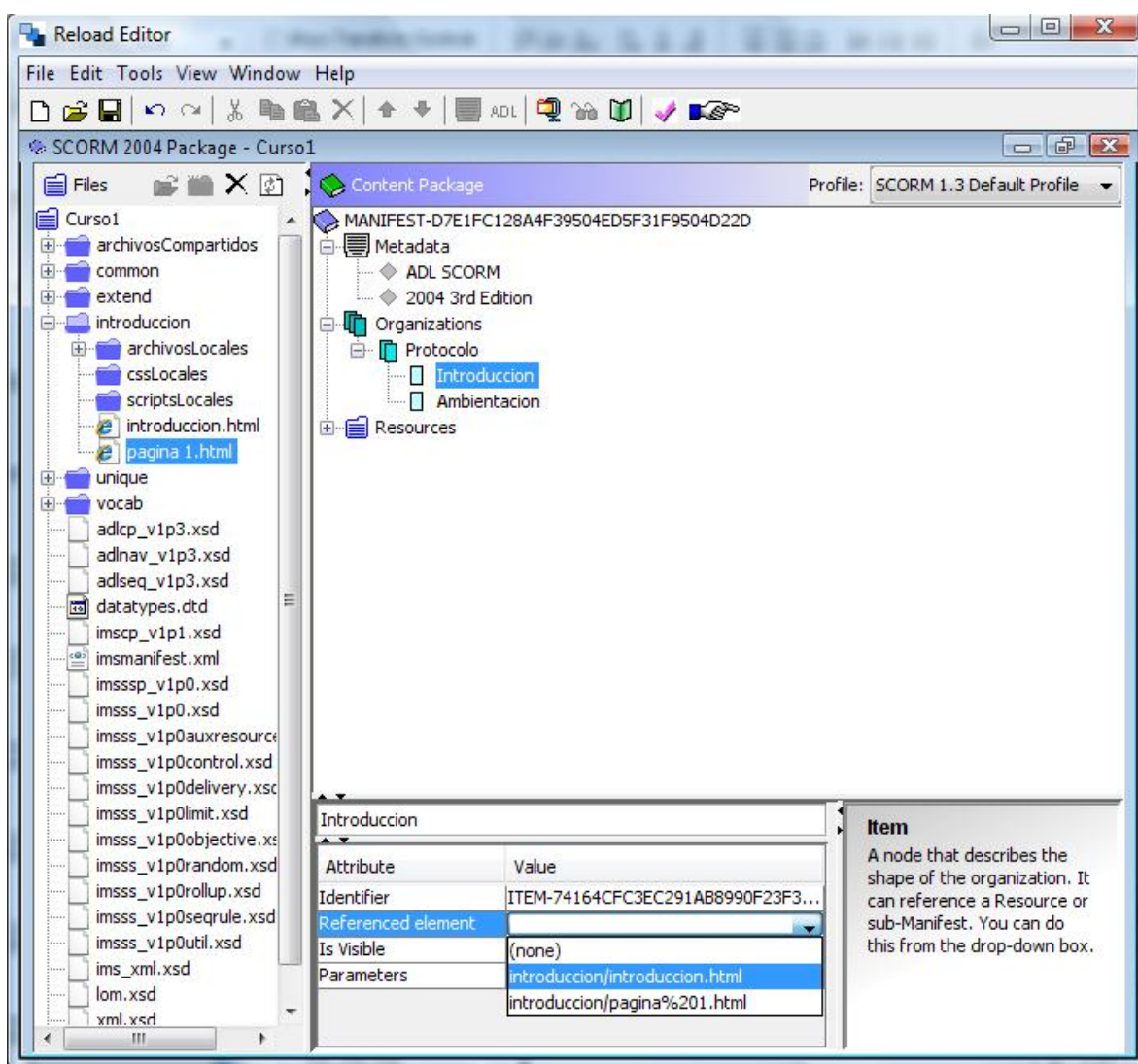


Figura 76. Pagina HTML que se ejecutara para el ítem introducción

Paso 7: Guardar, pre-visualizar y generar el paquete SCORM

En este paso ya se ha terminado de crear el paquete SCORM, por esto, ya se puede guardar el paquete, probarlo y generarlo.

Las acciones de este paso son:

☞ Seleccionar la opción “File” y hacer clic en la opción “Save”.

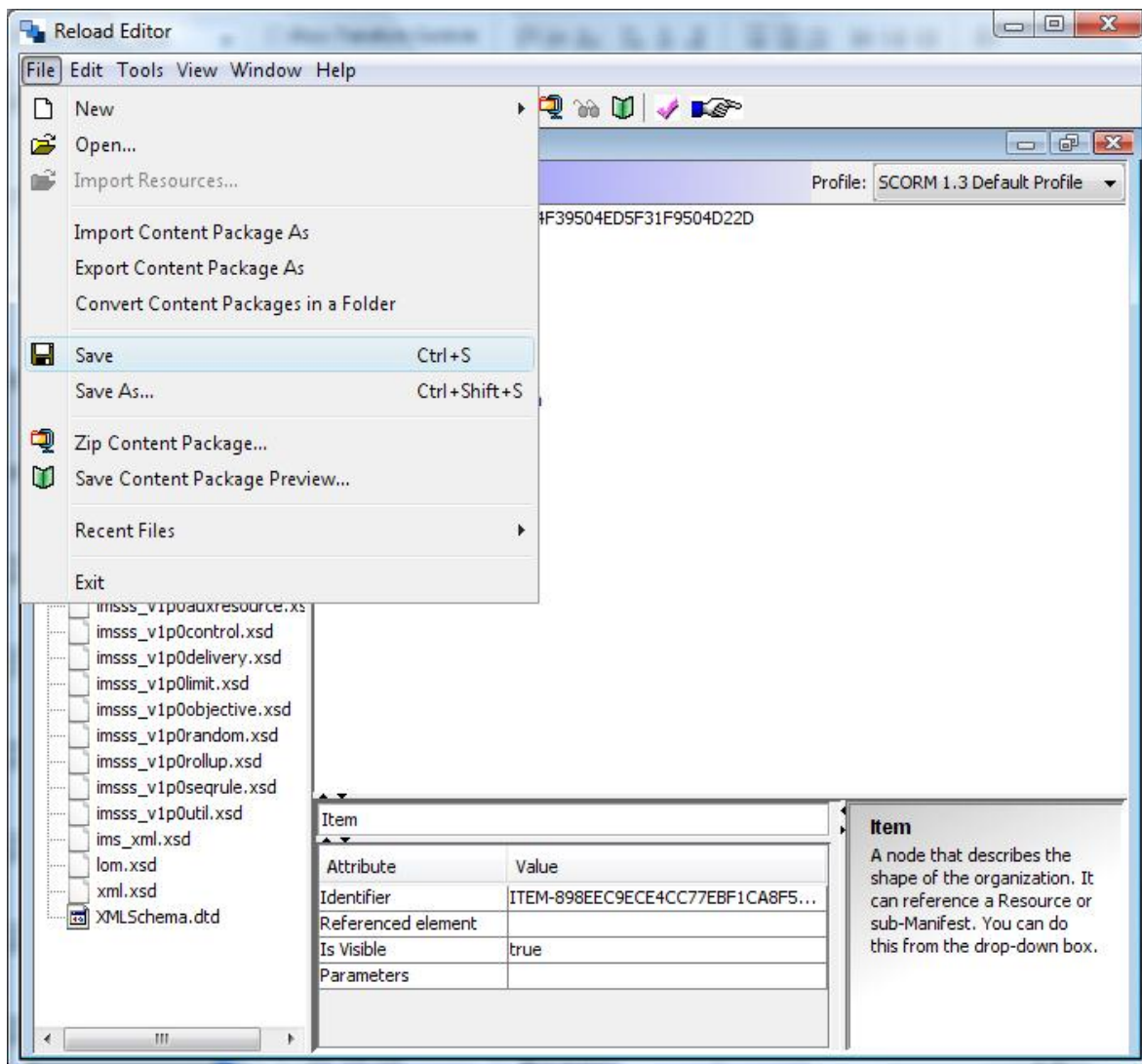


Figura 77. Pasos para guardar un paquete

☞ Seleccionar la opción “View” y hacer clic en “Preview Content Package”.

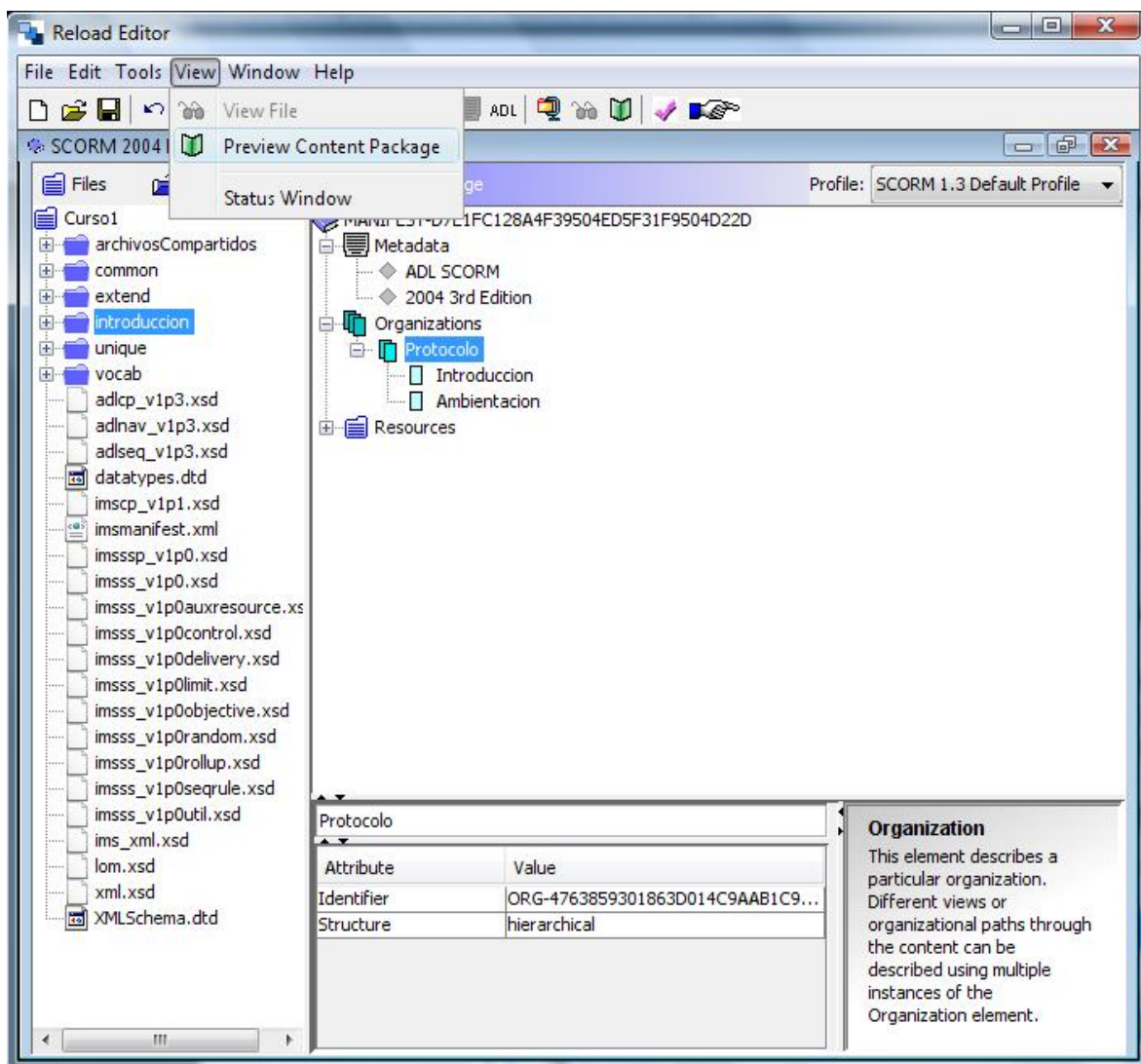


Figura 78. Pasos para ver el diseño final del curso


- Esta opción le permite tener una idea general de cómo será desplegado el curso en la plataforma de EAFIT Interactiva.



Figura 79. Pre-visualización del curso

Después de determinar que el curso se encuentra correcto, solo hace falta generar el paquete SCORM.

Las acciones para generar el paquete son:

-  Seleccionar la opción "File" y hacer clic en la opción "Zip Content Package".

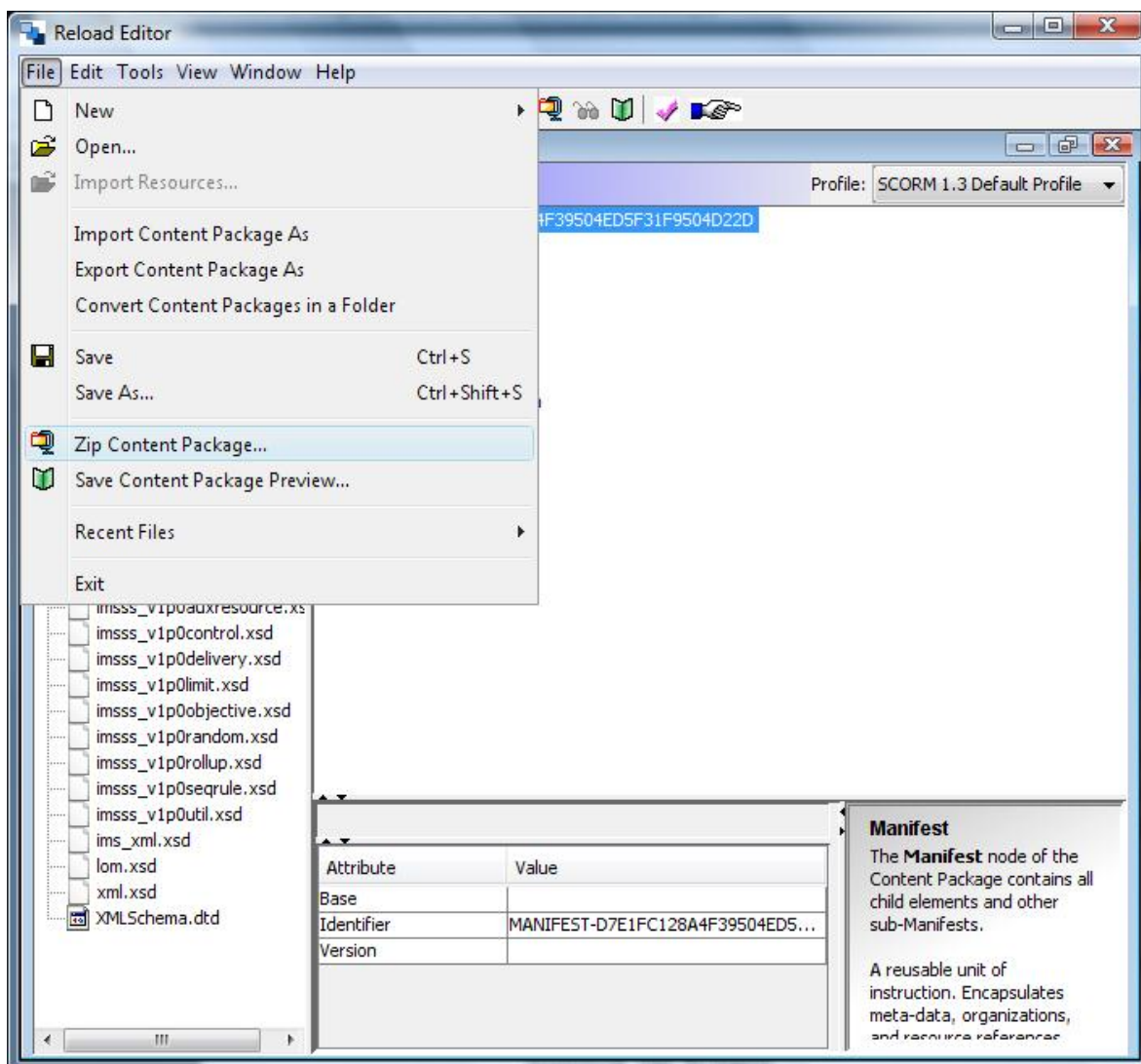


Figura 80. Pasos para empaquetar el curso

- 📁 En la ventana que aparece seleccionar la ubicación donde desea guardar el curso y el nombre del mismo.

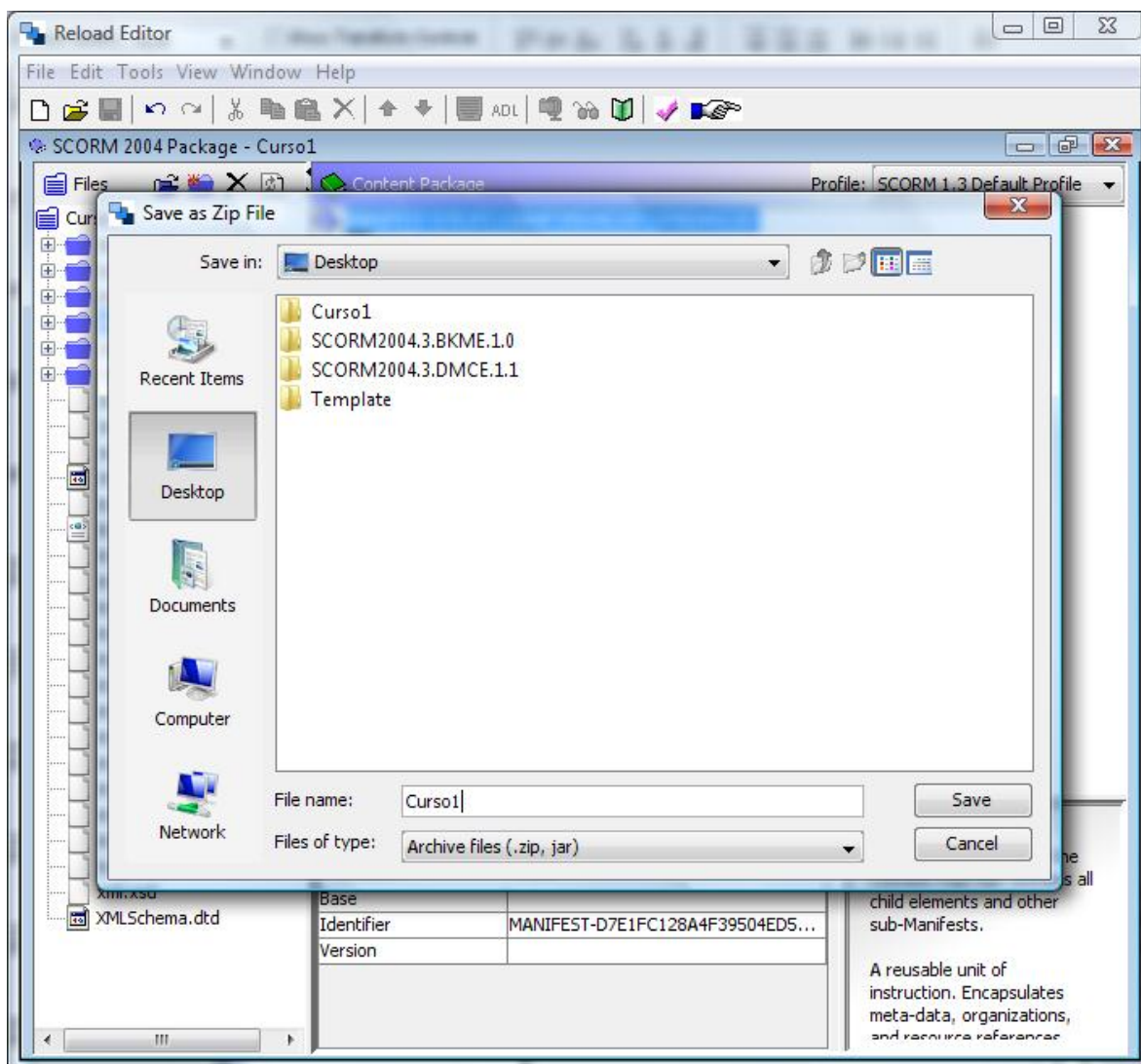


Figura 81. Ventana para guardar el paquete

Después de tener el paquete creado, solamente debe cargarlo en la plataforma EAFIT Interactiva y esta se encargará de mostrarlo.

8. RECOMENDACIONES FUTURAS

- ⌚ Una de las limitaciones que posee el prototipo entregado es que la creación de un curso que cumpla con el estándar SCORM es una tarea un poco complicada para un usuario que no esté familiarizado con el tema, pues se necesita de conocimientos básicos en informática y diseño de páginas Web. Por esta razón se recomienda diseñar una aplicación que facilite la creación de los cursos educativos de tipo SCORM. Este sistema podría integrarse a la plataforma EAFIT Interactiva y así, el usuario solamente se tendría que preocupar por proporcionar los recursos que desea mostrar en el curso (texto, videos, audio, imágenes, etc) y la acomodación de estas dentro de cada página Web.

- ⌚ En el presente trabajo de grado se entrega el modulo de protocolo de primer día de clase del curso virtual para la mentoría de docentes del centro de idiomas (SIMEVI), sin embargo, queda restando el desarrollo de los demás módulos del curso:
 - Planeación de clases: Plan de clase
 - Manejo de clase: Uso del tablero, asignación de turnos, técnicas de agrupación, lenguaje corporal, manejo del tiempo, estilos de aprendizaje.
 - Evaluación: Evaluación escrita, evaluación oral, retroalimentación, reporte de notas

Se adjunta el storyboard de los módulos faltantes.

- ⌚ Al tratarse de una aplicación pensada para el Centro de Idiomas, consideramos de gran importancia que el material desarrollado sea útil para todo tipo de docentes y esto incluye a los docentes nativos de la lengua extranjera y que no poseen un buen manejo del español. Por lo anterior es necesario que en un futuro la interfaces que permiten al usuario adicionar un curso de tipo SCORM a la plataforma EAFIT Interactiva sean desplegadas en diferentes idiomas.
- ⌚ Añadir una funcionalidad que permita modificar los cursos de tipo SCORM que ya han sido adjuntados previamente a EAFIT Interactiva sin tener que eliminar el curso y volverlo a cargar de nuevo.
- ⌚ Estudiar nuevas metodologías pedagógicas que permitan mejorar el curso entregado de protocolo de primer día de clase.
- ⌚ Crear una función de notificaciones que se apoye en el correo electrónico de la institución para notificar a los usuarios pertenecientes a una asignatura sobre la creación o eliminación de un curso de tipo SCORM.
- ⌚ Crear un manual en línea que facilite tanto la creación de un curso de tipo SCORM en EAFIT Interactiva como el diseño de estos de forma que cumplan el estándar requerido.
- ⌚ Realizar un estudio de usabilidad que permita concluir la forma en que los usuarios finales están percibiendo la funcionalidad desarrollada en EAFIT Interactiva y de esta forma considerar las mejoras que se le deben hacer en cuanto a interfaz y navegación.

- ⌚ Realizar pruebas piloto con los docentes del centro de idiomas al curso “Protocolo de primer día de clase” para modificar y mejorar el contenido presentado y el diseño del curso.

CONCLUSIONES

Se puede aprovechar las nuevas tecnologías de información, las teorías de ambientes virtuales de aprendizaje y las ventajas que ofrece la web 2.0 para potenciar las estrategias pedagógicas que tiene una institución educativa. De esta forma se puede observar que las plataformas web pueden ser un gran apoyo para llevar a cabo los procesos de enseñanza, propiciando entornos donde el tiempo y el espacio no son barreras para el aprendizaje.

Las herramientas tecnológicas de aprendizaje necesitan estar apoyadas en unos lineamientos metodológicos que sean acordes con las estrategias y los objetivos de la educación, de otra manera, tener una plataforma que apoye el proceso de aprendizaje no tendría sentido, pues se estaría entregando información sin ninguna finalidad. La tecnología es tan solo un apoyo, no la solución final.

Tener una buena comunicación con el cliente y los usuarios es crucial para lograr un buen desarrollo del producto. Al realizar este proyecto nos pudimos dar cuenta de que el intercambio de información debía ser clara y constante desde ambas partes. Desde el lado del cliente para poder comprender sus necesidades y deseos y desde la parte de los desarrolladores para entregar retroalimentaciones y avances sobre el estado de la aplicación. Así, la calidad del producto se puede mejorar porque el cliente conoce el estado del producto y puede ir haciendo las recomendaciones necesarias, dando como resultado un cliente satisfecho y más comprometido con el proyecto.

Aplicar un estándar para el manejo de contenidos de aprendizaje como el SCORM trae grandes ventajas y facilita el manejo de cursos virtuales, pues de esta forma la aplicación tendrá unas características mejor desarrolladas en cuanto a reusabilidad, accesibilidad y seguimiento al usuario.

Antes de desarrollar una aplicación es necesario conocer todo el modelo de negocio de la institución cliente, ya que sin hacer un análisis del entorno es imposible modelar una solución que se adapte a las necesidades específicas y a las características de los usuarios (para tal efecto se utilizaron las facilidades que provee construir los casos de uso en correlación con el cliente y sus requerimientos). Del mismo modo, realizar un proceso de planeación previo al desarrollo del producto es importante para distribuir de una forma mejor los recursos disponibles.

Estudiar y conocer las diferentes teorías de aprendizaje y los modelos pedagógicos propuestos por los diferentes estudiosos del área de la educación y la pedagogía ayuda a diseñar mejores cursos virtuales.

Utilizar herramientas y plataformas existentes para adaptarlas a las necesidades específicas de la aplicación solicitada trae mucha facilidad y ahorro de tiempo y esfuerzo, pues de esta forma se puede uno enfocar en las nuevas funcionalidades y en el modelado del negocio. En nuestro caso específico, utilizar y apoyarnos en EAFIT Interactiva para ofrecerle al cliente los servicios requeridos de interacción (chat, foros, correo, etc.) fue de gran utilidad porque no fue necesario implementar el sistema desde cero y no se necesitaron crear elementos ya existentes. De esta forma nos pudimos enfocar primordialmente en las necesidades específicas del centro de idiomas en cuanto al curso de mentoría y la forma de adaptar ese curso a EAFIT Interactiva, permitiendo acotar el problema y trayendo finalmente

beneficios que se vieron reflejados en el ahorro de tiempo y facilidad de la implementación.

BIBLIOGRAFÍA

1. **Sun Microsystems, Inc.** Code Samples and Apps - Applets. *Sun Developer Network (SDN)*. [En línea] [Citado el: 6 de Octubre de 2008.] <http://java.sun.com/applets/>.
2. XML DOM Tutorial. *W3 Schools Online Web Tutorials*. [En línea] [Citado el: 6 de Octubre de 2008.] <http://www.w3schools.com/dom/default.asp>.
3. **W3C Oficina Española.** Hojas de Estilo. *Oficina Española del W3C*. [En línea] 9 de Enero de 2008. [Citado el: 7 de Octubre de 2008.] <http://www.w3c.es/divulgacion/guiasbreves/HojasEstilo>.
4. DOM. *Mozilla*. [En línea] 8 de Agosto de 2008. [Citado el: 6 de Octubre de 2008.] <http://developer.mozilla.org/en/DOM>.
5. **Metacourse, Inc.** ¿Qué es eLearning? *Metacursos*. [En línea] Metacourse, Inc. [Citado el: 6 de Octubre de 2008.] <http://metacursos.com/elearning/index.htm>.
6. **Raggett, Dave, Le Hors, Arnaud y Jacobs, Ian.** HTML 4.01 Specification. *W3C - World Wide Web Consortium*. [En línea] W3C, 24 de Diciembre de 1999. [Citado el: 6 de Octubre de 2008.] <http://www.w3.org/TR/html401/present/frames.html>.

-
7. **Master, Web.** Acerca de IEEE. *IEEE Colombia*. [En línea] IEEE Colombia, 11 de Abril de 2007. [Citado el: 6 de Octubre de 2008.] <http://www.ieee.org.co/portal/content/view/12/27/>.
 8. **Imperial College London.** An E-Learning Glossary. *Information & Communication Technologies ICT*. [En línea] 10 de Diciembre de 2007. [Citado el: 6 de Octubre de 2008.] <http://www3.imperial.ac.uk/ict/services/teachingandresearchservices/elearning/aboutelearning/elearningglossary>.
 9. **Munera Sanchez, Mauricio.** *Implementación de una audionovela interactiva, un módulo de gramática y un software para la administración de ejercicios soportado en moodle y basado en el estándar SCORM para la creación de contenido para un curso de español como lengua extranjera*. Medellín : s.n., 2006.
 10. Tecnologías de la información y la comunicación. *Wikipedia La enciclopedia libre*. [Online] [Cited: Octubre 6, 2008.] http://es.wikipedia.org/wiki/Tecnolog%C3%ADas_de_la_informaci%C3%B3n_y_la_comunicaci%C3%B3n.
 11. Token (programación). *Wikipedia La enciclopedia libre*. [En línea] [Citado el: 6 de Octubre de 2008.] [http://es.wikipedia.org/wiki/Token_\(programaci%C3%B3n\)](http://es.wikipedia.org/wiki/Token_(programaci%C3%B3n)).
 12. **Diegues, Jorge.** Normativas y estándares para el tratamiento de contenidos. *WordPress*. [En línea] 2004. [Citado el: 5 de Octubre de 2008.] <http://jdiegues.wordpress.com/seleccion-de-recursos-scorm/normativas-y-estandares-para-el-tratamiento-de-contenidos/>.
 13. **Diaz Redondo, Rebeca P y Fernandez Vilas, Ana.** Analisis tecnico de estandares. *Proyecto SUMA, E-learning multimodal y adaptativo para la empresa*. 2007.

-
14. **Advanced Distributed Learning (ADL).** *Sharable Content Object Reference Model (SCORM®) 2004 3rd Edition Overview.* Virginia : s.n., 2006.
 15. —. *Sharable Content Object Reference Model (SCORM®) 2004 3rd Edition Content Aggregation Model Version 1.0.* Virginia : s.n., 2006.
 16. —. *Sharable Content Object Reference Model (SCORM®) 2004 3rd Edition Sequencing and Navigation Version 1.0.* Virginia : s.n., 2006.
 17. —. *Sharable Content Object Reference Model (SCORM®) 2004 3rd Edition Run-Time Environment Version 1.0.* Virginia : s.n., 2006.
 18. **Rosrnberg, M. J.** *E-learning strategies for delivering knowledge in the digital age.* s.l. : McGraw-Hill, 2001.
 19. **García Peñalvo, Francisco José.** Estado actual de los sistemas e-learning. *Universidad de Salamanca.* [En línea] [Citado el: 11 de Octubre de 2008.] http://www.usal.es/~teoriaeducacion/rev_numero_06_2/n6_02_art_garcia_penalvo.htm.
 20. Estándar SCORM. *Autore.* [En línea] 2005. [Citado el: 3 de Octubre de 2008.] <http://autore.ehu.es/term/term6.htm>.
 21. **SumTotal Systems.Inc.** Brief introduction to SCORM. *ToolBook.* [En línea] 2008. [Citado el: 3 de Octubre de 2008.] http://www.toolbook.com/community_scorm.php.
 22. **Arango Moreno, Carolina, Holguin Cano, Juan David y Moncada Fernandez, David.** *Implementación del prototipo funcional de una wiki en la plataforma académica EAFIT Interactiva.* Medellín : s.n., 2008.
 23. **Chadid Botero, Gerardo Alberto, Fenández López, Gustavo Adolfo y Pabón Ramírez, Carolina.** *Diseño de la arquitectura y prototipo funcional para*

comunidades educativas virtuales basadas en protocolos de internet. Medellín : s.n., 2003.

24. Advanced Distributed Learning. SCORM 2004 3rd Edition© Sample Run-Time Environment - API Documentation. 2007.

ANEXOS

Anexo 1: Código de los Java Script utilizados en el template del curso.

```
/*  
**  
** Advanced Distributed Learning Co-Laboratory (ADL Co-Lab) grants you  
** ("Licensee") a non-exclusive, royalty free, license to use and redistribute  
** this software in source and binary code form, provided that i) this copyright  
** notice and license appear on all copies of the software; and ii) Licensee  
** does not utilize the software in a manner which is disparaging to ADL Co-Lab.  
**  
** This software is provided "AS IS," without a warranty of any kind. ALL  
** EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND  
** WARRANTIES, INCLUDING ANY  
** IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A  
** PARTICULAR PURPOSE OR  
** NON-INFRINGEMENT, ARE HEREBY EXCLUDED. ADL Co-Lab AND ITS  
** LICENSORS SHALL  
** NOT BE LIABLE FOR ANY DAMAGES SUFFERED BY LICENSEE AS A  
** RESULT OF USING,  
** MODIFYING OR DISTRIBUTING THE SOFTWARE OR ITS DERIVATIVES. IN  
** NO EVENT WILL
```

** ADL Co-Lab OR ITS LICENSORS BE LIABLE FOR ANY LOST REVENUE,
PROFIT OR DATA,
** OR FOR DIRECT, INDIRECT, SPECIAL, CONSEQUENTIAL, INCIDENTAL OR
PUNITIVE
** DAMAGES, HOWEVER CAUSED AND REGARDLESS OF THE THEORY OF
LIABILITY, ARISING
** OUT OF THE USE OF OR INABILITY TO USE SOFTWARE, EVEN IF ADL Co-
Lab HAS BEEN
** ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

**

*****/

/*****

**

** This file is being presented to Content Developers, Content Programmers and
** Instructional Designers to demonstrate one way to abstract API calls from the
** actual content to allow for uniformity and reuse of content fragments.

**

** The purpose in wrapping the calls to the API is to (1) provide a
** consistent means of finding the LMS API adapter within the window
** hierarchy, (2) to ensure that the method calls are called correctly by the
** SCO and (3) to make possible changes to the actual API Specifications and
** Standards easier to implement/change quickly.

**

** This is just one possible example for implementing the API guidelines for
** runtime communication between an LMS and executable content components.
** There are many other possible implementations.

**

*****/

```
// local variable definitions used for finding the API
var apiHandle = null;
var findAPITries = 0;
var noAPIFound = "false";

// local variable used to keep from calling Terminate() more than once
var terminated = "false";

// local variable used by the content developer to debug
// This should be set to true during development to find errors. However,
// This should be set to false prior to deployment.
var _debug = false;

/*****
**
** This function looks for an object named API in parent and opener windows
**
** Inputs: Object - The Window Object
**
** Return: Object - If the API object is found, it's returned, otherwise null
**          is returned
**
*****/
function findAPI( win )
{
    while ( (win.API_1484_11 == null) &&
            (win.parent != null) &&
            (win.parent != win) )
```

```
{
    findAPITries++;

    if ( findAPITries > 500 )
    {
        alert( "Error finding API -- too deeply nested." );
        return null;
    }

    win = win.parent;
}

return win.API_1484_11;
}

/*****
**
** This function looks for an object named API, first in the current window's
** frame hierarchy and then, if necessary, in the current window's opener window
** hierarchy (if there is an opener window).
**
** Inputs: none
**
** Return: Object - If the API object is found, it's returned, otherwise null
**          is returned
**
*****/
function getAPI()
{
```

```
window = this;
var theAPI = findAPI( window );

if ( (theAPI == null) &&
    (window.opener != null) &&
    (typeof(window.opener) != "undefined") )
{
    theAPI = findAPI( window.opener );
}

if (theAPI == null)
{
    alert( "Unable to locate the LMS's API Implementation.\n" +
        "Communication with the LMS will not occur." );

    noAPIFound = "true";
}

return theAPI
}

/*****
**
** Returns the handle to API object if it was previously set, otherwise it
** returns null
**
** Inputs: None
**
** Return: Object - The value contained by the apiHandle variable.
**
*****/
```

```
**
*****/

function getAPIHandle()
{
    if ( apiHandle == null )
    {
        if ( noAPIFound == "false" )
        {
            apiHandle = getAPI();
        }
    }
}

    return apiHandle;
}

/*****
**
** This function is used to tell the LMS to initiate the communication session.
**
** Inputs: None
**
** Return: String - "true" if the initialization was successful, or
**         "false" if the initialization failed.
**
*****/

function initializeCommunication()
{
    var api = getAPIHandle();
```

```
if ( api == null )
{
    return "false";
}
else
{
    var result = api.Initialize("");

    if ( result != "true" )
    {
        var errCode = retrieveLastErrorCode();

        displayErrorInfo( errCode );

        // may want to do some error handling
    }
}

return result;
}

/*****
**
** This function is used to tell the LMS to terminate the communication session
**
** Inputs: None
**
** Return: String - "true" if successful or
**          "false" if failed.
**
*****/
```



```
**
*****/
function terminateCommunication()
{
    var api = getAPIHandle();

    if ( api == null )
    {
        return "false";
    }
    else
    {
        // call Terminate only if it was not previously called
        if ( terminated != "true" )
        {
            // call the Terminate function that should be implemented by
            // the API
            var result = api.Terminate("");

            if ( result != "true" )
            {
                var errCode = retrieveLastErrorCode();

                displayErrorInfo( errCode );

                // may want to do some error handling
            }
            else // terminate was successful
            {
```

```
        terminated = "true";
    }
}

return result;
}

/*****
**
** This function requests information from the LMS.
**
** Inputs: String - Name of the data model defined category or element
**          (e.g. cmi.core.learner_id)
**
** Return: String - The value presently assigned to the specified data model
**           element.
**
*****/
function retrieveDataValue( name )
{
    // do not call a set after finish was called
    if ( terminated != "true" )
    {
        var api = getAPIHandle();

        if ( api == null )
        {
            return "";
        }
    }
}
```

```
}  
else  
{  
    var value = api.GetValue( name );  
  
    var errCode = api.GetLastError();  
  
    if ( errCode != "0" )  
    {  
        var errCode = retrieveLastErrorCode();  
  
        displayErrorInfo( errCode );  
    }  
    else  
    {  
        return value;  
    }  
}  
}  
  
return;  
}  
  
/*****  
**  
** This function is used to tell the LMS to assign the value to the named data  
** model element.  
**  
** Inputs: String - Name of the data model defined category or element value
```

```
**  
**      String - The value that the named element or category will be  
**      assigned  
**  
** Return: String - "true" if successful or  
**      "false" if failed.  
**  
*****/  
function storeDataValue( name, value )  
{  
    // do not call a set after finish was called  
    if ( terminated != "true" )  
    {  
        var api = getAPIHandle();  
  
        if ( api == null )  
        {  
            return;  
        }  
        else  
        {  
            var result = api.SetValue( name, value );  
  
            if ( result != "true" )  
            {  
                var errCode = retrieveLastErrorCode();  
  
                displayErrorInfo( errCode );  
            }  
        }  
    }  
}
```

```
        // may want to do some error handling
    }
}

return;
}

/*****
**
** This function requests the error code for the current error state from the
** LMS.
**
** Inputs: None
**
** Return: String - The last error code.
**
*****/
function retrieveLastErrorCode()
{
    // It is permitted to call GetLastError() after Terminate()

    var api = getAPIHandle();

    if ( api == null )
    {
        return "";
    }
    else
```

```
{
    return api.GetLastError();
}
}

/*****
**
** This function requests a textual description of the current error state from
** the LMS
**
** Inputs: String - The error code.
**
** Return: String - Textual description of the given error state.
**
*****/
function retrieveErrorInfo( errCode )
{
    // It is permitted to call GetLastError() after Terminate()

    var api = getAPIHandle();

    if ( api == null )
    {
        return "";
    }
    else
    {

        return api.GetErrorString( errCode );
    }
}
```

```
}  
}  
  
/*****  
**  
** This function requests additional diagnostic information about the given  
** error code. This information is LMS specific, but can help a developer find  
** errors in the SCO.  
**  
** Inputs: String - The error code.  
**  
** Return: String - Additional diagnostic information about the given error  
**         code  
**  
*****/  
function retrieveDiagnosticInfo( error )  
{  
    // It is permitted to call GetLastError() after Terminate()  
  
    var api = getAPIHandle();  
  
    if ( api == null )  
    {  
        return "";  
    }  
    else  
    {  
        return api.GetDiagnostic( error );  
    }  
}
```

```
}

/*****
**
** This function requests that the LMS persist all data to this point in the
** session.
**
** Inputs: None
**
** Return: None
**
*****/

function persistData()
{
    // do not call a set after Terminate() was called
    if ( terminated != "true" )
    {
        var api = getAPIHandle();

        if ( api == null )
        {
            return "";
        }
        else
        {
            return api.Commit();
        }
    }
    else
```



```
{
    return "";
}
}

/*****
**
** Display the last error code, error description and diagnostic information.
**
** Inputs: String - The error code
**
** Return: None
**
*****/

function displayErrorInfo( errCode )
{
    if ( _debug )
    {
        var errString = retrieveErrorInfo( errCode );
        var errDiagnostic = retrieveDiagnosticInfo( errCode );

        alert( "ERROR: " + errCode + " - " + errString + "\n" +
            "DIAGNOSTIC: " + errDiagnostic );
    }
}

// JavaScript Document

/*****
**
```

```
** Course functions for the Advanced Distributed Learning
** Sequencing Essentials Content Example (SECE).
** Sharable Content Object Reference Model (SCORM®) 2004 Sequencing.
**
*****/

// Global variables

// Keep track of location in content by div number
var CurrentPage;
// Track initialized status so it is only called once
var AlreadyInitialized = false;

/*****
**
** This function asks the LMS if there exists a previous SCO or Asset to go to.
** If a SCO or Asset exists, then the previous button is displayed.
**
** Inputs: None
**
** Return: String - "true" if the previous button should be displayed
**         "false" if failed.
**
*****/
function RenderPreviousButton() {
    var value = retrieveDataValue("adl.nav.request_valid.previous");
    return value;
}
```

```

/*****
**
** This function asks the LMS if there exists a next SCO or Asset to continue
** to. If a SCO or Asset exists, then the continue button is displayed.
**
** Inputs: None
**
** Return: String - "true" if the continue button should be displayed
**          "false" if failed.
**
*****/
function RenderContinueButton() {
    var value = retrieveDataValue("adl.nav.request_valid.continue");
    return value;
}

/*****
**
** This function is used to go to a previous SCO
**
*****/
function PreviousSCO() {
    // we request the previous SCO from the LMS
    storeDataValue( "adl.nav.request", "previous" );
    // we terminate this SCO's communication with the LMS
    terminateCommunication();
}

/*****

```

```
**  
** This function is used to go to a next SCO  
**  
*****/  
function ContinueSCO() {  
    // we request the previous SCO from the LMS  
    storeDataValue( "adl.nav.request", "continue" );  
    // we terminate this SCO's communication with the LMS  
    terminateCommunication();  
}  
  
/*****  
**  
** This function is used to tell the LMS to initiate the communication session  
** using the APIWrapper.js file as a pass through. It uses the global variable  
** AlreadyInitialized so initialize is only called once.  
**  
** Inputs: None  
**  
** Return: None  
**  
*****/  
function CallInitialize(){  
    if(!AlreadyInitialized){  
        initializeCommunication();  
        AlreadyInitialized = true;  
    }  
}
```

```
/******  
**  
** This function sets the state of the sco.  
**  
** Inputs: None  
**  
** Return: None  
**  
*****/  
  
function Initialize() {  
    // make initialize call  
    CallInitialize();  
  
    // set completion status to incomplete  
    SetIncomplete();  
  
    // set exit to suspended  
    storeDataValue( "cmi.exit","suspend" );  
  
    // check for resumed entry state  
    var entryMode = retrieveDataValue( "cmi.entry" );  
  
    // set a local variable to page 1  
    var location = 1;  
  
    // check whether resuming SCO  
    if (entryMode == "resume") {  
        // check if a prior location was set  
        location = retrieveDataValue( "cmi.location" );  
    }  
}
```

```
// get the Error code from the last call
var errorCode = retrieveLastErrorCode();

// if not set or at the last page, go to first page
if (errorCode == "403" || location == TotalPages()) {
    location = 1;
}
}
// present page to learner
DisplayPage( location );
}

/*****
**
** This function is used to get the total number of pages (divs with the class
** "page") in the sco
**
** Inputs: None
**
** Return: String - total number of divs with the class name "page"
**
*****/
function TotalPages(){
    // initial setup of variables
    var pages = 0
    var divs = document.getElementsByTagName("div");

    for ( var i = 0; i < divs.length; i++ )    {
```

```
        if (divs[i].className == "page"){
            pages++;
        }
    }
    return pages;
}

/*****
**
** Makes the appropriate calls for a normal exit calling Terminate
**
** Inputs: None
**
** Return: None
**
*****/

function Terminate() {
    terminateCommunication();
}

/*****
**
** Sets the SCO completion status to incomplete.
**
** Inputs: None
**
** Return: None
*****/
```

```
**
*****/
function SetIncomplete (){
    retrieveDataValue( "cmi.completion_status" );
    if (status != "completed"){
        storeDataValue( "cmi.completion_status", "incomplete" );
    }
}

/*****
**
** Sets the SCO completion status to complete.
**
** Inputs: None
**
** Return: None
**
*****/
function SetComplete (){
    storeDataValue( "cmi.completion_status", "completed" );
}

/*****
**
** Shows and hides divs to create the appearance of paging through a sco.
**
** Inputs: Int (or a String formatted as an Int) - div to display
**
** Return: None
```



```
**
*****/
function DisplayPage( pn )
{
    pageNumber = parseInt(pn);

    // catch out of range pages

    if (pageNumber <1 || pageNumber > TotalPages()){
        pageNumber = 1;
    }

    //check and stop flash

    if (SwfLoaded(document["swf" + CurrentPage ])) {
        document["swf" + CurrentPage ].GotoFrame(1);
    }

    // set location value for bookmark
    storeDataValue( "cmi.location", pageNumber ) ;

    var divs = document.getElementsByTagName("div");

    for ( var i = 0; i < divs.length; i++ )    {
        var div = divs[i];
        var id = div.id;
        var className = div.className;
```

```
        if ( className == "page" ){
            if ( id == "p" + pageNumber ){
                // show requested page
                div.style.visibility = "visible";
            }
            else {
                // hide other pages
                div.style.visibility = "hidden";
            }
        }
    }

    // set completion status to completed when the user hits the last page
    // check whether to display continue button at end of sco for navigation to
next sco
    if ( pageNumber == TotalPages() ) {

        SetComplete();

        if ( RenderContinueButton() != "true" ) {
            document.getElementById("nextBtn").style.visibility = "hidden";
        }
    }
    else{
        document.getElementById("nextBtn").style.visibility = "visible";
    }
}
```

```
// check whether to display previous button at beginning of sco for
navigation to previous sco
if ( pageNumber == 1 ) {
    if ( RenderPreviousButton() != "true" ) {
        document.getElementById("previousBtn").style.visibility =
"hidden";
    }
}
else{
    document.getElementById("previousBtn").style.visibility = "visible";
}

// check and start flash
if (SwfLoaded(document["swf" + pageNumber])) {
    document["swf" + pageNumber].Play();
}

// set global page
CurrentPage = pageNumber;

}
```

```
/******
**
** Navigation button continue function. Handles page to page and sco to sco
navigation.
** (Buttons are hidden when sco to sco navigation is not allowed.)
**
```

** Inputs: None

**

** Return: None

**

*****/

```
function NextPage() {
    if (CurrentPage+1 <= TotalPages()){
        DisplayPage( CurrentPage + 1 )
    }
    else{
        ContinueSCO();
    }
}
```

/******

**

** Navigation button previous function. Handles page to page and sco to sco navigation.

** (Buttons are hidden when sco to sco navigation is not allowed.)

**

** Inputs: None

**

** Return: None

**

*****/

```
function PreviousPage() {
    if (CurrentPage-1 >= 1){
        DisplayPage( CurrentPage - 1 )
    }
}
```

```
        else{
            PreviousSCO();
        }
    }

/*****
**
** Checks if Flash object is finished loading.
**
** Inputs: Object - DOM Reference
**
** Return: Boolean
**
*****/

function SwfLoaded (swfRef) {
    if (typeof(swfRef) != "undefined") {
        return swfRef.PercentLoaded() == 100;
    } else {
        return false;
    }
}

/*****
**
** Hides and shows a div on a page.
** Used in the Tracking Model SCO for additional information about keywords.
**
** Inputs: Object - DOM Reference
**
```

```
** Return: Void
**
*****/
function SwitchMenu(divId)
{
var element = document.getElementById(divId);
    if(element.style.display != "block"){
        element.style.display = "block";
    }
    else{
        element.style.display = "none";
    }
}
```

Anexo 2: Código Java Script que se posee la implementación de la comunicación con el API de SCORM.

```
/******
**
** FileName: APIWrapper.js
**
*****/

/******
**
** Advanced Distributed Learning Co-Laboratory (ADL Co-Lab) Hub grants you
** ("Licensee") a non-exclusive, royalty free, license to use, modify and
** redistribute this software in source and binary code form, provided that
** i) this copyright notice and license appear on all copies of the software;
```

** and ii) Licensee does not utilize the software in a manner which is
** disparaging to ADL Co-Lab Hub.

**

** This software is provided "AS IS," without a warranty of any kind. ALL
** EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND
WARRANTIES, INCLUDING
** ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A
PARTICULAR PURPOSE
** OR NON-INFRINGEMENT, ARE HEREBY EXCLUDED. ADL Co-Lab Hub AND
ITS LICENSORS
** SHALL NOT BE LIABLE FOR ANY DAMAGES SUFFERED BY LICENSEE AS
A RESULT OF
** USING, MODIFYING OR DISTRIBUTING THE SOFTWARE OR ITS
DERIVATIVES. IN NO
** EVENT WILL ADL Co-Lab Hub OR ITS LICENSORS BE LIABLE FOR ANY
LOST REVENUE,
** PROFIT OR DATA, OR FOR DIRECT, INDIRECT, SPECIAL,
CONSEQUENTIAL,
** INCIDENTAL OR PUNITIVE DAMAGES, HOWEVER CAUSED AND
REGARDLESS OF THE
** THEORY OF LIABILITY, ARISING OUT OF THE USE OF OR INABILITY TO
USE
** SOFTWARE, EVEN IF ADL Co-Lab Hub HAS BEEN ADVISED OF THE
POSSIBILITY OF SUCH
** DAMAGES.

**

*****/

/*****

```
** This file is part of the ADL Sample API Implementation intended to provide  
** an elementary example of the concepts presented in the ADL Sharable  
** Content Object Reference Model (SCORM).
```

```
**
```

```
** The file is used by the run time environment to maintain internal communication  
** within the frames of the environment.
```

```
**
```

```
*****/
```

```
var _Debug = false; // set this to false to turn debugging off  
                    // and get rid of those annoying alert boxes.
```

```
// local variable definitions
```

```
var apiHandle = null;  
var API = null;  
var findAPITries = 0;
```

```
/******/
```

```
**
```

```
** Function: dolnit()
```

```
** Inputs: none
```

```
** Return: false if the API handle cannot be located
```

```
**
```

```
** Description: gets the API handle and sets up necessary api variable values
```

```
**
```

```
**
```

```
*****/
```



```
function initAPI()
{
    var api = getAPIHandle();

    if (api == null)
    {
        alert("No se puede localizar la implementacion del RTE del API");
        return "false";
    }
    else
    {
        api.setActivityID( document.getElementById("activityID").value );
        api.setCourseID( document.getElementById("courseID").value );
        api.setStateID( document.getElementById("stateID").value );
        api.setUserID( document.getElementById("userID").value );
        api.setUsername( document.getElementById("userName").value );
        api.setNumAttempts( document.getElementById("numAttempts").value );
        api.clearState();
    }
}

/*****
**
** Function getAPIHandle()
** Inputs: None
** Return: value contained by APIHandle
**
** Description:
```

```
** Returns the handle to API object if it was previously set,  
** otherwise it returns null  
**  
*****/  
function getAPIHandle()  
{  
    if (apiHandle == null)  
    {  
        apiHandle = getAPI();  
    }  
    return apiHandle;  
}  
  
/*****  
**  
** Function findAPI(win)  
** Inputs: win - a Window Object  
** Return: If an API object is found, it's returned, otherwise null is returned  
**  
** Description:  
** This function looks for an object named API in parent and opener windows  
**  
*****/  
function findAPI(win)  
{  
    while ((win.API_1484_11 == null) && (win.parent != null) && (win.parent != win))  
    {  
        findAPITries++;  
    }  
}
```

```
// Note: 500 is a number based on the IEEE API Standards.
if ( findAPITries > 500 )
{
    alert("Error encontrando el API -- Muy anidado.");
    return null;
}

win = win.parent;

}

return win.API_1484_11;

}

/*****
**
** Function getAPI()
** Inputs: none
** Return: If an API object is found, it's returned, otherwise null is returned
**
** Description:
** This function looks for an object named API, first in the current window's
** frame hierarchy and then, if necessary, in the current window's opener window
** hierarchy (if there is an opener window).
**
*****/
```

```
function getAPI()
{
    window = this;
    var theAPI = findAPI(window);

    if ((theAPI == null) && (window.opener != null) && (typeof(window.opener) !=
"undefined"))
    {
        theAPI = findAPI(window.opener);
    }
    if (theAPI == null)
    {
        alert("RTE - No se puede localizar el adaptador del API");
    }
    return theAPI
}
```