

Universidad de las Ciencias Informáticas

Facultad 4



Análisis y Diseño de un visor de paquetes SCORM para la Plataforma Educativa Zera

Trabajo de diploma para optar por el título de Ingeniero en Ciencias Informáticas

Autor

Ramón Verdecia Espinosa

Tutores

Ing. Irina Ivis Santiesteban Pérez

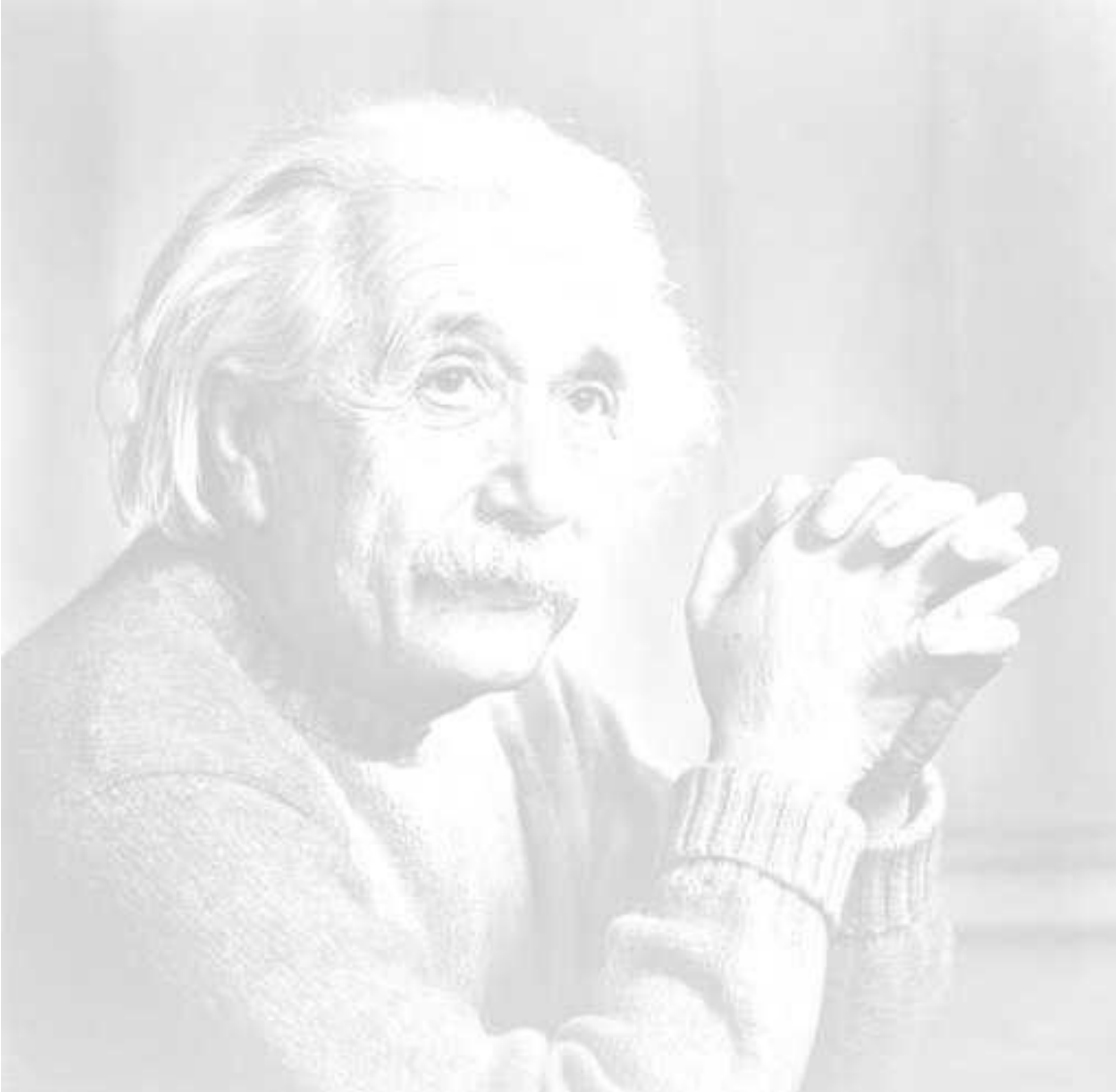
Ing. Yerandy Manso Guerra

Cotutora

Ing. Yenisleidy Piloto Lastra

La Habana, junio de 2012

“Año 54 de la Revolución”



“Nunca consideres el estudio como una obligación sino como una oportunidad para penetrar en el bello y maravilloso mundo del saber”

Albert Einstein

Declaración de autoría

Declaro que soy el único autor del trabajo “Análisis y Diseño de un visor de paquetes SCORM para la Plataforma Educativa Zera” y autorizo a la Facultad 4 de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

Firma del Autor

Ramón Verdecia Espinosa

Firma del Tutor

Ing. Irina Ivis Santiesteban Pérez

Firma del Tutor

Ing. Yerandy Manso Guerra

Firma del Cotutor

Ing. Yenisleidy Piloto Lastra

Agradecimientos

A mi mamá por todo su amor, comprensión y cariño durante toda mi vida, por confiar siempre en mí.

A mi papá por apoyarme durante todos estos años.

A mis abuelos, hermanos, tíos y primos por todo su cariño y ayuda brindada.

A mis tutores Irina y Yerandy por todo el apoyo y ayuda brindada, sin ellos este trabajo no hubiera sido posible. A pesar de su poca experiencia me supieron guiar de forma inigualable, les doy mil gracias.

A mis compañeros de grupo Yisel, Abdel, Leticia, José Antonio, Dunet, Leo, Dayanis, Alejandro, Gisela, Argel, Coralí y Orestes, los mejores que tuve.

A Dorys y Aleyda por apoyarme en los buenos y malos momentos, por brindarme su amistad.

A Lilibett y Yanet mis mejores compañeras de estudio.

A los profesores que me ayudaron siempre que los necesite, Piña, Yoseti y Thaymi.

A Yuleisy por siempre estar dispuesta ayudarme y brindarme apoyo.

A Yusdel, la primera persona que conocí de la facultad, por revisarme la tesis.

A mis amistades del apto Ernesto, Geiber, Manuel y Leonardo.

A todos los que de una forma u otra colaboraron y revisaron la tesis.

Dedicatoria

A mi mamá por todo su sacrificio y ser la razón de mi vida.

A mis abuelos por darme siempre sabios consejos y apoyarme en todo momento.

A mis hermanos, especialmente a Raisa por estar siempre en los momentos más difíciles.

A mi papá y el resto de la familia por siempre estar pendiente de mí y darme todo su apoyo.

A mi tío Osmanis por toda su ayuda durante mis estudios.

A todos los que de una forma u otra han contribuido con mi formación como profesional y como persona.

Resumen

En las plataformas educativas existentes a nivel internacional, se integran diversas herramientas y estándares como es el caso de SCORM, el cual es un conjunto de especificaciones usadas para la creación, distribución y empaquetamiento de contenido. Para visualizar el contenido de los paquetes las plataformas utilizan visores SCORM. La Plataforma Educativa Zera tiene entre sus características la importación y exportación de paquetes SCORM, sin embargo no posee un visor, por tanto la siguiente investigación se hace necesaria con el objetivo de desarrollar el análisis y diseño de un visor de paquetes SCORM que cumpla con lo establecido por el estándar SCORM 2004. Durante la investigación se estudiaron los visores que son utilizados por plataformas a nivel mundial, se analizaron sus características, ventajas y desventajas, lo que permitió obtener una mejor comprensión del funcionamiento del negocio. A partir del análisis anterior, se tomaron una serie de características para ser incorporadas a la propuesta de solución. Para el desarrollo de la investigación se utilizaron las tecnologías definidas por el proyecto Alfaomega. Además se confeccionaron los artefactos que propone la metodología RUP en sus primeros tres flujos de trabajo, validando dos de ellos. Los métodos investigativos empleados fueron la modelación, el analítico-sintético y el histórico-lógico.

Palabras claves: curso SCORM, navegación, visor

Índice

Introducción.....	1
Capítulo I: Fundamentación Teórica	5
1.1 Conceptos y definiciones asociados al problema de investigación.....	5
1.1.1 E-learning	5
1.1.2 Sistema de Gestión de Aprendizaje	6
1.1.3 Estándar.....	6
1.1.4 Especificación	7
1.1.5 Visor	7
1.2 Características del estándar SCORM	8
1.2.1 Evolución del estándar	8
1.2.2 Entorno de ejecución de SCORM.....	9
1.3 Visores de paquetes SCORM usados por plataformas educativas.....	14
1.3.1 Sakai	15
1.3.2 Moodle	16
1.3.3 Blackboard.....	17
1.3.4 Dokeos.....	19
1.3.5 Tabla resumen de los visores estudiados	20
1.3.6 Conclusiones del estudio de los visores SCORM.....	21
1.4 Metodología, lenguaje y herramientas	21
1.4.1 Metodología.....	22
1.4.2 Lenguaje de modelado	24
1.4.3 Herramienta de modelado	25
1.4.4 Herramienta para el diseño de prototipos de interfaz de usuario.....	25
1.4.5 Conclusiones de la metodología, el lenguaje y las herramientas.....	26
1.5 Conclusiones.....	26

Capítulo II: Características del Sistema.....	28
2.1 Modelo de dominio	28
2.1.2 Diagrama del Modelo de Dominio	28
2.2 Descripción del sistema propuesto.....	30
2.3 Requerimientos del software	30
2.3.1 Requerimientos funcionales	30
2.3.2 Requerimientos no funcionales.....	31
2.4 Modelo de Casos de Uso del Sistema	32
2.4.1 Patrones de casos de uso	33
2.4.2 Diagrama de Actores	33
2.4.4 Diagrama de Casos de Uso del Sistema.....	35
2.4.5 Descripción de los casos de uso del sistema	35
2.5 Conclusiones.....	41
Capítulo III: Análisis y Diseño.....	42
3.1 Modelo de Análisis	42
3.1.1 Clases del análisis	42
3.1.2 Diagramas de Clases del Análisis	43
3.1.3 Diagramas de Colaboración	44
3.2 Paquetes del visor	45
3.3 Modelo de Diseño.....	45
3.3.1 Arquitectura.....	45
3.3.2 Diagramas de Clases del Diseño	50
3.3.3 Diagramas de Secuencia del Diseño	51
3.4 Diseño de la base de datos	52
3.4.1 Descripción de las tablas del Modelo de Datos	52
3.5 Conclusiones.....	54

Capítulo 4 Validación de la Solución Propuesta	55
4.1 Métricas de la Calidad de Especificación de los Requisitos.....	55
4.2 Validación mediante prototipos de interfaz de usuario.....	58
4.4 Validación de la propuesta por el método de expertos	60
4.5 Conclusiones.....	63
Conclusiones.....	64
Recomendaciones	65
Referencias bibliográficas	66
Glosario de Términos	71

Introducción

Las Tecnologías de la Información y las Comunicaciones, en lo adelante TICs, han impactado todas las esferas de la sociedad, revolucionando la forma de vivir y pensar de las personas. En una de las áreas donde ha influido con mayor peso ha sido en la educación, donde el proceso de enseñanza-aprendizaje le abrió las puertas a una nueva modalidad, el “e-learning¹”. El e-learning es una estrategia educativa que rompe con las barreras geográficas y temporales, además facilita la comunicación entre el docente y el alumno. Para facilitar el proceso de esta nueva forma de educar se emplean herramientas que se conocen como Sistemas de Gestión de Aprendizaje, en inglés Learning Management Systems, en lo adelante LMS.

Los LMS permiten gestionar materiales educativos, cursos y estudiantes, realizar evaluaciones, administrar servicios de comunicación, así como controlar el proceso de aprendizaje. Estos beneficios pueden ser mejorados con el uso de estándares como el Modelo de Referencia de Contenidos Compartibles, en inglés Sharable Content Object Reference Model, en lo adelante SCORM, debido a que este permite la reutilización de materiales formativos, son adaptables a las necesidades de uso, son interoperables entre diferentes plataformas y accesibles.

Cuba se encuentra en un proceso de informatización de la sociedad y como parte de ello la esfera educacional ha tenido un avance significativo, que va desde la aparición de los laboratorios de computación en cada una de las escuelas, hasta el uso de plataformas educativas. Las universidades explotan sustancialmente el avance tecnológico, un ejemplo de ello es la Universidad de las Ciencias Informáticas, en lo adelante UCI.

La UCI cuenta con un Centro de Tecnologías para la Formación, dentro de este se está desarrollando una plataforma educativa denominada Zera. La plataforma permite la gestión de hiperentornos de aprendizaje, los cuales constituyen medios de enseñanza que “le permiten al estudiante apropiarse del contenido de manera reflexiva y consciente”. (1) Además constituyen un soporte informático pleno para el proceso docente, para cada asignatura y grado a que va dirigido, cubriendo completamente los programas de estudio de cada asignatura en los diferentes niveles educacionales. En la plataforma se integran los principales conceptos de los hiperentornos, las mejores prácticas y elementos arquitectónicos de soluciones similares y las principales especificaciones y estándares educativos desarrollados y utilizados a nivel mundial en plataformas de aprendizaje. Entre las funcionalidades que presenta la misma están la

¹ Educación electrónica

exportación e importación de recursos educativos cumpliendo con lo establecido en el estándar SCORM 2004 3ra Edición, incorporadas a través de una tesis² de pregrado en el curso 2010-2011. Sin embargo, a pesar de lo mencionado, los paquetes SCORM solo se usan para importar recursos a la plataforma provocando que se pierda la intención pedagógica con la que fueron creados los paquetes y el objetivo principal del estándar, importar y exportar cursos, para que puedan ser reutilizados por diferentes plataformas. Además, una vez que se cuenta con el paquete en la plataforma, no hay modo de navegar por el contenido del curso que este posee. Tampoco existe la posibilidad de que se comunique el paquete o curso con la plataforma, lo que provoca que las actividades que se realicen no queden registradas en la plataforma.

Teniendo en cuenta lo expuesto anteriormente, se identificó como **problema a resolver**: ¿Cómo facilitar la implementación de un módulo que permita la visualización y navegación de los cursos que poseen los paquetes SCORM importados a la Plataforma Educativa Zera, así como la comunicación entre el curso y la plataforma?

El **objetivo general** de la investigación es: desarrollar el análisis y diseño de un visor de paquetes SCORM para la Plataforma Educativa Zera que cumpla con lo establecido por el estándar SCORM 2004.

Para dar cumplimiento al objetivo general se plantean los **objetivos específicos** siguientes:

- ✓ Caracterizar el estado del arte referente a los visores de paquetes SCORM.
- ✓ Definir los requerimientos funcionales y no funcionales del visor de paquetes SCORM.
- ✓ Diseñar el visor de paquetes SCORM.
- ✓ Validar la propuesta de solución.

Resultados esperados:

Una vez finalizado el presente trabajo, se tendrá como resultado el análisis y diseño del visor de paquetes SCORM, que ayudará en gran medida en su futura implementación. Además de una amplia documentación generada a partir del desarrollo de esta investigación.

El **objeto de estudio** son los procesos de visualización de contenidos de paquetes SCORM.

² Se puede encontrar en la siguiente dirección,
http://repositorio_institucional.uci.cu/jspui/bitstream/ident/TD_04642_11/1/TD_04642_11.pdf

La investigación se enmarca en el **campo de acción** de la visualización de los contenidos de los paquetes SCORM en plataformas educativas.

Como **idea a defender** se plantea que con el desarrollo del análisis y diseño de un visor de paquetes SCORM 2004, se facilitará la implementación de un módulo para la visualización y navegación de los cursos SCORM en la Plataforma Educativa Zera, así como la comunicación entre el curso y la plataforma.

Durante la investigación se utilizaron los siguientes **métodos teóricos**: el **Histórico-Lógico** para estudiar la evolución del estándar SCORM y observar cómo están siendo usados los visores de paquetes SCORM en la actualidad. Además en el estudio de las tecnologías que se emplearán para el desarrollo de la propuesta. Otro de los métodos empleado es el **Analítico-Sintético**, el cual se empleó en el análisis de la documentación, analizando las características, forma de uso y utilidad de los diferentes visores de paquetes SCORM. Todo esto permitió comprender mejor el funcionamiento del negocio y realizar una buena captura de requisitos. También se utilizó la **Modelación** para la representación de las características estructurales y de comportamiento del visor de paquetes SCORM, correspondientes a la etapa de análisis y diseño de RUP.

Estructura capitular

Capítulo I: Fundamentación Teórica. Se presenta un estudio del estado del arte. Se realiza un estudio sobre la visualización de paquetes SCORM en aplicaciones de escritorio, repositorios de objetos de aprendizaje y plataformas educativas. También se enuncian algunas de las características que presentan las tecnologías a emplear en el desarrollo de la investigación, que fueron definidas por el proyecto Alfaomega.

Capítulo II: Característica del Sistema. Se realiza el levantamiento de requisitos, se estructura el diagrama de casos de uso del sistema y se describen cada uno de ellos.

Capítulo III: Análisis y Diseño. Se realiza el análisis y diseño del sistema, compuesto por los diagramas de clases del análisis, colaboración del análisis, clases del diseño y secuencia del diseño. Además se define la arquitectura y el modelo de datos. Se sientan las bases para la futura implementación.

Capítulo IV: Validación de la Solución Propuesta. Se realiza la validación de los requisitos y los casos de uso. También se describe la validación de la propuesta por el método de expertos y se plasman los resultados obtenidos.

Capítulo I: Fundamentación Teórica

En el capítulo se presentará un estudio del estado del arte. Se analizarán una serie de conceptos que ayudarán a comprender el problema de investigación. También se realizará un estudio sobre los procesos de visualización de paquetes SCORM, profundizando en las soluciones similares usadas a nivel mundial. Además se estudiarán las diferentes tecnologías que se emplearán para el desarrollo del tema de investigación.

1.1 Conceptos y definiciones asociados al problema de investigación

Como parte del desarrollo de la investigación, fue necesario identificar los siguientes conceptos y definiciones que ayudarán en la familiarización con el tema.

1.1.1 E-learning

A medida que crece la educación asistida por ordenadores, son más las personas que definen esta nueva modalidad de enseñanza. A continuación se muestran algunos de los conceptos que han definido diferentes autores:

“Una modalidad formativa que permite una formación completamente a distancia o semipresencial que integra el uso de las TICs y otros elementos didácticos para la docencia, donde los estudiantes acceden a los contenidos, actividades, recursos y tutores del curso a través de las plataformas tecnológicas, que le permiten interactuar con los participantes del proceso sin compartir el mismo espacio físico”. (2)

“Enseñanza a distancia caracterizada por una separación física entre profesorado y alumnado sin excluir encuentros físicos puntuales, entre los que predomina una comunicación de doble vía, asíncrona donde se usa preferentemente Internet como medio de comunicación y de distribución del conocimiento, de tal manera que el estudiante es el centro de una formación independiente y flexible, al tener que gestionar su propio aprendizaje generalmente con ayuda de tutores externos”. (3)

Estas definiciones concuerdan en que se trata de una educación a distancia, donde pueden existir encuentros de forma presencial, en las dos se dice que se utilizan las tecnologías de la información y las comunicaciones para el aprendizaje especialmente Internet y que permiten la comunicación entre estudiantes y docentes para el aprendizaje.

El e-learning comprende fundamentalmente los siguientes aspectos: (4)

- ✓ El pedagógico, referido a la Tecnología Educativa como disciplina de las ciencias de la educación, vinculada a los medios tecnológicos, la psicología educativa y la didáctica.
- ✓ El tecnológico, referido a las TICs, mediante la selección, diseño, personalización, implementación, alojamiento y mantenimiento de soluciones en donde se integran tecnologías propietarias y de código abierto.

1.1.2 Sistema de Gestión de Aprendizaje

Los Sistemas de Gestión de Aprendizaje son definidos por diferentes autores como:

“...herramienta informática, habitualmente de gran tamaño, que permite la gestión y presentación de materiales educativos a estudiantes. El objetivo de esta herramienta es permitir el aprendizaje en cualquier parte y en cualquier momento. La mayoría de estas herramientas son herramientas web, es decir, herramientas que se usan a través de Internet utilizando un navegador web”. (5)

“...programa que permite organizar materiales y actividades de formación en cursos, gestionar la matrícula de los estudiantes, hacer seguimiento de su proceso de aprendizaje, evaluarlos, comunicarse con ellos mediante foros de discusión, chat o correo electrónico. Permite hacer todas aquellas funciones necesarias para gestionar cursos de formación a distancia (aunque pueden usarse como complemento en la enseñanza presencial)”. (6)

Después de analizar estas definiciones, se puede decir que los LMS son una herramienta que permiten la gestión de cursos, materiales educativos y estudiantes, que ofrecen varios servicios como los foros y el chat que hacen que el aprendizaje sea más colaborativo y se consiga de forma rápida.

1.1.3 Estándar

Diversas instituciones encargadas de la estandarización, han definido el término estándar como se muestra a continuación.

“...acuerdo documentado que contiene especificaciones técnicas u otros criterios, para ser utilizados constantemente como reglas, lineamientos o definiciones de características, para asegurar que materiales, productos, procesos y servicios son adecuados para sus propósitos.”
(7)

Según la Entidad de Normalización Británica, en inglés British Standard Institute (BSI), un estándar “es una especificación publicada que establece un lenguaje común y contiene una especificación técnica diseñada para ser usada constantemente como una regla, una definición.”

De forma general se puede concluir que un estándar es un documento que contiene una serie de normas o reglas para un fin común.

1.1.4 Especificación

Muchas veces se confunden los términos estándar y especificación, debido a que son documentos técnicos con reglas y normas, pero no son lo mismo. “Una especificación es una descripción documentada. Algunas llegan a ser un estándar, es decir, reciben alguna acreditación, pero las crean comités no acreditados, como el Grupo Especial sobre Ingeniería de Internet, en inglés Internet Engineering Task Force (IETF) y el Grupo de Gestión de Objetos, en inglés Object Management Group (OMG)”. (8)

1.1.5 Visor

El término visor se puede encontrar en diferentes áreas como la fotografía, las armas de fuego y la informática. Dentro de la informática un visor es una aplicación utilizada para visualizar algún archivo o recurso, como por ejemplo:

- ✓ **Adobe Acrobat Reader:** permite leer, navegar e imprimir los ficheros de documentos en formato PDF. (9)
- ✓ **VLC media player v1.2.0:** reproduce videos y sonidos, es compatible con extensiones como MPEG-1, MPEG-2, MPEG-4, DivX, XviD, H.264, MP3 y OGG. (10)
- ✓ **Visor de imágenes y fax de Windows:** ofrece opciones como la visión secuencial controlada, pulsando sobre anterior o siguiente, mostrando las imágenes que se encuentran en la misma carpeta. También permite ajustar la imagen a la pantalla, ampliar las zonas de la imagen a través de la lupa, girarla, borrarla y enviarla a la impresora. (11)

En los visores anteriores se puede apreciar que las acciones de visualizar el recurso y navegar a través de él son aspectos comunes. También que los visores pueden brindar otras opciones, que estarán en dependencia de las características del recurso.

El estudio de los conceptos expuestos anteriormente, servirá para lograr un mejor entendimiento de los elementos que se analizarán a continuación.

1.2 Características del estándar SCORM

SCORM es un conjunto de especificaciones usadas para desarrollar, empaquetar y distribuir contenido educativo en cualquier momento y en cualquier lugar.

Un paquete SCORM es un archivo comprimido en formato .zip, que contiene recursos como imágenes, videos, textos, animaciones, sonidos, objetos de evaluación y páginas web. En su raíz contiene un archivo llamado manifiesto, que se describe a continuación.

El manifiesto contiene la información necesaria para describir la estructura del paquete, dividido en cuatro secciones:

- ✓ **Metadatos:** describen los objetos educativos, la versión del estándar que se utiliza, los recursos y los objetivos didácticos.
- ✓ **Organizaciones:** forma en que están organizados los contenidos y su desglose en actividades. Las actividades se encuentran conectadas con los recursos a través de su identificador. Dentro de las organizaciones se encuentran las instrucciones de secuenciación y navegación.
- ✓ **Recursos:** describen los recursos externos y locales que utiliza el paquete. Los recursos locales se encontrarán en el fichero comprimido. Para que un recurso pueda comunicarse con una plataforma debe ser un Objeto de Contenido Intercambiable, en inglés Sharable Content Object, en lo adelante SCO.
- ✓ **SubManifiestos:** está compuesto por otros manifiestos anidados.

En el modelo SCORM un SCO, es la unidad mínima de contenido, es un objeto de aprendizaje que contiene código javascript, que le permite comunicarse con un LMS al ser ejecutado.

1.2.1 Evolución del estándar

SCORM 1.0 (enero de 2000): fue el inicio del estándar y contribuyó a formar lo que es hoy, se introdujo el concepto de SCO y el modelo de Interfaz de Programación de Aplicaciones, en inglés Application Program Interface, en lo adelante API. SCORM 1.0 no es relevante hoy en día y no existen implementaciones importantes.

SCORM 1.1 (enero de 2001): se considera la primera versión de SCORM, se emplea una estructura de curso. La estructura del contenido se describe en un archivo en formato XML, basado en las especificaciones del Comité de Capacitación por Computadora de la Industria de

la Aviación, en inglés Aviation Industry CBT Committee, en lo adelante AICC. Su empleo se vio limitado al no contar con un manifiesto robusto de empaquetado ni un soporte para metadatos.

SCORM 1.2 (Octubre de 2001): es la versión más universal y es soportado por un número mayor de herramientas, tanto plataformas como herramientas de autor.

SCORM 2004 (2004-2009): esta versión cuenta con 4 ediciones que a diferencia de la 1.2 permite: (12)

Secuenciación: secuencia de contenidos en función de determinadas condiciones, como puede ser el cumplimiento por el alumno de determinados objetivos formativos o el paso obligatorio por determinadas actividades. Esto permite desarrollar contenidos más dinámicos y no tan lineales. Esta posibilidad hace que tanto los contenidos como los LMS sean mucho más complejos.

Control sobre la navegación: dentro de una unidad (SCO) se pueden colocar elementos de interfaz de usuario como son botones y enlaces, que permitan la navegación a otras unidades del curso.

Compartir información entre las unidades (solo en SCORM 2004, 4ª edición): esta característica permite que el contenido traspase alguna información del usuario de una unidad SCO a otra, por ejemplo mostrar en una pantalla del último módulo de un curso, las distintas puntuaciones obtenidas por el usuario en los otros módulos del curso.

1.2.2 Entorno de ejecución de SCORM

Entre los objetivos de SCORM están que los recursos de aprendizaje sean reusables e interoperables entre múltiples LMS. Para que esto pueda ser posible, debe existir: (13)

1. Una forma común para iniciar recursos de aprendizaje (lanzamiento³). Este mecanismo define los procedimientos y las responsabilidades para el establecimiento de la comunicación entre el recurso de aprendizaje y el LMS. El protocolo de comunicación está estandarizado a través del uso de una API común.
2. Un mecanismo común para comunicarse con el LMS (API). La comunicación gira en torno al estado del recurso de aprendizaje (inicializado, terminado, condición de error) y además permite obtener y fijar datos entre el LMS y el SCO.

³ Mostrar el recurso en un navegador.

3. Un lenguaje predefinido (vocabulario) que forme las bases de la comunicación (Modelo de Datos, en inglés Data Model). En su forma más simple, el modelo de datos define elementos que tanto el LMS como el SCO están esperando conocer. El LMS debe mantener el estado de los elementos requeridos a través de sesiones, y el contenido de aprendizaje debe utilizar solo estos elementos predefinidos para garantizar la reusabilidad en diversos sistemas, que cumplan con la especificación de SCORM.

“El uso de una API común satisface muchos de los requerimientos de más alto nivel de SCORM para la reusabilidad y la interoperabilidad. Proporciona una forma estandarizada a los SCO para comunicarse con los LMS. El adaptador de la API es una pieza de software funcional, implementado por el LMS, que implementa las funciones de la API y por medio de interfaces las hace disponibles a los clientes SCO. Una vez que la comunicación entre el SCO y el LMS se establece, el SCO puede obtener y fijar información en el LMS. Toda la comunicación entre el adaptador de la API y el SCO es iniciada por el SCO.” (13)

Los métodos disponibles por el SCO para interactuar con el LMS están divididos en tres tipos como se muestra en la siguiente tabla: (14)

<p>Métodos de sesión. Inician o finalizan la sesión.</p>	<p>Initialize(“”)</p> <ul style="list-style-type: none"> ✓ Inicia la sesión de comunicación. ✓ Parámetros: la cadena vacía. ✓ Devuelve verdadero o falso, según la conexión con el LMS se haya realizado correctamente o no. En caso de error fija el código de error al valor correspondiente. <p>Terminate(“”)</p> <ul style="list-style-type: none"> ✓ Termina la sesión de comunicación. Al finalizar envía la información generada desde el SCO, desde el último envío o desde la inicialización de sesión. ✓ Como parámetro recibe la cadena vacía. ✓ Devuelve verdadero si el cierre de la conexión es correcto, o falso en el caso de que ocurra algún error. Si esto se produce fija el código de error al valor correspondiente.
--	--

<p>Métodos de transferencia de datos. Solo pueden llamarse mientras haya una sesión iniciada.</p>	<p>GetValue(dato)</p> <ul style="list-style-type: none">✓ Solicita por parte del SCO información al LMS. La información solicitada puede ser valores de datos del modelo, la versión del modelo que se utiliza y los elementos soportados.✓ Recibe como parámetro el identificador del dato que se quiere recoger.✓ Devuelve el valor solicitado o la cadena vacía en caso de error. En este último caso, la API se encarga de asignar el código de error apropiado. <p>SetValue(dato, valor)</p> <ul style="list-style-type: none">✓ Envía al LMS el <i>valor</i> pasado como parámetro al <i>dato</i> que se indica en el primer parámetro. El API puede enviar al servidor el dato inmediatamente o almacenarlo en una caché local.✓ Envía como parámetros el nombre del <i>dato</i> que se desea asignar y su <i>valor</i>. El valor se envía como texto, pero debe ser convertible al tipo definido para el dato.✓ Devuelve verdadero o falso según todo haya ido correcto o no. En caso de error el API asigna el código de error adecuado. <p>Commit(“”)</p> <ul style="list-style-type: none">✓ Envía al servidor cualquier dato que estuviera cacheado. Si no hubiera caché retorna verdadero y fija el código de error a cero, sin hacer más procesamiento.✓ Recibe la cadena vacía como parámetro.✓ Devuelve verdadero en caso de éxito o falso si ocurre algún error, en este caso asigna el código de error adecuado.
---	---

<p>Métodos de soporte. Aportan soporte de errores, incluso fuera de sesión.</p>	<p>GetLastError()</p> <ul style="list-style-type: none"> ✓ Solicita el código de error para el estado actual de la API. Este método no afecta el estado actual de la API al ser llamado. ✓ No acepta parámetros. ✓ Devuelve una cadena de texto convertible a un número entero en el rango de cero a 65536. <p>GetErrorString(error)</p> <ul style="list-style-type: none"> ✓ Obtiene una descripción textual de un error. ✓ Recibe como parámetro una cadena de texto con el código del error. ✓ Devuelve la descripción del error de hasta 255 caracteres o la cadena vacía si el código de error es desconocido. El estándar solo fija los códigos de error, los mensajes textuales son responsabilidades del LMS. <p>GetDiagnostic(parámetro)</p> <ul style="list-style-type: none"> ✓ Función de uso específico del LMS. Permite al LMS definir información para diagnóstico a través de la API. ✓ El parámetro puede ser un código de error, pero no tiene que serlo. Es una cadena de texto definida por el desarrollador de la API de hasta 255 caracteres. ✓ Devuelve la información de diagnóstico o la cadena vacía si el parámetro no se reconoce.
---	--

Tabla 1.1 Descripción de los métodos de la API

Modelo de datos del entorno de ejecución

Los elementos utilizados por el modelo que utiliza SCORM fueron creados por el AICC CMI, por ello es que los elementos comienzan con la palabra "cmi". Luego al unirle a SCORM la secuenciación y la navegación, fue necesario incorporarle nuevos valores al modelo, comenzando en este caso por la palabra "adl". Los elementos del modelo de datos se clasifican en dos grandes grupos, que son "obligatorios" y "opcionales". (13)

Todo LMS está obligado a soportar: (13)

- ✓ Todos los elementos del grupo obligatorio (id, nombre, ubicación de la lección y estado de la lección).
- ✓ Todos los métodos y elementos que indiquen el desempeño del estudiante como el tiempo acumulado total, indicadores sobre cómo o por qué se abandona un SCO y duración de la última sesión.
- ✓ Los datos de suspensión y lanzamiento de un SCO.

Las obligaciones del SCO son las siguientes:

- ✓ Localizar el API para iniciar la comunicación con la plataforma.
- ✓ De forma opcional, leer y escribir datos en la plataforma.
- ✓ Terminar la comunicación con la plataforma.

Después de analizar el estándar SCORM, se pudo constatar que ha estado en constante evolución en los últimos años y que se le han incorporado nuevos elementos, como es el caso de la secuenciación y la navegación en su última versión. También se vio como SCORM para su entorno de ejecución exige: una vía para mostrar los contenidos (un navegador), un API que permita la comunicación del curso con la plataforma y un modelo de datos que permita almacenar las interacciones del estudiante con el curso.

1.3 Visores de paquetes SCORM

Una vez que se ha creado un paquete SCORM en una herramienta de autor o en una plataforma educativa, se necesita una vía para acceder (ver) y navegar por su contenido, es aquí donde aparece el visor. Los visores de paquetes SCORM actualmente se pueden encontrar como aplicaciones de escritorio, o como módulos integrados a repositorios de objetos de aprendizaje o plataformas educativas. A continuación se muestran algunos de ellos.

Aplicaciones de escritorio

Vi-Sco es un visor de paquetes SCORM que muestra los objetos de aprendizaje que son descargados de Agrega en formato SCORM o que han sido creados en herramientas como eXeLearning. Este visor al ser una aplicación de escritorio, para emplearla solo hay que ejecutarla y cargar el archivo SCORM. Luego se genera un menú con toda la secuencia de contenidos que contiene el paquete SCORM, que se emplea para la navegación. (15)

Reload SCORM Player verifica que el archivo imsmanifest.xml esté correcto, simula la comunicación con una plataforma educativa y almacena los resultados hasta que se reinicien las pruebas. (16) Es una herramienta muy útil para probar contenidos SCORM. Permite

importar un paquete SCORM 1.2, ver el árbol de contenidos descrito en el manifiesto y ejecutar los SCO importados.

Visualización de paquetes SCORM en repositorios de objetos de aprendizaje

El repositorio de acceso público basado en SCORM (SBPAR)⁴, visualiza el contenido de los paquetes SCORM 1.2, para ello se debe seleccionar el nombre del recurso y su contenido se muestra en una nueva pestaña del navegador, permitiéndole al usuario ver la estructura del contenido y navegar a través de él.

En la UCI existe un **Repositorio de Objetos de Aprendizaje (RHODA)** que almacena a los objetos creados en la herramienta de autor CRODA, en formato .zip siguiendo el estándar SCORM. Para acceder al contenido de uno de estos objetos o cursos se debe seleccionar la opción que permite visualizarlo, mostrándose en una nueva pestaña del navegador. En la pestaña se mostrará el árbol de contenido y opciones que faciliten la navegación. Otra de las opciones que brinda el repositorio es la exportar los cursos a SCORM 1.2 y SCORM 2004. Con características similares a las citadas anteriormente cumple un repositorio usado en Venezuela⁵.

En las plataformas educativas se pueden encontrar varios de estos visores, pero sobre esto se hablará en el próximo epígrafe.

1.3 Visores de paquetes SCORM usados por plataformas educativas

El uso de visores de paquetes SCORM se ha extendido entre varias plataformas. Para la presente investigación se analizarán cuatro de ellas, aunque existen otras, se consideró que las características que estas presentaron son de interés para la investigación y además la información que proporcionaron fue suficiente. Se comenzará por Sakai, la cual se actualiza cada tres o cuatro meses y se le añaden herramientas constantemente para su integración, (17) como los visores SCORM. Luego a Moodle por ser “la plataforma educativa libre más conocida y extendida, con 67 000 sitios registrados, que ofrecen 5,5 millones de cursos, en los que participan 54 millones de usuarios”. (18) Como propietaria se escogió a Blackboard, por tener como uno de sus puntos focales más importantes la compatibilidad con los estándares

⁴ Se puede acceder a través de la siguiente dirección, <http://spar.unicauca.edu.co/spar/default.aspx>

⁵ Se puede acceder a través de la siguiente dirección, <http://roa.mppeu.gob.ve>

globales, como SCORM. (19) La última seleccionada fue Dokeos, esta permite la importación y exportación de archivos SCORM, IMS⁶ y AICC. (20)

1.3.1 Sakai

Sakai es una plataforma en línea desarrollada por una comunidad de educadores para facilitar la enseñanza y el aprendizaje colaborativo en instituciones educativas. El proyecto originado en la Universidad de Michigan, fue anunciado oficialmente en EDUCAUSE, en noviembre del 2003. La plataforma es distribuida gratuitamente como un programa de código abierto bajo la licencia de comunidad educativa. (21)

La plataforma es utilizada para el manejo de cursos, pero también es un ambiente de aprendizaje y colaboración que añade herramientas para el desarrollo de contenidos. Permite colocar el prontuario (anotación o resumen) y presentaciones dentro de la plataforma, y ofrece plantillas para diseñar lecciones de manera secuencial, para crear asignaciones, exámenes y concursos en línea. (21)

La plataforma cuenta con los visores SCORM que se muestran a continuación:

1.3.1.1 CRIM SCORM Player es basado en el estándar de la ADL SCORM 2004. Permite la validación de cursos SCORM 2004, es de código abierto. Está integrado totalmente en Sakai como una herramienta. Entre las limitaciones que presenta está que únicamente funciona con una base de datos que tenga como gestor MySQL⁷. Almacena estadísticas, pero la explotación es pobre. Presenta problemas al montar varios servidores en cluster⁸. (22)

1.3.1.2 Icodeon SCORM Player es un visor robusto, la configuración es flexible y de sencilla adaptación. La comunidad ha desarrollado una integración oficial (Integración de Sakai e Icodeon), por lo que es posible que se convierta en una solución muy utilizada. La integración añade un nuevo tipo de contenido a la herramienta de “Recursos” que permite subir un paquete SCORM y lo registra en Icodeon. Haciendo clic en el nombre del recurso se lanza el visor SCORM. Una funcionalidad importante es que Icodeon registra eventos en las tablas de Sakai, aunque todavía no existen informes ni otras características que aprovechen esta información.

⁶ Tiene como objetivo principal el desarrollo y promoción de especificaciones libres en formato XML, que faciliten el aprendizaje electrónico.

⁷ Sistema de gestión de bases de datos relacional, licenciado bajo la GPL de la GNU, es probablemente el gestor más usado en el mundo del software libre, debido a su gran rapidez y facilidad de uso.

⁸ Es un grupo de múltiples ordenadores unidos mediante una red de alta velocidad, de tal forma que el conjunto es visto como un único ordenador, más potente que los comunes de escritorio.

Puede ser configurable con múltiples bases de datos. No tiene soporte para la explotación de estadísticas. (22)

1.3.2 Moodle

Moodle es un Sistema de Gestión de Cursos de Código Abierto, en inglés Open Source Course Management System (CMS), conocido también como Sistema de Gestión de Aprendizaje o como Entorno de Aprendizaje Virtual. Es una herramienta para producir cursos basados en Internet, páginas web y procedimientos que permitan fácilmente la comunicación a través de Internet y el trabajo colaborativo. (23)

Algunas de las características que presenta la plataforma son: (24)

- ✓ Lleva registro y seguimiento completo de los accesos del alumno. Se dispone de informes de actividad de cada estudiante, con gráficos y detalles sobre su paso por cada módulo (último acceso, número de veces que lo ha leído).
- ✓ Los recursos que el docente entrega a sus estudiantes pueden ser de cualquier fuente y con cualquier formato, puesto que su programación está orientada a objetos. Es decir, soporta objetos como una característica fundamental del mismo, y es necesario tener la fuente del mismo para poder ejecutarlo.
- ✓ Responde a los estándares internacionales SCORM, el cual permite importar y exportar los contenidos a otras plataformas.

Moodle utiliza el visor de paquetes SCORM que se muestra a continuación:

1.3.2.1 Flex SCORM Player 2.0 es el visor que muestra los SCO construidos a través del Flex SCORM Builder, otorgándole la apariencia de un libro virtual. Está compuesto por una serie de herramientas que hacen del estudio una experiencia amigable y cómoda, optimizando los recursos que el alumno tiene a su disposición. Es extremadamente sencillo, rápido y ágil de usar. El libro virtual se basa en un documento en formato PDF para su visualización. Cumple con el estándar SCORM 2004. (25)

El visor conserva la correspondiente interacción con el estándar SCORM, pudiendo consultarse el seguimiento del alumno y evaluaciones. Cuenta con herramientas que complementan las existentes en Moodle, pero reduciendo en un alto porcentaje su complejidad, como los puntos de control o las estadísticas del alumno. El visor muestra la estadística del alumno local relativo al curso, combinándola con las herramientas nativas de estadísticas que la plataforma e-

learning disponga. Permite guardar como un acceso directo, aquellas páginas que el alumno considere como favoritas y recuperarlas de una forma fácil, directa y sencilla para su consulta. La herramienta ojeador de páginas permite pre-visualizar cualquier página del libro, pudiendo acceder a ella pulsando sobre la misma. (25)

Este visor también es utilizado por la plataforma Sakai para la visualización de los contenidos y seguimiento de los estudiantes.

1.3.3 Blackboard

Blackboard se fundó con la visión de transformar Internet en un potente entorno educativo. Es el principal proveedor de soluciones de educación electrónica. Presta sus servicios a varios niveles de enseñanza como primaria, secundaria y educación superior, además a agencias del gobierno o de las empresas. Blackboard tiene su sede central en Washington D.C., y cuenta además con oficinas y personal en Norteamérica, Europa y Asia. (19)

Para darle soporte al estándar SCORM, cuenta con el visor siguiente:

1.3.3.1 Content Player Building Block

El Content Player Building Block de Blackboard fue desarrollado internamente por medio de los equipos de ingenieros y seguridad de Blackboard, y cuenta con certificado ADL⁹. Permite a un instructor agregar contenidos que se ajusten a SCORM 1.2 y 2004, IMS, o las normas NLN¹⁰ a un curso. Los instructores pueden ver los IMS, SCORM y NLN dependiendo de lo que el administrador haya puesto a disposición.

Los instructores pueden importar en su curso un SCO y luego realizar actividades sobre este. Cuando un artículo del libro de calificaciones se asocia con el elemento de contenido SCORM, IMS o NLN, el instructor podrá ver los datos relacionados con las interacciones de los usuarios con el contenido. Esto se conoce como datos de intento. Los detalles pueden incluir el tiempo total que el usuario ha visto cada objeto de aprendizaje, el estado de finalización, las respuestas a las preguntas contenidas en el paquete y si la respuesta era correcta. El propósito de los datos de intento es ayudar al instructor en la determinación de la evaluación. (26)

⁹ Advance Distributed Learning es una iniciativa del Departamento de Defensa de los Estados Unidos, y sus socios en la industria, la educación, y los sectores privados y federales. El propósito fue desarrollar una estructura técnica común que contenga contenido en forma de objetos de aprendizaje reutilizables. (49)

¹⁰ National Learning Network, se encarga de proveer una infraestructura de red robusta y una amplia gama de programas de apoyo, información y asesoramiento, así como de desarrollar materiales digitales para la enseñanza y el aprendizaje.

Este visor cuenta con las siguientes opciones avanzadas: (27)

Los controles de navegación: puede ser de alguna utilidad para los profesores con un nivel básico de experiencia con SCORM, ya que estos ajustes pueden controlar el comportamiento, las características disponibles y la funcionalidad del reproductor de contenidos. Les permite incluir botones, barras y otras ayudas a la navegación, que puedan ser aprovechadas por los estudiantes.

Opción	Función
Mostrar barra de navegación	Determinar si en el reproductor SCORM se mostrará una barra de navegación para el alumno.
Mostrar el botón Terminar	Muestra un botón de salida del curso en la barra de navegación.
Mostrar el botón cerrar SCO	Aparece en la barra de navegación, debe permanecer apagada debido a que no es útil para la mayoría de los estudiantes.
Activar Anterior / Siguiente	Incluir controles anterior y siguiente en la barra de navegación, que le permite al estudiante avanzar o retroceder en el contenido.
Mostrar barra de progreso	Mostrar una barra de progreso para el contenido, por lo que los estudiantes puedan medir su progreso.
Mostrar ayuda	Muestra un botón de ayuda para el estudiante.

Tabla 1.2. Controles de navegación.

Comportamiento del lanzamiento

Opción	Función
Tipo de lanzamiento SCO	<ul style="list-style-type: none"> ✓ Marcos: lanzamiento del SCO en línea, en un marco del navegador, en lugar de abrirlo en una nueva ventana. ✓ Nueva ventana: lanzamiento del SCO en una nueva ventana del navegador. ✓ Nueva ventana sin barra de navegación: lanzamiento del SCO en una nueva ventana sin barra de herramientas.
Tipo de reproductor de lanzamiento	<ul style="list-style-type: none"> ✓ Marcos: iniciar reproductor SCORM en línea, en un marco del navegador, en lugar de abrirlo en una nueva ventana. ✓ Nueva ventana: lanzamiento del reproductor en una nueva ventana del navegador. ✓ Nueva ventana sin barra de navegación: lanzamiento del reproductor en una nueva ventana sin barra de herramientas.

Tabla 1.3. Comportamiento del lanzamiento.

1.3.4 Dokeos

Dokeos es una herramienta libre, colaborativa y creada para la administración de contenidos educativos. Incluye distribución de contenidos, calendario, proceso de entrenamiento, chat en texto, audio y video, administración de pruebas y guardado de registros.

La plataforma soporta el estándar SCORM, por lo que cuenta con el visor que a continuación se describe:

1.3.4.1 SCORM Nube

La Gran Nube de SCORM añadido a la instalación de Dokeos le proporciona un sistema de gestión del aprendizaje con la versión más actualizada de SCORM y la capacidad de importar y exportar paquetes SCORM, la secuencia de cursos y la compatibilidad con casi cualquier contenido.

Permite que los estudiantes puedan aprender en cualquier lugar en la Web, pueden acceder al visor sin la necesidad de entrar a la plataforma, debido a que solo necesitan de un navegador. (28)

“SCORM Nube no solo entrega y le da seguimiento al contenido. También puede crear simples pruebas y exámenes con Quizzage¹¹, o convertir archivos de video de paquetes con SCORM ScoTube¹²”. (28)

Cuenta con una herramienta de informes que permite ver el historial de los alumnos, el tiempo que pasaron en los cursos y qué preguntas han contestado correcta o incorrectamente. Brinda la posibilidad de ver los informes de diferentes maneras, de forma general con todos los estudiantes y cursos a la vez, y de forma individual de un curso y de un estudiante.

La pista de pruebas es otro de los servicios que ofrece el visor y se emplea para ejecutar el curso y verificar que todo funciona sin problemas.

1.3.5 Tabla resumen de los visores estudiados

Acción	Validar	Lanzar el visor	Navegar	Importar ¹³	Almacenar seguimiento	Consultar seguimiento
CRIM SCORM Player	Si	Desconocida	Si	No	Si	No
Icodeon SCORM Player	No	En una nueva ventana	Si	No	Si	No
Flex SCORM Player 2.0	No	Documento en forma de PDF	Si	No	Si	Si
SCORM Nube	Si	En un Navegador	Si	Si	Si	Si
Content	Si	En un marco	Si	Si	Si	Si

¹¹ Herramienta utilizada para crear concursos, pruebas y encuestas.

¹² Herramienta que permite añadir un video ya sea importándolo directamente o indicando la url del mismo en Youtube.

¹³ Los que tienen “No”, el proceso de importación se realiza a través de la plataforma.

Player Building Block		del navegador o en una nueva ventana				
-----------------------------	--	---	--	--	--	--

Tabla 1.4. Funcionalidades de los visores

Descripción de las acciones

Validar: se refiere a si antes de mostrar el contenido, se valida el paquete SCORM.

Lanzar el visor: forma en que aparecerá el visor.

Navegar: permite que el usuario navegue por el contenido.

Importar: brinda la posibilidad de importar los paquetes SCORM.

Almacenar seguimiento: almacena los datos de la interacción del estudiante con el contenido.

Consultar seguimiento: permite ver los datos de seguimiento del estudiante.

1.3.6 Conclusiones del estudio de los visores SCORM

El análisis anterior permitió identificar de qué forma se llevan a cabo los procesos en los visores, facilitando el análisis y diseño del visor de paquetes SCORM que se desea realizar para la Plataforma Educativa Zera. No se tendrán en cuenta las funcionalidades de importar y validar el paquete desde el visor, debido a que en la plataforma antes mencionada, está definida la manera de importar los recursos. Los cursos SCORM serán importados a través del macro índice de la materia, seleccionando en esta, el capítulo, tema o subtema que contará con el curso; antes de importar el paquete a la plataforma este será validado. Se tomará de Icodeon SCORM Player la idea de utilizar el paquete SCORM en su conjunto como un recurso más de la plataforma, además de que al hacer clic en el nombre del recurso se lance el visor. De Content Player Building Block se emplearán algunas de las opciones que presenta en cuanto al lanzamiento del visor en una nueva ventana y la de brindar opciones de navegación. Todos los visores estudiados almacenan el seguimiento de los estudiantes, demostrando que es una característica de suma importancia y que no debe faltar, por lo que será incorporado a la propuesta de solución. En el proyecto existe un subsistema llamado Reportes, que es el encargado de agrupar los reportes relacionados con la actividad del estudiante en la plataforma. Por esta razón el visor, no contará con la responsabilidad de mostrar las trazas.

1.4 Metodología, lenguaje y herramientas

Esta investigación forma parte del proyecto Alfaomega, donde ya han sido definidos la metodología que guía el proceso de desarrollo, el lenguaje y las herramientas para desarrollar los artefactos que conforman el visor. A continuación se muestran cada uno de ellos.

1.4.1 Metodología

Todo proyecto de software necesita hacer uso de las metodologías de desarrollo de software, teniendo en cuenta que las mismas establecen un conjunto de pasos y procedimientos que deben seguirse, si se desea obtener un proyecto exitoso. Permiten que el trabajo sea más organizado, que se ahorren recursos, tiempo y se obtenga un producto con mayor calidad.

Rational Unified Process

El Proceso Unificado de Desarrollo, en lo adelante RUP, es el resultado de varios años de desarrollo y uso práctico en el que se han unificado técnicas de desarrollo, a través del Lenguaje Unificado de Modelado, en lo adelante UML, y trabajo de muchas metodologías utilizadas por los clientes. Al ser un proceso en su modelación define los elementos: trabajadores, actividades, artefactos y flujos de trabajo.

Características principales: (29)

- 1- Guiado por Casos de Uso:** los casos de uso reflejan lo que los usuarios futuros necesitan y desean. Los casos de uso guían el proceso de desarrollo. Los modelos que se obtienen, como resultado de los diferentes flujos de trabajo, representan la realización de los casos de uso.
- 2- Centrado en la Arquitectura:** la arquitectura muestra la visión común del sistema completo en la que el equipo de proyecto y los usuarios deben estar de acuerdo, por lo que describe los elementos del modelo que son más importantes para su construcción, los cimientos del sistema que son necesarios como base para comprenderlo, desarrollarlo y producirlo económicamente.
- 3- Iterativo e incremental:** RUP propone que cada fase se desarrolle en iteraciones. Una iteración involucra actividades de todos los flujos de trabajo, aunque desarrolla fundamentalmente algunos más que otros. Es práctico dividir el trabajo en partes más pequeñas o mini-proyectos. Cada mini-proyecto es una iteración que resulta en un incremento. Las iteraciones hacen referencia a pasos en los flujos de trabajo, y los incrementos, al crecimiento del producto.

En RUP se han agrupado las actividades en grupos lógicos definiéndose nueve flujos de trabajo principales. Los seis primeros son conocidos como flujos de ingeniería y los tres últimos como de apoyo, ellos son: (29)

- ✓ **Modelamiento del Negocio:** describe los procesos de negocio, identificando quiénes participan y las actividades que requieren automatización.
- ✓ **Requerimientos:** define qué es lo que el sistema debe hacer, para lo cual se identifican las funcionalidades requeridas y las restricciones que se imponen.
- ✓ **Análisis y Diseño:** describe cómo el sistema será realizado a partir de las funcionalidades previstas y las restricciones impuestas (requerimientos), por lo que indica con precisión lo que se debe programar.
- ✓ **Implementación:** define cómo se organizan las clases y objetos en componentes, cuáles nodos se utilizarán y la ubicación en ellos de los componentes y la estructura de capas de la aplicación.
- ✓ **Prueba:** busca los defectos a lo largo del ciclo de vida.
- ✓ **Instalación:** produce la liberación del producto y realiza actividades (empaquete, instalación y asistencia a usuarios) para entregar el software a los usuarios finales.
- ✓ **Administración del Proyecto:** involucra actividades con las que se busca producir un producto que satisfaga las necesidades de los clientes.
- ✓ **Administración de Configuración y Cambios:** describe cómo controlar los elementos producidos por todos los integrantes del equipo de proyecto en cuanto a: utilización/actualización concurrente de elementos y control de versiones.
- ✓ **Ambiente:** contiene actividades que describen los procesos y herramientas que soportará el equipo de trabajo del proyecto; así como el procedimiento para implementar el proceso en una organización.

Está compuesto por cuatro fases que representan un ciclo de desarrollo en la vida de un producto de software ellas son: Inicio, Elaboración, Construcción y Transición.

La metodología RUP fue seleccionada en Alfaomega, teniendo en cuenta las ventajas que esta ofrece y la magnitud del proyecto, donde el número de miembros es elevado, el tiempo para finalizar el producto es largo, y el cliente y el equipo de desarrollo se encuentran separados geográficamente.

El empleo de esta metodología en el proyecto ha posibilitado que:

- ✓ Los clientes no han tenido que esperar a que el producto sea terminado para utilizarlo.

- ✓ A medida que se realizan las entregas, el cliente ha modificado e incorporado nuevos requerimientos.
- ✓ Se disminuya el riesgo de fracaso del proyecto completo, debido a que se va entregando por incrementos.
- ✓ Las entregas queden bien documentadas.

1.4.2 Lenguaje de modelado

Antes de comenzar a desarrollar un software es necesario haber hecho primero un modelado del sistema, para ello se emplean los lenguajes de modelado. Los modelos que se obtienen son usados por arquitectos, líderes de proyecto, analistas, probadores, programadores, en general por todo el equipo de desarrollo. Estos lenguajes permiten que muchos errores sean detectados desde etapas tempranas y que los beneficios económicos sean mayores.

UML

“UML puede utilizarse para visualizar, especificar, construir y documentar los artefactos de un sistema que involucra gran cantidad de software. El UML soporta un conjunto rico en elementos de notación gráficos. Describe la notación para clases, componentes, nodos, actividades, flujos de trabajo, casos de uso, objetos, estados y cómo modelar la relación entre esos elementos”.

(30)

Características. (30)

- ✓ Proporcionar a los usuarios un lenguaje de modelado visual expresivo y utilizable para el desarrollo e intercambio de modelos significativos.
- ✓ Ser independiente del proceso de desarrollo y de los lenguajes de programación.
- ✓ Soportar conceptos de desarrollo de alto nivel como pueden ser colaboraciones, frameworks, patrones y componentes.

Los principales beneficios de UML son: (31)

- ✓ Mejores tiempos totales de desarrollo (de 50 % o más).
- ✓ Modelar sistemas (y no solo de software) utilizando conceptos orientados a objetos.
- ✓ Establecer conceptos y artefactos ejecutables.
- ✓ Encaminar el desarrollo del escalamiento en sistemas complejos de misión crítica.
- ✓ Crear un lenguaje de modelado utilizado tanto por humanos como por máquinas.
- ✓ Mejor soporte a la planeación y al control de proyectos.
- ✓ Alta reutilización y minimización de costos.

UML cuenta con una gran variedad de modelos, fáciles de entender y visualmente atractivos que permiten que el tiempo empleado para el desarrollo del software sea menor.

Para desarrollar los diagramas de clases del diseño con estereotipos web, se empleará la extensión de UML propuesta por Conallen. (32)

1.4.3 Herramienta de modelado

“Las herramientas de Ingeniería de Software Asistida por Ordenador, en inglés Computer Aided Software Engineering, en lo adelante CASE, son un conjunto de programas y ayudas que dan asistencia a los analistas, ingenieros de software y desarrolladores, durante todos los pasos del ciclo de vida de desarrollo de un software.” (33)

Visual Paradigm

“Visual Paradigm es una herramienta que sirve para realizar modelado UML. Presenta características gráficas muy cómodas que facilitan la realización de los diagramas siguientes: diagramas de clases, casos de uso, comunicación, secuencia, estado, entidad-interrelación, actividades y componentes. Permite la exportación de imágenes a JPG, PNG y SVG. Se encuentra integrado con diversos entornos de desarrollo integrado como NetBeans, JDeveloper, Eclipse, JBuilder.” (34)

Esta herramienta está diseñada para desarrollar software con programación orientada a objetos. Al ser integrada con Eclipse permite sincronizar el modelo de diseño con el código en todo el ciclo de desarrollo, se pueden realizar cambios en el código generado y actualizar estos cambios en el diseño.

Visual Paradigm es fácil de emplear, cuenta con todos los diagramas que el equipo necesita. El equipo se siente identificado y en un ambiente favorable, debido a que los objetos que se modelan son una abstracción de la realidad.

1.4.4 Herramienta para el diseño de prototipos de interfaz de usuario

Como parte de las tareas de análisis y diseño de un software, está el diseño de los prototipos de interfaz de usuario. Estos permiten enseñarle al cliente una primera versión de cómo quedará el software una vez implementado, y que estos puedan comprender y evaluar las funcionalidades que el sistema ofrecerá.

Balsamiq Mockups

“Balsamiq Mockups es una herramienta que permite diseñar prototipos de interfaz de usuario de manera sencilla y rápida. Dispone de más de 75 modelos ya definidos con los diferentes elementos de las interfaces de usuario para montar los prototipos. Todos los dibujos tienen un estilo que se asemeja a cuando lo hacemos a mano para dotarlo de un poco más de personalidad y realismo.” (35)

Con la herramienta Balsamiq Mockups se obtienen diseños que son de gran calidad y aceptación, la herramienta es fácil de utilizar y amigable. Los que la emplean se encuentran seguros, confiados y lo consideran un entorno ameno. Los usuarios finales y los desarrolladores entienden lo que se muestra en los prototipos, posibilitando la realización de cambios tempranos en los requerimientos y así implementar lo que realmente desea el cliente.

La herramienta cuenta con dos licencias una comercial y otra totalmente gratuita ¹⁴, empleándose esta última.

1.4.5 Conclusiones de la metodología, el lenguaje y las herramientas

Las tecnologías antes estudiadas permitirán un desarrollo exitoso de la investigación, debido a que cuentan con características y ventajas ya probadas en el proyecto Alfaomega. De RUP destacar la capacidad de permitir organizar el equipo y el trabajo, a través de la definición de los roles y los artefactos que estos deben generar en cada uno de los flujos de trabajo. La experiencia con UML y Visual Paradigm permitirá ahorrar tiempo y obtener resultados rápidamente. En el caso de Balsamiq a pesar de no llevar mucho tiempo empleándose, cuenta con una interfaz intuitiva y amigable, que hace que el trabajo en esta se realice de forma sencilla.

1.5 Conclusiones

En este capítulo se realizó un estudio del estado del arte relacionado con los conceptos vinculados al problema de investigación. Se estudiaron las diferentes modalidades de visualizar los paquetes SCORM. Además se identificaron las características de soluciones similares utilizadas a nivel mundial en plataformas educativas, donde se pudo apreciar que cuentan con servicios de gran utilidad para estudiantes y profesores, y que han estado en los últimos años en constante evolución. De estas soluciones se tomaron algunas de sus funciones que serán adaptadas para el presente diseño como son: la navegabilidad a través del contenido, el

¹⁴ Se puede encontrar en <http://www.balsamiq.com/download>

almacenamiento de la interacción del estudiante con el visor y tomar el curso como un recurso más de la plataforma. También se mencionaron algunas de las características de la metodología, lenguajes y herramientas que se emplearán en el presente trabajo, que ya se encontraban definidas dentro del proyecto Alfaomega.

Capítulo II: Características del Sistema

Uno de los primeros pasos que se debe realizar cuando se desea construir un software es comprender el contexto del sistema, para ello se debe realizar el modelo de negocio y/o modelo de dominio. Con el modelo de negocio se describen los procesos y con el de dominio se describen los conceptos más representativos del negocio.

Cuando se entiende el funcionamiento del negocio, se procede a realizar la captura de los requerimientos funcionales y no funcionales, que son las capacidades y cualidades que debe cumplir el sistema. Luego se identifican y describen actores y casos de uso, y se refleja su interacción mediante el diagrama de casos de uso del sistema.

2.1 Modelo de dominio

Durante el análisis del negocio no se logró determinar fronteras bien establecidas de los procesos que se llevan a cabo, por lo que se identificaron conceptos que permitieron realizar el modelo de dominio. En este los conceptos son representados como clases y entre estas se establecen relaciones.

El modelo de dominio ayuda a comprender los conceptos claves del negocio y a disminuir la brecha de representación, entre cómo lo ven los clientes y la representación real de la solución.

2.1.2 Diagrama del Modelo de Dominio

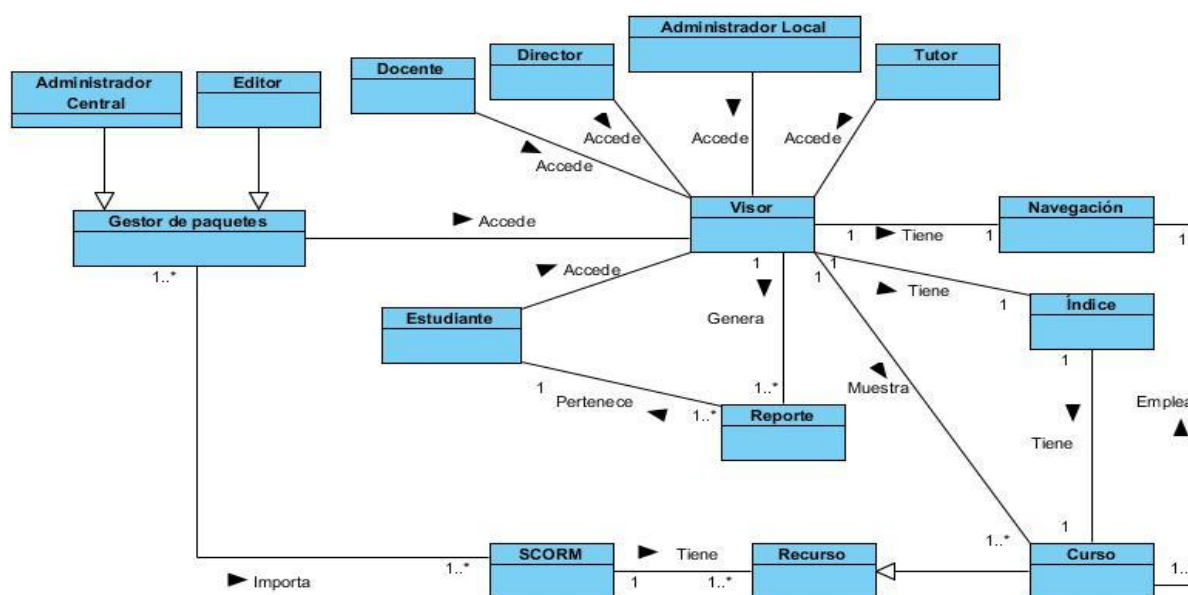


Figura 2.1 Modelo de dominio

Descripción de las clases del diagrama de dominio

Gestor de paquetes: usuario que representa la generalización de los actores: Editor y Administrador Central. El cual puede tanto importar paquetes SCORM como acceder al visor.

Editor: persona encargada de gestionar el contenido de las materias que se imparten en la escuela.

Docente: persona que puede impartir una o varias materias en uno o muchos grupos. Puede asignar actividades al grupo o individualmente, hace el seguimiento a las actividades de todos los estudiantes que estén a su cargo y evalúa a los estudiantes que le imparte el contenido.

Administrador Central: se encarga de la administración global del sitio y tiene permiso sobre todos los componentes de la plataforma, genera contraseñas, nuevos reportes, usuarios, crea los cursos y carga el material que los compone.

Director: persona encargada de dirigir una escuela. Puede acceder a los diferentes servicios que ofrece el visor de paquetes SCORM.

Administrador local: persona que realiza las tareas de administración de la plataforma que se encuentra en la escuela. Puede acceder a los diferentes servicios que ofrece el visor de paquetes SCORM.

Tutor: persona que supervisa y guía al estudiante en las actividades docentes. Puede acceder a los diferentes servicios que ofrece el visor de paquetes SCORM.

Navegación: opciones que serán de utilidad para el usuario al navegar por el curso SCORM.

Estudiante: usuario a quien va dirigido el curso, por lo que podrá acceder a todas las funcionalidades del visor.

Reporte: constituyen las trazas que se generan a partir de la interacción del estudiante con el contenido.

Índice: representa la estructura del contenido del curso a través del índice de contenido.

SCORM: paquete que contiene recursos que serán utilizados en la plataforma.

Recurso: elemento que permite enriquecer los contenidos de las diferentes materias.

Curso: recurso que contribuirá con el aprendizaje de los estudiantes y que podrá ser consultado por todos los usuarios de la plataforma.

Visor: herramienta que permitirá que los cursos que poseen los paquetes SCORM puedan ser aprovechados en la plataforma, permitiendo la navegabilidad a través de los mismos.

2.2 Descripción del sistema propuesto

Antes de aplicar la solución que a continuación se explica, se debe cambiar la forma de importar un paquete SCORM, ofreciéndole al usuario la posibilidad de seleccionar lo que va a importar, si los recursos del paquete por separado o el curso en su totalidad. El visor propuesto tiene como objetivo, permitir la visualización de los cursos que contienen los paquetes SCORM, navegar por su contenido y permitir la comunicación de estos con la plataforma. Para navegar a través del contenido del paquete, se debe seleccionar previamente el curso, para que sea lanzado el reproductor. Desde que se ejecute el visor hasta su cierre, se debe ofrecer una vía que permita la comunicación entre el curso y la plataforma. El visor ofrecerá opciones de navegación, un listado de los cursos asociados al programa de estudio, permitiéndole al usuario acceder a ellos al seleccionarlos, sin tener que salir del visor. Todos los usuarios de la plataforma tienen acceso al visor.

2.3 Requerimientos del software

Los requerimientos son una descripción de las necesidades de un producto. Su objetivo es identificar y documentar lo que en realidad se necesita. Estos deben ser definidos de manera inequívoca de modo que se detecten los riesgos y no se presenten sorpresas al momento de entregar el producto. También deben cumplir las siguientes características, especificados por escrito, descritos como una característica y posibles de probar o verificar.

2.3.1 Requerimientos funcionales

Los requerimientos funcionales, en lo adelante RF, especifican qué debe hacer un sistema.

RF 1. Importar paquete SCORM como curso

El usuario podrá seleccionar la opción que le permite importar un curso SCORM, este es validado¹⁵ y es agregado a la base de datos como un recurso más de la plataforma.

RF 2. Ejecutar el visor de paquetes SCORM

¹⁵ Se valida que:

1. Sea un archivo .zip y que exista en su raíz el fichero "imsmanifest.xml".
2. El "imsmanifest.xml" esté correctamente formado.
 - 2.1 El "schema version" del "imsmanifest" corresponda con SCORM 2004.
 - 2.2 Contenga las únicas etiquetas obligatorias "manifest", "schema" "organizations", al menos una etiqueta "organization" con un "item" y un "title", y "resources".

Una vez que el usuario seleccione el curso SCORM se debe lanzar el reproductor del paquete SCORM en una nueva ventana.

RF 3. Permitir la comunicación del curso con la plataforma.

Se debe garantizar la implementación de la API descrita en el estándar.

RF 4. Acceder a cursos asociados

Se le brinda la posibilidad al usuario de ver los demás cursos SCORM asociados al programa de estudio y de acceder a ellos al seleccionarlos.

RF 5. Visualizar y ocultar el árbol de contenido.

En el visor el usuario tendrá la posibilidad de ver el árbol de contenido del curso SCORM y de ocultarlo cuando lo desee, de forma tal que pueda ser aprovechado el mayor espacio posible en la visualización de la información.

RF 6. Visualizar y ocultar la barra de navegación.

En el visor el usuario tendrá la posibilidad de ver y ocultar la barra que contiene los botones de navegación Anterior, Siguiente y Salir.

RF 7. Permitir la navegación por los contenidos

Permite a los usuarios la navegación por cada uno de los contenidos del curso a través del índice del contenido y también a través de las opciones de navegación Anterior, Siguiente y Salir, respetando las reglas de secuenciación.

RF 8. Guardar avance del contenido.

Cuando el usuario que accede al visor es un estudiante y el curso contiene la implementación de almacenar trazas, se guardará el avance en el contenido.

2.3.2 Requerimientos no funcionales

Los requerimientos no funcionales, en lo adelante RNF, especifican cómo debe ser el sistema, las propiedades o cualidades que debe tener, controlan la especificación, implementación y ejecución de los requisitos funcionales. Muchas veces influyen en la aceptación o no del producto por parte del cliente. Estos requerimientos hacen al producto atractivo, usable, rápido o confiable.

RNF 1. Apariencia o interfaz externa

- ✓ La información aparecerá correctamente organizada de forma tal que el usuario, pueda encontrar lo que busca rápidamente.
- ✓ El diseño de las interfaces será sencillo, con pocas imágenes y colores.
- ✓ Consistente: siempre que sea posible las acciones similares deben ejecutarse de similar forma.

- ✓ El sistema debe notificar al usuario ante la presencia de un error.

RNF 2. Portabilidad

- ✓ Se necesita un navegador web como Mozilla Firefox 3.6 o superior, Internet Explorer 7 o superior, Chrome 7 o superior, Safari u Opera 10 o superior.

RNF 3. Usabilidad

- ✓ El sistema podrá ser utilizado por cualquier usuario que tenga conocimientos básicos en el uso de aplicaciones web.
- ✓ Las acciones a realizar en el visor serán visibles.

RNF 4. Seguridad

- ✓ A los usuarios autorizados se les garantizará el acceso a la información.
- ✓ El nivel de acceso a la información dependerá del rol con que se autentique el usuario.

RNF 5. Rendimiento

El tiempo de respuesta cuando haya hasta 200 usuarios accediendo simultáneamente a los cursos SCORM, no debe ser mayor a dos segundos.

RNF 6. Restricciones de diseño e implementación:

- ✓ El lenguaje de programación deberá ser PHP en su versión 5.2.5 o superior.
- ✓ Como marco de trabajo se utilizará Symfony 1.4.15.
- ✓ Como servidor web se utilizará Apache 2.2.6.
- ✓ El sistema gestor de bases de datos deberá ser PostgreSQL 9.0.

2.4 Modelo de Casos de Uso del Sistema

El Modelo de Casos de Uso del Sistema está compuesto por actores, casos de uso y sus relaciones.

Los actores del sistema: (36)

- ✓ No son parte de él.
- ✓ Pueden intercambiar información con él.
- ✓ Pueden ser un recipiente pasivo de información.
- ✓ Pueden representar el rol que juega una o varias personas, un equipo o un sistema automatizado.

Los casos de uso son artefactos narrativos que describen, bajo la forma de acciones y reacciones, el comportamiento del sistema desde el punto de vista del usuario.

2.4.1 Patrones de casos de uso

Con el avance de la Ingeniería de Software y la práctica en la utilización de casos de uso, surgen los patrones de casos de uso, los cuales permiten agrupar los casos de uso y estructurar los diagramas. Además permiten reducir el tiempo que se necesita para desarrollar un software.

A continuación se muestra uno de estos patrones:

- ✓ Múltiples Actores - Roles comunes.
 - Se emplea cuando a un mismo caso de uso pueden acceder dos o más actores y estos comparten el mismo rol. La aplicación del patrón se puede ver a través del caso de uso Importar curso, que pueden acceder los actores Administrador Central y Editor para realizar las mismas acciones, por lo que se crea el actor Gestor de paquetes.

2.4.2 Diagrama de Actores

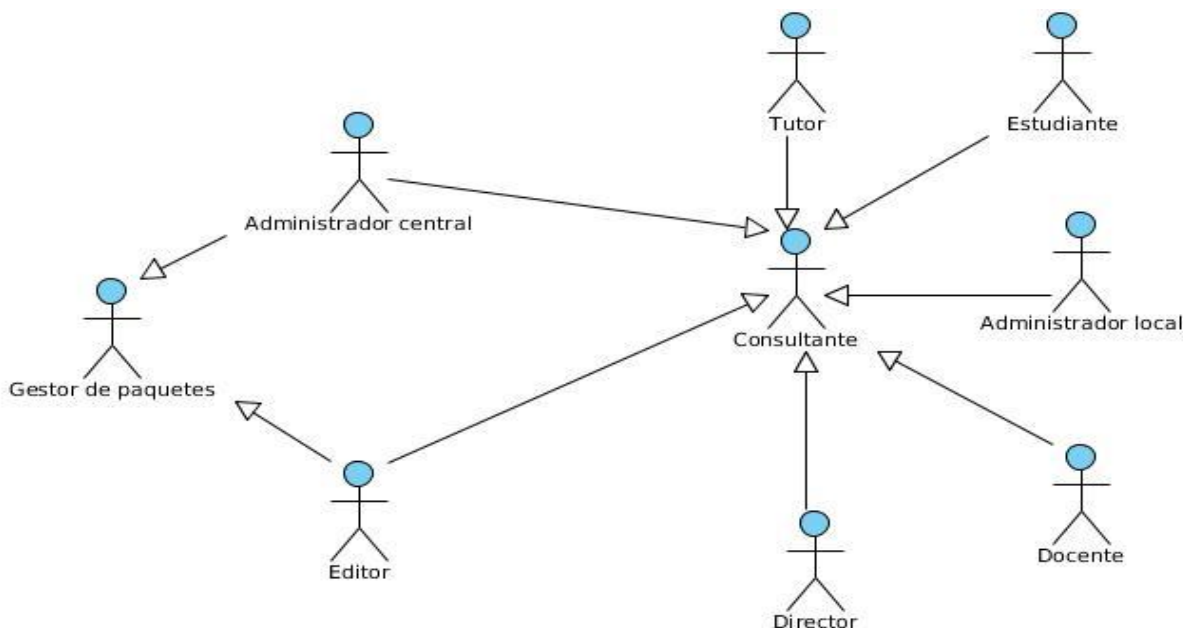


Figura 2.2 Diagrama de actores

2.4.3 Descripción de los actores del sistema

Actor	Descripción
Docente	Persona que imparte contenidos, evalúa y da seguimiento a sus estudiantes.

Gestor de paquetes	Usuario que representa la generalización de los actores: Editor y Administrador central, el cual puede importar paquetes SCORM a la plataforma, seleccionando previamente lo que desea, si los recursos del paquete por separado o el curso en su totalidad. También podrá asociar el curso SCORM a una página como un recurso más.
Consultante	Usuario que representa la generalización de los actores: Docente, Administrador central, Administrador local, Editor, Estudiante, Director y Tutor, el cual puede usar todas las funcionalidades presentes en el visor.
Administrador local	Realiza las tareas de administración en la plataforma instalada en la escuela.
Administrador central	Se encarga de la administración global del sitio y tiene permiso sobre todos los componentes de la plataforma, genera contraseñas, nuevos reportes, usuarios, crea los cursos y carga el material que los compone.
Director	Persona encargada de dirigir una escuela, ya sea porque es el dueño o porque se le asignó la tarea. Puede revisar y analizar la información a nivel de estudiante, curso, docente y escuela.
Editor	Persona encargada de gestionar el contenido de las materias que se imparten en la escuela.
Tutor	Persona que supervisa y guía al estudiante en las actividades docentes.
Estudiante	Persona que recibe contenidos educativos en uno o varios grupos de una escuela.

Tabla 2.1. Descripción de los actores del sistema.

2.4.4 Diagrama de Casos de Uso del Sistema

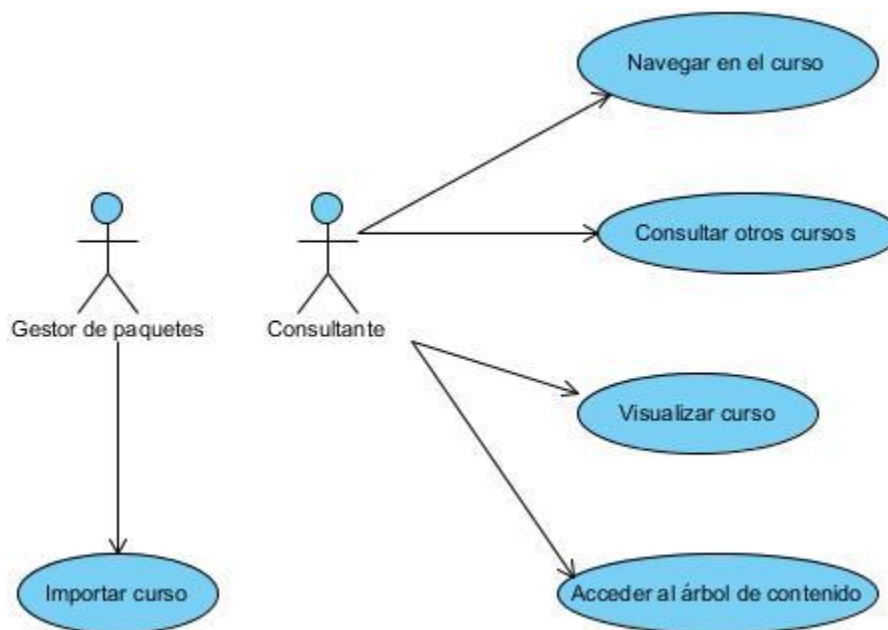


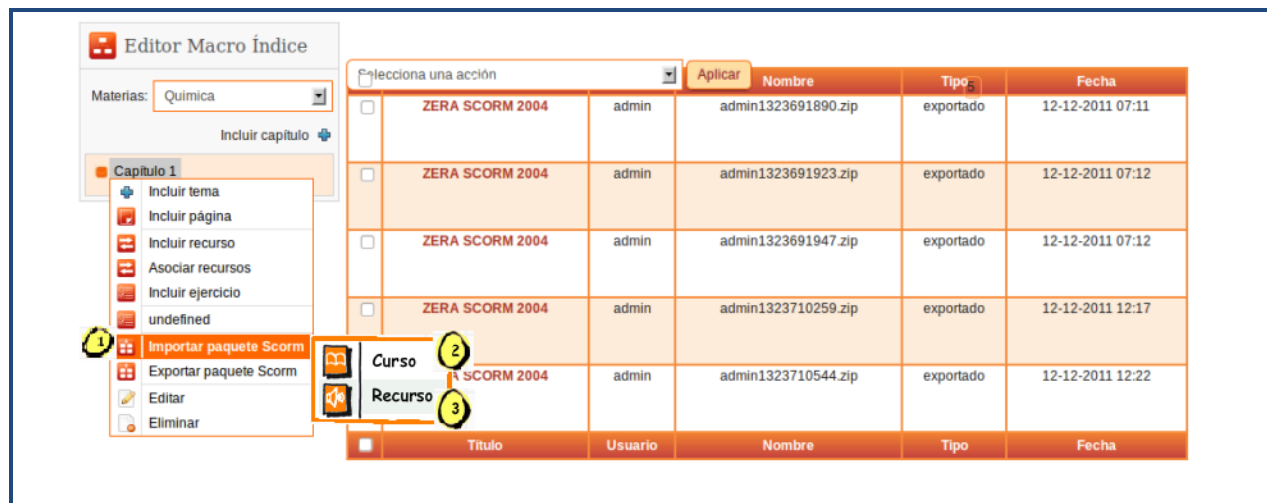
Figura 2.3 Diagrama de Casos de Uso del Sistema

2.4.5 Descripción de los casos de uso del sistema

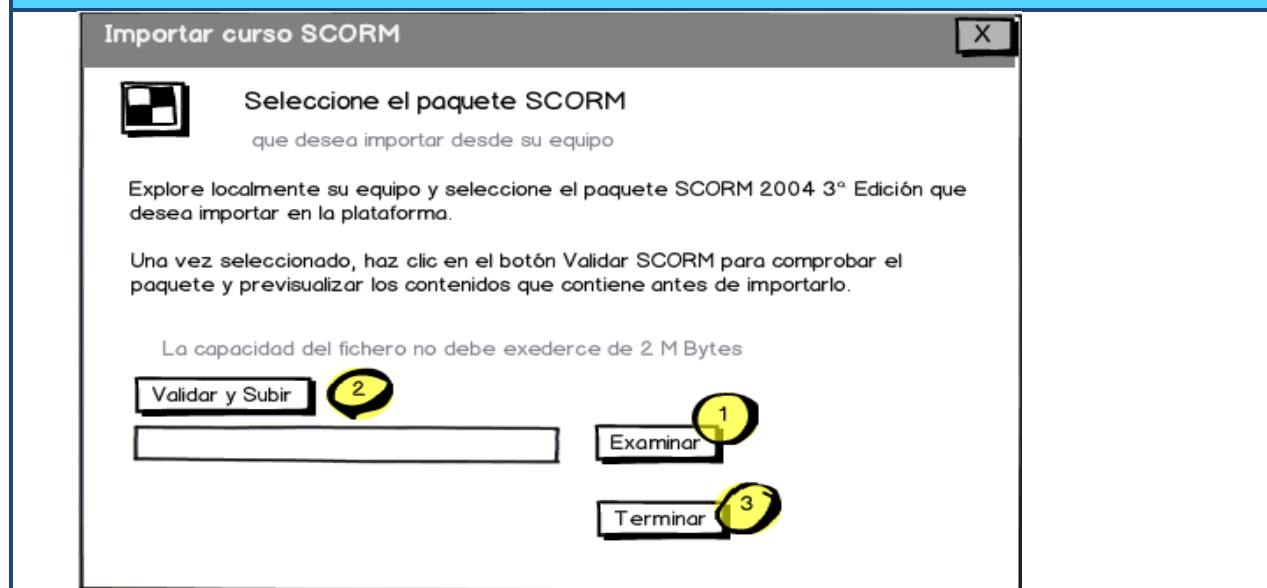
2.4.5.1 Descripción del CU Importar curso

Nombre	Importar curso
Objetivo	Importar un curso SCORM a la plataforma.
Actor	Gestor de paquetes: Administrador central y Editor.
Precondiciones	Debe haberse generado el escritorio de trabajo.
Postcondiciones	Se importó un curso SCORM válido a la plataforma.
Resumen del caso de uso	El caso de uso se inicia cuando el actor decide importar un paquete SCORM, para ello debe seleccionar un elemento del índice, luego escoger la opción Importar paquete SCORM y después seleccionar Curso. Una vez realizada esta acción el actor puede seleccionar y validar el paquete, el sistema ofrece como respuesta un mensaje notificando la valides o invalides del paquete, si es permitido lo descomprime y lo guarda en la carpeta correspondiente (uploads/resources/referencia/SCORM/id-de-la-BD). El caso de uso finaliza cuando se selecciona la opción Siguiente y el paquete es incorporado a la

	plataforma como un recurso más y el sistema agrega el curso SCORM al elemento seleccionado en el índice.
Referencia	RF 1
Flujo Básico.	
Acciones del actor	Respuesta del sistema
1. El caso de uso se inicia cuando el actor selecciona la opción de Importar paquete SCORM como un curso a un elemento del índice.	
	2. Muestra la interfaz de importar un curso SCORM.
3. Selecciona el paquete SCORM que desea importar y selecciona la opción que le permite validar y subir el paquete.	
	4. Valida que el paquete cumpla con las especificaciones del estándar y con el tamaño establecido.
	5. Muestra un mensaje con la valides del paquete. <ul style="list-style-type: none"> • Paquete válido.
	6. Descomprime el paquete y lo guarda en la carpeta correspondiente (uploads/resources/referencia/SCORM/id-de-la-BD)
	7. Incorpora el curso SCORM al capítulo, tema o subtema seleccionado.
	8. Incorpora el paquete a la base de datos como un recurso más.
	9. El caso de uso termina.
"Prototipo de interfaz 1.a"	



“Prototipo de interfaz 2.a”



Flujo Alterno

4.a El paquete no cumple con las especificaciones

Acciones del actor	Respuesta del sistema
	<p>4. a.1 Muestra un mensaje con los posibles errores.</p> <ul style="list-style-type: none"> • El paquete no cumple con las especificaciones del estándar. • El tamaño del paquete no está permitido.

	4. a.2 Regresa al paso 2 del flujo básico.
--	--

Tabla 2.2. Descripción del caso de uso del sistema, Importar curso SCORM.

2.4.5.2 Descripción del CU Visualizar curso

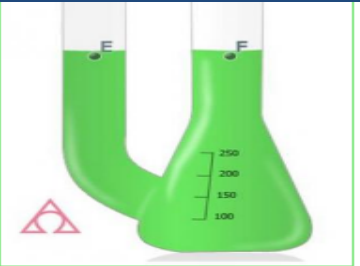
Nombre	Visualizar curso
Objetivo	Visualizar el curso.
Actor	Consultante
Precondiciones	Debe existir un curso.
Postcondiciones	Se visualizó el curso.
Resumen del caso de uso	El caso de uso se inicia cuando el actor selecciona un curso SCORM para ser visualizado, el sistema accede a la carpeta (uploads/resources/referencia/SCORM/id-de-la-BD), luego lee el fichero imsmanifes.xml contenido dentro del paquete y construye el árbol de los contenidos. Además ofrece los servicios implementados, correspondientes al API de SCORM 2004 que le permitirá al SCO comunicarse con el LMS.
Referencia	RF 2 y RF 3
Flujo Básico	
Acciones del actor	Respuesta del sistema
1. El caso de uso se inicia cuando el actor selecciona un curso.	
	2. Realiza las siguientes acciones: <ul style="list-style-type: none"> • Accede a la carpeta (uploads/resources/referencia/SCORM/id-de-la-BD). • Accede al archivo imsmanifiest.xml para tomar la estructura del contenido: <ul style="list-style-type: none"> ✓ Título del curso. ✓ Nombre de los elementos


	<p>que conforman el curso.</p> <ul style="list-style-type: none"> ✓ Organización. • Además busca en el manifiesto los datos necesarios para cumplir con las reglas de navegación. • Construye el árbol de contenido.
	<p>3. Ofrece varios servicios correspondientes al API que pueden ser usados por el SCO para comunicarse con el LMS tales como:</p> <ul style="list-style-type: none"> • Inicializar la comunicación del SCO con el LMS. Se inicializan varios datos del modelo como: <ul style="list-style-type: none"> ✓ cmi.learner_id ✓ cmi.learner_name ✓ cmi.location ✓ cmi.credit ✓ cmi.entry ✓ cmi.score.raw ✓ cmi.total_time ✓ cmi.suspend_data ✓ cmi.launch_data • Terminar la comunicación del SCO con el LMS. • Obtener valores de seguimiento del estudiante de la base de datos tales como: <ul style="list-style-type: none"> ✓ Ubicación actual del estudiante. ✓ Si el estudiante ha completado el SCO. ✓ Progreso del estudiante. • Modificar y guardar valores en la base de datos del LMS.

	<ul style="list-style-type: none"> ✓ Cantidad de tiempo en el curso. ✓ Valores máximo y mínimo del rango de puntuación. • Gestionar los errores. <ul style="list-style-type: none"> ✓ Código del error. ✓ Descripción corta del código de error. ✓ Información detallada del error.
	4. El caso de uso termina.

“Prototipo de interfaz 1.a”

recipientes unidos entre sí por un tubo. La superficie libre de un líquido en reposo que permanece en vasos comunicantes de cualquier forma, se encuentra al mismo nivel. En la vida cotidiana es muy común encontrarnos con vasos comunicantes, ejemplos: las redes hidráulicas de una casa o edificio forman vasos comunicantes, las cafeteras, las fuentes de los parques, diferentes sistemas de riego, entre otros.



Vaso comunicante 

La acción que ejerce el aire sobre los cuerpos se denomina presión atmosférica. La primera comprobación experimental de una presión asociada al aire fue realizada por Torricelli. El valor que él determinó para 1 atmósfera de presión fue el equivalente a una altura de 760 mm en una columna de mercurio. La presión atmosférica medida a 0 °C es equivalente a 101 300 Pa, que es la unidad en el Sistema Internacional

La presión atmosférica disminuye con la altura, por lo que se deben tomar medidas para contrarrestar los efectos en la salud al practicar alpinismo o al viajar entre ciudades que se diferencien notablemente por su altura al nivel del mar. Con un popote no succionas el refresco, puedes beberlo ya que es empujado por la presión atmosférica.

C Curso
1
|<
<
11/11
>
>|

“Prototipo de interfaz 2.a”

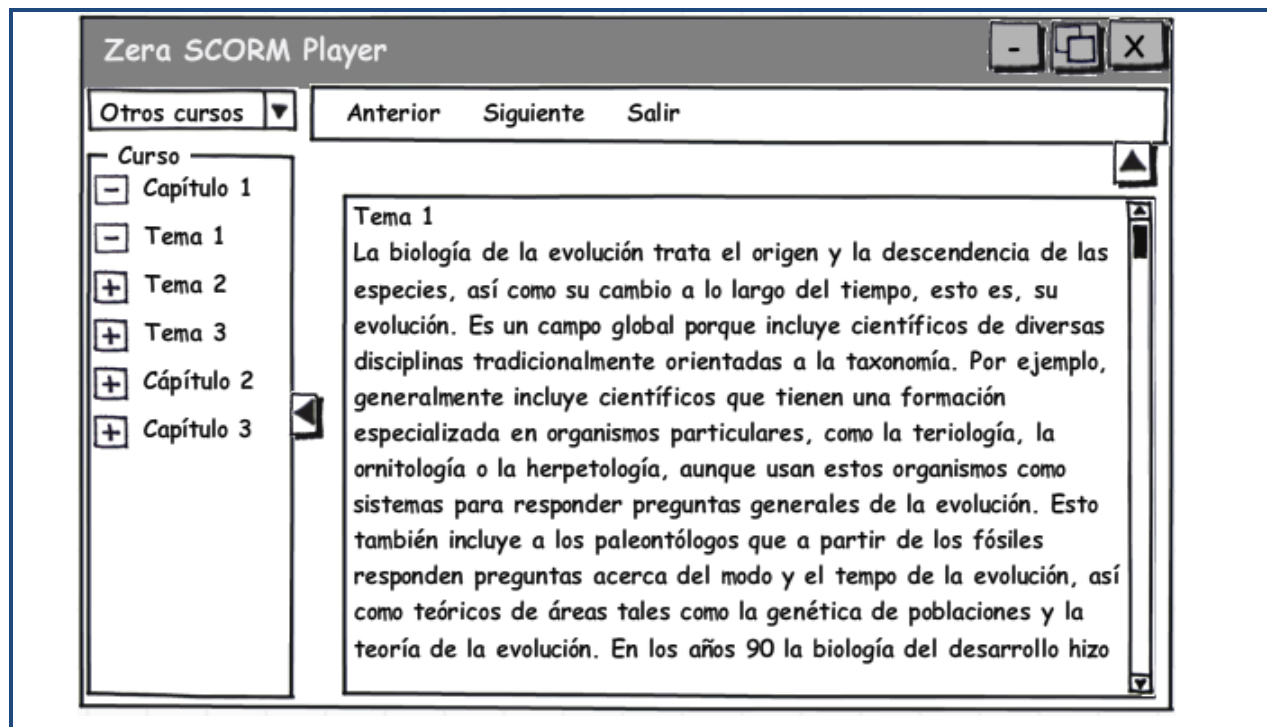


Tabla 2.3. Descripción del caso de uso del sistema, Visualizar curso

2.5 Conclusiones

La realización del modelo de dominio permitió la comprensión del funcionamiento del negocio y la captura de los requerimientos funcionales y no funcionales. Se logró tener un entendimiento de lo que realmente se desea implementar. A partir del análisis de la relación y complejidad de los requerimientos se identificaron los casos de uso y se estructuró el diagrama de casos de uso del sistema, este artefacto unido a la descripción de los casos de uso serán de gran aporte para las actividades de análisis y diseño.

Capítulo III: Análisis y Diseño

En el presente capítulo se profundizará en los casos de usos identificados, detallándolos de manera que permitan reflejar la vista interna del sistema descrita con el lenguaje de los desarrolladores. Los casos de uso serán mejor especificados y se establecerán las clases necesarias para llevar a cabo las funcionalidades que ellos contienen.

3.1 Modelo de Análisis

El modelo de análisis es un modelo conceptual, debido a que está compuesto por clases y relaciones, que en su conjunto permiten interpretar la parte del sistema modelada. Es utilizado fundamentalmente por los desarrolladores para comprender cómo debería darse forma al sistema, es decir, cómo debería ser diseñado e implementado. Sirve como una primera aproximación del diseño y su objetivo es comprender perfectamente los requisitos del software y no precisar cómo se implementará la solución. (37)

3.1.1 Clases del análisis

Las clases del análisis se centran en los requisitos funcionales y son evidentes en el dominio del problema porque representan conceptos y relaciones del dominio. RUP propone clasificar las clases en: interfaz, control y entidad.

- ✓ Las clases interfaz modelan la interacción entre el sistema y sus actores. Representan ventanas, formularios, comunicación con otros sistemas o dispositivos.
- ✓ Las clases entidad modelan la información que posee larga vida y que es a menudo persistente.
- ✓ Las clases control se encargan de coordinar la realización de uno o pocos casos de uso, coordinando las actividades de los objetos que implementan la funcionalidad del caso de uso.

3.1.2 Diagramas de Clases del Análisis

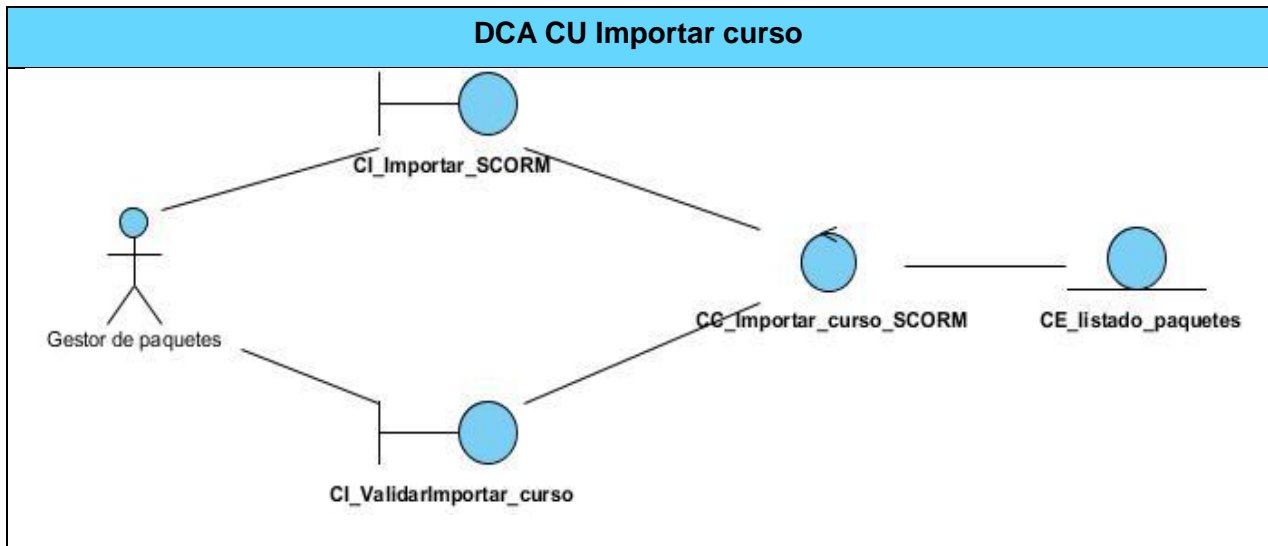


Figura 3.1. DCA CU Importar curso

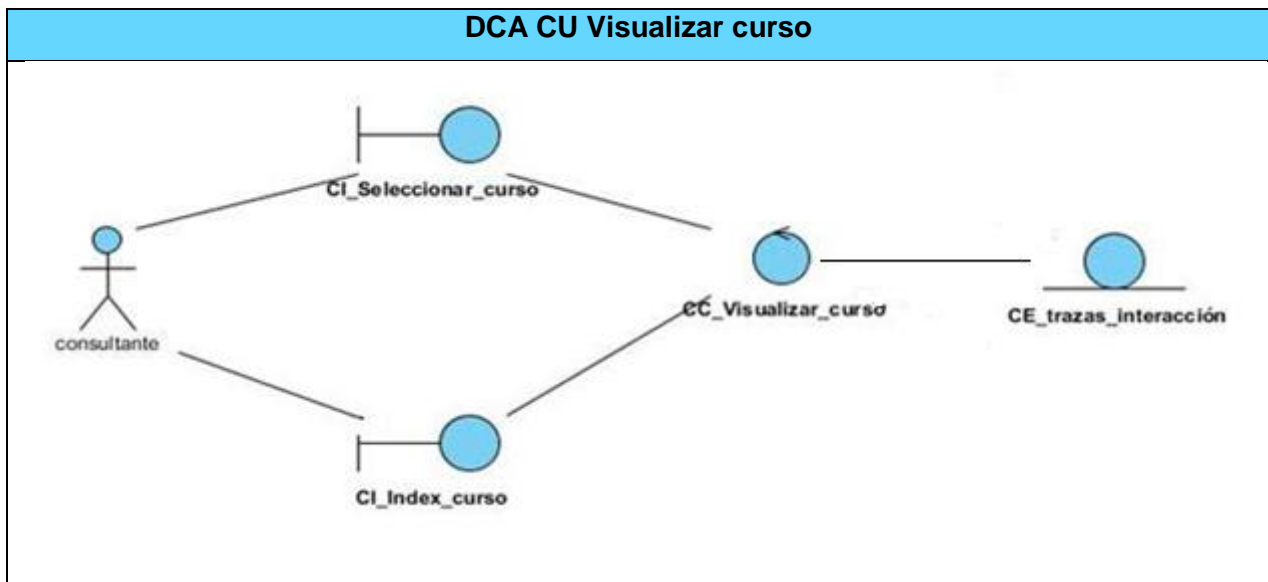


Figura 3.2. DCA CU Visualizar curso

3.1.3 Diagramas de Colaboración

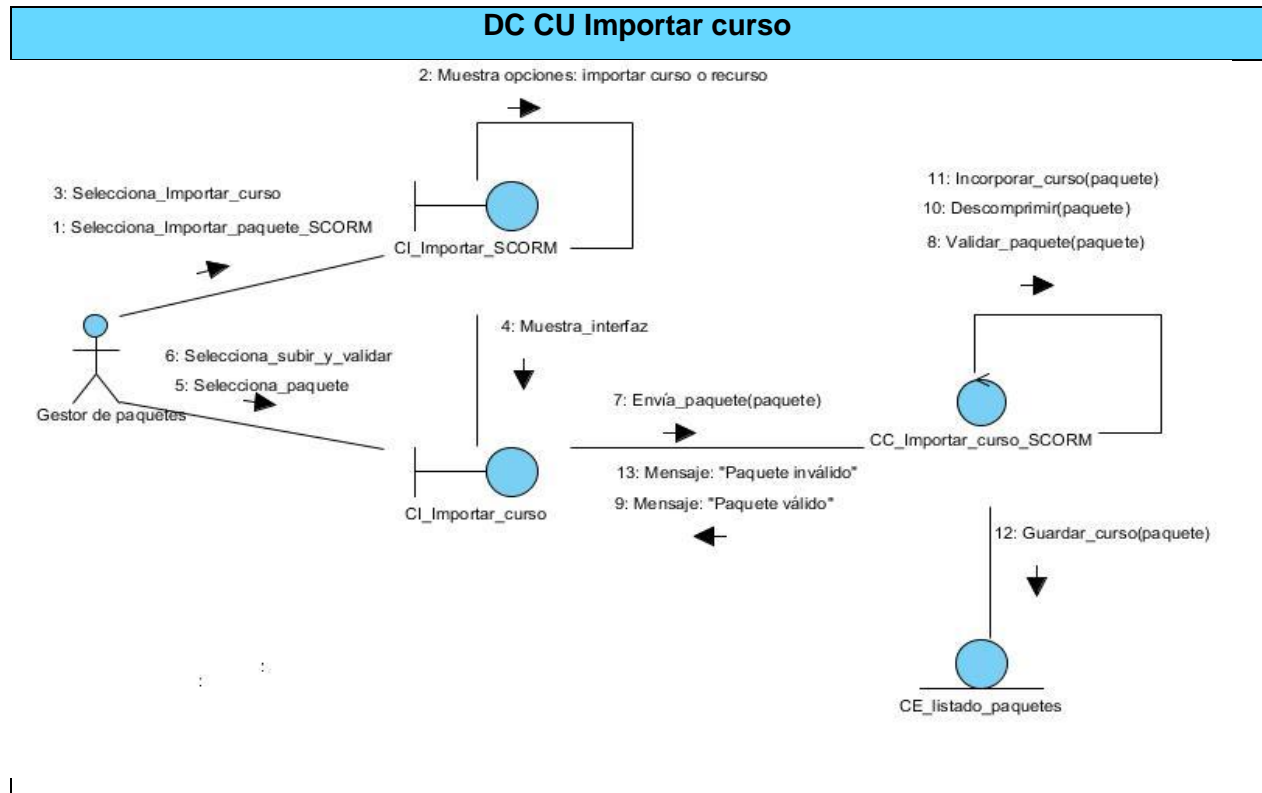


Figura 3.4. DC CU Importar curso

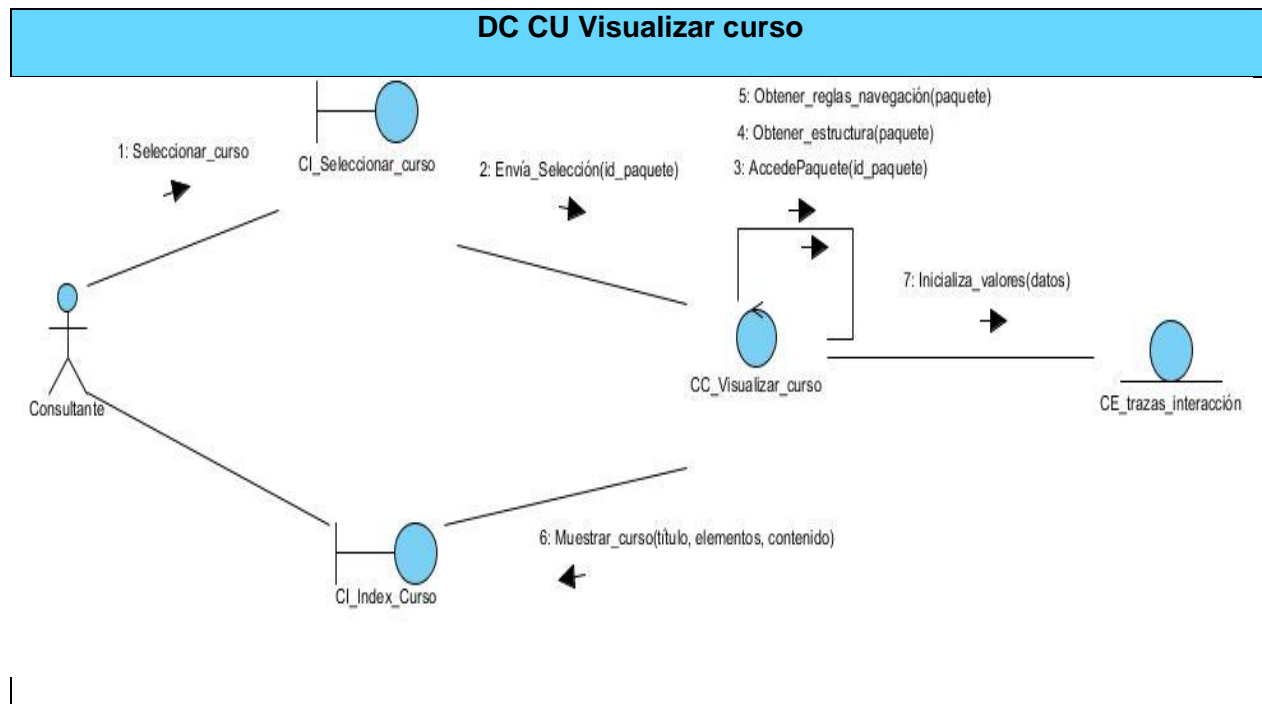


Figura 3.5. DC CU Visualizar curso

3.2 Paquetes del visor

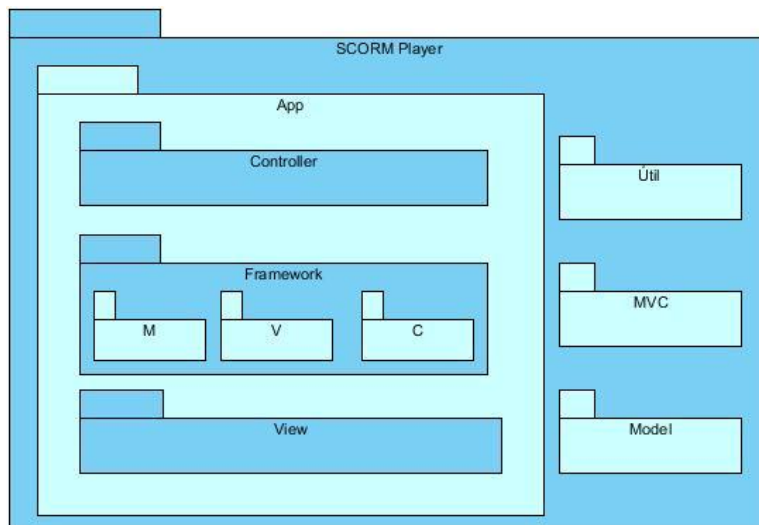


Figura 3.7. Diagrama de paquetes del visor

Descripción de los paquetes:

Controller: está compuesto por las clases controladoras.

View: contiene todas las interfaces que se utilizarán para elaborar la solución.

Framework: contiene las clases relacionada con el framework Symfony, separadas en tres paquetes que representan la forma en que el framework implementa el patrón arquitectónico Modelo Vista Controlador, en lo adelante MVC.

MVC: se representa como un paquete genérico.

Útil: contiene todas las clases auxiliares que van a ser utilizadas por las clases del paquete *Controller*, con el objetivo de complementar la implementación del negocio.

Model: contiene las entidades que conforman el modelo de datos.

3.3 Modelo de Diseño

El modelo de diseño es un modelo de objetos que describe la realización física de los casos de uso, centrándose en cómo los requisitos funcionales y no funcionales, junto con otras restricciones relacionadas con el entorno de implementación, tienen impacto en el sistema a considerar. Se utiliza como entrada fundamental a las actividades de implementación.

3.3.1 Arquitectura

“La arquitectura del software de un programa o sistema de cómputo es la estructura o las estructuras del sistema, que incluyen los componentes del software, las propiedades visibles externamente de esos componentes y las relaciones entre ellos.” (38)

Los patrones arquitectónicos transforman el diseño de la arquitectura, definen el comportamiento del sistema en determinadas circunstancias.

MVC es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario y la lógica de control en tres componentes distintos. En las aplicaciones web la vista la forman las páginas HTML, el controlador el código que obtiene datos dinámicos y genera el contenido HTML, y el modelo la información almacenada en una base de datos o en XML y las reglas de negocio que transforman esa información. (39)

Pasos dentro del patrón:

- ✓ El usuario realiza una acción en la interfaz.
- ✓ El controlador trata el evento de entrada.
- ✓ El controlador notifica al modelo la acción del usuario, lo que puede implicar un cambio del estado del modelo.
- ✓ Se genera una nueva vista. La vista toma los datos del modelo.
- ✓ La interfaz de usuario espera otra interacción del usuario, que comenzará otro nuevo ciclo.

Symfony implementa el patrón MVC distribuyendo los componentes de la siguiente manera: en el modelo, la abstracción de la base de datos y el acceso a los datos; en la vista las plantillas, los layouts¹⁶ y las vistas, y en el controlador las acciones y el controlador frontal. “El controlador frontal y el layout son comunes para todas las acciones de la aplicación. Se pueden tener varios controladores y varios layouts, pero se debe tener de manera obligatoria al menos uno. El controlador frontal es un componente que solo tiene código relativo al MVC, por lo que no es necesario crear uno, ya que Symfony lo genera de forma automática.” (40)

Symfony dentro de la arquitectura MVC implementa varias clases como son: (40)

- ✓ `sfController`: es la clase del controlador. Se encarga de decodificar la petición y transferirla a la acción correspondiente.
- ✓ `sfRequest`: almacena todos los elementos que forman la petición (parámetros, cookies y cabeceras).
- ✓ `sfResponse`: contiene las cabeceras de la respuesta y los contenidos. El contenido de este objeto se transforma en la respuesta HTML que se envía al usuario.
- ✓ El singleton de contexto: almacena una referencia a todos los objetos que forman el núcleo de Symfony y puede ser accedido desde cualquier punto de la aplicación.

¹⁶ Plantillas que definen los elementos comunes, como menús y cabecera.

Los patrones de diseño contribuyen a disminuir los esfuerzos de desarrollo y mantenimiento, mejoran la eficiencia y seguridad de los diseños. Además proporcionan un considerable ahorro en la inversión, a continuación se muestran algunos de ellos.

Singleton

Se encuentra dentro de los patrones de creación, su nombre en español significa instancia única. Está diseñado para garantizar que una clase tenga solo un objeto. Para ello la clase debe tener el constructor privado y además contar con un método estático para devolver la única instancia. Este patrón se evidencia a través de la clase *sfContext* de Symfony, la cual hace referencias a otras clases como *user*, *response* y *request*, y se pueden acceder a ellas a través del método *sfContext::getInstance()*.

Decorator

Pertenece al grupo de los patrones estructurales, su nombre en español significa decorador. La aplicación de este patrón se puede observar en el método “*decorator*” de la clase abstracta *sfView*, padre de todas las vistas, que contienen un decorador para permitir agregar funcionalidades dinámicamente. El archivo *layout.php* o plantilla global guarda el código HTML que es usual en todas las páginas del sistema, para no tener que repetirlo en cada página. El contenido de la plantilla se integra en el *layout*, o sea el *layout* decora la plantilla.

Se hace uso de los Patrones Generales de Software para Asignación de Responsabilidades, en inglés General Responsibility Assignment Software Patterns (GRASP) para la asignación de responsabilidades, a continuación se muestran los empleados en el presente trabajo.

Bajo acoplamiento

Las clases *sfScormActions*, *iResourcesActions* y *sfScormPlayerActions* heredan únicamente de *sfActions* para alcanzar un bajo acoplamiento de clases. Las clases que implementan la lógica del negocio y de acceso a datos se encuentran en el modelo, las cuales no tienen asociaciones con las de la vista o el controlador, lo que proporciona que la dependencia en este caso sea baja.

Alta cohesión

Un ejemplo del patrón es la clase *sfAction*, la cual está formada por varias funcionalidades que están estrechamente relacionadas, siendo la misma la responsable de definir las acciones para las plantillas y colaborar con otras para realizar diferentes operaciones, instanciar objetos y acceder a las propiedades.

Controlador

La clase *sfActions* manipula todas las peticiones web, siendo el único punto de entrada a toda la aplicación en un entorno determinado. Este patrón se evidencia en la clase *sfContex* y los *Actions*.

Experto

“Es uno de los patrones que más se utiliza cuando se trabaja con Symfony, con la inclusión de la librería Doctrine para mapear la base de datos. Symfony utiliza esta librería para realizar su capa de abstracción en el modelo, encapsular toda la lógica de los datos y generar las clases con todas las funcionalidades comunes de las entidades, las clases de abstracción de datos poseen un grupo de funcionalidades, que están relacionadas directamente con la entidad que representan y contienen la información necesaria de la tabla que representan.” (41)

Creador

“En la clase *Actions* se encuentran las acciones definidas para el sistema y se ejecutan en cada una de ellas. En dichas acciones se crean los objetos de las clases que representan las entidades.” (41)

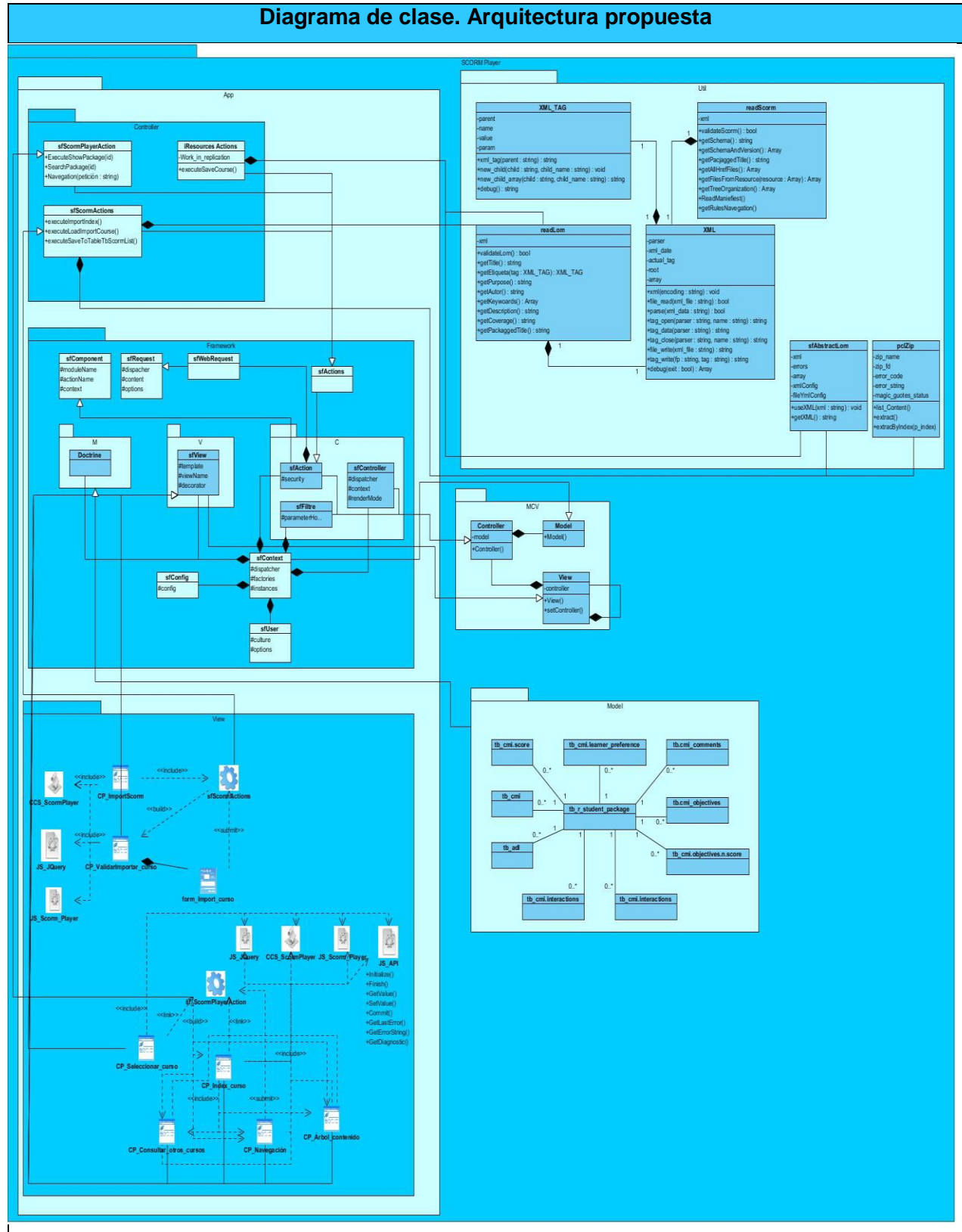


Figura 3.8 Diagrama de clases de la arquitectura propuesta

3.3.2 Diagramas de Clases del Diseño

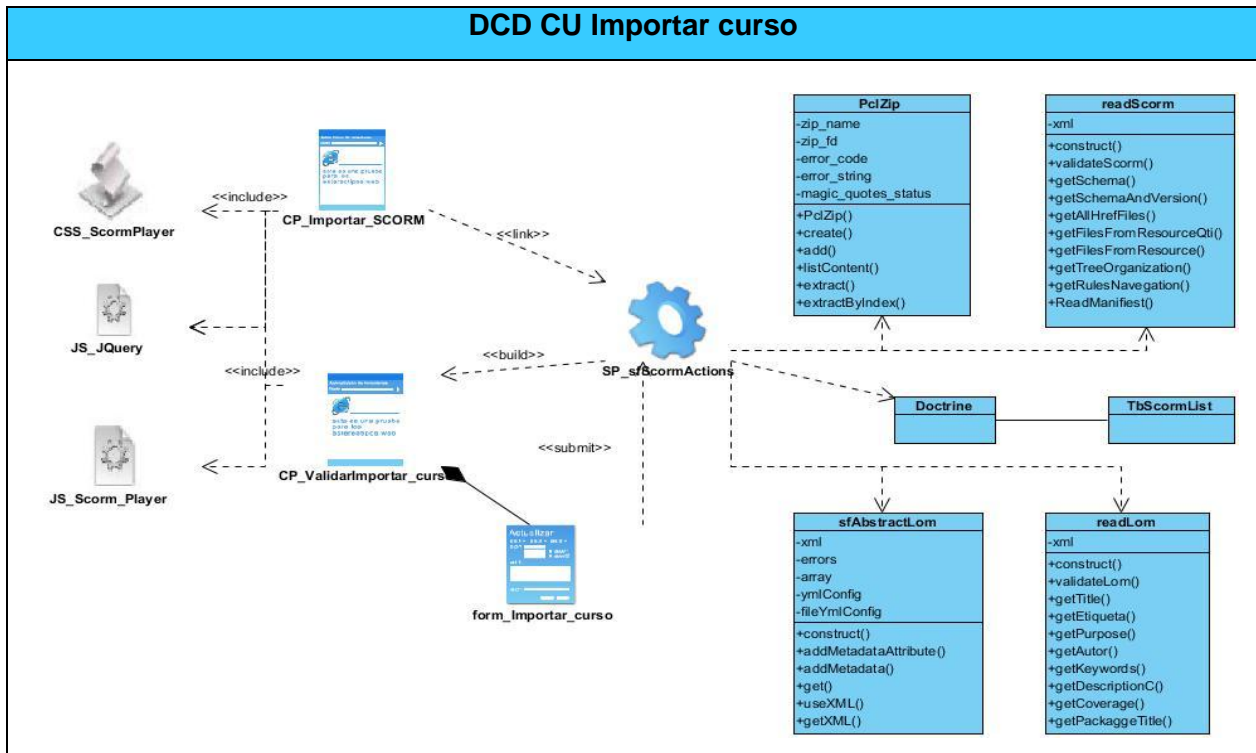


Figura 3.9. DCD CU Importar curso

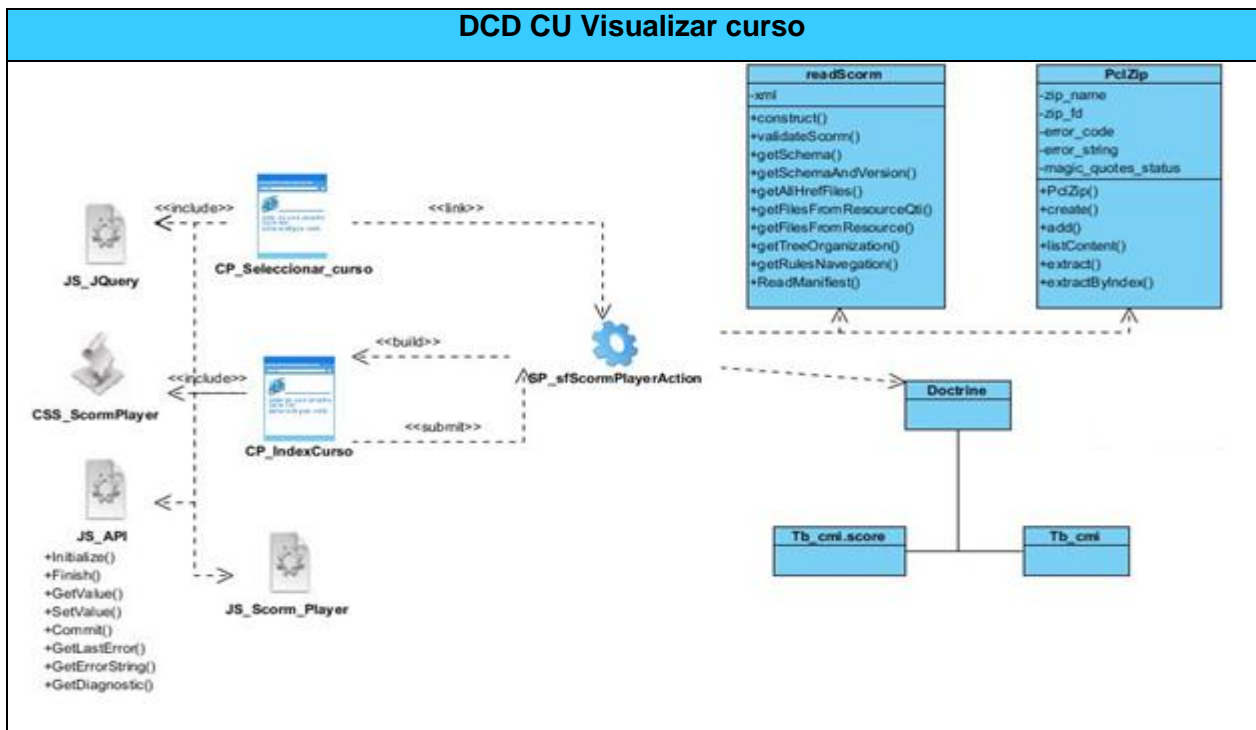


Figura 3.10. DCD CU Visualizar curso

3.3.3 Diagramas de Secuencia del Diseño

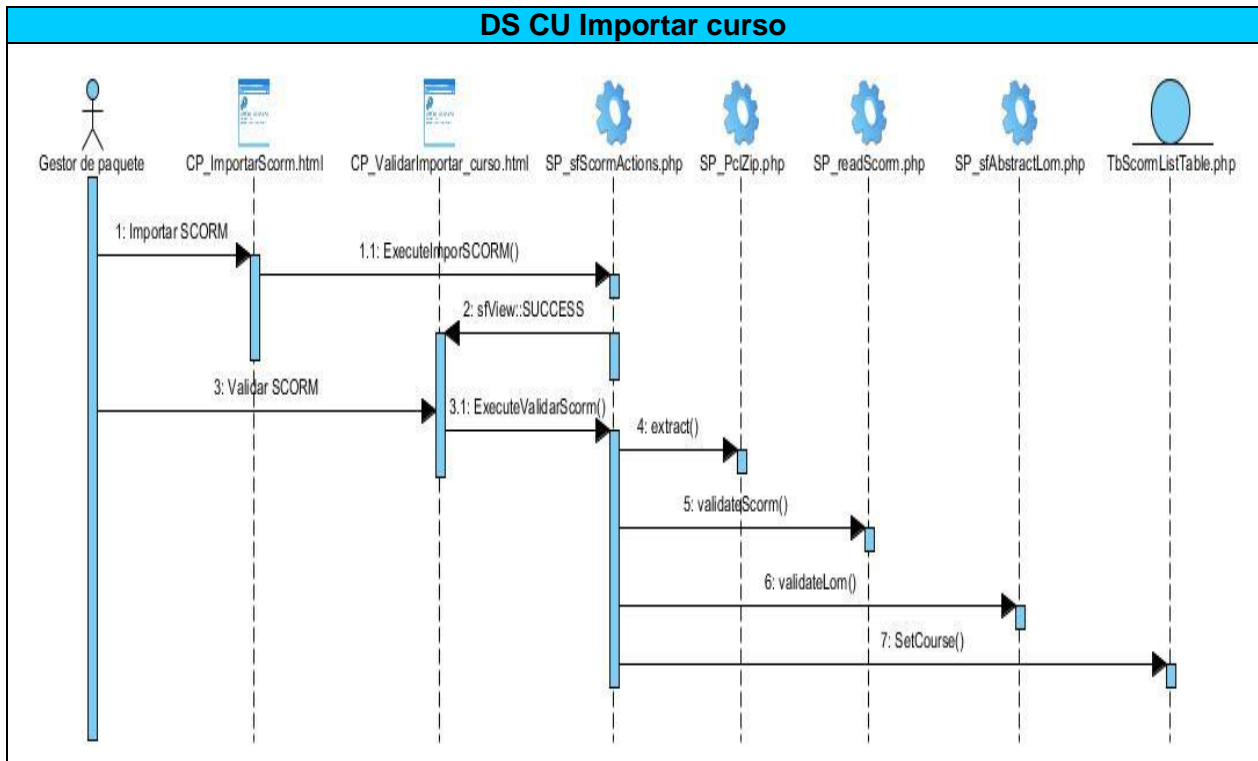


Figura 3.11. DS CU Importar curso

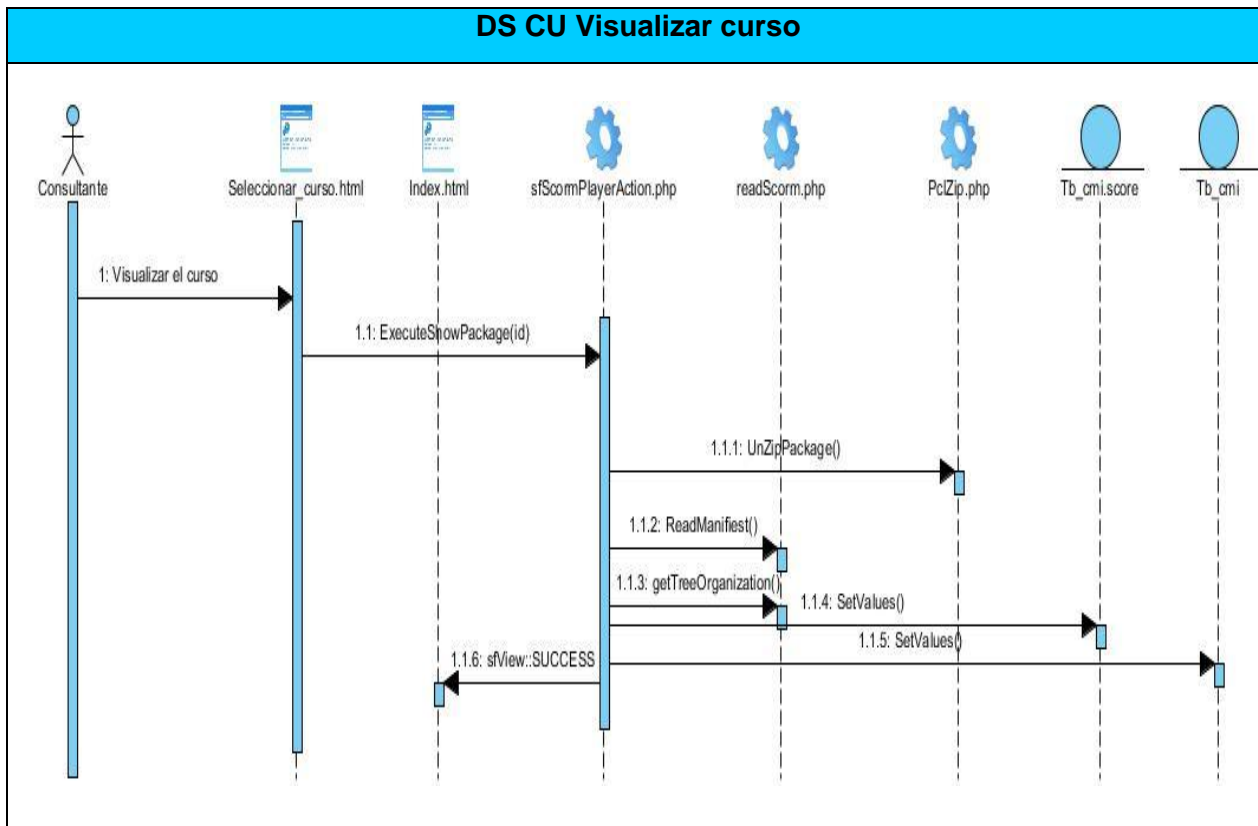


Figura 3.12. DS CU Visualizar curso

3.4 Diseño de la base de datos

Para la elaboración del modelo de datos que se muestra en la figura 3.12, los campos fueron tomados a partir del vocabulario que define el estándar SCORM 2004 3ra Edición para su entorno de ejecución. (42) Las trazas que se almacenarán por cada interacción del estudiante con el curso, dependerá de las funciones que le haya agregado su creador, donde especifica los datos que desea intercambiar con la plataforma.

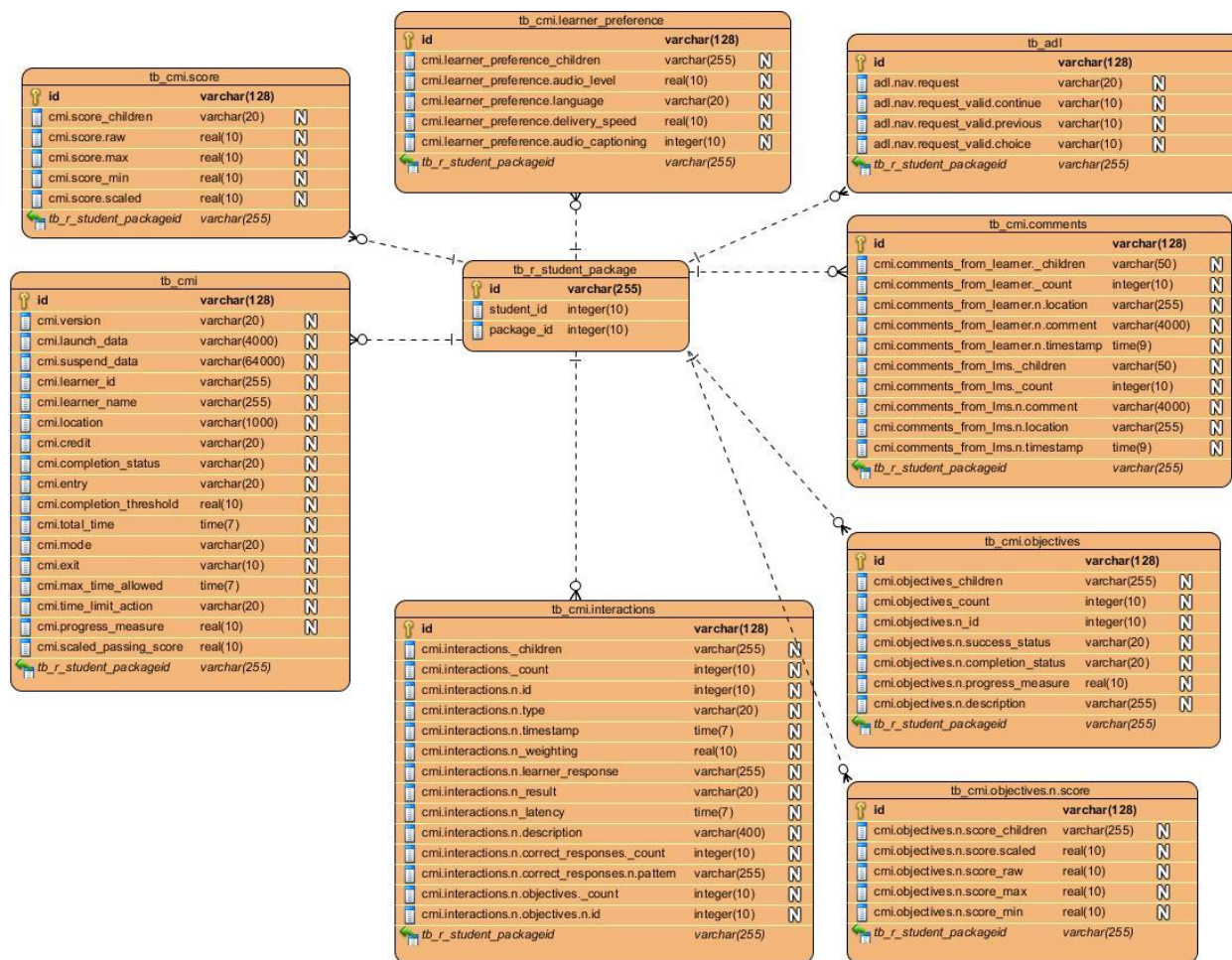


Figura 3.12 Modelo de datos

3.4.1 Descripción de las tablas del Modelo de Datos

tb_cmi
<p>Descripción: esta tabla registra los datos generales y más importantes del estudiante y su interacción con el curso.</p>

Atributo	Tipo	Descripción
cmi._version	varchar(20)	Versión del modelo de datos.
cmi.launch_data	varchar(4000)	Datos proporcionados al SCO después de su lanzamiento.
cmi.suspend_data	varchar(64000)	Para almacenar y recuperar datos entre las sesiones de aprendizaje.
cmi.learner_id	varchar(255)	Identificador del estudiante que interactúa con el curso.
cmi.learner_name	varchar(255)	Nombre del estudiante que visualiza el curso.
cmi.location	varchar(1000)	Punto de salida que tuvo el alumno la última vez que estuvo trabajando con el SCO.
cmi.credit	varchar(20)	Se usa para guardar los valores "credit" si el estudiante va hacer evaluado y "no-credit" en caso contrario.
cmi.completion_status	varchar(20)	Indica el estado de completitud del SCO: "Terminado", "Incompleto", "No se ha intentado" y "Desconocido".
cmi.entry	varchar(20)	Tiene el valor "ab-initio" si es la primera vez que se visita el SCO y "resume" o "" si se ha consultado anteriormente.
cmi.completion_threshold	real(10)	Se utiliza para determinar si el SCO debe determinarse como completado.
cmi.total_time	time(7)	Tiempo total del estudiante en el curso.
cmi.mode	varchar(20)	Indica el modo en que se presentará el SCO: <ul style="list-style-type: none"> • "browse": visita previa, no evaluar. • "normal": para evaluar. • "review: ya ha sido evaluado.

cmi.exit	varchar(10)	Indica cómo o por qué el estudiante salió del SCO: <ul style="list-style-type: none">• “time-out”: tiempo excesivo o tiempo máximo excedido.• “suspend”: salida para luego continuar.• “logout”: salida desde dentro del SCO.• “”: estado normal.
cmi.max_time_allowed	time(7)	Máximo tiempo permitido para utilizar un SCO.
cmi.time_limit_action	varchar(20)	Indica lo que el SCO debe hacer cuando se excede el tiempo máximo permitido.
cmi.progress.measure	real(10)	Medida de progreso del alumno en el SCO.
cmi.scaled_passing_score	real(10)	Calificación de aprobado.

Tabla 3.1. Descripción de la tabla tb_cmi

3.5 Conclusiones

Los artefactos generados en este capítulo permitieron transformar los requerimientos en diagramas, que indican cómo debe ser implementado el sistema. También como parte de las actividades de análisis y diseño se identificó el patrón arquitectónico y los patrones de diseño que se deben emplear para el desarrollo de las funcionalidades identificadas. Además se realizó el modelo de base datos, donde quedaron plasmadas todas las relaciones de las tablas necesarias para almacenar la interacción del SCO y el LMS.

Capítulo 4 Validación de la Solución Propuesta

Para lograr un producto exitoso se debe satisfacer las necesidades del cliente, para ello los requerimientos deben estar correctos y completos. La utilización de métricas en algunos de los artefactos generados permitirá identificar y erradicar errores en los requisitos y los casos de uso. Otro aspecto importante dentro de la investigación es lograr que esta sea validada por especialistas, por lo que se aplicará la técnica del criterio de expertos.

Para aplicar la técnica de las métricas de los requerimientos y los casos de uso fue necesario seleccionar a varios revisores, dos analistas y un programador.

Analistas: Ing. Yuleisy González Pérez e Ing. Irina Ivis Santiesteban Pérez.

Programador: Ing. Yerandy Manso Guerra.

4.1 Métricas de la Calidad de Especificación de los Requisitos

La métrica de la Calidad de Especificación de los Requisitos, (43) mide la especificidad de los requisitos, haciendo que la parte interesada pueda entenderlos de manera fácil y se puedan probar.

Para comenzar a utilizar la técnica de validación, se necesita conocer el número total de requisitos.

R_f : número de requisitos funcionales.

R_{nf} : número de requisitos no funcionales.

R_t : total de requisitos.

$$R_t = R_f + R_{nf}$$

$$R_t = 8 + 6 = 14$$

Especificidad

Para determinar la especificidad o falta de ambigüedad de los requisitos, se sugiere una métrica basada en la consistencia de la interpretación de los revisores de cada requisito. (44)

$$Q_1 = R_{ij} / R_t$$

Donde R_{ij} es el número de requisitos que todos los revisores interpretaron igual. Cuanto más cercano esté el valor de Q_1 (grado de especificación de los requisitos) a 1, menor será la ambigüedad de la especificación.

Primera revisión

$$Q_1 = 12/14$$

$Q_1 = 0.86$

Resultado: se detectó que el 14.3% de los requerimientos presentaban ambigüedades, por lo que se realizó una reelaboración de los requerimientos, con términos más entendibles por todos.

Segunda revisión

$Q_1 = 14/14$

$Q_1 = 1$

Resultado: no se detectaron ambigüedades.

Compleción

El resultado de esta métrica está siempre entre 0 y 1. El valor óptimo de esta métrica es el más cercano a 1 e indica un alto nivel de completitud en la definición de los requisitos. Este valor se calcula como se muestra a continuación:

$$Q_2 = R_c / (R_c + R_{pe})$$

Donde Q_2 es el grado de completitud de los requisitos, R_c es el número de requisitos completos y R_{pe} el número de requisitos pobremente especificados.

Primera revisión

$$Q_2 = R_c / (R_c + R_{pe})$$

$$Q_2 = 10 / (10 + 4)$$

$$Q_2 = 0.71$$

Resultado: se detectó que el 28.6% de las definiciones de los requisitos se encontraban pobremente especificadas, debido a esto fue necesario estudiar más a fondo los procesos de negocio para luego realizar una completa definición de los requerimientos. Después de esto se realizó una segunda revisión con resultados exitosos.

Corrección

El resultado de esta métrica está siempre entre 0 y 1. El valor óptimo de esta métrica es el más cercano a 1 e indica un alto nivel de corrección en la definición de los requisitos. Este valor se calcula como se muestra a continuación:

$$Q_3 = (R_c - R_i) / R_c$$

Donde Q_3 es el grado de validación de los requisitos, R_c número de requisitos validados como correctos y R_i número de requisitos validados como incorrectos.

Primera revisión

$$Q_3 = (14 - 2)/14$$

$$Q_3 = 0.86$$

Resultado: se detectó que el 14.3% de los requisitos no estaban definidos de forma correcta, debido a esto los dos requisitos incorrectos fueron nuevamente definidos.

Consistencia interna

El resultado de esta métrica está siempre entre 0 y 1. El valor óptimo la métrica es el más cercano a 1 y expresa que no existen subconjuntos de requisitos contradictorios. Este valor se calcula como se muestra a continuación:

$$Q_4 = (R_e - R_c)/R_e$$

Donde Q_4 es el grado de consistencia interna, R_e es el número de requisitos especificados y R_c número de requisitos en conflicto con otros requisitos.

$$Q_4 = (14 - 0)/14$$

$$Q_4 = 1$$

Resultado: no hay requisitos en conflictos con otros.

Estabilidad

Para medir la estabilidad de los requisitos, en el presente trabajo se aplicó la métrica propia para esto, la cual ofrece valores entre 0 y 1. El mejor valor es el más cercano a 1. La estabilidad de los requisitos se calcula de la siguiente forma:

$$E = (R_t - R_m)/R_t$$

Donde E es el valor de la estabilidad de los requisitos y R_m es el número de requisitos modificados.

Clasificación de la estabilidad

Clasificación	Intervalo
Alta	$0.90 \leq E \leq 1$
Media	$0.80 \leq E < 0.90$
Baja	$E < 0.80$

Tabla 4.1. Intervalo y Clasificación.

$$E = (14 - 2)/14$$

$$E = 0.86$$

Resultado: se detectó que el 14.3% de los requisitos han sido modificados, por lo que la estabilidad ha sido media.

4.2 Validación mediante prototipos de interfaz de usuario

“Los prototipos consisten en la creación de una maqueta o versión del producto final, se utilizan fundamentalmente para comprobar la corrección y completitud de la especificación de los requerimientos.” (45) Para llevar a cabo esta técnica se realizaron los siguientes pasos:

- ✓ Seleccionar las personas que validarán el prototipo: las personas encargadas de validar los prototipos fueron los profesores de los tribunales de los seminarios de tesis.
- ✓ Desarrollar escenarios de validación: se identificaron una serie de escenarios de manera que el usuario pueda observar todo lo que va pasando.
- ✓ Ejecutar los escenarios: el usuario ejecutó cada uno de los escenarios definidos.
- ✓ Documentar los problemas: se registraron todos los problemas encontrados.

La aplicación de la técnica permitió detectar errores en tres requerimientos, debido a que no se plasmaba lo que realmente se deseaba. También se eliminaron requerimientos que no se querían para el visor, como el de contar con una ayuda y permitir realizar búsquedas dentro del curso SCORM.

Los prototipos cuando estuvieron correctos, fueron anexados a la descripción de los casos de uso del sistema, con el objetivo de hacer más explícita su descripción.

4.3 Métricas de casos de uso

Para validar los casos de uso se realizó la técnica de las métricas, en la que se emplearán 4 atributos: (46)

Compleitud

Un caso de uso es completo si especifica todo lo que deben hacer el actor y el sistema para alcanzar el objetivo del caso de uso y si se consideran todas las respuestas del sistema a situaciones anormales. Para verificarlo se plantean las siguientes interrogantes:

- ✓ ¿Hay respuestas a todas las peticiones que el actor del caso de uso hace al sistema y viceversa?
- ✓ ¿Se contemplan todos los posibles escenarios para poder alcanzar el objetivo del caso de uso?
- ✓ ¿Se especifican todas las secuencias alternativas a la secuencia normal?
- ✓ ¿Se contemplan todas las posibles excepciones a la secuencia normal?

Comprensibilidad

Un caso de uso es comprensible si todos los tipos de lectores pueden entenderlo fácilmente con una mínima explicación del autor. Para verificarlo se plantean las siguientes interrogantes:

- ✓ ¿Es posible leer el caso de uso sin volver atrás en repetidas ocasiones?
- ✓ ¿Es difícil seguir la secuencia normal del caso de uso por la presencia de las relaciones include o extend?
- ✓ ¿Es difícil seguir la secuencia de pasos por la existencia de demasiados pasos alternativos?
- ✓ ¿Se han desglosado demasiado los pasos de algún actor o del sistema provocando que el caso de uso avance a un ritmo muy lento?
- ✓ ¿Aparecen pasos condicionales para expresar que el sistema comprueba una situación que permite al caso de uso continuar su realización?

Concisión

Un caso de uso es conciso si no incluye información innecesaria. Para verificarlo se plantean las siguientes interrogantes:

- ✓ ¿Podría el caso de uso ser expresado con menos palabras?
- ✓ ¿Existen elementos que se puede obviar o aparecen anotaciones innecesarias y que dificultan la lectura del caso de uso?
- ✓ ¿Aparecen demasiadas interacciones entre el actor principal del caso de uso y otros elementos del entorno?

No trivialidad

Un caso de uso es no trivial si su secuencia de pasos conduce al actor a conseguir el objetivo que persigue, la realización del caso de uso. Para verificarlo se plantean las siguientes interrogantes:

- ✓ ¿Expresa el nombre del caso de uso un objetivo de un usuario que el sistema debe implementar?
- ✓ ¿Conduce el caso de uso al actor a conseguir alguno de sus objetivos sin representar un conjunto de interacciones triviales?

Después de la aplicación de la técnica en dos revisiones se obtuvieron los siguientes resultados:

Primera revisión:

Análisis de la comprensibilidad: se encontró que el caso de uso Visualizar curso contenía información que dificultaba el entendimiento, por lo que surgió un error del 20%, lo que implica que el 80% de los casos de uso cumplen debidamente con el atributo.

Análisis de la concisión: se encontró que los casos de uso Acceder al árbol de contenido y Navegar en el curso presentaban información innecesaria, lo que representa que un 40% de las descripciones presentaban errores.

Segunda revisión:

Después de haber rectificados los errores encontrados en la primera revisión, se efectuó una segunda, de la cual se obtuvo que los casos de uso quedarán correctos al 100% en cuanto a los atributos completitud, comprensibilidad, concisión y no trivialidad.

La aplicación de la técnica de métricas de casos de uso permitió descubrir y corregir errores en cuanto a la comprensibilidad y la concisión de las especificaciones de los casos de uso.

4.4 Validación de la propuesta por el método de expertos

Un experto es un individuo con un conocimiento amplio en un área particular del conocimiento, capaz de ofrecer valoraciones conclusivas de un problema en cuestión. (47) Para evaluar la calidad y efectividad de la investigación fueron seleccionados un grupo de personas con estas capacidades, para emitir una valoración tendrán en cuenta el aporte, la novedad, los resultados teóricos y la posibilidad real de llevarlos a la práctica.

Para la aplicación del método se realizaron los siguientes pasos.

- 1- Selección de los expertos.

Para la selección de los expertos se tuvo en cuenta su experiencia en el desarrollo de software educativo, la efectividad de su labor como profesional y sus conocimientos de la metodología de la investigación y la Ingeniería de Software.

Listado de expertos:

ID	Expertos
E₁	Dpto. Producción De Herramientas Educativas. Ing. Jorge Luis Piña González
E₂	Dpto. Producción De Herramientas Educativas. Ing. Yuleisy González Pérez
E₃	Dpto. Programación e Ingeniería de Software. Ing. Rosalba Carralero Medina
E₄	Dpto. Programación e Ingeniería de Software. MSc. Ismael A. Nodarse Mora
E₅	Dpto. Ciencias Básicas. Ing. Bettys Leidy Arias Valdés

E₆	Dpto. Teleformación. Lic. Thaymi Mazorra Ofarrill
----------------------	--

Tabla 4.2. Identificadores y Expertos seleccionados.

Información de los expertos:

ID	Proyectos	Rol	Años de experiencia	Publicaciones científicas	Asignaturas
E₁	Alfaomega	Desarrollador	5	10	Introducción a las Tecnologías para la Formación y Framework de Desarrollo
E₂	Alfaomega	Analista	3	2	Ingeniería de Software
E₃	Multisaber	Desarrolladora	6	4	Ingeniería de Software
E₄	Multisaber	Jefe de módulo	5	11	Ingeniería de Software
E₅	Infodrez e Innovación Universitaria	Analista	3	3	Matemática
E₆	-	-	7	-	Ética Informática y Metodología de la Investigación Científica

Tabla 4.3. Información de los expertos.

2- Enviar la información a los expertos.

A cada uno de los expertos se le envió un resumen de la investigación para su estudio y evaluación de acuerdo a criterios que fueron seleccionados en conjunto.

Tabla de alternativas:

Indicadores	Alternativas
A₁	Nivel de calidad de la investigación.
A₂	Aportes científicos.
A₃	Novedad científica de la investigación.
A₄	Necesidad de uso del visor.
A₅	Nivel de comprensión de los artefactos generados.
A₆	Contribución a las actividades de implementación.
A₇	Contribución al proceso de enseñanza-aprendizaje.

A₈	Contribución al desarrollo de software educativo.
----------------------	---

Tabla 4.4. Indicadores y Alternativas para la evaluación.

La evaluación de las alternativas se realizó como se muestra en la siguiente tabla:

Valor	Peso
Ninguno	1
Poco	2
Medio	3
Moderado	4
Alto	5

Tabla 4.5. Peso y Valor.

Evaluación dada por los expertos:

Alternativas	E ₁	E ₂	E ₃	E ₄	E ₅	E ₆	E _a
A₁	4	4	5	4	5	5	4.5
A₂	4	3	5	4	4	5	4.2
A₃	4	4	5	2	4	5	4
A₄	5	4	5	5	5	5	4.8
A₅	5	5	4	3	5	5	4.5
A₆	5	5	5	3	5	5	4.7
A₇	5	4	4	4	4	5	4.3
A₈	5	4	5	5	4	5	4.7

Tabla 4.6. Evaluación de las alternativas.

E_a: evaluación promedio de las alternativas.

- 3- Una vez que se conoce la evaluación promedio de las alternativas (**E_a**), se puede calcular el índice de aceptación (**IA**) y ubicarlo en los rangos definidos, determinando así la probabilidad de éxito.

$$IA = \sum E_a / 40$$

$$IA = 35.7 / 40$$

$$IA = 0.89$$

Probabilidad de éxito	Rangos
Ninguno	IA < 0.3

Baja	$0.3 \geq IA > 0.5$
Media	$0.5 \geq IA > 0.7$
Alta	$0.7 > IA$

Tabla 4.7. Probabilidad de éxito

4- Resultados obtenidos.

La investigación cuenta con un nivel de calidad alta, los artefactos generados tienen buena comprensión y se destaca la necesidad del uso del visor para favorecer el proceso de enseñanza-aprendizaje.

De acuerdo al índice de aceptación calculado se demuestra que la propuesta tiene alta probabilidad de éxito.

4.5 Conclusiones

Las técnicas de validación aplicadas a los artefactos especificación de los requerimientos y casos de uso permitieron detectar y corregir problemas, favoreciendo así al buen desarrollo del resto de los artefactos. Se logró un mayor acercamiento y entendimiento entre los miembros del equipo de desarrollo, a partir de los prototipos realizados. La validación de la propuesta mostró excelentes resultados, de acuerdo a los resultados del método aplicado, que de forma general ratifican el nivel de calidad, funcionalidad y relevancia de la propuesta.

Conclusiones

El estudio bibliográfico referente a los visores de paquetes SCORM, permitió comprender su funcionamiento y la importancia de su empleo en las plataformas educativas. Además a partir de las soluciones similares, se seleccionaron las características más relevantes y se incorporaron a la presente solución.

El uso de la metodología RUP, permitió guiar el proceso y posibilitó la creación de artefactos fundamentales para sus tres primeros flujos de trabajo. Las herramientas empleadas ayudaron a realizar las interfaces de usuario, los modelos y diagramas útiles para la futura implementación de la propuesta de solución.

La definición y especificación de los requerimientos de forma completa y consistente, posibilitó la buena comprensión de las funcionalidades y la reducción de los cambios. Durante la realización del análisis y diseño del visor se realizaron los diagramas de clases del análisis y del diseño, y se definió la arquitectura a seguir para la posterior implementación.

Se comprobó la factibilidad del visor a partir de la validación de los requerimientos y los casos de uso mediante métricas. La validación de la propuesta por el método de expertos, dio como resultado un índice de aceptación de 0.89, para una probabilidad de éxito alta, demostrando así la calidad de la investigación y de los artefactos generados.

Recomendaciones

- ✓ Realizar la implementación del visor SCORM.
- ✓ Monitorizar las próximas versiones de SCORM, debido a que se le pueden añadir nuevos elementos al modelo de datos, que sean de gran aporte para el control y seguimiento del aprendizaje.
- ✓ Valorar la edición de los metadatos del paquete desde el propio visor, para el rol editor de la Plataforma Educativa Zera.

Referencias bibliográficas

1. **Marrero Zamora, Anibal.** Cuadernos de Educación y Desarrollo. [En línea] Mayo de 2011. [Citado el: 20 de Noviembre de 2011.] <http://www.eumed.net/rev/ced/27/amz2.htm>.
2. **Morgado, Erla Mariela Morales.** Dialnet. [En línea] Octubre de 2007. [Citado el: 7 de Diciembre de 2011.] <http://dialnet.unirioja.es/servlet/tesis?codigo=18457> .
3. **Ruipérez, Germán.** Sistemas e-learning. *Sistemas e-learning*. [En línea] 2003. [Citado el: 16 de Marzo de 2012.] <http://www.sistemaselearning.com/contenidos/definicion-de-elearning.asp>.
4. **Biscay, Carlos Emilio.** e-ABC. [En línea] [Citado el: 7 de Diciembre de 2011.] <http://www.e-abclearning.com/definicion-e-learning>.
5. **Manjón, Baltasar Fernández, Moreno Ger, Pablo y Sierra Rodríguez, José Luis.** Educación. [En línea] [Citado el: 8 de Diciembre de 2011.] <http://ares.cnice.mec.es/informes/16/contenido/47.htm>.
6. **Alfredo.** Aprendizaje a Distancia. [En línea] 25 de Abril de 2009. <http://aprendizajedistancia.blogspot.com/2009/04/que-es-un-lms.html>.
7. **Santini, Vicente de J. Bihouet.** Organismo de Certificación de Establecimientos TIF. [En línea] 2007. [Citado el: 20 de Mayo de 2012.] <http://www.ocetif.org/iso.html>.
8. **Pedruelo, Miguel Rebollo.** *Cuaderno de Investigación*. [En línea] Diciembre de 2004. [Citado el: 30 de Abril de 2012.] www.mrebollo.es/pubs/tesina.pdf.
9. R. [En línea] [Citado el: 2 de Junio de 2012.] http://descargas.mundo-r.com/ficha/1141648416060_es.html.
10. Argentina Warez. [En línea] 17 de Agosto de 2011. [Citado el: 1 de Junio de 2012.] <http://www.argentinawarez.com/programas-gratis/1542091-descarga-gratis-vlc-media-player-v1-2-0-nightly-portable.html>.
11. **González, Fabriciano.** Lecciones Ordenadores e Internet. [En línea] 14 de Mayo de 2009. [Citado el: 4 de Junio de 2012.] <http://lecciones.batiburrillo.net/2009/05/14/acceso-directo-al-visor-de-imagenes-y-fax-de-windows.htm>.

12. **Diéguez, Jorge.** raccoon. [En línea] 23 de Abril de 2012. [Citado el: 2012 de Abril de 25.] http://www.raccoon-learning.com/conocimiento_raccoon/la-norma-scorm-un-acercamiento-practico/.
13. **Lucía, Diario de Sara.** La tesis. [En línea] 21 de Febrero de 2005. [Citado el: 17 de Marzo de 2012.] <http://latesis.blogspot.com/2005/02/el-rte-de-scorm.html>.
14. **Peso, Daniel Hernández del.** AdictosAlTrabajo. *AdictosAlTrabajo*. [En línea] 8 de Abril de 2008. [Citado el: 13 de Marzo de 2012.] <http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=elearningScorm>.
15. agrega. [En línea] 13 de Abril de 2012. [Citado el: 2 de Junio de 2012.] <http://agrega.catedu.es/portada/noticias/Esta/disponible/en/la/web/del/CATEDU/el/visor/de/paquetes/SCORM,/Vi-Sco./9>.
16. **Victorhgtzr.** Blog sobre tecnología, e-learning, viajes y experiencias personales. [En línea] 3 de Febrero de 2009. [Citado el: 20 de Octubre de 2011.] <http://victorhgtzr.wordpress.com/2009/02/03/programas-para-crear-contenidos-scorm-12-6/>.
17. **Informática, Alumnos del Master de Formación del Profesorado en la especialidad de.** Instalación de LMS. [En línea] 2009. [Citado el: 4 de Junio de 2012.] joseperezlozano.com/lms/?page_id=178.
18. **ysalinas.** Cosolig comunidad de software libre. [En línea] 13 de Abril de 2012. [Citado el: 4 de Junio de 2012.] <http://www.cosolig.org/2012/04/13/blackboard-no-pudo-con-la-competencia-y-compra-moodle/>.
19. PR Newswire. [En línea] [Citado el: 5 de Junio de 2012.] <http://www.prnewswire.co.uk/news-releases/blackboard-supera-el-test-de-certificado-adlscorm-gracias-a-su-producto-content-player-building-block-155162835.html>.
20. **Pariante, Javier Prieto.** Scribd. [En línea] Junio de 2009. [Citado el: 5 de Junio de 2012.] <http://es.scribd.com/doc/34158569/Comparacion-entre-distintas-plataformas-e-learning-Moodle-Dokeos-Caroline-ATutor-y-EFront>.
21. **López, Nitza M. Hernández.** TECNOINFO. [En línea] 14 de Octubre de 2010. [Citado el: 8 de Diciembre de 2011.] <http://tecnoinfo6300.blogspot.com/2010/10/sakai-plataforma-educativa-de-codigo.html>.

22. **Blanco, Diego del y Carro, Miguel.** Sakai. [En línea] [Citado el: 25 de Octubre de 2011.] <http://webcache.googleusercontent.com/search?q=cache:Ai5QsmG5MfIJ:https://confluence.sakaiproject.org/download/attachments/67111992/Diego%2BDe%2BBlanco,%2BMiguel%2BCarro%2B-%2BHerramientas%2Bde%2Bautor%2Bpor%2BSCORM.pdf%3Fversion%3D1%26modificationDate%3D1>.
23. **Torre, Aníbal de la.** adelat. [En línea] 2006. [Citado el: 14 de Diciembre de 2011.] www.adelat.org/media/docum/moodle/docum/23_cap01.pdf.
24. Ecured. [En línea] [Citado el: 13 de Diciembre de 2011.] http://www.ecured.cu/index.php/Plataformas_Educativas#Plataforma_Moodle.
25. Blue Core Software. [En línea] [Citado el: 2 de Noviembre de 2011.] http://bluecoresoftware.es/mediapool/103/1033199/data/flex_scorm_player.pdf.
26. **library.blackboard.** [En línea] [Citado el: 15 de Diciembre de 2011.] http://library.blackboard.com/docs/r6/6_3/es_ES/admin/bbas_r6_3_admin/the_open_standards_content_player.htm.
27. Blackboard help. [En línea] [Citado el: 15 de Diciembre de 2011.] http://help.blackboard.com/instructor/Content/_instructor_content/instructor_content_build_content.htm.
28. Rustici Software. [En línea] [Citado el: 6 de Enero de 2012.] <http://scorm.com/es/scorm-resuelto/scorm-nube/>.
29. EcuRed. [En línea] [Citado el: 17 de Noviembre de 2011.] http://www.ecured.cu/index.php/Proceso_Unificado_de_Desarrollo.
30. scribd. [En línea] [Citado el: 10 de Noviembre de 2011.] <http://www.scribd.com/doc/57494332/10/%C2%BFQue-es-UML>.
31. **Zamitiz, Ing. Carlos Alberto Román.** [En línea] [Citado el: 10 de Noviembre de 2011.] <http://profesores.fi-b.unam.mx/carlos/aydoo/uml.html>.

32. **Sanchez, Rocio Santiago.** slideshare. [En línea] 13 de Febrero de 2009. [Citado el: 15 de Marzo de 2012.] <http://www.slideshare.net/2008PA2Info3/extensiones-uml-para-aplicaciones-web-roco-santiago>.
33. **Sub-Jefatura de Informática.** inei. [En línea] Noviembre de 1999. [Citado el: 11 de Noviembre de 2011.] <http://www.inei.gov.pe/biblioineipub/bancopub/Inf/Lib5103/Libro.pdf>.
34. **Valdez, Altamirano Alfonso.** [En línea] [Citado el: 11 de Noviembre de 2011.] http://docs.google.com/viewer?a=v&q=cache:vo4SGq-b8gkJ:ubicuos.com/wp-content/uploads/2009/05/comparativoides.pdf+visual+paradigm%2Bcaracteristicas&hl=es-419&gl=cu&pid=bl&srcid=ADGEESjOKndsLdCU-0Z17o_n848kp13V_LIheRI4A05G0DaKdlybcXHv2jMdSKfL6ftczr9XZstav_.
35. Tuna Web Design. [En línea] 23 de febrero de 2009. [Citado el: 15 de Noviembre de 2011.] <http://www.tunawebdesign.com/blog/2009/02/23/balsamic-mockups-herramienta-para-planear-sitios-web>.
36. EcuRed. [En línea] [Citado el: 24 de Febrero de 2012.] http://www.ecured.cu/index.php/Flujo_de_Trabajo_Requerimiento#Actores_del_Sistema.
37. **Jacobson, Ivar, Rumbaugh, James and Booch, Grady.** *El Proceso Unificado de Desarrollo de Software*. s.l. : Addison Wesley, 2000.
38. **Pressman, Roger S.** *Ingeniería de Software Un enfoque práctico*. [Libro] Sexta edición .
39. **Mestras, Juan Pavón.** *Estructura de las aplicaciones Orientadas a Objeto*.
40. **Eguíluz, Javier.** librosweb.es. [En línea] [Citado el: 15 de Marzo de 2012.] http://www.librosweb.es/symfony/capitulo2/el_patron_mvc.html.
41. **Hernández, Maylen Fiffe.** EcuRed. [En línea] 13 de Mayo de 2011. [Citado el: 4 de Abril de 2012.] http://www.ecured.cu/index.php?title=Patrones_en_Symfony&oldid=585394.
42. RUSTICI SOFTWARE. [En línea] 26 de Junio de 2009. [Citado el: 10 de Diciembre de 2011.] <http://scorm.com/es/scorm-explic%C3%B3/t%C3%A9cnica-scorm/en-tiempo-de-ejecuci%C3%B3n/ejecutar-referencia-de-tiempo/>.

43. **González, Danirys Ortega y Rodríguez Veitia, Reinaldo.** Dirección de Información. [En línea] 2010. [Citado el: 13 de Marzo de 2012.]
http://repositorio_institucional.uci.cu/jspui/bitstream/ident/TD_03395_10/1/TD_03395_10.pdf.
44. **Angelica.** [En línea] [Citado el: 20 de Abril de 2012.]
<http://delfin.mxl.uabc.mx/~angelica/Metricas.pdf> .
45. Ecured. [En línea] 24 de Mayo de 2012. [Citado el: 30 de Mayo de 2012.]
<http://www.ecured.cu/index.php/Prototipo>.
46. **B. Bernárdez, A. Durán, M. Toro.** aemes TI. [En línea] Agosto de 2004. [Citado el: 30 de Abril de 2012.]
<http://www.google.com.cu/url?sa=t&rct=j&q=+Hay+respuestas+a+todas+las+peticiones+que+el+actor+del+caso+de+uso&source=web&cd=1&ved=0CEYQFjAA&url=http%3A%2F%2Fwww.aemes.org%2Findex.php%2Frevista-de-procesos-y-metricas%2Fnumeros-publicados%2Fano-2004-volumen>. ISSN: 1698-2029.
47. **Oberto, César Prieto.** aporrea.org. [En línea] 10 de Agosto de 2009. [Citado el: 20 de Febrero de 2012.] <http://www.aporrea.org/actualidad/a84219.html>.
48. **ADL.** scribd. [En línea] 6 de Febrero de 2011. [Citado el: 13 de Marzo de 2012.]
<http://es.scribd.com/doc/56971827/12/Manual-introductorio-de-SCORM-2004>.
49. adlnet. [En línea] [Citado el: 10 de Diciembre de 2011.] <http://www.adlnet.org/>.

Glosario de Términos

API: mecanismo de comunicación para informarle al LMS el estado de un objeto de contenido, por ejemplo, inicializado, concluido o en error, y utilizado para obtener y configurar los datos como la calificación y los límites de tiempo entre el Sistema de Administración de Aprendizaje y el SCO. (48)

Árbol de contenido: estructura con que fue concebido el curso.

El Ambiente de Ejecución de SCORM: ofrece un medio para la interoperabilidad entre el contenido de aprendizaje basado en el SCO y los Sistemas de Administración de Aprendizaje. (48)

Framework: esqueleto de un programa de software que sirve como patrón para el desarrollo o implementación de una aplicación.

Interoperabilidad: capacidad para ser utilizado el contenido en cualquier plataforma sin importar cómo fueron creados.

Modelo de Datos de Ambiente de Ejecución de SCORM: conjunto de elementos de datos usados para definir la información que se comunica. En su forma más básica, se refiere a elementos que tanto el Sistema de Administración de Aprendizaje como el SCO deben conocer. El Sistema de Administración de Aprendizaje debe mantener el estado de los elementos de datos requeridos a lo largo de las sesiones, y el contenido de aprendizaje debe utilizar solo los elementos de datos predefinidos si se da la reutilización en múltiples sistemas. (48)

XML: lenguaje de etiquetado simple que se encarga de describir datos.