



“Mecanismo de integración entre ATcnea- Plataforma Educativa ZERA 2.0”

**Trabajo de Diploma para optar por el título de Ingeniero en
Ciencias Informáticas.**



Autores:

Ronnay Castillo Andino

Frank Lorenzo Morfa Valdes

Tutores:

Dra.C. Roxana Cañizares González

Ing. Agustín Castillo Cordero

Ing. Yunier Broche Guevara

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

FACULTAD 4

LA HABANA, JUNIO DE 2017

“Año 59 de la Revolución”



“Ciencia es el conjunto de conocimiento humano aplicables a un orden de objetos, íntima y particularmente relacionados entre sí.”

DECLARACIÓN DE AUTORÍA

Declaramos ser los autores de la presente tesis y autorizamos a la Universidad de las Ciencias Informáticas para que haga uso de la misma, como estime pertinente, cediéndole de esta forma los derechos patrimoniales de la misma, con carácter exclusivo.

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

Frank Lorenzo Morfa Valdes

Ronnay Castillo Andino

Firma del Autor

Firma del Autor

Dra.C. Roxana Cañizares González

Ing. Agustín Castillo Cordero

Firma del Tutor

Firma del Tutor

Ing. Yunier Broche Guevara

Firma del Tutor

Agradezco a mis familiares y amigos, por haber hecho realidad este sueño. Gracias por acompañarme en esta ardua tarea y agradecido por el apoyo brindado. Agradecer a esos amigos que me brindaron su ayuda en todo momento y ante cualquier circunstancia. Gracias a mi compañero Frank por el apoyo en el desarrollo de este trabajo, por la ayuda y el esfuerzo realizado a lo largo del ciclo del desarrollo. También quiero agradecer a mi madre, mis abuelos, mi novia y su familia que me apoyaron todos estos años. Agradecer a Armando Afa quien me enseñó el camino para llegar a donde uno quiere llegar, gracias por los conocimientos transmitidos.

Ronnay

En primer lugar quiero darle gracias a Dios por mantenerse a mi lado en todo momento. Doy gracias a mi mamá porque todo esto ha sido por y para ella, porque ha vivido tristezas y alegrías conmigo a lo largo de estos años, y al final hemos llegado a la meta juntos. Agradezco también a mis dos padres Lorenzo y Marcos, por todo. A mi hermano y mi hermana ya que cada detalle cuenta. A mi novia Gabriela que me ha apoyado en todo momento, porque me ha comprendido y entendido a lo largo de este proceso. A toda mi familia, desde el primero hasta el último, porque de una forma u otra me han apoyado y ayudado. También doy gracias a cada uno de mis amigos, desde la infancia hasta la actualidad, de cada uno he aprendido algo. Quiero dar gracias a mi compañero Ronnay por cada segundo de apoyo, de trabajo, de esfuerzo, de sacrificio, por ser tan buen compañero de tesis y el amigo en el que se ha convertido. Gracias a mis tutores por cada ayuda que nos brindaron. Y doy gracias a cada una de las personas que de una forma u otra tuvo que ver con el desarrollo de este trabajo.

Frank

RESUMEN

Las Tecnologías de la Información y las Comunicaciones constituyen un pilar básico en el desarrollo de la sociedad, donde la educación no se queda al margen de los nuevos cambios. El Centro de Tecnologías para la Formación (FORTES) cuenta con productos como la Plataforma Educativa ZERA 2.0 y ATcnea que evidencian este desarrollo en la educación. En la actualidad, la reutilización y la integración de contenidos docentes es una de las prioridades en el mundo del *e-learning*. Actualmente no existe comunicación entre la Plataforma Educativa ZERA 2.0 y ATcnea por lo que no se permite el intercambio y accesibilidad a recursos, cuestionarios y evaluaciones entre ambos sistemas como alternativa para dar seguimiento al estudiante, así como la reutilización de contenidos a través de un acceso único. Por ello se decide desarrollar un mecanismo de integración basado en el uso de servicios web que permita la interoperabilidad y reutilización entre ATcnea y la Plataforma Educativa ZERA 2.0. Se seleccionó la metodología AUP-UCI para el desarrollo del *software*. Como lenguajes de programación están PHP 7 y Java, y se cuenta con Symfony 2.7 como *framework* para PHP, JavaFX como *framework* para interfaces en Java y NetBeans 8.0 como IDE de desarrollo. El servidor web es NGINX, y los gestores de bases de datos son HSQLDB y PostgreSQL 9.5 con PgAdmin III para la administración en PostgreSQL 9.5. El lenguaje de modelado es UML 2.0 y la herramienta UML CASE empleada es Visual Paradigm V8.0.

Palabras claves: acceso único, cuestionarios, evaluaciones, integración, interoperabilidad, mecanismo, recursos, reutilización, servicios web.

Índice

Introducción 1

Capítulo 1 Fundamentación Teórica 7

1.1 Plataformas de gestión del aprendizaje 7

1.2 Interoperabilidad entre sistemas 7

1.3 Servicios web 8

 1.3.1 SOAP (*Simple Object Access Protocol*) 10

 1.3.2 REST (*Representational State Transfer*) 11

1.4 Estándares y especificaciones para e-learning 16

 1.4.1 IMS Global Consortium Inc. 18

1.5 Estudio comparativo de herramientas con soluciones similares 21

 1.5.1 Docebo LMS (33): 21

 1.5.2 Integración Joomla y Moodle (34): 21

 1.5.3 Interoperabilidad entre el LMS Moodle y las aplicaciones educativas de propósito específico (11): 22

1.6 Herramientas y tecnologías 23

 1.6.1 Librería FosRestBundle 24

 1.6.2 Librería okhttp 3.6.0 24

 1.6.3 Okio-1.11.0 25

 1.6.4 XML (*Extensible Markup Language*) 27

 1.6.5 JSON (*JavaScript Object Notation*) 28

 1.6.6 Comparación de JSON con XML 28

1.7 Contextualización del sistema 29

1.8 Conclusiones del capítulo 30

Capítulo 2 Análisis y Diseño 31

2.1 Modelo de dominio 31

 2.1.1 *Conceptos del dominio* 31

2.2 Diagrama de clases del modelo de dominio 31

2.3 Propuesta de solución 32

2.4 Requerimientos del sistema 33

2.5 Requisitos funcionales 33

2.6	Requisitos no funcionales.....	45
2.7	Historias de usuario.....	45
2.8	Patrones de diseño	47
2.8.1	<i>Patrones GRASP</i>	47
2.8.2	<i>Patrones GoF</i>	48
2.9	Modelo de diseño.....	49
2.9.1	<i>Diagramas de clases del diseño</i>	49
2.9.2	<i>Diagramas de secuencia del diseño</i>	49
2.10.	Diagrama de despliegue.....	52
2.11	Modelo de datos.....	52
2.11.1	<i>Descripción de las tablas de la Base de Datos</i>	53
2.12	Conclusiones del capítulo.....	54
Capítulo 3	Implementación y Pruebas	55
3.1	Modelo de implementación.....	55
3.1.1	<i>Diagrama de componentes</i>	55
3.2	Estándares de codificación.....	56
3.3	Pruebas de software	59
3.3.1	<i>Niveles de prueba</i>	59
3.3.2	<i>Métodos de prueba</i>	60
3.4	Diseño de casos de prueba.....	60
3.5	Resultados obtenidos.....	61
3.5.1	<i>Pruebas de integración</i>	61
3.5.2	<i>Resultados de las pruebas de caja negra</i>	63
3.5.3	<i>Pruebas de aceptación</i>	66
3.6	Conclusiones del capítulo.....	66
Conclusiones generales	67
Recomendaciones	68
Referencias bibliográficas	69

INTRODUCCIÓN

Con el desarrollo tecnológico de los últimos tiempos, las Tecnologías de la Información y las Comunicaciones (TIC) han llegado a ser uno de los pilares básicos en el desarrollo de la sociedad, donde la educación no se ha quedado al margen de los nuevos cambios. Estas brindan nuevas posibilidades de instrumentación de los conocimientos que las formas tradicionales de aprendizaje no pueden cubrir. Además, diversifican el conocimiento con el uso de herramientas telemáticas y de teleformación como las enciclopedias multimedia, los videos, el software educativo, la realidad virtual entre otras, que elevan la productividad y la flexibilidad del proceso de enseñanza-aprendizaje con lo que facilita la tarea de difundir, transmitir y crear conocimientos.

El uso eficiente de las TIC ha elevado el nivel de desarrollo y desempeño en la enseñanza de los estudiantes. Estas permiten incrementar el desarrollo íntegro de los conocimientos con el uso de objetos de aprendizaje, videoconferencias y la vinculación a proyectos productivos, lo que permite desarrollar habilidades y un mejor desempeño. Estas tecnologías contribuyen a la preparación y juegan un papel fundamental en el desarrollo de la educación cubana. El uso de tecnologías en la educación tiene la finalidad de desarrollar las capacidades cognitivas del individuo. Se enfoca en el trabajo colaborativo y en red para el desarrollo de la participación y atiende la demanda de conocimiento de los recursos informáticos disponibles.(1)

La Universidad de las Ciencias Informáticas (UCI) fue creada en el 2002, con la necesidad de formar jóvenes profesionales, con el objetivo de informatizar el país y desarrollar la industria cubana del software. En la actualidad la UCI está comprometida con el propósito de llamar la atención sobre las tendencias pedagógicas en la virtualidad, la convergencia tecnológica en la educación, la educación abierta, la analítica del aprendizaje en red, así como la producción de recursos para el aprendizaje. Frente a este propósito sobresale la interrelación, cada vez más marcada, entre educación, conocimiento, ciencia, tecnología e innovación. Una cuestión sumamente apreciable a la hora de pensar los esquemas que deben describir el camino del estudiante en el proceso de aprendizaje que transitará. Estudios desde la universidad cubana señalan que “las TIC propician la construcción social del conocimiento y nuevas formas de trabajo colaborativo en el proceso de enseñanza-aprendizaje, por lo que es incuestionable el cambio en la concepción del mismo hacia un nuevo modelo en los proyectos educativos”.(2)

Las plataformas educativas constituyen un conjunto de herramientas agrupadas y optimizadas con fines educativos. El uso de las mismas permite la gestión de cursos en línea y contribuyen en la evolución de los procesos de enseñanza aprendizaje en los estudiantes. Se entiende por

plataforma educativa como un sitio en la Web, que permite a un profesor contar con un espacio virtual en Internet donde sea capaz de colocar todos los materiales de su curso, enlazar otros, incluir foros, wikis, recibir tareas de sus alumnos, desarrollar test, promover debates, chats, obtener estadísticas de evaluación y uso -entre otros recursos que crea necesarios incluir en su curso- a partir de un diseño previo que le permita establecer actividades de aprendizaje y que ayude a sus estudiantes a lograr los objetivos planteados.(3)

Además, están las plataformas de mayor complejidad que pretenden cubrir todas las necesidades de los usuarios, denominados Entornos virtuales o Sistemas para la Gestión de Aprendizaje (LMS, siglas en inglés de *Learning Management System*). Un LMS es un sistema *online* que permite administrar, distribuir, monitorear, evaluar y apoyar las diferentes actividades de un proceso de aprendizaje. Estos sistemas pueden utilizarse como núcleo del aprendizaje a distancia o como un complemento del aprendizaje presencial, una de las razones por lo que son utilizados en varias instituciones de la educación superior.(4)

Otros sistemas existentes son las aulas tecnológicas, estas son espacios de enseñanza donde los usuarios (docentes y alumnos) tienen una relación participativa utilizando medios tecnológicos, los cuales son amigables y de fácil interacción. Un aula tecnológica facilita el acceso a contenidos, recursos, materiales, evaluaciones y ejercicios prácticos a través de implementos tecnológicos. Estas aulas no son más que el resultado de la transición de las aulas tradicionales a las aulas basadas en la tecnología lo cual hace un traslado de un modelo de enseñanza autoritario a uno más colaborativo. Las aulas tecnológicas realizan un uso más amplio de las TIC y matizan la correlación entre la tecnología y los nuevos diseños en los procesos de enseñanza-aprendizaje, ya que el empleo de estas tecnologías apoya el proceso de cambio necesario para el uso eficaz de las nuevas capacidades de enseñanza que se despliegan en el mundo. (5)

El informe Horizon 2017 sobre tendencias de base tecnológica en la innovación educativa, analiza los actuales proyectos educativos, analiza los retos a los que se enfrenta la comunidad educativa y descubre las tendencias metodológicas y los cambios que buscan provocar a 5 años vista. Una de las propuestas de este informe es rediseñar los espacios de aprendizaje que no es más que con el provecho de los avances en tecnologías multimedia e inalámbricas, crear con ellas espacios más “inteligentes”, motivadores y que permitan un aprendizaje más interactivo, colaborativo y motivador. Otra propuesta de este informe a tener en consideración son los LMS de nueva generación, donde los sistemas de gestión del aprendizaje (por sus siglas en inglés – LMS), también seguirán en evolución a un ritmo vertiginoso. Ya no serán solo sitios web donde colgar información estática de una clase o un curso. Sino que tienden a convertirse en espacios virtuales

totalmente personalizables, interoperables con otros sistemas, y más enfocados en el aprendizaje en sí y no solo en la administración del contenido.(6)

Con el análisis de varias fuentes de referencias (7) (8) (9) (10), se define interoperabilidad en este trabajo como la capacidad de diferentes productos o sistemas para el intercambio de datos con el fin de intercambiar información de una manera útil y significativa.

En la actualidad, la reutilización y la integración de contenidos docentes es una de las prioridades en el mundo del E-Learning. De poco sirve un objeto de aprendizaje con un alto nivel de calidad, si solo es accesible por unos cuantos usuarios de una determinada plataforma. Las instituciones educativas requieren de mecanismos de interoperabilidad, ya que no tiene sentido invertir gran cantidad de tiempo y trabajo en la construcción de un recurso para mantenerlo aislado en un mundo cada vez más interconectado y globalizado. Moodle es un ejemplo de esto ya que cuenta con un recurso denominado "*external tool*", un componente que crea un vínculo entre el LMS y las herramientas externas de aprendizaje basadas en web que sean compatibles con cierto estándar de interoperabilidad. Su objetivo es la integración de cualquier LMS con toda aplicación de aprendizaje desarrollada por terceros.(11)

Uno de los métodos para realizar la interoperabilidad entre plataformas educativas y aulas tecnológicas es mediante el uso de la Arquitectura Orientada a Servicios (SOA, siglas del inglés *Service Oriented Architecture*), es un modelo de arquitectura basado en el paradigma de diseño de orientación a servicios. SOA apuesta por los servicios como base de la arquitectura. Los servicios son unidades de lógica con diferentes características en común como: la reutilización, capacidad de composición, estandarización, abstracción, autonomía, etc...(12)

El Centro de Tecnologías para la Formación (FORTES), perteneciente a la facultad 4 de la Universidad de las Ciencias Informáticas (UCI), cuenta con el desarrollo de varios productos como son el aula tecnológica ATcnea y la Plataforma Educativa ZERA 2.0.

ATcnea es el componente fundamental del aula, proporciona una solución para la creación de un aula interactiva compuesta por una pizarra inteligente, una laptop para el profesor y una tableta por cada estudiante. ATcnea garantiza la interacción entre las tabletas y la laptop a través de la conexión inalámbrica. Esta aula tecnológica brinda a los educadores la posibilidad de ser más creativos e innovadores, capaces de implementar nuevos recursos didácticos, así como nuevas estrategias de aprendizaje y la aplicación de innovadores modelos didácticos y pedagógicos. A su vez le da la posibilidad al estudiante de formar parte de la clase de una forma más dinámica a través del uso de esas tecnologías y en estrecha correlación conjunta con todos los demás estudiantes y con el profesor a través de ATcnea.

La Plataforma Educativa ZERA 2.0 (ZERA 2.0) es un sistema para la gestión académica, el aprendizaje y la gestión de cursos en línea. En ZERA 2.0 los cursos creados constituyen un soporte informático para el proceso docente con lo que cubren completamente los programas de estudio de cada asignatura en los diferentes niveles educacionales. ZERA 2.0 cuenta con cursos definidos por asignaturas que contienen recursos tales como ficheros, documentos, fotos, videos y cuestionarios. Los contenidos educativos que se muestran son asesorados por profesores y especialistas del Ministerio de Educación de Cuba. En su diseño pedagógico, ZERA 2.0 implementa un conjunto de requisitos que permite al profesor analizar y personalizar los contenidos al ritmo y estilo de aprendizaje del alumno en forma individual o grupal.(13)

En la actualidad no existe una comunicación entre la Plataforma Educativa ZERA 2.0 y ATcnea lo que provoca las siguientes limitaciones:

1. Se duplican los esfuerzos en el diseño de las clases en ATcnea por no contar con acceso a la nube donde se comparta recursos, videos, ficheros, fotos y documentos que han sido diseñado con anterioridad.
2. No se permite la reutilización de cuestionarios y ejercicios existentes para asignarlos en ATcnea, por lo que el profesor debe conformar estas actividades de forma local.
3. Actualmente las evaluaciones generadas en ATcnea, no pueden ser utilizadas en la Plataforma Educativa ZERA 2.0, para darles seguimiento a los estudiantes.
4. No existe un seguimiento de sesión durante el proceso de autenticación de los estudiantes y profesores entre ATcnea y la Plataforma Educativa ZERA 2.0.

Por lo que se puede resumir que no se permite el intercambio y accesibilidad a los recursos, cuestionarios y evaluaciones entre ambos sistemas como una alternativa para dar seguimiento del estudiante, así como la reutilización de los contenidos a través de un mecanismo de acceso único.

A partir de lo antes mencionado, surge como **problema de la investigación**: ¿Cómo contribuir al seguimiento de la evaluación del estudiante, la reutilización de recursos y el acceso único entre ATcnea y la Plataforma Educativa ZERA 2.0?

En consecuencia, con el problema expuesto se puede definir como **objeto de estudio**, la reutilización e interoperabilidad de los sistemas de gestión de aprendizaje con otros sistemas. Identificándose como **campo de acción**, la interoperabilidad y reutilización entre ATcnea y la Plataforma Educativa ZERA 2.0.

De esta manera el **objetivo general** del presente trabajo es, desarrollar un mecanismo de integración entre ATcnea y la Plataforma Educativa ZERA 2.0 para contribuir al seguimiento de la evaluación del estudiante, la reutilización de recursos y el acceso único del usuario.

En la presente investigación se sustentan las siguientes **Preguntas de investigación**:

1. ¿Cómo se establecen los mecanismos de integración entre los LMS y las aulas tecnológicas?
2. ¿De qué forma y con qué objetivos se realiza la reutilización de recursos entre los LMS y las aulas tecnológicas?
3. ¿Qué ventajas aportan la interoperabilidad entre sistemas informáticos y el acceso único entre ellos?

A partir de lo anterior quedan definidas las siguientes **tareas de investigación**:

1. Análisis de sistemas similares ya existentes a nivel nacional e internacional.
2. Análisis de herramientas, tecnologías y metodología a utilizar en el desarrollo.
3. Generación de los artefactos.
4. Confección de la concepción del mecanismo de la integración.
5. Definición de los requisitos funcionales y no funcionales.
6. Análisis y diseño del mecanismo de integración entre ATcnea y la Plataforma Educativa ZERA 2.0 para contribuir al seguimiento de la evaluación del estudiante, la reutilización de recursos y el acceso único del usuario.
7. Implementación del mecanismo de integración entre ATcnea y la Plataforma Educativa ZERA 2.0 para contribuir al seguimiento de la evaluación del estudiante, la reutilización de recursos y el acceso único del usuario.
8. Realización de pruebas de software.

Para el desarrollo de la investigación fueron empleados **métodos teóricos** y **empíricos**, con el propósito dar cumplimiento a los objetivos y las tareas propuestas. Estos se evidencian a continuación:

Métodos teóricos: Análisis histórico- Lógico: es utilizado con el fin de analizar la evolución histórica de soluciones similares y las tendencias más recientes sobre la integración entre aulas tecnológicas y de LMS. **Modelación:** es utilizado para crear una abstracción de la solución para

comprender mejor el problema, es usado en el proceso de desarrollo de software para realizar los diagramas necesarios que modelarán la solución. **Analítico-Sintético:** permitió estudiar y realizar el análisis de las referencias teóricas y bibliográficas acerca del tema, donde se hizo una extracción y síntesis de los elementos más significativos relacionados con los sistemas de aulas tecnológicas y plataformas educativas. **Inductivo-Deductivo:** es utilizado para ver la interoperabilidad de forma general, para luego enfocarla y específicamente entre aulas tecnológicas y plataformas educativas.

Métodos empíricos: Observación: permitió estudiar más de cerca cómo funciona el proceso de interoperabilidad entre sistemas, posibilitando estudiar de cerca el objeto de la investigación.

La presente investigación está conformada por tres capítulos, que queda estructurada de la siguiente manera:

En el capítulo 1 se exponen los elementos teóricos que sustentan el problema científico y los objetivos del trabajo. En este se realiza un estudio de las soluciones similares y se analizan las metodologías y herramientas que se ajustan al desarrollo de la investigación, para justificar la selección y utilización de cada una de ellas, así como los principales conceptos involucrados, para una mejor comprensión.

En el capítulo 2 se explican las características de la solución propuesta. Se describen los conceptos asociados al ámbito del sistema. Se definen los requisitos funcionales y no funcionales que debe cumplir el sistema. Se genera el modelo de dominio, las historias de usuarios y el modelo de datos. Además, se realiza el diseño del sistema. Se determina el enfoque constructivo que tendrá el sistema y sobre cuales patrones de diseño se basará la propuesta de solución.

En el capítulo 3 se describe todo el proceso de implementación y posteriormente las pruebas realizadas a la integración. Además, se definen los tipos de pruebas y los casos de prueba que para apoyar en el proceso de enseñanza-aprendizaje y llevar un registro de las evaluaciones que se generan en el ámbito presencial del aula tecnológica en la Plataforma Educativa ZERA 2.0.

Capítulo 1 FUNDAMENTACIÓN TEÓRICA

En el presente capítulo se hace un estudio acerca del dominio del problema, donde se exponen los fundamentos teóricos que sustentan la investigación. Este incluye los principales conceptos relacionados con el tema del presente trabajo, así como el resultado del estudio del estado del arte sobre soluciones similares. Se describen las tecnologías, metodologías, herramientas y lenguajes de programación utilizados en el desarrollo de la solución.

1.1 Plataformas de gestión del aprendizaje

Plataformas LMS: las plataformas LMS son espacios virtuales de aprendizaje orientados a facilitar la experiencia de capacitación a distancia, tanto para instituciones educativas como empresas. LMS es el acrónimo en inglés de *Learning Management System*, que podría traducirse como sistemas para la gestión de aprendizaje. Este sistema permite la creación de aulas virtuales donde se produce la interacción entre tutores y alumnos. También se pueden hacer evaluaciones, intercambiar archivos y participar en foros y chats, además de otras muchas herramientas adicionales.(10)

Las plataformas de *e-learning* ofrecen ambientes de aprendizaje ya diseñados e integrados. A ellos acceden los alumnos a través de una clave personal. Por ello, se trata de un espacio privado, dotado de las herramientas necesarias para aprender (comunicación, documentación, contenidos, interacción, etc.) Además, las plataformas permiten hacer un mejor seguimiento del progreso de los alumnos. Los LMS constan de un entorno de aprendizaje y relación social, al que acceden los alumnos, profesores y coordinadores y un entorno de administración, desde dónde se configuran los cursos, se dan de alta los alumnos, se importan contenidos y se habilitan servicios.(14)

1.2 Interoperabilidad entre sistemas

Interoperabilidad: según el artículo “Interoperabilidad de sistemas de organización del conocimiento: el estado del arte”; “la interoperabilidad es la capacidad que deben tener dos o más sistemas o componentes de intercambiar información y poder utilizar esa información”.(7)

La Real Academia de Ingeniería define la interoperabilidad como la “condición que permite a equipos o municiones ser usados por varios sistemas, aunque con menos prestaciones que los completamente intercambiables”.(8)

La red temática IST IDEAS ha definido la interoperabilidad como: “la habilidad de un sistema o producto de trabajar con otros sistemas o productos sin esfuerzo especial de la parte del usuario”.(9)

Por otra parte el libro “Evolución y retos de la educación Virtual Construyendo el *e-learning* del siglo XXI”, precisa la interoperabilidad como “la condición que hace posible que las diferencias entre dos o más sistemas de información no sean una barrera para que estos puedan comunicarse y utilizar los contenidos y servicios respectivos, con el fin de desarrollar una tarea determinada”.(10)

Con el uso de todas las referencias anteriores, para ser usada en el marco de la investigación se decide conformar en una síntesis para definir interoperabilidad, como la capacidad de diferentes productos o sistemas para el intercambio de datos con el fin de intercambiar información de una manera útil y significativa. La interoperabilidad entre plataformas tecnológicas hace posible que los contenidos y las herramientas de aprendizaje se puedan intercambiar y utilizar desde distintos entornos (10). La interoperabilidad en los LMS se podría garantizar con el uso de un medio y un canal de comunicación, la Arquitectura Orientada a Servicios (SOA, siglas del inglés *Service Oriented Architecture*) que es esencialmente una colección de servicios, podría establecer el canal de comunicación, mientras que la utilización de estándares podría constituir el medio de comunicación.

1.3 Servicios web

Servicios Web (*Web Services*): un servicio web es una tecnología que utiliza un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones. Distintas aplicaciones de software desarrolladas en lenguajes de programación diferentes, y ejecutadas sobre cualquier plataforma, pueden utilizar los servicios web para intercambiar datos en redes de ordenadores como Internet. La interoperabilidad se consigue mediante la adopción de estándares abiertos. La forma más habitual de implementar una aplicación orientada a servicios es mediante Servicios Web. Básicamente, una aplicación orientada a servicios es una colección de servicios web distribuidos. Estos servicios se comunican entre sí. Esta comunicación puede involucrar simplemente el paso de datos o la coordinación de alguna actividad entre varios servicios. En general, los servicios web son solo APIs Web que pueden ser accedidas en una red, como internet, y ejecutadas en un sistema de hosting remoto. Los servicios web, constituyen un avance tecnológico que no está ligado a ninguna tecnología, ya que una de sus premisas es la interoperabilidad multiplataforma. Un servicio web es más una idea, un concepto

cuya implementación debe respetar una serie de reglas e interfaces para poder ser accedido por cualquier sistema conectado a la red, siempre que disponga de permiso para ello.(15)

Un servicio web debe satisfacer los siguientes requisitos (15):

- Proporcionarnos un servicio remoto para poder ser usado desde distintas plataformas (Interoperabilidad)
- Adecuarse al medio actual (Internet)
- No debe haber ambigüedad en el tipo de datos del mensaje enviado y recibido (Interfaces tipificadas)
- Debería poder ser construida sobre un estándar de Internet ampliamente adoptado para aprovechar los conjuntos de herramientas y productos existentes creados para dicha tecnología.
- Tiene que dar soporte a distintos lenguajes de programación (Java, PHP, etc)
- No debe estar ligada a ninguna estructura de componentes; los protocolos subyacentes deberían proporcionar un nivel base de comunicación entre infraestructura de objeto distribuidos existentes.

Estándares empleados para los servicios web (15):

- *Web Services Protocol Stack*: conjunto de servicios y protocolos de los servicios web.
- *XML (Extensible Markup Language)*: formato estándar para los datos que se vayan a intercambiar.
- *SOAP (Simple Object Access Protocol)* o *XML-RPC (XML Remote Procedure Call)*: protocolos sobre los que se establece el intercambio.
- *WSDL (Web Services Description Language)*: es el lenguaje de la interfaz pública para los servicios web. Es una descripción basada en XML de los requisitos funcionales necesarios para establecer una comunicación con los servicios web.
- *UDDI (Universal Description, Discovery and Integration)*: protocolo para publicar la información de los servicios web. Permite comprobar qué servicios web están disponibles.

- *WS-Security (Web Service Security)*: protocolo de seguridad aceptado como estándar por OASIS (*Organization for the Advancement of Structured Information Standards*). Garantiza la autenticación de los actores y la confidencialidad de los mensajes enviados.
- *REST (Representational State Transfer)*: arquitectura que, con el uso del protocolo HTTP, proporciona una API que utiliza cada uno de sus métodos (GET, POST, PUT, DELETE, etcétera) para poder realizar diferentes operaciones entre la aplicación que ofrece el servicio web y el cliente.

1.3.1 SOAP (*Simple Object Access Protocol*)

SOAP es un protocolo de comunicación, por paso de mensajes XML, se utiliza para la formación de los mensajes intercambiados entre los sistemas distribuidos y la red. Los mensajes SOAP son independientes de los sistemas operativos y de los protocolos, y pueden ser transportados con el uso de una variedad de protocolos de Internet, incluyendo SMTP, y HTTP. El mensaje está compuesto de tres partes: un sobre, un encabezado y el cuerpo como se describe a continuación(16):

- *Sobre (Envelope)*: elemento raíz del documento. Contiene dos subelementos: el *Body* y el *Header*. También puede contener otros elementos hijo.
- *Cuerpo (Body)*: el cuerpo *Body* contiene la información principal del mensaje, es decir, la carga de datos del mensaje que se conoce como carga útil.
- *Encabezado (Header)*: la cabecera *Header* es opcional y contiene información que describe el mensaje, datos adicionales que no necesariamente pertenecen al cuerpo del mensaje, y también metadatos sobre enrutamiento (*routing*), seguridad o transacciones.

Un mensaje SOAP define de una manera uniforme como dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos codificados mediante XML de forma estructurada y tipada. SOAP es uno de los principales protocolos utilizados en los servicios web. También define una forma de realizar invocaciones a procedimientos remotos (RPC) con el empleo de HTTP.

Este protocolo está pensado para el intercambio de información en entornos descentralizados y distribuidos. Los dos objetivos de diseño principales de SOAP son la simplicidad y la extensibilidad. Para alcanzar estos objetivos, SOAP simplemente elimina de su arquitectura aquellos aspectos que con más frecuencia se encuentra en los sistemas distribuidos. Se

pueden agregar las características que se deseen simplemente con extender la especificación(16).

Las ventajas según (17):

- No está asociado con ningún lenguaje.
- No se encuentra fuertemente asociado a ningún protocolo de transporte.
- No está atado a ninguna infraestructura de objeto distribuido.
- Aprovecha los estándares existentes en la industria.
- Permite la interoperabilidad entre múltiples entornos.

Desventajas según (16):

- SOAP consumirá mayor ancho de banda que otros protocolos que tienen su misma finalidad. Esto es debido a que SOAP al estar basado en XML tendrá un mayor tamaño que el resto de protocolos basado en enviar datos binarios.
- SOAP es más costoso de desarrollar que otros protocolos. Es un protocolo basado en ASCII por lo que los datos necesitan ser convertidos a secuencias más bien que ser convertidos a su forma binaria. Esto consume un gran número de ciclos del servidor.
- Requiere más memoria. Construir las secuencias y analizarlas conllevará un mayor uso de memoria y posiblemente genere más “basura” que otros protocolos.
- Seguridad. SOAP usa el mismo punto de entrada en los firewalls corporativos que el puerto HTTP-80, por lo que sus mensajes pasan fácilmente a través de los firewalls.

1.3.2 REST (*Representational State Transfer*)

REST es una técnica de arquitectura de software para sistemas web más restringido y fiable. El término se originó en el año 2000, en una tesis doctoral sobre la web escrita por Roy Fielding, uno de los principales autores de la especificación del protocolo HTTP y ha pasado a ser ampliamente utilizado por la comunidad de desarrollo. REST se sustenta sobre los estándares de HTTP y URI (“*uniform resource identifier*”), que sirve para identificar recursos en Internet. Un concepto importante en REST es la existencia de recursos; para manipular estos recursos, los componentes de la red (clientes y servidores) se comunican a través de una interfaz estándar (HTTP) e intercambian representaciones de estos recursos. (18)

Si bien el término REST se refería originalmente a un conjunto de principios de arquitectura, en la actualidad se usa en el sentido más amplio para describir cualquier interfaz entre sistemas que utilice directamente HTTP para obtener datos o indicar la ejecución de operaciones sobre los datos, en cualquier formato (XML, JSON, etc) sin las abstracciones adicionales de los protocolos basados en patrones de intercambio de mensajes, como por ejemplo SOAP.

Principios básicos de REST

Para considerar un sistema como REST se debe apoyar en unos principios básicos o restricciones (18):

- **Cliente-Servidor:** La primera restricción que tiene REST está en común con el estilo Cliente-Servidor. Separar lo que es competencia del cliente y el servidor es el principio de todo. Hay que distinguir lo que concierne al interfaz del usuario del almacenamiento de datos. De esta manera se ayuda a mejorar la portabilidad de la interfaz de usuario a través de múltiples plataformas. Además, también se mejora la escalabilidad porque se simplifican las componentes del servidor al no tener que implementar las funcionalidades que van asociadas a la interfaz del usuario. Otro factor importante es que la separación permite a los componentes desarrollarse independientemente.
- **Sin estado:** Cada petición del cliente debe contener toda la información necesaria para que el servidor la comprenda y no necesite mirar ningún dato almacenado previamente sobre el contexto de la comunicación. Esta restricción mejora la visibilidad, eficiencia y escalabilidad. La visibilidad mejora porque el servidor no tiene que ocuparse de mirar en otros sitios ni realizar más operaciones para comprender la naturaleza de una simple petición. La eficiencia aumenta porque se hace más fácil recuperarse de errores parciales. La escalabilidad se ve también afectada porque al no hacer falta almacenar los estados entre las peticiones, los componentes pueden liberar recursos más rápidamente.
- **Caché:** Las respuestas a una petición deben poder ser etiquetadas como cacheable o no-cacheable. Si una respuesta es cacheable, entonces al cliente cache se le da permiso para reutilizar la respuesta más tarde si se hace una petición equivalente.
- **Interfaz uniforme:** La principal característica que distingue a REST del resto de estilos de arquitecturas de red es el énfasis de usar una interfaz uniforme entre los componentes. Las operaciones disponibles sobre los recursos son siempre las mismas y su semántica es conocida por todos los clientes y servicios.

Operaciones en REST sobre HTTP:

- POST: Crea nuevos recursos. Como retorno, se ofrece el ID automáticamente creado.
- GET: Lista un recurso.
- PUT: Reemplaza un recurso con otro (Útil para el update).
- DELETE: Elimina un recurso.

La desventaja de usar una interfaz uniforme es que degrada la eficiencia porque la información transferida está en una forma estandarizada y no según las necesidades que tenga la aplicación. Un diseño de interfaz que utiliza REST será eficiente con transferencias de datos web (suelen ser datos voluminosos).

Recursos: los recursos se definen como fuentes de información específica y son muy importantes en REST. Un cliente accede a los recursos que el servidor dispone. Imagine una aplicación empresarial con información de clientes. Los recursos generalmente son transmitidos entre el servidor y el cliente con el empleo de formatos como HTML, XML o JSON. No obstante, los recursos también pueden ser imágenes, texto plano o cualquier otro formato.

Las ventajas según (18):

- Separación cliente/servidor: Al ser sistemas independientes (solo se comunican con un lenguaje de intercambio como JSON) pueden desarrollarlos proyectos autónomos.
- REST es muy ligero, sus respuestas contienen exactamente la información que se necesita.
- Independencia de tecnologías y lenguajes: Se puede desarrollar en cualquier tipo de tecnología y lenguaje en la que el desarrollador se encuentre a gusto o que pueda minimizar su tiempo de desarrollo, o que coincida con las necesidades de su proyecto.
- Fiabilidad, escalabilidad, flexibilidad.
- La aplicación es más simple de mantener, los vínculos son más estructurados y su forma es más común. Es también más común que los vínculos sean el modo de llevar el estado de una aplicación Web.
- La ausencia de gestión de estado de cliente en el servidor conduce un consumo más eficiente de la memoria, una mayor simplificación de la funcionalidad y una capacidad más grande de gestionar un gran número de *requests* al mismo tiempo.

- La ausencia de la gestión de estado del cliente en el servidor permite implementar un balance de carga más simple: una sesión del cliente no tiene que ser mantenida por un servidor en particular, ni propagada a otros servidores. Ésta permite también una mejor evolución y tolerancia a fallos.
- La utilización de protocolo HTTP y todas las partes de sus encabezamientos (la mayor parte de la gramática HTTP).

Las desventajas según (18):

- El envío de datos de un formulario HTML que usa un método como DELETE no está comprendida en todos los navegadores. Para aliviar este problema se emula el comportamiento usando POST.
- No hay un estándar en sus respuestas por lo que no se definen tipos de datos.
- Puede producirse en determinadas circunstancias mayor rigidez en el desarrollo, sobre todo al ser dos proyectos independientes, pueden surgir situaciones donde falle la sincronización.

Tabla 1 Comparación entre REST y SOAP

	REST	SOAP
Tecnología	<ul style="list-style-type: none"> -Interacción dirigida por el usuario por medio de formularios. -Mecanismo consistente de nombrado de recursos (URL). -Centrado en la escalabilidad y rendimiento a gran escala para sistemas distribuidos hipermedia. 	<ul style="list-style-type: none"> - Flujo de eventos orquestados. -Falta de un mecanismo de nombrado. -Se centra en el diseño de aplicaciones distribuidas.
Protocolo	<ul style="list-style-type: none"> -XML autodescriptivo -HTTP es un protocolo de aplicación. -Síncrono. 	<ul style="list-style-type: none"> - Tipado fuerte, XML Schema. - HTTP es un protocolo de transporte. - Síncrono y Asíncrono
Descripción del servicio	<ul style="list-style-type: none"> -Documentos orientados al usuario que definen las direcciones de 	<ul style="list-style-type: none"> -WSDL. -Se pueden construir

	<p>petición y las respuestas.</p> <ul style="list-style-type: none"> -Interactuar con el servicio supone horas de testado y depuración de URLs. -No es necesario el tipado fuerte, si ambos lados están de acuerdo con el contenido. 	<p>automáticamente clientes por medio del WSDL.</p> <ul style="list-style-type: none"> -Tipado fuerte.
Gestión del estado	<ul style="list-style-type: none"> -El servidor no tiene estado (<i>stateless</i>). -Los recursos contienen datos y enlaces que representan transiciones a estados válidos. -Técnicas para añadir sesiones como cookies. 	<ul style="list-style-type: none"> -El servidor puede mantener el estado de la conversación. - Los mensajes solo contienen datos. - Técnicas para añadir sesiones como cabecera de sesión
Seguridad	<ul style="list-style-type: none"> -<i>Hypertext Transfer Protocol Secure</i>. -Implementado desde hace muchos años. -Comunicación punto a punto segura. 	<ul style="list-style-type: none"> -<i>WS-Security</i>. -Las implementaciones comienzan a aparecer ahora. - Comunicación origen a destino segura.
Metodología de diseño	<ul style="list-style-type: none"> -Identificar recursos a ser expuestos como Servicios. -Definir <i>Uniform Resource Locator</i> (URL) para direccionarlos. - Distinguir los recursos de solo lectura de los modificables. -Implementar e implantar el servidor web. 	<ul style="list-style-type: none"> - Listar las operaciones del servicio en el documento WSDL. - Definir un modelo de datos para el contenido de los mensajes. - Elegir un protocolo de transporte apropiado y definir las correspondientes políticas. - Implementar e implantar el contenedor del servicio web.

En resumen, los servicios basados en REST han surgido como una alternativa prometedora y diferente de los servicios basados en SOAP, esto se debe a su capacidad de transmitir datos directamente sobre HTTP y además a su simplicidad y naturaleza liviana. REST constituye una arquitectura simple, escalable, eficiente, segura, confiable y extensible. La arquitectura multinivel tanto para servicios web como para aplicaciones web dinámicas conlleva a la reutilización, simpleza, extensibilidad y a una clara separación de las responsabilidades de los componentes. Por lo antes expuesto, para garantizar la interoperabilidad entre ATcnea y la Plataforma Educativa ZERA 2.0 se utilizarán los servicios basados en REST.

1.4 Estándares y especificaciones para e-learning

Un estándar es un patrón, una tipificación o una norma de cómo realizar algo (19) y los hay de dos tipos: estándares de jure, cuando provienen de una organización acreditada que certifica una especificación, y estándares de facto, cuando la especificación se adoptan por un grupo mayoritario de individuos (20). Es claro entonces que un estándar regularmente proviene de una especificación, esto es, un conjunto de declaraciones detalladas y exactas de los requisitos funcionales y particularidades de algo que quiere construirse, instalarse o manufacturarse.

El reto de los estándares es acordar de qué forma compartir, comunicar o desarrollar modelos y sistemas con la finalidad de lograr la interoperabilidad entre los diversos componentes.

Ventajas del uso de estándares según (20)

- Contenidos reutilizables.
- Consistencia en la descripción de los contenidos.
- Normalización en la organización de sus recursos.
- Acceso a más contenidos, de más fuentes y más fácilmente localizables.
- Migración sencilla de sus sistemas a nuevas versiones, e incluso a una nueva plataforma.
- Comunicación e intercambio de información con otros sistemas.
- Administración de la información apropiada tanto del recurso como del estudiante.
- Extensión de los servicios y de las capacidades de las plataformas.

La utilización de estándares amplía las opciones de los usuarios finales, para esto reduce las restricciones de los sistemas propietarios y de soluciones aisladas (20). Las instituciones, los

docentes, los estudiantes y los proveedores se verán beneficiados al contar con contenidos flexibles, plataformas homogéneas y bases de datos compartidas y distribuidas.

Especificaciones y estándares (20):

Las organizaciones e instituciones que desarrollan especificaciones y estándares están orientadas hacia una forma común de identificar, definir y comunicar a todos los recursos involucrados en un entorno *e-learning* (contenidos, docentes, estudiantes, aplicaciones, proveedores, etcétera). Estos trabajos tuvieron sus inicios en grupos que comenzaron a trabajar diferentes áreas de los estándares. A continuación, se describen brevemente las organizaciones que trabajan en el desarrollo de propuestas para la estandarización del *e-learning*:

- *AICC (Aviation Industry Computer-Based Training Comitee)*, (21). Es una asociación de entrenamiento profesional basado en tecnología, especializado en el sector de la aviación pero que se ha permeado también a otros sectores. Se reconoce como una de los precursores de la estandarización de materiales del entrenamiento profesional.
- *IMS Global Consortium Inc.*, (22). Cuenta con miembros de organizaciones comerciales, educativas y gubernamentales dedicadas a definir y distribuir arquitecturas abiertas para actividades de educación en línea. Uno de sus resultados es lo que se conoce como el estándar IMS.
- *Advanced Distributed Learning (ADL)*, (23). La misión de ADL es proveer acceso de la más alta calidad en educación y entrenamiento, en cualquier lugar y en cualquier momento. Para cumplir con estos objetivos crean el modelo SCORM.
- *ARIADNE (Alliance of Remote Instructional Authoring and Distribution Networks for Europe)*, (24). Es un proyecto de investigación y de desarrollo tecnológico de telemática para la educación y el entrenamiento, patrocinado por la Unión Europea. El proyecto se enfoca al desarrollo de herramientas y metodologías para producir, administrar y reutilizar elementos pedagógicos basados en computadora.
- *IEEE/LTSC (Institute of Electrical and Electronics Engineers/Learning Technology Standards Committee)*, (25). El IEEE es una asociación internacional, cuya misión es promover los procesos ingenieriles para la creación, desarrollo, integración, compartición y aplicación del conocimiento sobre tecnologías electrónicas y de información. Dentro de su organización cuenta con el Comité de Estándares para Tecnología del Aprendizaje o

LTSC, que se encarga de desarrollar estándares técnicos, recomendaciones y guías para la tecnología educativa.

- W3C. (*World Wide Web Consortium*), (26). El consorcio W3C se encarga del desarrollo de tecnologías interoperables (especificaciones, normas, software y herramientas) para aprovechar todo el potencial del Web. Aunque este consorcio no está directamente vinculado con el desarrollo del *e-learning* es importante mencionarlo ya que de la interoperabilidad de la web dependen muchas de las funciones de la educación en línea.

De estos grupos de desarrollo las propuestas más ampliamente adoptadas, han sido las propuestas de IMS *Learning Consortium* y la de ADL, que han tenido como resultado las especificaciones IMS y el modelo SCORM, respectivamente.

1.4.1 IMS Global Consortium Inc

Las especificaciones IMS son el resultado de una activa iniciativa que desarrolla y propone especificaciones basadas en tecnologías abiertas (XML, JSON) para facilitar las actividades de aprendizaje sobre tecnología web, principalmente para el intercambio de contenidos y de información sobre los estudiantes. Es una propuesta ambiciosa que cubre, entre otros rubros, accesibilidad y adaptación del estudiante, la definición de competencias, el empaquetamiento de contenidos, información de agentes del proceso educativo, el diseño del aprendizaje a través de un lenguaje para expresar diferentes modelos pedagógicos, así como la formación de repositorios de contenidos digitales (22). Las especificaciones disponibles son:

- *IMS Question & Test Interoperability* (IMS QTI): permite el intercambio de contenido de artículos, pruebas y datos de resultados entre herramientas de autoría, bancos de artículos, herramientas de construcción de pruebas, plataformas de aprendizaje, sistemas de entrega de evaluaciones y motores de puntuación y análisis. El objetivo principal de la especificación QTI es definir un modelo de información y vinculación asociada que se puede utilizar para representar e intercambiar elementos de evaluación. Para los propósitos de QTI, un *ítem* es un conjunto de interacciones (posiblemente vacías) recopiladas junto con cualquier material de apoyo y un conjunto opcional de reglas para convertir las respuestas del candidato en resultados de evaluación.(27)
- *IMS Learning Tools Interoperability* (IMS LTI): su concepto principal es establecer una forma estándar de integrar aplicaciones de aprendizaje ricas (a menudo alojadas remotamente y proporcionadas a través de servicios de terceros) con plataformas como sistemas de gestión de aprendizaje, portales, repositorios de objetos de aprendizaje u

otros entornos educativos. En LTI estas aplicaciones de aprendizaje se llaman Herramientas (suministradas por Proveedores de Herramientas) y el LMS, o plataformas, se llaman Consumidores de Herramientas. El caso de uso básico detrás del desarrollo de la especificación LTI es permitir la conexión sin fisuras de aplicaciones basadas en la web, alojada externamente y contenido o herramientas (desde simples aplicaciones de comunicación como chat, hasta entornos de aprendizaje específicos de dominio para temas complejos como matemáticas o ciencia) a las plataformas que los presentan a los usuarios. En otras palabras, si tiene una aplicación de evaluación interactiva o un laboratorio de química virtual, puede conectarse de forma segura a una plataforma educativa de manera estándar sin tener que desarrollar y mantener integraciones personalizadas para cada plataforma.(28)

- *IMS Enterprise Services (IMS ES)*: define la interoperabilidad entre sistemas dentro de la misma organización. El intercambio de datos entre empresas u organizaciones es posible, pero la especificación no está desarrollada para ello, ya que no considera integridad, comunicación, seguridad y otros aspectos inherentes al intercambio de datos entre organizaciones independientes.(29)
- *IMS Learner Information Package (IMS LIP)*: corresponde a la interoperabilidad de sistemas con información del estudiante con otros sistemas que soportan el ambiente de aprendizaje en Internet. Es un conjunto de información del estudiante o de un productor de contenido de aprendizaje (autores, proveedores). La intención de la especificación es definir un conjunto de paquetes que pueden ser usados para importar y extraer datos de estudiantes de un servidor compatible con IMS.(20)
- *IMS Learning Resources Meta-Data (IMS LRM)*: esta especificación hace más eficiente el proceso de búsqueda y uso de los recursos, ya que proporciona una estructura para los elementos (metadatos) que describen o catalogan los recursos de aprendizaje, incluye también cómo los elementos deben ser usados, representados y organizados.(30)
- *IMS Resource List Interoperability (IMS RLI)*: la especificación de Interoperabilidad de la Lista de Recursos (RLI) especifica cómo se pueden intercambiar metadatos estructurados entre sistemas que almacenan y exponen recursos con el propósito de crear listas de recursos y aquellos que recopilan y organizan esas Listas de Recursos con fines educativos o formativos.(31)

- *IMS General Web Services (IMS GWS)*: el perfil de base de servicios web generales promueve la interoperabilidad para implementaciones de especificaciones basadas en servicios web en diferentes plataformas de software y proveedores. Se centra en un conjunto básico de especificaciones de servicios web y los problemas más comunes que se experimentan al implementar las especificaciones de servicio web identificadas. No es un objetivo del perfil de base de servicios web generales crear una arquitectura *plug-and-play* para servicios web o garantizar una interoperabilidad completa. El perfil de base de servicios web generales trata la interoperabilidad en la capa de aplicación, en particular, la descripción de comportamientos expuestos a través de servicios web.(32)

Cada una de las especificaciones cuenta con al menos tres documentos:

- *Information Model (Modelo de Información)*. Describe de manera conceptual la estructura de los datos, elementos y demás componentes que dan cuerpo a la especificación.
- *Best Practice and Implementation Guide (Guía de buenas prácticas y de implementación)*. Explica cómo implementar una especificación IMS a un sistema específico, orienta al equipo de desarrollo durante el proceso de implementación.
- *XML Binding (Ligadura XML)*. Este documento es una guía de notas e información sobre cómo representar el Modelo de Información a través de elementos XML, explica las convenciones de los esquemas XML que se utilizan e incluyendo el código.

Para aplicaciones específicas se puede utilizar solo una o varias de las especificaciones, lo que facilita que la incorporación de IMS a un sistema ya en marcha pueda hacerse por etapas, esto le permite irse filtrando paulatinamente en diversas aplicaciones. La investigación de estas especificaciones arrojó a la conclusión de que por el momento no es posible su empleo debido a características propias que tienen ya definidos los proyectos de ATcnea y ZERA 2.0 que están en contraste con esta especificaciones, pero para una posible mejora del mecanismo en el futuro sería conveniente utilizar varias de estas especificaciones las cuáles pueden ir incluyéndose por etapas en ambos proyectos para el beneficio del mecanismo y otras integraciones con otros sistemas que deseen realizarse más adelante.

1.5 Estudio comparativo de herramientas con soluciones similares

1.5.1 Docebo LMS (33):

En Docebo aprecian que la integración de su LMS con otros sistemas de terceros es clave, por lo que ofrecen un sistema LMS API para que los clientes aprovechen para realizar integraciones.

Docebo no implementa un paradigma estrictamente REST, sino uno que es más "REST-like", ya que soporta la conexión sin estado o *stateful* (depende de la configuración de Docebo). Para llamar a una API de Docebo necesitará agregar el URL del método a la URL de la plataforma estándar y permite:

- Administrar cuentas LMS de usuario
- Administrar cursos y materiales de capacitación
- Inscribir a los usuarios en los cursos
- Obtenga estadísticas sobre usuarios y cursos

Funcionamiento de las API en la práctica: si cierta entidad tiene un portal de intranet utilizado por sus usuarios para acceder a otros servicios o herramientas y desea que continúen el uso del portal para su formación, puede utilizar el módulo API "usuario" para mantener las cuentas de usuario sincronizadas entre el portal y el LMS. Esto significa que cualquier usuario nuevo creado en el portal puede ser enviado directamente a Docebo con toda su información personal ya llenada y cualquier actualización o eliminación del usuario puede ser aplicada en tiempo real al LMS sin tener que activar ninguna tarea adicional.

También puede mostrar a los usuarios los cursos que se supone que deben tomar desde dentro del portal para que puedan acceder a ellos en un solo clic. Esto puede lograrse con unas pocas API y el soporte SSO (*Single Sign On*) de Docebo.

Además, las API son independientes de la tecnología, por lo que puede utilizar cualquier lenguaje o marco de programación para integrar su sistema o aplicación con Docebo.

1.5.2 Integración Joomla y Moodle (34):

Joomla está enfocado a gestionar toda la información de un proyecto web, Moodle está enfocado a la gestión de la formación *online*, y puede gestionar entre otras funcionalidades: Alumnos, cursos, exámenes, unidades didácticas, etc.

La aproximación que suele tener cuando se trata de proyectos en los que intervienen las dos partes CMS y *e-learning*: la parte de gestión de contenidos web (CMS) con el uso de Joomla y la parte de gestión de contenidos de *e-learning* (LMS) con el empleo de Moodle, es la de aprovechar lo mejor de las dos plataformas e integrándolas entre sí, para así compartir usuarios, cursos, notas y matriculaciones entre los dos sistemas, así eliminamos las necesidades de que los usuarios tengan que volver a entrar en el otro sistema.

Existen varias extensiones de Joomla para realizar la integración entre Joomla y Moodle como Joomla y JFusion.

Ejemplos de integración de Joomla con Moodle:

- *Sign-on*: al hacer *login* en Joomla, el usuario queda automáticamente logueado en la otra plataforma, de esta forma podríamos tener un usuario que se ha logueado en una web en Joomla automáticamente logueado en la plataforma Moodle, donde podría acceder por ejemplo a sus cursos sin necesidad de tener que volver a hacer *login*.
- Registro de usuarios unificado: cuando el usuario realice el registro en Joomla automáticamente quedará registrado en la otra plataforma. Esto evita tener que registrarse dos veces.

Ejemplos de proyectos de integración Joomla:

- Integración Joomla y Moodle para la academia de Inglés *MissionEnglish.com*
- Integración Joomla y Moodle para la escuela Náutica *Greenwichnautica.com*

1.5.3 Interoperabilidad entre el LMS Moodle y las aplicaciones educativas de propósito específico (11):

Moodle es un LMS muy usado en el ámbito del *e-learning* por la diversidad de herramientas que ofrece en un entorno integrado, lo que permite establecer múltiples escenarios de aprendizaje; presenta una interfaz basada en web que ayuda a que los aprendices, tutores y administradores inicien sesión de manera permanente y ejecutar sus actividades de enseñanza y aprendizaje diarias, además de operar con una amplia variedad de tecnologías de servidores web y bases de datos.

Entre los recursos y las herramientas educativas que ofrece Moodle, se encuentra una denominada "*external tool*", un componente que crea un vínculo entre el LMS y las herramientas externas de aprendizaje basadas en web que sean compatibles con el estándar

de interoperabilidad IMS-LTI. Su objetivo es la integración de cualquier LMS con toda aplicación de aprendizaje desarrollada por terceros.

El estándar IMS-LTI favorece la conexión de un LMS con aplicaciones alojadas en forma externa; técnicamente, las aplicaciones externas son denominadas herramientas proveedoras (TP:*toolsproviders*) y al LMS se le conoce como herramienta consumidora (TC:*toolconsumer*), ya que consume la solicitud o recurso generado por el proveedor.

Esta interconectividad se presenta cuando el LMS institucional (por ejemplo, Moodle) de un profesor de física en Santiago de Cuba accede a un recurso denominado MAT1, creado por un profesor en la UCI y alojado en un servidor en este centro, con el cual es posible generar simulaciones a través de un equipo especializado en la UCI, que no son posibles de exportar a través del SCORM; por lo tanto, se decide utilizar LTI para comunicar ambas aplicaciones, que seguirán en su ubicación original y simplemente se les vincula para brindar al LMS de Santiago de Cuba accesibilidad al recurso MAT1 alojado en la UCI.

En este contexto de interoperabilidad, cuando un usuario cambia del LMS a otro sistema externo, es posible compartir información como la siguiente: detalles del perfil (nombre, correo electrónico), detalles del contexto instruccional (LMS utilizado) y del contexto específico del que provienen (curso) y rol dentro del contexto (alumno). Este proceso de "lanzamiento" se produce de manera segura con el empleo de esquemas de cifrado de datos mediante el navegador web del usuario. Una conexión entre los dos sistemas se crea con solo introducir la URL, la identificación del consumidor (*consumer key*) y una llave secreta (*shared secret*) para asegurar la conexión con el LMS.

1.6 Herramientas y tecnologías

Son varias las herramientas y tecnologías que fueron analizadas para una posterior selección para su empleo en la confección y el desarrollo del mecanismo de integración entre ATcnea y la Plataforma Educativa ZERA 2.0. La selección de estas define el perfil escogido por el desarrollador para implementar el producto que desea realizar. Además, la UCI ha determinado una estrategia marcaria para sus productos, la cual define los componentes del diseño con el uso de un conjunto de herramientas y tecnologías que se emplean para desarrollar el mecanismo de integración.

1.6.1 Librería FosRestBundle

FosRestBundle es un paquete bajo la licencia del MIT para crear APIs REST con Symfony2. FosRestBundle ofrece varias herramientas para ayudar en la creación de aplicaciones REST (35):

- Una capa de vista para habilitar controladores agnósticos de salida y formato.
- Controlador de excepciones para enviar códigos de estado HTTP apropiados.
- Generación automática de rutas: solo controlador RESTful (para recursos sencillos).
- Generación automática de rutas: múltiples controladores RESTful (para recursos con sub-recursos secundarios).
- Definición manual de rutas.
- Un cargador de rutas personalizado para generar las siguientes convenciones REST de URL.
- Aceptar la negociación de formato de encabezado, incluyendo el manejo de tipos de *mime* personalizados.
- Descodificación RESTful del cuerpo de solicitud HTTP y encabezados *Accept*.

1.6.2 Librería okhttp 3.6.0

Okhttp es una biblioteca de terceros desarrollada por *Square* para enviar y recibir solicitudes de red basadas en HTTP. Está construido en la parte superior de la biblioteca de Okio, que trata de ser más eficiente acerca de la lectura y escritura de datos que las bibliotecas estándar de E / S de Java mediante la creación de un grupo de memoria compartida. También es la biblioteca subyacente para la biblioteca Retrofit que proporciona seguridad de tipo para consumir API basadas en REST. Okhttp normalmente crea un nuevo subproceso de trabajo para enviar la solicitud de red y utiliza el mismo subproceso para manejar la respuesta.

Además, Okhttp v2.4 también proporciona una forma más actualizada de administrar URL internamente. En lugar de las clases `java.net.URL`, `java.net.URI` o `android.net.Uri`, proporciona una nueva clase `HttpUrl` que facilita obtener un puerto HTTP, analizar URLs y normalizar las cadenas de URL.(36)

Okhttp es un cliente HTTP que es eficiente por defecto (37):

- El soporte HTTP / 2 permite que todas las peticiones al mismo host compartan un socket.

- El agrupamiento de conexiones reduce la latencia de la solicitud (si HTTP / 2 no está disponible).
- GZIP transparente reduce los tamaños de descarga.
- El almacenamiento en caché de respuestas evita la red completamente para las solicitudes de repetición.

Okhttp persevera cuando la red es problemática: se recuperará silenciosamente de problemas comunes de conexión. Si un servicio tiene varias direcciones IP, Okhttp intentará direcciones alternativas si falla la primera conexión. Esto es necesario para IPv4 + IPv6 y para servicios alojados en centros de datos redundantes. Okhttp inicia nuevas conexiones con características TLS modernas (SNI, ALPN) y vuelve a TLS 1.0 si el anterior falla.(37)

Usar Okhttp es fácil. Su API de solicitud / respuesta está diseñada con constructores fluidos e inmutabilidad. Soporta llamadas de bloqueo síncrono y llamadas asíncronas con devoluciones de llamada.(37)

Okhttp es compatible con Android 2.3 y superior. Para Java, el requisito mínimo es 1.7.

1.6.3 Okio-1.11.0

Okio es una biblioteca que complementa `java.io` y `java.nio` para facilitar el acceso, almacenamiento y procesamiento de sus datos.(38)

Okio se construye alrededor de dos tipos que embalan mucha capacidad en una API directa *ByteStrings* y búferes(38)

- *ByteString* es una secuencia inmutable de bytes. Para los datos de caracteres, *String* es fundamental. *ByteString* facilita tratar los datos binarios como un valor. Esta clase es ergonómica: sabe codificar y descodificarse como hex, base64 y UTF-8.
- *Buffer* es una secuencia mutable de bytes. Al igual que *ArrayList*, no es necesario el tamaño de su *buffer* de antemano. Se lee y escribe los búferes como cola: se escriben los datos al final y se leen desde el frente. No hay obligación de administrar posiciones, límites o capacidades.

Internamente, *ByteString* y *Buffer* hacen algunas cosas inteligentes para ahorrar CPU y memoria. Si se codifica una cadena UTF-8 como *ByteString*, almacena en caché una referencia a esa cadena para que si se decodifica más tarde, no hay trabajo que hacer.

El búfer se implementa como una lista enlazada de segmentos. Cuando mueve los datos de un búfer a otro, reasigna la propiedad de los segmentos en lugar de copiar los datos. Este enfoque es particularmente útil para los programas multiproceso: un hilo que habla con la red puede intercambiar datos con un hilo de trabajo sin ninguna copia o ceremonia.

Sources and Sinks (38): una parte elegante del diseño de java.io es cómo las secuencias se pueden superponer para transformaciones, como cifrado y compresión. Okio incluye sus propios tipos de flujo llamados *Source* y *Sink* que funcionan como *InputStream* y *OutputStream*, pero con algunas diferencias clave:

- Tiempo de espera. Los flujos proporcionan acceso a los tiempos de espera del mecanismo de E / S subyacente. A diferencia de las secuencias de *socket* java.io, las llamadas *read ()* y *write ()* cumplen los tiempos de espera.
- Fácil de implementar. El origen declara tres métodos: *read ()*, *close ()*, y *timeout ()*. No hay peligros como las lecturas disponibles o de un solo byte que causan sorpresas de corrección y rendimiento.
- Fácil de usar. Aunque las implementaciones de *Source* y *Sink* solo tienen tres métodos para escribir, a los llamadores se les da una API rica con las interfaces *BufferedSource* y *BufferedSink*. Estas interfaces le dan todo lo que necesita en un solo lugar.
- No hay distinción artificial entre los flujos de bytes y los flujos de caracteres. Es toda la información. Se lee y escribe como bytes, cadenas UTF-8, números enteros de 32 bytes de tamaño grande, cortos *little-endian*; lo que se desee.
- Fácil de probar. La clase *Buffer* implementa *BufferedSource* y *BufferedSink* para que el código de prueba sea simple y claro.

Sources and *Sinks* interoperan con *InputStream* y *OutputStream*. Puede ver cualquier fuente como un *InputStream*, y puede ver cualquier *InputStream* como un origen. Similarmente para *Sink* y *OutputStream*.

Se usan las librerías FosRestBundle, Okhttp 3.6.0 y Okio 1.11.0. FosRestBundle por las herramientas que brinda para el trabajo con REST, Okhttp 3.6.0 ya que brinda un complemento de seguridad con respecto a APIs basadas en REST, además de las funcionalidades que brinda al trabajo con URL y Okio 1.11.0 por las facilidades que brinda al acceso, almacenamiento y procesamiento de datos.

1.6.4 XML (*Extensible Markup Language*)

XML es un estándar para describir datos y crear etiquetas. Las características especiales son la independencia de datos, o la separación de los contenidos de su presentación. El lenguaje XML es una forma, estándar industrial e independiente del sistema, de representar datos. Los datos que se representan con el uso de XML se pueden publicar en múltiples medios porque XML describe la estructura de los datos, no su formato, al contrario que el HTML, los datos de XML se pueden pasar entre aplicaciones porque la estructura de los datos se puede especificar en un esquema, lo que permite que un analizador de sintaxis valide y procese los datos que siguen el esquema. A diferencia de otros lenguajes, XML da soporte a bases de datos, por lo que es útil cuando varias aplicaciones deben comunicarse entre sí o integrar información. XML no ha nacido únicamente para su aplicación en Internet, sino que se propone como un estándar para el intercambio de información estructurada entre diferentes plataformas. Se puede usar en bases de datos, editores de texto, hojas de cálculo y casi cualquier cosa imaginable. XML es una tecnología sencilla que tiene a su alrededor otras que la complementan y la hacen mucho más grande, con unas posibilidades mucho mayores. Tiene un papel muy importante en la actualidad ya que permite la compatibilidad entre sistemas para compartir la información de una manera segura, fiable y fácil.(39)

Ventajas según (39):

- Es extensible: después de diseñado y puesto en producción, es posible extender XML con la adición de nuevas etiquetas, de modo que se pueda continuar su uso sin complicación alguna.
- El analizador es un componente estándar, no es necesario crear un analizador específico para cada versión de lenguaje XML. Esto posibilita el empleo de cualquiera de los analizadores disponibles. De esta manera se evitan *bugs* y se acelera el desarrollo de aplicaciones.
- Si un tercero decide usar un documento creado en XML, es sencillo entender su estructura y procesarla. Mejora la compatibilidad entre aplicaciones. Podemos comunicar aplicaciones de distintas plataformas, sin que importe el origen de los datos, es decir, podríamos tener una aplicación en Linux con una base de datos Postgres y comunicarla con otra aplicación en Windows y Base de Datos MS-SQL Server.
- Transformamos datos en información, pues se les añade un significado concreto y los asociamos a un contexto, con lo cual tenemos flexibilidad para estructurar documentos.

1.6.5 JSON (*JavaScript Object Notation*)

JSON es un formato ligero de intercambio de datos. Leerlo y escribirlo es simple para humanos, mientras que para las máquinas es simple interpretarlo y generarlo. Está basado en un subconjunto del Lenguaje de Programación JavaScript aunque hoy, debido a su amplia adopción como alternativa a XML, se considera un formato de lenguaje independiente. JSON es un formato de texto que es completamente independiente del lenguaje, pero utiliza convenciones que son ampliamente conocidos por los programadores de la familia de lenguajes C, incluyendo C++, C#, Java, JavaScript, Perl, Python, y muchos otros. Estas propiedades hacen que JSON sea un lenguaje ideal para el intercambio de datos. Si bien es frecuente ver JSON posicionado contra XML, también es frecuente el uso de JSON y XML en la misma aplicación. Una de las supuestas ventajas de JSON sobre XML como formato de intercambio de datos es que es mucho más sencillo escribir un analizador sintáctico (*parser*) de JSON.(40)

JSON está constituido por dos estructuras (41):

- Una colección de pares de nombre/valor. En varios lenguajes, esto es conocido como un objeto, registro, estructura, diccionario, tabla *hash*, lista de claves o un arreglo asociativo.
- Una lista ordenada de valores. En la mayoría de los lenguajes, esto se implementa como arreglos, vectores, listas o secuencias.

Estas son estructuras universales; virtualmente todos los lenguajes de programación las soportan de una forma u otra. Es razonable que un formato de intercambio de datos que es independiente del lenguaje de programación se base en estas estructuras.

1.6.6 Comparación de JSON con XML

- Una ventaja que tiene JSON con respecto a XML es que los datos son más sencillos de analizar sintácticamente.
- Otra ventaja de JSON es que es más rápido de procesar en cualquier navegador.
- Cuando existe un flujo de datos alto entre el cliente y el servidor y estos datos son vitales, JSON es más rápido que XML ya que optimiza los tiempos debido a que es menos pesado.
- UN aspecto común que comparte JSON y XML es la implementación de datos auto-descriptos e internacionalización. Ambos de estos estándares utilizan el estándar Unicode

y ambos crean datos de forma que permiten a herramientas genéricas manipular los datos.

- XML permite almacenar cualquier tipo de dato, en cambio JSON limita a almacenar datos numéricos y texto.
- Con XML es posible adjuntar cualquier fichero de cualquier formato. Por otro lado, JSON solo soporta formatos de datos tradicionales. Esto significa que es posible incluir fotos, audio, vídeo, y otros ficheros dentro de un fichero XML. Mientras que esto puede parecer algo bueno al principio, también puede ser peligroso. Eso es porque podría incluir un fichero ejecutable que podría tener peligrosas consecuencias para la seguridad. La simplicidad de las estructuras de datos que JSON soporta, hace imposible los ataques si se usa este formato.

1.7 Contextualización del sistema

Debido a indicaciones y especificaciones que propone la UCI para buscar estandarizar el proceso de desarrollo de software de la universidad se define que se debe cumplir con la metodología de desarrollo AUP-UCI. Para la ayuda en los aspectos del ciclo de vida de desarrollo del mecanismo de integración la herramienta UML CASE que se emplea es Visual Paradigm v8.0. El lenguaje de modelado definido es *Unified Modeling Language 2.0* (UML 2.0, por sus siglas en inglés), utilizado para visualizar, especificar, construir y documentar el mecanismo de integración desarrollado. Están establecidos previamente para el proyecto ZERA como IDE de desarrollo el NetBeans 8.0 el cual es software libre. El lenguaje de programación utilizado por este proyecto es PHP v7.0.13 y cuenta con Symfony v2.7 como *framework* de desarrollo. Como gestor de bases de datos está definido PostgreSQL 9.5 que cuenta con PgAdmin III v1.18.1 como herramienta para la administración de bases de datos para PostgreSQL. El servidor web que se utiliza es el NGINX v1.9.5, de código abierto, ligero y de alto rendimiento. Por su parte, el proyecto ATcnea establece para su desarrollo el IDE NetBeans 8.0. Emplea el lenguaje de programación Java 8 y utiliza a JavaFX 2.0 como *framework* para el trabajo con las interfaces del sistema. Además, HSQLDB v2.0 es el gestor de base de datos utilizado por el proyecto ATcnea.

1.8 Conclusiones del capítulo

Al haber realizado un estudio sobre las tecnologías a emplearse, sistemas similares y que sea posible una integración factible entre ATcnea y la Plataforma Educativa ZERA 2.0. Se decide optar por REST como arquitectura de *software* para sistemas web, se elige JSON como formato para el intercambio de datos. Se utiliza la librería Okhttp por las ventajas que brinda en concordancia con REST, Okio 1.11.0 por sus características con respecto al manejo de datos y FosRestBundle por sus herramientas para ayudar en aplicaciones REST. Se hace uso del lenguaje Java 8, con el empleo de JavaFX 2.0 como *framework* y el lenguaje PHP que hace uso del *framework* Symfony 2.7. De igual forma para la integración se hace uso del IDE NetBeans 8.0 en el proyecto ZERA y en ATcnea, debido a la flexibilidad que presentan para el desarrollo aplicaciones *Desktop* y *Web*. Para el desarrollo del modelado se hará uso del Visual Paradigm v8.0. Para la gestión de los datos en la Plataforma Educativa ZERA 2.0 se hará uso de una base de datos PostgreSQL, para gestionar los datos en ATcnea se hará uso de una base de datos HSQLDB. Como servidor web que soporte la Plataforma Educativa ZERA 2.0 se hace uso de un servidor NGINX en su versión 1.9.5. Todo el proceso será controlado y orientado por la metodología AUP-UCI la cual es una variante de la metodología AUP, de forma tal que se adapte al ciclo de vida definido para la actividad productiva de la UCI. Como lenguaje de modelado se utilizará UML en su versión 2.0, donde se hará uso del Visual Paradigm v8.0 para el desarrollo de este modelado.

Capítulo 2 ANÁLISIS Y DISEÑO

En este capítulo se realiza una descripción de la solución propuesta. Se representa el diagrama del modelo del dominio con la correspondiente descripción de sus elementos. Además se especifican los requisitos funcionales y no funcionales que debe cumplir el sistema. Se modelan las Historias de Usuarios (HU), así como los diagramas de clases del diseño y los diagramas de secuencia. Por otra parte se representan también el modelo de despliegue y el modelo de datos de la solución propuesta.

2.1 Modelo de dominio

Es una representación de las clases conceptuales del mundo real, no de componentes de *software*. No se trata de un conjunto de diagramas que describen clases de *software*, u objetos de *software* con responsabilidades.(42)

Puede utilizarse para capturar y expresar el entendimiento ganado en un área bajo análisis como paso previo al diseño de un sistema, ya sea de *software* o de otro tipo. Similares a los mapas mentales utilizados en el aprendizaje, el modelo de dominio es utilizado por el analista como un medio para comprender el sector industrial o de negocios. (43)

2.1.1 Conceptos del dominio

La definición correcta del modelo contribuirá a lograr una comunicación entre los usuarios, los desarrolladores y el cliente al establecer un lenguaje común para el entendimiento del mismo.

Plataforma Educativa ZERA 2.0: plataforma educativa encargada de la gestión de cursos, esta proporciona todos los servicios necesarios para el intercambio de información con ATcnea.

ATcnea Ordenador Profesor: encargado de hacer toda la gestión dentro del aula tecnológica para impartir una clase en tiempo real.

ATcnea Terminal Estudiante: encargado de la interacción con el ordenador del profesor, con el objetivo de comunicar al estudiante con el profesor y establecer una vía de intercambio de información.

2.2 Diagrama de clases del modelo de dominio

El diagrama de la Figura 1 muestra la relación que existe entre los conceptos identificados del dominio para un mejor entendimiento del problema.

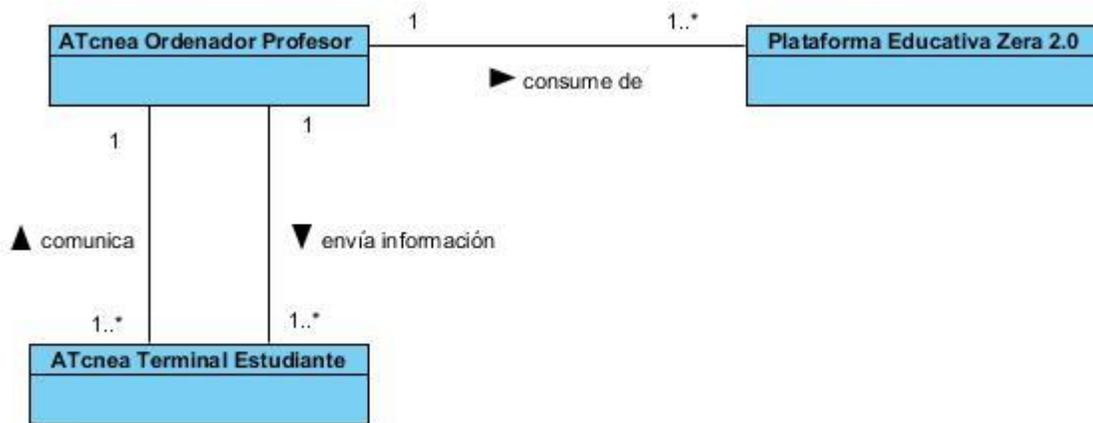


Figura 1 Modelo de dominio

2.3 Propuesta de solución

El desarrollo del mecanismo de integración propuesto tiene como objetivos el acceso único de los usuarios, la comunicación, reutilización e intercambio de información y recursos entre la Plataforma Educativa ZERA 2.0 y ATcnea. Para lograr estos objetivos se establece la comunicación entre ambos sistemas a partir del empleo de servicios web basados en REST, que no es más que un tipo de arquitectura de desarrollo web que se apoya totalmente en el estándar HTTP. Los servicios web son brindados por la Plataforma Educativa ZERA 2.0 e implementados como parte del desarrollo del mecanismo de integración, ya que hasta el momento no se encontraban implementados. Estos servicios tienen los objetivos de autenticar usuarios, obtener los cursos de un usuario, obtener recursos de tipo video y documento, obtener cuestionarios, obtener el rol de usuario, obtener todos los datos de un usuario y además existe un servicio para enviar evaluaciones desde ATcnea hacia la Plataforma Educativa ZERA 2.0. El mecanismo permitirá la autenticación de un usuario en ATcnea a partir del servicio web brindado desde la Plataforma Educativa ZERA 2.0. A partir de los cursos que tenga registrado en la Plataforma Educativa ZERA 2.0 el usuario logueado en ATcnea, creará en ATcnea las clases respectivas a dichos cursos. Además, el mecanismo de integración permite reutilizar en ATcnea recursos de tipo video y documento ya existentes en la Plataforma Educativa ZERA 2.0. Estos recursos se listan en la clase de ATcnea y podrán ser descargados, visualizados o compartidos en ATcnea. A través de este mecanismo también se hace posible en ATcnea obtener y reutilizar cuestionarios pertenecientes a la Plataforma Educativa ZERA 2.0. En ATcnea estos cuestionarios al ser resueltos generan evaluaciones que serán enviadas hacia la Plataforma Educativa ZERA 2.0. Para enviar estas evaluaciones, el mecanismo primero comprueba que el usuario que desea enviar las evaluaciones tenga los permisos necesarios

para realizar la acción requerida. Una vez validado el envío, son llevadas las evaluaciones hacia la Plataforma Educativa ZERA 2.0 a través del formato de intercambio de datos JSON.

2.4 Requerimientos del sistema

Son la descripción de los servicios proporcionados por el sistema y sus restricciones operativas. Estos requerimientos reflejan las necesidades de los clientes de un sistema que ayude a resolver algún problema.(44)

2.5 Requisitos funcionales

Los requisitos funcionales(RF) son declaraciones de los servicios que debe proporcionar el sistema, de la manera en que éste debe reaccionar a entradas particulares y de cómo se debe comportar en situaciones particulares, describen con detalle la función de éste, sus entradas y salidas.(45)

Para el correcto funcionamiento de la solución propuesta se hace la siguiente especificación de requisitos.

Tabla 2 Requisitos funcionales

°	Nombre	Descripción	Prioridad	Complejidad	Referencias cruzadas
RF1	Autenticar usuario en ATcnea mediante el servicio web brindado por la Plataforma Educativa ZERA 2.0	<p>El sistema debe permitir a los usuarios iniciar sesión en el sistema para lo que hace solicitud de los siguientes datos:</p> <ul style="list-style-type: none"> • (*) Usuario • (*) Contraseña <p>Usuario: Campo de texto que admite caracteres alfabéticos.</p> <p>Contraseña: campo de texto que permite cualquier carácter.</p> <p>En caso de error en la entrada de los campos se mostrará la notificación:</p> <ul style="list-style-type: none"> • Usuario o contraseña incorrecto 	Alta	Alta	HU_Autenticar usuario en ATcnea mediante el servicio web brindado por la Plataforma Educativa ZERA 2.0
RF2	Crear en ATcnea las clases respectivas a los cursos en ZERA	El sistema crea automáticamente en ATcnea las clases correspondientes a los cursos en ZERA 2.0 en los que esté registrado el usuario	Alta	Alta	HU_Crear en ATcnea las clases respectivas a los cursos en ZERA 2.0

	2.0	<p>actualmente logueado en ATcnea.</p> <p>Por cada curso en ZERA 2.0 se crea una clase en ATcnea.</p> <p>Las clases en ATcnea van a mostrar el nombre del curso el cual corresponde con el de ZERA 2.0 pero con las siglas LMS al final del nombre.</p> <p>Si la conexión con el LMS es satisfactoria se muestra el mensaje:</p> <ul style="list-style-type: none"> • Se han creado las clases en ATcnea contenidas en el LMS. <p>En caso que exista un error de conexión con el LMS se muestra el mensaje:</p> <ul style="list-style-type: none"> • Se produjo un error de conexión al intentar crear las clases en ATcnea desde el LMS. 			
RF3	Listar en una clase de ATcnea,	Permite listar videos en la sección Videos_LMS, perteneciente a una	Alta	Media	HU_Listar en una clase de ATcnea, recursos de tipo video

	<p>recursos de tipo video pertenecientes a un curso en ZERA 2.0</p>	<p>clase en ATcnea, los videos que contiene el curso en ZERA 2.0 que corresponde a esta clase de ATcnea.</p> <p>Esta lista se actualiza cuando el usuario selecciona la siguiente opción:</p> <ul style="list-style-type: none">• Videos_LMS <p>Las extensiones para los videos son mpeg, mp4, flv, ogg y webm.</p> <p>Se va a mostrar en la lista de videos los siguientes atributos:</p> <ul style="list-style-type: none">• nombre• descripción• nombre del archivo <p>La sección Videos_LMS tiene las opciones:</p> <ul style="list-style-type: none">• Descargar• Visualizar• Compartir			<p>pertenecientes a un curso en ZERA 2.0</p>
--	---	--	--	--	--

		<ul style="list-style-type: none"> • Cancelar <p>En caso de error en la conexión al intentar listar los recursos contenidos en el LMS se muestra el mensaje:</p> <ul style="list-style-type: none"> • Se produjo un error en la conexión al intentar listar los recursos. 			
RF4	Listar en una clase de ATcnea, recursos de tipo documentos pertenecientes a un curso en ZERA 2.0	<p>Permite listar documentos en la sección Documentos_LMS, perteneciente a una clase en ATcnea, los documentos que contiene el curso en ZERA 2.0 que corresponde a esta clase de ATcnea.</p> <p>Esta lista se actualiza cuando el usuario selecciona la siguiente opción:</p> <ul style="list-style-type: none"> • Presentaciones_LMS <p>Las extensiones para los documentos son xls, odt, pdf, doc, ppt y rtf.</p>	Alta	Media	HU_Listar en una clase de ATcnea, recursos de tipo documentos pertenecientes a un curso en ZERA 2.0

		<p>Se mostrará en la lista de documentos los siguientes atributos:</p> <ul style="list-style-type: none"> • nombre • descripción • nombre del archivo <p>La sección Videos_LMS tiene las opciones:</p> <ul style="list-style-type: none"> • Descargar • Visualizar • Compartir • Cancelar <p>En caso de error en la conexión al intentar listar los recursos contenidos en el LMS se muestra el mensaje:</p> <ul style="list-style-type: none"> • Se produjo un error en la conexión al intentar listar los recursos. 			
RF5	Compartir en una clase de ATcnea,	Permite al profesor compartir videos correspondientes a la sección	Alta	Alta	HU_Compartir en una clase de ATcnea, recursos de tipo video

	<p>recursos de tipo video pertenecientes a un curso en ZERA 2.0</p>	<p>Videos LMS, pertenecientes a una clase en ATcnea. Estos videos pertenecen al curso en ZERA 2.0 que corresponde a esta clase de ATcnea.</p> <p>Para compartir un video, el usuario debe seleccionar el video que desea compartir y luego seleccionar la opción:</p> <ul style="list-style-type: none"> • Compartir <p>Las extensiones que soporta de los videos para compartir son mpeg, mp4 y flv.</p>			<p>pertenecientes a un curso en ZERA 2.0</p>
RF6	<p>Compartir en una clase de ATcnea, recursos de tipo documento pertenecientes a un curso en ZERA 2.0</p>	<p>Permite al profesor compartir documentos correspondientes a la sección Documentos_LMS, perteneciente a una clase en ATcnea. Estos documentos pertenecen al curso en ZERA 2.0 que corresponde a esta clase de ATcnea.</p> <p>Para compartir un documento, el usuario debe seleccionar el</p>	Alta	Alta	<p>HU_Compartir en una clase de ATcnea, recursos de tipo documento pertenecientes a un curso en ZERA 2.0</p>

		<p>documento que desea compartir y luego seleccionar la opción:</p> <ul style="list-style-type: none"> • Compartir <p>Las extensiones de los documentos que soporta para compartir son pptx, pdf y ppt.</p>			
RF7	<p>Descargar en una clase de ATcnea, recursos de tipo video pertenecientes a un curso en ZERA 2.0</p>	<p>Permite descargar recursos de tipo video pertenecientes a ZERA 2.0, hacia la laptop del profesor en ATcnea.</p> <p>Para descargar un video, el usuario debe seleccionar el video que desea descargar y luego seleccionar la opción:</p> <ul style="list-style-type: none"> • Descargar <p>Cuando finaliza la descarga se muestra la notificación:</p> <ul style="list-style-type: none"> • Archivo descargado exitosamente 	Media	Alta	HU_Descargar en una clase de ATcnea, recursos de tipo video pertenecientes a un curso en ZERA 2.0
RF8	<p>Descargar en una clase de ATcnea, recursos de tipo</p>	<p>Permite descargar recursos de tipo documento, pertenecientes a ZERA 2.0, hacia la laptop del profesor en</p>	Media	Alta	HU_Descargar en una clase de ATcnea, recursos de tipo documento pertenecientes a un

	documento pertenecientes a un curso en ZERA 2.0	<p>ATcnea.</p> <p>Para descargar un documento, el usuario debe seleccionar el documento que desea descargar y luego seleccionar la opción:</p> <ul style="list-style-type: none"> • Descargar <p>Cuando finaliza la descarga se muestra la notificación:</p> <ul style="list-style-type: none"> • Archivo descargado exitosamente 			curso en ZERA 2.0
RF9	Visualizar en una clase de ATcnea, recursos de tipo documento pertenecientes a un curso en ZERA 2.0	<p>Permite visualizar recursos de tipo documento, correspondientes a la sección Documentos_LMS, perteneciente a una clase en ATcnea. Estos documentos pertenecen al curso en ZERA 2.0 que corresponde a esta clase de ATcnea.</p> <p>Para visualizar un documento, el usuario debe seleccionar el documento que desea visualizar y luego seleccionar la opción:</p>	Alta	Alta	HU_Visualizar en una clase de ATcnea, recursos de tipo documento pertenecientes a un curso en ZERA 2.0

		<ul style="list-style-type: none"> • Visualizar <p>Si existe un fallo en el instante que el profesor en ATcnea desea ver el video, entonces se mostrará la notificación:</p> <ul style="list-style-type: none"> • Se produjo un error en la conexión a la hora de ver el documento. <p>La extensión para los documentos es pdf.</p>			
RF10	Visualizar en una clase de ATcnea, recursos de tipo video pertenecientes a un curso en ZERA 2.0	<p>Permite visualizar recursos de tipo videos correspondientes a la sección Videos_LMS, perteneciente a una clase en ATcnea. Estos videos pertenecen al curso en ZERA 2.0 que corresponde a esta clase de ATcnea.</p> <p>Para visualizar un video, el usuario debe seleccionar el video que desea visualizar y luego seleccionar la opción:</p> <ul style="list-style-type: none"> • Visualizar 	Alta	Alta	HU_Visualizar en una clase de ATcnea, recursos de tipo video pertenecientes a un curso en ZERA 2.0

		<p>Si existe un fallo en el instante que el profesor en ATcnea desea ver el video, entonces se mostrará la notificación:</p> <ul style="list-style-type: none"> • Se produjo un error en la conexión a la hora de ver el video. <p>Las extensiones para los videos son mpeg, mp4, flv, ogg y webm.</p>			
RF11	Adicionar a una clase de ATcnea, cuestionarios pertenecientes a un curso en ZERA 2.0	<p>Permite adicionar a una clase en ATcnea, cuestionarios ya existentes en el curso de ZERA 2.0 correspondiente a esa clase en ATcnea.</p> <p>Los cuestionarios se adicionan en ATcnea a la sección:</p> <ul style="list-style-type: none"> • Banco de exámenes <p>Los datos mostrados de los cuestionarios son:</p> <ul style="list-style-type: none"> • Título • Forma de calificación 	Alta	Alta	HU_Adicionar a una clase de ATcnea, cuestionarios pertenecientes a un curso en ZERA 2.0

		<ul style="list-style-type: none"> • Intentos • Ejercicios 			
RF12	Enviar hacia ZERA 2.0 las evaluaciones generadas en ATcnea	<p>Envía hacia ZERA 2.0 para que sean almacenadas, las evaluaciones que reciben los estudiantes en ATcnea.</p> <p>A esta funcionalidad solo tendrán acceso los usuarios con rol de profesor.</p> <p>En la sección Enviar Evaluaciones hacia LMS, el profesor debe escoger las evaluaciones que desee enviar hacia ZERA 2.0.</p> <p>Si desea cancelar la selección para el envío, el profesor debe seleccionar la opción:</p> <ul style="list-style-type: none"> • Cancelar <p>Una vez seleccionados las evaluaciones el profesor debe seleccionar la opción:</p> <ul style="list-style-type: none"> • Enviar 	Alta	Alta	HU_Enviar hacia ZERA 2.0 las evaluaciones generadas en ATcnea.

2.6 Requisitos no funcionales

Los requisitos no funcionales (RNF) son restricciones de los servicios o funciones ofrecidos por el sistema, estos no se refieren directamente a las funciones específicas que proporciona el sistema, sino a las propiedades emergentes de éste como la fiabilidad, el tiempo de respuesta, hardware, software, seguridad y la capacidad de almacenamiento. (45)

Software:

- Para Java, el requisito mínimo es 1.7.

Interfaz:

- Las clases correspondientes a un curso se mostrarán de color azul oscuro y con las siglas LMS al final del nombre de la clase.

Confiabilidad:

- Verificar la capacidad del software para mantener su nivel de ejecución cuando se usa bajo las condiciones especificadas por el proyecto ATcnea.

Usabilidad:

- El tiempo de aprendizaje del sistema por un usuario deberá ser menor a 2 horas.
- Se reutilizarán diseños similares a los que presenta ATcnea para aprovechar la experiencia de usuario.
- El sistema debe proporcionar mensajes que sean informativos y orientados al usuario.

Seguridad:

- Garantizar el acceso a las funcionalidades definidas para los usuarios de acuerdo al tipo de usuario (profesor y estudiante).
- Mantener el sistema disponible evitando que los mecanismos de seguridad impidan el acceso a la información requerida por los usuarios autorizados.
- El nuevo sistema debe desarrollarse con el empleo de patrones y recomendaciones de programación que incrementen la seguridad de datos.

2.7 Historias de usuario

Las historias de usuario (HU) representan fragmentos de funcionalidad o comportamiento que se desea que posea un producto. El término fue introducido por Kent Beck como parte de la metodología *Extreme Programming* para desarrollar una forma de definición de requisitos

informal que pudiera favorecer el entendimiento entre todo tipo de perfiles, ya fueran técnicos o no.(46)

A continuación, se muestra la historia de usuario del requisito funcional Crear en ATcnea las clases respectivas a los cursos en ZERA 2.0. Las restantes historias de usuario se describen en los Anexos del presente documento.

Tabla 3 HU Crear en ATcnea las clases respectivas a los cursos en ZERA 2.0

Número: 2	Nombre del requisito: Crear en ATcnea las clases respectivas a los cursos en ZERA 2.0
Programador: Ronnay Castillo Andino	Iteración Asignada: 1era
Prioridad: Alta	Tiempo Estimado: 15 días
Riesgo en Desarrollo: N/A	Tiempo Real: 12 días
<p>Descripción:</p> <p>1- Objetivo: Permitir crear en ATcnea las clases respectivas a los cursos en ZERA 2.0.</p> <p>2- Acciones para lograr el objetivo (precondiciones y datos): Para crear en ATcnea las clases respectivas a los cursos en ZERA 2.0 hay que:</p> <ul style="list-style-type: none"> • Estar autenticado en ATcnea con el rol Profesor, y acceder a través del servicio autenticación que brinda la Plataforma Educativa ZERA 2.0. • Estar registrado en cursos de ZERA 2.0. <p>3- Comportamientos válidos y no válidos (flujo central y alternos):</p> <ul style="list-style-type: none"> • Las clases en ATcnea van a mostrar el nombre del curso el cual corresponde con el de ZERA 2.0. <p>4- Flujo de la acción a realizar:</p> <ul style="list-style-type: none"> • Una vez que un usuario se autentica en ATcnea, el sistema verifica el rol que posee este usuario a través del servicio web obtener el rol de un usuario. • Si el usuario es de rol Profesor, entonces el sistema crea automáticamente en ATcnea las clases correspondientes a los cursos en ZERA 2.0 en los que esté registrado el usuario. • Si la conexión con el LMS es satisfactoria se mostrará el mensaje: “Se han creado las 	

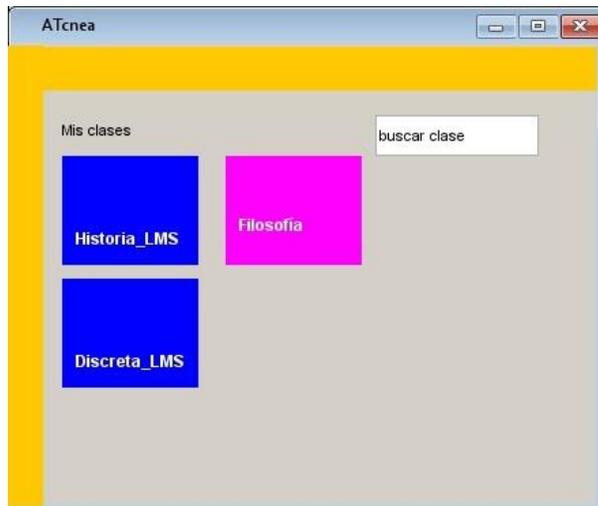
clases en ATcnea contenidas en el LMS.”

- En caso que exista un error de conexión con el LMS se muestra el mensaje: “Se produjo un error de conexión al intentar crear las clases en ATcnea desde el LMS.”

Observaciones:

- Por cada curso en ZERA 2.0 se crea una clase en ATcnea.

Prototipo de interfaz:



2.8 Patrones de diseño

Un Patrón de Diseño define un esquema para refinar los subsistemas o componentes de un sistema de software, o las relaciones entre ellos. Describe la estructura comúnmente recurrente de los componentes en comunicación, que resuelve un problema general de diseño dentro de un contexto particular.

Los patrones de diseño son el esqueleto de las soluciones a problemas comunes en el desarrollo de software. En otras palabras, brindan una solución ya probada y documentada a problemas de desarrollo de software que están sujetos a contextos similares. Los patrones de diseño son utilizados durante la fase de diseño de un sistema. (47)

2.8.1 Patrones GRASP

GRASP es el acrónimo de *General Responsibility Assignment Software Patterns*. Los patrones GRASP constituyen un apoyo para la enseñanza que ayuda a uno a entender el diseño de objetos esencial, y aplica el razonamiento para el diseño de una forma sistemática, racional y explicable. Este enfoque para la comprensión y utilización de los principios de diseño se basa en los patrones de asignación de responsabilidades.(48)

A continuación, se presenta el patrón GRASP utilizado para el diseño del sistema:

Controlador: asignar la responsabilidad de controlar el flujo de eventos del sistema, a clases específicas. Esto facilita la centralización de actividades. El controlador no realiza estas actividades, las delega en otras clases con las que mantiene un modelo de alta cohesión.

En la solución propuesta este patrón GRASP está evidenciado en las clases controladoras DocumentsLmsController, VideosLmsController, MainScreenLessonController y MainScreenController, SendEvaluationController, QuestionaryLmsController.

```
public SendEvaluationLmsController(ScreenManager myScreenManager) {
    setScreenParent(myScreenManager);
    ((MainScreenController) Main.getControlledScreen()).addToolBarButton(new HMenuToolBarButton("\uE954", HMenuToolBarButton.Section.BOX_INTEGRATION_LMS,
    @Override
```

Imagen 1 Uso del patrón Controlador en la clase SendEvaluationLmsController.java

2.8.2 Patrones GoF

Los patrones de diseño son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de software y otros ámbitos referentes al diseño de interacción o interfaces, estos deben ser aplicable a diferentes problemas de diseño en distintas circunstancias.(49)

Los patrones de diseño *Gang-of-Four* (GoF) ofrecen soluciones flexibles a problemas comunes de desarrollo de software. (50)

A continuación, se presenta el patrón GoF utilizado para el diseño del sistema:

Singleton (Instancia única): garantiza la existencia de una única instancia para una clase y la creación de un mecanismo de acceso global a dicha instancia. Permite asegurar que de una clase concreta existe una única instancia y proporciona un método único que la devuelve.

En la solución propuesta este patrón GoF está evidenciado en la clase ATcneaSingleton.

```
public void onMouseClicked() {
    myScreenManager.openWindow(IntegrationSceneConfig.QUESTIONARY, ATcneaSingleton.getInstance().getI18N().getString("lms.list.evaluat
    StageStyle.UTILITY, QuestionaryLmsController.this);
}
```

Imagen 2 Patrón Singleton

2.9 Modelo de diseño

Este modelo describe todos los diagramas contenidos dentro del diseño para una mejor comprensión del análisis propuesto.

2.9.1 Diagramas de clases del diseño

El diagrama de clases del diseño (DCD) es el encargado de mostrar las clases participantes, subsistemas y sus relaciones, así como los atributos y operaciones correspondientes a cada clase.(51)

A continuación, se presenta el DCD de la historia de usuario Crear en ATcnea las clases respectivas a los cursos en ZERA 2.0. Para el estudio de los demás diagramas remitirse a los Anexos.

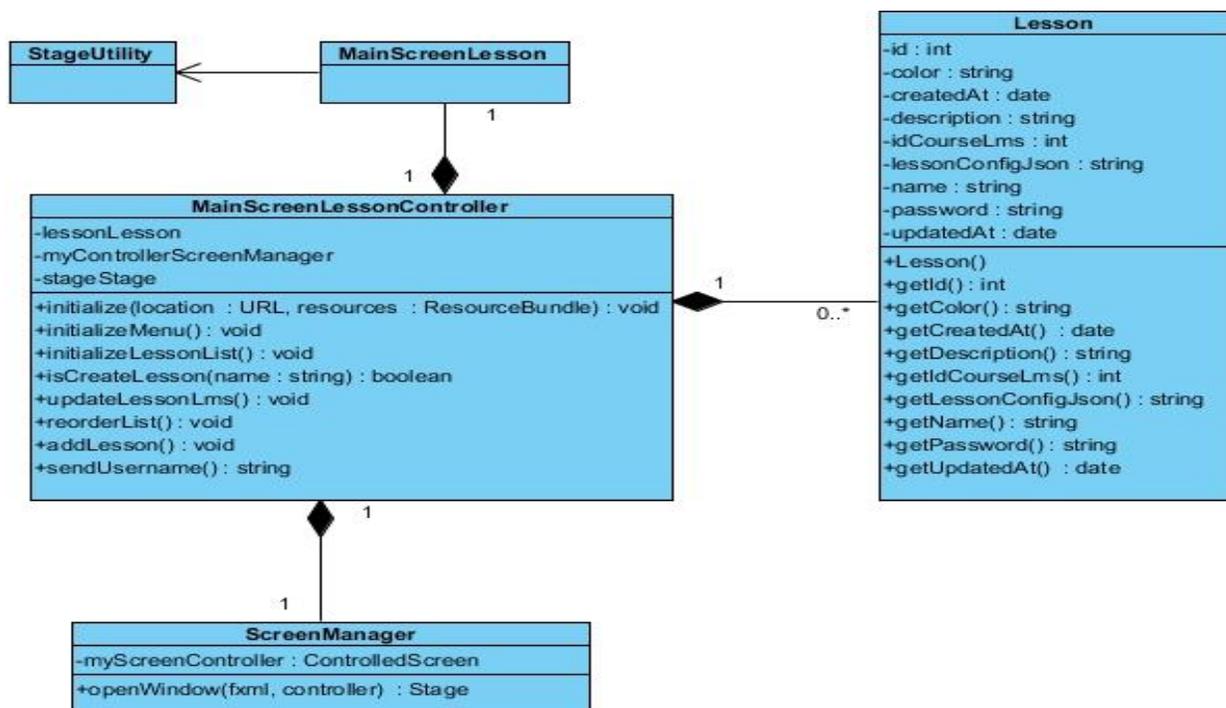


Figura 2 DCD_HU_Crear en ATcnea las clases respectivas a los cursos en ZERA 2.0

2.9.2 Diagramas de secuencia del diseño

El diagrama de secuencia del diseño (DSD) describe el curso particular de los eventos de una funcionalidad, los actores externos que interactúan directamente con el sistema y con los eventos del sistema generados por los actores.

Cada objeto tiene asociados una línea de vida y focos de control. La línea de vida indica el intervalo de tiempo durante el que existe ese objeto. Un foco de control o activación muestra el

periodo de tiempo en el cual el objeto se encuentra ejecutando alguna operación, ya sea directamente o mediante un procedimiento concurrente.(52)

A continuación, se presenta el DSD de la historia de usuario Crear en ATcnea las clases respectivas a los cursos en ZERA 2.0. Para consultar los demás diagramas remitirse a los Anexos.

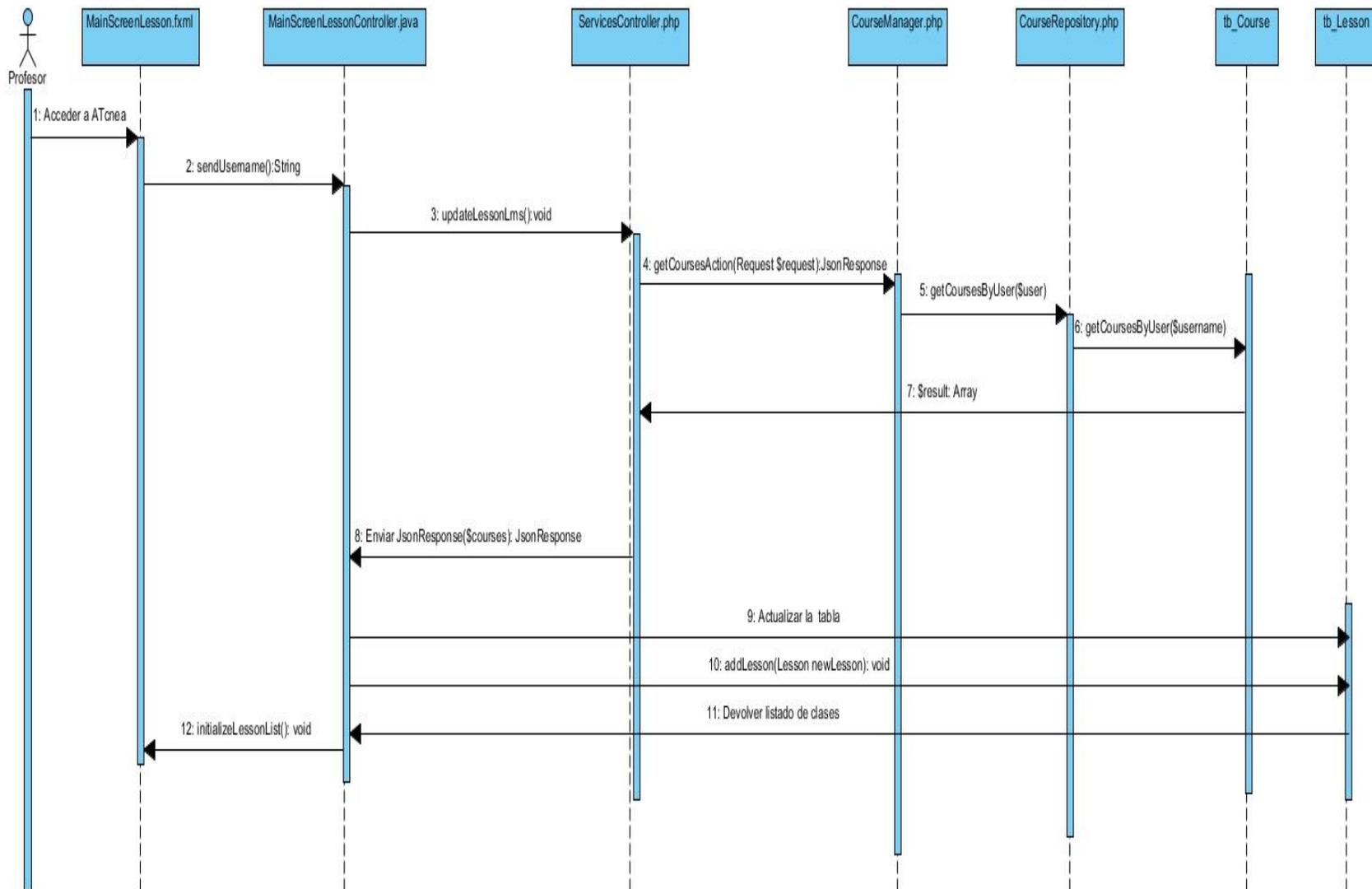


Figura 3 DSD_HU_Crear en ATcnea las clases respectivas a los cursos en ZERA 2.0

2.10 Diagrama de despliegue

En el diagrama de despliegue se representan dos tipos de elementos, nodos y conexiones, además de la distribución de componentes del sistema de información con respecto a la partición física del sistema.

El diagrama de despliegue que se muestra a continuación representa la distribución física del sistema a través de nodos.

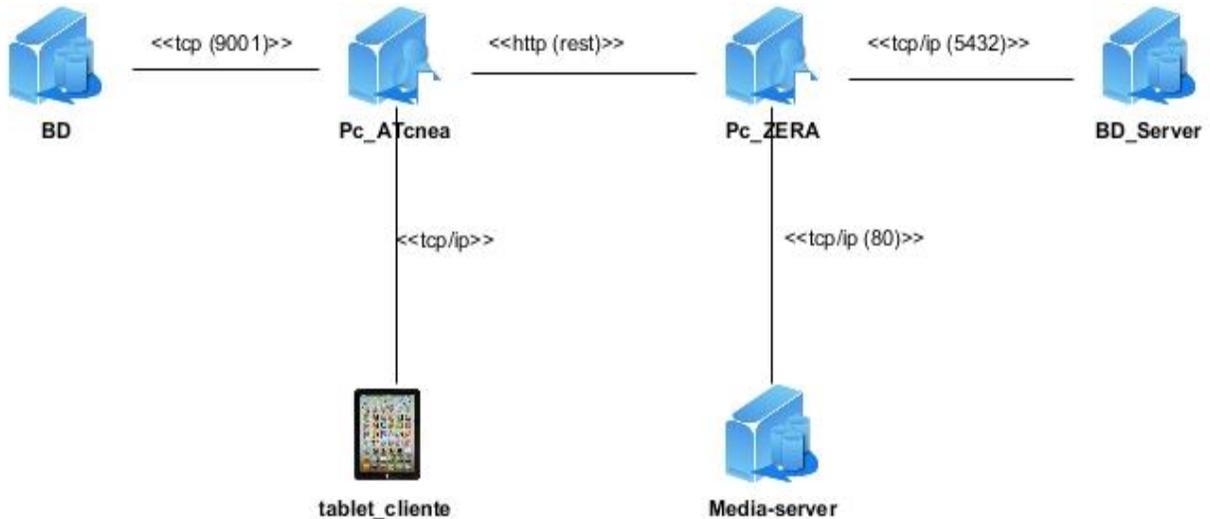


Figura 4 Diagrama de Despliegue

2.11 Modelo de datos

Este diagrama muestra la representación de las tablas que persisten en la base de datos y sus relaciones.

En la próxima figura se muestra la relación entre las tablas usadas de la base de datos:

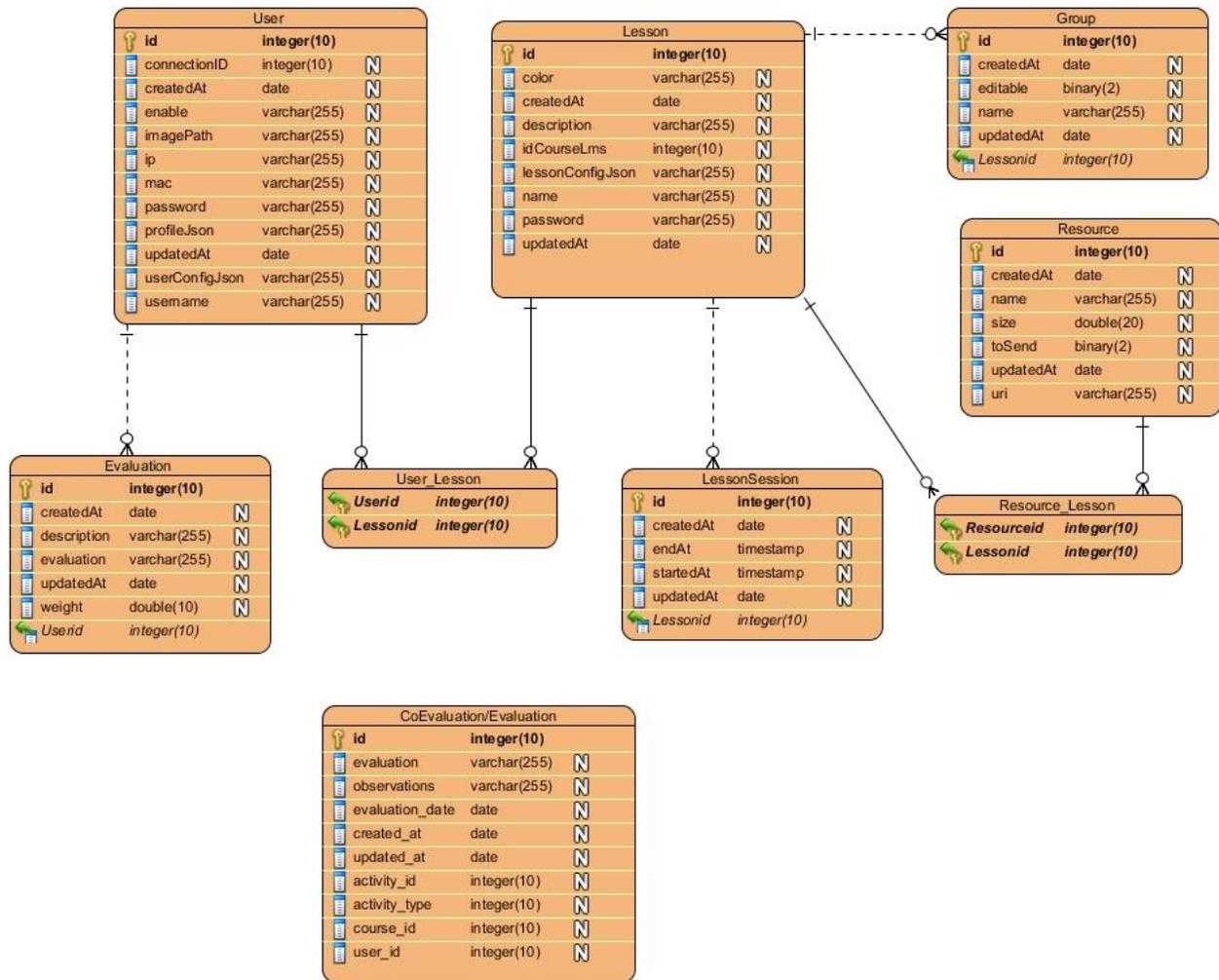


Figura 5 Modelo de datos

2.11.1 Descripción de las tablas de la Base de Datos

A continuación se presenta la descripción de la tabla Lesson. El resto de las descripciones se encuentran en los Anexos.

Tabla 4 Descripción de la tabla Lesson

Lesson		
Descripción: en esta tabla se agrupa la información correspondiente a los datos de las clases correspondientes a una sesión en ATcnea.		
Atributo	Tipo	Descripción
id	integer	Etiqueta única que identifica el objeto en la tabla.

color	character varying(255)	Almacena el color de la clase.
createdAt	date	Almacena la fecha en que se creó la clase.
description	character varying(255)	Almacena la descripción de la clase.
idCourseLms	integer	Almacena el identificador del curso de ZERA 2.0 que está creada la clase.
lessonConfigJson	character varying(255)	Almacena la configuración de la clase
name	character varying(255)	Almacena el nombre de la clase.
password	character varying(255)	Almacena la contraseña de acceso a la clase.
updatedAt	date	Almacena la fecha en que se edita la clase.

2.12 Conclusiones del capítulo

Se identificaron los principales conceptos asociados a la solución, y se reflejaron en el modelo de dominio posibilitando así definir los requisitos funcionales que sustentan la propuesta de solución representados a través de las Historias de Usuario, así como también se plantearon los requisitos no funcionales que están presentes en dicha propuesta. El desarrollo y diseño del sistema, permitió definir las bases necesarias para la implementación de la solución y permitió que se obtuvieran detallada y concretamente las relaciones que se establecen entre clases. Quedaron bien definidos y argumentados los patrones de diseño que se emplean en el desarrollo de las funcionalidades. Además, se realizó el modelo de base datos donde quedaron plasmadas todas las relaciones de las tablas que serán objetos de persistencia en la Base de Dato para las entidades que se manejan en la solución, así como el diagrama de despliegue. De esta forma, se dejan creadas las bases para comenzar con la construcción de la propuesta, con el cuidado del cumplimiento de los requerimientos identificados.

CAPÍTULO 3 IMPLEMENTACIÓN Y PRUEBAS

3.1 Modelo de implementación

El modelo de implementación describe cómo los elementos del diseño, se implementan en términos de componentes. Describe también cómo se organizan los componentes de acuerdo con los mecanismos de estructuración y modularización disponibles en el entorno de implementación y los lenguajes de programación utilizados, y cómo dependen los componentes unos de otros.(53)

3.1.1 Diagrama de componentes

Los diagramas de componentes se utilizan para modelar la vida estática de un sistema. Muestra las organizaciones y las dependencias entre un conjunto de componentes de software, además organiza los subsistemas de implementación en capas. Los componentes constituyen su elemento central, un componente es el empaquetamiento físico de los elementos de un modelo, como son las clases en el modelo de diseño.(53)

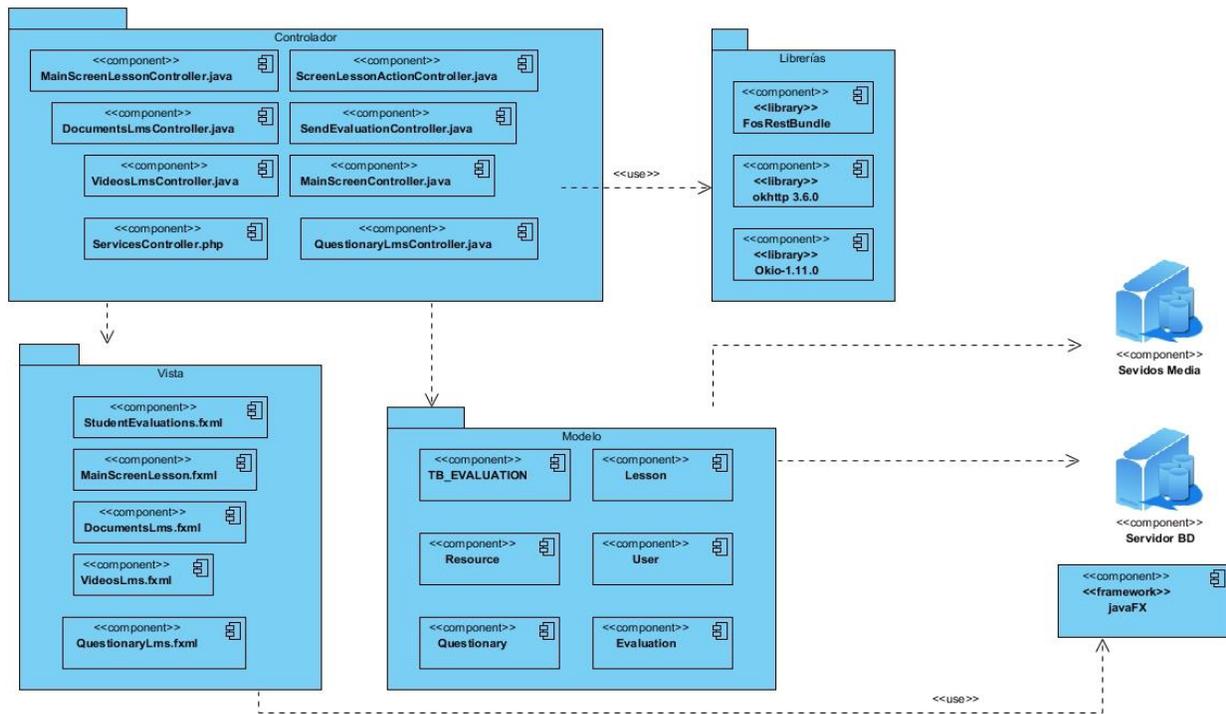


Figura 6 Diagrama de componentes

3.2 Estándares de codificación

Inicialización: intentar inicializar las variables locales al ser declaradas. La única razón para no inicializar una variable donde se declara es si el valor inicial depende de algunos cálculos que deben ocurrir.

Colocación: poner las declaraciones solo al principio de los bloques (un bloque es cualquier código encerrado por llaves "{" y "}"). No esperar al primer uso para declararlas; puede confundir a programadores no preavisados y limitar la portabilidad del código dentro de su ámbito de visibilidad.

```
void myMethod() {
    int int1 = 0; // comienzo del bloque del método

    if (condición) {
        int int2 = 0; // comienzo del bloque del "if"
        ...
    }
}
```

La excepción de la regla son los índices de bucles for, que en Java se pueden declarar en la sentencia for.

```
for (int i = 0; i < cantidadVueltas; i++) {}
```

Evitar las declaraciones locales que ocultan declaraciones de niveles superiores, por ejemplo, no declarar la misma variable en un bloque interno:

```
int cuenta;
...
myMethod() {
    if (condición) {
        int cuenta = 0; // evitar esto
        ...
    }
}
```

...

}

Declaraciones de clases e interfaces: al codificar clases e interfaces de Java, se siguen las siguientes reglas de formato:

Ningún espacio en blanco entre el nombre de un método y el paréntesis "(" que abre su lista de parámetros.

La llave de apertura "{" aparece al final de la misma línea de la sentencia declaración.

La llave de cierre "}" empieza una nueva línea para ajustarse a su sentencia de apertura correspondiente, excepto si no existen sentencias entre ambas, que debe aparecer inmediatamente después de la de apertura "{".

```
class Ejemplo extends Object {
```

```
int ivar1;
```

```
int ivar2;
```

```
Ejemplo (int i, int j) {
```

```
ivar1 = i;
```

```
ivar2 = j;
```

```
}
```

```
int emptyMethod() {}
```

...

}

Sentencias if, if-else, if else-if else: la clase de sentencias if-else debe tener la siguiente forma:

```
if (
```

```
condición
```

```
) {
```

```
sentencias;
```

```
}
```

```
if (  
condición  
) {  
sentencias;  
} else {  
sentencias;  
}  
if (  
condición  
) {  
sentencia;  
} else if (  
condición  
) {  
sentencia;  
} else {  
sentencia;  
}
```

Nota: las sentencias if usan siempre llaves {}, evitar la siguiente forma, propensa a errores:

```
if (condición)  
sentencia; //evitar esto
```

Sentencias for: una sentencia for debe tener la siguiente forma:

```
for (inicialización; condición; actualización) {  
sentencias;  
}
```

Variabes: Excepto las constantes, todas las instancias y variables de clase o método empezarán con minúscula. Las palabras internas que lo forman (si son compuestas) empiezan con su primera letra en mayúsculas. Los nombres de variables no deben empezar con los caracteres subguión "_" o signo del dólar "\$", aunque ambos están permitidos por el lenguaje. Los nombres de las variables deben ser cortos, pero con significado. Los nombres de variables de un solo carácter se deben evitar, excepto para variables índices temporales.

3.3 Pruebas de software

Las pruebas son un elemento importante dentro del ciclo de vida de un proyecto que proporciona una medida de la calidad del software. Estas son actividades en las cuales un sistema o componente es ejecutado bajo condiciones o requerimientos especificados, los resultados son observados y registrados, y una evaluación es hecha de algún aspecto del sistema o componente.(54)

Estas pruebas constituyen un elemento crucial para verificar la calidad e integridad del software, ya que permiten a los desarrolladores identificar, documentar y corregir un conjunto de no conformidades antes de liberar el producto, esto garantiza que el producto final funcione de acuerdo a los fines para los cuales fue diseñado e implemente de manera correcta los requerimientos identificados.

3.3.1 Niveles de prueba

La estrategia que se ha de seguir a la hora de evaluar dinámicamente un sistema de software debe permitir comenzar por los componentes más simples y más pequeños y avanzar progresivamente hasta probar todo el software en su conjunto. Más concretamente, los pasos a seguir son: pruebas unitarias, pruebas de integración, pruebas del sistema y pruebas de aceptación.

- **Pruebas Unitarias:** centra el proceso de verificación en la menor unidad del diseño del software (componente de software o módulo). Se prueba la interfaz del módulo para asegurar que la información fluye de forma adecuada hacia y desde la unidad de programa que está probándose.(54)
- **Pruebas de Integración:** es una técnica sistemática para construir la estructura del programa mientras que, al mismo tiempo, se llevan a cabo pruebas para detectar errores asociados con la interacción. El objetivo es tomar los módulos probados mediante la prueba de unidad y construir una estructura de programa que esté de acuerdo con lo que dicta el diseño.(54)

- **Pruebas del Sistema:** está constituida por una serie de pruebas diferentes cuyo propósito primordial es ejercitar profundamente el sistema basado en computadora. Aunque cada prueba tiene un propósito diferente, todas trabajan para verificar que se han integrado adecuadamente todos los elementos del sistema y que realizan las funciones apropiadas.(54)
- **Pruebas de Aceptación:** son realizadas por el usuario final en lugar del responsable del desarrollo del sistema. Una prueba de aceptación puede ir desde un informal paso de prueba hasta la ejecución sistemática de una serie de pruebas bien planificadas. Además, se llevan a cabo con el objetivo de que el cliente valide todos los requisitos que debe poseer el sistema.(54)

3.3.2 Métodos de prueba

Los métodos de prueba del software tienen el objetivo de diseñar pruebas que descubran diferentes tipos de errores con menor tiempo y esfuerzo. La importancia de estas pruebas es que son la garantía de la calidad del software. Realizar dichas pruebas tiene varios objetivos como detectar un error, tener un buen caso de prueba o descubrir un error no descubierto antes. Las técnicas existentes para realizar dichas pruebas son:

- **Pruebas de caja blanca:** es cuando el test es concebido a partir del conocimiento interno del software(55). Estas pruebas aseguran que la operación interna se ajusta a las especificaciones y que todos los componentes internos se han comprobado de forma adecuada. Se basan en un examen detallado de los procedimientos y caminos lógicos del sistema para determinar si el estado real coincide con el esperado.(51)
- **Pruebas de caja negra:** es cuando el método de prueba prescinde de la estructura interna del software y solo se diseña el test a partir del conocimiento expresado en su especificación(55). También son denominadas pruebas funcionales o de comportamiento y se centran en los requisitos funcionales del software. Se llevan a cabo sobre la interfaz del software para buscar errores en cada una de las funcionalidades. Con la aplicación de estas se trata de demostrar que las funciones del software son completamente operativas, que las entradas se manejan de forma adecuada y que se produce el resultado esperado.(51)

3.4 Diseño de casos de prueba

El diseño de casos de prueba es una parte de las pruebas de componentes y sistemas en las que se diseñan los casos de prueba (entradas y salidas esperadas) para probar el sistema. El

objetivo del proceso de diseño de casos de prueba es crear un conjunto de casos de prueba que sean efectivos para descubrir defectos en los programas y muestren que el sistema satisface sus requerimientos.(54)

A continuación se muestra el diseño de caso de prueba correspondiente a la HU Descargar en una clase de ATcnea, recursos de tipo video pertenecientes a un curso en ZERA 2.0. El resto de los otros casos de pruebas del diseño se encuentran en los Anexos.

Tabla 5 DCP de la HU Descargar en una clase de ATcnea, recursos de tipo video pertenecientes a un curso en ZERA 2.0

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Opción Descargar	Selecciona la opción Descargar que permite descargar un video.	El botón "Descargar" solo se activará si se selecciona un video de la lista. Brinda la posibilidad de escoger el directorio donde se desee descargar el video.	Videos_LMS/Descargar/Seleccionar_directorio/Aceptar
EC 1.2 Descarga exitosa	La descarga se ha realizado correctamente	Muestra la notificación: Archivo descargado exitosamente.	Videos_LMS/Descargar/Seleccionar_directorio/Aceptar
EC 1.3 Descarga fallida	Existe un error en la descarga	Muestra la notificación: La descarga ha fallado.	Videos_LMS/Descargar/Seleccionar_directorio/Aceptar
EC 1.4 Opción Cancelar	Selecciona la opción Cancelar	Regresa a la vista principal de la clase.	Videos_LMS/Cancelar

3.5 Resultados obtenidos

3.5.1 Pruebas de integración

Para probar la interacción entre las unidades del software se realizaron pruebas de integración. Se emplearon estas pruebas para verificar que los componentes del mecanismo funcionan correctamente si actúan en conjunto entre ellos y con todo el sistema en general. Los tres tipos más comunes de realizarse este tipo de pruebas son:

- *Big bang*: se acoplan todos los módulos de una sola vez, lo que reduce la cantidad de pruebas.

- *Bottom-up*: se prueban primero los componentes de bajo nivel y luego se avanza hacia los de mayor nivel.
- *Top-down*: el enfoque es exactamente inverso a *bottom-up*, es decir, primero se prueban los componentes de más alto nivel y posteriormente se avanza a los de menor nivel.

Este tipo de pruebas tiene varias ventajas entre las que se encuentran:

- La interacción entre módulos puede ser evaluada para agilizar los procesos.
- Útil para la detección de errores en el instante que se encuentren todos los componentes en construcción.
- Es apta para aplicar diversos escenarios para poder hacer un análisis del funcionamiento de todos los componentes en diversas situaciones.
- Se aplica antes de entregar el proyecto para evaluar el trabajo de todo el sistema con diversos escenarios.(51)

También presenta las siguientes desventajas:

- Si el proyecto es de corto plazo no se cuenta con mucho tiempo para aplicar este tipo de prueba y realizar correcciones.
- Solo es posible aplicarla hasta que esté avanzado el proyecto.
- Se tienen que trabajar con todos los módulos, no se puede hacer un análisis individual.(51)

Para las pruebas de integración realizadas al mecanismo se usó el método del *Big bang*, el cual arrojó los siguientes resultados:

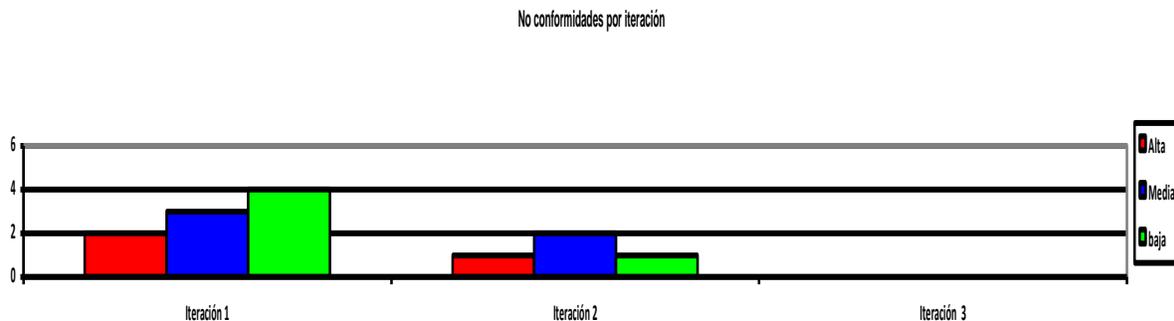


Figura 7 Resultados de las pruebas de integración

En la figura 7 se muestra la ejecución de tres iteraciones de pruebas de integración, donde cada una de las no conformidades identificadas en cada iteración fue resuelta, hasta que todas fueron corregidas. Las no conformidades identificadas se dividieron en prioridad alta, media y baja. Las principales dificultades resueltas fueron:

- La actualización de datos.
- La no uniformidad en el diseño de la interfaz.
- Error en la estructura de intercambio de datos.
- Duplicidad en nombres de variables.

3.5.2 Resultados de las pruebas de caja negra

En las pruebas de caja negra, los casos de prueba intentan exponer que son operativas las funciones del software, que la entrada se acepta de forma adecuada y que se produce una salida correcta. El mecanismo desarrollado fue sujeto a tres iteraciones de pruebas funcionales a nivel de sistema, para ello se emplearon los casos de prueba diseñados. Con la ejecución de las mismas se pudo constatar el cumplimiento de los requisitos funcionales y no funcionales del mecanismo.

Como resultado de estas pruebas se obtuvo un conjunto de 14 No Conformidades (NC) en la 1ra iteración, que fueron evaluadas en un rango comprendido entre: Alta (5), Media (5), Baja (4) según su nivel de complejidad.

A continuación se presenta el resultado de las pruebas realizadas en la 1ra iteración:

Tabla 6 Resultado de las pruebas de caja negra en la 1ra iteración

No. CP	Caso de Prueba	No conformidades			
		Alta	Media	Baja	Total
CP 2	CP Crear en ATcnea las clases respectivas a los cursos en ZERA 2.0	0	1	1	2
CP 3	Listar en una clase de ATcnea, recursos de tipo video pertenecientes a un curso en ZERA 2.0	1	0	1	2

CP 4	Listar en una clase de ATcnea, recursos de tipo documento, pertenecientes a un curso en ZERA 2.0	1	0	1	2
CP 5	Compartir en una clase de ATcnea, recursos de tipo video pertenecientes a un curso en ZERA 2.0	0	1	0	1
CP 6	Compartir en una clase de ATcnea, recursos de tipo documento pertenecientes a un curso en ZERA 2.0	0	1	0	1
CP 7	Descargar en una clase de ATcnea, recursos de tipo video pertenecientes a un curso en ZERA 2.0	1	0	0	1
CP 11	Adicionar a una clase de ATcnea, cuestionarios pertenecientes a un curso en ZERA 2.0	1	1	0	2
CP 12	Enviar hacia ZERA 2.0 las evaluaciones generadas en ATcnea.	1	1	1	3
Total de No conformidades					14

Con el objetivo de verificar las NC detectadas en la primera iteración se presenta una tabla con los siguientes datos: NC detectada, Descripción (clasificada en Alta, Media o Baja) y estado con respecto a la Solución (RA: Resuelta y Aprobada por el revisor, PD: Pendiente por solución del equipo de desarrollo, NP: No Procede, AV: Aplazada para resolver en próximas versiones, NR: Nuevo Requisito).

Tabla 7 Descripción de las No Conformidades detectadas en la 1ra iteración

No. NC	No. CP	Descripción	Complejidad	Estado
1	CP 2	Falta la tilde en la palabra <<Conexión>>	Media	RA
2	CP 2	Las clases correspondientes a cursos de ZERA 2.0 no muestran las siglas LMS al final del nombre	Baja	PD

3	CP 3	Los videos se listan dobles	Alta	RA
4	CP 3	Falta la acción <<Cancelar>>	Baja	RA
5	CP 4	Los documentos se listan dobles	Alta	RA
6	CP 4	Falta la acción <<Cancelar>>	Baja	RA
7	CP 5	No se habilita la acción <<Compartir>>	Media	RA
8	CP 6	No se habilita la acción <<Compartir>>	Media	RA
9	CP 7	No funciona la acción <<Descargar>>	Alta	RA
10	CP 11	Acepta añadir más de una vez un mismo cuestionario al banco de exámenes.	Media	PD
11	CP 11	Los ejercicios de tipo <<Selección múltiple>> solo muestran el encabezado	Alta	PD
12	CP 12	Se muestra la fecha actual de la pc, y no la fecha del envío de la evaluación	Media	RA
13	CP 12	Falta la acción <<Cancelar>>	Baja	RA
14	CP 12	No actualiza el estado de la evaluación	Alta	PD

En la 2da iteración y 3ra iteración se detectó un número menor de No conformidades, que fueron resueltas por el equipo de desarrollo. La siguiente gráfica muestra el número de No conformidades por iteración.

No conformidades por iteración

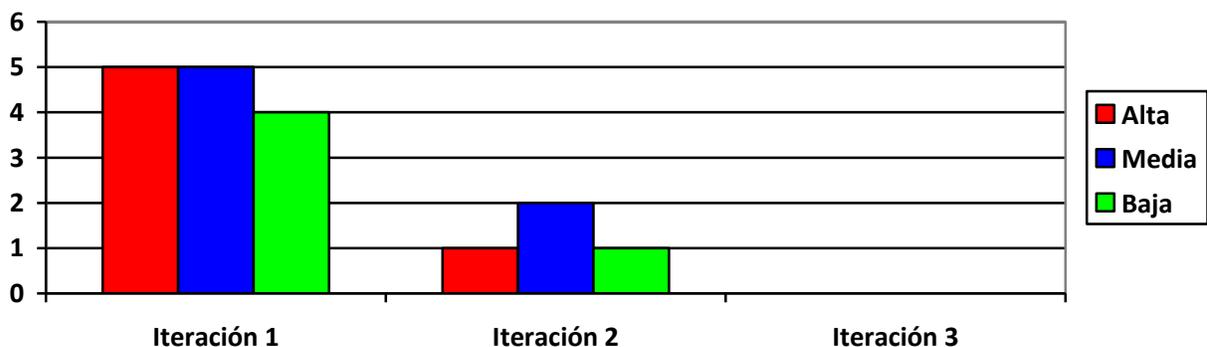


Figura 8 Resultados de las pruebas de caja negra

3.5.3 Pruebas de aceptación

Las pruebas de aceptación fueron realizadas al mecanismo que conforma la propuesta de solución. Para ello se efectuaron tres iteraciones. En la primera iteración se detectaron 4 no conformidades; de estas 0 se clasificaron de complejidad alta, 1 de complejidad media y 3 de complejidad baja. A continuación, se presenta un gráfico que muestra la relación entre las no conformidades detectadas de complejidad alta, media y baja.

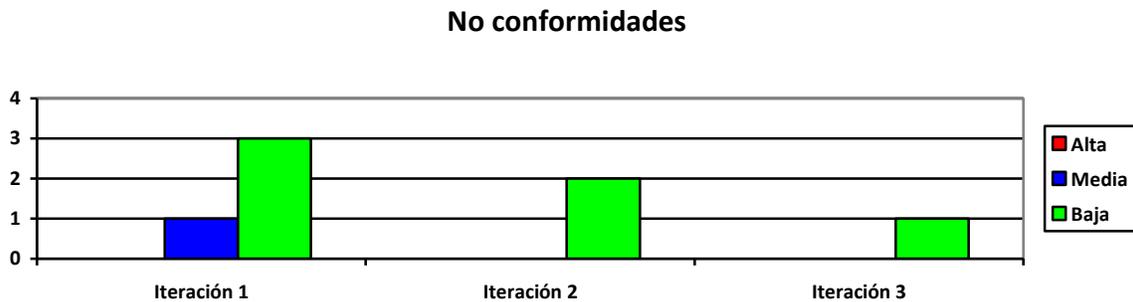


Figura 9 Resultados de las pruebas de aceptación

Concluida cada iteración de pruebas se analizaron las no conformidades encontradas y se determinaron las que constituían fallas en la integración con los sistemas ATcnea y la Plataforma Educativa ZERA 2.0 o representaban variaciones respecto a la especificación de requisitos.

3.6 Conclusiones del capítulo

En el capítulo actual se generaron los diversos artefactos que conforman el modelo de implementación. Los diagramas de componentes fueron uno de estos artefactos, los cuales son los encargados de describir cómo se organizan y cómo se implementan en términos de componentes los elementos del modelo del diseño. Se realizaron además pruebas de software con el objetivo de corregir los errores y entregar al cliente un software con buena calidad.

CONCLUSIONES GENERALES

Con el desarrollo del mecanismo de integración entre ATcnea y la Plataforma Educativa ZERA 2.0 se arriba a las siguientes conclusiones:

- Las herramientas, tecnologías y lenguajes utilizados permitieron el desarrollo del mecanismo de integración, que estuvo guiado por los artefactos generados en correspondencia con la Metodología AUP-UCI.
- El desarrollo del mecanismo de integración entre ATcnea y la Plataforma Educativa ZERA 2.0 dota a ambos sistemas de una comunicación entre ellos, proporcionándole al profesor la posibilidad de reutilizar cursos, recursos, cuestionarios y dar seguimiento en ZERA 2.0 a las evaluaciones generadas en ATcnea.
- La realización de las pruebas a la propuesta de solución permitió detectar y corregir a tiempo los errores presentes, obteniéndose un producto con las funcionalidades requeridas.

RECOMENDACIONES

Finalizada la investigación y con el objetivo de perfeccionar la aplicación y utilizar mucho más sus potencialidades en el futuro, se recomienda:

- El uso de estándares y especificaciones con la finalidad de facilitar la interoperabilidad entre ATcnea y la Plataforma Educativa ZERA 2.0, además de utilizarse para futuras interoperabilidades con otros sistemas.

REFERENCIAS BIBLIOGRÁFICAS

1. Naryana Linares, Edistio Yoel Verdecia Martínez and Eduardo Alfonso Álvarez Sánchez. Tendencias en el desarrollo de las TIC y su impacto en el campo de la enseñanza | Linares | Revista Cubana de Ciencias Informáticas. [online]. Vol. 8, no. 1. [Accessed 10 May 2017]. Available from: <http://rcci.uci.cu/?journal=rcci&page=article&op=view&path%5B%5D=517&path%5B%5D=253>
2. Portal de la Comunicación InCom-UAB · Agenda · XVI Congreso Internacional de Informática en la Educación (INFOREDU2016): Conectando sociedades. [online]. [Accessed 10 May 2017]. Available from: http://www.portalcomunicacion.com/agenda_det.asp?cal=2016-3&type=1&id=4792
3. Plataformas educativas. [online]. [Accessed 31 May 2017]. Available from: <http://aprendeonline.udea.edu.co/banco/html/plataformaseducativas/>
4. William R Watson. An Argument for Clarity: What are Learning Management Systems, What are They Not, and What Should They Become? *TechTrends*. 1 March 2007. Vol. 51, no. 2, p. 28–34. DOI 10.1007/s11528-007-0023-y.
5. Riedl, Joan. *The Integrated Technology Classroom: Building Self-Reliant Learners*. 1995. ED385220. ISBN ISBN-0-205-16157-X.
6. NMC Horizon | The New Media Consortium. [online]. [Accessed 9 May 2017]. Available from: <https://www.nmc.org/nmc-horizon/>
7. Martínez Tamayo, Ana M., Valdez, Julia C., Stubbs, Edgardo A., González Terán, Yanina and Kessler, María Inés. Interoperabilidad de sistemas de organización del conocimiento: el estado del arte. *Información, cultura y sociedad*. 2011. No. 24, p. 15–37.
8. Home | Real Academia de Ingeniería. [online]. [Accessed 31 May 2017]. Available from: <http://www.raing.es/es>
9. Raúl Poler, Ricardo Chalmeta, Sergio Palomero and Verónica Pazos. Interoperability in Business Information Systems. *University of Oldenburg* [online]. Available from: http://www.ibis.uni-oldenburg.de/download/issues/10/ibis_10.pdf
10. Begoña Gros. *Evolución y retos de la educación virtual. Construyendo el e-learning del siglo XXI*. UOC, 2011. ISBN 978-84-9788-406-8.

11. Interoperabilidad entre el LMS Moodle y las aplicaciones educativas de propósito específico utilizando servicios del IMS-LTI. [online]. [Accessed 9 May 2017]. Available from: http://www.scielo.org.mx/scielo.php?script=sci_arttext&pid=S1665-61802016000100024
12. Hurwitz, Judith, Bloor, Robin, Baroudi, Carol and Kaufman, Marcia. *Service Oriented Architecture for Dummies*. 2007.
13. Plataforma Educativa ZERA - Uciencia2048.pdf. [online]. [Accessed 9 May 2017]. Available from: <https://uciencia.eventos.uci.cu/sites/default/files/public/ponencia/ponencia/node/Uciencia2048.pdf>
14. Fernández-Pampillón Cesteros, Ana. Las plataformas e-learning para la enseñanza y el aprendizaje universitario en Internet. In: López Alonso, Covadonga, Matesanz Del Barrio, María (eds.), *Las plataformas de aprendizaje. Del mito a la realidad* [online]. Madrid: Biblioteca Nueva, 2009. p. 45–73. [Accessed 31 May 2017]. ISBN 978-84-9742-944-3. Available from: <http://eprints.ucm.es/10682/>
15. Gustavo Alonso, Fabio Casati, Harumi Kuno and Vijay Machiraju. *Web Services*. Springer Berlin Heidelberg, 2004. ISBN 978-3-642-07888-0.
16. What is SOAP? Why is SOAP required? Advantages of SOAP. [online]. [Accessed 31 March 2017]. Available from: <http://careerride.com/SOAP-What-is-SOAP.aspx>
17. DOUGLAS K BARRY. *Web Services, Service-Oriented Architectures, and Cloud Computing: The Savvy Manager's Guide*. 2. Morgan Kaufmann Publishers, [no date]. ISBN ISBN 978-0123983572.
18. Masse, Mark. *REST API Design Rulebook: Designing Consistent RESTful Web Service Interfaces*. O'Reilly Media, Inc., 2011. ISBN 978-1-4493-1990-8. Google-Books-ID: eABpzyTcJNIC
19. RAE: Real Academia Española. *Diccionario de la Lengua Española*. 22th. Madrid: Espasa Calpe SA.
20. Estándares y especificaciones para e-learning. [online]. [Accessed 18 May 2017]. Available from: <http://www.biblioweb.tic.unam.mx/libros/repositorios/estandares.htm>
21. Home. [online]. [Accessed 19 May 2017]. Available from: <https://www.aiccbbox.org/>
22. IMS Global Learning Consortium |. [online]. [Accessed 19 May 2017]. Available from: <https://www.imsglobal.org/>

23. - ADL Net. [online]. [Accessed 19 May 2017]. Available from: <http://www.adlnet.org/>
24. About | Ariadne. [online]. [Accessed 19 May 2017]. Available from: <http://www.ariadne-eu.org/>
25. IEEE - The world's largest technical professional organization dedicated to advancing technology for the benefit of humanity. [online]. [Accessed 19 May 2017]. Available from: <https://www.ieee.org/index.html>
26. World Wide Web Consortium (W3C). [online]. [Accessed 19 May 2017]. Available from: <https://www.w3.org/>
27. IMS Question & Test Interoperability Implementation Guide Version 2.1 Final | IMS Global Learning Consortium. [online]. [Accessed 19 May 2017]. Available from: http://www.imsglobal.org/question/ktiv2p1/imsqti_implv2p1.html
28. Learning Tools Interoperability | IMS Global Learning Consortium. [online]. [Accessed 19 May 2017]. Available from: <http://www.imsglobal.org/activity/learning-tools-interoperability>
29. IMS Enterprise Services Specification | IMS Global Learning Consortium. [online]. [Accessed 19 May 2017]. Available from: <http://www.imsglobal.org/es/index.html>
30. Learning Resource Meta-data Specification | IMS Global Learning Consortium. [online]. [Accessed 19 May 2017]. Available from: <http://www.imsglobal.org/metadata/index.html>
31. Resource List Interoperability | IMS Global Learning Consortium. [online]. [Accessed 19 May 2017]. Available from: <http://www.imsglobal.org/rli/index.html>
32. General Web Services | IMS Global Learning Consortium. [online]. [Accessed 19 May 2017]. Available from: <http://www.imsglobal.org/gws/index.html>
33. Docebo LMS API Documentation and third party Integrations. [online]. [Accessed 12 May 2017]. Available from: <https://www.docebo.com/lms-docebo-api-third-party-integration/#onlinedoc>
34. Integración Joomla y Moodle. [online]. [Accessed 12 May 2017]. Available from: <http://www.webactualizable.com/integracion-joomla-y-moodle>
35. Getting Started With FOSRestBundle (The Symfony Bundles Documentation). [online]. [Accessed 18 May 2017]. Available from: <http://symfony.com/doc/current/bundles/FOSRestBundle/index.html>

36. Using OkHttp · codepath/android_guides Wiki · GitHub. [online]. [Accessed 12 May 2017]. Available from: https://github.com/codepath/android_guides/wiki/Using-OkHttp
37. OkHttp. [online]. [Accessed 18 May 2017]. Available from: <http://square.github.io/okhttp/>
38. GitHub - square/okio: A modern I/O API for Java. [online]. [Accessed 18 May 2017]. Available from: <https://github.com/square/okio>
39. Extensible Markup Language (XML) 1.1 (Second Edition) - REC-xml11-20060816.pdf. [online]. [Accessed 31 March 2017]. Available from: <http://www.w3pdf.com/W3cSpec/XML/2/REC-xml11-20060816.pdf>
40. Final draft of the TC39 “The JSON Data Interchange Format” standard - ECMA-404.pdf. [online]. [Accessed 31 March 2017]. Available from: <http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-404.pdf>
41. JSON. [online]. [Accessed 31 March 2017]. Available from: <http://www.json.org/json-es.html>
42. Microsoft Word - Modelo del Dominio.docx - ModeloDominio.pdf [online]. [Accessed 7 March 2017]. Available from: <http://is.ls.fi.upm.es/docencia/is2/documentacion/ModeloDominio.pdf>
43. Modelo de Dominio | Tecnología y Synergix. [online]. [Accessed 7 March 2017]. Available from: <https://synergix.wordpress.com/2008/07/10/modelo-de-dominio/>
44. Sommerville, Ian. *Ingeniería del Software*. 2005.
45. Requerimientos Funcionales y No Funcionales (RF/RNF). [online]. [Accessed 15 February 2017]. Available from: <http://ingenieriadesoftware.bligoo.com.mx/requerimientos-funcionales-y-no-funcionales-rf-rnf>
46. Artefactos de Scrum. Historias de Usuario en el Product Backlog - Becoming An Agile Architect. [online]. [Accessed 14 March 2017]. Available from: <http://www.becominganagilearchitect.com/scrum-product-backlog-historias-de-usuario-empirismo>
47. Tedeschi, Nicolás. ¿Qué es un Patrón de Diseño? [online]. [Accessed 14 March 2017]. Available from: <https://msdn.microsoft.com/es-es/library/bb972240.aspx>

48. *PatronesGrasp.PDF - patronesgrasp.pdf* [online]. [Accessed 15 March 2017]. Available from: <http://www.inf-cr.uclm.es/www/mpolo/asig/0304/0102/patronesgrasp.pdf>
49. Patrones GoF. [online]. [Accessed 15 March 2017]. Available from: <http://geektheplanet.net/5462/patrones-gof.xhtml>
50. *OOPSLA2002-final.doc - Design Pattern Implementation in Java and AspectJ.pdf* [online]. [Accessed 15 March 2017]. Available from: <file:///C:/Users/LenovoT410/Desktop/Design%20Pattern%20Implementation%20in%20Java%20and%20AspectJ.pdf>
51. Pressman, Roger S. *Ingeniería del software. Un enfoque práctico*. McGraw-Hill Interamericana de España S.L., 2010. ISBN 978-607-15-0314-5. Google-Books-ID: deuwcQAACAAJ
52. Diagrama de secuencia - manuel.cillero.es. [online]. [Accessed 15 March 2017]. Available from: <https://manuel.cillero.es/doc/metrica-3/tecnicas/diagrama-de-interaccion/diagrama-de-secuencia/>
53. Jacobson, I., Booch, G. and Rumbaugh, J. *El proceso unificado de desarrollo de software*. 2000.
54. Sommerville, Ian. *Ingeniería del software*. Pearson Educación, 2005. ISBN 978-84-7829-074-1. Ingeniería del Software Google-Books-ID: gQWd49zSut4C
55. Guillermo, Pantaleo and Rinaudo, Ludmila. *Ingeniería de Software*. Alfaomega Grupo Editor, 2015. ISBN 978-987-1609-78-9. Google-Books-ID: a8j2DQAAQBAJ

ANEXOS

Historias de usuario

Tabla 8 HU_Autenticar usuario en ATcnea mediante el servicio web brindado por la Plataforma Educativa Zera 2.0

Número: 1	Nombre del requisito: Autenticar usuario en ATcnea mediante el servicio web brindado por la Plataforma Educativa Zera 2.0
Programador: Frank Lorenzo Morfa Valdes	Iteración Asignada: 1era
Prioridad: Alta	Tiempo Estimado: 14
Riesgo en Desarrollo: N/A	Tiempo Real: 10
<p>Descripción:</p> <p>1- Objetivo:</p> <p>Permitir autenticarse al usuario en ATcnea para mantener un seguimiento de sesión en ZERA 2.0.</p> <p>2- Acciones para lograr el objetivo (precondiciones y datos):</p> <p>Para Autenticar el usuario a través del servicio web hay que:</p> <ul style="list-style-type: none"> • Seleccionar el checkbox "Autenticación desde LMS". <p>3- Comportamientos válidos y no válidos (flujo central y alternos):</p> <p>Los campos "Usuario" y "Contraseña" son obligatorios.</p> <ul style="list-style-type: none"> • Usuario: Campo de texto que admite caracteres alfabéticos. • Contraseña: campo de texto que permite cualquier carácter. <p>4- Flujo de la acción a realizar:</p> <ul style="list-style-type: none"> • Se introducen los datos en los campos "Usuario" y "Contraseña". • Se selecciona el botón "Iniciar sesión". 	

- En caso de error en la entrada de los campos se mostrará la notificación: “Usuario o contraseña incorrecto”

Observaciones:

Prototipo de interfaz:



Tabla 9 HU_Listar en una clase de ATcnea, recursos de tipo video pertenecientes a un curso en ZERA 2.0

Número: 3	Nombre del requisito: Listar en una clase de ATcnea, recursos de tipo video pertenecientes a un curso en ZERA 2.0
Programador: Ronnay Castillo Andino	Iteración Asignada: 1era
Prioridad: Alta	Tiempo Estimado: 14 días
Riesgo en Desarrollo: N/A	Tiempo Real: 10 días
Descripción:	
1- Objetivo:	
Permitir listar recursos de tipo video pertenecientes a ZERA 2.0 para utilizarlos en una clase de ATcnea.	
2- Acciones para lograr el objetivo (precondiciones y datos):	
Para listar en ATcnea videos pertenecientes a ZERA 2.0 hay que:	
<ul style="list-style-type: none"> Estar autenticado en Atcnea con el rol Profesor. 	

3- Comportamientos válidos y no válidos (flujo central y alternos):

- Lista de videos: Se listan en una tabla.
- Se mostrarán de los videos los atributos nombre, descripción, nombre del archivo.
- Se listarán videos con los formatos mpeg, mp4, flv, ogg, webm.
- La sección Videos _LMS tiene las opciones: “Descargar”, “Visualizar” y “Compartir”

4- Flujo de la acción a realizar:

- Se accede a la sección Videos_LMS perteneciente a una clase en ATcnea.
- Se actualiza la lista de videos automáticamente.
- En caso de error en la conexión al intentar listar los recursos contenidos en el LMS se muestra el mensaje: “Se produjo un error en la conexión al intentar listar los recursos.”

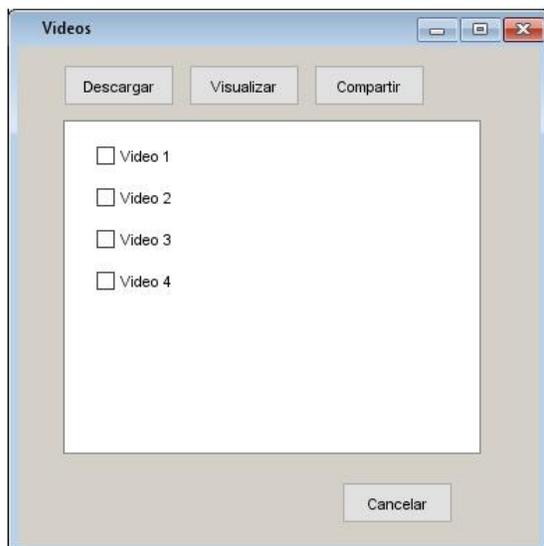
Observaciones:**Prototipo de interfaz:**

Tabla 10 HU_Listar en una clase de ATcnea, recursos de tipo documento pertenecientes a un curso en ZERA 2.0

Número: 4	Nombre del requisito: Listar en una clase de ATcnea, recursos de tipo documento pertenecientes a un curso en ZERA 2.0	
Programador: Ronnay Castillo Andino	Iteración Asignada: 1era	
Prioridad: Alta	Tiempo Estimado: 14 días	
Riesgo en Desarrollo: N/A	Tiempo Real: 10 días	
Descripción:		
1- Objetivo:		
Permitir listar recursos de tipo documento, pertenecientes a ZERA 2.0 para utilizarlos en una clase de ATcnea.		
2- Acciones para lograr el objetivo (precondiciones y datos):		
Para listar en ATcnea documentos pertenecientes a ZERA 2.0 hay que:		
<ul style="list-style-type: none"> • Estar autenticado en Atcnea con el rol Profesor. 		
3- Comportamientos válidos y no válidos (flujo central y alternos):		
<ul style="list-style-type: none"> • Lista de videos: Se listan en una tabla. • Se mostrarán de los documentos los atributos nombre, descripción, nombre del archivo. • Se listarán videos con las extensiones xls, odt, pdf, doc, ppt, rtf. • La sección Documentos _LMS tiene las opciones: "Descargar", "Visualizar" y "Compartir" 		
4- Flujo de la acción a realizar:		
<ul style="list-style-type: none"> • Se accede a la sección Documentos_LMS perteneciente a una clase en ATcnea. • Se actualiza la lista de documentos automáticamente. • En caso de error en la conexión al intentar listar los recursos contenidos en el LMS se muestra el mensaje: "Se produjo un error en la conexión al intentar listar los recursos." 		
Observaciones:		
Prototipo de interfaz:		

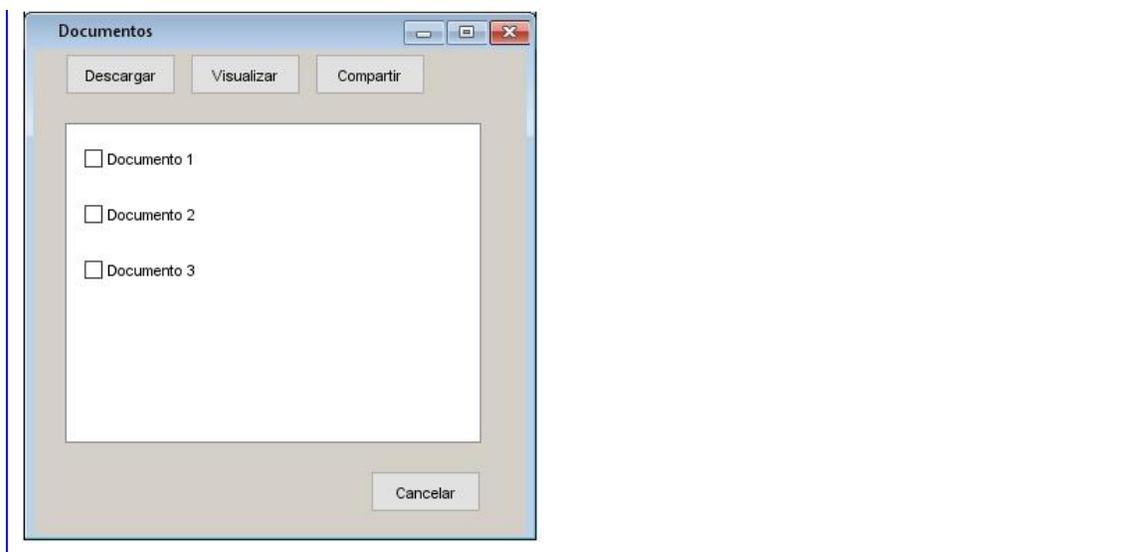


Tabla 11 HU_Compartir en una clase de ATcnea, recursos de tipo video pertenecientes a un curso en ZERA 2.0

Número: 5	Nombre del requisito: Compartir en una clase de ATcnea, recursos de tipo video pertenecientes a un curso en ZERA 2.0
Programador: Frank Lorenzo Morfa Valdés	Iteración Asignada: 1era
Prioridad: Alta	Tiempo Estimado: 15 días
Riesgo en Desarrollo: N/A	Tiempo Real: 12 días
<p>Descripción:</p> <p>1- Objetivo:</p> <p>Permitir compartir en ATcnea, recursos de tipo video pertenecientes a ZERA 2.0.</p> <p>2- Acciones para lograr el objetivo (precondiciones y datos):</p> <p>Para compartir en ATcnea, recursos de tipo video pertenecientes a un curso en ZERA 2.0 hay que:</p> <ul style="list-style-type: none"> • Estar autenticado en ATcnea con el rol Profesor. • Las extensiones de los videos que soporta para compartir son mpeg, mp4 y flv. 	

3- Comportamientos válidos y no válidos (flujo central y alternos):

- El botón “Compartir” no se habilitará hasta que no sea seleccionado un archivo en la lista de videos.

4- Flujo de la acción a realizar:

- El usuario selecciona un archivo de la lista de videos.
- Se habilita el botón “Compartir”.
- Seleccionar el botón “Compartir”..

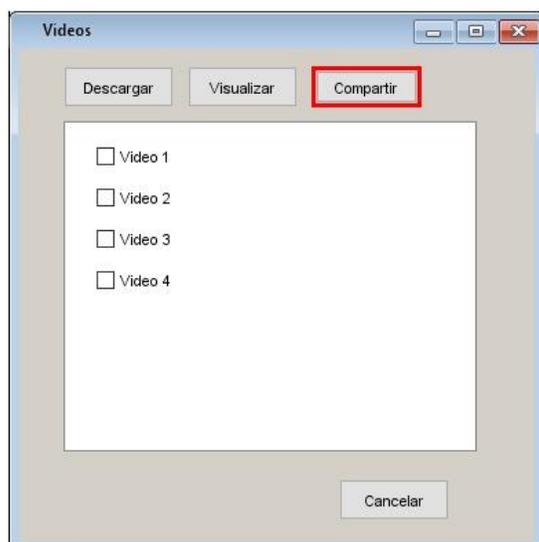
Observaciones:**Prototipo de interfaz:**

Tabla 12 HU_Compartir en una clase de ATcnea, recursos de tipo documento pertenecientes a un curso en ZERA 2.0

Número: 6	Nombre del requisito: Compartir en una clase de ATcnea, recursos de tipo documento pertenecientes a un curso en ZERA 2.0
Programador: Frank Lorenzo Morfa Valdés	Iteración Asignada: 1era
Prioridad: Alta	Tiempo Estimado: 15 días

Riesgo en Desarrollo: N/A**Tiempo Real: 12 días****Descripción:****1- Objetivo:**

Permitir compartir en ATcnea, recursos de tipo documento pertenecientes a ZERA 2.0.

2- Acciones para lograr el objetivo (precondiciones y datos):

Para compartir en ATcnea, recursos de tipo documento pertenecientes a un curso en ZERA 2.0 hay que:

- Estar autenticado en ATcnea con el rol Profesor.
- Las extensiones de los videos que soporta para compartir son ppt, pptx y pdf.

3- Comportamientos válidos y no válidos (flujo central y alternos):

- El botón “Compartir” no se habilitará hasta que no sea seleccionado un archivo en la lista de documentos.

4- Flujo de la acción a realizar:

- El usuario selecciona un archivo de la lista de documentos.
- Se habilita el botón “Compartir”.
- Seleccionar el botón “Compartir”.

Observaciones:**Prototipo de interfaz:**

Tabla 13 HU_Descargar en una clase de ATcnea, recursos de tipo video pertenecientes a un curso en ZERA 2.0

Número: 7	Nombre del requisito: Descargar en una clase de ATcnea, recursos de tipo video pertenecientes a un curso en ZERA 2.0
Programador: Frank Lorenzo Morfa Valdés	Iteración Asignada: 1era
Prioridad: Media	Tiempo Estimado: 14 días
Riesgo en Desarrollo: N/A	Tiempo Real: 12 días
<p>Descripción:</p> <p>1- Objetivo:</p> <p>Permitir descargar en ATcnea, recursos de tipo video pertenecientes a ZERA 2.0.</p> <p>2- Acciones para lograr el objetivo (precondiciones y datos):</p> <p>Para compartir en ATcnea, recursos de tipo video pertenecientes a un curso en ZERA 2.0 hay que:</p> <ul style="list-style-type: none"> • Estar autenticado en ATcnea con el rol Profesor. <p>3- Comportamientos válidos y no válidos (flujo central y alternos):</p> <ul style="list-style-type: none"> • El botón “Descargar” no se habilitará hasta que no sea seleccionado un archivo en la lista de videos. <p>4- Flujo de la acción a realizar:</p> <ul style="list-style-type: none"> • El usuario selecciona un archivo de la lista de videos. • Se habilita el botón “Descargar”. • Seleccionar el botón “Descargar”. • Al finalizar la descarga se muestra la notificación “Archivo descargado exitosamente”. • En caso de fallo en la descarga se muestra la notificación “La descarga ha fallado” 	
Observaciones:	
Prototipo de interfaz:	

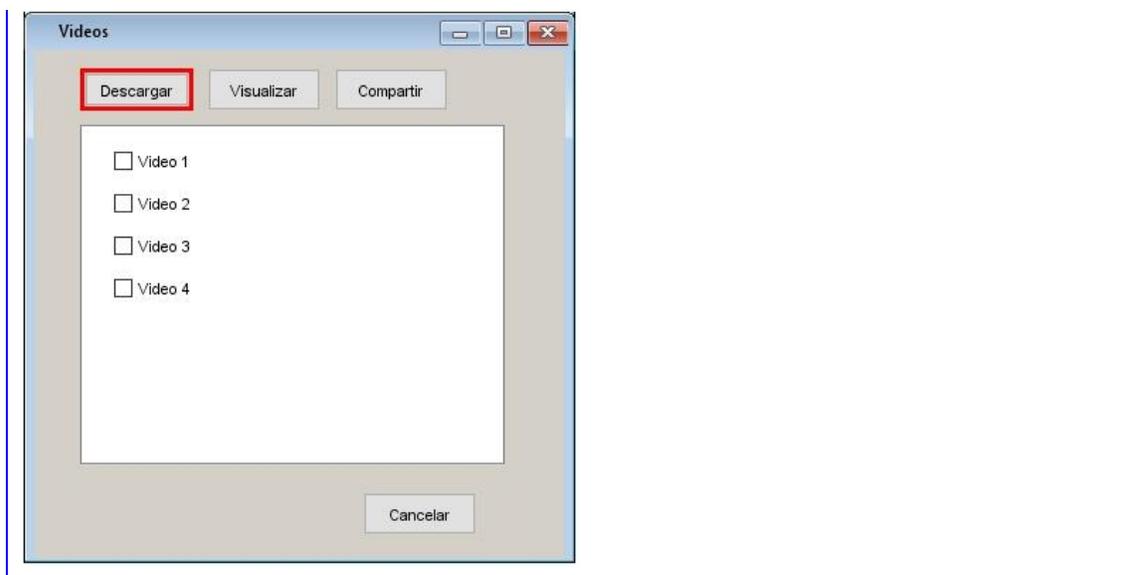


Tabla 14 HU_Descargar en una clase de ATcnea, recursos de tipo documento pertenecientes a un curso en ZERA 2.0

Número: 8	Nombre del requisito: Descargar en una clase de ATcnea, recursos de tipo documento pertenecientes a un curso en ZERA 2.0
Programador: Frank Lorenzo Morfa Valdés	Iteración Asignada: 1era
Prioridad: Media	Tiempo Estimado: 14 días
Riesgo en Desarrollo: N/A	Tiempo Real: 12 días
Descripción:	
1- Objetivo:	
Permitir descargar en ATcnea, recursos de tipo documento pertenecientes a ZERA 2.0.	
2- Acciones para lograr el objetivo (precondiciones y datos):	
Para compartir en ATcnea, recursos de tipo documento pertenecientes a un curso en ZERA 2.0 hay que:	
<ul style="list-style-type: none"> • Estar autenticado en ATcnea con el rol Profesor. 	

3- Comportamientos válidos y no válidos (flujo central y alternos):

- El botón “Descargar” no se habilitará hasta que no sea seleccionado un archivo en la lista de documentos.

4- Flujo de la acción a realizar:

- El usuario selecciona un archivo de la lista de documentos.
- Se habilita el botón “Descargar”.
- Seleccionar el botón “Descargar”.
- Al finalizar la descarga se muestra la notificación “Archivo descargado exitosamente”.
- En caso de algún fallo se muestra la notificación

Observaciones:

Prototipo de interfaz:

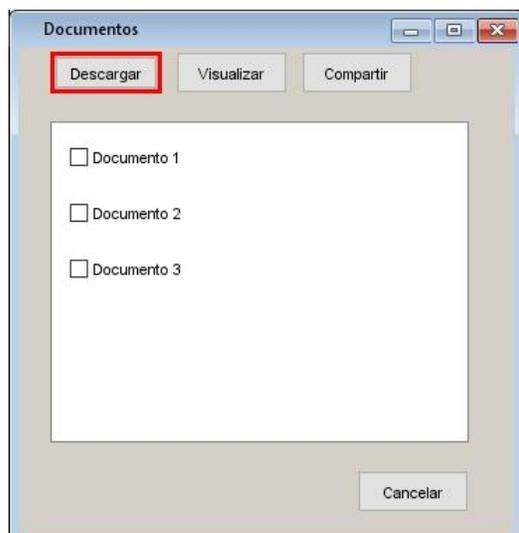


Tabla 15 HU_Visualizar en una clase de ATcnea, recursos de tipo documento pertenecientes a un curso en ZERA 2.0

Número: 9	Nombre del requisito: Visualizar en una clase de ATcnea, recursos de tipo documento pertenecientes a un curso en ZERA 2.0
Programador: Ronnay Castillo Andino	Iteración Asignada: 1era

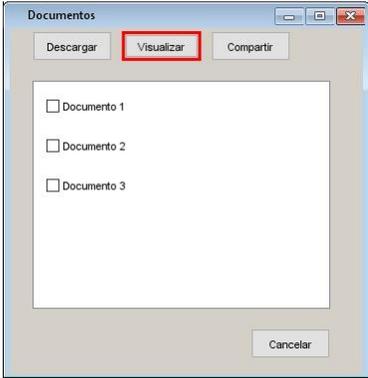
Prioridad: Media	Tiempo Estimado: 20 días
Riesgo en Desarrollo: N/A	Tiempo Real: 16 días
Descripción: 1- Objetivo: Permitir visualizar en ATcnea, recursos de tipo documento pertenecientes a ZERA 2.0. 2- Acciones para lograr el objetivo (precondiciones y datos): Para compartir en ATcnea, recursos de tipo documento pertenecientes a un curso en ZERA 2.0 hay que: <ul style="list-style-type: none">• Estar autenticado en ATcnea con el rol Profesor.• Las extensiones de los documentos deben pdf. 3- Comportamientos válidos y no válidos (flujo central y alternos): <ul style="list-style-type: none">• El botón “Visualizar” no se habilitará hasta que no sea seleccionado un archivo en la lista de documentos. 4- Flujo de la acción a realizar: <ul style="list-style-type: none">• El usuario selecciona un archivo de la lista de documentos.• Se habilita el botón “Visualizar”.• Seleccionar el botón “Visualizar”.	
Observaciones:	
Prototipo de interfaz: 	

Tabla 16 HU_Visualizar en una clase de ATcnea, recursos de tipo video pertenecientes a un curso en ZERA 2.0

Número: 10	Nombre del requisito: Visualizar en una clase de ATcnea, recursos de tipo video pertenecientes a un curso en ZERA 2.0
Programador: Ronnay Castillo Andino	Iteración Asignada: 1era
Prioridad: Media	Tiempo Estimado: 20 días
Riesgo en Desarrollo: N/A	Tiempo Real: 16 días
<p>Descripción:</p> <p>1- Objetivo:</p> <p>Permitir visualizar en ATcnea, recursos de tipo video pertenecientes a ZERA 2.0.</p> <p>2- Acciones para lograr el objetivo (precondiciones y datos):</p> <p>Para compartir en ATcnea, recursos de tipo video pertenecientes a un curso en ZERA 2.0 hay que:</p> <ul style="list-style-type: none"> • Estar autenticado en ATcnea con el rol Profesor. • Las extensiones de los videos deben ser mpeg, mp4, flv, ogg y webm. <p>3- Comportamientos válidos y no válidos (flujo central y alternos):</p> <ul style="list-style-type: none"> • El botón “Visualizar” no se habilitará hasta que no sea seleccionado un archivo en la lista de videos. <p>4- Flujo de la acción a realizar:</p> <ul style="list-style-type: none"> • El usuario selecciona un archivo de la lista de videos. • Se habilita el botón “Visualizar”. • Seleccionar el botón “Visualizar”. 	
Observaciones:	

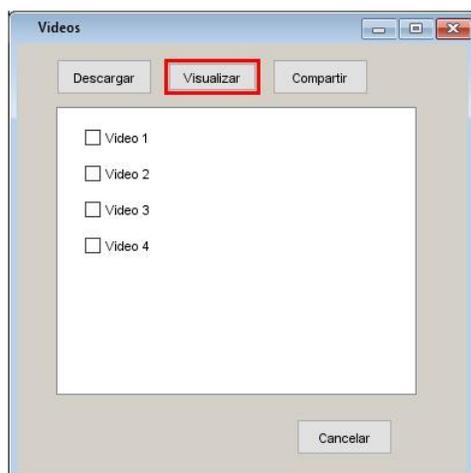
Prototipo de interfaz:

Tabla 17 HU_Adicionar a una clase de ATcnea, cuestionarios pertenecientes a un curso en ZERA 2.0

Número: 11	Nombre del requisito: Adicionar a una clase de ATcnea, cuestionarios pertenecientes a un curso en ZERA 2.0.
Programador: Ronnay Castillo Andino	Iteración Asignada: 1era
Prioridad: Alta	Tiempo Estimado: 30 días
Riesgo en Desarrollo: N/A	Tiempo Real: 28 días
Descripción:	
1- Objetivo:	
Permitir adicionar en una clase de ATcnea, cuestionarios pertenecientes a un curso de ZERA 2.0.	
2- Acciones para lograr el objetivo (precondiciones y datos):	
Para adicionar a una clase de ATcnea, cuestionarios pertenecientes a un curso en ZERA 2.0 hay que:	
<ul style="list-style-type: none"> • Estar autenticado en ATcnea con el rol Profesor. 	

3- Comportamientos válidos y no válidos (flujo central y alternos):

- Para almacenar en el banco de exámenes de una clase de ATcnea, cuestionarios procedentes de un curso de ZERA 2.0 se debe seleccionar antes al menos un cuestionario.

4- Flujo de la acción a realizar:

- El usuario debe seleccionar la opción Cuestionarios_LMS.
- En esa ventana debe seleccionar los cuestionarios que desea agregar al banco de exámenes de la clase en ATcnea.
- Si desea cancelar toda la selección y salir de la sección Cuestionarios_LMS, seleccionar el botón "Cancelar".
- Seleccionar el botón "Agregar" para adicionar los cuestionarios seleccionados.
- Los cuestionarios seleccionados se agregan al banco de exámenes de la clase en ATcnea.
- Se muestra la notificación:
- En caso de algún fallo se muestra la notificación:

Observaciones:

Prototipo de interfaz:

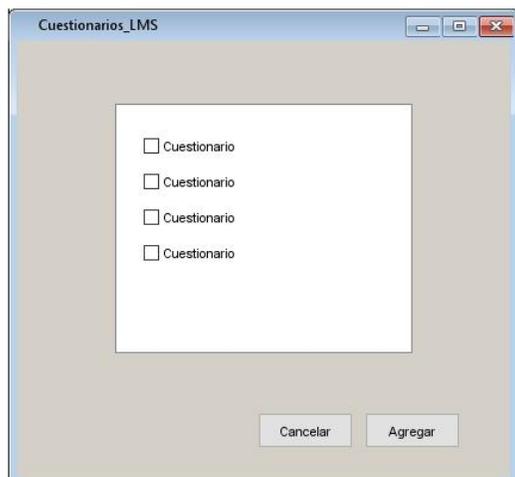
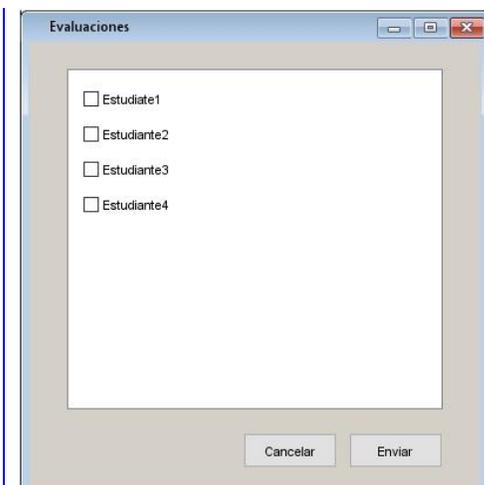


Tabla 18 HU_Enviar hacia ZERA 2.0 las evaluaciones generadas en ATcnea

Número: 12	Nombre del requisito: Enviar hacia ZERA 2.0 las evaluaciones generadas en ATcnea	
Programador: Frank Lorenzo Morfa Valdes	Iteración Asignada: 1era	
Prioridad: Alta	Tiempo Estimado: 30 días	
Riesgo en Desarrollo: N/A	Tiempo Real: 24 días	
Descripción:		
1- Objetivo:		
Permitir enviar a ZERA 2.0 para que sean almacenadas, las evaluaciones de los estudiantes en ATcnea.		
2- Acciones para lograr el objetivo (precondiciones y datos):		
Para enviar a la Plataforma Educativa ZERA 2.0 las notas de los estudiantes de una clase de ATcnea hay que:		
<ul style="list-style-type: none"> • Estar autenticado en ATcnea con el rol Profesor. 		
3- Comportamientos válidos y no válidos (flujo central y alternos):		
4- Flujo de la acción a realizar:		
<ul style="list-style-type: none"> • El usuario debe seleccionar la sección “Enviar evaluaciones hacia LMS”. • Seleccionar las evaluaciones que se desean enviar y almacenar en ZERA 2.0. • Seleccionar el botón “Cancelar” para cancelar la selección. • Seleccionar el botón “Enviar” para enviar las evaluaciones hacia ZERA 2.0 para almacenarlas. • Se mostrará la notificación: • En caso de fallo en el envío se muestra la notificación: 		
Observaciones:		
Prototipo de interfaz:		



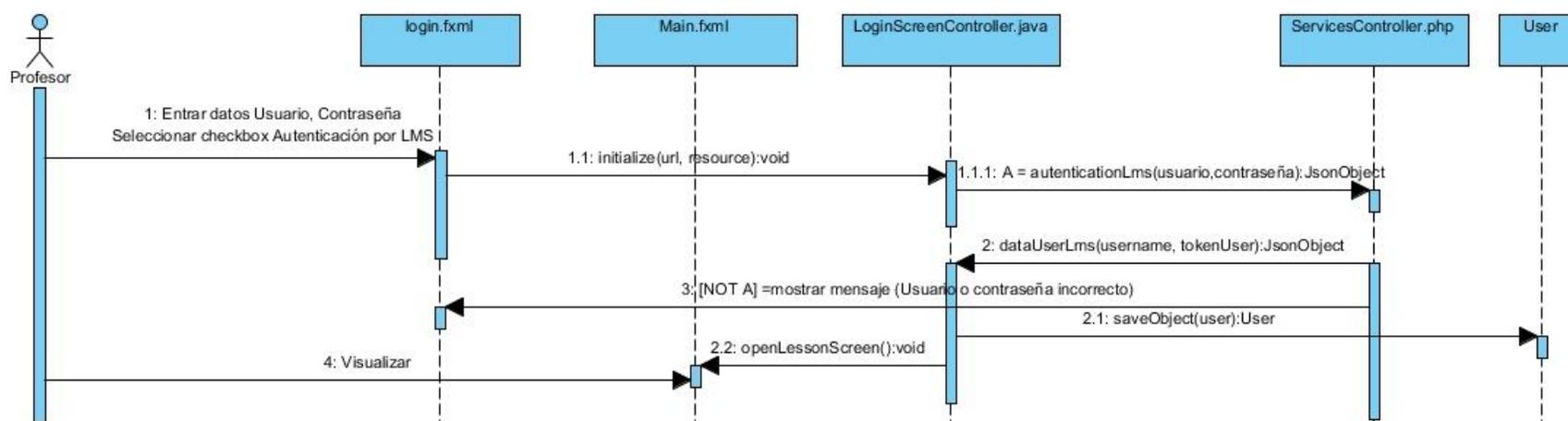


Figura 10 Diagrama de Secuencia-Authenticar usuario en ATcnea mediante el servicio web brindado por la Plataforma Educativa ZERA 2.0

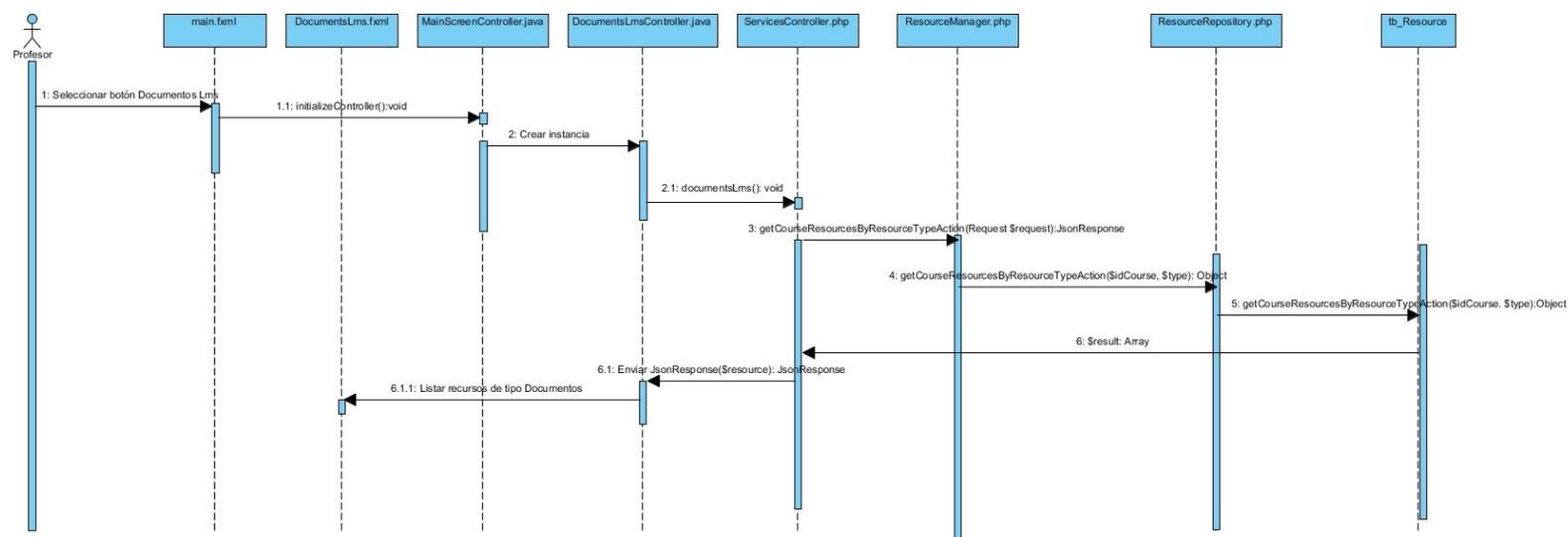


Figura 11 Diagrama de Secuencia-Listar en una clase de ATcnea, recursos de tipo documento pertenecientes a un curso en ZERA 2.0

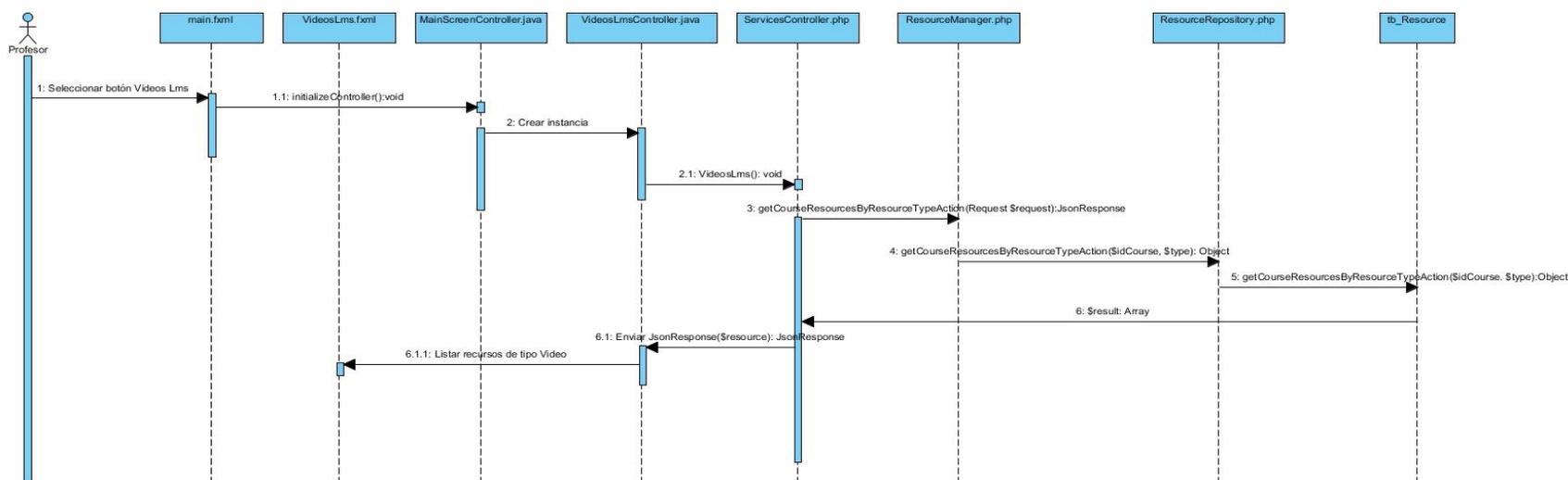


Figura 12 Diagrama de Secuencia-Listar en una clase de ATcnea, recursos de tipo video pertenecientes a un curso en ZERA 2.0

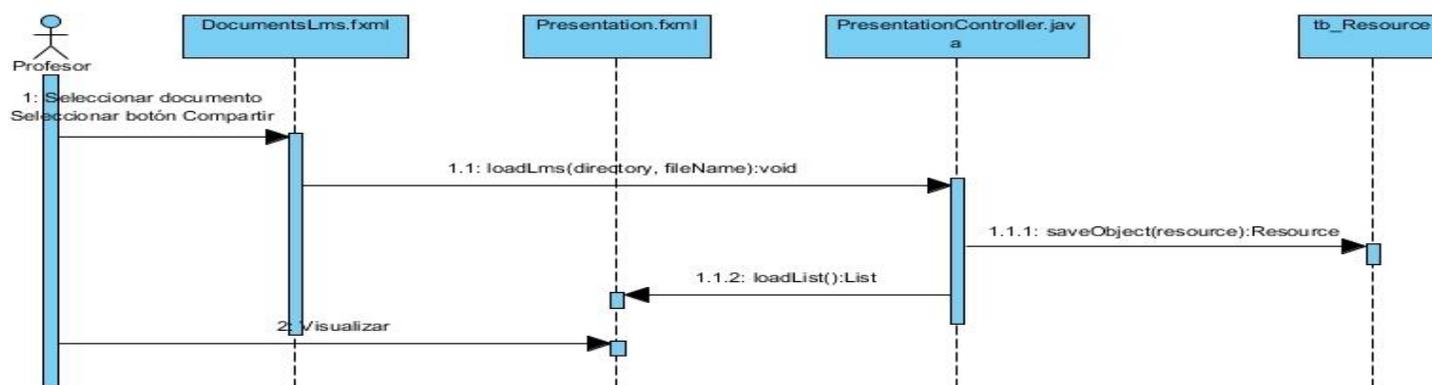


Figura 13 Diagrama de Secuencia-Compartir en una clase de ATcnea, recursos de tipo documento pertenecientes a un curso en ZERA 2.0

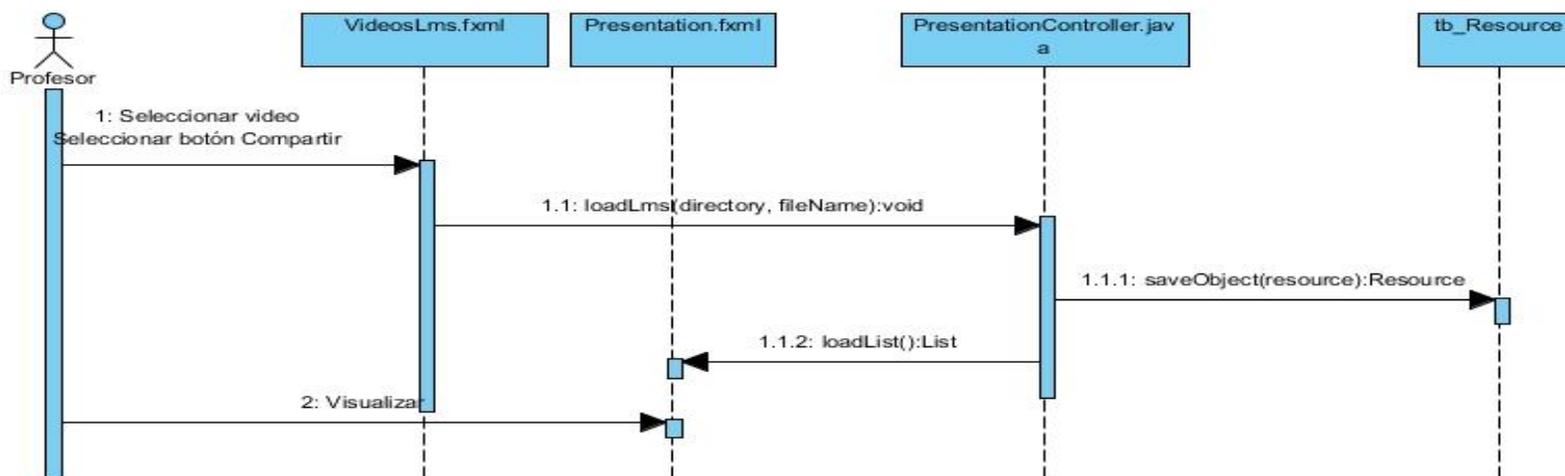


Figura 14 Diagrama de Secuencia-Compartir en una clase de ATcnea, recursos de tipo video pertenecientes a un curso en ZERA 2.0

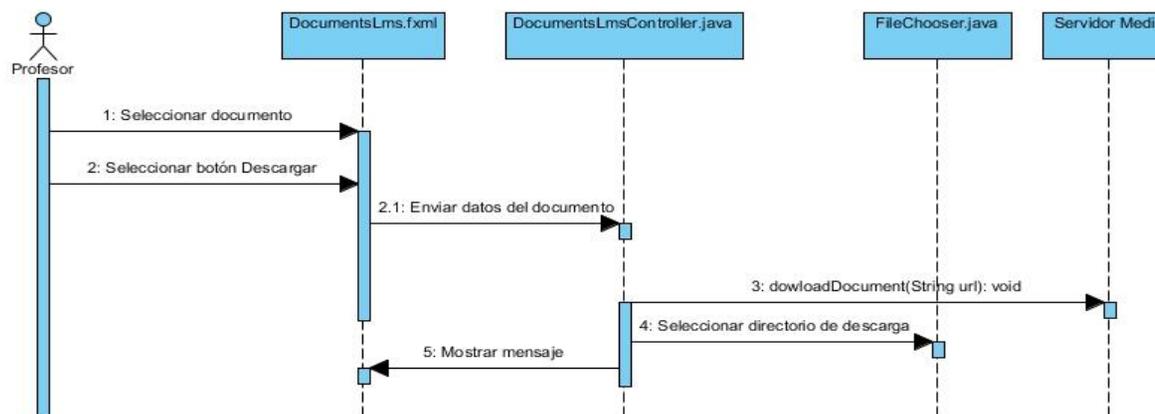


Figura 15 Diagrama de Secuencia-Descargar en una clase de ATcnea, recursos de tipo documento pertenecientes a un curso en ZERA 2.0

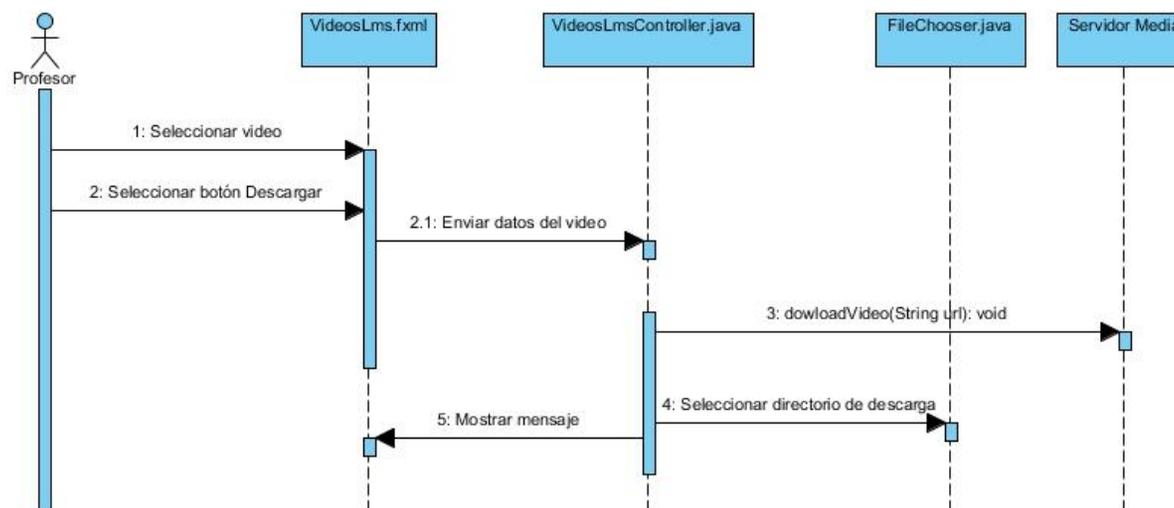


Figura 16 Diagrama de Secuencia-Descargar en una clase de ATcnea, recursos de tipo video pertenecientes a un curso en ZERA 2.0

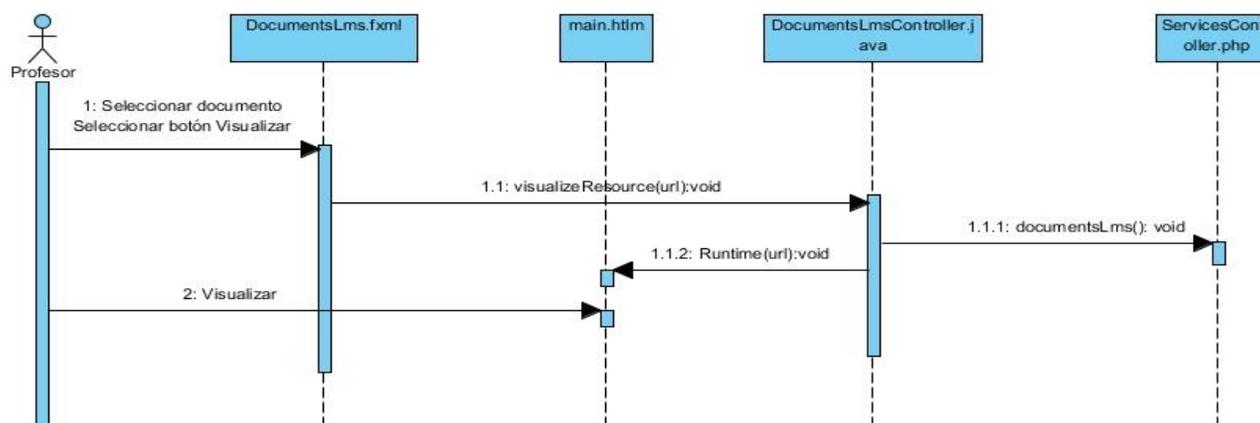


Figura 17 Diagrama de Secuencia-Visualizar en una clase de ATcnea, recursos de tipo documento pertenecientes a un curso en ZERA 2.0

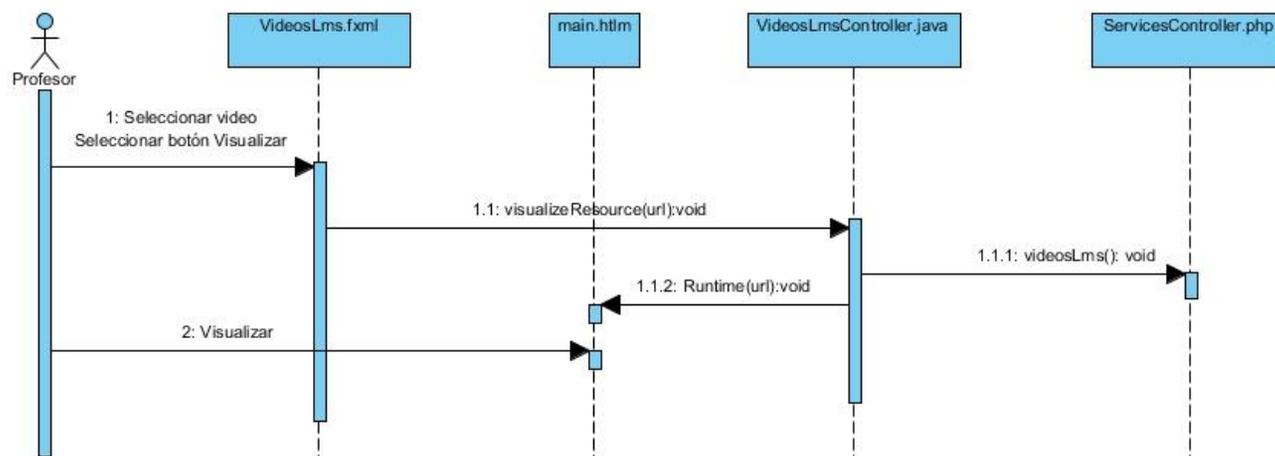


Figura 18 Diagrama de Secuencia-Visualizar en una clase de ATcnea, recursos de tipo video pertenecientes a un curso en ZERA 2.0

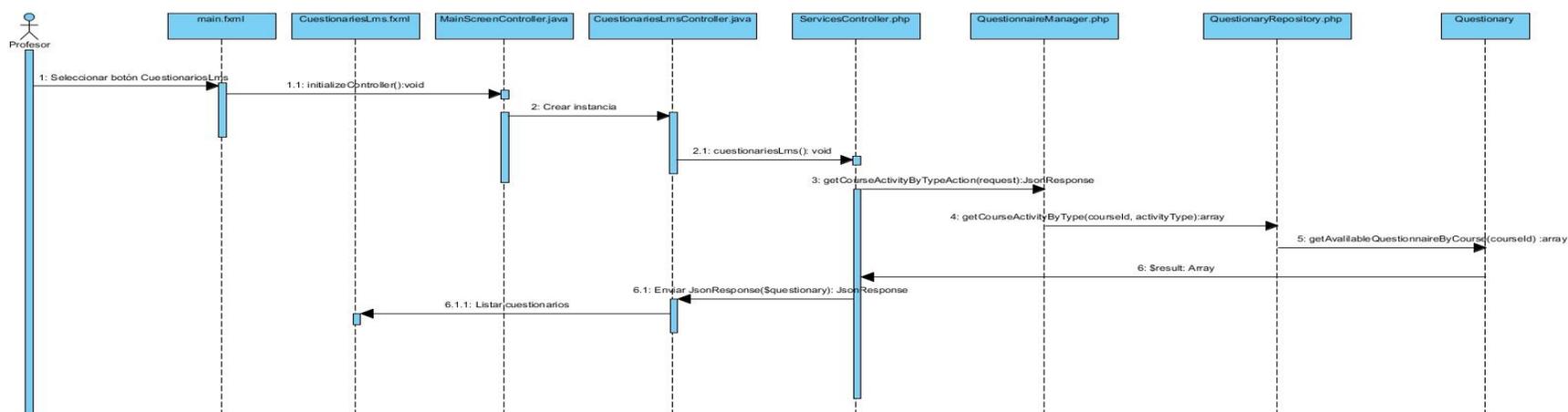


Figura 19 Diagrama de Secuencia-Adicionar a una clase de ATcnea, cuestionarios pertenecientes a un curso en ZERA 2.0

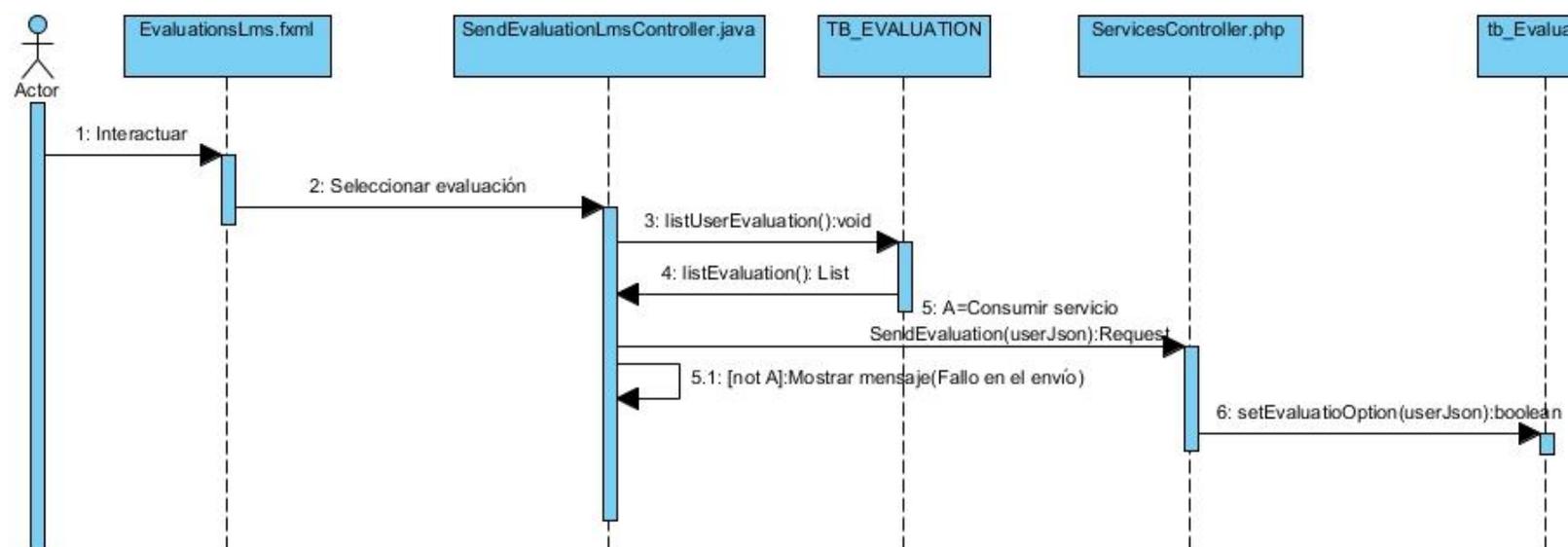


Figura 20 Diagrama de Secuencia-Enviar hacia ZERA 2.0 las evaluaciones generadas en ATcnea

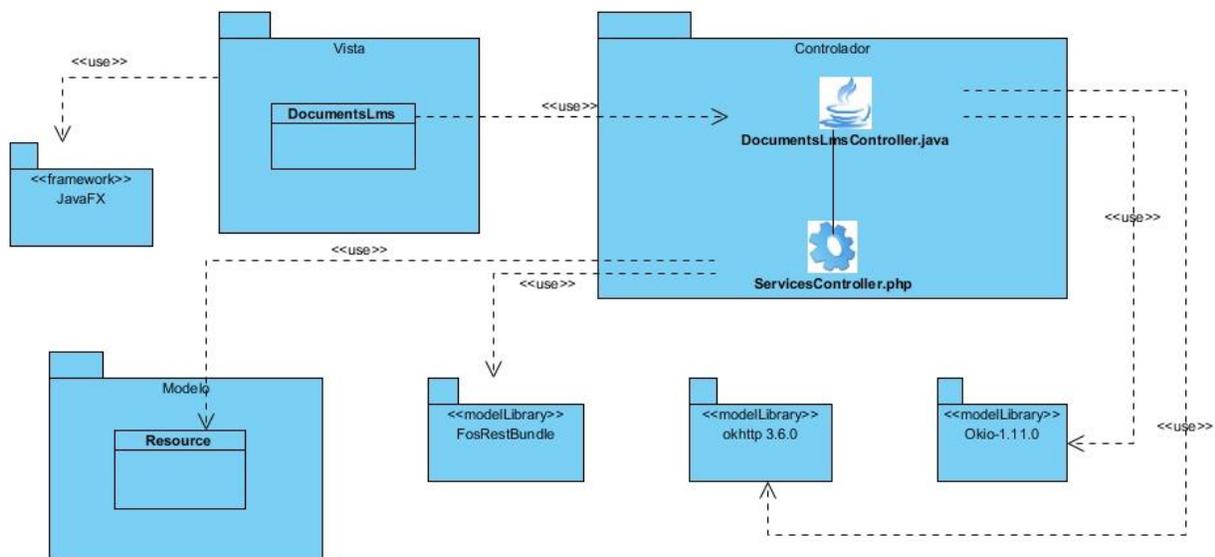


Figura 21 Diagrama de clases del diseño-Listar en una clase de ATcnea, recursos de tipo documento pertenecientes a un curso en ZERA 2.0

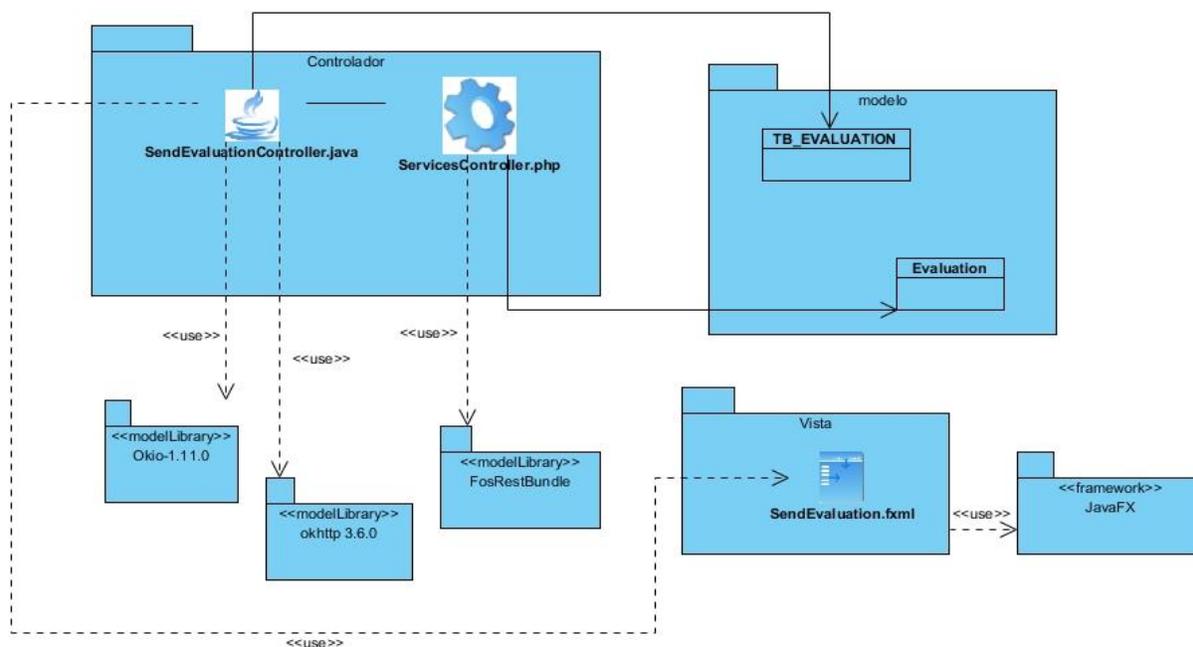


Figura 22 Diagrama de Clases del diseño-Enviar hacia ZERA 2.0 las evaluaciones generadas en ATcnea

Descripción de las tablas de la base de datos

Tabla 19 Descripción de la tabla User

User		
Descripción: en esta tabla se agrupa la información correspondiente a los datos de los usuarios de una clase correspondiente a ATcnea.		
Atributo	Tipo	Descripción
id	integer	Etiqueta única que identifica el objeto en la tabla.
connectionID	integer(10)	Identificador de la conexión.
createdAt	date	Almacena la fecha en que se registró el usuario.
enable	character varying(255)	Almacena el estado del usuario.
imagePath	character varying(255)	Almacena la dirección de la imagen.
ip	character varying(255)	Almacena el ip del usuario.
mac	character varying(255)	Almacena la mac del usuario.
password	character varying(255)	Almacena la contraseña del estudiante.
profileJson	character varying(255)	Almacena los datos del usuario.
updatedAt	date	Almacena la fecha en que el usuario actualizó sus datos.
userConfigJson	character varying(255)	Almacena la configuración del usuario.
username	character varying(255)	Almacena el nombre del usuario.

Tabla 20 Descripción de la tabla Evaluation

Evaluation		
Descripción: en esta tabla se agrupa la información correspondiente a los datos de las evaluaciones		
Atributo	Tipo	Descripción
id	integer	Etiqueta única que identifica el objeto en la tabla.
createdAt	date	Almacena la fecha en que se creó la evaluación.
description	character varying(255)	Almacena la descripción de la evaluación.
evaluation	character varying(255)	Almacena el valor de la evaluación.
updatedAt	date	Almacena la fecha en que se actualizó la evaluación.
weight	double	Almacena la evaluación por rangos.
Userid	integer	Almacena identificador único del usuario.

Tabla 21 Descripción de la tabla Group

Group		
Descripción: en esta tabla se agrupa la información correspondiente a los grupos.		
Atributo	Tipo	Descripción
id	integer	Etiqueta única que identifica el objeto en la tabla.
createdAt	date	Almacena la fecha en que se creó el grupo.
editable	boolean	Almacena si el grupo ha sido editado.

name	character varying(255)	Almacena el nombre del grupo.
updatedAt	date	Almacena la fecha en que se actualizó la evaluación.
Lessonid : integer(10)	double	Almacena el identificador único de la clase.

Tabla 22 Descripción de la tabla Resource

Resource		
Descripción: en esta tabla se agrupa la información correspondiente a los recursos.		
Atributo	Tipo	Descripción
id	integer	Etiqueta única que identifica el objeto en la tabla.
createdAt	date	Almacena la fecha en que se creó el recurso.
size	double	Almacena el tamaño del recurso.
name	character varying(255)	Almacena el nombre del recurso.
updatedAt	date	Almacena la fecha en que se actualizó la evaluación.
toSend	boolean	Almacena si el recurso fue enviado.
url	character varying(255)	Almacena la url con la dirección del recurso.

Tabla 23 Descripción de la tabla LessonSession

LessonSession		
Descripción: en esta tabla se agrupa la información correspondiente a la sesión de una clase.		
Atributo	Tipo	Descripción

id	integer	Etiqueta única que identifica el objeto en la tabla.
createdAt	date	Almacena la fecha en que se creó la sesión.
endAt	timestamp	Almacena la hora y fecha en que finaliza la clase.
startedAt	timestamp	Almacena la hora y fecha en que comienza la clase.
updatedAt	date	Almacena la fecha en que se actualizó la evaluación.
Lessonid	integer	Almacena el identificador único de la clase.

Tabla 24 Descripción de la tabla User_Lesson

User_Lesson		
Descripción: en esta tabla se agrupa la información correspondiente al usuario de una clase.		
Atributo	Tipo	Descripción
Userid	integer	Etiqueta única que identifica al usuario.
Lessonid	integer	Etiqueta única que identifica la clase.

Tabla 25 Descripción de la tabla Resource_Lesson

Resource_Lesson		
Descripción: en esta tabla se agrupa la información correspondiente al recurso de una clase.		
Atributo	Tipo	Descripción
Resourceid	integer	Etiqueta única que identifica al recurso.
Lessonid	integer	Etiqueta única que identifica la clase.

Tabla 26 Descripción de la tabla CoEvaluation/Evaluation

CoEvaluation/Evaluation		
Descripción: en esta tabla se agrupa la información correspondiente a los datos en ZERA 2.0 de las evaluaciones de tipo cuestionario y tareas.		
Atributo	Tipo	Descripción
id	integer	Etiqueta única que identifica el objeto en la tabla.
evaluation	character varying(255)	Almacena la evaluación.
observations	character varying(255)	Almacena la observación de la evaluación.
evaluation_date	date	Almacena la fecha de la evaluación.
created_at	date	Almacena la fecha en que se crea la evaluación.
updated_at	date	Almacena la fecha en que se actualiza la evaluación.
activity_id	integer	Almacena la identificación única de la actividad a la que pertenece la evaluación.
activity_type	integer	Almacena el tipo de actividad a la que pertenece la evaluación.
course_id	integer	Almacena la identificación única del curso al que pertenece la evaluación.
user_id	integer	Almacena la identificación única del usuario al que pertenece la evaluación.

Diseños de casos de prueba

Tabla 27 DCP_Autenticar un usuario en ATcnea mediante el servicio web brindado por la Plataforma Educativa Zera 2.0

Descripción general					
Permitir autenticar un usuario en ATcnea mediante el servicio web brindado por la Plataforma Educativa Zera 2.0.					
Condiciones de ejecución					
Para autenticar el usuario debe existir conexión entre ATcnea y ZERA 2.0 en ese momento.					
SC 1 Autenticar usuario					
Escenario	Descripción	Nombre	Contraseña	Respuesta del sistema	Flujo central
EC 1.1 Opción Aceptar	Accede al sistema	V	V	Autentica al usuario y accede a la clase.	Autenticación/Aceptar
EC 1.2 Datos incorrectos	Existen datos incorrectos	V	V	Se muestra la notificación: Su usuario o contraseña es incorrecto.	Autenticación/Aceptar

Tabla 28 Descripción de las variables del DCP_Autenticar un usuario en ATcnea mediante el servicio web brindado por la Plataforma Educativa ZERA 2.0

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	Nombre	campo de texto	No	Campo de carácter obligatorio que representa el nombre del usuario, puede contener caracteres alfanuméricos. Ejemplo: ghernandez

2	Contraseña	campo de selección	No	Campo de carácter obligatorio que representa la contraseña del usuario, puede contener cualquier tipo de caracter. Ejemplo: 1L4m@s
---	------------	--------------------	----	--

Tabla 29 DCP_Crear en ATcnea las clases respectivas a los cursos en ZERA 2.0

Descripción general			
Permitir crear en ATcnea las clases respectivas a los cursos en ZERA 2.0			
Condiciones de ejecución			
Estar autenticado en ATcnea con el rol Profesor.			
SC 1 Crear clase			
Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Clases creadas correctamente	Las clases se han creado correctamente.	Se muestra la notificación: Se han creado las clases en ATcnea contenidas en el LMS.	MainScreenLesson
EC 1.2 Siglas LMS	Las siglas se mostrarán al final del nombre de las clases provenientes de ZERA 2.0.	Se mostrarán las siglas LMS al final del nombre de las clases. Ejemplo: Historia_LMS	MainScreenLesson
EC 1.3 Las clases no han sido creadas	Fallo en la creación de clases	Se muestra la notificación: Se produjo un error de conexión al intentar crear las clases en ATcnea desde el LMS.	MainScreenLesson

Tabla 30 DCP_Listar en una clase de ATcnea, recursos de tipo video pertenecientes a un curso en ZERA 2.0

Descripción general
Permitir listar en una clase de ATcnea, recursos de tipo video pertenecientes a un curso en ZERA 2.0
Condiciones de ejecución
Estar autenticado en ATcnea con el rol Profesor.

SC 1 Listar videos

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Opción Videos_LMS	Selecciona la sección Videos_LMS que permite listar los videos que pertenecen al curso de ZERA 2.0.	Abre la sección Videos_LMS y lista los videos.	MainScreenLesson/Videos_LMS
EC 1.3 Listado fallido	Existe un error en listar los videos.	Muestra la notificación: Se produjo un error en la conexión al intentar listar los recursos.	MainScreenLesson/Videos_LMS

Tabla 31 DCP_Listar en una clase de ATcnea, recursos de tipo documento pertenecientes a un curso en ZERA 2.0

Descripción general

Permitir listar en una clase de ATcnea, recursos de tipo documento pertenecientes a un curso en ZERA 2.0

Condiciones de ejecución

Estar autenticado en ATcnea con el rol Profesor.

SC 1 Listar documentos

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Opción Documentos_LMS	Selecciona la sección Documentos_LMS que permite listar los documentos que pertenecen al curso de ZERA 2.0.	Abre la sección Documentos_LMS y lista los documentos.	MainScreenLesson/Documentos_LMS
EC 1.3 Listado fallido	Existe un error en listar los documentos.	Muestra la notificación: Se produjo un error en la conexión al intentar listar los recursos.	MainScreenLesson/Documentos_LMS

Tabla 32 DCP_Compartir en una clase de ATcnea, recursos de tipo video pertenecientes a un curso en ZERA 2.0

Descripción general			
Permitir compartir en una clase de ATcnea, recursos de tipo video pertenecientes a un curso en ZERA 2.0			
Condiciones de ejecución			
Estar autenticado en ATcnea con el rol Profesor.			
Las extensiones de los videos que soporta para compartir son mpeg, mp4 y flv			
SC 1 Compartir video			
Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Opción Compartir	Selecciona la opción Compartir que permite añadir un video a la sección Compartir video.	El botón "Compartir" solo se activará si se selecciona un video de la lista. El video se añade a la sección Compartir video.	Videos_LMS/Compartir
EC 1.2 Opción Cancelar	Selecciona la opción Cancelar	Regresa a la vista principal de la clase.	Videos_LMS/Cancelar

Tabla 33 DCP_Compartir en una clase de ATcnea, recursos de tipo documento pertenecientes a un curso en ZERA 2.0

Descripción general			
Permitir compartir en una clase de ATcnea, recursos de tipo documento pertenecientes a un curso en ZERA 2.0			
Condiciones de ejecución			
Estar autenticado en ATcnea con el rol Profesor.			
Las extensiones de los documentos que soporta para compartir son ppt, pptx y pdf.			
SC 1 Compartir documento			
Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Opción Compartir	Selecciona la opción Compartir que permite añadir un documento a la	El botón "Compartir" solo se activará si se selecciona un documento de la lista. El documento se añade a la	Documentos_LMS/Compartir

	sección Presentación.	sección Presentación.	
EC 1.2 Opción Cancelar	Selecciona la opción Cancelar	Regresa a la vista principal de la clase.	Documentos_LMS/Cancelar

Tabla 34 DCP_Descargar en una clase de ATcnea, recursos de tipo documento pertenecientes a un curso en ZERA 2.0

<p>Descripción general Permitir descargar en una clase de ATcnea, recursos de tipo documento pertenecientes a un curso en ZERA 2.0</p> <p>Condiciones de ejecución Estar autenticado en ATcnea con el rol Profesor.</p> <p>SC 1 Descargar documentos</p>			
Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Opción Descargar	Selecciona la opción Descargar que permite descargar un documento.	El botón "Descargar" solo se activará si se selecciona un documento de la lista. Brinda la posibilidad de escoger el directorio donde se desee descargar el documento.	Documentos_LMS/Descargar/Seleccionar_directorio/Aceptar
EC 1.2 Descarga exitosa	La descarga se ha realizado correctamente	Muestra la notificación: Archivo descargado exitosamente.	Documentos_LMS/Descargar/Seleccionar_directorio/Aceptar
EC 1.3 Descarga fallida	Existe un error en la descarga	Muestra la notificación: La descarga ha fallado.	Documentos_LMS/Descargar/Seleccionar_directorio/Aceptar
EC 1.4 Opción Cancelar	Selecciona la opción Cancelar	Regresa a la vista principal de la clase.	Documentos_LMS/Cancelar

Tabla 35 DCP_Visualizar en una clase de ATcnea, recursos de tipo documento pertenecientes a un curso en ZERA 2.0

<p>Descripción general Permitir visualizar en una clase de ATcnea, recursos de tipo documento pertenecientes a un curso en ZERA 2.0</p>			
<p>Condiciones de ejecución Estar autenticado en ATcnea con el rol Profesor. La extensión para los documentos es pdf.</p>			
<p>SC 1 Visualizar documento</p>			
Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Opción Visualizar	Selecciona la opción Visualizar que permite visualizar un documento.	El botón "Visualizar" solo se activará si se selecciona un documento de la lista. Brinda la posibilidad de visualizar un documento de la lista con el navegador.	Documentos_LMS/Visualizar
EC 1.3 Visualización fallida	Existe un error en la visualización.	Muestra la notificación: Se produjo un error en la conexión a la hora de ver el documento.	Documentos_LMS/Visualizar
EC 1.4 Opción Cancelar	Selecciona la opción Cancelar.	Regresa a la vista principal de la clase.	Documentos_LMS/Cancelar

Tabla 36 DCP_Visualizar en una clase de ATcnea, recursos de tipo video pertenecientes a un curso en ZERA 2.0

<p>Descripción general Permitir visualizar en una clase de ATcnea, recursos de tipo video pertenecientes a un curso en ZERA 2.0</p>			
<p>Condiciones de ejecución Estar autenticado en ATcnea con el rol Profesor. Las extensiones para los videos son mpeg, mp4, flv, ogg y webm.</p>			
<p>SC 1 Visualizar video</p>			

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Opción Visualizar	Selecciona la opción Visualizar que permite visualizar un video.	El botón "Visualizar" solo se activará si se selecciona un video de la lista. Brinda la posibilidad de visualizar un video de la lista con el navegador.	Videos_LMS/Visualizar
EC 1.3 Visualización fallida	Existe un error en la visualización.	Muestra la notificación: Se produjo un error en la conexión a la hora de ver el video.	Videos_LMS/Visualizar
EC 1.4 Opción Cancelar	Selecciona la opción Cancelar.	Regresa a la vista principal de la clase.	Videos_LMS/Cancelar

Tabla 37 DCP_Adicionar a una clase de ATcnea, cuestionarios pertenecientes a un curso en ZERA 2.0

Descripción general				
Permitir adicionar a una clase de ATcnea, cuestionarios pertenecientes a un curso en ZERA 2.0				
Condiciones de ejecución				
Estar autenticado en ATcnea con el rol Profesor.				
SC 1 Adicionar cuestionarios				
Escenario	Descripción	Campo	Respuesta del sistema	Flujo central
EC 1.1 Opción Cuestionarios_LMS	Permite listar en la sección Cuestionarios_LMS los cuestionarios pertenecientes a un curso de ZERA 2.0.	N/A	Abre la sección Cuestionarios_LMS y lista los cuestionarios.	MainScreenLesson/Cuestionarios_LMS
EC 1.2 Opción Adicionar	Permite adicionar los cuestionarios seleccionados a la sección Banco de exámenes.	V	Añade a la sección Banco de exámenes los cuestionarios seleccionados.	MainScreenLesson/Cuestionarios_LMS/Adicionar
EC 1.3 Opción Cancelar	Permite abandonar la sección Cuestionarios_L	N/A	Regresa a la sección principal de la clase.	MainScreenLesson/Cuestionarios_LMS/Cancelar

MS

Tabla 38 Descripción de variables del DCP_Agregar a una clase de ATcnea, cuestionarios pertenecientes a un curso en ZERA 2.0

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	Campo	campo de selección	Sí	Campo que representa a los cuestionarios pertenecientes a ZERA 2.0, es un campo de selección. Ejemplo: Cuestionario de Historia

Tabla 39 DCP_Enviar hacia ZERA 2.0 las evaluaciones generadas en ATcnea

Descripción general				
Permitir enviar hacia ZERA 2.0 las evaluaciones generadas en ATcnea				
Condiciones de ejecución				
Estar autenticado en ATcnea con el rol Profesor.				
SC 1 Enviar evaluaciones				
Escenario	Descripción	Campo	Respuesta del sistema	Flujo central
EC 1.1 Opción Enviar Evaluaciones hacia LMS	Permite listar en la sección Enviar Evaluaciones hacia LMS las evaluaciones de los cuestionarios provenientes de ZERA 2.0	N/A	Abre la sección Enviar Evaluaciones hacia LMS y lista las evaluaciones de los cuestionarios provenientes de ZERA 2.0.	MainScreenLesson/Enviar Evaluaciones hacia LMS
EC 1.2 Opción Enviar	Permite enviar hacia ZERA 2.0 las evaluaciones seleccionadas.	V	Envía hacia ZERA 2.0 las evaluaciones de los cuestionarios pertenecientes a ZERA 2.0.	MainScreenLesson/Enviar Evaluaciones hacia LMS/Enviar
EC 1.3 Opción Cancelar	Permite abandonar la sección Enviar Evaluaciones hacia LMS.	N/A	Regresa a la sección principal de la clase.	MainScreenLesson/Enviar Evaluaciones hacia LMS/Cancelar

Tabla 40 Descripción de variables del DCP_Enviar hacia ZERA 2.0 las evaluaciones generadas en ATcnea

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	Campo	campo de selección	No	Campo que representa a las evaluaciones de los cuestionarios pertenecientes a ZERA 2.0, es un campo de selección. Ejemplo: Cuestionario de Historia