

# Universidad Austral de Chile

Facultad de Ciencias de la Ingeniería  
Escuela de Ingeniería Civil en Informática

## “QUIZ: UNA FUNCIONALIDAD PARA AQRATE”

Tesis para optar al título de:  
Ingeniero Civil en Informática

Profesor Patrocinante:  
Sr. Luis Alberto Álvarez González  
Ingeniero Civil Electricista  
Magíster en Ingeniería Informática

Profesor Informante:  
Sra. Alicia Campos Hebrero  
Ingeniero Técnico en Informática de Gestión

Profesor Informante:  
Srta. Natalia Escobar Lovera  
Ingeniero Civil en Informática

RODRIGO ANDRÉS MORALES ORTEGA  
VALDIVIA – CHILE  
2008



# Universidad Austral de Chile

Instituto de Informática

Valdivia, 6 de octubre del 2008.

Luis Alberto Álvarez González.  
Patrocinante

Juan Pablo Salazar Fernández.  
Director Escuela de Ingeniería Civil en Informática

Ref: Calificación proyecto de título

consideración:

Habiendo revisado el trabajo de titulación “**QUIZ: UNA FUNCIONALIDAD AQUARATE**”, presentado por el estudiante **SR. RODRIGO ANDRÉS MORALES GA**, mi evaluación es la siguiente:

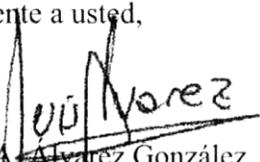
**Nota: 7,0 (SIETE COMA CERO).**

#### Fundamento de la nota:

El sistema de Titulación se enmarcó dentro del proyecto AquRate, uno de los pocos proyectos de investigación a nivel mundial que desarrolla una herramienta de evaluación para la última versión del estándar IMS QTI. La funcionalidad que desarrolló era una parte importante del proyecto y para ello se incorporó por ocho semanas al equipo de investigación en la Universidad de Kingstone, Londres, donde su trabajo fue muy bien valorado por los investigadores responsables del proyecto. Muy buen lenguaje técnico.

Aspecto	Evaluación
Cumplimiento de objetivos	7
Satisfacción de alguna necesidad	7
Aplicación del método científico	7
Interpretación de los datos y obtención de conclusiones	7
Originalidad	7
Aplicación de criterios de análisis y diseño	7
Perspectivas del trabajo	7
Coherencia y rigurosidad lógica	7
Precisión del lenguaje técnico	7

Sin otro particular, saluda atentamente a usted,

  
Luis A. Álvarez González  
Instituto de Informática

Kingston Upon Thames, 24 de Octubre del 2008

De: Alicia María Campos Hebrero  
Informante

A: Juan Pablo Salazar Fernández  
Director  
Escuela de Ingeniería Civil en Informática

Ref: Calificación proyecto de título

De mi consideración:

Habiendo revisado el trabajo de titulación **“QUIZ: UNA FUNCIONALIDAD PARA AQURATE”**, presentado por el alumno **SR. RODRIGO ANDRÉS MORALES ORTEGA**, mi evaluación del mismo es la siguiente:

**Nota: 7,0 (SIETE COMA CERO).**

**Fundamento de la nota:**

En la tesis de su titulación el estudiante demuestra haber adquirido un amplio conocimiento en lo referente a las tecnologías de aprendizaje y los estándares actuales que les dan soporte. Define y explica coherentemente la estructura y características de la especificación IMS QTI 2.1 en relación con la herramienta AQURATE y los otros dos proyectos relacionados, ASDEL y MINIBIX.

Como miembro del equipo de desarrollo del proyecto AQURATE, he seguido de cerca el trabajo de Rodrigo durante su estancia en nuestra facultad. La funcionalidad que se requería fue implementada bajo un proceso de diseño rápido, eficaz y documentado, cumpliendo así los objetivos que se habían definido para su proyecto fin de carrera y las expectativas del equipo.

<b>Aspecto</b>	<b>Evaluación</b>
Cumplimiento de objetivos	7
Satisfacción de alguna necesidad	7
Aplicación del método científico	7
Interpretación de los datos y obtención de conclusiones	7
Originalidad	7
Aplicación de criterios de análisis y diseño	7
Perspectivas del trabajo	7
Coherencia y rigurosidad lógica	7
Precisión del lenguaje técnico	7

Sin otro particular, saluda atentamente a usted,

A handwritten signature in black ink, appearing to read 'Alicia M. Campos Hebrero', is written over a solid horizontal line.

Alicia M. Campos Hebrero  
Learning Technologies Group  
Faculty of Computing, Information  
Systems and Mathematics  
Kingston university

Valdivia, 27 de Octubre de 2008

De: Natalia Escobar Lovera  
Co-patrocinante

A: Juan Pablo Salazar Fernández  
Director  
Escuela de Ingeniería Civil en Informática

Ref: Calificación proyecto de título

De mi consideración:

Habiendo revisado el trabajo de titulación **“QUIZ: Una funcionalidad para AQURATE”**, presentado por el alumno **Sr. Rodrigo Andrés Morales Ortega**, mi evaluación del mismo es la siguiente:

**Nota:** 7,0 (Siete coma cero).

**Fundamento de la nota:**

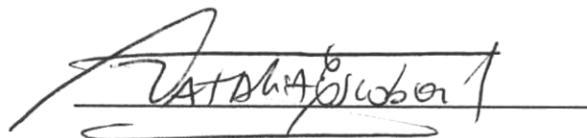
Se observa un estudio acabado de conceptos relacionados a Tecnologías del Aprendizaje y a sus estándares asociados, así como un análisis de la arquitectura y de las funcionalidades del proyecto AQURATE.

Las adecuaciones realizadas al proyecto AQURATE y el desarrollo de nuevas funcionalidades, cumplen con todas las etapas de un proyecto de desarrollo de software, análisis, diseño, implementación y testeó.

Finalmente cabe destacar que este proyecto cumple con los objetivos planteados inicialmente y que la participación del estudiante a cargo, refleja profesionalismo, seriedad y por sobre todo compromiso con el trabajo realizado en la Universidad de Kignston.

Aspecto	Evaluación
Cumplimiento de objetivos	7.0
Satisfacción de alguna necesidad	7.0
Aplicación del método científico	7.0
Interpretación de los datos y obtención de conclusiones	7.0
Originalidad	7.0
Aplicación de criterios de análisis y diseño	7.0
Perspectivas del trabajo	7.0
Coherencia y rigurosidad lógica	7.0
Precisión del lenguaje técnico	7.0

Sin otro particular, saluda atentamente a usted,



Natalia Escobar Lovera  
Docente del Instituto de Informática  
Universidad Austral de Chile

Professor Luis Alvarez-Gonzalez  
Universidad Austral de Chile (UACH)  
Luis Alberto Álvarez González  
Director Instituto de Informática  
Universidad Austral de Chile General Lagos 2086  
Campus Miraflores  
Valdivia  
Chile  
Fono : 56 - 63 - 221 427  
Fax : 56 - 63 - 293 115  
email : [lalvarez@uach.cl](mailto:lalvarez@uach.cl)

1<sup>st</sup> August 2008

Dear Luis

**Reference for Rodrigo Morales-Ortega**

As you know Rodrigo came to visit our University between the 8<sup>th</sup> June and 1<sup>st</sup> August 2008. During this visit he contributed to research into authoring and rendering QTI 2.1 interaction items. Specifically he greatly aided the grouping of items and rendering of groups of items for the AQuRate project.

His input was impressive, and was of benefit to the Learning Technology Research Group. The output of the research can be seen at [aqr.rate.kingston.ac.uk](http://aqr.rate.kingston.ac.uk). His programming skills were of a high level of competence, as was his ability to grasp how the authoring tool and project worked.

He has contributed to not only the functionality of the tool, but also its quality.

If I can be of any further help please do contact me.

The University accepts no liability, in negligence or otherwise, for the statements or information contained in this reference although they are provided in good faith.

Yours sincerely



**Graham Alsop**  
Learning Technology Research Group

## ÍNDICE

ÍNDICE DE FIGURAS.....	4
Resumen.....	6
Summary .....	7
<b>1. INTRODUCCIÓN .....</b>	<b>8</b>
1.1 Motivación .....	8
1.2 Metodología .....	9
1.3 Objetivos .....	9
1.3.1 Objetivos Generales .....	9
1.3.2 Objetivos Específicos.....	9
1.4 Convenciones de Escritura.....	10
1.5 Organización del Documento.....	10
<b>2. TECNOLOGÍAS DE APRENDIZAJE, ESTÁNDARES Y ESPECIFICACIONES ..</b>	<b>11</b>
2.1 ¿Qué es un Estándar? .....	11
2.2 Ventajas Aportadas por los Estándares a las Tecnologías de Aprendizaje .....	12
2.3 Organismos e Instituciones que Participan en los Procesos de Estandarización .....	14
2.3.1 Comité de Capacitación Basado en Computador de la Industria de Aviación ...	14
2.3.2 Aprendizaje Distribuido Avanzado (ADL).....	15
2.3.3 Estructura de las Especificaciones de IMS .....	15
2.3.4 Comité de Estándares para Tecnologías de Aprendizaje del IEEE.....	16
2.3.5 La Alianza ARIADNE.....	17
2.4 Estándares y Especificaciones más Utilizados.....	17
2.4.1 Especificación IMS <i>Content Packaging</i> .....	18
2.4.2 Especificación IMS <i>Learning Design</i> .....	20
2.4.3 Especificación IMS <i>Learner Information Packaging</i> .....	20
2.4.4 Especificación IMS <i>Simple Sequencing</i> .....	21
2.4.5 Estándar IEEE LOM .....	21
2.4.6 Modelo de Referencia SCORM .....	22
2.4.7 Especificación IMS Vocabulary Definition and Exchange .....	24
<b>3. ESPECIFICACIÓN IMS QUESTION AND TEST INTROPERABILITY.....</b>	<b>25</b>
3.1 Introducción .....	25
3.2 Historia de IMS QTI .....	26
3.3 Conceptos Básicos de IMS QTI V 2.X.....	28
3.3.1 Las Preguntas .....	28
3.3.2 Interacciones .....	30
3.3.2.1 Interacciones Simples .....	31
3.3.2.2 Interacciones de Texto .....	33
3.3.2.3 Interacciones Gráficas.....	35
3.3.2.4 Otros Tipos de Interacciones.....	40
3.3.3 Exámenes en IMS QTI V 2.1 .....	41
3.3.4 Resultados y Estadísticas .....	42
3.4 Conceptos Avanzados .....	43
3.4.1 Intentos y Sesiones de Interacción .....	44
3.4.2 Preguntas Dinámicas.....	44
3.4.3 Exámenes Dinámicos.....	46
3.4.4 Bancos de preguntas.....	47
3.5 Relación con otras Especificaciones .....	48

3.6 Herramientas Relacionadas.....	49
4. EL PROYECTO AQURATE.....	51
4.1 Introducción .....	51
4.2 Descripción del Proyecto .....	52
4.3 Integración de AQURATE con ASDEL y MINIBIX.....	54
4.3.1 El Proyecto ASDEL.....	55
4.3.2 El Proyecto MINIBIX.....	58
4.4 Características de AQURATE .....	59
4.4.1 El Modelo de Datos.....	60
4.4.2 Arquitectura de AQURATE.....	63
4.4.3 La Interfaz de AQURATE .....	67
4.4.4 El <i>renderer</i> de AQURATE .....	68
4.5 Funcionalidades Principales de AQURATE.....	71
4.5.1 Creación de Preguntas.....	72
4.5.2 Guardar Preguntas.....	73
4.6 Limitaciones de AQURATE.....	75
5. LA FUNCIONALIDAD QUIZ.....	76
5.1 Introducción .....	76
5.2 Desarrollo de la Funcionalidad .....	77
5.2.1 Modificación del <i>Content Package</i> .....	77
5.2.2 Interfaz de la Funcionalidad.....	78
5.2.3 Funcionalidad Preview QUIZ .....	79
5.2.4 Funcionalidad Generate QUIZ.....	83
5.2.5 Modificación del Menú de AQURATE.....	84
5.3 Errores Encontrados.....	85
5.4 Actualización del Código.....	86
5.5 Herramientas Utilizadas .....	87
6. CONCLUSIONES .....	88
6.1 Trabajos Futuros .....	88
7. REFERENCIAS BIBLIOGRÁFICAS.....	90
8. ANEXOS DIGITALES.....	94
8.1 Anexo Digital A .....	94
8.2 Anexo Digital B .....	94
8.3 Anexo Digital C .....	94

## ÍNDICE DE FIGURAS

Figura 2.1: Estructura archivo PIF. IMS CP .....	19
Figura 3.1: Estructura de un <i>assessmentItem</i> y su proceso de evaluación. ....	29
Figura 3.2: AssesmentItem XML en QTI. ....	30
Figura 3.3: Ejemplo de <i>choiceInteraction</i> . ....	31
Figura 3.4: Ejemplo de <i>orderInteraction</i> . ....	32
Figura 3.5: Ejemplo de <i>associateInteraction</i> .....	32
Figura 3.6: Ejemplo de <i>matchInteraction</i> . ....	33
Figura 3.7: Ejemplo de <i>gapMatchInteraction</i> . ....	33
Figura 3.8: Ejemplo de <i>inlineChoiceInteraction</i> . ....	34
Figura 3.9: Ejemplo de <i>textEntryInteraction</i> . ....	34
Figura 3.10: Ejemplo de <i>extendedTextInteraction</i> .....	35
Figura 3.11: Ejemplo de <i>hottextInteraction</i> . ....	35
Figura 3.12: Ejemplo de <i>hotspotInteraction</i> .....	36
Figura 3.13: Ejemplo de <i>selectpointInteraction</i> .....	36
Figura 3.14: Ejemplo de <i>graphicOrderInteraction</i> .....	37
Figura 3.15: Ejemplo de <i>graphicAssociateInteraction</i> .....	38
Figura 3.16: Ejemplo de <i>graphicGapMatchInteraction</i> .....	39
Figura 3.17: Ejemplo de <i>positionObjectInteraction</i> .....	40
Figura 3.18: Ejemplo de <i>sliderInteraction</i> .....	40
Figura 3.19: Estructura de un examen ( <i>assessmentTest</i> ) completo.....	42
Figura 3.20: Ejemplo de examen dinámico.....	47
Figura 3.21: Paquete IMS <i>Content Packaging</i> .....	48
Figura 4.1: Equipo de trabajo del proyecto AQURATE.....	54
Figura 4.2: Integración de Aqurate con los proyectos relacionados .....	55
Figura 4.3: Visión General del procesamiento de un Test en ASDEL .....	56
Figura 4.4: Integración de ASDEL con Aqurate y Minibix.....	58
Figura 4.5: Vinculación del esquema a través del compilador de JAXB.....	62
Figura 4.6: Esquema gráfico de <i>Unmarshalling</i> .....	62
Figura 4.7: Esquema gráfico de <i>Marshalling</i> .....	63
Figura 4.8: JAXB y QTI. ....	63
Figura 4.9: Vista de edición de una pregunta en AQURATE.....	65
Figura 4.10: Vista XML de una pregunta en AQURATE .....	65
Figura 4.11: Arquitectura de AQURATE .....	66
Figura 4.12: Organización de clases de las principales clases de AQURATE .....	67
Figura 4.13: Diagrama de secuencia del Inicio de AQURATE .....	67
Figura 4.14: Interfaz de AQURATE. ....	69
Figura 4.15: Diagrama de Secuencia del funcionamiento de Aqurate y el <i>renderer</i> . ....	72
Figura 4.16: Organización de archivos generados por el <i>renderer</i> . ....	72
Figura 4.17: Funcionalidades de AQURATE .....	73
Figura 4.18: Tipos de pregunta disponibles en AQURATE para su creación .....	73
Figura 4.19: Diagrama de Secuencia de la creación de una pregunta.....	74
Figura 4.20: Diagrama de Actividad de la creación de un CP. ....	75
Figura 4.21: Contenido del CP creado. ....	76
Figura 5.1: Varias preguntas en un CP .....	78
Figura 5.2: Funcionalidad <i>Save Pool</i> . ....	79
Figura 5.3: Interfaz de la funcionalidad QUIZ.....	80
Figura 5.4: Nueva organización de archivos generados por el <i>renderer</i> . ....	81
Figura 5.5: Diagrama de secuencia extendido de la funcionalidad Preview QUIZ. ....	82
Figura 5.6: Prototipo de la página Web generada por la funcionalidad QUIZ .....	83

Figura 5.7: Vista en el explorador de la funcionalidad QUIZ.....	84
Figura 5.8: Generación del QUIZ en un archivo formato ZIP.....	85
Figura 5.9: Nuevo menú de AQURATE.....	86

## Resumen

El presente proyecto tiene como objetivo principal el desarrollo de una funcionalidad para la herramienta AQURATE, desarrollada por la Universidad de Kingston en Londres, Inglaterra. La funcionalidad es llamada QUIZ ya que permite al usuario crear una secuencia de un determinado número de preguntas el cual puede ser distribuido a los usuarios y ser visualizado en un explorador Web.

Para llevar a cabo el desarrollo de la funcionalidad, fue necesario el estudio de los principales estándares y especificaciones con los que interactúa la herramienta, específicamente la especificación IMS QTI (*Question and Test Interoperability*). Esta especificación contempla una estructura básica que define la forma de representar preguntas individuales y gestionar evaluaciones o exámenes completos.

El estudio de la herramienta, así como el desarrollo de la funcionalidad, se realizaron en la Universidad de Kingston junto al equipo de trabajo y desarrolladores de AQURATE. Es por esto que este trabajo detalla el proyecto AQURATE en cuanto a su arquitectura así como la interacción con otros proyectos.

Finalmente se detalla cómo fue implementada la funcionalidad y los detalles que fueron encontrados durante el desarrollo.

## **Summary**

This project aims at the development of a functionality for the AQURATE tool, developed by the University of Kingston in London, England. The Functionality is called QUIZ since it allows the user creating a sequence of a determined number of questions which can be distributed to the users and being visualized in a Web explorer.

In order to accomplish the development of the functionality, the study of the main standards and specifications that the tool interacts with was necessary, specifically the IMS QTI (Question and Test Interoperability) specification. This specification contemplates a basic structure that defines the way to represent individual questions and to try to obtain evaluations or complete assessments.

The study of the tool, as well as the development of the functionality, they were realized in the University of Kingston together with the team of work and developers of AQURATE. It is for this that this work details the project AQURATE as for its architecture as well as the interaction with other projects.

Finally there is detailed how it was implemented the functionality and the details that were found during the development.

# 1. INTRODUCCIÓN

Los últimos años han visto el crecimiento enorme del uso de tecnologías en educación, ya en 1999 se consideraba según [Kin99] como una disciplina a las Tecnologías de Aprendizaje. Aunque en esos años el uso de tecnologías en educación no era nuevo, la integración de varias tecnologías en las instituciones fueron reflejando cambios en los planes de estudio de las universidades acompañados por el surgimiento de grupos de investigación tales como el LTSC (del inglés, *Learning Technology Standards Committee*) que comenzó con el desarrollo de estándares técnicos en las áreas de tecnologías de aprendizaje. Por otro lado ALT (del inglés, *Association for Learning Technology*) comenzó en 1993 en Inglaterra como un conjunto de investigadores y desarrolladores en el estudio de tecnologías y definen a las Tecnologías de Aprendizaje como una amplia gama de comunicación y tecnologías de información que pueden ser usadas para apoyar el aprendizaje.

Contrariamente a lo que muchas personas consideran, la enseñanza utilizando nuevas tecnologías no pretende sustituir a los profesores, aunque sí cambia su papel principal ya que pasa a ser más un dinamizador y un supervisor que un transmisor de conocimiento [Fer05]. Los alumnos son los participantes centrales, ya que dependiendo de su rendimiento o satisfacción, se podrá evaluar el éxito de la enseñanza.

En Valdivia, el año 2004 el grupo GITA<sup>1</sup> (Grupo de Investigación en Tecnologías de Aprendizaje) nace con las mismas inquietudes relacionadas en como apoyar la educación con tecnologías. Este proyecto es parte de GITA como proyecto de titulación trabajando con una de las especificaciones de IMS, que actualmente es el principal promotor y desarrollador de especificaciones abiertas orientadas a la enseñanza electrónica [Fer05]. En conjunto con el LTG<sup>2</sup> (*Learning Technologies Research Group*) de la Universidad de Kingston en Londres, Inglaterra se lleva a cabo el proyecto que se detalla en el presente trabajo.

## 1.1 Motivación

Debido al número de instituciones y empresas dedicadas a la proliferación de estándares para Tecnologías de Aprendizaje que faciliten la interoperabilidad de recursos de aprendizaje entre distintos sistemas y plataformas, el número de iniciativas es cada día mayor. Por lo que es importante destacar que la estandarización no tiene importancia

---

<sup>1</sup> <http://www.gita.cl>

<sup>2</sup> <http://cism.kingston.ac.uk/research/groups/people/index.php?GroupID=12>

únicamente en educación que utilice la Web como único medio de distribución, ya que está influyendo en otros tipos de educación. Por ejemplo los contenidos desarrollados para clases presenciales se están empaquetando como cursos para simplificar su distribución y reutilización. En el otro sentido, contenidos desarrollados para cursos en línea se están reutilizando en clases presenciales, ya que es más sencillo su localización y uso.

El proyecto AQURATE nace como una necesidad de una herramienta que desarrollara preguntas para la versión 2.1 de la especificación IMS QTI. En el grupo GITA surge como proyecto la investigación sobre la especificación IMS QTI, cuando la investigación estaba en curso se descubre AQURATE como la única aplicación de código abierto que en ese entonces trabajaba con la especificación por lo que se comienzan a unir lazos con la Universidad de Kingston en Londres, Inglaterra, para un posible trabajo colaborativo. Con estos antecedentes se comienza a investigar las posibles funcionalidades necesarias que pudiera necesitar la herramienta.

## **1.2 Metodología**

En el mes de Abril del año 2008 se recibe una invitación para trabajar en el proyecto AQURATE en la Universidad de Kingston por lo que el alumno que escribe este trabajo de tesis concurre a Londres, Inglaterra, por ocho semanas a trabajar en las dependencias de la Universidad junto al LTG el cual desarrolló la herramienta AQURATE.

## **1.3 Objetivos**

### **1.3.1 Objetivos Generales**

Desarrollo e implementación de la funcionalidad QUIZ para la herramienta AQURATE.

### **1.3.2 Objetivos Específicos**

- Estudio de los conceptos relacionados con las Tecnologías de Aprendizaje así como las organizaciones dedicadas a desarrollar estándares y especificaciones
- Estudio de la especificación IMS QTI (*Question and Test Interoperability*).
- Estudio de la arquitectura y funcionalidades de la herramienta AQURATE y los proyectos relacionados.

- Implementación de la funcionalidad para la herramienta AQURATE.

## 1.4 Convenciones de Escritura

Las referencias son escritas de la forma [Abc01], donde las tres letras representan el apellido del autor del libro, artículo o texto usado como referencia y los números el año de publicación. En el caso de que un documento haya sido hecho por más de un autor, se considerará el apellido del primer autor, en forma alfabética, en caso de no definirse un autor principal de la obra referida. En los casos en que exista más de una publicación del mismo autor y el mismo año, se utilizará una letra minúscula, de forma correlativa, a continuación del año.

Las imágenes y diagramas serán referenciadas como Figura c,n, siendo c el número del capítulo y n un número correlativo único asociado a cada imagen o diagrama. Todas las imágenes serán listadas en el índice de figuras de la página 4.

Términos en otros idiomas serán denotados por letras cursivas. Ejemplos y descripciones de código serán mostrados en letra Courier.

## 1.5 Organización del Documento

El contenido del documento se encuentra dividido en 6 capítulos, siendo éste el primero e introductorio al proyecto.

En el capítulo 2 se realiza una descripción de las Tecnologías de Aprendizaje, estándares y especificaciones y un acercamiento a los elementos principales de este.

En el capítulo 3 se analizará la especificación IMS QTI que es la especificación base de la herramienta.

El capítulo 4 profundiza en la herramienta AQURATE, detallando su arquitectura como las funcionalidades principales, así como los proyectos con que está relacionada.

Luego el capítulo 5 contiene el desarrollo e implementación de la funcionalidad QUIZ.

Finalmente el capítulo 6 contiene las conclusiones obtenidas, además de delinear posibles áreas de trabajo futuro.

## **2. TECNOLOGÍAS DE APRENDIZAJE, ESTÁNDARES Y ESPECIFICACIONES**

El término Tecnologías de Aprendizaje abarca información y tecnología instructiva, así como también herramientas de telecomunicaciones, aplicaciones, y sistemas que le dan apoyo al aprendizaje, enseñanza y evaluación. Para maximizar el efecto transformativo de las tecnologías de aprendizaje, es necesario que estas herramientas sean replicables a través de organizaciones para así, proveer liderazgo y aumentar la capacidad, rapidez y desempeño de desarrollo sin importar el lugar donde operen las organizaciones. Un amplio rango de personas en el sector público y privado de la educación tiene a las tecnologías de aprendizaje como núcleo de su trabajo.[ALT][EDU]

Así como las Tecnologías de Información cambiaron fundamentalmente la forma de trabajar en las organizaciones, las tecnologías de aprendizaje están cambiando la naturaleza de cómo deben los usuarios aprender a hacer ese trabajo. [Fer05]

El modelo fundamental de aprendizaje no ha cambiado: Los Profesionales de Aprendizaje aún enseñan a otros a hacer cosas que antes no podían hacer. Las tecnologías de Aprendizaje son sofisticadas herramientas que permiten que cada profesor sea más productivo al enseñar.

### **2.1 ¿Qué es un Estándar?**

Según el diccionario de la Real Academia Española<sup>3</sup>, un estándar es lo “que sirve como tipo, modelo, norma, patrón o referencia”. En el campo técnico, la estandarización es el proceso por el cual se establecen normas comúnmente aceptadas que permiten la cooperación de diferentes empresas o instituciones sin menoscabar su posibilidad de competir. Un estándar proporciona ventajas a las empresas y a los usuarios, particularmente a los usuarios, ya que les posibilita un amplio rango de elección entre distintos proveedores que cumplen con un estándar determinado y no limitarse con un proveedor en particular. Es por esto que las empresas crean productos que son compatibles.

Existen dos tipos de estándares los oficiales o “de jure” y los “de facto” [Fer05]. Los estándares oficiales son aquellos que han sido aprobados y sancionados por un organismo oficial de estandarización, ya sea nacional o internacional. Estos estándares

---

<sup>3</sup> Diccionario online <http://www.rae.es>

en algunos casos son de obligado cumplimiento como, por ejemplo, que todas las páginas Web oficiales deben cumplir un determinado nivel de accesibilidad para discapacitados. Los estándares de facto son aquellos que se usan por voluntad propia o conveniencia y tienen una amplia aceptación, aunque no hayan sido sancionados por un organismo de estandarización. El caso más conocido en Internet son las recomendaciones realizadas por el W3C<sup>4</sup>, que crea las normas probablemente más utilizadas en Internet como, por ejemplo, el lenguaje HTML<sup>5</sup> (y que en muchos casos después de publicadas pasan a ser reconocidas como estándares formales).

¿Porque deberían preocuparse las organizaciones por la aparición y convergencia de estándares de aprendizaje? La respuesta se reduce a la protección de las organizaciones en cuanto al retorno de las inversiones en tecnologías de aprendizaje así como los contenidos de aprendizaje y servicios que desarrollan. Millones de dólares serán invertidos en tecnologías, contenidos y servicios para mejorar el conocimiento y capacidades de las personas. Si los sistemas no pueden crecer, ser mantenidos, actualizados y no ser útiles a los profesores, la inversión será un fracaso o seriamente menos efectiva en retornar resultados.

## **2.2 Ventajas Aportadas por los Estándares a las Tecnologías de Aprendizaje**

Para las tecnologías de aprendizaje, una de las principales funciones de los estándares es servir como facilitadores de la durabilidad y de la reutilización en el tiempo de los contenidos y de la interoperabilidad, es decir, facilitar el intercambio de los contenidos entre diversas plataformas y sistemas. Hay que evitar caer en el error de ver el estándar como un limitador de la iniciativa o creatividad personal. En muchos casos, cuando los educadores oyen la palabra estándar suelen tener una reacción adversa, ya que tienden a considerar que es una norma de obligado cumplimiento que coartará su creatividad respecto a la creación de nuevos cursos, o su forma habitual de planificar una acción formativa o una clase. Otra circunstancia es considerar que su uso es sólo en educación a distancia y que no son útiles para otros planteamientos educativos. Esto no es cierto, ya que la existencia de contenidos educativos reutilizables puede ser de gran ayuda para simplificar el trabajo de los docentes, aunque lo utilicen en educación

---

<sup>4</sup> World Wide Web Consortium. <http://www.w3c.org>

<sup>5</sup> Hyper Text Markup Lenguaje <http://www.w3c.org> (Lenguaje de marcado de hipertexto)

presencial o en un formato mixto presencial-Web (llamado *blended learning* o *b-learning*). [Fer05]

Existen varias ventajas asociadas a la utilización generalizada de estándares para todas las partes implicadas en el proceso de aprendizaje. Entre ellas cabe mencionar las siguientes [Sun02]:

- Desde la perspectiva del consumidor, los estándares evitan que éste quede atrapado en ciertos vendedores o productos en particular. Los costos se ven rebajados al evitar instalaciones de software específicas y a su vez poder integrarlas el software como un módulo.
- Desde la perspectiva del vendedor, los métodos estandarizados eliminan la necesidad de desarrollar interfaces específicas para productos diferentes. Esto implica un menor costo de desarrollo y aumenta el tamaño del potencial mercado. Los productores pueden competir en calidad y valor.
- Desde la perspectiva del creador de contenidos de aprendizaje, los estándares permiten producir contenido en un único formato para ser usado en cualquier sistema.
- Desde la perspectiva del estudiante, los estándares posibilitan que el resultado del aprendizaje (Certificados, créditos, etc.) sean más portables entre los LMS (Sistemas de Gestión de Aprendizaje del inglés *Learning Management System*) de las Universidades.
- Desde la perspectiva del diseñador, los estándares en e-learning harán sus trabajos más fáciles al posibilitar el acceso a grandes repositorios de contenido reusable, reduciendo la necesidad de desarrollar múltiples sistemas y permitiendo crear contenidos modularizados que pueden ser fácilmente actualizados y mantenidos.

Los estándares protegen los desarrollos en e-learning con 5 “habilidades” que se mencionan a continuación [Mas03]:

- Interoperabilidad. La interoperabilidad es la habilidad para intercambiar y usar de una forma transparente contenidos desarrollados en distintos lugares, con diferentes herramientas o plataformas.

- Re-Usabilidad. La flexibilidad de incorporar contenidos en múltiples aplicaciones y contextos distintos a aquél para el que fueron inicialmente diseñados.
- Gestionabilidad. Es la habilidad del Sistema como un LMS, de entregar información apropiada sobre el estudiante y los contenidos que tiene asociados.
- Accesibilidad. Acceder el contenido apropiado en el momento justo y en el dispositivo correcto sin importar las barreras geográficas o discapacidades físicas.
- Durabilidad. La habilidad resistir a los cambios de tecnologías sin tener que rediseñar, reconfigurar o re-codificar el Sistema.

## **2.3 Organismos e Instituciones que Participan en los Procesos de Estandarización**

Tener una idea clara del proceso de estandarización para las tecnologías de aprendizaje es una tarea compleja, debido principalmente al poco tiempo que se lleva realizando el proceso y a la profusión de grupos, instituciones y consorcios que trabajan en el tema.

No obstante el escenario está mejorando, ya que cada vez más se llegan a acuerdos de colaboración entre distintas iniciativas. Por tanto se está cada vez más cerca de una estandarización real y que tenga un impacto efectivo en la industria.

A continuación se presentarán brevemente algunas de las iniciativas más importantes o que están teniendo una mayor repercusión. [Fer05]

### **2.3.1 Comité de Capacitación Basado en Computador de la Industria de Aviación (AICC)**

El Comité de la Industria de Aviación con Capacitación Basado en Computador (*AICC*<sup>6</sup>, del inglés *Aviation Industry Computer Based Training Committee*) es una asociación internacional de capacitación de profesionales basado en tecnologías. La *AICC* fue creada en 1988 por la necesidad de estandarizar el hardware de las plataformas de entrenamiento que se usan en aviación.

---

<sup>6</sup> <http://www.aicc.org>

Su objetivo es crear aplicaciones educativas que sean eficientes, que tengan un costo razonable y que sean mantenibles ó actualizables a lo largo del tiempo. AICC publica recomendaciones en muchos aspectos del e-learning (incluido el hardware), pero quizás la que ha tenido mayor impacto ha sido la recomendación para interoperabilidad *CMI (Computer-Managed Instruction, 1993)*. Es una especificación sobre cómo crear contenido que se pueda comunicar con el mayor número de sistemas LMS.

### **2.3.2 Aprendizaje Distribuido Avanzado (ADL)**

En Noviembre de 1997 el Departamento de Defensa de EE.UU. y la oficina de Ciencia y Tecnología de la Casa Blanca lanzaron la iniciativa de Aprendizaje Distribuido Avanzado (del inglés *Advanced Distributed Learning*). El propósito de ADL<sup>7</sup> es desarrollar el e-learning para asegurar el acceso a materiales educativos y de alta calidad que puedan ser adaptados a las necesidades individuales y que sean de distribución sencilla. ADL surge como respuesta a las necesidades de uno de los mayores consumidores de software del mundo, y forma parte del esfuerzo que el gobierno de Estados Unidos viene realizando con el objetivo de conseguir una enseñanza de calidad, en el que también están implicados los departamentos de Educación y Trabajo.

Su principal resultado es un conjunto de especificaciones que, bajo la denominación *Shareable Content Object Reference Model (SCORM)* propone un modelo de agregación de contenidos (*Content Aggregation Model, CAM*), un entorno de tiempo de ejecución (*Run-Time Environment, RTE*) y la secuenciación y navegación (*Sequencing and Navigation, SN*) de los contenidos.

### **2.3.3 Estructura de las Especificaciones de IMS**

El Consorcio Global IMS<sup>8</sup> (del inglés *Global Learning Consortium*) es un grupo independiente, sin ánimo de lucro en la cual participan instituciones educativas de todo el mundo (desde universidades a pequeñas empresas de formación), fabricantes, y vendedores de aplicaciones software para la educación.

---

<sup>7</sup> <http://www.adlnet.org>.

<sup>8</sup> <http://www.imsglobal.org>

IMS tiene muchas especificaciones ya que cada una de ellas está enfocada en una necesidad distinta del proceso de enseñanza. Hay especificaciones que se refieren a meta datos de los objetos educativos, al formato de empaquetamiento y distribución de los cursos, a la información del usuario, a la secuenciación de contenidos educativos, o incluso al diseño de la actividad educativa en su conjunto.

El objetivo de IMS de definir especificaciones que hagan posible la interoperabilidad de aplicaciones y servicios de enseñanza distribuida, se ha concretado, al día de hoy, en más de 16 especificaciones principales. Normalmente cada una de ellas se encuentra detallada al menos en tres documentos: [Fer05]

- Guía de Implementación y consejos. En ésta se incluyen: la forma de uso de la especificación, ejemplos, la relación con otras especificaciones, y cualquier tipo de información complementaria que pueda servir de ayuda. Normalmente es el documento que se recomienda leer primero para entender los conceptos generales con los que se trata.
- Modelo de Información. Documento que describe de manera formal, los datos así como su estructuración, detallando cada uno de los elementos considerados en la especificación. El modelo que se propone en este documento es independiente del formato físico en el que finalmente se representa la información.
- Documento de Enlace. Documento que ofrece la forma de representar la estructura de datos de la especificación, generalmente, en XML. Adicionalmente se proporciona el esquema documental XML que permite comprobar la validez de la estructura de un documento que haya sido creado, respecto a la especificación a la que está asociado.

### **2.3.4 Comité de Estándares para Tecnologías de Aprendizaje del IEEE**

El comité de estandarización de las tecnologías de aprendizaje, (LTSC<sup>9</sup>, del inglés *Learning Technologies Standardization Committee*), perteneciente al Instituto de Ingenieros Eléctricos y Electrónicos (IEEE<sup>10</sup>, del inglés *Institute of Electrical and Electronic Engineers*), cubre prácticamente todos los aspectos del aprendizaje basado en computador. Su misión principal es desarrollar estándares técnicos, prácticas

---

<sup>9</sup> <http://ieeeltsc.org/>.

<sup>10</sup> <http://www.ieee.org/>.

recomendadas y guías para componentes software, herramientas, tecnologías y métodos de diseño que faciliten el desarrollo, implantación, mantenimiento e interoperabilidad de implementación de sistemas educativos.

LTSC está organizado en subcomités que se encargan de áreas de trabajo determinadas como la definición de la arquitectura de sistemas de aprendizaje o la definición de meta datos para objetos educativos. En estos momentos el área de mayor impacto es la relacionada con los meta datos de los recursos educativos, en donde se especifica un esquema de datos conceptual que define la estructura de los meta datos de un objeto de aprendizaje. [LTSC][OJE08]

### **2.3.5 La Alianza ARIADNE**

La alianza ARIADNE<sup>11</sup> (del inglés *Alliance of Remote Instructional Distribution Networks for Europe*) es una fundación que surge a raíz de dos proyectos con financiación de la Unión Europea y que está compuesta por miembros de la industria y de las instituciones educativas. La misión básica de ARIADNE es permitir la mejora de la calidad de las tecnologías de aprendizaje mediante el desarrollo de herramientas y metodologías que permitan la interoperabilidad y reutilización de objetos de aprendizaje. Están desarrollando guías y recomendaciones para la aplicación de estándares, siendo muy activos en aspectos como la indexación multilingüe y los almacenes o repositorios de objetos de aprendizaje. Además han colaborado activamente en la elaboración del estándar LOM (del inglés *Learning Object Metadata*).

## **2.4 Estándares y Especificaciones más Utilizados**

Como se ha mostrado anteriormente hay un gran número de iniciativas de estandarización de modo que no es muy realista tratar de hacer una descripción completa de todas ellas en este trabajo de tesis. Además existen relaciones entre las especificaciones realizadas por diferentes grupos que a veces se solapan, y otras son simplemente adaptaciones o perfiles de aplicación para adaptarse a un campo o uso específico.

Fernández (2005) considera en [Fer05] que actualmente IMS es el principal promotor y desarrollador de especificaciones abiertas, y que cubren más aspectos de la enseñanza electrónica. Este trabajo, conjuntamente con el desarrollado por ADL en su

---

<sup>11</sup> <http://www.ariadne-eu.org/>

modelo de referencia SCORM, y por IEEE LTSC con su propuesta de metadatos para objetos de aprendizaje, son los que están teniendo una mayor repercusión. De hecho es especialmente interesante analizar las propuestas de IMS, ya que su amplio número de colaboraciones con otras entidades, y especialmente con IEEE LTSC, hace muy previsible que sus especificaciones sean la base para nuevos estándares. Por este motivo, a continuación sólo se analizarán brevemente los principales estándares en el ámbito de las tecnologías de aprendizaje: [Fer05] [OJE08]

### **2.4.1 Especificación IMS *Content Packaging***

La recolección y el empaquetado de los contenidos educativos en formato digital es un requisito básico para muchos de los procesos involucrados en el despliegue, gestión, distribución y agregación de dichos contenidos. La especificación *Content Packaging* de IMS define un formato digital estándar para representar dichos paquetes de contenidos educativos. [IMS04]

El objetivo de esta especificación es permitir la distribución de contenidos reutilizables e intercambiables, es decir, describe el modo en el que se debe empaquetar el contenido educativo para que pueda ser procesado por otro sistema diferente. Ofrece una forma de empaquetar (en un archivo comprimido tipo .ZIP<sup>12</sup>) los contenidos educativos tales como cursos individuales, conjuntos de cursos, o cualquier tipo de recurso necesario en el proceso educativo (por ejemplo, evaluaciones o exámenes). Al distribuir una serie de contenidos empaquetados, existe un documento fundamental que es el Manifiesto. Dicho documento es un archivo XML<sup>13</sup> en el que se describe la estructura de los contenidos incluidos en el paquete (figura 2.1). Dicha descripción se realiza a dos niveles diferentes: [Fer05]

- Por un lado, se describe cada uno de los Recursos del paquete. En una primera aproximación se puede hacer una relación casi directa entre un Recurso y un archivo con contenidos visualizables como pueden ser archivos HTML, animaciones en Flash, etc. En realidad, en cada Recurso se puede incluir información sobre los archivos que componen dicho Recurso, el tipo de los mismos y, opcionalmente, metadatos con información adicional sobre dicho Recurso.

---

<sup>12</sup> <http://www.pkware.com/support/zip-application-note>

<sup>13</sup> <http://www.w3.org/XML/>

- Por otro lado, en el Manifiesto se describe como están organizados dichos Recursos, es decir, como se estructura el contenido del paquete. Esto se implementa mediante las Organizaciones. Una organización de una posible ordenación jerárquica (actualmente en forma de árbol) de los Recursos de un paquete. El estándar permite que un Manifiesto contenga distintas organizaciones sobre los Recursos del paquete, dando así lugar a distintas vistas o “cursos” a partir de los mismos contenidos. El elemento básico de estructuración que se usa al definir las organizaciones son los Ítems. A cada Ítem se le puede asociar un Recurso, de modo que el árbol de Ítems es, efectivamente, una estructuración de los Recursos del paquete.

En resumen, el Manifiesto es un fichero XML que describe y organiza los contenidos de un paquete, añadiendo información adicional en forma de metadatos que pueden ser procesados y aprovechados en tareas de catalogación de contenidos. Luego para la distribución e intercambio efectivo de los cursos, se crea un Archivo de Intercambio de Paquetes (PIF, *Package Interchange File*). Dicho archivo PIF, posee en su interior el archivo manifiesto y todos los recursos que el archivo manifiesto referencia. Este archivo PIF es un archivo de intercambio con formato ZIP. La figura 2.1 muestra la estructura de un paquete PIF.

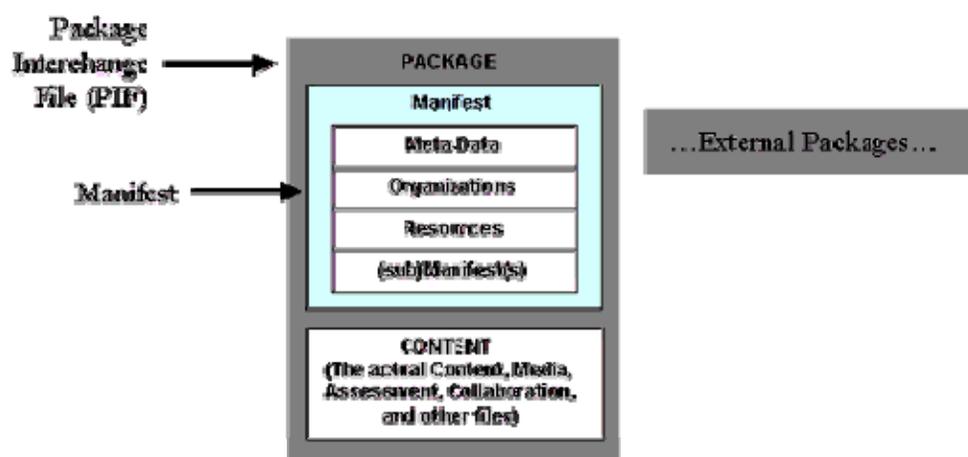


Figura 2.1: Estructura archivo PIF. IMS CP. Obtenida de [IMS04]

IMS CP es una especificación básica para facilitar la *interoperabilidad* entre los sistemas de que utilizan tecnologías de aprendizaje, ya que dichos sistemas pueden intercambiar materiales empaquetados de acuerdo a IMS CP: un sistema que soporta IMS CP (por ejemplo, una herramienta de autor, un sistema de gestión del aprendizaje, una biblioteca digital de recursos educativos, etc.) será capaz de abrir los paquetes IMS,

independientemente de la forma y el lugar en los que dichos paquetes hayan sido producidos.

### **2.4.2 Especificación IMS *Learning Design***

La especificación IMS LD [IMS03a] (del inglés *Learning Design*) se ocupa de describir y codificar el diseño pedagógico, es decir las metodologías educativas implícitas en un proceso de enseñanza, de forma que sean procesables por un LMS. En este caso se utiliza un nuevo concepto, la *unidad de aprendizaje* (UoL, del inglés *Unit of Learning*), ya que se considera que lo importante no son tanto los objetos de aprendizaje por sí mismos, si no las actividades en las que se encuentran implicados. El elemento clave de una Unidad de Aprendizaje es la *actividad o tarea*, que se concibe como uno o más actores (por ejemplo: alumnos, profesores) que trabajan para lograr un cierto objetivo educativo en un determinado entorno. El entorno contiene los recursos y los servicios necesarios para realizar la actividad propuesta. El principio subyacente es que los alumnos aprenden realizando actividades en un entorno, en el cual los objetos de aprendizaje son recursos que permiten o facilitan la tarea. La visión es más amplia que la de los objetos de aprendizaje básicos, ya que se contempla el uso de herramientas o de procesos, como la comunicación entre alumnos o entre alumnos y profesores. De hecho el rol o papel de un alumno podría cambiar en un determinado momento, por ejemplo, para supervisar el trabajo realizado por otros alumnos. La unidad de aprendizaje es la nueva unidad mínima de intercambio entre sistemas, ya que se considera que si se descompone en sus elementos básicos se pierde el diseño pedagógico que permite alcanzar el resultado deseado.

### **2.4.3 Especificación IMS *Learner Information Packaging***

Especificación IMS LIP [IMS02] que nos indica qué información se almacena referente a un alumno (o grupo de alumnos) o incluso a un autor de contenido educativo, y cómo debe almacenarse. El objetivo de esta especificación es definir una estructura que permita el intercambio de paquetes con información entre distintos LMS, sistemas de recursos humanos, sistemas de información de estudiantes y otros sistemas relativos a sistemas de enseñanza. La existencia de formatos consensuados para la definición de expedientes de alumnos permite su exportación entre sistemas educativos heterogéneos.

Es necesario decidir qué información debe incluirse en el expediente y el formato para representarla.

#### **2.4.4 Especificación IMS *Simple Sequencing***

La especificación IMS *Simple Sequencing* [IMS03b] se ocupa de la definición de mecanismos que permitan la secuenciación de los recursos educativos dentro de cualquier sistema de aprendizaje que lo implemente. El objetivo es poder definir, por ejemplo, el orden en el que se presentan los objetos de aprendizaje o las reglas para seleccionar un objeto de aprendizaje entre varios posibles en función del comportamiento del alumno con el contenido.

#### **2.4.5 Estándar IEEE LOM**

LOM [IEE02] es un modelo de datos, codificado en XML, usado para describir un objeto de aprendizaje y otros recursos digitales similares usados para el apoyo al aprendizaje. Estos metadatos proporcionan descripciones, propiedades e información sobre los objetos de aprendizaje que permiten caracterizarlos de forma que se simplifique su uso y gestión [IMS06a]. Lo que se busca, es poder saber el contenido y el propósito de un determinado objeto de aprendizaje sin la necesidad de acceder a dicho contenido. Por lo tanto, los metadatos proporcionan información que permite realizar búsquedas más eficientes de objetos de aprendizaje o diseños de aprendizaje en repositorios especializados. Los metadatos pueden aplicarse tanto a objetos de aprendizaje como a cursos completos o recursos aislados. [OJE08][Fer05]

El objetivo de LOM es la creación de descripciones estructuradas de recursos educativos. Su modelo de datos especifica qué aspectos de un objeto de aprendizaje deberían ser descritos y qué vocabularios se pueden utilizar en dicha descripción.

Esta es una descripción jerárquica con nueve apartados principales que agrupan el resto de campos. A continuación describimos cada una de estas categorías: [Fer05]

- *General*: Aquí se describe el objeto educativo. Incluye campos como identificadores, título, descripción, etc.

- *Lifecycle*: Almacena un histórico del objeto y su estado actual. Detalla quiénes han interactuado con este objeto desde que fue creado, y el tipo de interacción que han realizado.
- *Meta-Metadata*: Agrupa información sobre los metadatos. Esto puede parecer redundante a primera vista pero resulta muy interesante tener información como quién ha contribuido a la creación de los metadatos y el tipo de contribución que ha realizado.
- *Technical*. Incluye la información técnica del recurso de aprendizaje, tal como tamaño, ubicación, o formato en el que se encuentra. Además, en este elemento se almacenan los posibles requisitos técnicos necesarios para poder usar el objeto al que se refieren los metadatos.
- *Educational*. En este elemento se encuentran las diferentes características pedagógicas del objeto. Típicamente se incluyen campos como tipo de recurso, nivel de interactividad entre el usuario y el objeto, o el contexto de uso del recurso, entre otros.
- *Rights*. Se incluyen los detalles sobre la propiedad intelectual del recurso. También se detallan las condiciones de utilización y el precio en caso de tenerlo.
- *Relation*. Explica el tipo de relación que tiene el recurso de aprendizaje con otros. Posee un par nombre-valor en el que detalla el nombre del objeto de aprendizaje relacionado y el tipo de relación.
- *Annotation*. Incluye comentarios sobre la utilización del objeto de aprendizaje, además de su autor y la fecha de creación.
- *Cassification*. Nos informa si el objeto de aprendizaje pertenece a algún tema en concreto. Permite tanto detalle cómo se quiera mediante anidamiento de temas.

#### **2.4.6 Modelo de Referencia SCORM**

El modelo SCORM (del inglés *Shareable Content Object referente Model*) [ADL04a] es un conjunto de estándares y especificaciones para compartir, reutilizar, importar y exportar Objetos de Aprendizaje. Este modelo describe cómo las unidades de contenidos se relacionan unas con otras a diferentes niveles de granularidad, cómo se comunican los contenidos con el LMS, define también cómo empaquetar los contenidos para importarse y exportarse entre plataformas, y describe las reglas que un LMS debe seguir a fin de presentar un aprendizaje específico. SCORM es expandible e incluye a trabajos de IEEE, AICC y de IMS para algunas de sus funciones. Maneja las unidades de

contenido con el nombre de SCO (*Sharable Content Object*) que son simplemente objetos de aprendizaje que cumplen con la especificación SCORM. [OJE08]

Los detalles de la especificación se encuentran en cuatro documentos a los que se les da mantenimiento de manera independiente:

- *SCORM Overview* [ADL04a]. Este documento describe la historia y los objetivos de la Iniciativa ADL y de SCORM, incluye las especificaciones y los estándares que SCORM ha adoptado para su definición. También describe cómo se relacionan los otros libros o documentos de la especificación SCORM
- *SCORM Content Aggregation Model, CAM* [ADL04c]. Describe los componentes utilizados en el aprendizaje, cómo empaquetar esos componentes para el intercambio entre sistemas, cómo describir esos componentes para permitir la búsqueda y la recuperación, y cómo definir las reglas de secuencia de los componentes. El CAM promueve consistencia en el almacenamiento, etiquetado, empaquetado, intercambio y recuperación de contenidos. Este documento también define las responsabilidades y requisitos para construir contenidos agregados como cursos, lecciones o módulos. Asimismo, contiene información para crear paquetes de contenido, aplicando metadatos y una secuenciación y detalles de navegación. Un paquete de contenido (content packaging) está formado por un archivo XML con descriptores del objeto y el archivo del objeto. Entre los descriptores se encuentra información para identificar, organizar y procesar el objeto en un LMS.
- *SCORM Run-Time Environment, RTE* [ADL04d]. Este libro describe el medio para interoperar contenidos de aprendizaje basados en SCO y los LMS. Define los requerimientos de un LMS para administrar actividades de tiempo de ejecución (run-time) en el entorno, como arranque de procesos de contenidos y comunicación entre contenidos, así como los elementos del modelo de datos utilizados para transmitir los contenidos al estudiante. RTE provee el medio para que los contenidos puedan ser interoperables entre diversas plataformas LMS, sin importar la herramienta con la que fueron creados.
- *SCORM Sequencing and Navigation, SN* [ADL04b]. El documento SN describe las reglas que un LMS debe seguir a fin de presentar un aprendizaje específico. El desarrollador del contenido es responsable de definir las reglas a las que el LMS debe adherirse. Las reglas se expresan en la estructura del contenido y se codifican en una sección del paquete del contenido. Con este mecanismo, el

comportamiento esperado de una colección de recursos de aprendizaje puede ser transferido con un paquete del entorno de un LMS a otro.

ADL se ha centrado desde un principio en el aprendizaje sobre la Web. Actualmente es el modelo más utilizado en la industria y que cuenta con mayor cantidad de herramientas que lo soportan. Es un perfil de aplicación, ya que combina muchas especificaciones (IMS, AICC, IEEE) y las particulariza para un caso concreto. Las especificaciones, por su generalidad, dejan sin fijar aspectos que son necesarios para facilitar la implementación final, y SCORM trata de ser más preciso para lograr una mayor compatibilidad. [Fer05]

#### **2.4.7 Especificación IMS *Vocabulary Definition and Exchange***

IMS VDEX (del inglés Vocabulary Definition and Exchange) define una gramática para el intercambio de listas de valores o vocabularios, que puedan ser procesables automáticamente y entendibles por las personas. Permite por ejemplo definir valores para ser utilizados en IEEE LOM, IMS LIP o en ADL/SCORM.

En la presente tesis, la especificación con mayor relevancia es IMS *Question And Test Interoperability* que se detallará en el próximo capítulo

### 3. ESPECIFICACIÓN IMS QUESTION AND TEST INTEROPERABILITY

#### 3.1 Introducción

Las Tecnologías de Aprendizaje han contribuido a que los exámenes sean utilizados ampliamente como herramienta para evaluar si un alumno ha asimilado los conceptos que le han sido presentados y como herramienta de auto-evaluación para el alumno, de manera que éste pueda reforzar aquella parte de la materia en la que no tenga un dominio suficiente. Un gran número de tipos de preguntas pueden ser corregidas de manera automática mediante el uso de este tipo de herramientas, de manera que un profesor puede crear una serie de preguntas que el sistema informático puede utilizar para preparar exámenes y corregirlos automáticamente.

De hecho, en la actualidad, gran parte de las plataformas de aprendizaje incluyen con menor o mayor funcionalidad una herramienta para la creación, gestión y realización de exámenes en línea. Sin embargo, el esfuerzo realizado por el profesor en la elaboración de preguntas y exámenes puede llegar a perderse si es necesario cambiar de plataforma de aprendizaje. [Fer05]

Asimismo, ya que la creación de dichos exámenes requiere invertir cierto esfuerzo, sería deseable tener la posibilidad de compartir este esfuerzo permitiendo el intercambio de exámenes completos, o poder crear repositorios de preguntas. Así mismo, estas preguntas deberían estar clasificadas por materias y por dificultades para simplificar su localización y reutilización en la formulación de nuevos exámenes.

La especificación de IMS *Question and Test Interoperability* (IMS QTI) describe una estructura para representar preguntas individuales o ítems (*assessment Item*), gestionar evaluaciones o exámenes completos (*assessment Test*) y reportes con los resultados correspondientes. [IMS06c] El objetivo principal de esta especificación es permitir el intercambio de preguntas, evaluaciones y resultados entre distintas herramientas. Con este propósito IMS QTI plantea un modelo en el que se definen los componentes principales que intervienen en el proceso de evaluación y, adicionalmente a este modelo, se proporciona un formato de contenido para almacenar las preguntas de manera independientemente del sistema o herramienta de autoría utilizada para crearlas. [Fer05] [IMS06b]

El modelo de datos está descrito abstractamente, usando UML<sup>14</sup> (del inglés *Unified Modelling Language*) para facilitar la compatibilidad con una amplia gama de herramientas de modelado de datos y lenguajes de programación, sin embargo, para la conexión entre sistemas se provee y recomienda el uso de XML. La especificación IMS QTI ha sido diseñada para mantener la interoperabilidad e innovación a través de puntos de extensión y escalabilidad bien definidos. Estos puntos de extensión pueden ser usados para representar ítems directamente sin importar la procedencia de los datos. Esto permite, por ejemplo, el uso de las mismas preguntas en diversos LMS o en sistemas de evaluación electrónica, o la integración en un único LMS de preguntas o exámenes desarrollados con distintas herramientas. Por otro lado se propone un sistema coherente para que los sistemas puedan informar de cuál es el resultado de una evaluación.

### **3.2 Historia de IMS QTI.**

IMS QTI es una de las especificaciones en las que el consorcio IMS trabajó más tempranamente una inicial versión 0.5 de la especificación fue puesta a disposición para discusión en Marzo de 1999 que fue aceptada en Noviembre del mismo año para desarrollar la versión 1.0 de la especificación, que fue lanzada como un borrador público en Febrero del 2000 y como una especificación final en Mayo del 2000. La especificación fue extendida y actualizada dos veces, en Marzo del 2001 y Enero del 2002. Desde entonces, surgieron varias modificaciones por parte del equipo del proyecto QTI. Muchas de ellas fueron puestas en un suplemento, lo que definió la versión 1.2.1 de la especificación que fue lanzada en Marzo del 2003. Quedaron fuera varios cambios debido a que estos mismos requerirían cambios a la especificación, los cuales no serían compatibles por temas fundamentales que no habían sido cubiertos y se necesitaba una extensiva clarificación.

Desde que la especificación QTI fue publicada, las especificaciones de IMS han ido creciendo y se ha ido trabajando en *IMS Content Packaging*, *IMS Simple Sequencing* y últimamente *IMS Learning Design*, por lo que se creó la necesidad de una revisión cruzada de las especificaciones. Esta revisión tomo lugar en el 2003 y se identificaron una gran cantidad de situaciones que afectaban a QTI. En septiembre ese año se definió proyecto para dirigir las mejoras de la versión 1.x y la armonización con el resto de las especificaciones para bosquejar QTI V2.0.

---

<sup>14</sup> <http://www.uml.org/>

Para hacer el trabajo manejable y asegurar que los resultados fueran devueltos a la comunidad cuanto antes, algunas restricciones fueron puestas sobre el alcance del trabajo recomendado. Por lo tanto, el lanzamiento de la versión 2.0 en Enero del 2005 de la especificación se concentró sólo en las preguntas individuales (*assessmentItem*) y no se actualizaron las partes de la especificación que tenían relación con la composición de dichas preguntas, es decir, la creación de exámenes completos. La versión 2.1 completa la actualización desde 1.x y 2.x reemplazando las partes faltantes de la especificación QTI en Julio del 2006.

IMS QTI versión 2.1 está actualmente en proceso de evolución en modo borrador sobre el que la comunidad, tanto educativa como técnica, puede opinar. El objetivo de esta nueva versión es seguir con el proceso de simplificación y evolución de la especificación, esta vez dando soporte a los exámenes completos y al intercambio de los resultados de los mismos. Además también se incluye información para clarificar la compatibilidad y el uso de IMS QTI con algunas otras de las especificaciones ya existentes.

Sin embargo, el potencial de la especificación IMS QTI no se ha explotado completamente debido en parte a su complejidad y a la escasa existencia de herramientas informáticas que pusieran en práctica (habitualmente de manera parcial) la especificación.

En resumen, se puede destacar que en versiones anteriores se centraba principalmente en cómo se presentaba finalmente la pregunta, ahora se definen los posibles tipos de interacciones por parte del usuario (ejemplo: seleccionar uno o más elementos de una lista, crear asociaciones entre elementos de dos listas, introducir texto, seleccionar un trozo de texto de un texto más largo, etc.) Además de todas las interacciones contempladas, introduce un tipo de interacción propio para poder extender el modelo y crear nuevas formas de interacción para poder introducir nuevos tipo de preguntas. También tiene plantillas de preguntas para crear preguntas similares, pero en las que hay partes variables que se seleccionan aleatoriamente entre un conjunto de valores predefinidos. Otra de las novedades que introduce son los ítems adaptativos, que permite su corrección adaptativa en función de una secuencia de intentos. Esto permite, por ejemplo, que el alumno pueda alterar su respuesta debido a la realimentación, o que se le planteen preguntas adicionales en función de su respuesta actual.

### 3.3 Conceptos Básicos de IMS QTI V 2.X

La versión 2 de la especificación intenta simplificar su uso tanto desde el punto de vista técnico como desde el punto de vista del usuario de dicha especificación. Para ello, se han definido de manera completamente independiente tres conceptos:

- Las preguntas. Las preguntas individuales podrán ser utilizadas como un recurso educativo independiente, por ejemplo, como un recurso más dentro de un paquete IMS.
- Los exámenes. Los exámenes son agrupaciones de preguntas que permitirán resumir las evaluaciones conseguidas en las preguntas individuales en una única evaluación del examen.
- Resultados de los exámenes. La interacción de los alumnos con las preguntas individuales y con los exámenes generará diferentes registros de información que puede ser recolectada para su posterior estudio.

#### 3.3.1 Las Preguntas

Las preguntas individuales (*assessmentItem*) en QTI son auto-contenidas, es decir, incluyen toda la información necesaria para su presentación al alumno y su corrección automática. La información relativa a la ó las preguntas correctas es agrupada al principio de la pregunta bajo la etiqueta *ResponseDeclaration*. Toda la información relativa a la presentación ha sido agrupada en el cuerpo (*itemBody*) de las preguntas. En la presentación de la pregunta están involucrados dos aspectos:

- El enunciado de la pregunta. Obviamente, la pregunta debe contener el enunciado de la misma y, de manera adicional, puede contener material explicativo complementario que permita al docente indicar el contexto en el que se realiza la pregunta. En la especificación IMS QTI v 2.X, los contenidos que podemos utilizar dentro del cuerpo de la pregunta siguen el estándar XHTML<sup>15</sup>, es decir, contenido Web y además es posible utilizar el estándar MathML<sup>16</sup> para la representación de ecuaciones matemáticas.

---

<sup>15</sup> <http://www.w3.org/TR/xhtml11/>

<sup>16</sup> <http://www.w3.org/Math/>

- La construcción de la respuesta. De manera adicional al enunciado de la pregunta, debemos dotar al alumno del equivalente del lápiz y papel para poder construir la respuesta. En el caso de IMS QTI v 2 se ha introducido el concepto de interacción (*interaction*). Dependiendo del tipo la herramienta informática generará una presentación acorde.

Adicionalmente a la definición de la presentación de las preguntas y junto a la definición de la interacción se llevará a cabo para recolectar las respuestas, es necesario especificar cómo corregir la pregunta. El funcionamiento de la corrección es simple y directo (figura 3.1.). Cuando un alumno tiene que responder a una pregunta, la herramienta presentará la pregunta y la interacción al alumno. Como resultado de esta interacción obtendremos una representación de la respuesta del alumno (figura 3.1, 1). Estas respuestas servirán como información de entrada para el proceso de corrección (figura 3.1, 2), generando finalmente una evaluación (figura 3.1, 3).

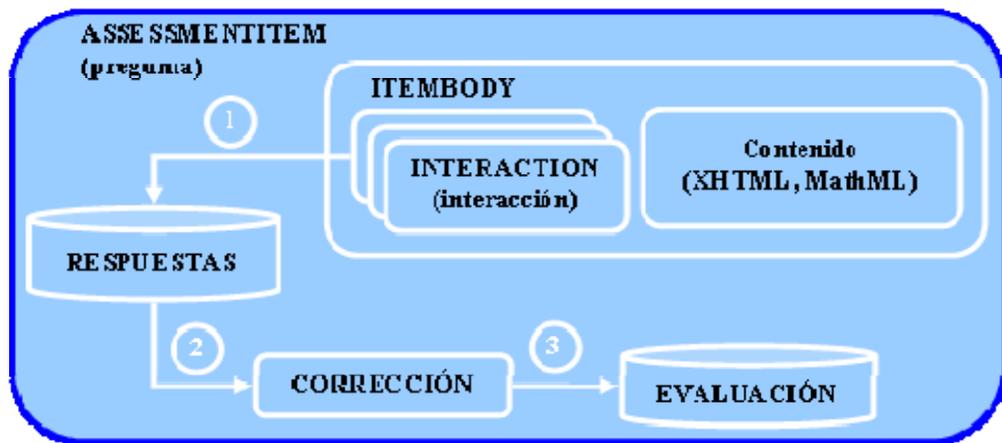


Figura 3.1: Estructura de un *assessmentItem* y su proceso de evaluación. Obtenida de [Fer05]

IMS QTI proporciona un arsenal de herramientas que permiten crear métodos de evaluación con alto nivel de personalización, incluyendo la posibilidad de forzar una determinada presentación. Sin embargo, esto lleva a que el creador de la pregunta necesite conocer en detalle la especificación IMS QTI e incluso tener conocimientos de programación. La figura 3.2 muestra una pregunta codificada en XML. Para simplificar tanto la tarea de autoría como la creación de herramientas que pongan en práctica IMS QTI se han definido un conjunto de plantillas de corrección en la que se han tenido en cuenta los casos típicos de corrección.

Finalmente, también es posible que cuando se cree una pregunta que no se indique como corregirla automáticamente, en este caso, es necesaria la intervención externa (por ejemplo del profesor) para llevar a cabo la corrección de la misma, es decir, para crear la evaluación de la pregunta. Habitualmente sólo se necesita indicar cuales de las opciones son correctas.

```

<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
- <assessmentItem timeDependent="false" adaptive="false" label="mylabel" title="Cerros de C
  xsi:schemaLocation="http://www.imsglobal.org/xsd/imsqti_v2p1_imsqti_v2p1.xsd" x
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.imsg
- <responseDeclaration baseType="identifier" cardinality="single" identifier="RESPONSE">
  - <correctResponse>
    <value>ID0</value>
  </correctResponse>
  - <mapping defaultValue="0.0">
    <mapEntry mappedValue="1.0" mapKey="ID0" />
  </mapping>
</responseDeclaration>
<outcomeDeclaration baseType="integer" cardinality="single" identifier="SCORE" />
<outcomeDeclaration baseType="identifier" cardinality="single" identifier="FEEDBACK" />
- <itemBody>
  - <choiceInteraction responseIdentifier="RESPONSE" maxChoices="1" shuffle="true">
    - <prompt>
      <br />
      
      Cerro de la figura
    </prompt>
    - <simpleChoice fixed="false" identifier="ID0">
      Cerro Castillo
      <feedbackInline identifier="ID0" showHide="show" outcomeIdentifier="FEEDBACK">co
      <feedbackInline identifier="ID0" showHide="hide" outcomeIdentifier="FEEDBACK" />
    </simpleChoice>
  </choiceInteraction>

```

Figura 3.2: AssesmentItem XML en QTI.

### 3.3.2 Interacciones

En el caso de IMS QTI no está contemplado el concepto de tipo de pregunta, existiendo en su lugar el concepto de interacción. Las interacciones permiten al profesor especificar las herramientas que tendrá el alumno disponible para poder construir la respuesta. Al igual que existen múltiples tipos de pregunta, también existen múltiples tipos de interacción. A continuación se describirán los tipos de interacciones posibles que pueden utilizarse dentro de una pregunta.

Hay que remarcar que existen dos grupos de interacciones, las interacciones en línea y las interacciones en bloque. Las interacciones en línea son un tipo de interacción que pueden incluirse en medio del enunciado de la pregunta. Por otra parte las interacciones de tipo bloque están pensadas para ser presentadas de manera independiente al enunciado de la pregunta.

Para ejemplificar las posibles interacciones que se nos ofrecen y debido a la falta de herramientas que proporcionen un soporte completo a la especificación, se hará uso de ejemplos basados a los que se ofrecen en la propia especificación.

### 3.3.2.1 Interacciones Simples

Las interacciones simples con aquellas interacciones en las que la corrección de las mismas se realiza en base a la selección de una opción o varias opciones disponibles. Las interacciones que pertenecen a esta categoría son: [IMS06d] [Fer05]

#### *choiceInteraction*

Esta interacción muestra al alumno un conjunto de posibles opciones. El alumno podrá seleccionar una o varias posibles opciones como respuesta. Es posible indicar que el conjunto de posibles opciones sea barajado entre distintos intentos del alumno.

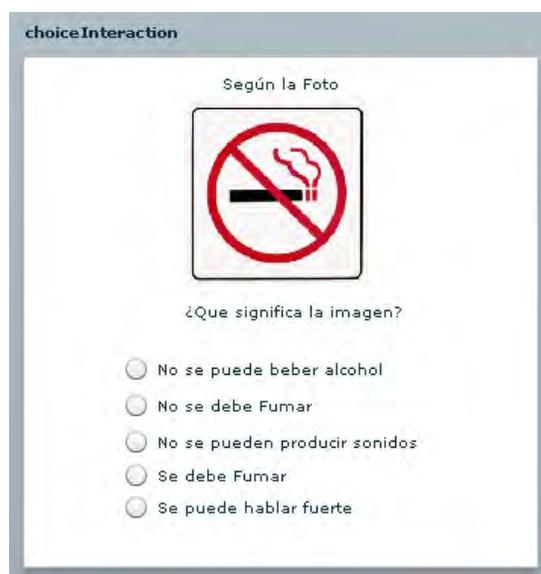


Figura 3.3: Ejemplo de *choiceInteraction*. Basado en [IMS06d]

#### *orderInteraction*

En esta interacción el objetivo del alumno es reordenar el conjunto de soluciones proporcionada. Además, es posible un número mínimo y un número máximo de opciones que conforman la solución, de manera que se realizaría una selección sobre las opciones disponibles y posteriormente se realizaría una ordenación de los elementos de dicha selección.



Figura 3.4: Ejemplo de *orderInteraction*. Basado en [IMS06d]

### ***associateInteraction***

Esta interacción presenta al alumno un conjunto de opciones y permite crear asociaciones por parejas entre dichas opciones. Es posible indicar el número mínimo y máximo de asociaciones que deben crearse como parte de la respuesta. Además, también es posible indicar el número mínimo y máximo de veces que una de las opciones puede aparecer dentro de una asociación.



Figura 3.5: Ejemplo de *associateInteraction*. Basado en [IMS06d]

### ***matchInteraction***

Esta interacción presenta al alumno dos conjuntos de opciones y le permite crear pares de asociaciones entre ellas. Al igual que en la interacción anterior es posible indicar el número mínimo y máximo de asociaciones posibles o el número mínimo y máximo de apariciones de una de las opciones en las asociaciones creadas.



Figura 3.6: Ejemplo de *matchInteraction*. Basado en [IMS06d]

### *gapMatchInteraction*

Esta interacción permite definir un conjunto de espacios en blanco dentro del enunciado de la pregunta a mostrar al alumno. Además se permitirá al alumno asociar a cada uno de los espacios una de las posibles opciones de respuesta. Hay que destacar que las opciones posibles son compartidas por todos los espacios. Como posibles respuestas es posible utilizar texto o también es posible utilizar imágenes. Además es posible restringir el número mínimo y máximo de veces que es utilizada cada una de las posibles opciones del conjunto de respuestas.

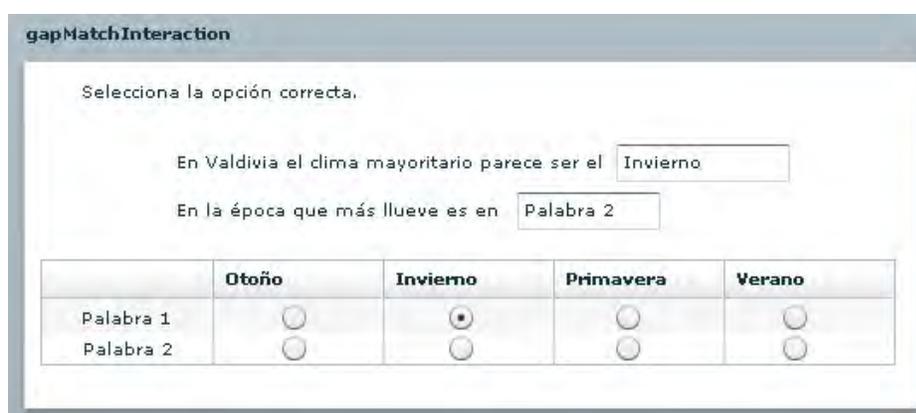


Figura 3.7: Ejemplo de *gapMatchInteraction*. Basado en [IMS06d]

### 3.3.2.2 Interacciones de Texto

En esta categoría se encuentran las interacciones en las que la respuesta que construirá el alumno puede ser una única palabra, una frase corta o un párrafo de texto completo. Estas interacciones permiten que durante el proceso de corrección se tenga en cuenta la respuesta en forma de texto que ha construido el alumno. Las interacciones que pertenecen a esta categoría son:

### ***inlineChoiceInteraction***

Esta interacción está pensada para definir un espacio donde se permitirá al usuario escoger entre un conjunto de opciones, donde cada una de estas opciones una palabra o frase corta. A diferencia de *gapMatchInteraction*, esta interacción está ideada para que cada uno de los espacios pueda tener un conjunto de opciones independiente. Es posible definir que las respuestas sean barajadas entre distintos intentos del alumno.

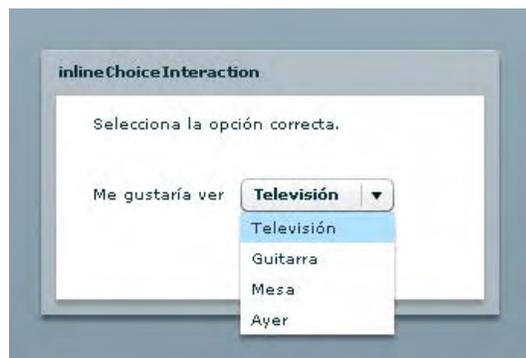


Figura 3.8: Ejemplo de *inlineChoiceInteraction*. Basado en [IMS06d]

### ***textEntryInteraction***

Al igual que la interacción anterior, esta interacción tiene como objetivo crear un espacio donde se permitirá teclear una palabra o frase corta para poder construir la respuesta. Cuando se define una pregunta con esta interacción es posible especificar la longitud del texto que se espera que el alumno introduzca.

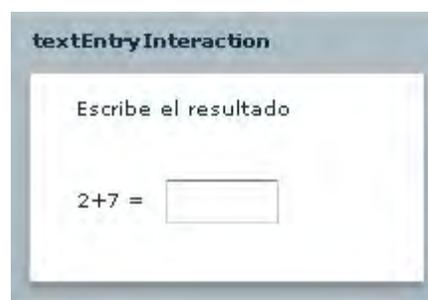


Figura 3.9: Ejemplo de *textEntryInteraction*. Basado en [IMS06d]

### ***extendedTextInteraction***

Esta interacción está pensada para que el alumno construya como respuesta un párrafo de texto. Es posible indicar el número mínimo y máximo de líneas de texto esperadas, junto con la longitud máxima de cada una de ellas.

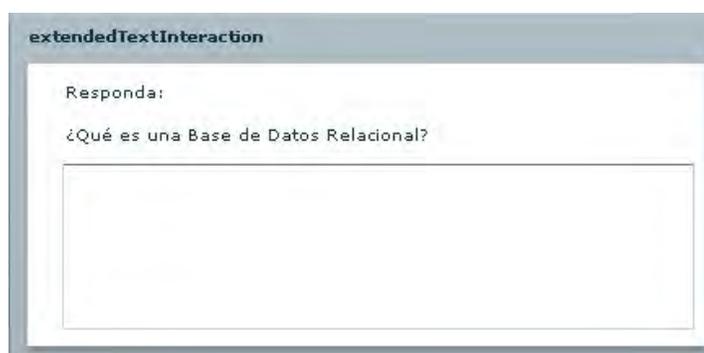


Figura 3.10: Ejemplo de *extendedTextInteraction*. Basado en [IMS06d]

### ***hottextInteraction***

El objetivo de esta interacción es que el alumno seleccione partes de texto que estarán resaltadas en el enunciado de la pregunta. Es posible indicar el número mínimo y máximo de elecciones que debe realizar el alumno, siendo 0, el valor mínimo, y 1, el valor máximo, de selecciones por defecto.

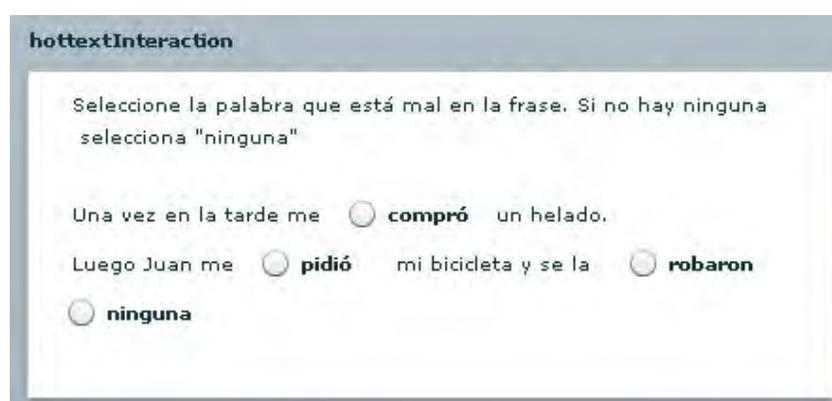


Figura 3.11: Ejemplo de *hottextInteraction*. Basado en [IMS06d]

### **3.3.2.3 Interacciones Gráficas**

Las interacciones gráficas tienen como elemento principal una imagen que se utilizará como fondo del enunciado y sobre la que se realizarán todas las acciones permitidas en las interacciones para que el usuario construya la respuesta. Las interacciones que pertenecen a esta categoría son:

### ***hotspotInteraction***

El objetivo de esta interacción es presentar al alumno un conjunto de “puntos críticos” (del inglés *hotspot*) sobre una imagen utilizada como fondo del enunciado. El

alumno deberá seleccionar uno o varios de estos puntos críticos para construir la respuesta. Es posible especificar el número mínimo y máximo de selecciones que debe realizar el alumno, siendo 0, el valor mínimo, y 1, el valor máximo, de selecciones por defecto.

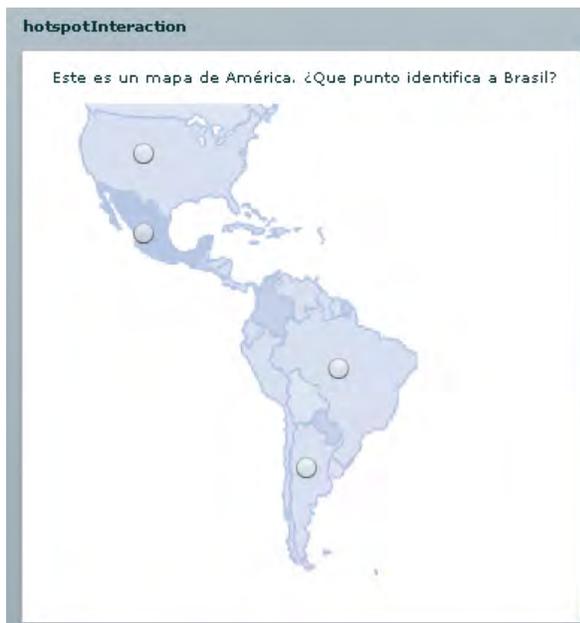


Figura 3.12: Ejemplo de *hotspotInteraction*. Basado en [IMS06d]

### ***selectPointInteraction***

El objetivo de esta interacción es que el usuario seleccione uno o varios puntos de una imagen utilizada como fondo del enunciado. Al contrario que en la interacción anterior, no se le presentará al alumno ninguna zona resaltada.

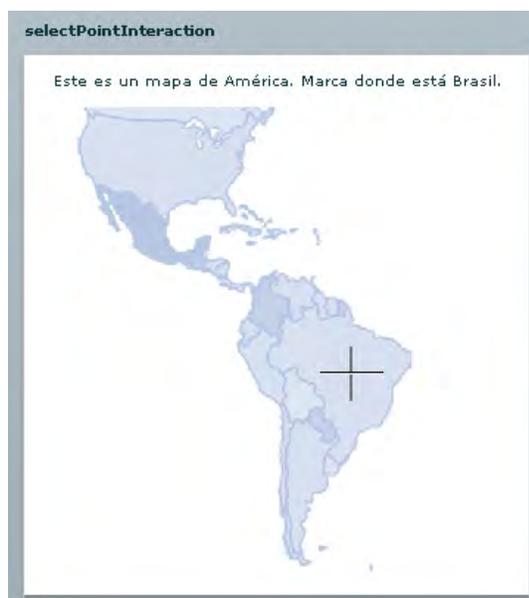


Figura 3.13: Ejemplo de *selectpointInteraction*. Basado en [IMS06d]

### ***graphicOrderInteraction***

Esta interacción mostrará un conjunto de puntos críticos sobre una imagen que será utilizada como fondo del enunciado. El objetivo es que el usuario realice una ordenación de estos puntos críticos. Al igual que la interacción *orderInteraction*, es posible definir el número mínimo y máximo de opciones que formarán parte de la respuesta, de manera que el usuario primero seleccionará las opciones y posteriormente realizará la ordenación de las mismas.



Figura 3.14: Ejemplo de *graphicOrderInteraction*. Basado en [IMS06d]

### ***graphicAssociateInteraction***

Esta interacción mostrará un conjunto de zonas seleccionables o puntos críticos sobre una imagen que será utilizada como fondo del enunciado. El objetivo de la interacción es permitir al alumno la creación de pares de asociación entre los puntos críticos. Es posible indicar el número máximo de asociaciones que el alumno puede crear. Asimismo también es posible indicar el número mínimo y máximo de cada uno de los puntos críticos dentro de las asociaciones creadas.



Figura 3.15: Ejemplo de *graphicAssociateInteraction*. Basado en [IMS06d]

### ***graphicGapMatchInteraction***

Esta interacción mostrará un conjunto de puntos críticos sobre una imagen que será utilizada como fondo del enunciado, además se proporcionará al usuario un conjunto de opciones. El objetivo es que el usuario construya parejas entre los puntos críticos y las opciones que le son proporcionadas. Hay que destacar que el conjunto de opciones disponibles es compartido por todos los puntos críticos. Asimismo, es posible definir el número mínimo y máximo de veces que puede aparecer una de las opciones en una de la parejas creadas por el usuario.



Figura 3.16: Ejemplo de graphicGapMatchInteraction. Basado en [IMS06d]

### ***positionObjectInteraction***

En esta interacción el alumno colocará una imagen le será proporcionada sobre alguna zona de otra imagen será utilizada como fondo del enunciado. Esta interacción es similar a la interacción *selectPointInteraction*, ya que tienen como objetivo seleccionar puntos de la imagen que se utiliza como fondo, en el caso de la interacción *positionObjectInteraction* esta posición seleccionada se marcará con la imagen que se le proporciona al usuario. Es posible definir el número mínimo y máximo de posibles selecciones que puede realizar el usuario.



Figura 3.17: Ejemplo de *positionObjectInteraction*. Basado en [IMS06d]

### 3.3.2.4 Otros Tipos de Interacciones

En esta categoría se encuentran interacciones relativamente avanzadas. Las interacciones que pertenecen a esta categoría son:

#### *sliderInteraction*

Esta interacción muestra al alumno una barra deslizante que permitirá al usuario seleccionar la respuesta correcta.

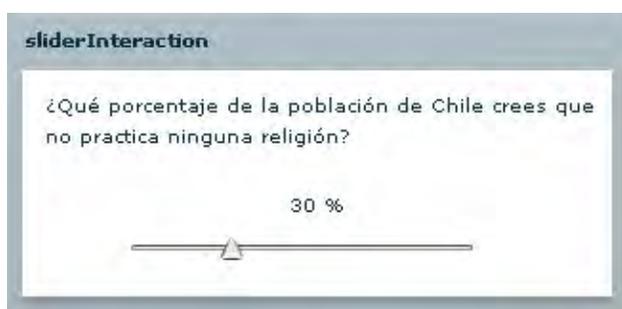


Figura 3.18: Ejemplo de *sliderInteraction*. Basado en [IMS06d]

#### *mediaInteraction*

Esta interacción está pensada para ser utilizada de manera conjunta con alguna otra interacción. El objetivo de esta interacción es permitir controlar el número de veces que un usuario visualiza un material multimedia, de esta manera es posible incluir esta interacción con un video como enunciado, e incluir alguna otra interacción que realmente realice una pregunta acerca del enunciado. El número de veces que el usuario

visualice el vídeo puede utilizarse para cambiar la interacción o simplemente puede ser utilizada para posteriormente realizar un estudio estadístico acerca del número de veces que es necesario visualizar el vídeo para poder responder a la pregunta.

### ***drawingInteraction***

Esta interacción tiene como objetivo permitir al usuario pintar sobre una imagen proporcionada en el enunciado. El resultado de la pregunta será la propia imagen modificada.

### ***uploadInteraction***

Esta interacción tiene como objetivo permitir al usuario crear una respuesta a partir de un fichero que será enviado a la herramienta desde el computador del usuario.

### ***customInteraction***

Esta interacción tiene como objetivo servir como base para la creación de interacciones particulares de cada herramienta que no se puedan encuadrar a ninguna de las interacciones descritas anteriormente.

## **3.3.3 Exámenes en IMS QTI V 2.1**

Un examen para IMS QTI es simplemente un grupo de preguntas. Durante el proceso de creación se puede estructurar el examen en distintas partes (testPart). Por ejemplo, si se ha impartido un módulo de un curso en el cual se discuten distintos temas, se puede crear un examen en el que se crean distintas partes por cada uno de esos temas. Asimismo, estas partes pueden dividirse en distintas secciones (sections) que representan diferentes secciones dentro de una parte del examen. Tanto las partes como las secciones de un examen pueden contener materiales que serán presentados a los usuarios durante la realización de dicha parte o sección. [IMS06d]

El objetivo de las partes del examen es doble:

- Desde el punto del usuario, cuando realice el examen se le presentarán cada una de las partes en las que está dividido el examen en el orden en el que aparecen en la definición del examen.
- Desde el punto de vista del creador del examen, el uso de distintas partes permite configurar distintas partes del examen, por ejemplo, indicando alguna limitación en el tiempo que puede emplear un alumno en una parte o sección (para mayor detalle ver figura 3.19).

Además de la estructuración, un objetivo adicional es la generación de una única evaluación, es decir, de una nota que agrupe todas las evaluaciones individuales de las preguntas, ponderándolas con algún factor si fuera necesario. Para ello, durante la creación del examen se puede definir como ha de realizarse la agrupación de las evaluaciones individuales.

Finalmente se puede destacar que desde el punto de vista de QTI la creación de un examen se realiza de manera completamente independiente de las preguntas de las que está compuesto. Al intercambiar un examen entre la herramienta de autoría y la herramienta que interpreta la descripción del examen, este intercambio se realiza creando un paquete de intercambio definido mediante el uso de la especificación IMS Content Packaging.

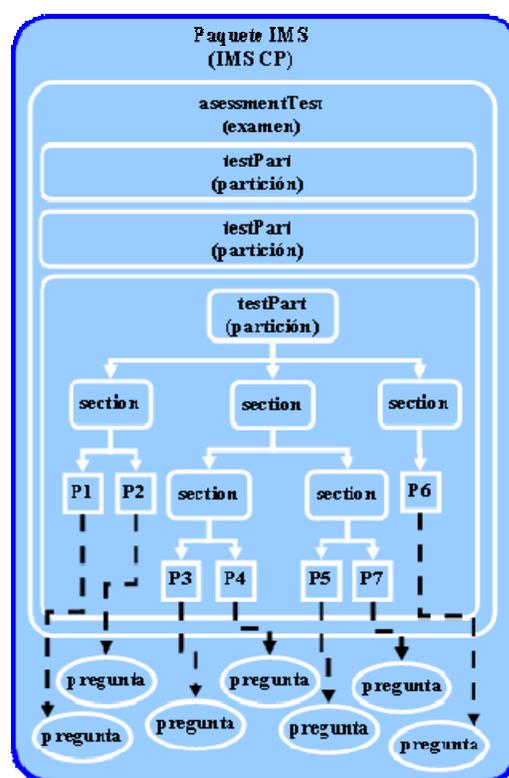


Figura 3.19: Estructura de un examen (*assessmentTest*) completo. Obtenida de [Fer05]

### 3.3.4 Resultados y Estadísticas

Además de permitirse el intercambio de la definición de las preguntas y de los exámenes, IMS QTI también permite el intercambio entre distintas herramientas de un informe en el que se incluyen los resultados que ha obtenido un alumno cuando ha realizado un examen concreto o cuando ha respondido a una o varias preguntas aisladas. Dentro del informe se puede incluir la siguiente información:

- Identificación del usuario. Se incluye la información necesaria para identificar al alumno, esta información se encuentra codificada utilizando la especificación *IMS Learning Information Profile*.
- Identificación del examen y/o preguntas. Se incluye la información necesaria para identificar que herramienta ha generado el informe, además de la información necesaria para identificar los exámenes y las preguntas a las que hace referencia este informe.
- Informe acerca de las preguntas individuales realizadas por el usuario. Esta información incluye las respuestas que ha creado el alumno a partir de las interacciones, la información relativa a las respuestas correctas, si se ésta encuentra disponible, y finalmente la información generada tras la corrección de las respuestas proporcionadas por los usuarios.
- Información acerca de los exámenes. Esta información incluye el resultado de la agregación de las evaluaciones de las preguntas individuales.

Además de la información específica para un usuario, examen y preguntas particulares, también es posible realizar el intercambio de información estadística acerca de los resultados obtenidos por los usuarios en exámenes y preguntas. La información que se intercambia incluye información como el número de muestras a partir de las que se ha generado la información estadística, la media aritmética, la varianza o a desviación típica. Además también es posible intercambiar los resultados de la aplicación de otros cálculos estadísticos habituales o incluso la aplicación de cálculos estadísticos adaptados a nuestras propias necesidades.

Para simplificar su intercambio se hace uso de la especificación *IMS Vocabulary Definition Exchange* para poder describir, tanto para las herramientas software, cómo para los actores humanos, los algoritmos estadísticos.

## **3.4 Conceptos Avanzados**

Además de las posibilidades vistas hasta ahora la especificación IMS QTI permite crear preguntas y exámenes utilizando funcionalidades avanzadas, las cuales requieren tanto un mayor conocimiento de la propia especificación como un mayor esfuerzo desde el punto de vista técnico. Debido a estos requisitos es posible que alguna de las características no estén disponibles en todas las herramientas de gestión de evaluaciones que sean compatibles con QTI.

A continuación se describen las características avanzadas más relevantes.

### **3.4.1 Intentos y Sesiones de Interacción**

El proceso de interacción típico de usuario con una pregunta, denominado intento, consta de tres pasos:

- Primero. Se presenta al usuario el enunciado de la pregunta, incluyendo algún contenido adicional que especifique el contexto de la pregunta.
- Segundo. El usuario construye la respuesta utilizando las herramientas proporcionadas por la interacción que el profesor ha elegido para la pregunta.
- Tercero. El usuario envía la respuesta para que esta sea corregida.

Habitualmente, sólo se lleva a cabo un intento por pregunta, sin embargo, en la especificación se ha tenido en cuenta la posibilidad de que un usuario realice varios intentos sobre la misma pregunta, denominándose a este conjunto de intentos como sesión de interacción. Además también está contemplado que un usuario pueda suspender el intento, es decir que el usuario puede construir parcialmente la respuesta pero sin enviar la respuesta para su corrección. De esta manera, el usuario puede pasar a otra pregunta del examen y posteriormente puede volver a la pregunta suspendida y terminar el intento.

Los intentos son importantes ya que dentro del proceso de corrección se va permitir la posibilidad de presentar nueva información al usuario entre intentos haciendo uso del mecanismo de realimentación. Asimismo, es posible modificar incluso la propia pregunta, por ejemplo, al fallar los dos primeros intentos podría darse un tercer intento

final en el cual el enunciado de la pregunta y la interacción a utilizar para construir la pregunta fueran distintas a la original.

### 3.4.2 Preguntas Dinámicas

Cuando se diseña una pregunta, se puede hacer considerando que un usuario pueda interactuar varias veces con dicha pregunta. La idea es que durante el proceso de creación de la pregunta se podrá añadir contenidos e interacciones adicionales que en el primer intento del usuario van a estar ocultas pero que en intentos sucesivos pueden aparecer según las respuestas del usuario y los resultados del proceso de corrección posterior.

Al tipo de preguntas donde el proceso de corrección y el contenido de la pregunta se modifica dependiendo de cada intento del alumno son denominadas preguntas adaptativas. [Fer05]

El mecanismo de realimentación puede ser utilizado en numerosas situaciones, algunas de ellas son:

- Cuando un usuario ha fallado una pregunta, es posible mostrar un mensaje indicando que ha fallado la pregunta, o se puede incluir alguna pista para que el usuario pueda responder correctamente a la pregunta.
- Tras finalizar el último intento de la pregunta, es posible mostrar un mensaje al alumno informándole de cual era la respuesta correcta.
- En el último intento de una pregunta, es posible hacer que aparezca una nueva interacción en la que se permita al usuario construir una respuesta de manera más simple, de forma que el resultado final de la pregunta será la evaluación de esta nueva interacción.

Algunas interacciones como, por ejemplo, la interacción *choiceInteraction* en la que se describe cada una de las posibles respuestas de la pregunta, es posible modificar dicha descripción. Además tras el proceso de corrección se puede mostrar un mensaje indicando que dicha opción es incorrecta, de manera que el mensaje se mantendrá durante los restantes intentos del alumno.

El uso de varias interacciones dentro de una pregunta, no está restringido sólo a las preguntas adaptativas. En el proceso de creación de una pregunta en la que el usuario no tenga varios intentos posibles, también es posible hacer uso de varias interacciones

independiente dentro de la definición de la misma pregunta, de forma que las interacciones son evaluadas de manera independiente. Este tipo de preguntas son denominadas *preguntas compuestas*. [Fer05]

Por ejemplo, el profesor puede utilizar esta característica para definir alguna pregunta adicional que utilice el mismo enunciado, por ejemplo, para poder asignar una puntuación extra. Es necesario resaltar que cada una de estas interacciones se corregirá de manera independiente.

### **3.4.3 Exámenes Dinámicos**

De manera adicional a la creación de preguntas dinámicas, es posible definir exámenes dinámicos. Cuando se diseña un examen, existen dos posibilidades para crear exámenes dinámicos dependiendo de si el examen se modifica antes de que el usuario intente responder a alguna de las preguntas que lo componen o bien la modificación del examen que se realizará durante la propia realización del examen. La modificación de un examen puede llevarse a cabo de manera previa a la realización del examen, donde en este caso el examen actúa como plantilla. El resultado final es que las secciones del examen se ven afectadas dependiendo del resultado de la aplicación de un conjunto de reglas. Estas reglas estarán definidas en las secciones del examen y afectarán a las otras secciones y preguntas que puede contener una sección. La idea es que dentro de las secciones se ordenarán de manera aleatoria las secciones y preguntas contenidas, siendo posible seleccionar un número de ellas posteriormente a la reordenación de las mismas.

Por ejemplo, en la figura 3.20 se observa la estructura de una parte de un examen en el momento de la autoría (figura 3.20, 1). Posteriormente (figura 3.20, 2) se seleccionan las dos primeras secciones de una parte del examen, donde también es posible observar como en la sección S4 las dos preguntas han sido reordenadas.

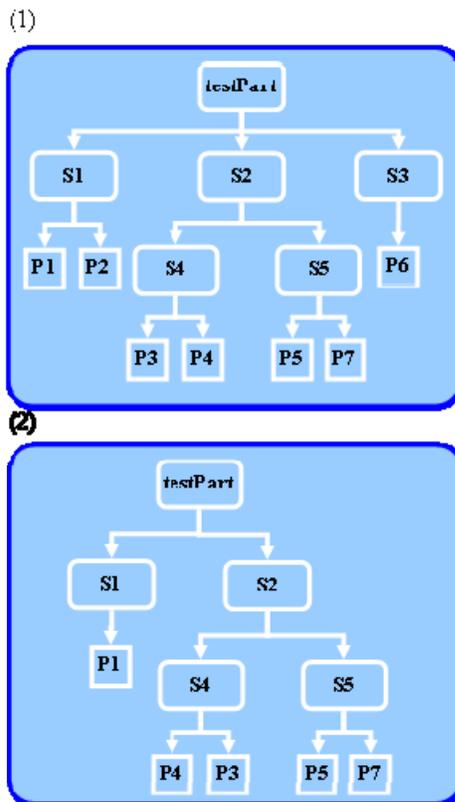


Figura 3.20: Ejemplo de examen dinámico. Obtenida de [Fer05]

Asimismo, los exámenes que se modifican dinámicamente durante la realización del examen son denominados exámenes adaptativos. El usuario atravesará las partes del examen dependiendo de los resultados de la corrección de las preguntas. Por ejemplo, es posible crear un examen con dos partes, de manera que si no se obtiene una puntuación mínima en una parte del examen no se puede avanzar a la siguiente parte del mismo.

### 3.4.4 Bancos de preguntas

En las versiones anteriores de IMS QTI se definía con una notación específica el formato de intercambio de los bancos de preguntas. Sin embargo, en la versión 2 de la especificación se ha tenido en cuenta el uso de otras especificaciones y la metodología utilizada para el intercambio de información realizada en el resto de especificaciones de IMS. [Fer05]

Por ello, para realizar el intercambio de bancos de preguntas, simplemente es necesario crear cada una de las preguntas de manera individual y posteriormente utilizar la especificación IMS *Content Packaging* para intercambiar cada una de las preguntas como un recurso independiente dentro de la definición del paquete IMS.

La figura 3.21 muestra un Paquete de intercambio compatible con la especificación IMS *Content Packaging*. El archivo *imsmanifest.xml* hará referencia a otros archivos que estarán dentro del paquete y que contendrán la definición de la pregunta.

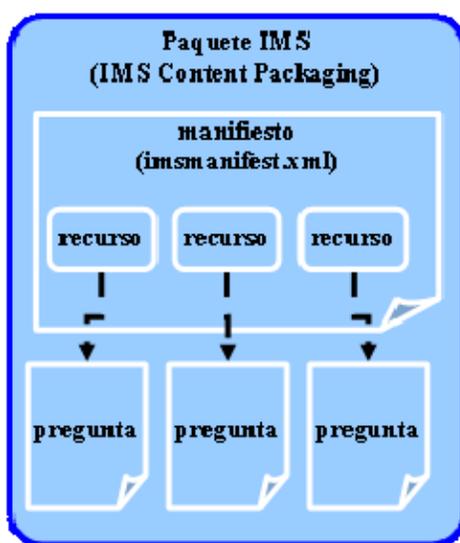


Figura 3.21: Paquete IMS *Content Packaging*. Obtenida de [Fer05]

### 3.5 Relación con otras Especificaciones

Parte de los objetivos de la versión 2 de QTI es hacer uso de las especificaciones existentes para evitar solapamientos en las funcionalidades que ofrecen.

Algunas de las especificaciones y estándares que están relacionados con QTI son: [Fer05]

- IMS *Metadata*, IEEE LOM. Al igual que el resto de contenidos educativos, es necesario la inclusión de meta información que permita clasificar los contenidos. En el caso de las preguntas y exámenes la meta información es muy útil a la hora de almacenar las preguntas dentro de los bancos de preguntas, de manera que se podría clasificar las preguntas según materias y dificultades, facilitando la búsqueda dentro de dichos repositorios y la posible generación automática de exámenes.
- IMS *Learning Information Package*. Como resultado de la realización de un examen se genera información relativa al expediente académico del alumno, además dentro de IMS QTI hace falta de alguna manera hacer referencia al alumno que ha realizado el examen.

- *IMS Learning Design*. La especificación hace uso de las condiciones para poder modificar la Unidad de Aprendizaje dependiendo de la evaluación de estas condiciones. Las condiciones incluyen la consulta de propiedades que han sido definidas por el creador de la Unidad de Aprendizaje para almacenar información según interactúe el alumno con la Unidad de Aprendizaje. Una posible actividad en una Unidad de Aprendizaje puede ser la realización de un examen, donde la puntuación de este examen será almacenada en alguna propiedad de la Unidad de Aprendizaje, de manera que el cambio de esta propiedad afecta a las condiciones y por tanto condicionará la puesta en práctica de otras actividades.
- *IMS Simple Sequencing, SCORM*. Al igual que con *IMS Learning Design*, tanto en *IMS Simple Sequencing* como en el caso particular del perfil de aplicación SCORM, existe un conjunto de propiedades que condicionan como se secuenciarán los Objetos de Aprendizaje. Al igual que en el caso de *IMS Learning Design*, podemos condicionar que se le presente al usuario uno o varios Objetos de Aprendizaje dependiendo de la nota que haya obtenido en el examen

### 3.6 Herramientas Relacionadas

Existen diversas herramientas que proporcionan soporte en mayor o menor medida a la especificación IMS QTI en sus diferentes versiones. Las herramientas pueden ser divididas en tres categorías: [Fer05]

- **Herramientas de Autoría.** Herramientas que permitirán crear preguntas y exámenes y finalmente salvarlos en el formato IMS QTI.
- **LMS.** En este caso los LMS incluirán una herramienta de gestión de evaluaciones en línea donde la herramienta permitirá importar y/o exportar las preguntas y exámenes en formato IMS QTI.
- **Reproductores y motores de ejecución.** Los reproductores interpretarán preguntas y exámenes compatibles con IMS QTI y permitirán realizar el examen. Los motores de ejecución, proporcionan infraestructura y componentes varios para la creación de herramientas de evaluación en línea, en este caso compatibles con IMS QTI.

Algunas herramientas de autoría son:

- Aqurate (<http://aqurate.kingston.ac.uk/>)
- Respondus (<http://www.respondus.com>)
- QuestionMark's Perception (<http://www.questionmark.com>)
- Assesst Designer (<http://www.xdlsoft.com/ad/>)

Algunos LMS que proporcionan soporte para IMS QTI:

- Dokeos (<http://www.dokeos.com/>)
- Claroline (<http://www.claroline.net>)
- SAKAI (<http://www.sakaiproject.org>)
- .LRN (<http://www.openacs.org>)
- OLAT (<http://www.olat.org>)
- WebCT (<http://www.webct.com>)

Sistemas que permiten la visualización y ejecución de QTI:

- ASDEL (<http://www.asdel.ecs.soton.ac.uk/>)
- <e-QTI> (<http://www.e-ucm.es/projects/eQTI>)
- Q-Player (<http://www.e-teach.ch/qplayer/>)
- APIS (<http://apis.sourceforge.net/>)
- R2Q2 (<http://www.r2q2.ecs.soton.ac.uk/>)
- MathQTI (<http://mqat.sourceforge.net/>)

La presente tesis se enfoca en la herramienta AQURATE, que se verá en detalle en el siguiente capítulo.

## 4. EL PROYECTO AQURATE

### 4.1 Introducción

AQURATE es un proyecto desarrollado desde Marzo del 2007 a Marzo del 2008 por la Universidad de Kingston en Londres, Inglaterra. El proyecto AQURATE fue fundado como parte de la sección del Programa Capital de Tecnologías de Aprendizaje del Comité Unido de Sistemas de Información, JISC<sup>17</sup> (del inglés *Joint Information Systems Committee*). JISC apoya los establecimientos educacionales (de alumnos por sobre los 16 años, referente a la educación media en Chile) y de educación superior de Reino Unido proveyendo liderazgo en el uso de Tecnologías de Información y Comunicación en el apoyo del aprendizaje, enseñanza, investigación y administración.

La meta de AQURATE es ayudar la aceleración en la adopción de la especificación IMS QTI a través de las distintas disciplinas en la educación superior de Reino Unido. Recientemente JISC ha auspiciado el desarrollo de la especificación IMS QTI y han financiado iniciativas para adelantar el uso de IMS QTI en el dominio de

---

<sup>17</sup> <http://www.jisc.ac.uk/>

exámenes. Uno de esos proyectos fue JELFAD<sup>18</sup> (del inglés *JISC eLearning Framework Assessment Demonstrator*), desarrollado en la Universidad de Kingston (2005), el que tenía como objetivo principal secuenciar un set de materiales de aprendizaje incorporando tipos de preguntas IMS QTI versión 2.1, lo que arrojó como resultado la necesidad de una interfaz amigable para la creación de preguntas.

El proyecto FREMA<sup>19</sup> (de inglés *e-learning Framework Reference Model for Assessment*), desarrollado en la Universidad de Southampton (2005), tenía como meta identificar como las tecnologías de aprendizaje pueden facilitar el aprendizaje y como tales tecnologías de aprendizaje deberían ser implementadas. Uno de los objetivos de FREMA era identificar los procesos en las exámenes electrónicas (*eAssessment*) las cuales arrojaron tres líneas de desarrollo: Creación, Bancos y entrega de Elementos.

AQURATE se desarrolló usando el Método Dinámico de Desarrollo de Sistemas, DSDM<sup>20</sup> (del inglés *Dynamic Systems Development Method*) para asegurar la administración, el involucramiento de los usuarios, riesgos y la entrega del trabajo a tiempo. DSDM es un desarrollo del software originalmente basado en la metodología de Desarrollo Ágil de Aplicaciones, RAD (del inglés *Rapid Application Development*). DSDM es un acercamiento iterativo e incremental que enfatiza involucramiento continuo del usuario. Su meta es entregar soportes lógicos a tiempo al ajustarse a cambios en los requisitos a lo largo del proceso de desarrollo.

## 4.2 Descripción del Proyecto

El proyecto AQURATE desarrolló una herramienta de código abierto y multiplataforma para la creación de varios tipos de preguntas en base a la especificación IMS QTI v 2.1. El principal objetivo de este proyecto es la implementación de una aplicación de escritorio que posea las siguientes funcionalidades: [JIS07a] [AQU07a]

- La creación y edición de preguntas o ítems que cumplan el estándar IMS QTI versión 2.1.
- La importación y exportación de preguntas desde y hacia otros sistemas externos a la herramienta.
- El empaquetamiento de preguntas de acuerdo con las especificaciones IMS *Content Packaging* y SCORM 2004 con el fin de facilitar su distribución.

---

<sup>18</sup> <http://www.jisc.ac.uk/publications/publications/jelfadfinalreport.aspx>

<sup>19</sup> <http://www.frema.ecs.soton.ac.uk/>

<sup>20</sup> <http://www.dsdm.org>

- El almacenamiento en repositorios remotos de preguntas QTI 2.1 así como la recuperación de las mismas.
- La posibilidad de comprobar la presentación y funcionamiento de las preguntas QTI 2.1 usando uno o varios sistemas de validación QTI, o bien usando otros sistemas de presentación y entrega de contenido.
- Garantizar la extensibilidad de la herramienta basando su diseño en una arquitectura de doble capa, de manera que se mantenga separados los componentes de interfaz de usuario de la lógica de la aplicación.

AQURATE produjo una aplicación de escritorio amigable en un ambiente de autoría de preguntas en QTI como una alternativa a aplicaciones comerciales de exámenes (*eAssessment*). El proyecto aspira a soportar todos los tipos de preguntas QTI pero el foco fue la producción de una aplicación que provea al usuario una manera fácil de crear y editar los tipos de preguntas más usados.

El proyecto trabajó con otras dos iniciativas fundadas por JISC: Banco de preguntas, MINIBIX (Universidad de Cambridge) y la entrega y procesamiento de exámenes, ASDEL (Universidad de Southampton). Estos dos proyectos y su interacción con AQURATE serán detallados en la sección 4.4.

AQURATE fue dividido en dos fases, ambas de aproximadamente seis meses de duración. [JIS07a] La primera fase corresponde al desarrollo del núcleo de la aplicación de autoría, primordialmente una fase de desarrollo técnico. Como la conformación del equipo interno de AQURATE incluyendo los desarrolladores del sistema y los Asesores de grupo en conjunto con los otros proyectos relacionados. Además de la producción de la primera versión de AQURATE para ser presentada.

La segunda fase trató primordialmente la integración con los otros dos proyectos en cuanto a pruebas y la interoperabilidad con otros sistemas de la comunidad de JISC que contribuyen al dominio de exámenes.

El Proyecto fue diseñado con los siguientes objetivos a futuro [JIS07a]

- Creación *offline* de exámenes
- Interpretación y Procesamiento *offline* de exámenes
- Desarrollo de nuevas interfaces (*plug-ins*, servicios Web)
- Extensión y desarrollo de nuevas funcionalidades del sistema (nuevos tipos de preguntas, uso de contenidos secuenciales)

El equipo de trabajo de Aqurate está conformado por: [AQU07b]

- Graham Alsop. Administrador del Proyecto/ Factores Educativos
- Zhen Cai. Desarrollador
- James A, Annesley. Desarrollador
- Alicia Campos. Desarrollador
- Martin Colbert. Factores Humanos
- Dave Livingstone. Coordinador Técnico
- James Orwell. Asesor de Usuarios

En donde los roles de cada uno pueden ser identificados en la figura 4.1.

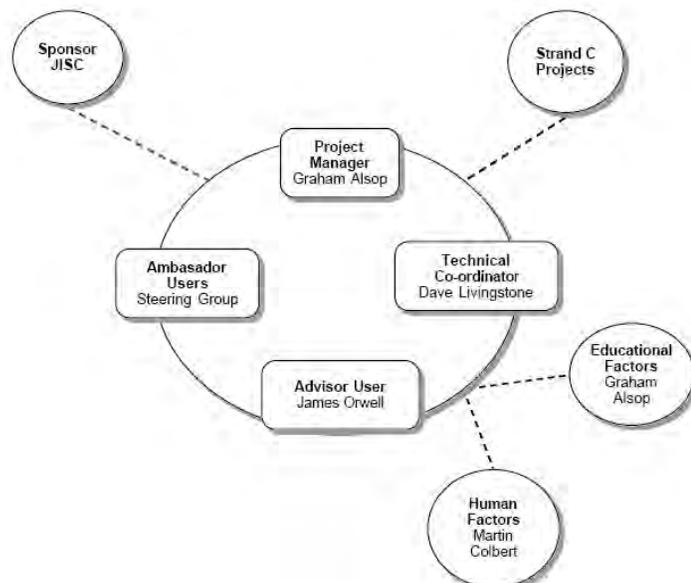


Figura 4.1: Equipo de trabajo del proyecto AQRATE. Obtenida de [JIS07a]

### 4.3 Integración de AQRATE con ASDEL y MINIBIX

AQRATE es uno de los tres proyectos financiados por JISC que, juntos, tenían como objetivo dar soporte a todo el proceso de evaluación académica, abarcando: la creación de preguntas (AQRATE, en la Universidad de Kingston), su almacenamiento en un repositorio (MINIBIX, en la Universidad de Cambridge) y finalmente, su entrega y presentación al usuario final (ASDEL, en la Universidad de Southampton). La figura 4.2 muestra un esquema general de como las diferentes herramientas se relacionan entre ellas y de cómo predominan claramente las tres líneas de desarrollo. La herramienta de Autoría, el Banco de Preguntas y la herramienta de entrega. Un escenario general sería: [JIS07a] [Cam08]

- Un Profesor ó Tutor escribirá preguntas (*items*). La herramienta de autoría proveerá una interfaz de usuario apropiada al usuario final, pero formatea y guarda los ítems de acuerdo a la especificación IMS QTI. Al ser guardados en formato IMS QTI permite que los mismos sean almacenados e intercambiados con otros sistemas
- Los usuarios pueden seleccionar artículos del banco de preguntas y agruparlos para construir un examen. El sistema construcción de exámenes, al igual que la herramienta de autoría, usará una interfaz de usuario apropiada y almacenará un examen de acuerdo a la especificación IMS QTI.
- Teniendo las preguntas y exámenes en la manera en que se especifica por IMS QTI, la implementación de preguntas, bancos de preguntas, y los exámenes de fuentes diversas pueden ser presentado a través del sistema de entrega a los usuarios vía un ambiente de enseñanza en un explorador de Internet.
- El usuario ahora puede tomar el examen, y los resultados serán reportados en una manera coherente.

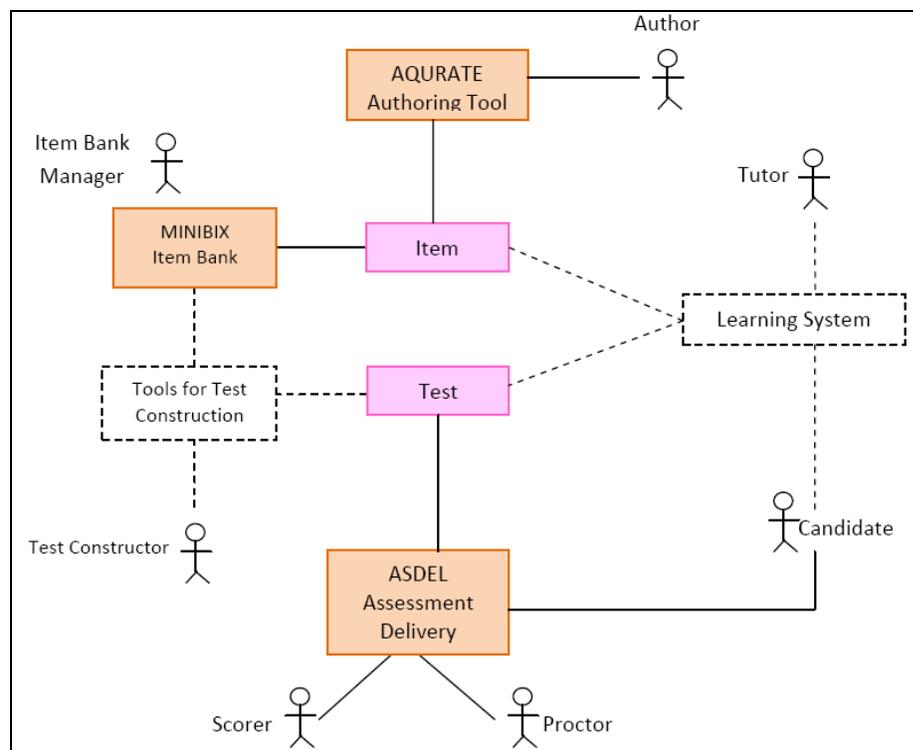


Figura 4.2: Integración de Aqurate con los proyectos relacionados. Obtenida de [Cam08]

### 4.3.1 El Proyecto ASDEL

El proyecto ASDEL (del inglés *Assesment Delivery Engine*) es un sistema que se propone para la presentación y procesamiento de preguntas bajo la especificación IMS QTI. En su arquitectura se especifica el uso de R2Q2, una herramienta financiada por JISC el año 2006. R2Q2, es un sistema de presentación y procesamiento de preguntas bajo la especificación IMS QTI, aunque sólo procesa preguntas individuales, puede tomarse como un componente esencial para futuros sistemas debido a su diseño y uso de servicios WEB, que facilitan el desarrollo y pueden fácilmente adaptarse a otras aplicaciones. [R2Q06]

Sin embargo, R2Q2 no implementa toda la especificación IMS QTI en términos de exámenes, por lo que ASDEL plantea una arquitectura como se muestra en la figura 4.3, en donde se tiene por un lado las acciones del usuario, el sistema de gestión de aprendizaje (LMS ó VLE del inglés *Virtual Learning Environment*) y el servicio de procesamiento de los exámenes (*Assessment Delivery Service*). Las interacciones del usuario con el sistema como Iniciar una sesión ó ver los resultados son parte de ASDEL, pero dejan a R2Q2 el despliegue y procesamiento de las preguntas.

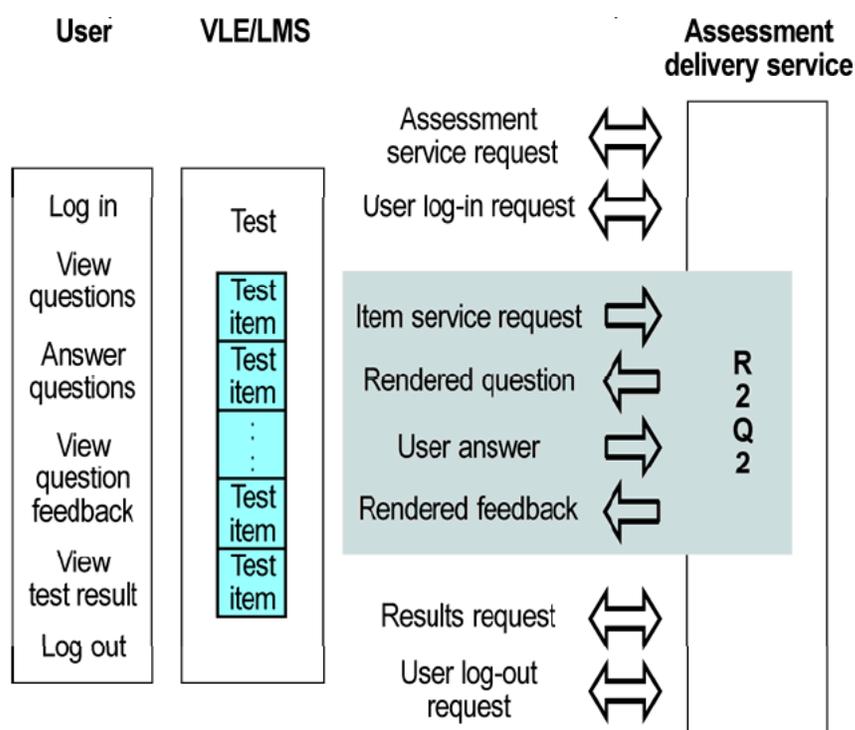


Figura 4.3: Visión General del procesamiento de un Test en ASDEL. Obtenida de [JIS07b]

El objetivo principal del proyecto es proveer de un motor de entrega y procesamiento de exámenes que se base completamente con la especificación IMS QTI Versión 2.1. El proyecto será desarrollado en dos fases.

- La primera fase es el desarrollo técnico del motor de procesamiento de acuerdo con la especificación IMS QTI, como por ejemplo, el uso de servicios WEB, la forma de empaquetar los contenidos, la forma de agregar contenidos (multimedia) a los exámenes, feedback y procesamiento del examen así como de los resultados.
- La segunda fase integrará el desarrollo de las herramientas AQURATE (Kingston) y MINIBIX (Cambridge) y producir una demostración en conjunto.

Durante el desarrollo del proyecto ASDEL publicó QTITOOLS<sup>21</sup>, que es una colección de herramientas, librerías y servicios Web de código abierto para la presentación, validación y procesamiento de preguntas y exámenes en base a la especificación IMS QTI versión 2.1

- JQTI<sup>22</sup>: Librería para la interpretación de documentos XML en base a la especificación IMS QTI versión 2.1. para el desarrollo de aplicación QTI.
- R2Q2<sup>23</sup>: Herramienta para la presentación y procesamiento de preguntas en QTI
- playr<sup>24</sup>: Herramienta para ejecutar exámenes en QTI.
- validatr<sup>25</sup>: Herramienta para validar exámenes QTI.
- assessr<sup>26</sup>: Herramienta para planificar exámenes y obtener los resultados.
- constructr<sup>27</sup>: Herramienta para construir exámenes con preguntas desde un banco de preguntas.

Con estas herramientas, la integración de las herramientas según [ASD07] se muestra en la figura 4.4.

---

<sup>21</sup> <http://www.qtitools.org>

<sup>22</sup> <http://jqti.qtitools.org/>

<sup>23</sup> <http://r2q2.qtitools.org/>

<sup>24</sup> <http://playr.qtitools.org/>

<sup>25</sup> <http://validatr.qtitools.org/>

<sup>26</sup> <http://assessr.qtitools.org/>

<sup>27</sup> <http://constructr.qtitools.org/>

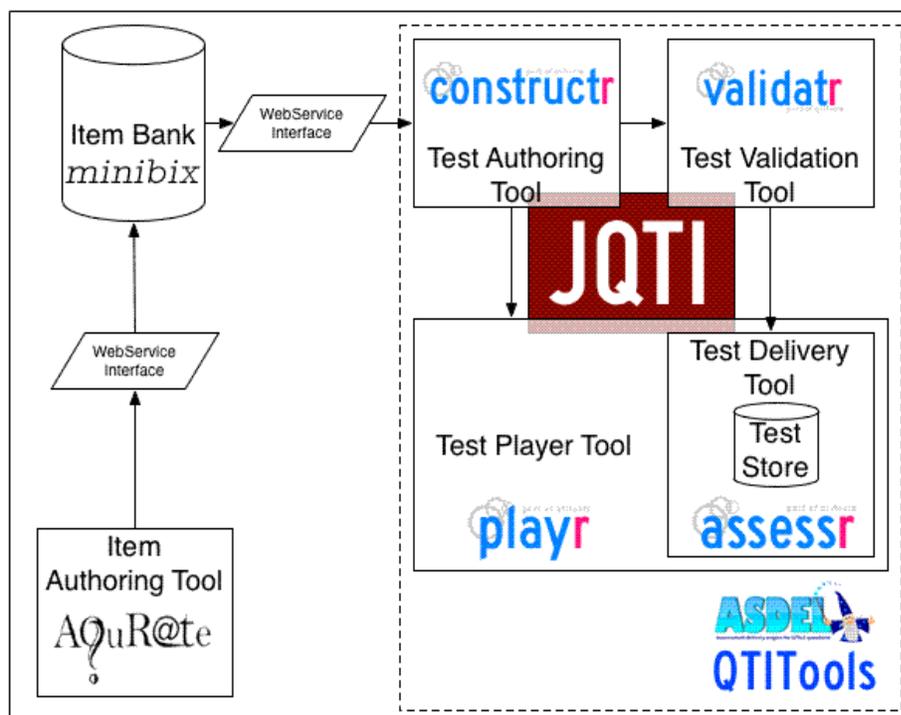


Figura 4.4: Integración de ASDEL con AQRATE y MINIBIX.

### 4.3.2 El Proyecto MINIBIX

La Universidad de Cambridge ha jugado un rol significativo en el desarrollo de la especificación IMS QTI. La Universidad ha sido miembro de IMS *Global Learning Consortium* desde el 2001, y últimamente bajo la marca *Cambridge Assessment*<sup>28</sup> del Sindicato Local de Examinaciones de la Universidad de Cambridge, UCLES (del inglés *University of Cambridge Local Examinations Syndicate*),

En el año 2003, comenzó un trabajo colaborativo entre UCLES y Centro para la Investigación Aplicada en Tecnologías Educativas, CARET<sup>29</sup> (del inglés *Centre for Applied Research in Educational Technologies*) en un sistema en línea para la administración y entrega de exámenes experimentales de admisión que son tomados por los interesados en ciertos cursos de pregrado al momento de la entrevista. El examen se llama Habilidades de Pensamiento para Admisiones, TSA (del inglés *Test Skills for Admissions*). A través del proyecto TSA, CARET desarrolló un sistema para la administración de modo dual, en línea y en papel. El examen se toma varios meses, teniendo la mayoría de los postulantes frente a una pantalla en Diciembre cada año. Esto

<sup>28</sup> <http://www.cambridgeassessment.org.uk/ca/>

<sup>29</sup> <http://www.caret.cam.ac.uk/>

requiere que existan varias versiones del mismo test cada año y por lo mismo un gran banco de preguntas.

El sistema de banco de preguntas que CARET desarrollo para apoyar el proyecto TSA fue el primer prototipo de un sistema banco de preguntas basado en QTI, en otras palabras “un banco de preguntas en XML” IBIX (del inglés *Item Bank in XML*). En ese entonces el desarrollo del sistema solo contemplaba simples preguntas de tipo *multiple Choice* para TSA y sólo soportaba la versión 1.0 de IMS QTI. Este prototipo del sistema de banco de preguntas pasó a llamarse MINIBIX.

El proyecto MINIBIX tiene la intención de desarrollar a un sistema Banco de Preguntas basado en la especificación IMS QTI Versión 2.1 capaz de soportar el intercambio de preguntas entre distintos sistemas. En el cual las funcionalidades previstas son las siguientes:

- Inserción, búsqueda y descarga de preguntas bajo la especificación IMS QTI.
- Seguimiento del uso de las preguntas, así como el etiquetado de preguntas por los usuarios.
- Administración de grupos de preguntas.
- Proveer los medios para que una herramienta de autoría de preguntas pueda agregar y actualizar preguntas.
- Recuperación de preguntas y grupos de preguntas usando la especificación IMS *Content Packaging*.

#### **4.4 Características de AQRATE**

Existen grandes diferencias entre las distintas versiones de la especificación QTI. Mientras que las primeras se centran en la definición y representación de la pregunta en sí, la última, IMS QTI versión 2.1, está enfocada a la definición de nuevos y distintos tipos de interacciones por parte del usuario (por ejemplo: ordenar un conjunto de elementos, introducir texto, emparejar los elementos de dos listas dadas, etc.) y a la creación de preguntas adaptativas, es decir, preguntas capaces de cambiar su forma, su puntuación y texto de retroalimentación en función de las respuestas dadas por el usuario.

Era esencial que la herramienta presentara las características de la nueva especificación completa y coherentemente. Sin embargo, debido al tiempo y recursos asignados al proyecto, era muy difícil abarcar todo el ámbito QTI en una sola aplicación

de escritorio. Se requería un análisis de los componentes de la especificación para establecer prioridades entre ellos y así, proporcionar el máximo beneficio a la comunidad de usuarios. Esto dio lugar a que la implementación se centrara en los tipos de interacción más demandados. [Cam08]

Por otro lado, tenía que encontrarse un equilibrio entre la usabilidad total de la herramienta y su capacidad de procesar elementos complejos del estándar (por ejemplo, “*ResponseProcessing*”). La gama resultante de preguntas que pueden crearse es una fortaleza de QTI versión 2.1, [Cam08], sin embargo:

- Desde la perspectiva del desarrollador, es muy complejo implementar una herramienta que incluya una funcionalidad completa en todos los escenarios definidos por la especificación.
- Desde la perspectiva del usuario, es difícil entender cómo configurar los elementos y atributos del estándar para cubrir sus necesidades y hacer un mejor uso de las posibilidades que éste ofrece.

La coherencia de QTI versión 2.1 ayuda a su implementación, pero al mismo tiempo su complejidad podría desanimar a algunos usuarios a trabajar con ella. De esta manera, se presentó la necesidad de implementar una herramienta de creación que sea funcional para usuarios principiantes (a través de asistentes) y para usuarios expertos.

#### **4.4.1 El Modelo de Datos**

La especificación IMS QTI, tal como se menciona en la estructura de las especificaciones, proporcionan el esquema documental XML que permite comprobar la validez de la estructura de un documento que haya sido creado. Un esquema es una especificación XML que gobierna los componentes admisibles de un documento XML y las relaciones entre los componentes. Por ejemplo, un esquema identifica los elementos que pueden aparecer en un documento XML, en qué orden deben aparecer, qué atributos que pueden tener, y cuáles elementos dependen de otros. Un documento XML válido debe conformar un esquema. El esquema de QTI define el modelo de datos de la especificación, pero para el desarrollo de software es necesario expresarlo en clases por lo que para implementarlas se utilizó la tecnología llamada JAXB<sup>30</sup> (del inglés *Java Architecture for XML Binding*). [Cam08] Esta tecnología proporciona una manera

---

<sup>30</sup> <http://java.sun.com/developer/technicalArticles/Webservices/jaxb/>

rápida y conveniente de crear vinculaciones entre documentos XML y objetos Java. Dado un esquema, que especifica la estructura de los datos XML, el compilador JAXB genera un conjunto de clases de Java que contienen todo el código para analizar los documentos XML basados en el esquema. JAXB requiere que el documento XML que se quiere acceder tenga un esquema, y ese esquema debe estar escrito en el lenguaje W3C XML<sup>31</sup>.

JAXB provee de dos funcionalidades principales. La habilidad de representar objetos Java en XML (*Marshall*) y viceversa (*Unmarshall*). En otras palabras, JAXB permite obtener y guardar datos en memoria en cualquier formato XML, sin la necesidad de implementar un set de métodos XML de carga y obtención de datos para la estructura de clases del programa en Java. [Ort03]

JAXB simplifica el acceso a un documento XML desde un programa escrito en Java representando el documento XML en el programa en formato Java. El primer paso es vincular el esquema XML, esto significa generar un set de clases que representa el esquema. Por ejemplo, el documento `books.xml` posee un esquema, `books.xsd`, que está escrito en lenguaje de esquemas W3C XML. Este esquema define `<Collection>` como un elemento que tiene un tipo complejo. Esto significa que tiene elementos hijos, en este caso elementos `<book>`. Cada elemento `<book>` también tiene un tipo complejo llamado `BookType`. El elemento `<book>` tiene elementos hijos como `<name>`, `<ISBN>`, y `<author>`.

Todas las implementaciones de JAXB proveen de un Compilador de Vinculaciones para vincular el esquema. Por ejemplo si se requiere vincular el esquema `books.xsd`, en respuesta, el compilador genera un set de interfaces y un set de clases que implementas esas interfases. Las interfaces generadas para el esquema `books.xsd` serían las siguientes: [Ort03]

- `CollectionType.java`. Representa el tipo complejo sin nombre del elemento `<Collection>`.
- `Collection.java`. Representa al elemento `<Collection>`.
- `BookType.java`. Representa el tipo complejo `BookType`.
- `ObjectFactory.java`. Contiene métodos para generar instancias de las interfaces.

---

<sup>31</sup> <http://www.w3.org/TR/xmlschema-0>

A continuación se listan las clases que implementan las interfaces (generadas en el subdirectorio *impl*)

- `impl/CollectionTypeImpl.java`. Implementa la interfaz `CollectionType` descrita en `CollectionType.java`.
- `impl/CollectionImpl.java`. Implementa la interfaz `Collection` descrita en `Collection.java`.
- `impl/BookTypeImpl.java`. Implementa la interfaz `BookType` descrita en `BookType.java`.

En total, las clases generadas representan todo el esquema `books.xsd`. Es importante tener en cuenta que las clases definen los métodos *get* y *set* que son usados para obtener y especificar datos respectivamente para cada tipo de elemento y atributo en el esquema. La figura 4.5 muestra gráficamente la operación.



Figura 4.5: Vinculación del esquema a través del compilador de JAXB. Obtenida de [Ort03]

Luego de generar las clases e interfaces están las dos funcionalidades que provee JAXB.

- *Unmarshalling* que significa crear un árbol de objetos de contenido que representan los contenidos y la organización del documento XML, y están directamente disponibles para el programa.

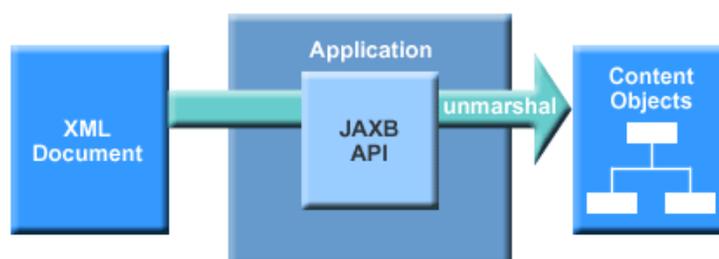


Figura 4.6: Esquema gráfico de *Unmarshalling*. Obtenida de [Ort03]

- *Marshalling* es el opuesto a *unmarshalling*, crea un documento XML desde un árbol de contenidos.

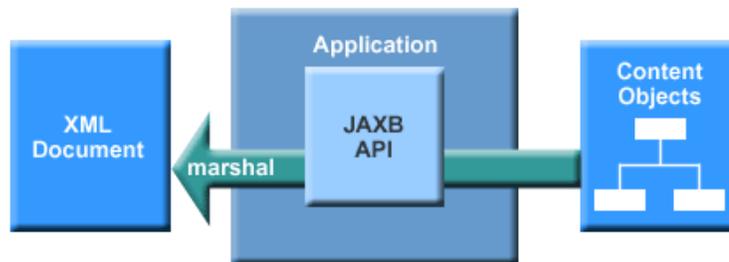


Figura 4.7: Esquema gráfico de *Marshalling*. Obtenida de [Ort03]

Los objetos de contenido son instancias de las clases generadas por el compilador. Además de proveer el compilador, JAXB provee de una API de ejecución para operaciones relacionadas como *unmarshalling* y *marshalling*, esta API es provista como parte del Marco de trabajo de vinculación. Este marco de trabajo comprende tres paquetes. El primer paquete *javax.xml.bind* contiene clases e interfaces para realizar operaciones como *unmarshalling*, *marshalling* y validaciones. Un segundo paquete *javax.xml.bind.util* que contiene clases con utilidades y por último el tercer paquete *javax.xml.bind.helper* diseñado para los desarrolladores para proveer de ayuda.

Esta tecnología supuso una gran ayuda a la hora de implementar la Interfaz de Programación de Aplicaciones API (del inglés *Application Programming Interface*) requerida por la aplicación ya que las librerías JAXB permitieron realizar un mapeo completo de los componentes del esquema QTI en clases Java. Además de generar todos los métodos que permiten la creación, carga y modificación de dichos componentes, estas librerías también facilitan la validación sintáctica en tiempo real de preguntas QTI v2.1 contra el esquema de la especificación. [Cam08]

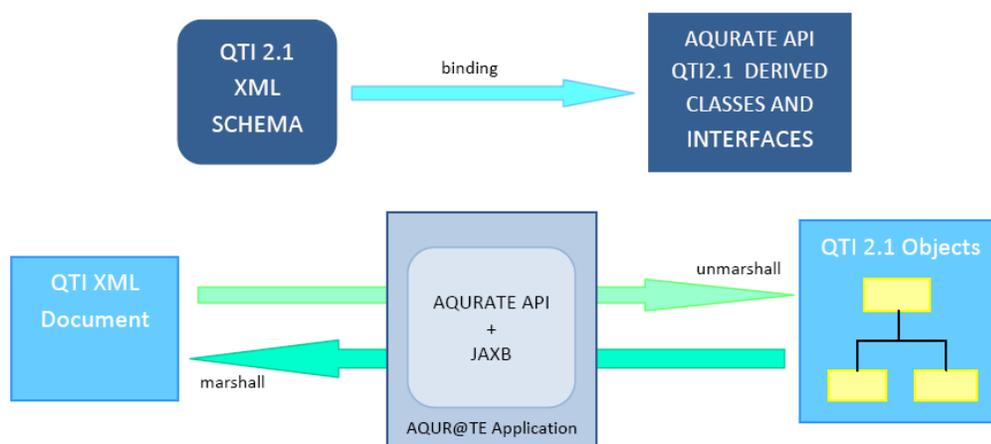


Figura 4.8: JAXB y QTI. Obtenida de [Cam08]

La estructura de las clases generadas por JAXB es de granularidad muy fina y, como consecuencia, genera un conjunto de clases demasiado grande e inmanejable para los desarrolladores, por lo que se prefiere trabajar con menos clases más complejas. Otra de las principales ventajas añadidas es la rapidez con la que se pueden re-implementar los cambios que se producen en la especificación. [Cam08]

#### **4.4.2 Arquitectura de AQURATE**

Con el fin de lograr la flexibilidad y escalabilidad especificadas en los requisitos para el desarrollo la herramienta, se diseñó una arquitectura basada en el paradigma Modelo-Vista-Controlador (MVC<sup>32</sup>). Esto asegura la independencia entre las clases Java que componen la lógica de la aplicación y las que componen la interfaz gráfica de usuario unidas por las clases controladoras.

Las clases que constituyen la API de AQURATE permiten crear y modificar los modelos de documentos XML de acuerdo con el estándar QTI v2.1. Por cada tipo de pregunta, hay definida una clase (“QTI *Document*”) que representa el correspondiente documento QTI. Estas clases son las responsables de inicializar el tipo de pregunta al que corresponda y realizar la validación semántica de los componentes QTI incluidos en dicha pregunta.

AQURATE presenta en su interfaz dos vistas hacia el usuario, una vista para interactuar con ella donde se definen todas las propiedades de algún tipo de pregunta que se esté creando o modificando y otra vista XML, en donde se muestra en formato XML QTI la pregunta con todos los componentes previamente modificados.

---

<sup>32</sup> <http://java.sun.com/blueprints/patterns/MVC-detailed.html>

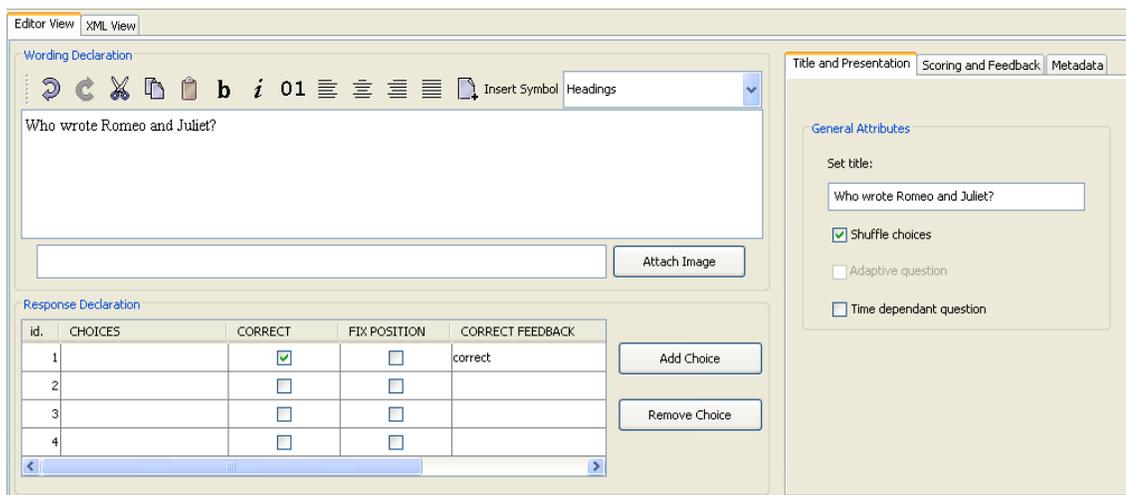


Figura 4.9: Vista de edición de una pregunta en AQRATE. Obtenida de [Cam08]

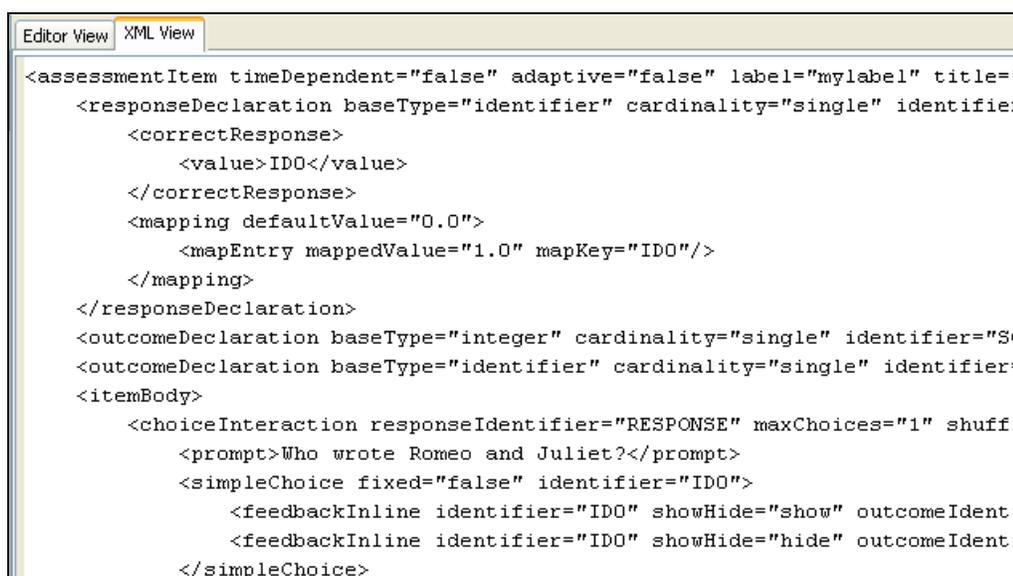


Figura 4.10: Vista XML de una pregunta en AQRATE

Con el fin de facilitar el intercambio de información entre las clases de la interfaz gráfica, la vista de edición (“*Edit View*”) y la vista XML (“*XML View*”) y las clases “*QTI Document*”, se definen las clases controladoras (“*Controller*”), existiendo una de ellas por cada tipo de pregunta. De esta forma, los cambios realizados por el usuario en el documento quedan reflejados simultáneamente en ambas vistas de la interfaz y viceversa. La figura 4.11 muestra el diagrama de la arquitectura de AQRATE bajo el enfoque de Modelo-Vista-Controlador en donde se muestra también la interacción con la API de AQRATE para utilizar las clases generadas por JAXB. Además se hace referencia a los archivos HTML generados por el *renderer* (detallado en sección 4.4.4) de AQRATE que finalmente son presentados en un Explorador Web.

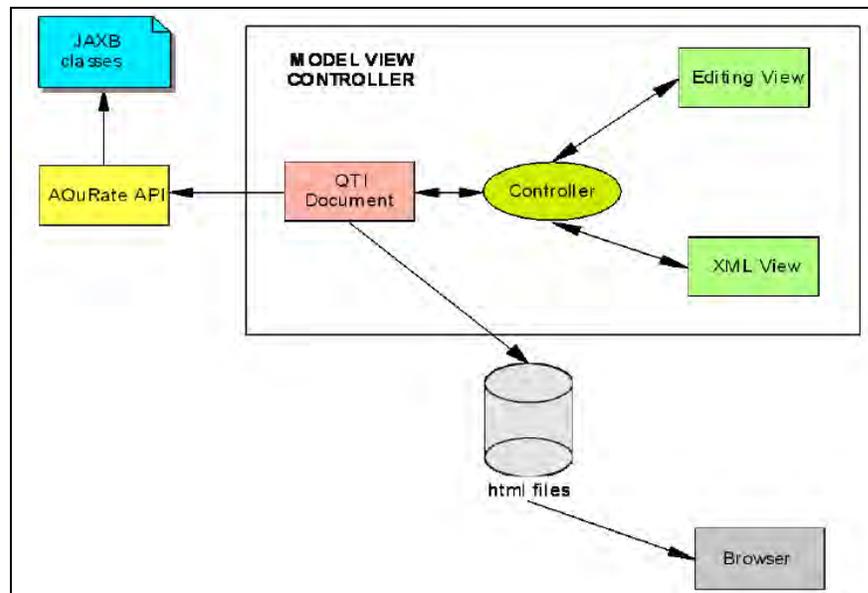


Figura 4.11: Arquitectura de AQRATE. Obtenida de [Cam08]

La clase principal de AQRATE es *AqurateFramework*, ésta clase es la que da inicio a la aplicación, tomando en cuenta el modelo MVC e posible apreciar en la figura 4.12 como está organizada la aplicación en la cual se crean instancias de:

- **AqurateObjectFactory:** Como se menciona en la sección 4.4.1, es necesario el uso de JAXB por lo que se crea una instancia de la API de AQRATE, para tener disponibles las clases que han sido creadas en base al esquema de la especificación IMS QTI.
- **AbstractController:** Clase abstracta que posee los métodos para controlar las lista de Vistas y Modelos inicializados, en otras palabras, las preguntas.
- **DefaultController:** Clase que define todas las variables de las preguntas así como los métodos para cambiar los valores de las variables.
- **AbstractDocumentModel:** Clase abstracta que observa los cambios en las propiedades de los modelos.
- **AssessmentItemDoc:** Clase Abstracta que define todas los métodos *get* y *set* para cambiar las propiedades de las preguntas.
- **AbstractViewPanel:** Clase Abstracta que implementa primordialmente los paneles de Preámbulo y Postdata de las preguntas.
- **XMLViewPanel:** Clase que implementa la vista XML, así como la coordinación a través del controlador para mantener la misma información en la vista de edición y en la vista XML.

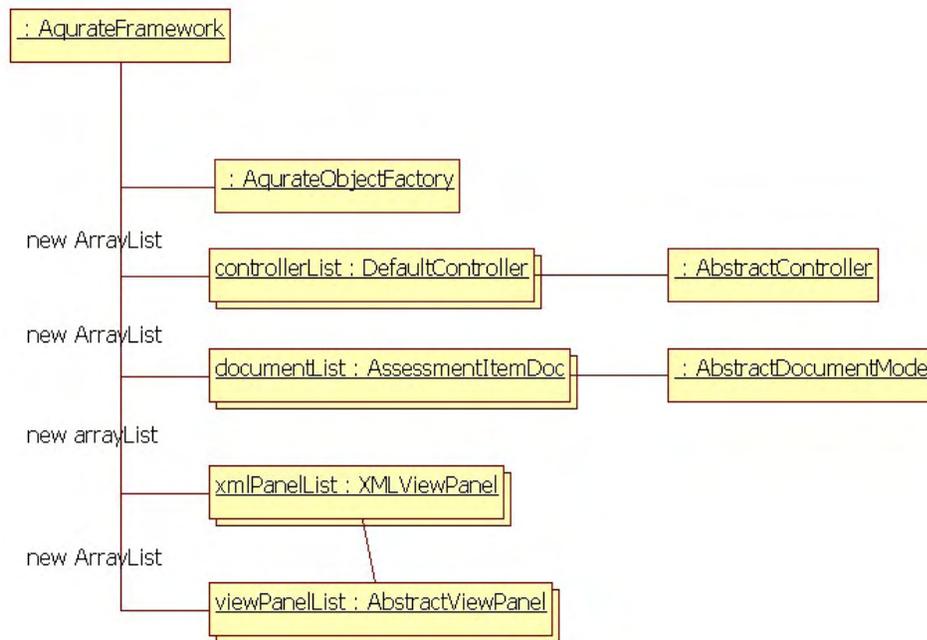


Figura 4.12: Organización de clases de las principales clases de AQRATE

Como es posible apreciar, se inicializan clases abstractas debido a que cuando el usuario elige un tipo de pregunta, no se utilizan todas las variables disponibles en las clases abstractas.

La figura 4.13 muestra el inicio de la aplicación como un diagrama de secuencia, luego de crear instancias de las clases, la aplicación espera la acción del usuario, por ejemplo, crear una pregunta.

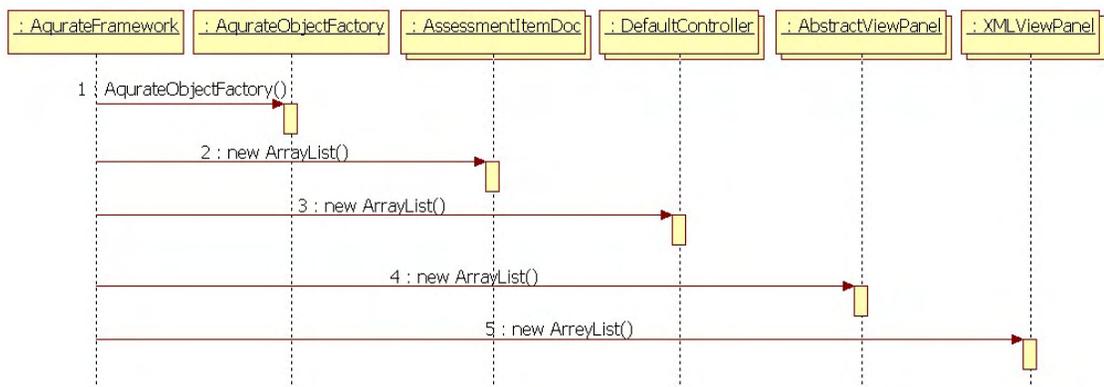


Figura 4.13: Diagrama se secuencia del Inicio de AQRATE

En este punto la aplicación está inicializada, esperando por el usuario a utilizar las funcionalidades de AQRATE que serán vistas en la sección 4.6.

### 4.4.3 La Interfaz de AQURATE

La interfaz de AQURATE, como se muestra en la figura 4.14, lista las preguntas creadas en un panel de preguntas a la izquierda (*Question pool*) de donde el usuario puede seleccionar la pregunta que desee editar, guardar o cerrar. El resto de la interfaz se utiliza para crear la pregunta, donde el usuario puede modificar los campos disponibles en cada tipo de pregunta. Estos son:

- *Add Preamble*: Es el preámbulo de la pregunta donde se puede indicar información adicional antes de presentar la pregunta.
- *Wording Declaration*: Es la pregunta que se presenta al usuario.
- *Attach Image*: Es la posibilidad de incluir imagen junto con la pregunta.
- *Response Declaration*: Son las respuestas posibles a la pregunta definida en *Wording Declaration*.
- *Add Postscript*: Es información adicional a continuación de la pregunta y las opciones que es posible incluir.
- La pestaña *title and presentation* muestra al usuario las opciones para cambiar el título de la pregunta, “*shuffle choices*” que significa reordenar las opciones al momento de ser presentadas, “*adaptive question*” o pregunta adaptativa, que significa que la pregunta puede ser reemplazada por otra dependiendo de las interacciones del usuario con la pregunta (por ejemplo, número de intentos) y “*time dependant*” que significa que la pregunta registra el tiempo que toma el usuario en responder la pregunta.
- La pestaña *score and feedback*, permite al usuario asignar puntaje mínimo y máximo a la pregunta así como el feedback al usuario luego de responder la pregunta (por ejemplo, si el usuario responde correctamente la pregunta, el sistema muestra el mensaje al usuario “correcto”).

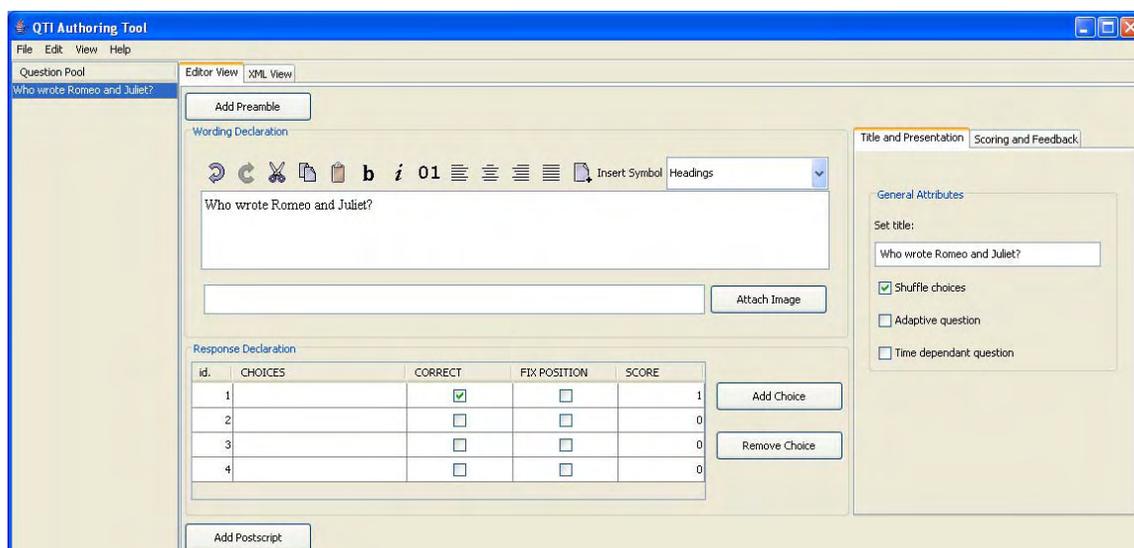


Figura 4.14: Interfaz de AQURATE. Obtenida de [AQU07b]

#### 4.4.4 El *renderer* de AQURATE

*Renderer* en español significa entregar, en este sentido para AQURATE el *renderer* es la funcionalidad que entrega o despliega al usuario la pregunta como una previsualización de la misma en el explorador Web. La necesidad de tener una vista previa de las preguntas recién creadas (*Preview Question*) fue lo que llevo a crear un sistema de entrega y procesamiento de las preguntas de forma local. Esta funcionalidad fue desarrollada por el Dr. Graham Smith en el transcurso del proyecto.

Según el Dr. Graham Smith en el informe entregado a AQURATE, en el cual explica las características técnicas del desarrollo del *renderer*, es probable que el mayor uso de las preguntas creadas sea vía Web. Es por esto que el diseño del *renderer* ha sido enfocado a tener la misma presentación y comportamiento de respuestas de preguntas QTI a como se verían vía WEB. Esto permite el uso de la tecnología de exploradores de Internet y familiares usos para el cliente a través del uso de formularios HTML. Además permite la reutilización de código de la aplicación JAssess<sup>33</sup> (una implementación en Java de la especificación IMS QTI versión 2 desarrollada por el Dr. Graham Smith), en donde el comportamiento de la presentación y procesamiento de las preguntas es idéntico.

El componente *rendering* consiste en una aplicación escrita en Java que permite la visualización de contenido QTI XML. El punto de entrada a ésta aplicación es una clase que maneja el contenido en XML y retorna un archivo en HTML que es almacenado en un directorio de AQURATE. En este directorio también son copiados archivos necesarios para la presentación y procesamiento de la pregunta como imágenes,

<sup>33</sup> <http://qtitools.caret.cam.ac.uk/>

*applets*<sup>34</sup> (mini-aplicación en Java que puede ser incluida en contenido HTML e interactuar con el usuario) y JavaScript<sup>35</sup>.

El funcionamiento del *renderer* se muestra en la figura 4.15:

- EL usuario luego de crear una pregunta selecciona la opción “previsualizar”.
- AQURATE tiene en la vista XML la pregunta seleccionada por lo tanto envía al *renderer* la pregunta empaquetada bajo la especificación IMS *Content Packaging* como un archivo en formato ZIP.
- El *renderer* procesa el archivo ZIP y desempaqueta el archivo XML QTI y lo procesa, generando la pregunta en HTML.
- La pregunta generada en HTML se presenta en un Explorador (*Browser*) utilizando *applets* para visualizarla y espera que el usuario interactúe con ella.
- El usuario visualiza la pregunta y la responde.
- Utilizando funciones JavaScript la respuesta ingresada se envía al *renderer*.
- El *renderer* procesa la pregunta y entrega la respuesta al explorador que el usuario puede visualizar.

Los archivos generados por el *renderer* son almacenados en una carpeta dentro de la estructura de directorios de la aplicación llamada “*PackageDirectory*”. Por cada una de las veces que se previsualiza una pregunta se sobrescribe la estructura de directorios. Esta estructura se describe a continuación y se puede observar en la figura 4.16.

El directorio zip contiene archivos temporales que se generan en tiempo de ejecución mediante AQURATE.

- `preview.zip`: Es el archivo almacenado bajo la especificación IMS *Content Packaging* que contiene la pregunta en formato XML QTI, imágenes si fuera necesario y el archivo `manifest.xml` que indica la organización de los archivos dentro del paquete.
- `imsmanifest.xml`: Es el archivo manifiesto del paquete.

El directorio XML contiene la pregunta en formato XML QTI.

El Directorio htm contiene todo lo necesario para visualizar la pregunta así como su procesamiento.

---

<sup>34</sup> <http://java.sun.com/applets/>

<sup>35</sup> [http://developer.mozilla.org/en/About\\_JavaScript](http://developer.mozilla.org/en/About_JavaScript)

- El subdirectorio *images* contiene las imágenes correspondientes a los logos de AQURATE junto con algunas imágenes necesarias por algunas interacciones de AQURATE.
- El subdirectorio *Jscript* contiene el archivo `QTIV2JSCRIPT.js` que es la unión entre las respuestas del usuario y el *renderer*. Cuando el usuario selecciona una respuesta, una función de este archivo recoge los datos y los envía al *renderer*. El *renderer* retorna valores y la misma función despliega la respuesta en el explorador.
- `preview.htm`: es la pregunta generada por el *renderer* para ser presentada al usuario.
- `Response_0.html`: es un archivo generado por el *renderer* luego de que el usuario selecciona una respuesta al interactuar con `preview.html`. En este archivo se muestra la misma pregunta con la respuesta que dio el usuario junto con el procesamiento de la misma, correcto o incorrecto, puntaje, feedback, etc.
- `DeskTopResponder`: es un *applet* del *renderer* necesario para interactuar con la pregunta así como para el procesamiento de la misma.
- Archivos *.class*: son los diferentes *applets* necesarios para que el usuario pueda visualizar las preguntas en el explorador, por ejemplo, *orderinteraction*, donde el usuario debe usar el *mouse* para ordenar las opciones que se le presentan para responder la pregunta.
- Archivos *.css*: son archivos de hojas de estilo para el formato de la pregunta, como color de fondo, tipo de letra, espaciamiento, etc.

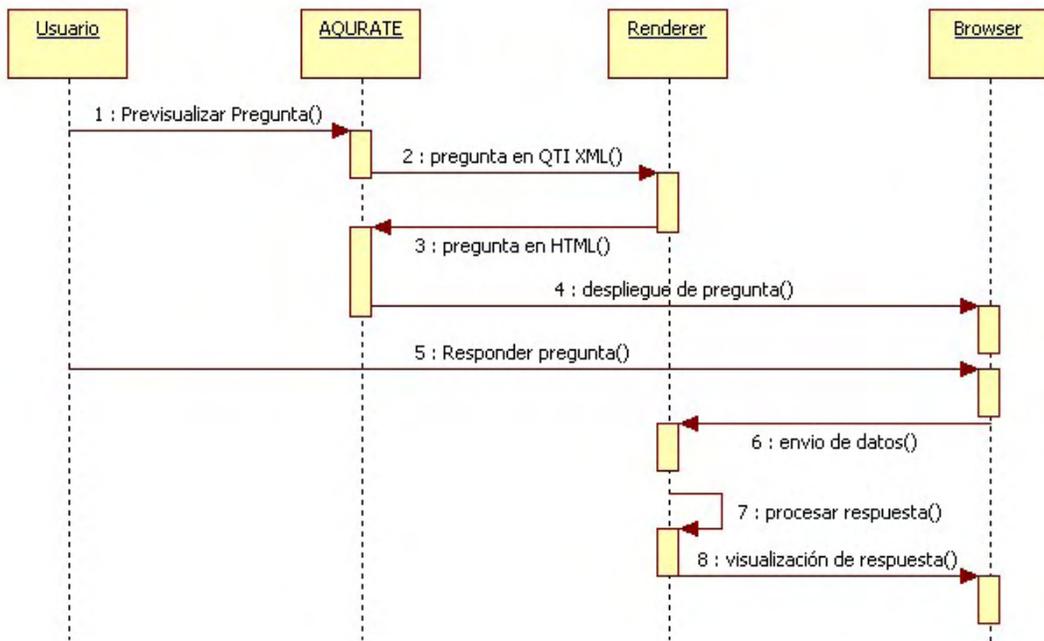


Figura 4.15: Diagrama de Secuencia del funcionamiento de Aqurate y el *renderer*.

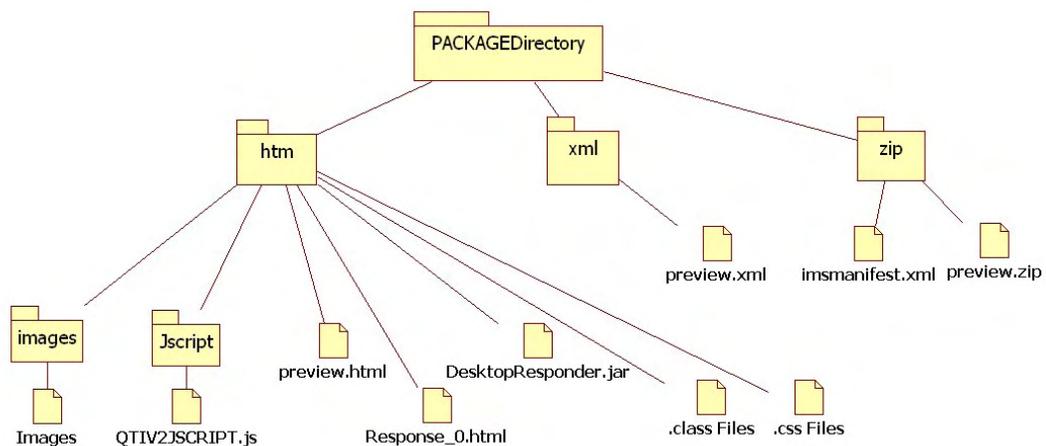


Figura 4.16: Organización de archivos generados por el *renderer*.

## 4.5 Funcionalidades Principales de AQURATE

El conjunto de funcionalidades de la última versión disponible de AQURATE en [AQU07b] se detalla en la figura 4.17. Las funciones a detallar en este trabajo de tesis, que se consideran las más importantes son, Crear (*New*) y Guardar (*Save*), la funcionalidad de Vista Previa (*Preview*) ya fue detallada en la sección 4.4.4.

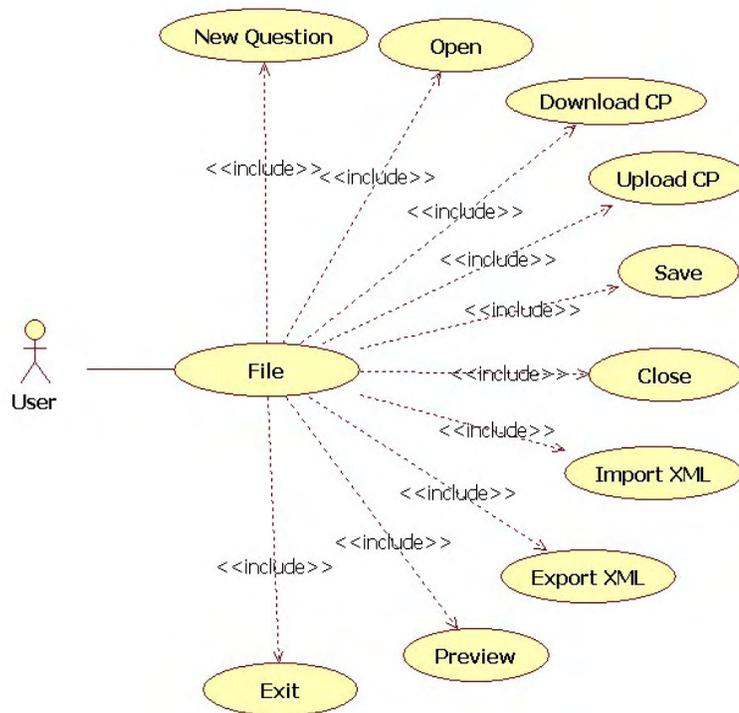


Figura 4.17: Funcionalidades de AQRATE. Obtenida de [AQU07b]

### 4.5.1 Creación de Preguntas

Aunque la especificación IMS QTI versión 2.1 define 21 tipos de interacciones [AQU07a], AQRATE dividió este set de preguntas en ocho principales para ser desarrolladas en la fase uno del proyecto, para dedicar tiempo al desarrollo de las posibles interacciones así como otros aspectos de la implementación del sistema. Las ocho interacciones principales se muestran en la figura 4.18, que es la pantalla de selección del tipo de pregunta a crear en AQRATE.

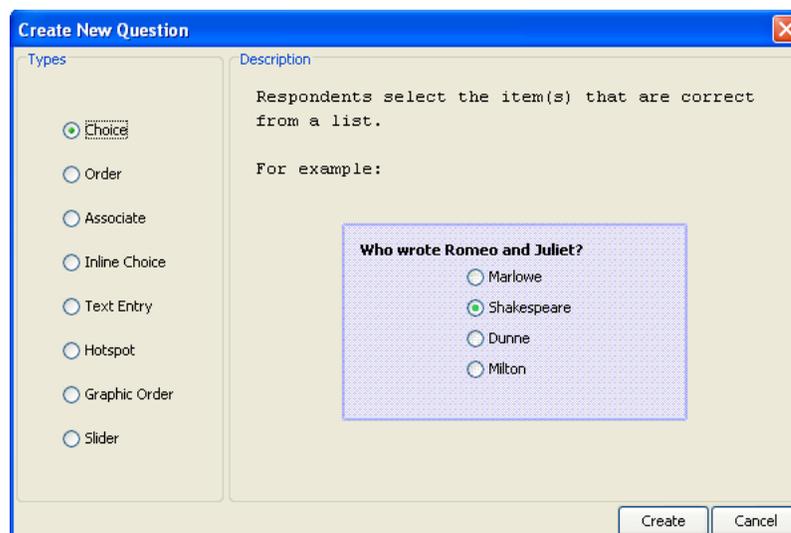


Figura 4.18: Tipos de pregunta disponibles en AQRATE para su creación. Obtenida de [AQU07b]

Como se detalló en la sección 4.4.2 en la arquitectura de AQURATE se utilizan distintas clases para inicializar la pregunta, siendo las principales las con relación al controlador, las vistas y el modelo del documento QTI. En la figura 4.19 se muestra el diagrama de secuencia de la creación de una pregunta donde se tiene como ejemplo el tipo de pregunta *choiceinteraction*, pero el proceso es el mismo para cualquier pregunta. En este punto la pregunta está inicializada y se despliega en pantalla esperando a que el usuario modifique las propiedades de la pregunta detalladas en la sección 4.4.3.

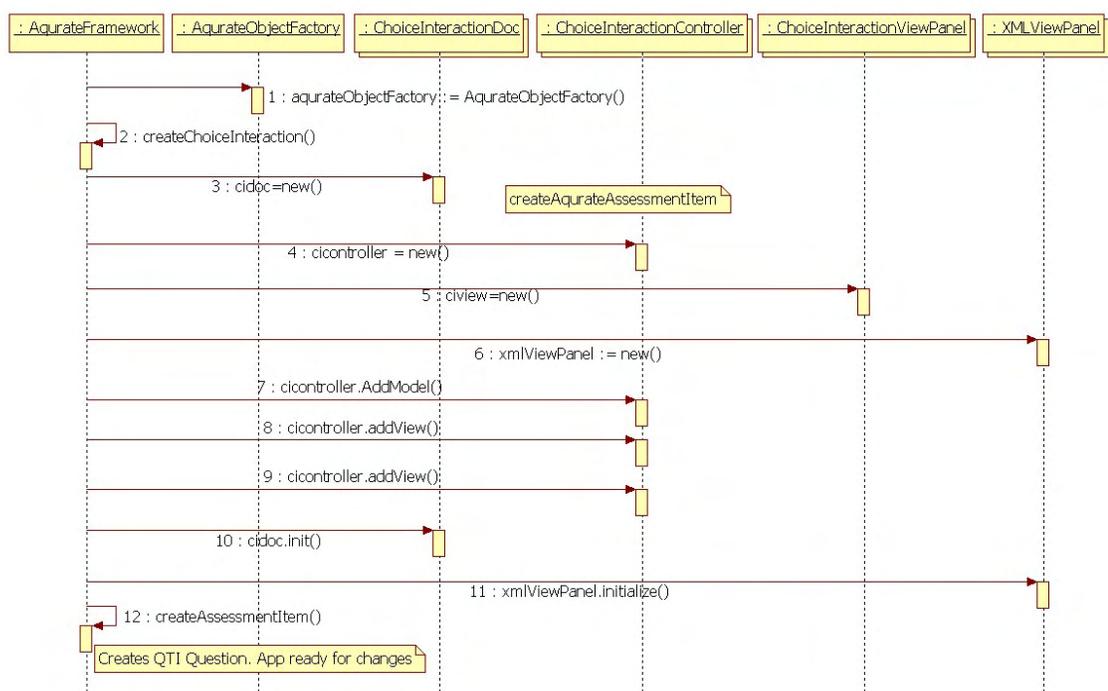


Figura 4.19: Diagrama de Secuencia de la creación de una pregunta.

## 4.5.2 Guardar Preguntas

Luego de crear una pregunta, lo más probable es que el usuario desee guardarla para luego seguir trabajando con la misma o llevarla a otro sistema. El archivo que genera es un archivo en formato ZIP llamado *Content Package* (paquete de contenidos, de ahora en adelante CP, El proceso que realiza AQURATE para guardar una pregunta se muestra en la figura 4.20 en donde:

- Se recibe la instrucción para guardar una pregunta.
- Se instancia un archivo en formato ZIP en el cual se empaquetara la pregunta.

- Se revisa la pregunta (que está en la vista XML) en busca de las etiquetas <img> ó <object>, imagen u objeto respectivamente.
- Si se encuentran imágenes u objetos se agregan al archivo ZIP, además de la pregunta en formato XML.
- Se crea el archivo manifiesto (*manifest*) al cual se le agrega información acerca del CP que se está creando, metadatos e información del creador de la pregunta. Además se asocian los recursos (imágenes u objetos) con la pregunta, en otras palabras la organización dentro del paquete.
- Se añade el archivo manifiesto al archivo ZIP y se almacena.

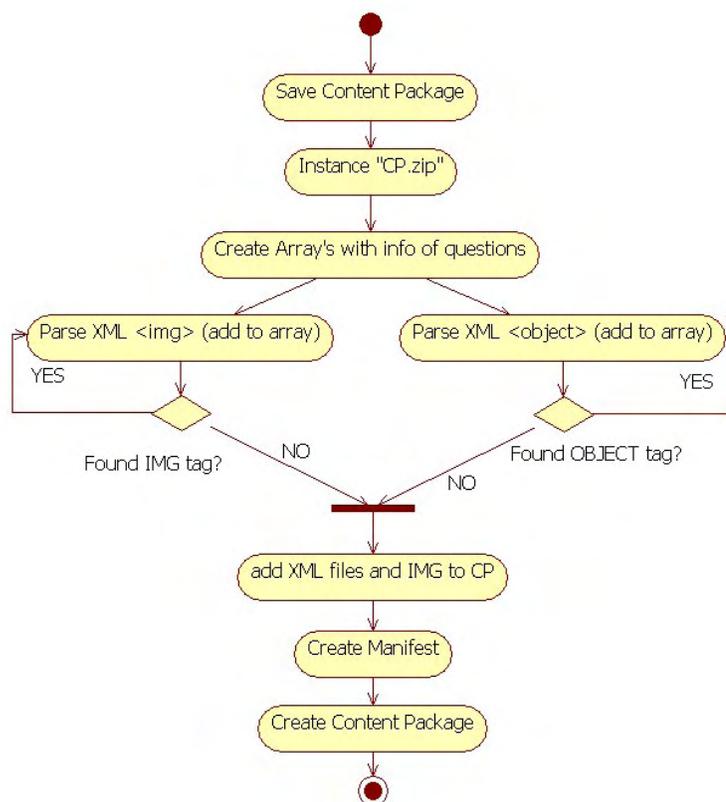


Figura 4.20: Diagrama de Actividad de la creación de un CP.

Luego de crear el CP se vería tal como en la figura 4.21 en donde:

- Cerro.jpg es una imagen que se mostrará junto a la pregunta.
- CerrosdeChile.xml es la pregunta que se le hace al usuario
- Imsmanifest.xml es el archivo manifiesto que indica la información acerca del CP así como de su organización de los recursos dentro del paquete.

Nombre	Tamaño	Comprimido	Tipo
..			Carpeta
cerro.jpg	3.855	3.829	Imagen JPEG
CerrosdeChile.xml	3.037	841	XML Document
imsmanifest.xml	2.628	750	XML Document

Figura 4.21: Contenido del CP creado.

## 4.6 Limitaciones de AQURATE

La sección 4.2 que se refiere a la descripción del proyecto AQURATE, lista los principales objetivos y en ellos se menciona la creación de preguntas bajo la especificación IMS QTI, pero no implica que puedan guardarse como un conjunto de preguntas. En AQURATE para guardar un grupo de preguntas sólo es posible hacerlo individualmente. Esto supone un mayor esfuerzo del usuario para guardar las preguntas que ha creado, así como para editarlas. Esto implica a su vez que la previsualización de preguntas se hace de forma individual.

Extender estas funcionalidades a que sea posible guardar un grupo de preguntas, así como previsualizarlas da una mayor usabilidad de la herramienta, debido a que el usuario podrá guardar una colección de preguntas sin necesidad de tener varios archivos para realizar un examen, sino que uno solamente, un *Content Package* en el cual el archivo manifiesto maneja toda la organización de las preguntas, así como los recursos que utilizan.

Esto da paso a que sea posible previsualizar un conjunto de preguntas, y es ésta es la funcionalidad que se detalla en el siguiente capítulo.

## 5. LA FUNCIONALIDAD QUIZ

### 5.1 Introducción

En un principio, AQURATE fue diseñada para generar y trabajar con varias preguntas QTI simultáneamente, de manera que, el autor es capaz de moverse entre ellas, referenciar, o cortar y pegar material de unas a otras. No obstante, esta capacidad de manipular varias preguntas a la vez (una colección de elementos), presentaba ciertas consideraciones a tener en cuenta en la edición de un examen así como en la edición de preguntas individuales.

Durante los dos meses de trabajo en la Universidad de Kingston se tuvo la oportunidad de trabajar junto a todo el equipo de trabajo de AQURATE quienes presentaron la inquietud de agregar una funcionalidad a la herramienta. Esta funcionalidad era la de crear una serie de preguntas y visualizarlas como un examen, debido a que las herramientas que supuestamente deberían funcionar como ASDEL no habían sido finalizadas, y no se habían realizado mayores pruebas en los sistemas.

Se propuso la creación de una funcionalidad llamada QUIZ que daría la posibilidad de crear varias preguntas y poder guardarlas en conjunto así como visualizarlas como una secuencia lineal de preguntas. La razón por la cual se llamó QUIZ, es para evitar confusiones con las palabras como *Assessment* ó *Test* debido a que la funcionalidad no significa que se pueda crear un examen en base a la especificación IMS QTI ya que no se trabaja con el elemento “*Assessment*” de la especificación, sino con un conjunto de preguntas o “*Assessment Item*”, que es permitido por la especificación ya que es posible tener en un CP varias preguntas en donde el archivo manifiesto maneja la organización de todas las preguntas y recursos asociados a la mismas. En IMS QTI, el elemento *Assessment* tiene una serie de propiedades al igual que las preguntas, y supuestamente fueron trabajadas por ASDEL, por lo que si se trabajaba como un examen implicaría mucho tiempo en investigación que desarrollo.

La funcionalidad no considera la evaluación del usuario con notas en una escala, sino que informa al usuario si su respuesta fue correcta o incorrecta, ya que da un mayor valor al usuario saber el resultado de cada respuesta que una nota final del QUIZ, tomando en cuenta que la funcionalidad no trabaja con el elemento *Assessment* de la especificación.

## 5.2 Desarrollo de la Funcionalidad

El desarrollo de la funcionalidad implicó la investigación de toda la herramienta AQURATE y su funcionamiento así como la estructura del funcionamiento del *renderer*. Es por esto que se ha dividido esta sección en los cambios que se realizaron en AQURATE así como la implementación de nuevas funcionalidades.

### 5.2.1 Modificación del *Content Package*

Debido a que AQURATE guardaba sólo una pregunta por CP, fue necesario extender su usabilidad y permitir que se guardaran varias preguntas en un solo CP, lo cual está permitido por la especificación IMS *Content Packaging* [IMS04]. Los cambios realizados permiten el funcionamiento original de AQURATE así como la nueva implementación. Esto implicó una extensión de los métodos en la clase *ContentPackageBuilder.java* que es la que recibe la pregunta y genera el CP. Las modificaciones a esta clase principalmente fueron en la adición de imágenes y asociación de las mismas con las preguntas que las referenciaban, así como la organización del archivo manifiesto. Como se muestra en la figura 5.1, el resultado de la modificación de la generación del *Content Package*, varias preguntas incluidas en un mismo paquete.



Nombre	Tamaño	Comprimido	Tipo
..			Carpeta
cerro.jpg	67.158	44.310	Imagen JPEG
imsmanifest.xml	13.745	955	XML Document
Question_0.xml	3.046	845	XML Document
Question_1.xml	2.769	692	XML Document
Question_2.xml	2.769	692	XML Document
Question_3.xml	2.769	692	XML Document
Question_4.xml	2.769	692	XML Document
Question_5.xml	2.769	692	XML Document

Figura 5.1: Varias preguntas en un CP

Estos cambios dieron paso a una nueva funcionalidad, *Save Pool* que en español significa guardar un grupo de preguntas, y que resulta muy útil para el usuario que crea el QUIZ, ya que puede guardar un grupo de preguntas y no solo preguntas individuales como se hacía anteriormente. La figura muestra la acción del usuario guardando un set de preguntas, esto genera un archivo *Content Package* tal como se muestra en la figura 5.1.

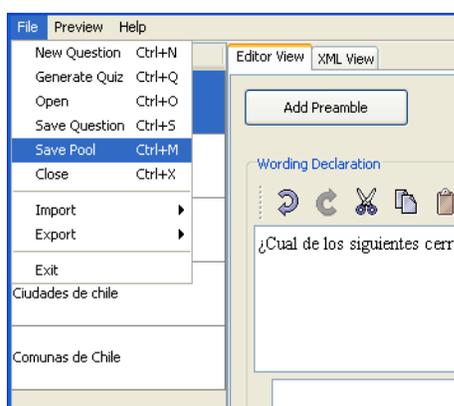


Figura 5.2: Funcionalidad *Save Pool*.

El hecho de guardar un grupo de preguntas, implicó abrir un grupo de preguntas. Para esto fue necesario modificar la clase principal `AqurateFramework.java`, específicamente el método que abre el *Content Package* `getJMenuItemOpen`. El método abría archivos XML individualmente, así como *Content Packages*, por lo que fue necesario extender su funcionalidad para abrir varios archivos contenidos dentro de un *Content Package* tal como se muestra en la figura 5.1. El método carga una a una las preguntas en grupo de preguntas del lado izquierdo de la interfaz de la aplicación.

## 5.2.2 Interfaz de la Funcionalidad

La nueva funcionalidad requirió de una nueva ventana para seleccionar las preguntas que el usuario previamente había creado. El panel creado pasó por revisiones del Dr. Martin Colbert quien estaba a cargo de la usabilidad en cuanto a interfaces de la herramienta, luego de varias pruebas sobre como implementar la interfaz para la funcionalidad sin invadir el resto de la aplicación, se decidió incluir una ventana en la cual se desplegaban las preguntas que el usuario estaba creando en ese momento y poder así elegir las preguntas que deseaba incluir en un QUIZ. Para esto se implementó la clase `JDialogNewQuiz.java` la cual contiene todos los métodos necesarios para desplegar un panel tal como se muestra en la figura 5.3, es posible seleccionar las preguntas que se quieren incluir en el test, como la primera pregunta es la que se muestra primero en el QUIZ, es posible ordenar la presentación de las mismas con los botones “*Up*” y “*Down*”, arriba y abajo respectivamente.

Las opciones de seleccionar y ordenar las preguntas fueron muy bien acogidas por el equipo de desarrollo de AQRATE ya que no se habían contemplado en el desarrollo original de la herramienta. Los botones *Preview* y *Generate* serán detallados a continuación.

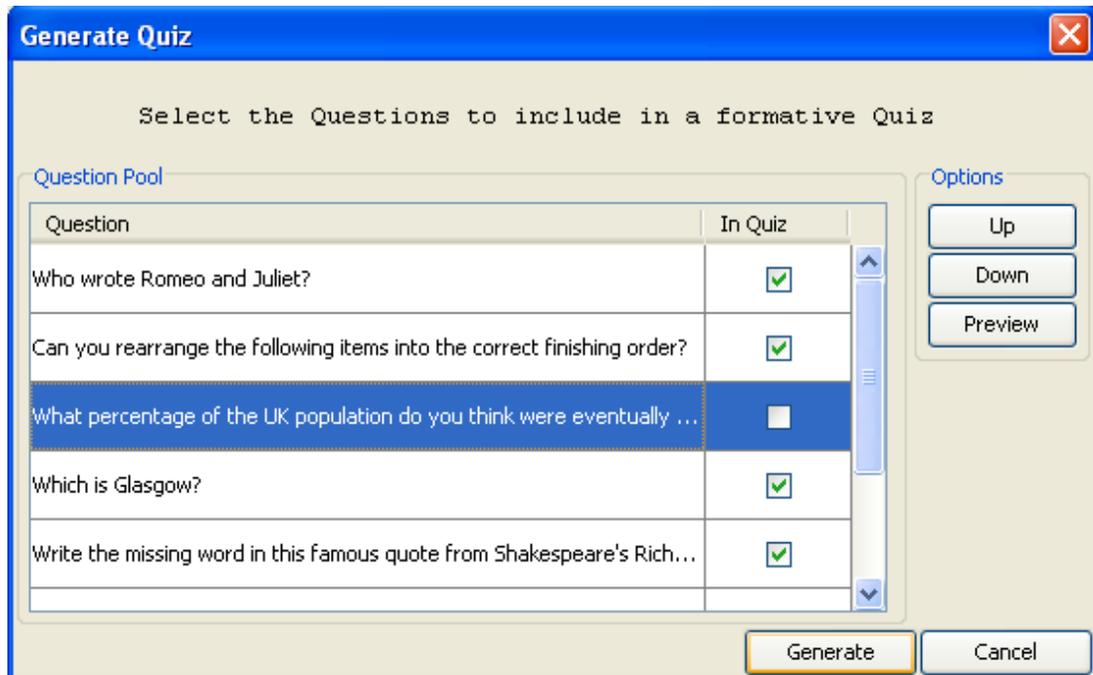


Figura 5.3: Interfaz de la funcionalidad QUIZ

### 5.2.3 Funcionalidad Preview QUIZ

Debido a que se extendieron las funcionalidades del CP a varias preguntas, en el *renderer* era necesario extenderlas también, por lo que fue necesario reestructurar la generación de los archivos para dar paso a la funcionalidad. Respecto de la figura 4.16 en la sección 4.4.4 sobre el *renderer* de AQUATE, la figura presenta cambios en la estructura agregando un nivel de jerarquía y creando carpetas por cada una de las previsualizaciones que se realizan, ya sea por preguntas individuales o un grupo de preguntas. Los cambios necesarios son mostrados en la figura 5.4, en donde se tienen:

- `TEST_{ZIPNUMBER}`: Es el directorio que contiene a toda la pregunta o QUIZ. `ZIPNUMBER` es un número temporal que se genera en tiempo de ejecución por el *renderer*.
- `TEST_{ZIPNUMBER}.html`: Es la página principal que llama al resto de las preguntas para secuenciarlas, en el caso que sea sólo una pregunta, se despliega sólo esa pregunta.
- `Question_{Number}` (`.xml` ó `.html`): Sol las preguntas en formato XML y HTML respectivamente. *Number* es el correlativo que le corresponde a cada pregunta creada.

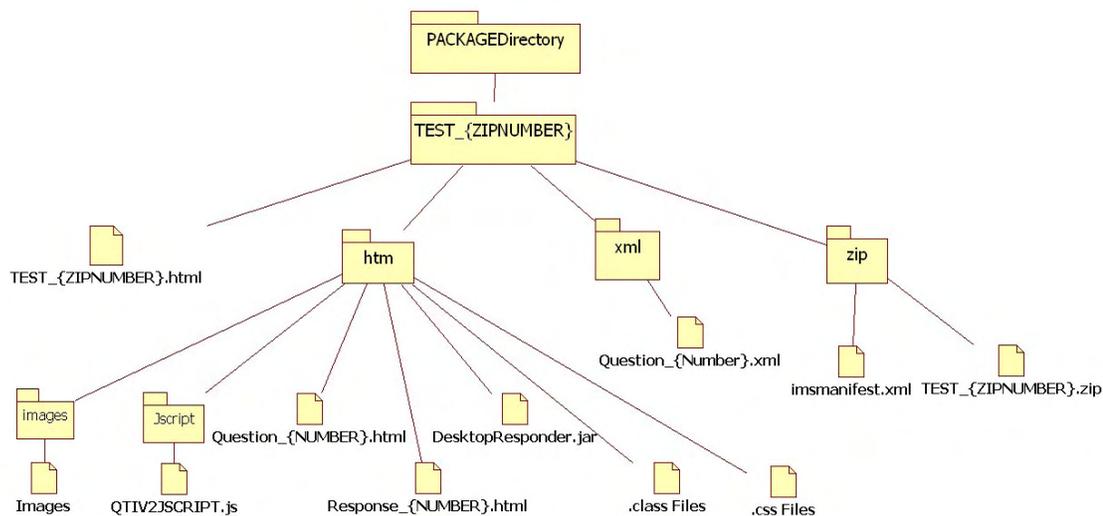


Figura 5.4: Nueva organización de archivos generados por el *renderer*.

Esta nueva organización de contenidos dio paso a que las funcionalidad *preview QUIZ* quedaran independientes unas de otras y no confundir las preguntas con otras hechas anteriormente. En la figura 5.3, se muestra el botón *preview* en la interfaz, que significa previsualizar las preguntas seleccionadas en el explorador como una secuencia lineal de preguntas. Esta funcionalidad requirió la mayoría del trabajo, ya que es necesario considerar varias modificaciones al código de AQRATE así como nuevas implementaciones. Para llevar a cabo esta implementación se consideran cuatro puntos principales:

- La conversión de las preguntas en formato XML QTI a HTML.
- La presentación de las preguntas en el explorador, lo que implica una página principal que vaya secuenciando las preguntas y poder recorrerlas.
- La recolección de las respuestas indicadas por el usuario y su envío a la aplicación para su posterior proceso.
- El proceso de las preguntas y su presentación nuevamente al usuario.

Extendiendo el diagrama de la figura 4.15 en la sección 4.4.4, es posible contemplar en la figura 5.5 la interacción de las tres principales clases de generación de contenidos y respuestas, así como de la comunicación del sistema con el Explorador por medio de JavaScript, éstas se detallan a continuación:

- `ItemBodyHandler.java`: La instrucción 5 de la figura 5.5, genera el contenido HTML presente en todos los tipos de preguntas, utilizando además clases específicas dependiendo del tipo de interacción.
- `QTIPkgeHandler.java`: La instrucción 6 de la figura 5.5 genera el contenido HTML para la página principal, que llama al resto de las preguntas para secuenciarlas y presentarlas en una sola pagina.
- `QTIV2JSCRIPT.js`: La instrucción 9 de la figura 5.5, indica la ejecución de un archivo escrito en JavaScript que recoge las respuestas indicadas por el usuario en los formularios HTML creados por el *renderer*. Luego de tener las respuestas del usuario las envía a la aplicación, específicamente a `QTIV2Responder` donde son procesadas las respuestas y se genera la página HTML de respuesta.
- `QTIV2Responder.java`: La instrucción 10 de la figura 5.5, indica que luego de que las preguntas han sido presentadas, es el momento de interacción con el usuario. Esta clase maneja las respuestas del usuario, procesándolas y presentando al usuario si su pregunta fue correcta o incorrecta.

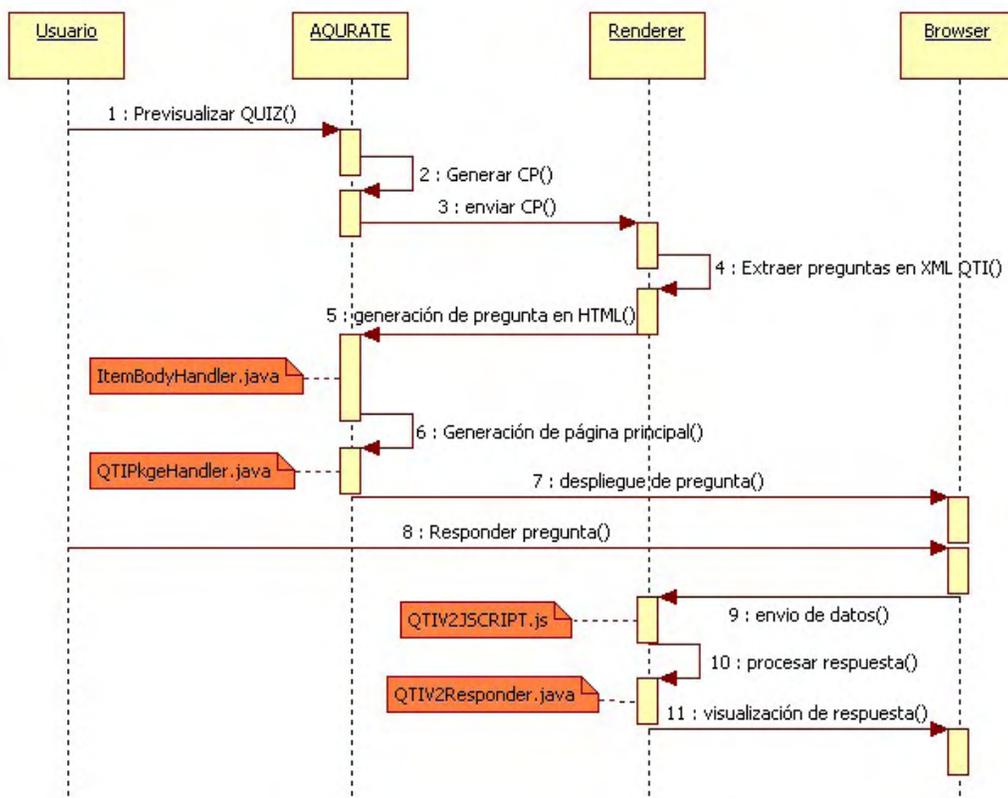


Figura 5.5: Diagrama de secuencia extendido de la funcionalidad Preview QUIZ.

La funcionalidad *Preview QUIZ*, como prototipo de salida, pretendía una página Web dividida en secciones, tal como se muestra en la figura 5.6. El lado izquierdo muestra las preguntas disponibles que el usuario ha seleccionado para el QUIZ y se puede navegar entre ellas con los botones *Previous* (Anterior), *Answer* (Respuesta) y *Next* (Siguiente) para contestar las preguntas.

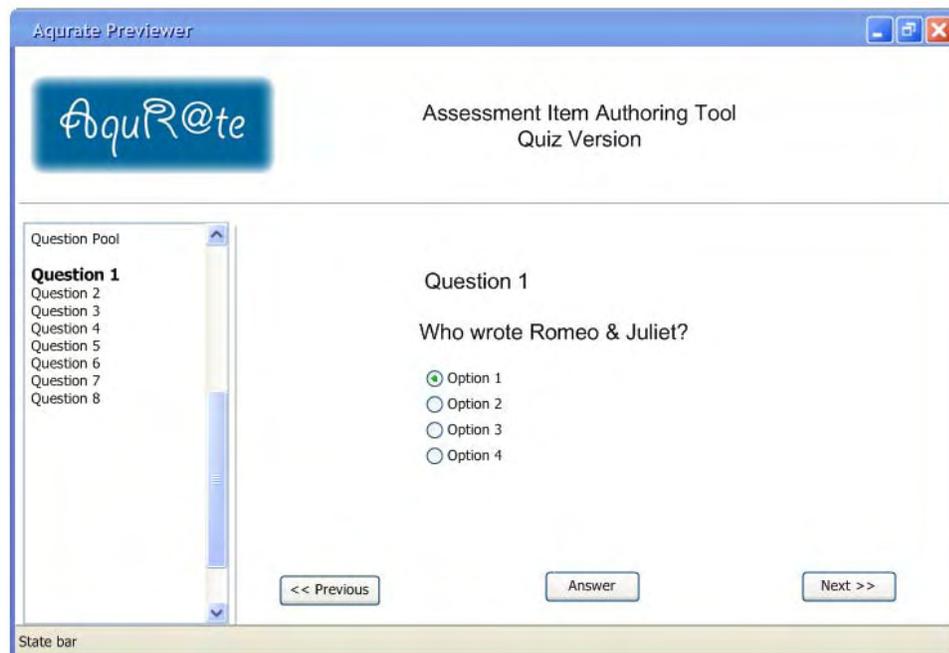


Figura 5.6: Prototipo de la página Web generada por la funcionalidad QUIZ

La navegación entre las preguntas fue programada en el archivo `QTIV2JSCRIPT.js` al cual se le agregaron los métodos para identificar las preguntas que el usuario respondía y la recolección de las respuestas tal como se especificó anteriormente.

Luego de que AQURATE realiza el proceso de previsualizar el QUIZ, la salida se ve tal como se muestra en la figura 5.7, no como un prototipo, sino como la página Web generada por AQURATE.

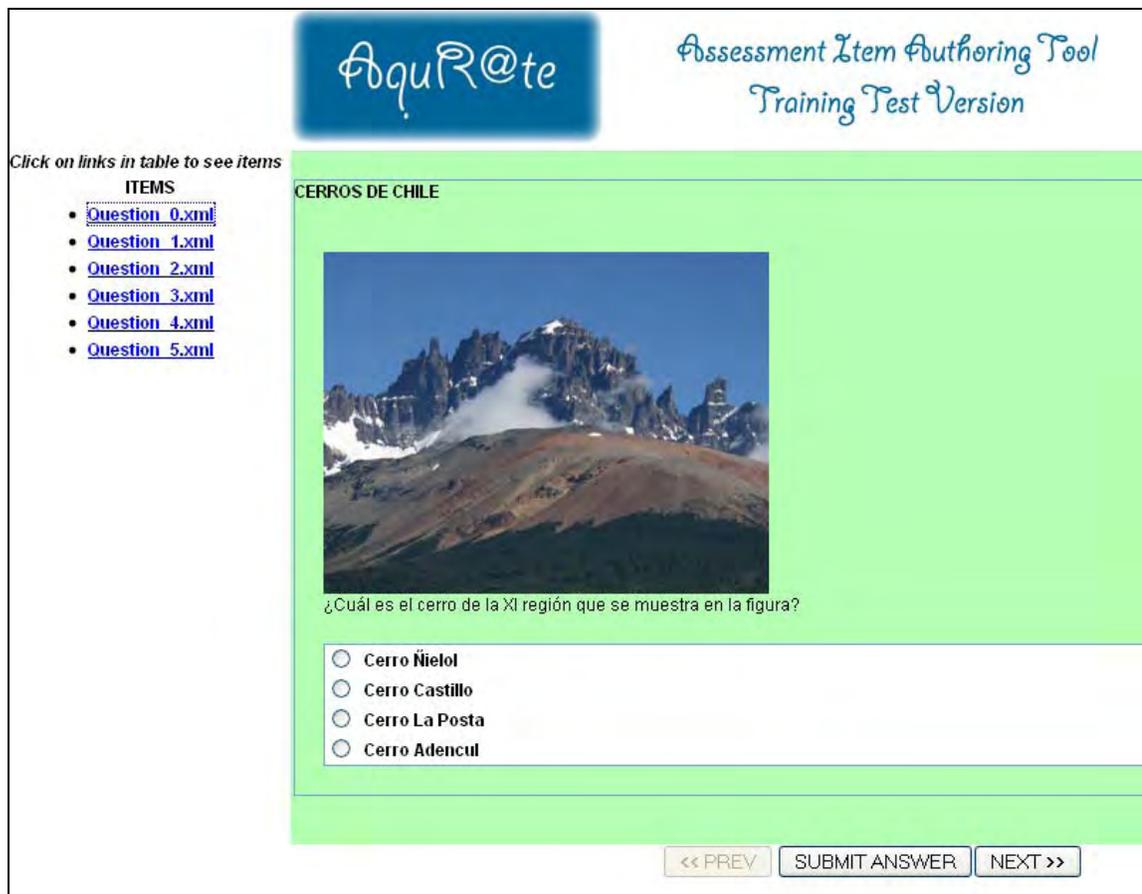


Figura 5.7: Vista en el explorador de la funcionalidad QUIZ.

## 5.2.4 Funcionalidad Generate QUIZ

El botón *Generate* en la figura 5.3 significa crear un archivo con formato ZIP e incluir en el todos los archivos necesarios para el funcionamiento del QUIZ. Esto significa que es posible realizar el QUIZ en cualquier plataforma debido a que sólo es contenido HTML el que verá el usuario, junto con *applets* en JAVA, tal como se muestra en la figura 5.7. Todos estos contenidos se empaquetan en un archivo ZIP, el cual da la posibilidad de ser distribuido por diversos medios. La figura 5.8 intenta mostrar gráficamente los contenidos que serán incluidos en el archivo ZIP, que son los mismos que la figura 5.4 comenzando por la carpeta `TEST_{ZIPNUMBER}`.

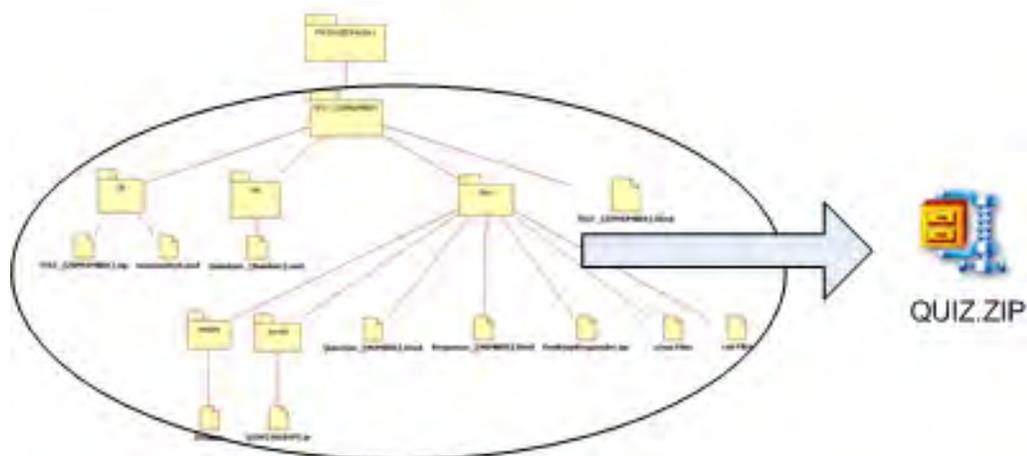


Figura 5.8: Generación del QUIZ en un archivo formato ZIP.

La inclusión en AQURATE de esta funcionalidad proporciona la oportunidad de generar recursos autosuficientes, ya que el paquete creado contiene las preguntas en formato XML, sus correspondientes archivos HTML y toda la lógica necesaria para procesar las respuestas. Los HTML se cargan en un navegador sin necesidad de poseer conexión con Internet, de forma que los estudiantes pueden avanzar y retroceder en la secuencia de cuestiones al mismo tiempo que las responden.

Durante su ejercicio no necesariamente se habla de puntuación, ya que el estudiante no recibe una nota, si no que cada respuesta es evaluada como “correcta” o “incorrecta”. Esto puede entenderse como un modo de entrenamiento o ejercicio utilizado, por ejemplo, para la evaluación formativa en la educación superior, o ser considerado como el material complementario de los libros de texto.

Este caso de uso, es útil en situaciones donde no se puede lograr o garantizar al usuario la conexión con la red. También posee la ventaja de proporcionar material cuya presentación e interacción han sido previamente comprobadas, reduciendo así la incertidumbre de cómo será presentado finalmente al estudiante. [Cam08]

### 5.2.5 Modificación del Menú de AQURATE

Aparte de trabajar con la funcionalidad QUIZ, se trabajó con el menú de AQURATE. Se propuso una reestructuración del menú organizándolo de distinta forma para disminuir su tamaño y no confundir al usuario.

Entre las modificaciones principales al menú de destacan la inclusión de las funcionalidades *Generate QUIZ* y *Save Pool*. El uso de estos nombres implicó pasar de un menú antiguo llamado *New* a ser llamado *New Question*, así evitar que el usuario piense que puede crear un QUIZ sino que dirigir al usuario a crear nuevas preguntas, y al

momento de tener más de dos preguntas creadas, activar la opción *Generate QUIZ*. Lo mismo ocurre con la antigua opción llamada *Save*, la cual pasó a llamarse *Save Question* debido a que el usuario podría pensar que *Save* significa guardar todas las preguntas que haya creado. Así se tiene *Save Question* y *Save Pool* para guardar una pregunta y toda la lista respectivamente.

Los cambios propuestos fueron aceptados por el equipo de AQRATE y se presenta a continuación en la figura 5.9.

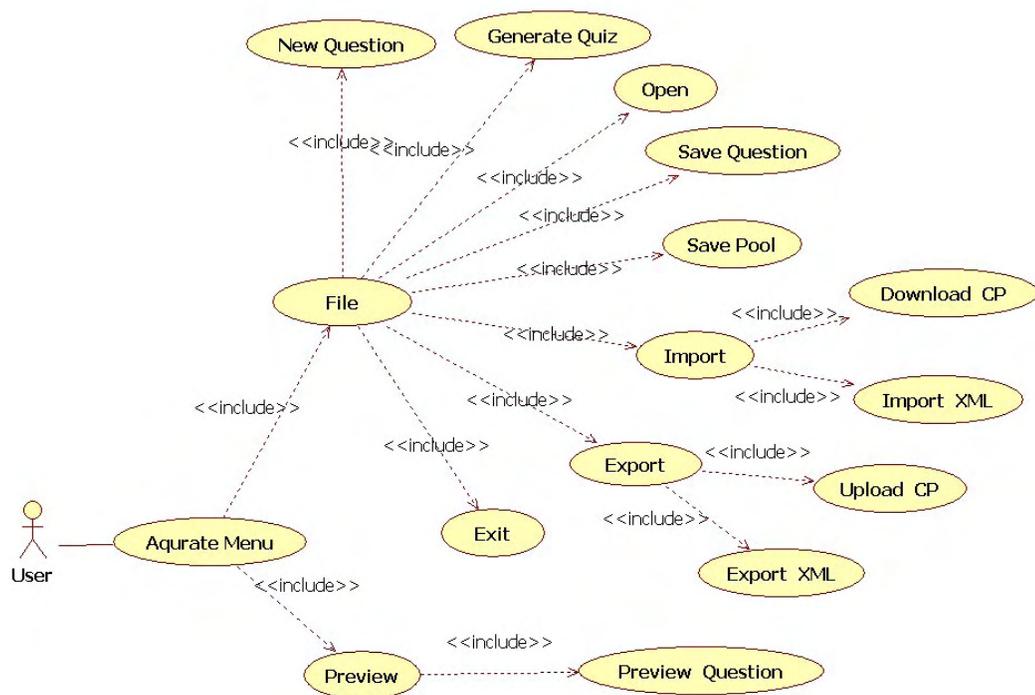


Figura 5.9: Nuevo menú de AQRATE.

### 5.3 Errores Encontrados.

Durante el estudio de AQRATE y del desarrollo de la funcionalidad se fueron encontrando errores en la aplicación, si bien algunos eran menores y fueron solucionados, para otros se requería mayor trabajo ya que tipos de preguntas como *InlineChoice* y *TextEntry* no estaban implementadas por completo, y que requerían además modificaciones a la API de AQRATE. Los errores incluían problemas con la generación del archivo manifiesto en inclusión de metadata, como autor y lenguaje. Por esta razón se hizo un estudio de todas las interacciones con las que trabaja AQRATE, que son ocho, como se menciona en la sección 4.5.1 respecto a la funcionalidad de creación de preguntas. De este estudio se llegó a la conclusión que de las ocho interacciones disponibles, sólo cuatro funcionaban totalmente debido a errores

ya sea en AQURATE como en el *renderer*. Esto se refiere a Abrir, Guardar y Previsualizar preguntas principalmente. Las interacciones con las que se trabajó en la funcionalidad fueron *choiceinteraction*, *orderinteracion*, *hotspotinteraction* y *sliderinteraction*.

El *renderer*, en el cual se trabajó en conjunto con el Dr. Graham Smith quien proveyó las últimas versiones del código, también contenía errores, principalmente para el funcionamiento de los *applets* y la generación de archivos HTML. Lo que imposibilitó, como se menciona anteriormente, el uso de algunas interacciones con la funcionalidad.

Todos estos errores fueron reportados y documentados al equipo de desarrollo de AQURATE. Se produjo un documento con el estudio realizado a las interacciones así como una lista de errores, el cual está disponible en el Anexo Digital B.

## 5.4 Actualización del Código

El equipo de desarrollo de AQURATE trabajó utilizando el servicio de repositorios de código de Google<sup>36</sup>, donde se manejaban las últimas versiones de código y documentación sobre la herramienta. En febrero del año 2008 todo el código fue transportado a otro repositorio llamado *Sourceforge*<sup>37</sup> de donde se obtuvo todo el código para trabajar la herramienta. El código puede ser descargado del repositorio, pero es necesario ser parte del equipo de desarrollo del proyecto para poder actualizarlo con nuevos cambios, es por esto que se concedieron permisos a través de una cuenta generada por el director del proyecto Prof. Graham Alsop. Los cambios efectuados al código de AQURATE fueron documentados e incluidos en el repositorio de documentos de Google del proyecto al cual es posible acceder también con los permisos necesarios para la publicación de documentos. El primer día de agosto del año 2008 se actualizó el repositorio con todos los cambios hechos incluyendo la documentación. El código fuente está disponible en el Anexo Digital A y la documentación en el Anexo Digital B.

---

<sup>36</sup> <http://code.google.com/p/qtidevelopment/>

<sup>37</sup> <http://www.sourceforge.net/>

## 5.5 Herramientas Utilizadas

Las herramientas utilizadas para el desarrollo de AQRATE fue el entorno de trabajo de código abierto Eclipse<sup>38</sup>, con el cual se manejaban los proyectos y las librerías. Se utilizó *Subversion*<sup>39</sup> para el manejo de versiones y poder así obtener y actualizar el código desde el repositorio en *Sourceforge*. Para la documentación se utilizaron herramientas comunes como Microsoft Word, Microsoft Power Point y Microsoft Visio. Para los diagramas vistos anteriormente en las figuras se utilizó *StarUML*<sup>40</sup> que es un software de código abierto para la generación de documentos UML. Para las previsualizaciones se utilizó Microsoft Internet Explorer y Mozilla Firefox, dos de los exploradores Web más utilizados en el mundo. Es importante destacar que para que el entorno de trabajo con Eclipse funcionara, era necesario tener instalado el *kit* de desarrollo de Java en su versión 1,5. Está disponible en el Anexo Digital C un manual para la instalación del entorno de desarrollo así como la descarga del código fuente y las librerías.

---

<sup>38</sup> <http://www.eclipse.org/>

<sup>39</sup> <http://subversion.tigris.org/>

<sup>40</sup> <http://www.staruml.com/>

## 6. CONCLUSIONES

El objetivo general de esta tesis era el desarrollo de una funcionalidad para la herramienta AQURATE a la cual se llamó QUIZ y se llevó a cabo durante el tiempo de estadía en la Universidad de Kingston en donde fue usada y aprobada por los desarrolladores de la herramienta. Todo el código y documentación forman parte del mismo repositorio de código que fue actualizado con la nueva funcionalidad, así como los cambios en la herramienta. En cuanto a los objetivos específicos, fueron alcanzados necesariamente para desarrollar la herramienta, esto es, el estudio de las principales especificaciones y estándares tales como IMS LOM, IMS *Content Packaging* e IMS QTI.

El hecho de que la Universidad de Kingston trabajó junto a las Universidades de Cambridge y Southampton, en un proyecto único a nivel mundial, en donde se trabajó con los estándares y especificaciones más importantes e influyentes en las Tecnologías de Aprendizaje, da un mayor trasfondo al trabajo realizado por el autor de este trabajo, debido a que las actualizaciones de código hechas, en el mismo repositorio que ocupan los proyectos como ASDEL y MINIBIX cumplen función de poner presente al grupo GITA así como la Universidad Austral de Chile. Estrechando lazos de comunicación y agradecimientos por el trabajo colaborativo realizado.

La herramienta y la funcionalidad fueron presentadas en *Aston Lakeside Conference*<sup>41</sup> en Birmingham, Inglaterra, realizado del 15 al 18 de Septiembre del 2008. En esta conferencia patrocinada por JISC e IMS se presentó la herramienta AQURATE como una de las implementaciones realizadas sobre la especificación IMS QTI presentado por Alicia Campos junto a David Livingstone. Además la funcionalidad realizada forma parte del artículo a presentar en el próximo congreso de la LACLO (*Latin American Comitee for Learning Objects*) a ser realizada en Octubre del año 2008 en México, en donde el autor de este trabajo es co-autor junto con el equipo de desarrollo de AQURATE.

### 6.1 Trabajos Futuros

La herramienta AQURATE es de código abierto y está disponible a la comunidad para ser descargada y estudiada. Considerando los errores que fueron reportados, es

---

<sup>41</sup> <http://www.abs.aston.ac.uk/conferenceaston/>

posible pensar en una versión nueva de la herramienta que incluya las soluciones a estos problemas, nuevas funcionalidades ó la implementación de más tipos de preguntas. Los proyectos como ASDEL y MINIBIX aún están trabajando por lo que es posible más adelante obtener un trabajo colaborativo entre los tres proyectos.

Considerando que el resultado de la funcionalidad es el uso de contenido HTML, sería posible incorporar o extender la funcionalidad utilizando dispositivos móviles en el uso o distribución del contenido. Funcionalidades que podrían considerarse como futuros trabajos de investigación para alumnos en el área informática y así aumentar el uso y usabilidad de las Tecnologías en Educación.

## 7. REFERENCIAS BIBLIOGRÁFICAS

- [ADL04a] Advanced Distributed Learning (2004), SCORM® Overview Version 1.3.1. Disponible en <http://www.adlnet.gov/scorm/>. Consultado el 17 de Agosto del 2008.
- [ADL04b] Advanced Distributed Learning (2004), (2006), SCORM® Sequencing and Navigation Version 1.0. Disponible en <http://www.adlnet.gov/scorm/>. Consultado el 17 de Agosto del 2008.
- [ADL04c] Advanced Distributed Learning (2004), SCORM® Content Aggregation Model Version 1.0. Disponible en <http://www.adlnet.gov/scorm/>. Consultado el 17 de Agosto del 2008.
- [ADL04d] Advanced Distributed Learning (2004), SCORM® Runtime Environment Version 1.0. Disponible en <http://www.adlnet.gov/scorm/>. Consultado el 17 de Agosto del 2008.
- [Als08] Alsop G., Annesley J., Cai Z., Campos A., Colbert M., Livingstone D., Smith G., and Orwell J. (2008). Issues raised developing Aqurate (an authoring tool that uses the IMS Question and Test Interoperability Version 2 Specification). 12th International CAA Conference. (Loughborough University, Leicestershire. UK, Julio 2008).
- [ALT] Association for Learning Technology (ALT). Disponible e [http://www.alt.ac.uk/learning\\_technology.html](http://www.alt.ac.uk/learning_technology.html) Consultado el 17 de Agosto del 2008.
- [AQU07b] Aqurate (2007), A QTI version 2,1 authoring Tool. Disponible en <http://aqurate.kingston.ac.uk/>. Consultado el 20 de Marzo del 2008.
- [AQU07a] AQURATE Project Proposal, (2007). AQuRate: A QTI Authoring Tool, Kingston University, London. England. Disponible en <http://aqurate.kingston.ac.uk/documentation/pdfs/AQuRateProposal.pdf>. Consultado el 25 de Agosto del 2008.
- [ASD07] ASDEL: Assessment Delivery Engine for QTIv2 questions Disponible en: <http://www.asdel.ecs.soton.ac.uk/overview/> Consultado el 20 de Julio del 2008.
- [Cam08] Campos A., Morales R., Alsop G., Annesley J., Colbert M., Livingstone D., Orwell J, Smith G. (2008). AQURATE Una Herramienta de Creación y Presentación de Objetos de Aprendizaje Bajo la Especificación IMS

QTI 2.1. Grupo de Investigación de Tecnologías de Aprendizaje. Universidad de Kingston. Londres, Inglaterra.

- [EDU] Iniciativa de Aprendizaje EDUCAUSE. Disponible en <http://www.educause.edu/node/5672>. Consultado el 17 de Agosto del 2008.
- [Fer05] Fernández B, Moreno P, Sierra J & Martínez I, (2005). Uso de estándares aplicados a TIC en educación. Serie de Informes del Centro Nacional de Información y Comunicación Educativa (CNICE-MEC). Ministerio de Educación y Ciencia. España. Disponible en <http://ares.cnice.mec.es/informes/16/>
- [IEE02] LOM, Learning Technology Standards Committee of the IEEE, (2002), Standard for Learning Object Metadata, IEEE 1484.12.1. Disponible en <http://ltsc.ieee.org/wg12/index.html>. Consultado el 16 de Agosto del 2008
- [IMS02] IMS Global Learning Consortium, (2002). IMS Learner Information Packaging Best Practice & Implementation Guide Final Specification Version 1.0. Disponible en <http://www.imsglobal.org/profiles/lipbest01.html>. Consultado el 16 de Agosto del 2008.
- [IMS03a] IMS Global Learning Consortium. (2003). IMS Learning Design. Best Practice and Implementation Guide. Version 1.0 Final Specification. Disponible en: [http://www.imsglobal.org/learningdesign/ldv1p0/imsld\\_bestv1p0.html](http://www.imsglobal.org/learningdesign/ldv1p0/imsld_bestv1p0.html). Consultado el 16 de Agosto del 2004.
- [IMS03b] IMS Global Learning Consortium, (2003). IMS Simple Sequencing Best Practice and Implementation Guide Version 1.0 Final Specification. Disponible en [http://www.imsglobal.org/simplesequencing/ssv1p0/imsss\\_bestv1p0.html](http://www.imsglobal.org/simplesequencing/ssv1p0/imsss_bestv1p0.html). Consultado el 16 de Agosto del 2008
- [IMS04] IMS Global Learning Consortium, (2004). IMS Content Packaging Best Practice and Implementation Guide. Disponible en [http://www.imsglobal.org/content/packaging/cpv1p1p4/imscp\\_bestv1p1p4.html](http://www.imsglobal.org/content/packaging/cpv1p1p4/imscp_bestv1p1p4.html). Consultado el 16 de Agosto del 2004.
- [IMS06a] IMS Global Learning Consortium, (2006). IMS Meta-data Best Practice Guide for IEEE 1484.12.1-2002 Standard for Learning Object Metadata. Version 1.3 Final Specification. Disponible en

[http://www.imsglobal.org/metadata/mdv1p3/imsmd\\_bestv1p3.html](http://www.imsglobal.org/metadata/mdv1p3/imsmd_bestv1p3.html).

Consultado el 16 de Agosto del 2008.

- [IMS06b] IMS Global Learning Consortium, (2006). IMS Question and Test Interoperability Overview Version 2.1 Public Draft (revision 2) Specification. Disponible en <http://www.imsglobal.org/question>. Consultado el 13 de Marzo del 2008.
- [IMS06c] IMS Global Learning Consortium, (2006). IMS Question and Test Interoperability Assessment Test, Section, and Item Information Model Version 2.1 Public Draft revision 2 Specification. Disponible en <http://www.imsglobal.org/question>. Consultado el 13 de Marzo del 2008.
- [IMS06d] IMS Global Learning Consortium, (2006). IMS Question and Test Interoperability Implementation Guide Version 2.1 Public Draft (revision 2) Specification. Disponible en <http://www.imsglobal.org/question>. Consultado el 13 de Marzo del 2008.
- [JIS07a] JISC Development Programmes. Project Document Cover Sheet. Aqurate Project Plan (2007), Project Manager: Graham Alsop, University of Kingston. Disponible en <http://aqurate.kingston.ac.uk/documentation/pdfs/Project%20Plan%201.3%20public.pdf>. Consultado el 20 de Julio del 2008.
- [JIS07b] JISC Development Programmes. Project Document Cover Sheet. ASDEL Project Plan (2007), Project Manager: Gary Wills, University of Southampton. Disponible en <http://www.jisc.ac.uk/media/documents/programmes/elearningcapital/asdelprojectplan.pdf>. Consultado el 20 de Julio del 2008.
- [JIS07c] JISC Development Programmes. Project Document Cover Sheet. Minibix Project Plan (2007), Project Manager: Steve Lay, University of Cambridge. Disponible en [http://www.jisc.ac.uk/media/documents/programmes/elearningcapital/minibixprojectplan\(caret\).pdf](http://www.jisc.ac.uk/media/documents/programmes/elearningcapital/minibixprojectplan(caret).pdf). Consultado el 20 de Julio del 2008.
- [Kin99] Kinshuk (1999). IEEE Computer Society. Learning Technology Newsletter, 1(1):1-2, Julio. Disponible en [http://lutf.ieee.org/learn\\_tech/issues/july99/learn\\_tech\\_july1999.pdf](http://lutf.ieee.org/learn_tech/issues/july99/learn_tech_july1999.pdf).
- [LTS] LTSC Learning Technologies Standards Committee (Comité de estándares para tecnologías de Aprendizaje). Disponible en <http://ieeeltsc.org/>.

- [Mas03] Masie (2003). Making Sense of Learning Specifications and Standards: A Decision Maker's Guide to their Adoption, 2nd ed. e-Learning Consortium Industry Report, The Masie Center. Disponible en: [http://www.masie.com/standards/s3\\_2nd\\_edition.pdf](http://www.masie.com/standards/s3_2nd_edition.pdf) (último acceso, 16 Octubre 2007).
- [MIN07] Minibix (2007): A QTI-based Item Bank. Disponible en: [http://qtitools.caret.cam.ac.uk/index.php?option=com\\_content&task=category&sectionid=4&id=39&Itemid=32](http://qtitools.caret.cam.ac.uk/index.php?option=com_content&task=category&sectionid=4&id=39&Itemid=32) . Consultado el 21 de Julio del 2008.
- [Ort03] Ed Ort and Bhakti Mehta, (2003). *Java Architecture for XML Binding* (JAXB), Tutorial. Disponible en <http://java.sun.com/developer/technicalArticles/Webservices/jaxb/>. Consultado el 22 de Agosto del 2008
- [R2Q06] R2Q2: Rendering and Response processing services for QTIv2 questions. Disponible en <http://www.r2q2.ecs.soton.ac.uk/overview/>. Consultado el 20 de Agosto del 2008.
- [Sun02] Sun Microsystems, (2002): "e-learning interoperability standards". Disponible en [http://www.sun.com/products-n-solutions/edu/whitepapers/pdf/eLearning\\_Interoperability\\_Standards\\_wp.pdf](http://www.sun.com/products-n-solutions/edu/whitepapers/pdf/eLearning_Interoperability_Standards_wp.pdf).

## **8. ANEXOS DIGITALES**

Debido a la cantidad de archivos de código y documentación, fue necesario incluirlos en formato digital junto a este trabajo. La organización de los anexos digitales que se adjuntan en un medio óptico, se dividen en tres carpetas, además del presente documento en formato digital.

### **8.1 Anexo Digital A**

El contenido de este anexo se almacena en la carpeta Anexo A. Aquí se encuentran todos los archivos de código fuente de la herramienta AQURATE, está incluida la funcionalidad QUIZ debido a que es la última versión.

El código fuente también puede ser descargado de la siguiente dirección: <https://qtitools.svn.sourceforge.net/svnroot/qtitools> utilizando el manual disponible en el Anexo Digital C.

### **8.2 Anexo Digital B**

El contenido de este anexo se almacena en la carpeta Anexo B y es la documentación existente sobre la herramienta, así como los generados durante el desarrollo como diagramas UML, cambios realizados a la herramienta y errores encontrados.

Estos documentos también pueden ser descargados de la siguiente dirección: <http://code.google.com/p/qtidevelopment/downloads/list>

### **8.3 Anexo Digital C**

El contenido de este anexo se almacena en la carpeta Anexo C y es un manual para instalar el entorno de desarrollo en Eclipse y la descarga del código para trabajar con la herramienta AQURATE.

Estos documentos también pueden ser descargados de la siguiente dirección:

<http://code.google.com/p/qtidevelopment/downloads/list>