

Universidad de las Ciencias Informáticas

Facultad 4



**API para el desarrollo de un reproductor de
cuestionarios basados en IMS QTI en la herramienta
de autor web CRODA**

**Trabajo de diploma para optar por al título de Ingenieros en
Ciencias Informáticas**

Autor:

Gleydis Correa Medina

Tutores:

Ing. Osvaldo E. Stable Vilches

Ing. Yisel Quiñones Guerrero

La Habana Junio 2013

Declaración de autoría

Declaración de Autoría:

Por este medio declaro que soy la única autora de este trabajo y autorizo a la Universidad de las Ciencias Informáticas (UCI) para que hagan el uso que estimen pertinente con este trabajo.

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

Gleydis Correa Medina

Firma del Autor

Ing. Osvaldo E. Stable Vilches

Firma del Tutor

Ing. Yisel Quiñones Guerrero

Firma del Tutor



"Antes era la impotencia y la ignorancia, antes eran las manos vacías, las manos sin armas y la mente sin educación: ¡Hoy armamos los brazos y armamos las mentes!"

Fidel Castro Ruz



"Para el logro del triunfo siempre ha sido indispensable pasar por la senda de los sacrificios"

Simón Bolívar



Agradecimientos

A toda mi familia por todo el apoyo me han dado incluso cuando todo parecía perdido para mí, gracias a mi mamá fundamentalmente por tener tanta calma conmigo, por desempeñar también su papel de madre y al mismo tiempo ser mi amiga, mi guía ya que me costó mucho darme cuenta de la importancia que tiene llegar a ser alguien en la vida y eso se lo debo en gran parte a ella.

A mi papá y mi hermano por todos sus buenos consejos por preocuparse por mis cosas y darme siempre mucho ánimo para seguir adelante.

A mi amigo y gran amor Osvaldo porque a pesar de todo tuvo fe en mí, por tener tanta paciencia conmigo y dedicarme por completo cada minuto de su vida, no solo a lo que se refiere a mis estudios sino también en el plano personal, es una persona muy fundamental en mi vida.

A Yusdel, Lizardo, Mayrin, Cindy gracias por el tiempo y el esfuerzo que una vez me dedicaron para que yo pudiera salir bien en una determinada materia eso yo no lo olvido y les voy a estar eternamente agradecida.



Agradecimientos

A Arcadio y Tony por darme la oportunidad de seguir mis estudios en un momento en que parecía imposible.

A mis tutores Osvaldo, Yisel y mi oponente Leonardo por guiarme, por su confianza y corregirme siempre que fue necesario.

A mis compañeros de grupo especialmente a Ramón Vega por dedicarme tanto tiempo estudiando programación, gracias a sus explicaciones y dedicación pude sacar programación 5, a todos mis compañeros muchas gracias por su ayuda.

Resumen

La incorporación de las Tecnologías de la Información y las Comunicaciones (TIC) al proceso enseñanza aprendizaje sirve de apoyo en la formación de estudiantes en varias materias. Dentro de los recursos que contribuyen a ello se encuentran los objetos de aprendizaje (OA), estos dentro de su composición pueden tener actividades de autoevaluación estandarizadas como las que pueden ser incorporadas en un OA durante su diseño en la herramienta de autor web CRODA. Pero durante su diseño no poseen un mecanismo que permita visualizar estas actividades como un proceso de comprobación de las mismas, trayendo consigo posibles errores. Por lo que surge la interrogante ¿Cómo contribuir al desarrollo de un reproductor que permita la visualización de las actividades basadas en IMS QTI en la herramienta de autor web CRODA? Para conseguir una solución a dicho problema se decide desarrollar una Interfaz de Programación de Aplicaciones API que contribuya a la reproducción de contenidos basados en la especificación IMS QTI en la herramienta de autor web CRODA.

Para el desarrollo de una API se analizaron las tecnologías a utilizar, se hizo un estudio detallado de la especificación IMS QTI en la cual estarán basadas las actividades que se quieren reproducir y se realizó en análisis y diseño de un reproductor de actividades basadas en la especificación IMS QTI para CRODA con el objetivo de identificar las funcionalidades que debe brindar una API. Por último se desarrolló una API que favorecerá la reproducción de actividades basadas en la especificación IMS QTI, permitiendo la visualización y verificación en tiempo de ejecución de las actividades estandarizadas diseñadas en la herramienta de autor web CRODA.

Palabras clave: actividad, API, autoevaluación, CRODA, especificación, IMS, tecnologías, QTI.

Introducción	1
Capítulo 1: Fundamentación Teórica	6
1.1 Introducción.....	6
1.2 La enseñanza virtual (e-learning)	6
1.3 Objetos de aprendizaje (OA).....	7
1.4 IMS Interoperabilidad de preguntas y exámenes	8
1.4.1 Versiones de la especificación IMS QTI	9
1.4.2 Perfiles de la especificación IMS QTI.....	10
1.4.3 IMS QTI 2.0.....	11
1.5 Herramientas relacionadas con la especificación IMS QTI.....	16
1.6 Reproductores de contenidos basados en IMS QTI.....	16
1.7 API (Interfaz para la Programación de Aplicaciones).....	18
1.8 Metodologías de desarrollo de software.....	19
1.9 Lenguajes de Programación	21
1.10 Framework de desarrollo	22
1.11 Entorno Integrado de Desarrollo (IDE)	23
1.12 Sistema gestor de base de datos	24
1.13 Servidor web.....	24
1.14 Herramienta CASE	25
1.15 Lenguaje Unificado de Modelado (UML).....	25
1.16 Conclusiones	26
Capítulo 2: Desarrollo de la propuesta solución.	27
2.1 Introducción.....	27
2.2 Modelo de dominio.....	27
2.2.1 Representación del Diagrama de Modelo de Dominio	28
2.2.2 Definición de las clases del Modelo de Dominio.....	28
2.3 Requerimientos	29

2.4 Modelo de casos de uso del sistema.....	33
2.4.1 Actores del sistema	33
2.4.2 Casos de uso del sistema	34
2.4.3 Diagrama de Casos de Uso del sistema.....	35
2.5 Descripción textual de los casos de uso del sistema	36
2.6 Modelo del análisis	42
2.6.1 Diagrama de clases del análisis	43
2.6.2 Diagramas de interacción	44
2.7 Diseño	45
2.7.1 Diagramas de clases del diseño.....	45
2.7.2 Modelos de Datos	47
2.8 API-QTI.....	47
2.9 Diagrama de paquetes	49
2.10 Diagrama de componentes	50
2.11 Aplicaciones de patrones de diseño	51
2.12 Conclusiones	52
Capítulo 3: Validación de la solución propuesta	53
3.1 Introducción.....	53
3.2 Métodos de validación.....	53
3.2.1 Métricas de la Calidad de Especificación de los Requisitos	53
3.2.2 Métricas de Casos de Uso	56
3.2.3 Aplicación de la técnica de ladov	58
3.3 Conclusiones	61
Conclusiones generales.....	62
Recomendaciones	63
Bibliografía	67
Glosario de Términos	67
Anexos.....	67

Índice

Anexo 1: Diagramas de clases del análisis.....	67
Anexo 2: Diagramas de colaboración.....	68
Anexo 3: Diagrama de secuencia.....	70
Anexo 4: Diagrama de clase del diseño.....	72
Anexo 5: Validación de Requerimientos.....	74
Anexo 6: Cuestionario para la aplicación de la técnica de ladov.....	78

Introducción

En la actualidad las Tecnologías de la Información y las Comunicaciones (TIC), brindan herramientas y materiales que apoyan el aprendizaje, y por consiguiente, el desarrollo de habilidades (1).

La incorporación de las TIC al proceso de enseñanza aprendizaje posibilitó el surgimiento del aprendizaje electrónico o e-learning el cual permite la entrega de material educativo vía cualquier medio electrónico incluyendo el intranets, internet, extranet, audio, video, red satelital, televisión interactiva, CD y DVD entre otros medios” (2).

Dentro del e-learning surge la necesidad de crear y distribuir recursos digitales con fines formativos, por lo que se crean los objetos de aprendizaje (OA) considerados recursos digitales estandarizados, descritos por metadatos, compuesto por uno o varios objetos de información, que responde a un único objetivo de aprendizaje y pueden ser utilizados en diversas situaciones de enseñanza aprendizaje (3).

Los OA brindan la posibilidad de ser recursos interoperables, accesibles, reutilizables y duraderos, permitiendo su utilización en aplicaciones, como los repositorios de objetos de aprendizaje (ROA) que hacen posible su almacenamiento y facilitan su búsqueda, las herramientas de autor que tienen como objetivo la creación de los OA y los sistemas gestores del aprendizaje (LMS) que garantizan la incorporación de estos recursos para el apoyo de un determinado proceso de enseñanza.

En Cuba específicamente en la Universidad de las Ciencias Informáticas (UCI) se realizan importantes acciones con el objetivo de apoyar el proceso docente con el uso de plataformas educativas. La UCI cuenta con un Centro de Tecnologías para la Formación (FORTES) que brinda servicios y productos para la implementación de soluciones para la formación aplicando el uso de las TIC.

En el centro FORTES se desarrollan varios proyectos, dentro de estos se encuentra CRODA (Creando Objetos de Aprendizaje), herramienta de autor web que se especializa en la creación de los OA. Este recurso dentro de su estructura en la mayoría de las ocasiones se hace

necesario crear un espacio para apoyar el proceso de autoevaluación y para ello existe la posibilidad de incluir algunas tipologías de ejercicios utilizando la especificación IMS QTI con el objetivo de mantener la interoperabilidad. Dentro de estas tipologías de ejercicios se encuentran las de selección, llenar espacios, enlazar columnas, asociación y de ordenamiento, las cuales son actividades que permiten llevar a cabo el proceso de autoevaluación de los contenidos adquiridos por los estudiantes.

En su versión actual CRODA no permite comprobar en tiempo de ejecución las actividades estandarizadas elaboradas con sus correspondientes peculiaridades de vista, evaluación y retroalimentación, lo cual puede traer posibles errores en la estructura o funcionamiento de los objetivos que quiere lograr el autor con el acabado de la actividad. Debido a que pueden ocurrir incoherencias en las opciones, las respuestas, la calificación o mensajes de retroalimentación. El editor de actividades creado tampoco cuenta con la posibilidad de reutilizar componentes o funcionalidades que sirvan de apoyo a la visualización de este tipo de actividades en tiempo de ejecución.

Por lo anteriormente expuesto se plantea como **problema a resolver**: ¿Cómo contribuir al desarrollo de un reproductor que permita la visualización de las actividades basadas en la especificación IMS QTI en la herramienta de autor web CRODA? Para darle solución a este problema se define como **objeto de estudio**: proceso de reproductores de actividades.

El **campo de acción** de esta investigación de enmarca en la reproducción de actividades basadas en la especificación IMS QTI.

Se propone como **objetivo general** desarrollar una API que contribuya al desarrollo de un reproductor de actividades basadas en la especificación IMS QTI en la herramienta de autor web CRODA. Se definen como **objetivos específicos**:

- Examinar el marco teórico para la creación de una API que contribuye al desarrollo de un reproductor de actividades basadas en IMS QTI.
- Desarrollar la propuesta de solución.
- Validar la solución desarrollada mediante

Se definieron para la presente investigación las siguientes **Preguntas científicas**:

- ¿Cuáles son las tecnologías necesarias para el desarrollo de una API que contribuya a la reproducción de actividades basadas en IMS QTI en CRODA?
- ¿Cómo determinar las funcionalidades que debe brindar una API para contribuir a la reproducción de actividades basadas en IMS QTI en CRODA?
- ¿Cómo crear los componentes y funcionalidades en una API que contribuya a la reproducción de actividades basadas en IMS QTI en CRODA?
- ¿Es recomendable la propuesta solución desarrollada?

Para dar cumplimiento a los objetivos se plantean las siguientes **tareas de investigación**:

- Estudio de la especificación IMS QTI.
- Estudio de reproductores de contenidos basados en IMS QTI.
- Análisis de los conceptos y tecnologías más utilizadas para conformar la base teórica metodológica de la investigación.
- Elaboración del análisis y diseño de un reproductor de OA basados en la especificación IMS QTI en CRODA, para determinar las necesidades que debe brindar una API que contribuirá a su desarrollo.
- Selección de la metodología, herramientas y tendencias actuales más factibles para desarrollar la solución del problema.
- Desarrollo de funcionalidades en una API que contribuya a la visualización de los contenidos IMS QTI en CRODA.
- Validación de la solución desarrollada mediante las Métricas de la Calidad de Especificación de los Requisitos y Casos de Uso, así como la técnica de ladov.

Los **métodos teóricos utilizados** en el transcurso de esta investigación son:

El **Analítico-Sintético**: Se aplica en la presente investigación durante el análisis de aplicaciones que permitan reproducir cuestionarios, el modelo de información generado por la especificación IMS QTI para la creación de cuestionarios y tecnologías existentes que permitan el desarrollo de una API que contribuya al desarrollo de un reproductor para la visualización de los contenidos basados en IMS QTI.

El **Histórico-Lógico**: El empleo de este método permite estudiar el origen y evolución de la especificación IMS QTI.

Modelación: Su aplicación permite el esbozo de los diferentes diagramas y modelos generados durante el proceso de desarrollo de la aplicación.

La presente investigación tiene la siguiente **Estructura Capitular**:

Capítulo 1.-Fundamentación Teórica:

En este capítulo se abarcan conceptos relacionados con la especificación IMS QTI. Además contiene un estudio del estado del arte de las herramientas y tecnologías utilizadas para el desarrollo de una API que contribuya a la solución del problema de la investigación. Este análisis posibilita la recopilación de información que definen las herramientas y metodología que se usaran para el desarrollo de esta investigación.

Capítulo 2.-Desarrollo de la propuesta de solución.

En este capítulo se elabora una descripción de la solución que se propone, donde se representa primeramente el modelo de dominio el cual constituye una descripción gráfica de la estructura del XML para modelar un cuestionario. Se desarrolla el flujo de trabajo Requerimientos de la metodología RUP, definiendo los requisitos funcionales y no funcionales del sistema a desarrollar, llegando así a los casos de uso, donde además se realiza una descripción textual de los mismos. Se muestran los principales artefactos generado por el flujo de trabajo Análisis y Diseño, los diagramas de paquetes y componentes, y se realiza una descripción de las clases utilizadas en la realización de una API.

Capítulo 3.-Validación de la solución propuesta.

Introducción

En este capítulo se realiza la validación de los requisitos y casos de uso, utilizando los métodos de métricas de la calidad de especificación de los requisitos y métricas para la especificación de los casos de uso, esto permite comprobar que estos artefactos están correctamente detallados. Se utiliza la técnica de ladov para validar los resultados obtenidos con el objetivo de determinar el índice de satisfacción grupal con respecto a la solución creada.

Capítulo 1: Fundamentación Teórica

1.1 Introducción

En el presente capítulo se precisan los elementos teóricos necesarios para el desarrollo de una API (Interfaz de Programación de Aplicaciones). Se abordan conceptos fundamentales referentes a la especificación IMS QTI, así como la selección de tecnologías, metodología y herramientas necesarias para lograr la solución planteada en la presente investigación.

1.2 La enseñanza virtual (e-learning)

En la actualidad las TIC desempeñan un papel fundamental en el proceso de aprendizaje. El desarrollo e introducción de estas en el ámbito de la educación ha propiciado el surgimiento del e-learning que se define como “conjunto de tecnologías, aplicaciones y servicios orientados a facilitar la enseñanza y el aprendizaje a través de Internet/Intranet, que facilitan el acceso a la información y la comunicación con otros participantes” (4).

El empleo del e-learning en la formación de la enseñanza superior aporta ventajas tales como (5):

- Flexibilidad en tiempo y espacio para el desarrollo de las actividades de enseñanza-aprendizaje.
- Posibilidad de interactuar con la información por parte de los diferentes agentes que intervienen en las actividades.
- Enseñanza más personalizada.
- Acceso rápido a la información.

Para que el proceso de enseñanza-aprendizaje mediante el e-learning alcance su máxima calidad este requiere de la utilización de estándares y especificaciones, los cuales van a garantizar la accesibilidad, interoperabilidad, adaptabilidad, reusabilidad, durabilidad y productividad de los contenidos.

Capítulo 1: Fundamentación Teórica

Se considera que de los estándares utilizados en el e-learning el IMS (*Global Learning Consortium, Inc*) es el desarrollador de varias especificaciones entre la que se encuentra IMS QTI. La especificación IMS QTI contempla una estructura básica que describe la forma de representar preguntas individuales y gestionar exámenes completos, su objetivo es conseguir que tanto las evaluaciones como los resultados sean intercambiables entre los diferentes sistemas de gestión de aprendizaje.

1.3 Objetos de aprendizaje (OA)

El surgimiento del *e-learning* como soporte al proceso de enseñanza aprendizaje, trae como consecuencia la creación los objetos de aprendizaje, con el objetivo de compartir y reutilizar sus contenidos, desde sus inicios han sido varios los autores que le han dado una definición diferente a su concepto. Un objeto de aprendizaje es “una pieza digital de material de aprendizaje que direcciona a un tema claramente identificable y que tiene el potencial de ser reutilizado en diferentes contextos”. La Institución de Ingenieros Eléctricos y Electrónicos (IEEE) lo define como “una entidad, digital o no digital que puede ser utilizada, reutilizada y referenciada durante el aprendizaje apoyado con tecnología” (6).

La UCI lo define como un: “Recurso didáctico digital estandarizado, descrito por metadatos, compuesto por uno o varios objetos de información, que responde a un único objetivo de aprendizaje y puede ser utilizado en diversas situaciones de enseñanza-aprendizaje”.

En la Universidad de las Ciencias Informáticas específicamente en los proyectos RHODA y CRODA se define la estructura de los OA de la siguiente manera para así brindar una mejor organización en la información que se le ofrecerá al usuario (7):

- **Objetos de Información:** Está formado por activos de aprendizaje como videos, gráficos, animación, entre otros. Contiene toda la información que permite al usuario poder interactuar con el OA para dar cumplimiento a sus objetivos. Esta información sería de navegación simple como cuadros de búsquedas, barra de enlaces y barra de herramientas, entre otras.

Capítulo 1: Fundamentación Teórica

- **Metadatos:** Los que incluyen información necesaria para catalogar el OA, tales como el título, autor, contenido, vista, objetivo, especificaciones técnicas y otras definidas según el estándar utilizado. Permitiendo así buscar rápidamente, organizar y actualizar los OA.
- **Estructura didáctica:** Es la organización del contenido educativo basado en una unidad de aprendizaje, definida pedagógicamente.

Son varios los componentes que estructuran los OA, la estructura didáctica es el componente el cual la autoevaluación forma parte de él, para el desarrollo de la misma se utilizará la especificación IMS QTI la cual tiene como objetivo plantear y documentar un formato de contenido donde se puedan almacenar las preguntas y respuestas con el fin de generar evaluaciones usando las herramientas de auditoría las cuales son las encargadas de crear los OA.

1.4 IMS Interoperabilidad de preguntas y exámenes

Los exámenes son herramientas de evaluación que le permite al profesor medir el conocimiento adquirido por el estudiante. Estas herramientas de evaluación y autoevaluación han avanzado en cuanto a su informatización donde la mayoría de las preguntas pueden ser revisadas o corregidas de manera automática.

En la actualidad muchas plataformas de aprendizaje incluyen en su funcionalidad herramientas para la creación, gestión y realización de exámenes en línea. Debido que la creación de dichos exámenes requiere de cierto esfuerzo, para compartir el mismo cabe la posibilidad de permitir intercambiar exámenes completos o crear repositorios de preguntas (8).

La especificación IMS Interoperabilidad de Preguntas y Exámenes por sus siglas en inglés (IMS QTI) permite crear preguntas individuales y exámenes completos. Su principal objetivo es permitir el intercambio de preguntas, exámenes y resultados entre las distintas aplicaciones con fines educativos. Con este propósito IMS QTI plantea un modelo en el que se definen los componentes principales que intervienen en el proceso de autoevaluación y adicionalmente a este modelo, se proporciona un formato de contenido XML (*Extensible Markup Language*) para almacenar las actividades de manera independientemente del sistema o herramienta de autoría utilizada para crearlas.

1.4.1 Versiones de la especificación IMS QTI

La especificación IMS QTI cuenta con cinco versiones las cuales son: 1.0, 1.1, 1.2, 2.0 y 2.1 de las cuales tres de estas versiones tuvieron gran repercusión: **IMS QTI versión 1.2, IMS QTI versión 2.0 y finalmente IMS QTI versión 2.1** (8).

IMS QTI 1.2: Esta versión surge en el año 2002, fue la última finalizada completamente en la que se abarcan tanto preguntas individuales como exámenes completos. Una vez publicada esta especificación surgen nuevos problemas a la hora de implementarla lo que trajo como consecuencia que se publicara un suplemento (IMS QTI 1.2.1) debido a los problemas surgidos se requirió de grandes cambios de manera que dichas modificaciones provocarían que se perdiera la compatibilidad con las versiones anteriores. Además, algunas otras partes de la especificación necesitaban clarificarse o extenderse para resolver los problemas que habían surgido durante su puesta en práctica.

IMS QTI 2.0: Con el nacimiento de IMS QTI 1.2 y de nuevas especificaciones como el *IMS Content Packaging*, *IMS Simple Sequencing* y *IMS Learning Design* surge la necesidad de hacer compatible IMS QTI a las nuevas iniciativas es así como aparece la versión 2.0 de la especificación en el año 2005 en esta versión IMS QTI logra tener compatibilidad con las demás especificaciones y da solución a los problemas generados en la versión anterior. Esta especificación se centró solo en las preguntas individuales.

IMS QTI 2.1: Esta versión surge en el año 2006 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.

Después de realizar un estudio de las versiones de la especificación IMS QTI y ver sus características se ha decidido trabajar con la versión 2.0 por las siguientes razones: La versión 1.2 presentó problemas a la hora de su implementación dando así paso a la versión 1.2.1 la cual experimentó grandes cambios que provocaron la pérdida de compatibilidad con las versiones anteriores. La 2.1 aún se encuentra en proceso de evolución en modo borrador por lo

cual no se considera estable, quedando así solo la versión 2.0 como la más adecuada para ser usada en este trabajo con el inconveniente de que no permite la creación de exámenes completos.

1.4.2 Perfiles de la especificación IMS QTI

La especificación IMS QTI determina dos perfiles, los cuales se utilizan para precisar si se soporta la especificación completa o solo parte de ella. Estos perfiles son denominados *QTI-Lite* (que solo se aplica al contenido) y *QTI-ALL* que conforma la especificación completa.

QTI-Lite se centra solo en la creación y modificación del contenido, este además no soporta los sistemas de banco de ítems, ni es compatible con todas las características de la especificación, sin embargo constituye un perfil adecuado ya que describe la forma más sencilla de la especificación compuesta por un conjunto de funcionalidades y características (8).

Las principales diferencias entre *QTI-Lite* y el *QTI-ALL* son:

- *QTI-Lite* solo permite una interacción por ítem.
- El único tipo de interacción que soporta *QTI-Lite* es *choiceInteraction* y es compatible solo con las tipologías de ejercicios (selección simple, respuesta múltiple y verdadera o falsa).
- *QTI-Lite* no soporta los comentarios integrados (*Feedback*).
- *QTI-Lite* limita los tipos de imágenes soportadas y los tipos de estructuras.
- *QTI-Lite* no soporta características avanzadas como ítems adaptativos (adaptive ítems), plantillas (templates) y puntuación basada en el tiempo.

Para el desarrollo de una API que contribuya a la reproducción de contenidos basados en IMS QTI el perfil a utilizar es el *QTI-ALL* ya que es compatible con las cinco preguntas que se quieren reproducir de las cuales dos de ellas no son compatible con *QTI-Lite*.

1.4.3 IMS QTI 2.0

La versión 2.0 de la especificación IMS QTI intenta simplificar su uso tanto desde el punto de vista técnico como desde el punto de vista del usuario de dicha especificación, para esto se han determinado los siguientes conceptos (8):

Las preguntas

Las preguntas individuales podrán ser utilizadas como un recurso educativo independiente, por ejemplo, como un recurso más dentro de un paquete IMS. Estas preguntas individuales (*assessmentItem*) son auto contenidas, van a incluir toda la información necesaria para su presentación al alumno y su corrección automática. Toda la información relacionada con la presentación es agrupada en el cuerpo (*itemBody*) de las preguntas. En dicha presentación se tendrán en cuenta dos aspectos:

El enunciado de la pregunta: Las preguntas deben contener un enunciado y de forma adicional un material explicativo complementario que indique al docente en qué contexto se realiza la pregunta. La especificación IMS QTI 2.0 los contenidos que se utilizan dentro del cuerpo de las preguntas sigue el estándar XHTML, es decir contenido web y además utiliza el estándar *MathML* para las representaciones de ecuaciones matemáticas.

La construcción de la respuesta: Al enunciado de la pregunta se le debe adicionar el equivalente de un lápiz y papel para que el docente pueda realizar la respuesta. De acuerdo a la versión 2.0 de la especificación se ha introducido el concepto de interacción (*interaction*).

Las interacciones en IMS QTI

Las interacciones permiten al profesor especificar las herramientas con las que va a contar el alumno para poder construir las respuestas. Existen múltiples interacciones que pueden utilizarse dentro de las preguntas (8).

Se debe especificar que existen dos clases de interacciones, las interacciones en líneas y las interacciones en bloque. Las interacciones en líneas son las que se pueden incluir al enunciado y las interacciones en bloque son las que se representan independientes al enunciado.

Capítulo 1: Fundamentación Teórica

Para ejemplificar las posibles interacciones debido a la falta de herramientas que brinden un soporte completo a la especificación, se hará uso de los ejemplos que proporciona la propia especificación.

Interacciones Simples

Las interacciones simples con aquellas 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 (8):

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 sean intercambiadas entre distintos intentos del alumno.

The image shows a user interface for a 'ChoiceInteraction'. At the top, the title 'UNATTENDED LUGGAGE' is displayed. Below it, the instruction 'Look at the text in the picture.' is shown. In the center, there is a rectangular box containing the text 'NEVER LEAVE LUGGAGE UNATTENDED'. Below this box, the question 'What does it say?' is presented. Underneath the question, there are three radio button options:

What does it say?	
<input type="radio"/> You must stay with your luggage at all times.	<input type="radio"/>
<input type="radio"/> Do not let someone else look after your luggage.	<input type="radio"/>
<input type="radio"/> Remember your luggage when you leave.	<input type="radio"/>

Ilustración 1: Ejemplo de choiceInteraction (selección simple)

Fuente: (IMS Global Learning Consortium, 2005)

Capítulo 1: Fundamentación Teórica

COMPOSITION OF WATER	
Which of the following elements are used to form water?	
Carbon	<input type="checkbox"/>
Oxygen	<input checked="" type="checkbox"/>
Hydrogen	<input checked="" type="checkbox"/>
Chlorine	<input type="checkbox"/>
Helium	<input type="checkbox"/>
Nitrogen	<input type="checkbox"/>

Ilustración 2: Ejemplo de choiceInteraction (respuesta múltiple)

Fuente: (IMS Global Learning Consortium, 2005)

OrderInteraction: En esta interacción el objetivo del alumno es reordenar el conjunto de soluciones proporcionadas. 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.

GRAND PRIX OF BAHRAIN		
The following F1 drivers finished on the podium in the first ever Grand Prix of Bahrain. Can you rearrange them into the correct finishing order?		
Rubens Barrichello	Jenson Button	Michael Schumacher

Ilustración 3: Ejemplo de OrderInteraction

Fuente: (IMS Global Learning Consortium, 2005)

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

Capítulo 1: Fundamentación Teórica

posible indicar el número mínimo y máximo de veces que una de las opciones puede aparecer dentro de una asociación.

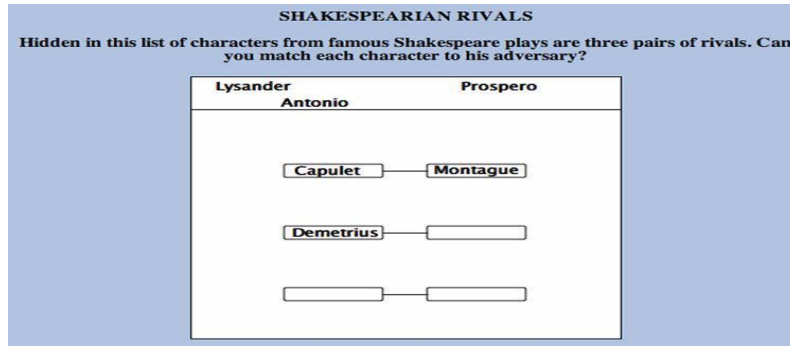


Ilustración 4: Ejemplo de AssociateInteraction

Fuente: (IMS Global Learning Consortium, 2005)

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.

CHARACTERS AND PLAYS

Match the following characters to the Shakespeare play they appeared in:	The Tempest	Romeo and Juliet	A Midsummer-Night's Dream
Prospero			
Capulet			
Demetrius			
Lysander			

Ilustración 5: Ejemplo de MatchInteraction

Fuente: (IMS Global Learning Consortium, 2005)

GapMatchInteraction: Esta interacción permite definir un conjunto de huecos dentro del enunciado de la pregunta a mostrar al alumno. Además se permitirá al alumno asociar a cada uno de los huecos una de las opciones de respuesta. Hay que destacar que las opciones son

Capítulo 1: Fundamentación Teórica

compartidas por todos los huecos. Como posibles respuestas se puede utilizar texto o también utilizar imágenes. Además se restringe el número mínimo y máximo de veces que es utilizada cada una de las posibles opciones del conjunto de respuestas.

RICHARD III (TAKE 1)

Identify the missing words in this famous quotation from Shakespeare's Richard III.

Now is the of our discontent
Made glorious by this sun of York;
And all the clouds that lour'd upon our house
In the deep bosom of the ocean buried.

Use the table below to select the missing words.

	winter	spring	summer	autumn
Word 1	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Word 2	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Ilustración 6: Ejemplo de GapMatchInteraction

Fuente: (IMS Global Learning Consortium, 2005)

La especificación cuenta con más interacciones como las de gráficas de texto entre otras, las cuales no serán tratadas en el presente trabajo, ya que las cinco tipos de preguntas que se quieren llegar a reproducir están contenidas dentro de las interacciones simples.

Las interacciones que soportan las preguntas que se van a visualizar en el reproductor a través de una API son:

- *ChoiceInteraction* (verdadero o falso, selección simple y respuestas múltiples).
- *OrderInteraction* (ordenamiento).
- *MatchInteraction* (pares de asociaciones).
- *GapMatchInteraction* (espacios en blanco).
- *AssociateInteraction* (asociación por parejas).

1.5 Herramientas relacionadas con la especificación IMS QTI

La especificación IMS QTI cuenta con varias herramientas que proporcionan soporte en mayor o menor medida a la especificación en sus diferentes versiones. Las herramientas pueden ser divididas en tres categorías.

Herramientas de Autoría. Herramientas que permitirán crear exámenes y finalmente salvarlos en el formato IMS QTI entre las que se encuentra *Respondus, QuestionMark's Perception, Assesst Designer, AquRate, Mathqurate.*

LMS (Sistemas Gestores de Aprendizaje). 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 entre las que se encuentran *Alphastudy, ANGEL Learning Management Suite, DB Primary, ILIAS, Claroline, SAKAI, .LRN, OLAT, WebCT.*

Reproductores de ejecución. Los reproductores interpretarán preguntas y exámenes compatibles con IMS QTI y permitirán realizar el examen.

1.6 Reproductores de contenidos basados en IMS QTI

Los exámenes son utilizados ampliamente como herramienta para evaluar el avance de un estudiante en una determinada materia. En este sentido, el uso de aplicaciones informáticas ha evolucionado. Varias tipologías de actividades pueden ser corregidas de manera automática mediante el uso de este tipo de herramientas, de manera que un profesor puede crear una batería de preguntas que el sistema informático, puede utilizar para preparar exámenes y corregirlos automáticamente.

En la presente investigación se hacen un estudio algunos sistemas gestores de aprendizaje que permiten la reproducción de contenidos basados en la especificación IMS QTI estos son:

Moodle

Moodle es una aplicación de código abierto basadas en módulos, denominado Entorno Virtual de Enseñanza Aprendizaje (EVEA) y cuenta con una creciente base de usuarios. Para su desarrollo se utilizaron las siguientes herramientas PHP, Apache y la base de datos MySQL (9).

Capítulo 1: Fundamentación Teórica

Permite gestionar y desarrollar un proceso de enseñanza aprendizaje a través de internet utilizando herramientas de comunicación, trabajo colaborativo, gestión y seguimientos de los alumnos. Brinda la posibilidad de gestionar usuarios, materiales, recursos didácticos y herramientas para la evaluación (10).

Esta plataforma en su versión 1.9 permitió la creación y exportación de las actividades en formato IMS QTI en su versión 2.0 no permitiendo así importar dichas actividades.

Mientras que en su versión 2.0 esta plataforma no soporta la especificación IMS QTI.

Claroline

Claroline es un Sistema de Gestión de Cursos Basados en Web. Esta plataforma permite a los profesores crear y administrar sitios web de cursos desde un navegador (*Internet Explorer, Netscape, Firefox, Mozilla,...*). Esta herramienta permite publicar documentos en cualquier formato, componer ejercicios, consultar las estadísticas de frecuentación y el éxito en los ejercicios (11).

Las deficiencias presentes en esta plataforma con respecto a la especificación IMS QTI:

- No proporciona en la mayoría de los ejercicios mensajes de retroalimentación, excepto en los de selección simple.
- No brinda la posibilidad de mostrar los ejercicios de enlazar utilizando imágenes.
- No cuenta con todas las interacciones que soporta la especificación IMS QTI v2.0 necesaria para dicha plataforma (no cuenta con la interacción de ordenar).
- No existe la funcionalidad que permita exportar un conjunto de ejercicios, haciendo tedioso el trabajo de exportarlo uno a uno (12).

Sakai

Sakai es una plataforma educativa orientada a la creación y administración de cursos, es un ambiente de aprendizaje y colaboración que incluye una herramienta para el desarrollo de

Capítulo 1: Fundamentación Teórica

contenidos. Esta brinda la posibilidad de colocar presentaciones, ofrece plantilla para diseñar lecciones de manera secuencial, para crear asignaciones y exámenes.

Las principales dificultades presentes en esta plataforma con respecto a la especificación IMS QTI:

- Es insuficiente el uso de los mensajes de retroalimentación de acuerdo a las necesidades de la plataforma y a la especificación IMS QTI.
- No gestiona ni almacena los metadatos relacionados con los ejercicios creados.
- No tiene en cuenta las interacciones necesarias en la plataforma que soporta la especificación IMS QTI (13).

Con el análisis de estos reproductores se vieron reflejadas sus características y dentro de ellas sus deficiencias con respecto a la especificación IMS QTI. La presente investigación tiene como principal objetivo crear una API (Interfaz para la Programación de Aplicaciones), con el fin de que contribuya al desarrollo de un reproductor que visualice las actividades basadas en IMS QTI, con sus peculiaridades de imágenes, retroalimentación y evaluación venciendo así algunas de las deficiencias antes mencionadas en los reproductores examinados.

1.7 API (Interfaz para la Programación de Aplicaciones)

API o Interfaz de Programación de Aplicaciones: es una interfaz de comunicación entre componentes de software. Uno de los principales consiste en proporcionar un conjunto de funciones de uso general además de representar un método para conseguir abstracción en la programación.

Una API es un conjunto de funciones que permite al programador acceder a servicios de una aplicación a través del uso de un lenguaje de programación. Esta le ofrece al programador un cierto nivel de abstracción que enmascara la complejidad de acceso a un sistema o aplicación, proponiéndole un conjunto de funciones de las cuales solo se conocen los parámetros y los valores devueltos (14).

Capítulo 1: Fundamentación Teórica

Los programadores se benefician de las ventajas de una API haciendo uso de su funcionalidad, evitándose el trabajo de programar todo desde el principio. Una API presenta varias ventajas entre ellas (14).

- Una API hace más fácil el trabajo de desarrollo de un programa, ya que debe proveer todos los bloques para construirlo. El programador lo único que hace es poner todos los bloques juntos.
- Una API está diseñado especialmente para los programadores, ya que garantiza que todos los programas que utilizan API, tendrán interfaces similares. Asimismo, esto le facilita al usuario aprender la lógica de nuevos programas.
- Cuando se realiza una petición, el servidor llamará al API, brindando la ventaja de disponer de una mayor cantidad de servicios.

Para el desarrollo de la presente investigación se implementará una API con el objetivo de facilitarle el trabajo al desarrollador con la creación de un reproductor para la visualización de las actividades basadas en IMS QTI.

1.8 Metodologías de desarrollo de software

Las Metodologías de desarrollo de software surgen debido a la necesidad de utilizar una serie de procedimientos, técnicas, herramientas y soporte documental a la hora de desarrollar un producto de software.

Estas metodologías tienen como propósito guiar a los desarrolladores en la creación de un software. Los requisitos de un software a otro son tan variado y cambiante que ha dado lugar a que exista una gran variedad de metodologías para el desarrollo de software. Estas se podrían clasificar en dos grupos, Las metodologías orientadas al control de los procesos, estableciendo rigurosamente las actividades a desarrollar, herramientas a utilizar y notaciones que se usarán, estas metodologías son llamadas Metodologías pesadas o robustas. Las metodologías orientadas a la interacción con el cliente y el desarrollo incremental del software, mostrando versiones parcialmente funcionales del software al cliente en intervalos cortos de tiempo, para

Capítulo 1: Fundamentación Teórica

que pueda evaluar y sugerir cambios en el producto según se va desarrollando. Estas son llamadas Metodologías ligeras o ágiles (15).

Proceso unificado de desarrollo (RUP)

El Proceso Unificado de Desarrollo (RUP) es una de las metodologías más tradicionales, se centran en la definición detallada de los procesos y tareas a realizar, herramientas a utilizar, y requiere una extensa documentación. Este tipo de metodologías son más eficaces y necesarias cuando mayor es el proyecto que se pretende realizar respecto a tiempo y recursos que son necesarios emplear, donde una gran organización es requerida (15).

La Metodología RUP que divide el desarrollo en cuatro fases que definen su ciclo de vida:

- Inicio: El objetivo es determinar la visión del proyecto y definir lo que se desea realizar.
- Elaboración: Etapa en la que se determina la arquitectura óptima del proyecto.
- Construcción: Se obtiene la capacidad operacional inicial.
- Transición: Obtener el producto acabado y definido.

Esta metodología posee seis principios claves:

- Adaptación del proceso: El proceso debe adaptarse a las características de la organización para la que se está desarrollando el software.
- Balancear prioridades: Debe encontrarse un balance que satisfaga a todos los inversores del proyecto.
- Colaboración entre equipos: Debe haber una comunicación fluida para coordinar requerimientos, desarrollo, evaluaciones, planes y resultados.
- Demostrar valor iterativamente: Los proyectos se entregan, aunque sea de una forma interna, en etapas iteradas. En cada iteración se evaluará la calidad y estabilidad del producto y analizará la opinión y sugerencias de los inversores.
- Elevar el nivel de abstracción: Motivar el uso de conceptos reutilizables.

- Enfocarse en la calidad: La calidad del producto debe verificarse en cada aspecto de la producción.

Para el desarrollo de este trabajo se decidió utilizar esta metodología debido a sus características, ya que la misma genera una gran cantidad de informes, los cuales contienen informaciones necesarias relacionadas con el desarrollo del software.

1.9 Lenguajes de Programación

Los lenguajes de programación pueden dividirse en dos grupos, los lenguajes del lado del servidor y los lenguajes del lado del cliente, donde ambos reconocen la filosofía de la arquitectura Cliente/Servidor para las plataformas web. En el lado del cliente existen tecnologías como: *Java Script* además de *Cascading Style Sheets (CSS)* y *HyperText Markup Language (HTML)*, destinados a ofrecer dinamismo a la aplicación contribuyendo a la no recarga en servidores, lo que proporciona rapidez y optimización en los canales de comunicación. Aunque también existen técnicas de desarrollo para la web como lo es Ajax y XML.

Extensible Markup Language (XML)

XML (*eXtensible Markup Language*) es un lenguaje que permite jerarquizar y estructurar la información y describir los contenidos dentro del propio documento, así como la reutilización de partes del mismo. La información estructurada presenta varios contenidos: texto, imágenes, audio, etc. Es un conjunto de reglas que se usan para definir etiquetas semánticas las cuales organizan un documento en diferentes partes. Siendo así un meta lenguaje que define sintaxis para definir otros lenguajes etiquetados estructurados (16).

El XML permite crear etiquetas adaptadas a las necesidades y es especialmente estricto en cuanto a lo que está permitido y lo que no, todo documento debe cumplir dos condiciones: ser válido y estar bien formado.

Este lenguaje se usará en el presente trabajo para modelar las preguntas basadas en la especificación IMS QTI.

Procesador de hipertexto (PHP)

Capítulo 1: Fundamentación Teórica

PHP es un lenguaje de programación del lado del servidor, este permite acceder a los recursos que contenga el servidor como por ejemplo podría ser una base de datos. Puede ser utilizado para la creación de otro tipo de programas incluyendo aplicaciones con interfaz gráficas.

Algunas de las ventajas que presenta PHP son las siguientes (17):

- Es sencillo de aprender.
- Permite las técnicas de programación orientadas a objetos (POO).
- Es un software libre.
- Se puede obtener en la Web y su código está disponible bajo la Licencia Pública General.
- Cuenta con una excelente biblioteca de funciones que permite realizar cualquier labor (acceso a base de datos, encriptación y creación de PDF).
- Es un lenguaje multiplataforma.
- Capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad.
- Capacidad de expandir su potencial utilizando la enorme cantidad de módulos (llamados extensiones).

En la presente investigación se hará uso de este lenguaje con el objetivo de desarrollar una API que permita desarrollar un reproductor para visualizar las actividades basadas en la especificación IMS QTI.

1.10 Framework de desarrollo

Un framework es como una aplicación genérica incompleta y configurable a la que se le añaden las últimas piezas para construir una aplicación concreta. Entre los objetivos principales de un framework están: acelerar el proceso de desarrollo, reutilizar código ya existente y promover buenas prácticas de desarrollo como el uso de patrones (18).

Entonces se define como framework web, un conjunto de componentes que integran un diseño reutilizable que facilita y agiliza el desarrollo de sistemas web.

Symfony

Symfony es uno de los frameworks más populares para PHP en la actualidad. Este separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja. Además, automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación.

Es compatible con la mayoría de los gestores de bases de datos, como *MySQL*, *PostgreSQL*, *Oracle* y *SQL Server de Microsoft*. Se puede ejecutar tanto en plataformas *nix (*Unix*, *Linux*, etc.) como en plataformas Windows. Todo lo anterior hace de symfony un robusto framework que cada vez es más utilizado en el desarrollo de sistemas para la web (19).

En la presente investigación se hace uso de este framework php, ya que es el utilizado en el proyecto CRODA para la programación del lado del servidor, y por sus características antes expuestas.

1.11 Entorno Integrado de Desarrollo (IDE)

Un entorno de desarrollo integrado, llamado también IDE (por sus siglas en inglés *Integrated Development Environment*), es un programa informático compuesto por un conjunto de herramientas de programación. Es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica (GUI) (20).

Netbeans

El IDE Netbeans es un entorno de desarrollo, una herramienta para que los programadores puedan escribir, compilar, depurar y ejecutar programas. Es de código abierto y soporta lenguajes como *Java*, *PHP*, *Java Script* y *Ajax*, *Groovy* y *Grails*, y *C / C + +*. Es un producto gratuito sin restricciones de uso, disponible para Windows, Mac, Linux y Solaris. Brinda facilidades de integración con la subversión (21).

En la presente investigación por las características antes mencionada se selecciona Netbeans como entorno de desarrollo.

1.12 Sistema gestor de base de datos

Un sistema gestor de base de datos es un software que funciona como interfaz entre la base de datos, el usuario y la aplicación. El mismo está compuesto por un lenguaje de definición de datos, un lenguaje de manipulación de datos y un lenguaje de consulta. Estos permiten manejar de forma clara y ordenada la construcción y manipulación de una base de datos para diversas aplicaciones. Entre los empleados en el proyecto CRODA se encuentra PostgreSQL como base de datos relacional y eXist-db como base de datos nativa XML (22).

EXist-db

EXist-db es una base de datos de código abierto, sistema de gestión construido utilizando la tecnología XML. Permite el almacenamiento de los esquemas XML de acuerdo con el modelo especificado por IMS QTI. Facilita el trabajo con dichos esquemas con la utilización de Xquery como lenguaje basado en Xpath.

Debido a que las actividades de autoevaluación basadas en la especificación IMS QTI están modeladas en formato XML para el desarrollo de la presente investigación se hará uso de la base de datos eXist-db por todo lo anteriormente planteado (23).

1.13 Servidor web

Un servidor es un ordenador remoto que provee los datos solicitados por parte de los navegadores de otras computadoras. El mismo es un software que permite configurar una computadora como servidor para facilitar el acceso a la red y sus recursos. Los servidores almacenan información en forma de páginas web y a través del protocolo HTTP lo entregan a petición de los clientes en formato HTML.

Apache

Apache es un servidor web de software libre desarrollado por la Fundación de Software Apache cuyo objetivo es servir a los clientes web o navegadores que las solicitan. Es un proyecto de

uso gratuito, multiplataforma, muy robusto y que destaca por su seguridad y rendimiento. En él se definirá el comportamiento general del servidor de páginas web y además es el encargado de acceder a los distintos módulos que amplían la funcionalidad del servidor.

Este servidor Apache puede ser adaptado a diferentes entornos y necesidades, con los diferentes módulos de apoyo que proporciona, y con la API de programación de módulos, para el desarrollo de módulos específicos. Es extensible, gracias a ser modular se han desarrollado diversas extensiones entre las que destaca PHP, un lenguaje de programación del lado del servidor (24).

1.14 Herramienta CASE

La Herramienta CASE Visual Paradigm es multiplataforma, debido a esta característica es posible su utilización en el proyecto CRODA ya que el mismo se trabaja sobre el sistema operativo Linux. Esta herramienta modela el ciclo de desarrollo de software que propone RUP completo. Visual Paradigm permite crear los diagramas de clases necesarios, tiene integración con diferentes entornos de desarrollo (IDE), generación de código y documentación.

VP posee un diseño de interfaz gráfica, enfocado a facilitar la interacción y el trabajo del usuario. La herramienta también proporciona abundantes tutoriales, demostraciones interactivas y proyectos de UML.

Es en conclusión esta herramienta es la adecuada a las necesidades y características de la investigación en cuestión.

1.15 Lenguaje Unificado de Modelado (UML)

El lenguaje UML es un estándar diseñado para visualizar, especificar, construir y documentar software orientado a objetos. El mismo proporciona los planos de un sistema y puede ser más o menos detallado, en función de los elementos que sean relevantes en cada momento.

UML es esencial en la construcción de software para comunicar la estructura de un sistema complejo, especificar el comportamiento deseado del sistema, comprender mejor lo que se está construyendo y descubrir oportunidades de simplificación y reutilización.

En la presente investigación se hace uso del lenguaje unificado de modelado por las características antes expuestas (25).

1.16 Conclusiones

La especificación IMS QTI brinda la posibilidad de crear actividades individuales de autoevaluación en la herramienta de autor web CRODA, pero la herramienta no cuenta con un reproductor que permita visualizar las actividades basadas en dicha especificación. Los reproductores de actividades que se estudió en el presente capítulo presentaron deficiencias relacionadas con la especificación IMS QTI, específicamente con su versión 2.0. Mediante un estudio de las versiones de IMS QTI se determinó para esta investigación trabajar con la versión 2.0, debido a las dificultades que presentaban las otras versiones. Se definió que la metodología a utilizar es RUP así como Visual Paradigm 6.4 para el modelado de la propuesta. Como servidor web se propone emplear Apache 2.2 y como gestores de base de datos eXist-db para guardar los archivos XML 1.0. Para la implementación se propone utilizar el framework de desarrollo Symfony, el entorno integrador de desarrollo Netbeans y PHP como lenguaje de programación.

Capítulo 2: Desarrollo de la propuesta solución.

2.1 Introducción

En el presente capítulo se expone la propuesta de análisis y diseño para un reproductor de cuestionarios basados en IMS QTI en la herramienta de autor web CRODA, con el objetivo de crear las bases del reproductor y determinar las funcionalidades que debe brindar una API que sirva como mediador entre la lógica del negocio y las actividades almacenadas en la base de datos XML nativa eXist. Para la descripción de la solución se desarrolla el modelo de dominio, representando las clases y relaciones que lo conforman, con el objetivo de conceder un vocabulario común. Se desarrolla el flujo de trabajo Requerimientos según la metodología (RUP) propuesta en el capítulo anterior, donde se especifican los requisitos funcionales, se definen e identifican los actores y los casos de usos así como las relaciones que se establecen entre estos.

2.2 Modelo de dominio

Un modelo de dominio se utiliza frecuentemente como fuente de inspiración para el diseño de los objetos de software. El modelo de dominio muestra clases conceptuales significativas, es el artefacto más importante que se crea durante el análisis orientado a objetos. Un modelo del dominio es una representación visual de las clases conceptuales u objetos del mundo real en un dominio de interés (26).

Este modelo agrupa los conceptos de un dominio. Mecanismo fundamental para comprender el dominio del problema y para establecer conceptos comunes (27).

El modelo de dominio que a continuación se presenta constituye una representación gráfica de la estructura del XML para modelar una actividad (question).

2.2.1 Representación del Diagrama de Modelo de Dominio

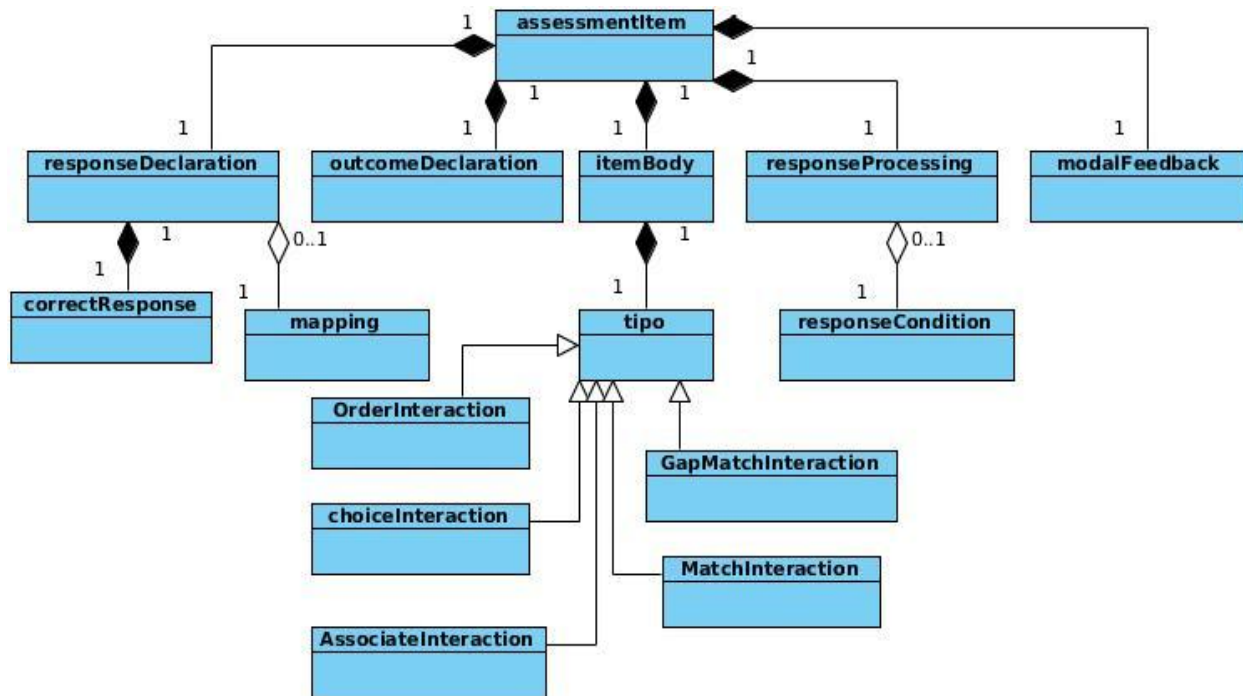


Ilustración 7: Modelo de dominio

2.2.2 Definición de las clases del Modelo de Dominio

- **AssessmentItem (evaluación):** Objeto que representa la estructura de datos Assessment.
- **ResponseDeclaration (declaración de respuesta):** Declaración de las respuestas de los usuarios.
- **OutcomeDeclaration (declaración de salidas):** Declara todas las evaluaciones de salida producidas para conseguir una evaluación.
- **ItemBody (cuerpo del ejercicio):** Objeto que representa la estructura de datos ítem.
- **Response Processing (procesamiento de respuestas):** Procesamiento y evaluación de las respuestas de los usuarios.

Capítulo 2: Desarrollo de la propuesta solución

- **ModalFeedback (retroalimentación):** Muestra información más detallada acerca de la respuesta que contiene la pregunta.
- **CorrectResponse (respuesta correcta):** Contiene la respuesta correcta de la actividad.
- **Mapping (mapeo):** Contiene el valor de la respuesta.
- **Type (tipo):** Posee la categoría de la actividad.
- **ResponseCondition (respuesta con condición):** Analiza la respuesta para darle una puntuación.

2.3 Requerimientos

“Los requerimientos son una especificación de lo que debe ser implementado. Estos son descripciones de cómo el sistema se debe comportar, de las propiedades y atributos del mismo. Deben ser una restricción del proceso de desarrollo del sistema” (28).

Requerimientos Funcionales

Los requerimientos funcionales son declaraciones de servicios que debe generar el sistema, es decir determina que hace el sistema. A continuación se presentan los requerimientos funcionales identificados durante el proceso de levantamiento de requisitos correspondiente al flujo de trabajo Requerimientos de la metodología RUP.

En este trabajo se definieron los siguientes requisitos funcionales:

ChoiceInteraction

RF1-El sistema debe permitir mostrar los elementos del enunciado de la pregunta ChoiceInteraction.

RF2-El sistema debe permitir mostrar las respuestas que se muestran como posibles opciones a seleccionar de la pregunta ChoiceInteraction.

RF3-El sistema debe permitir mostrar el valor que tiene la opción dentro de la pregunta ChoiceInteraction.

Capítulo 2: Desarrollo de la propuesta solución

RF4-El sistema debe permitir mostrar el resultado de la respuesta de la pregunta ChoiceInteraction.

RF5- El sistema debe permitir mostrar la retroalimentación de la pregunta ChoiceInteraction.

RF6- El sistema debe permitir mostrar la categoría de la pregunta ChoiceInteraction.

RF7-El sistema debe permitir mostrar la opción enviar respuesta de la pregunta ChoiceInteraction.

AssociateInteraction

RF8-El sistema debe permitir mostrar los elementos del enunciado de la pregunta AssociateInteraction.

RF9-El sistema debe permitir mostrar los elementos a relacionar de la pregunta AssociateInteraction.

RF10-El sistema debe permitir mostrar el valor que tiene la relación dentro de la pregunta AssociateInteraction.

RF11-El sistema debe permitir mostrar el resultado de la respuesta de la pregunta AssociateInteraction.

RF12-El sistema debe permitir mostrar la retroalimentación de la pregunta AssociateInteraction.

RF13- El sistema debe permitir mostrar la categoría de la pregunta AssociateInteraction.

RF14-El sistema debe permitir mostrar la opción enviar respuesta de la pregunta AssociateInteraction.

MatchInteraction

RF15-El sistema debe permitir mostrar los elementos del enunciado de la pregunta MatchInteraction.

RF16-El sistema debe permitir mostrar los pares de elementos a relacionar de la pregunta MatchInteraction.

Capítulo 2: Desarrollo de la propuesta solución

RF17-El sistema debe permitir mostrar el valor que tiene las relaciones dentro de la pregunta MatchInteraction.

RF18-El sistema debe permitir mostrar el resultado de la respuesta de la pregunta MatchInteraction.

RF19- El sistema debe permitir mostrar la retroalimentación de la pregunta MatchInteraction.

RF20- El sistema debe permitir mostrar la categoría de la pregunta MatchInteraction.

RF21-El sistema debe permitir mostrar la opción enviar respuesta de la pregunta MatchInteraction.

OrderInteraction

RF22-El sistema debe permitir mostrar los elementos del enunciado de la pregunta OrderInteraction.

RF23-El sistema debe permitir mostrar un conjunto de soluciones que deberán ser organizadas en la pregunta OrderInteraction.

RF24-El sistema debe permitir mostrar el valor que tiene el ordenamiento dentro de la pregunta OrderInteraction.

RF25-El sistema debe permitir mostrar el resultado de la respuesta de la pregunta OrderInteraction.

RF26- El sistema debe permitir mostrar la retroalimentación de la pregunta OrderInteraction.

RF27- El sistema debe permitir mostrar la categoría de la pregunta OrderInteraction.

RF28- El sistema debe permitir mostrar la opción enviar respuesta de la pregunta OrderInteraction.

GapMatchInteraction

Capítulo 2: Desarrollo de la propuesta solución

RF29-El sistema debe permitir mostrar los elementos del enunciado de la pregunta GapMatchInteraction.

RF30-El sistema debe permitir mostrar un conjunto de espacios en blanco dentro del enunciado de la pregunta GapMatchInteraction.

RF31-El sistema debe permitir mostrar el valor que tiene llenar cada espacio en blanco dentro de la pregunta GapMatchInteraction.

RF32-El sistema debe permitir mostrar el resultado de la respuesta de la pregunta GapMatchInteraction.

RF33-El sistema debe permitir mostrar la retroalimentación de la pregunta GapMatchInteraction.

RF34- El sistema debe permitir mostrar la categoría de la pregunta GapMatchInteraction.

RF35- El sistema debe permitir mostrar la opción enviar respuesta de la pregunta GapMatchInteraction.

Requerimientos no Funcionales

Los requisitos no funcionales son propiedades o cualidades que el producto debe tener. Deben pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable (29).

En este trabajo se definieron los siguientes requisitos no funcionales a tener en cuenta en una API que contribuya al desarrollo de un reproductor para la visualización de actividades de evaluación basadas en la especificación IMS QTI en CRODA:

RNF1. Usabilidad

RNF1.1. El API podrá ser usado por cualquier persona que posea conocimientos en el trabajo de desarrollo de software con tecnologías web.

RNF1.2. Debe presentar unos servicios fáciles de utilizar por los demás desarrolladores.

Capítulo 2: Desarrollo de la propuesta solución

RNF1.3. En la interacción con la API mostrará mensajes que sean de entendimiento común con los desarrolladores, para de esta forma elevar la comunicación con el mismo.

RNF2. Fiabilidad

RNF2.1. Las salidas que ofrece la API como resultado del manejo y procesamiento de la información estarán adecuadamente validadas durante los diferentes hitos de prueba, siendo estas fidedignas y con total exactitud a los valores esperados.

RNF3. Software

El sistema que porte la API desarrollada debe tener instalado:

RNF3.1. EXist-db versión 1.4 como base de datos de XML nativo.

RNF3.2. PHP 5.3 para poder soportar el lenguaje con el cual se encuentra implementado dicho producto.

2.4 Modelo de casos de uso del sistema

Los diagramas de casos de uso describen las relaciones y las dependencias entre un grupo de casos de uso y los actores que participan en el proceso. Un caso de uso es iniciado por un actor. A partir de ese momento, ese actor, junto con otros actores, intercambia datos o control con el sistema, participando de ese caso de uso.

Los diagramas de casos de uso del sistema se caracterizan por estar expresados desde el punto de vista del actor estos se documentan con un texto informal y describen tanto lo que hace el actor como lo que hace el sistema cuando interactúa con él, aunque el énfasis está puesto en la interacción. Estos son iniciados por un único actor.

2.4.1 Actores del sistema

Un actor es una agrupación uniforme de personas, sistemas o máquinas que interactúan con el sistema que se está construyendo los mismos va a ser externos al sistema (30). En la siguiente tabla se representa los actores del sistema.

Capítulo 2: Desarrollo de la propuesta solución

Tabla 1 : Actores del sistema

Actores	Descripción
Diseñador de actividades	Es el usuario responsable de definir cuáles actividades de autoevaluación implementar y elegir los métodos para las mismas, estas servirán de apoyo al proceso de autoevaluación y medir los conocimientos adquiridos por los estudiantes.

2.4.2 Casos de uso del sistema

Un caso de uso es una secuencia de acciones que el sistema lleva a cabo con el objetivo de ofrecer un resultado de valor para un actor, es decir, un caso de uso proporciona un resultado observable para el usuario (31).

Tabla 2: Casos de uso (CU) del sistema

CU-1	Reproducir cuestionarios de selección
Actor	Diseñador de actividades
Descripción	Permite al diseñador de actividades, visualizar las actividades de selección de una o varias repuestas.
Referencia	RF1

Tabla 3: Casos de uso del sistema

CU-2	Reproducir cuestionarios de asociación
Actor	Diseñador de actividades
Descripción	Permite al diseñador de actividades, visualizar las actividades de asociación.
Referencia	RF8

Capítulo 2: Desarrollo de la propuesta solución

Tabla 4: Casos de uso del sistema

CU-3	Reproducir cuestionarios de ordenamiento
Actor	Diseñador de actividades
Descripción	Permite al diseñador de actividades, visualizar las actividades de ordenamiento.
Referencia	RF15

Tabla 5: Casos de uso del sistema

CU-4	Reproducir cuestionarios de enlazar columnas.
Actor	Diseñador de actividades
Descripción	Permite al diseñador de actividades, visualizar las actividades enlazar columnas.
Referencia	RF22

Tabla 6: Casos de uso del sistema

CU-5	Reproducir cuestionarios de llenar espacios
Actor	Diseñador de actividades
Descripción	Permite al diseñador de actividades, visualizar las actividades llenar espacios en blanco.
Referencia	RF29

2.4.3 Diagrama de Casos de Uso del sistema

A continuación se presenta el diagrama de caso de uso del sistema en la Ilustración 8.

Capítulo 2: Desarrollo de la propuesta solución

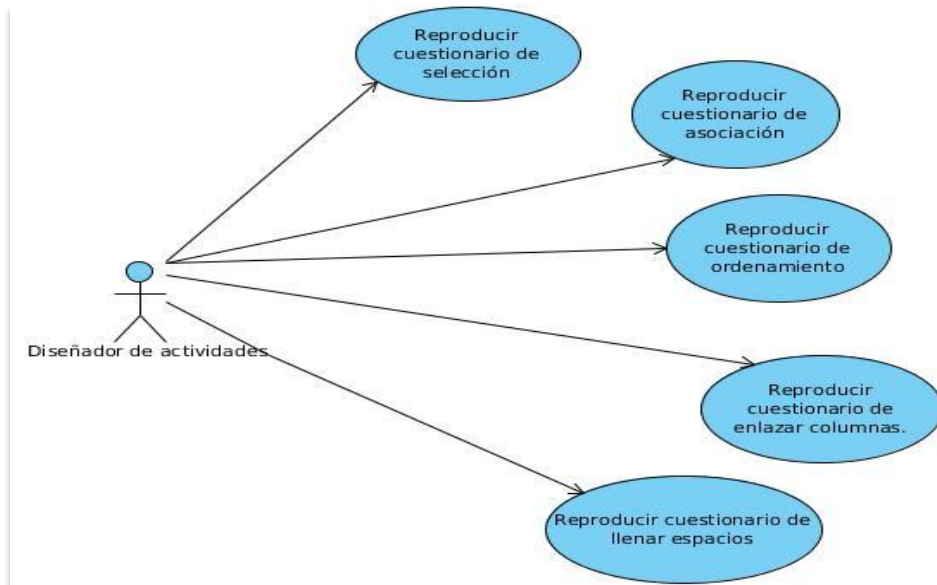


Ilustración 9: Diagrama de Casos de Uso del sistema

2.5 Descripción textual de los casos de uso del sistema

Las descripciones textuales de los casos de uso (CU) define el flujo de eventos a realizar en el sistema, el actor que lo inicializa, las precondiciones y poscondiciones del caso de uso, así como los flujos alternos y las referencias a los requisitos funcionales que los satisfacen. Estas son fundamentales para obtener una mejor comprensión del sistema dentro del equipo de desarrollo.

A continuación, se representa la descripción textual de los CU del sistema definido en la presente investigación.

Tabla 4: CU 1 Reproducir cuestionario de selección.

Caso de Uso:	Reproducir cuestionario de selección.
Actores:	Diseñador de actividades.
Resumen:	El CU se inicia cuando el diseñador de actividades selecciona la opción que le permite visualizar un cuestionario de selección.
Precondiciones:	Debe haber generado el entorno de trabajo del usuario

Capítulo 2: Desarrollo de la propuesta solución

	autenticado.
Referencia:	RF1.
Prioridad:	Crítica
Flujo Normal de Eventos	
Acción del Actor	Respuestas del sistema
1. El diseñador de actividades selecciona la opción visualizar cuestionario.	2. Brinda la posibilidad de realizar las acciones. •Ver las instrucciones del cuestionario. •Seleccionar la o las posibles respuestas de las opciones del cuestionario. •Permite enviar las posibles respuestas. •Permite cancelar la operación en cualquier momento.
3. Selecciona la o las posibles opciones para darle respuesta al cuestionario.	
4. Selecciona la opción de enviar.	5. Validar los datos
	6. Muestra resultado.
	7. Muestra retroalimentación.
Flujos Alternos	
Acción del Actor	Respuestas del sistema
1. El actor selecciona la opción cancelar.	2. El sistema cierra el entorno de reproducción del cuestionario y vuelve el entorno de diseño de cuestionario.

Tabla 5: CU 2 Reproducir cuestionario de asociación.

Caso de Uso:	Reproducir cuestionario de asociación.
---------------------	--

Capítulo 2: Desarrollo de la propuesta solución

Actores:	Diseñador de actividades.
Resumen:	El CU se inicia cuando el diseñador de actividades selecciona la opción que le permite visualizar un cuestionario de asociaciones en parejas.
Precondiciones:	Debe haber generado el entorno de trabajo del usuario autenticado.
Referencia:	RF8.
Prioridad:	Crítica.

Flujo Normal de Eventos

Acción del Actor	Respuestas del sistema
1. El diseñador de actividades selecciona la opción visualizar cuestionario.	2. Brinda la posibilidad de realizar las acciones. <ul style="list-style-type: none"> • Ver las instrucciones del cuestionario. • Asociar los elementos de cada fila del cuestionario. • Permite enviar las posibles respuestas. • Permite cancelar la operación en cualquier momento.
3. Asociar los elementos de cada fila del cuestionario para darle respuesta al cuestionario.	
4. Selecciona la opción de enviar.	5. Validar los datos.
	6. Muestra resultado.
	7. Muestra retroalimentación.

Flujos Alternos

Acción del Actor	Respuestas del sistema
------------------	------------------------

Capítulo 2: Desarrollo de la propuesta solución

1. El actor selecciona la opción cancelar.	2. El sistema cierra el entorno de reproducción del cuestionario y vuelve el entorno de diseño de cuestionario.
--	---

Tabla 6: CU 3 Reproducir cuestionario de ordenamiento.

Caso de Uso:	Reproducir cuestionario de ordenamiento.
Actores:	Diseñador de actividades.
Resumen:	El CU se inicia cuando el diseñador de actividades selecciona la opción que le permite visualizar un cuestionario de ordenamiento.
Precondiciones:	Debe haber generado el entorno de trabajo del usuario autenticado.
Referencia:	RF15.
Prioridad:	Crítica.
Flujo Normal de Eventos	
Acción del Actor	Respuestas del sistema
1. El diseñador de actividades selecciona la opción visualizar cuestionario.	2. Brinda la posibilidad de realizar las acciones. <ul style="list-style-type: none"> Ver las instrucciones del cuestionario. Ordenar los elementos del cuestionario. Permite enviar las posibles respuestas. Permite cancelar la operación en cualquier momento.
3. Ordenar los elementos para darle	

Capítulo 2: Desarrollo de la propuesta solución

respuesta al cuestionario.	
4. Selecciona la opción enviar.	5. Validar los datos.
	6. Muestra resultado.
	7. Muestra retroalimentación.
Flujos Alternos	
Acción del Actor	Respuestas del sistema
1. El actor selecciona la opción cancelar.	2. El sistema cierra el entorno de reproducción del cuestionario y vuelve el entorno de diseño de cuestionario.

Tabla 7: CU 4: Reproducir cuestionario de enlazar columnas.

Caso de Uso:	Reproducir cuestionario de enlazar columnas.
Actores:	Diseñador de actividades.
Resumen:	El CU se inicia cuando el diseñador de actividades selecciona la opción que le permite visualizar un cuestionario de asociaciones impares.
Precondiciones:	Debe haber generado el entorno de trabajo del usuario autenticado.
Referencia:	RF22.
Prioridad:	Crítica.
Flujo Normal de Eventos	
Acción del Actor	Respuestas del sistema
1. El diseñador de actividades selecciona la opción visualizar cuestionario.	2. Brinda la posibilidad de realizar las acciones. <ul style="list-style-type: none"> • Ver las instrucciones del cuestionario. • Enlazar los elementos de las filas del

Capítulo 2: Desarrollo de la propuesta solución

	<p>cuestionario.</p> <ul style="list-style-type: none"> • Permite enviar las posibles respuestas. • Permite cancelar la operación en cualquier momento.
3. Enlazar los elementos de cada fila del cuestionario para darle respuesta al cuestionario.	
4. Selecciona la opción enviar.	5. Validar los datos.
	6. Muestra resultado.
	7. Muestra retroalimentación.
Flujos Alternos	
Acción del Actor	Respuestas del sistema
1. El actor selecciona la opción cancelar.	2. El sistema cierra el entorno de reproducción del cuestionario y vuelve el entorno de diseño de cuestionario.

Tabla 8: CU 5 Reproducir cuestionario de llenar espacios.

Caso de Uso:	Reproducir cuestionario de llenar espacios.
Actores:	Diseñador de actividades.
Resumen:	El CU se inicia cuando el diseñador de actividades selecciona la opción que le permite visualizar un cuestionario de llenar espacios.
Precondiciones:	Debe haber generado el entorno de trabajo del usuario autenticado.
Referencia:	RF29.

Capítulo 2: Desarrollo de la propuesta solución

Prioridad:	Crítica.
Flujo Normal de Eventos	
Acción del Actor	Respuestas del sistema
1. El diseñador de actividades selecciona la opción visualizar cuestionario.	2. Brinda la posibilidad de realizar las acciones. <ul style="list-style-type: none">• Ver las instrucciones del cuestionario.• Llenar los espacios en blancos que se encuentran en el texto del cuestionario.• Permite enviar las posibles respuestas.• Permite cancelar la operación en cualquier momento.
3. Llenar los espacios en blancos que se encuentran en el texto del cuestionario para darle respuesta al cuestionario.	
4. Selecciona la opción enviar.	5. Validar los datos.
	6. Muestra resultado.
	7. Muestra retroalimentación.
Flujos Alternos	
Acción del Actor	Respuestas del sistema
1. El actor selecciona la opción cancelar.	2. El sistema cierra el entorno de reproducción del cuestionario y vuelve el entorno de diseño de cuestionario.

2.6 Modelo del análisis

Partiendo de los requisitos funcionales definidos anteriormente se realizará un esbozo preliminar del sistema a realizar en el Modelo de Análisis. En este flujo de trabajo se tiene

Capítulo 2: Desarrollo de la propuesta solución

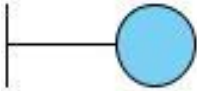


como objetivo adquirir una comprensión más específica de los requisitos, estos se refinarán, serán estructurados y profundizados y se proporcionará una visión general del sistema.

En el análisis, el estudio de los requisitos brindará una mejor comprensión antes de entrar al diseño del sistema garantizando una arquitectura robusta, eficaz y eficiente, resistente a cambios.

2.6.1 Diagrama de clases del análisis

En la elaboración del diagrama de clases del análisis se debe identificar las clases del análisis, las cuáles son la representación de los roles de los elementos del modelo. Este diagrama se representa mediante tres estereotipos que a continuación se presentan:

Tabla 9 : Estereotipos de las clases del análisis

Estereotipos	Descripción
	<u>Clase Interfaz:</u> Se encargan de la modelación de toda la interacción que puede existir entre los actores y el sistema; constituyen las fronteras del mismo.
	<u>Clase Controladora:</u> Representan la coordinación, secuenciación, transacciones y a veces la lógica del negocio; se emplean a menudo para encapsular el control referido a un CU.
	<u>Clase Entidad:</u> Representa la información de larga duración y a menudo persistente que se maneja en el sistema.

Capítulo 2: Desarrollo de la propuesta solución

A continuación se presentan el diagrama de clases del análisis del CU Reproducir cuestionario de selección en la Ilustración 10 el resto de los diagramas se encuentran en el **Anexo 1** desde la (Ilustración 22) hasta la figura (Ilustración 25).



Ilustración 11: Diagrama de clases del análisis CU1 Reproducir cuestionarios de selección.

En el presente trabajo se realizó un diagrama de clases del análisis para cada caso de uso identificado, permitiendo un mejor entendimiento y seguimiento de los requisitos involucrados en cada uno de ellos.

2.6.2 Diagramas de interacción

Los diagramas de interacción del análisis se utilizan para modelar los aspectos dinámicos de un sistema estos se representan a través de diagramas de colaboración o de secuencia. Los diagramas de secuencia destacan el orden temporal de los mensajes y los de colaboración destacan la organización estructural de los objetos que envían y reciben mensajes.

Un diagrama de interacción consiste en un conjunto de objetos y sus relaciones, incluyendo los mensajes que se pueden enviar entre ellos (32).

Para la elaboración de los diagramas se utilizaron los estereotipos representados en la **Tabla 12** Interfaz, Controladora y Entidad. A continuación se representa el diagrama de colaboración correspondiente al CU Reproducir cuestionario de selección en la **Ilustración 12** el resto de los diagramas de colaboración y de secuencia se encuentran en el Anexo 2 y Anexo3 desde la (Ilustración 26) hasta la (Ilustración 34).

Capítulo 2: Desarrollo de la propuesta solución

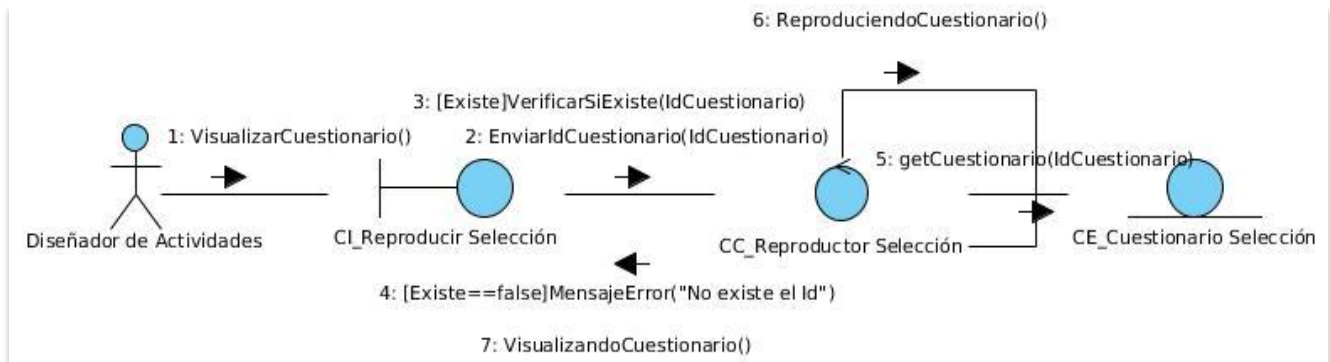


Ilustración 13: Diagrama de Colaboración CU1 Reproducir cuestionario de selección.

2.7 Diseño

El diseño es técnicamente la parte central de la ingeniería del software. Durante el diseño se desarrollan, revisan y se documentan los refinamientos progresivos de las estructuras de datos, de la estructura del programa y de los detalles procedimentales. El diseño da como resultado representaciones cuya calidad puede ser evaluada (33).




El modelo de diseño es uno de los artefactos fundamentales del flujo de trabajo de análisis y diseño. En él se representan los requisitos funcionales obtenidos en la etapa de análisis en un lenguaje técnico lo más cercano posible al lenguaje programación. El diseño debe proporcionar una idea completa de lo que es el software.

2.7.1 Diagramas de clases del diseño

Los estereotipos utilizados para la elaboración de los diagramas de clases del diseño son Server Page, Client Page y Form estos se relacionan entre sí, a continuación se expresan las descripciones de cada uno de los estereotipos y sus relaciones.

Capítulo 2: Desarrollo de la propuesta solución

Tabla 10: Estereotipos web

Estereotipos	Iconos	Descripción
<<Client Page>>		Una instancia de Página Cliente es una página web, con formato HTML. Mezcla de datos, presentación y lógica. Son interpretadas por el navegador.
<<Form>>		Colección de elementos de entrada que son parte de una página cliente.
<<Server Page>>		Representa la página web que tiene código que se ejecuta en el servidor.

A continuación se representa los diagramas de Clases del Diseño con los estereotipos web representados en la Tabla 13 correspondiente al CU Reproducir cuestionario de selección en la **Ilustración 14** el resto de los diagramas de clases del diseño se encuentran en el **Anexo 4** desde la (Ilustración 35) hasta la (Ilustración 38).

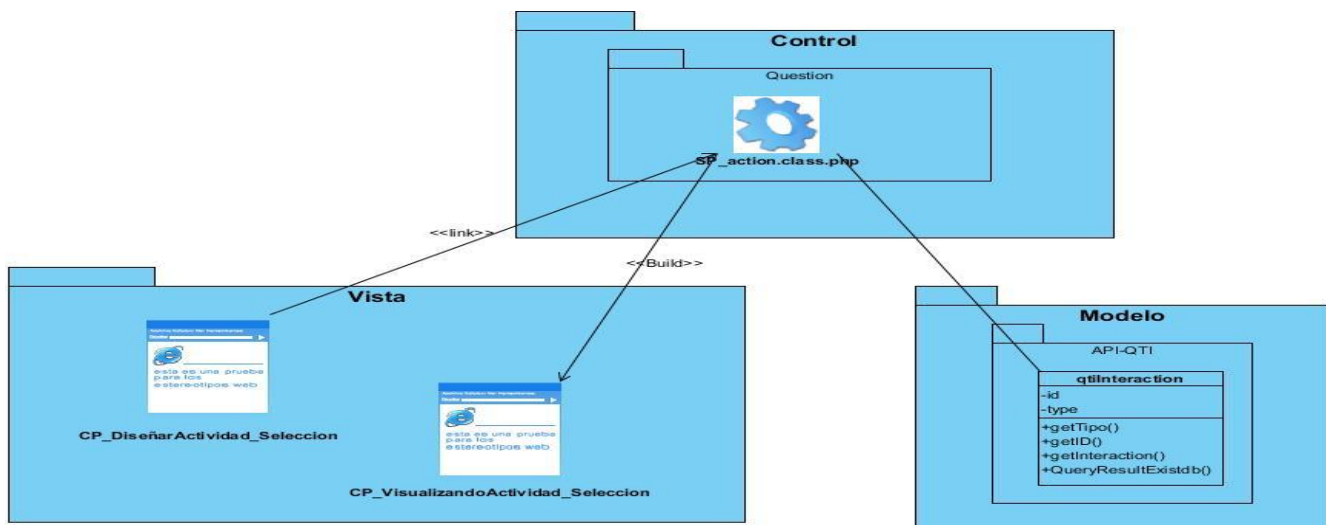


Ilustración 15: Diagrama de clases del diseño CU1 Reproducir cuestionario de selección.

2.7.2 Modelos de Datos

En la presente investigación no se trabaja con una base de datos relacional, se utiliza una base de datos nativa XML denominada eXist-db, la cual se encarga de almacenar cada una de las actividades diseñadas porque al estar basadas en IMS QTI se encuentran modeladas en sus respectivos esquemas XML.

A continuación se representa el esquema XML de la actividad soportada por la interacción ChoiceInteraction en la **Ilustración 16**.

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- This example adapted from the PET Handbook, copyright University of Cambridge ESOL Examinations -->
<assessmentItem xmlns="http://www.imsglobal.org/xsd/imsqti_v2p0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.imsglobal.org/xsd/imsqti_v2p0_imsqti_v2p0.xsd"
  identifier="choice" title="Unattended Luggage" adaptive="false" timeDependent="false">
  <responseDeclaration identifier="RESPONSE" cardinality="single" baseType="identifier">
    <correctResponse>
      <value>ChoiceA</value>
    </correctResponse>
  </responseDeclaration>
  <outcomeDeclaration identifier="SCORE" cardinality="single" baseType="integer">
    <defaultValue>
      <value>0</value>
    </defaultValue>
  </outcomeDeclaration>
  <itemBody>
    <p>Look at the text in the picture.</p>
    <p>
      
    </p>
    <choiceInteraction responseIdentifier="RESPONSE" shuffle="false" maxChoices="1">
      <prompt>What does it say?</prompt>
      <simpleChoice identifier="ChoiceA">You must stay with your luggage at all times.</simpleChoice>
      <simpleChoice identifier="ChoiceB">Do not let someone else look after your luggage.</simpleChoice>
      <simpleChoice identifier="ChoiceC">Remember your luggage when you leave.</simpleChoice>
    </choiceInteraction>
  </itemBody>
  <responseProcessing
    template="http://www.imsglobal.org/question/qti_v2p0/rptemplates/match_correct"/>
</assessmentItem>
```

Ilustración 17: Esquema XML de la actividad soportada por la interacción ChoiceInteraction

2.8 API-QTI

En el flujo de trabajo Análisis y Diseño se modelaron y definieron componentes y funcionalidades que servirán de bases para la posterior implementación de un reproductor de

Capítulo 2: Desarrollo de la propuesta solución

actividades basadas en IMS QTI. Sirve además, dicho flujo de trabajo para determinar cuáles funcionalidades serán necesarias implementar, en una API que actúe de intermediaria entre las actividades previamente diseñadas y almacenadas en la base de datos nativa eXist-db y el componente encargado de proyectarlas.

Esta API tiene como objetivo facilitarle el trabajo al desarrollador, proporcionándole un conjunto de funcionalidades necesarias para la creación de actividades basadas en IMS QTI y luego estas sean visualizadas o proyectadas en un reproductor.

A continuación en la **Ilustración 18** se definen las clases y relaciones entre ellas para el desarrollo de una API:

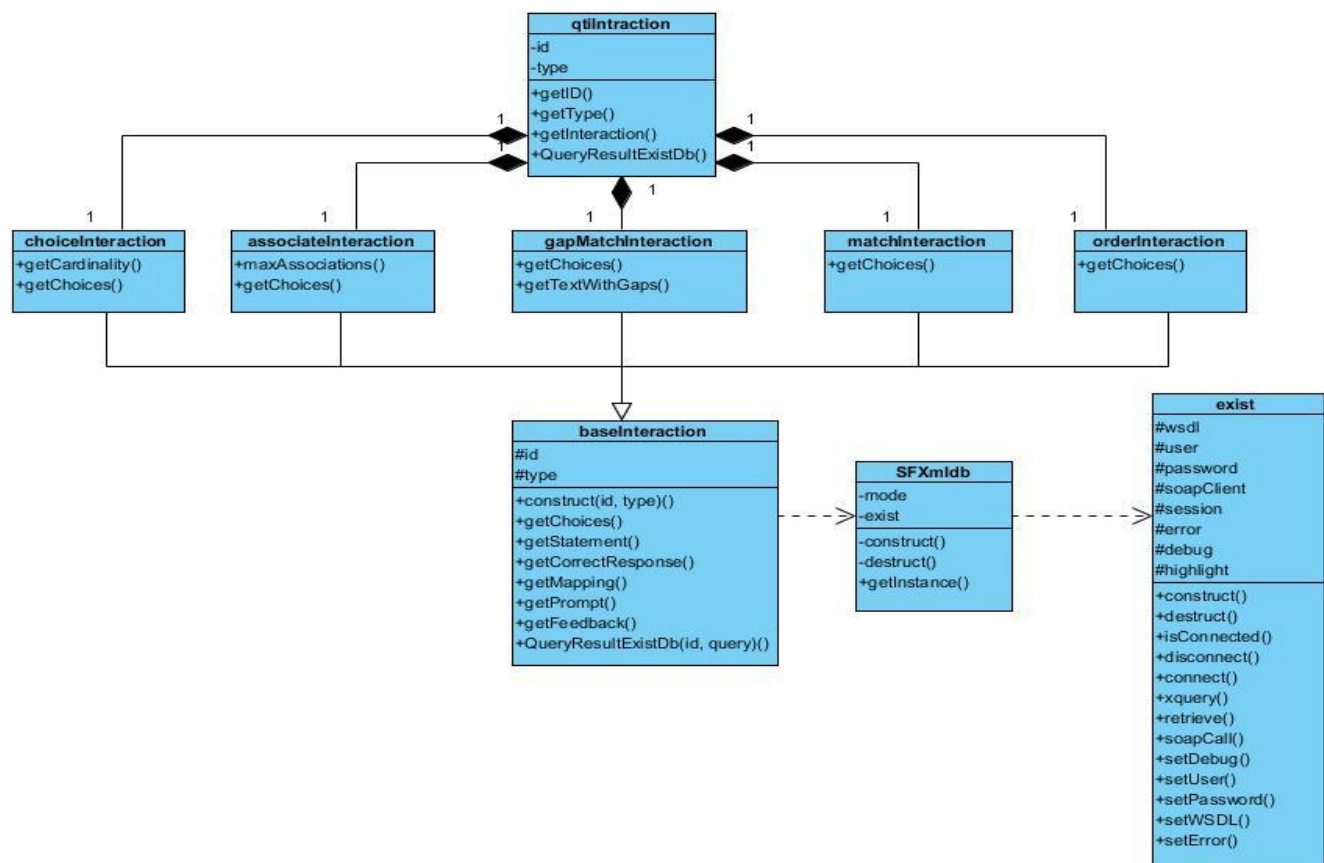


Ilustración 19: Diagrama de clases para el desarrollo de una API

Capítulo 2: Desarrollo de la propuesta solución

.qtInteraction: clase principal que se encarga de controlar las salidas en la API como: el tipo de actividad y la actividad que se decida obtener según su identificador.

.baseInteraction: es una clase abstracta en la cual se definen todas las funcionalidades comunes, pero que tienen sus peculiaridades en aquellas clases que se encargan de obtener los elementos según el tipo de actividad, como son: `choiceInteraction`, `associateInteraction`, `gapMatchInteraction`, `matchInteraction` y `orderInteraction`.

.choiceInteraction: se especializa en la obtención de los elementos correspondientes a las actividades de selección simple o múltiple.

.associateInteraction: se especializa en la obtención de los elementos correspondientes a las actividades de asociación.

.gapMatchInteraction: se especializa en la obtención de los elementos correspondientes a las actividades en las cuales dado un texto se requiera llenar los espacios en blanco o huecos.

.matchInteraction: se especializa en la obtención de los elementos correspondientes a las actividades en las que se enlazan elementos que se encuentran organizados en dos columnas.

.orderInteraction: se especializa en la obtención de los elementos correspondientes a las actividades en las que se requiere ordenar elementos.

2.9 Diagrama de paquetes

En la Ilustración 14 se observa un diagrama de paquetes general que esboza los dos grandes conjuntos lógicos en los que se divide el sistema: la parte relacionada con la librería ExtJs 3.4.0 y la parte que se asocia al framework Symfony en su versión 1.4.

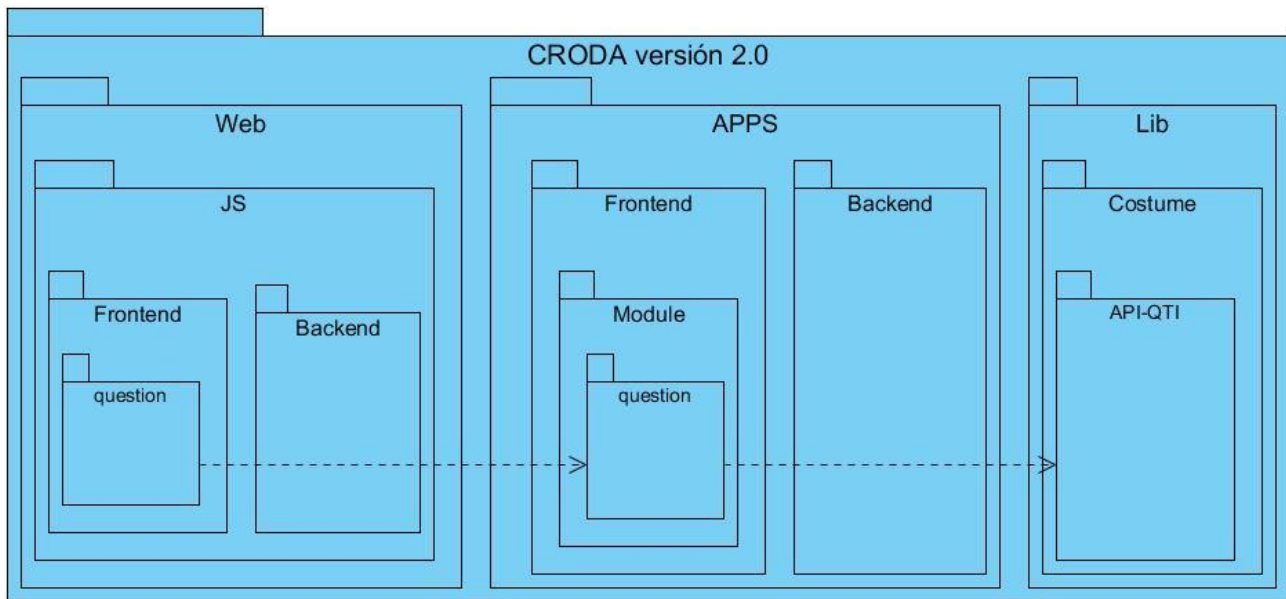


Ilustración 20: Diagrama de paquetes general del sistema.

Un proyecto basado en el framework Symfony genera una serie de paquetes de los cuales web, apps y lib, son los que se utilizan específicamente en la presente investigación. En el caso de web es un paquete que contiene todo lo relacionado con la librería ExtJS y demás aspectos relacionados con la presentación como imágenes y estilos. El paquete apps contiene la organización en módulos del proyecto y desde ellos se controla toda la lógica del negocio según lo requerido desde la vista. En el paquete lib se encuentra la API – QTI quien ofrece al módulo question todo lo relacionado con el acceso para la reproducción de una actividad basada en IMS QTI almacenada en la base de datos nativa eXist-db.

2.10 Diagrama de componentes

Durante el diseño de una actividad basada en IMS QTI el componente principal al que se hace referencia en la vista del proyecto es modquestion.js, desde el cual se inician todos los componentes necesarios para llevar a cabo la creación de este tipo de actividades, para ello es necesario la comunicación asincrónica entre los componentes js y el action.class.php ubicado en el módulo question. En el caso que se desee reproducir una actividad en creación o previamente creada se accederá desde el módulo question al API – QTI a través de la clase principal de este qtilInteracton.php para acceder a todos los elementos de la actividad.

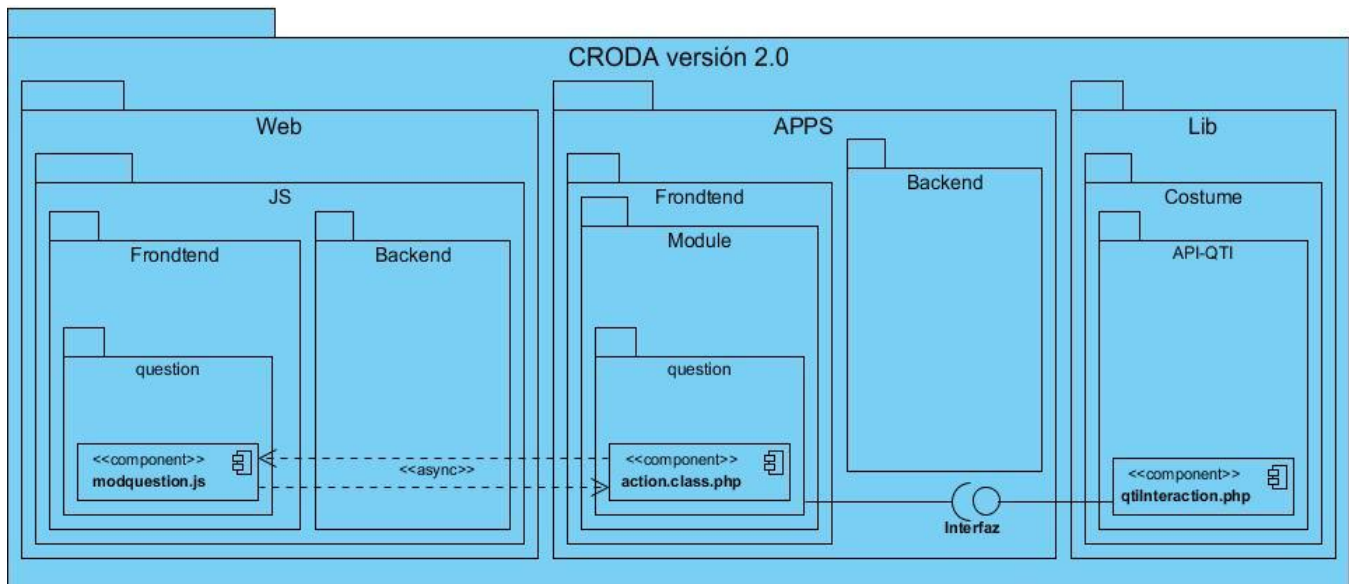


Ilustración 21: Diagrama de componentes del sistema.

2.11 Aplicaciones de patrones de diseño

Un patrón de diseño es una descripción del problema y la esencia de su solución, que se puede reutilizar en casos distintos. Es una solución adecuada a un problema común. Existen diferentes categorías de patrones, están los de creación que implica el proceso de instanciar objetos, los estructurales para la composición de objetos y los de comportamiento que muestra cómo se comunican los objetos, cooperan y distribuyen las responsabilidades para lograr sus objetivos (34).

Entre los patrones de diseños utilizados en la aplicación se puede hacer alusión a:

Patrones estructurales **Proxy** el cual proporciona un representante o sustituto de otro objeto para controlar el acceso a este y el **Adapter** que convierte la interfaz de una clase en otra que espera el cliente. El Adapter permite trabajar juntas a clases que de otra forma no podrían por tener interfaces incompatibles.

Patrón de comportamiento **Strategy** que define una familia de algoritmos, encapsula cada uno de ellos, y los hace intercambiables. Permite que un algoritmo varíe independientemente de los clientes que los usan.

Capítulo 2: Desarrollo de la propuesta solución

Patrones de asignación de responsabilidades (Grasp): Alta cohesión, Bajo acoplamiento, Creador, Experto, Controlador. Estos patrones se pueden evidenciar en los diagramas de interacción el cual es uno de los artefactos que se generan en la elaboración del análisis y diseño del software.

Con la elaboración de los diagramas de interacción se puede ver claramente cómo actúan los patrones de **Bajo acoplamiento** y de **Alta cohesión** ya que en él se determina el tipo de relación entre las clases y la responsabilidad que va a tener cada una de ellas.

El patrón **Experto** se refleja al definir una clase, la cual cuenta con la información necesaria para cumplir con sus responsabilidades.

El patrón **Creador** su principal objetivo es encontrar un creador que se conecte con el objeto producido en cualquier evento.

El patrón **Controlador** define quién debería encargarse de atender un evento del sistema.

2.12 Conclusiones

En este capítulo se expresaron las principales clases del dominio para una mayor comprensión del negocio dando paso a la definición de los requisitos funcionales y no funcionales que debe cumplir la aplicación. Estos elementos permitieron la creación de los actores y casos de uso representados en el diagrama de casos de uso del sistema, así como la descripción de cada uno de ellos. Se llevó a cabo el flujo de trabajo Análisis y Diseño generándose sus artefactos principales como, los diagramas de clases del análisis, diagrama de colaboración y diagrama de clases del diseño utilizando estereotipos web. Se definió como modelo de datos un esquema XML de las actividades de autoevaluación el cual refleja elementos contenidos en cada una de ellas, soportadas por sus correspondientes interacciones, luego se procedió con la representación de las clases determinadas para la creación de una API, las relaciones entre ellas y su descripción. Como parte de la implementación se elaboraron los diagramas de paquetes y componente dando una representación más detallada del sistema.

Capítulo 3: Validación de la solución propuesta

3.1 Introducción

En el presente capítulo se realizarán las validaciones correspondientes a los artefactos generados en el capítulo anterior utilizando Métricas de la Calidad de Especificación de los Requisitos y Casos de Uso.

3.2 Métodos de validación

Uno de los procesos más importantes que se llevan a cabo dentro del ciclo de vida un software es el proceso de Verificación y Validación (V&V). Este es un conjunto de proceso de comprobación y análisis que aseguran que el software que se desarrolla está acorde a su especificación y cumple con las necesidades de los clientes. La V&V es un proceso de ciclo de vida completo. Inicia con las revisiones de los requerimientos y continúa con las revisiones del diseño y las inspecciones del código hasta la prueba del producto (35).

La utilización de determinados métodos como métricas para la especificación de los requisitos y casos de uso en algunos de los artefactos generados permitirá identificar y erradicar errores en los requisitos y los casos de uso. Además de estos métodos de validación se empleará la técnica de ladov para medir el nivel de satisfacción del personal que utilizará el software.

3.2.1 Métricas de la Calidad de Especificación de los Requisitos

Las Métricas de la Calidad de Especificación de los Requisitos tienen como objetivo medir la especificidad de los requisitos, haciéndolos entendibles para la parte interesada y se puedan probar (36). Es necesario conocer el número total de requisitos para comenzar a utilizar la técnica de validación.

Rf: número de requisitos funcionales.

Rnf: número de requisitos no funcionales.

Rt: total de requisitos.

$$\mathbf{Rt = Rf + Rnf}$$

Capítulo 3: Validación de la solución propuesta

$$R_t = 35 + 4 = 39$$

Especificidad

Para determinar la especificidad o falta de ambigüedad de los requisitos, se sugiere una métrica basada en la consistencia de la interpretación de los revisores de cada requisito (37).

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

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

Primera revisión

$$Q_1 = 36 / 39$$

$$Q_1 = 0.93$$

Resultado: se detectó que el 7.1% de los requerimientos presentaban ambigüedades, por lo que se realizó una reelaboración de los requerimientos, con términos más entendibles por todos.

Segunda revisión

$$Q_1 = 39 / 39$$

$$Q_1 = 1$$

Resultado: no se detectaron ambigüedades.

Compleción

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

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

Capítulo 3: Validación de la solución propuesta

Donde Q2 es el grado de completitud de los requisitos, Rc es el número de requisitos completos y Rpe el número de requisitos pobremente especificados.

$$Q2 = 37 / 37 + 2$$

$$Q2 = 0.95$$

Resultado: se detectó que el 4.8% de los requisitos definidos no estaban correctamente especificados, se realizó una reelaboración de estos requisitos y para la segunda revisión realizada no se registró ningún inconveniente.

Corrección

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

$$Q3 = (Rc - Ri) / Rc$$

Donde Q3 es el grado de validación de los requisitos, Rc número de requisitos validados como correctos y Ri número de requisitos validados como incorrectos.

$$Q3 = (36 - 3) / 36$$

$$Q3 = 0.92$$

Resultado: se concluyó que el 7.6% de los requisitos no estaban correctamente definidos, se realizó una nueva descripción de estos requisitos arrojando un resultado satisfactorio.

Consistencia interna

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

$$Q4 = (Re - Rc) / Re$$

Capítulo 3: Validación de la solución propuesta

Donde Q4 es el grado de consistencia interna, Re es el número de requisitos especificados y Rc número de requisitos en conflicto con otros requisitos.

$$Q4 = (39 - 0) / 39$$

$$Q4 = 1$$

Resultado: no se hallaron requisitos en conflictos con otros.

Estabilidad

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

$$E = (Rt - Rm) / Rt$$

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

Tabla 11: Clasificación de la estabilidad

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

$$E = (Rt - Rm) / Rt$$

$$E = (39 - 0) / 39 = 1$$

Resultado: se detectó que los requisitos no han sido modificados en cuanto al indicador estabilidad, por lo que la estabilidad se clasifica como alta.

3.2.2 Métricas de Casos de Uso

Para la validación de los caso de uso se utilizó la técnica de las métricas de casos de uso, en las que se emplean 4 atributos (38).

Compleitud

Se define un caso de uso completo cuando especifica todo lo que debe hacer el actor y el sistema para alcanzar el propósito del caso de uso y si se considera todas las respuestas del sistema a situaciones anormales para esto se determinan las siguientes interrogantes:

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

Comprensibilidad

Se determina si un caso de uso es comprensible si todos los tipos de lectores pueden entenderlo fácilmente con una mínima explicación del autor para esto se plantean las interrogantes:

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

Concisión

Capítulo 3: Validación de la solución propuesta

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

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

No trivialidad

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

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

Con la aplicación de la técnica Métricas de Casos de Uso se requirió de solo una revisión debido a que los casos de uso no presentaban ninguna anomalía en cuanto a los atributos representados, utilizados para dicha validación.

3.2.3 Aplicación de la técnica de ladov

La técnica de ladov fue creada por (Kuzmina 1970) para el estudio de la satisfacción por la profesión en carreras pedagógicas. En la presente investigación la propuesta solución está destinada a los desarrolladores del proyecto CRODA, los cuales deben de emitir un criterio de satisfacción sobre el funcionamiento y servicios que brinda la API desarrollada, cualidades que solo no se garantizan con verificar que cada funcionalidad retorna uno o varios valores, por lo que es necesario la observación por parte de personas con dominio de la especificación IMS QTI y en el desarrollo de CRODA.

Capítulo 3: Validación de la solución propuesta

Esta técnica conforma una vía indirecta para el estudio de la satisfacción, ya que los criterios que se utilizan se fundamentan en las relaciones que se establecen entre tres preguntas cerradas que se intercalan dentro de un cuestionario. Estas tres preguntas se relacionan a través de lo que se denomina el “Cuadro Lógico de ladov” (39).

Tabla 12: Cuadro Lógico de ladov

3-¿Cuál es su opinión sobre la API desarrollada en CRODA para facilitar el acceso a las actividades basadas en IMS QTI?	1-¿Considera usted factible la utilización de la especificación IMS QTI sin la visualización de las actividades que esta soporta?								
	No			No sé			Si		
	2-¿Considera usted necesaria la utilización de una API que facilite el acceso a los cuestionarios basados en IMS QTI?								
	Si	No sé	No	Si	No sé	No	Si	No sé	Si
Me gusta mucho.	1	2	6	2	2	6	6	6	6
Me gusta más de lo que me disgusta.	2	2	3	2	3	3	6	3	6
Me es indiferente.	3	3	3	3	3	3	3	3	3
Me disgusta más de lo que me gusta.	6	3	6	3	4	4	3	4	4
No me gusta en lo absoluto.	6	6	6	6	4	4	6	4	5
No sé.	2	3	6	3	3	3	6	3	4

El número resultante de la interrelación de las tres preguntas define la posición de cada encuestado en la escala de satisfacción siguiente (40):

1. Clara satisfacción.
2. Más satisfecho que insatisfecho.
3. No definida.
4. Mas insatisfecho que satisfecho.

Capítulo 3: Validación de la solución propuesta

5. Clara insatisfacción.
6. Contradictoria.

Para obtener el índice de satisfacción grupal (ISG) se trabaja con los diferentes niveles de satisfacción que se expresan en la escala numérica que oscila entre +1 y - 1 de la siguiente forma:

Tabla 13: Niveles de satisfacción.

Valor	Interpretación
+1	Máximo de satisfacción.
0.5	Más satisfecho que insatisfecho.
0	No definido y contradictorio.
-0.5	Más insatisfecho que satisfecho.
-1	Máxima insatisfacción.

El índice de satisfacción grupal se calcula por la fórmula siguiente:

$$\text{ISG} = \frac{A(+1) + B(0.5) + C(0) + D(-0.5) + E(-1)}{N}$$

En esta fórmula A, B, C, D y E representan el número de sujetos con índice individual (1, 2, 3, 4, 5 ó 6) y N representa el número total de sujetos del grupo.

El ISG arroja valores entre 1 y -1. Los valores que se encuentran definidos entre -1 y -0.5 indican insatisfacción, los comprendidos entre -0.49 y +0.49 reflejan que existe contradicciones y los que se enmarcan en el rango de 0.5 y +1 indican que existe satisfacción.

Capítulo 3: Validación de la solución propuesta

Tabla 14: Eje numérico de satisfacción



El procedimiento de cálculo del Índice de Satisfacción Grupal (ISG) de la encuesta realizada es a 6 personas que reflejada a continuación:

$$\text{ISG} = \frac{4(+1) + 1(0.5) + 0(0) + 1(-0.5) + 0(-1)}{6}$$

$$\text{ISG} = 0.6$$

El resultado muestra una clara satisfacción con una ISG de 0.6 dentro del rango [0.5, 1].

3.3 Conclusiones

Como resultado del estudio realizado en este capítulo el cual corresponde a la validación de la solución propuesta, se determinó la utilización de métricas de calidad de los requisitos y casos de uso detallados en el capítulo anterior, arrojando resultados aceptables en cada una de las mediciones efectuadas. Se utilizó la técnica de ladov para el cálculo de la satisfacción personal. Luego de ejecutarle una encuesta a seis personas se determinó que el ISG el desarrollo de una API que contribuya a la visualización de actividades de autoevaluación basadas en la especificación IMS QTI cuenta con un buen nivel de aceptación entre sus usuarios

Conclusiones generales

El análisis de conceptos, estudio de herramientas similares así como la selección de la tecnología requerida permitieron la creación del conocimiento necesario para el trabajo con la especificación IMS QTI. Se definieron y desarrollaron las funcionalidades que brinda la API-QTI para acceder a cada uno de los elementos y sus relaciones en las actividades que permite diseñar la herramienta de autor web CRODA, para su posterior visualización. El API-QTI como productor creado es flexible porque permite aumentar su número de servicios sin influir en los ya existentes e independiente porque no afecta en su funcionamiento lo que sería la otra parte de la lógica del negocio en un reproductor de actividades basadas en IMS QTI. Se realizó la validación a la propuesta solución mediante las métricas de especificación y calidad de los casos de uso, lo cual permitió la definición correcta de los requisitos y casos de uso y mediante la técnica de ladov con el propósito de calcular el nivel de satisfacción obteniendo un nivel de 0.6 contenido dentro del rango definido como satisfactorio.

Recomendaciones

Se recomienda para la futura incorporación de esta investigación en la herramienta de autor web CRODA:

- La incorporación de nuevas funcionalidades que permitan obtener los elementos de otro tipo de actividades.
- Extender las funcionalidades de la API no solo a facilitar el trabajo con el desarrollo del reproductor sino también al editor de actividades basadas en la especificación IMS QTI.
- Darle seguimiento a la versión 2.1 de la especificación IMS QTI con el objetivo de adaptar la API QTI a dicha versión.

BIBLIOGRAFÍA

1. **Heller, Michael.** *El arte de enseñar con todo el cerebro.* Venezuela : Estudios, 2005. 11.
2. **A. Mendoza, Jorge.** *e- learning el futuro de la educación informática .* s.l. : Milenium, 2003.
3. **Rodriguez Gonzalez, Leonardo.** *Analisis y diseno de la version 3.0 de RHODA.* Habana : s.n., 2011.
4. **Tnet, Red.** La formacion sin distancia. Servicio publico de empleo estatal. [En línea] [en linea], 2005. [Citado el: 6 de octubre de 2010.] www.inem.es/otras/TTnet/pdfs/LIBRO_laformacionsindistancia.pdf.
5. *Ventajas del uso de las TIC en el proceso de Enseñanza-Aprendizaje desde la optica de los docentes universitarios españoles.* **Carlos Ferro Soto, Ana Isabel Martinez Senra, Maria Carmen Otero Neira.** 29, s.l. : EDUTEC, 2009.
6. **LTSC.** *Draft Standard for Learning Object Metadata.* New York : IEEE – SA. 2002.
7. **Mateos, Jorge E. Gil.** *Estrategia de Gestión de Recursos Educativos Abiertos en forma de Objetos de Aprendizaje en la Universidad de La Habana.* La Habana : s.n., 2010.
8. *Especificaciones y estándares en e-learning.* **Manjón, Baltazar Fernández.** 6, 2006.
9. **Caceres, Manuel.** *Moodle.*
10. *Principales características de Moodle.* **Gonzales, Mary Peñalver.** 2011.
11. *CLAROLINE 1.8 Manual del Profesor.* **Quan, Roberto Argueta.** 2009.
12. **Mercenier, Philippet.** [En línea] . *lab.claroline.ne.* [En línea] 14 de de Enero de 2011.
13. **Sakai, Group.** *Longsight.* trysakai.longsight.com/portal. *trysakai.longsight.com/portal.* [En línea] 14 de Enero de 2011.
14. **Comunicaciones, Ministerio de la Informatica y las.** *EcuRed.* *EcuRed.* [En línea] 7 de diciembre de 2011. <http://www.ecured.cu/index.php/API>.
15. **Isaías Carrillo Pérez, Rodrigo Pérez González, Aureliano David Rodríguez Martín.** *METODOLOGIA DE DESARROLLO DE SOFTWARE.* 2008.
16. **Bray, T.** *Extensible Markup Language (XML).* 2008.
17. *Procesador de hipertexto.* [En línea] 13 de Octubre de 2009 . <http://autorneto.com/tecnologia/software/procesador-de-hipertexto/>.
18. **Naula, Jorge.** *¿Qué es un framework de desarrollo web?* [En línea] 9 de Diciembre de 2010. <http://kokiing.blogspot.com/2010/12/que-es-un-framework-de-desarrollo-web.html>.

19. LibrosWeb.es. [En línea] 2013.
http://librosweb.es/symfony_1_2/capitulo_1/symfony_en_pocas_palabras.html.
20. **Cuba, Ministerio de la Informatica y las Comunicaciones en.** EcuRed. *EcuRed*. [En línea] 11 de mayo de 2011. http://www.ecured.cu/index.php/Eclipse,_entorno_de_desarrollo_integrado.
21. —. EcuRed. *EcuRed*. [En línea] 28 de marzo de 2013. <http://www.ecured.cu/index.php/NetBeans>.
22. **Asenjo, Jorge Sánchez.** *Sistemas Gestores de Base de Datos*. 2009.
23. **Pérez, Manuel Sotillo.** *Bases de datos XML nativas*.
24. **Foundation, The Apache Software.** Apache. *Apache*. [En línea] 10 de febrero de 2012.
<http://es.opensuse.org/Apache>.
25. **Grady Booch, Jim Rumbaugh e Ivar Jacobson.** *El Lenguaje Unificado de Modelado*.
26. *MODELO DEL DOMINIO*. **Hall., Craig Larman.** Prentice. s.l. : UML y Patrones. 2a Edición., 2003.
27. *Actores y sus roles Modelo del dominio*. **Gómez, Gloria Lucia Giraldo.** 2006.
28. **Sommerville, I. and P. Sawyer.** "Requirements Engineering A good practice guide.". 1997.
29. **Mendoza, M.** Metodología de Desarrollo de Software. [En línea] 2004.
http://www.informatizate.net/articulos/metodologias_de_desarrollo_de_software_07062004.html.
30. **Ceria, Santiago.** *Ingeniería de Software I, Casos de Uso, Un Método Práctico para Explorar Requerimientos*.
31. *El proceso unificado de desarrollo de software*. **Jacobson, Ivar, Booch, Grady y Rumbaugh, James.** Primera edición, Madrid : Pearson Educación S.A, 2000.
32. **Guerrero, Luis A.** *UML - Diagramas de interacción*. [CC61J] - Taller de UML] s.l. : Universidad de Chile, Departamento de Ciencias de la Computación.
33. *Fundamentos del diseño de software*.
34. *Patrones de Diseño*. s.l. : UAM, 2009.
35. **José M. Drake, Patricia López.** *Verificación y Validación*. 2009.
36. **Pressman, R.** *Ingeniería del Software. "Un enfoque práctico"*. 2005.
37. **Vargas, M. A. A.** *Métricas del software*. 2007.
38. *Revista de Procesos y Métricas de las Tecnologías de la Información*. **B. Bernárdez, A. D., M. Toro.** 2004.

39. *La satisfacción del profesor de Educación Física*. **Marta Bárbara Iznaola Cuscó, Jenniffer Gabriel Wells**. n. 27-2, 27-35, s.l. : Funámbulos Editores, 2008.
40. *La tecnica de ladov. Una aplicación para el estudio de la satisfacción de los alumnos para la clase de educación física*. . **RODRÍGUEZ, D. A. L. and MAURA, V. G.** s.l. : Revista Digital-Buenos Aires, 2002.
41. **Arselis Torres Dominguez, Diego Cárdenas Sáinz, Juan Enrique Gutiérrez Moreno**. Diseño de Objetos de Aprendizaje Utilizando la Herramienta de Modelado UML. 2006.
42. *LAS HERRAMIENTAS DE AUTOR EN EL PROCESO DE PRODUCCIÓN DE CURSOS EN FORMATO DIGITAL*. **José L. Montero O'Farrill, Elsa Herrero Tunis**. 33, La Habana : Pixel-Bit. Revista de Medios y Educación, 2008.
43. e-teach. *e-teach*. [En línea] <http://www.e-teach.ch/qplayer.php?url=qplayer.EN&page=faq>.
44. *API*.
45. *Extensible Markup Language (XML)*. s.l. : Balisage Conference , 2012.
46. **(2000), IEEE-Std-1471-2000**. *Recommended Practice for. Architectural Description of Software-Intensive Systems*. 2000.
47. **Moodle**. *demo.moodle.net. demo.moodle.net*. [En línea] 14 de Enero de 2011.

Glosario de Términos

Reutilización: capacidad de los OA para integrarse con otros formando una sola entidad. En términos de reutilización de metadatos se refiere a la capacidad de reutilizar los metadatos de un recurso para describir otro.

Accesibilidad: Capacidad que hace identificable y ubicable cuando se necesite al contenido de un recurso, para los requerimientos formativos necesarios, que se conozca su adecuación a los objetivos sin necesidad de obtener el propio contenido o pagar derechos por él, mediante la provisión de información suficiente.

Interoperabilidad: Capacidad de las plataformas para comunicarse e integrarse de una forma efectiva aun siendo diferentes.

ChoiceInteraction (Interacción Selección): Esta interacción soporta las actividades de selección simple, respuestas múltiple y verdadera o falsa.

OrderInteraction (Interacción Ordenamiento): Esta interacción soporta las actividades de ordenamiento de un conjunto de elementos.

AssociateInteraction (Interacción Asociación): Esta interacción soporta las actividades de asociar filas y columnas.

MatchInteraction (Interacción Enlazar Columnas): Esta interacción soporta las actividades de enlazar los elementos contenidos en dos columnas.

GapMatchInteraction (Interacción Llenar espacios en blanco): Esta interacción soporta las actividades de llenar huecos o espacios en blanco.

QTI (Interoperabilidad de preguntas y exámenes): Especificación la cual proporciona un lenguaje XML para representar preguntas y exámenes de forma que permita la interoperabilidad del contenido con los sistemas de evaluación.

IMS (Global Learning Consortium, Inc): IMS Global Learning Consortium es un grupo independiente, sin ánimo de lucro que inició su labor en 1997 impulsado por el NLII (National Learning Infrastructure Initiative) que es una organización apoyada por Educase. Ha publicado

numerosas especificaciones, cada una de ellas enfocada a una necesidad distinta del proceso de enseñanza.

Anexos

Anexo 1: Diagramas de clases del análisis.



Ilustración 22: Diagrama de clases del análisis CU2 Reproducir actividad de asociación.



Ilustración 23: Diagrama de clases del análisis CU3 Reproducir actividad de ordenamiento.

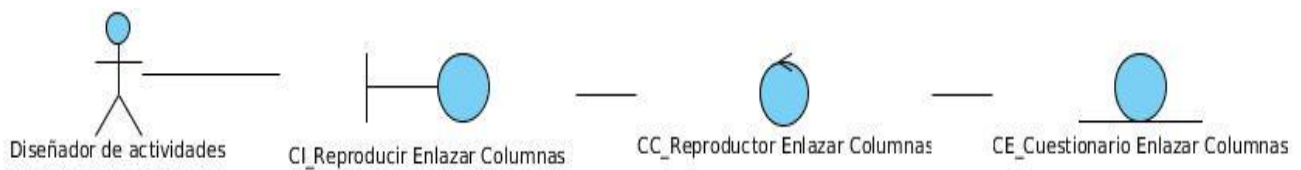


Ilustración 24: Diagrama de clases del análisis CU4 Reproducir actividad de enlazar columnas.

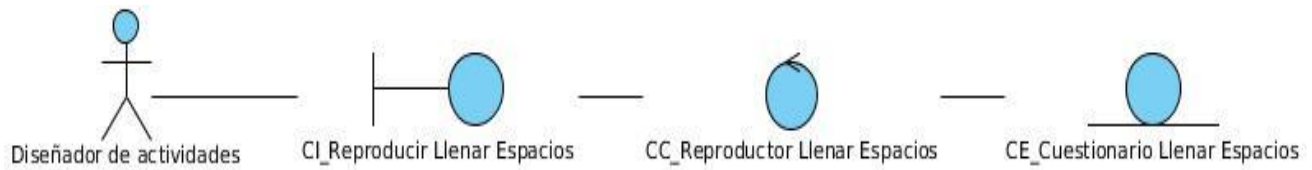


Ilustración 25: Diagrama de clases del análisis CU5 Reproducir actividad de llenar espacios.

Anexo 2: Diagramas de colaboración.

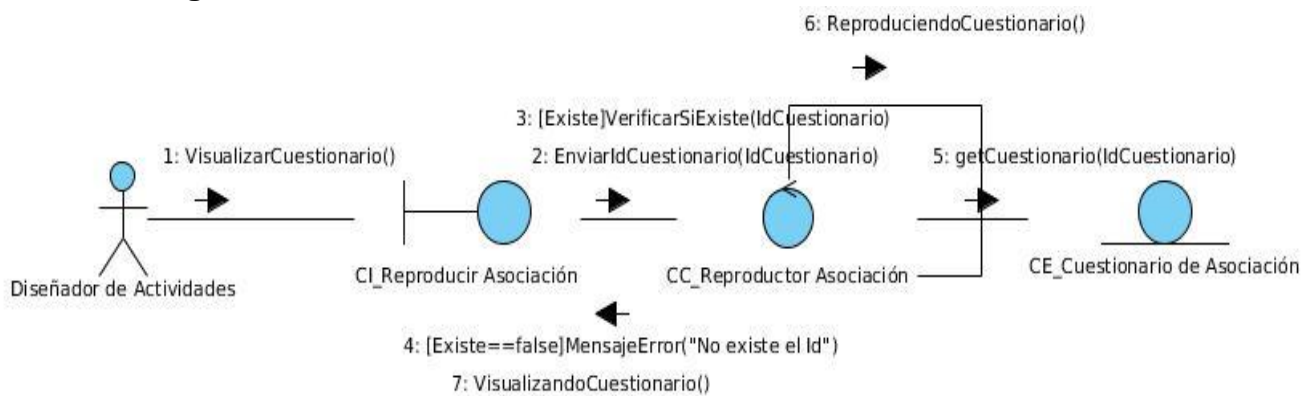


Ilustración 26: Diagrama de Colaboración CU2 Reproducir actividad de asociación.

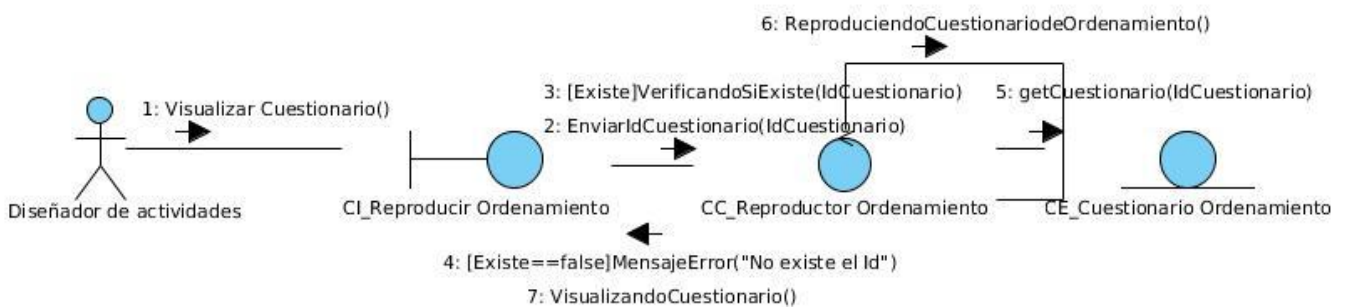


Ilustración 27: Diagrama de Colaboración CU3 Reproducir actividad de ordenamiento.

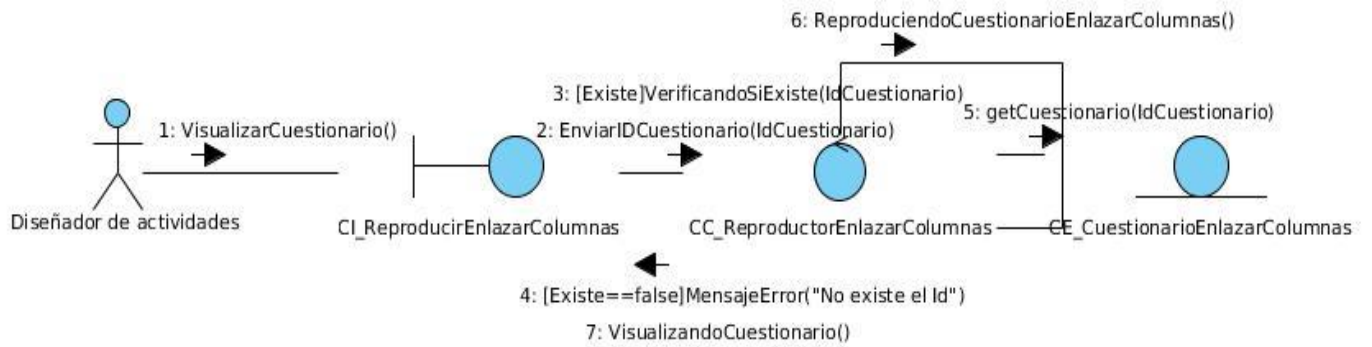


Ilustración 28: Diagrama de Colaboración CU4 Reproducir actividad de enlazar columnas.

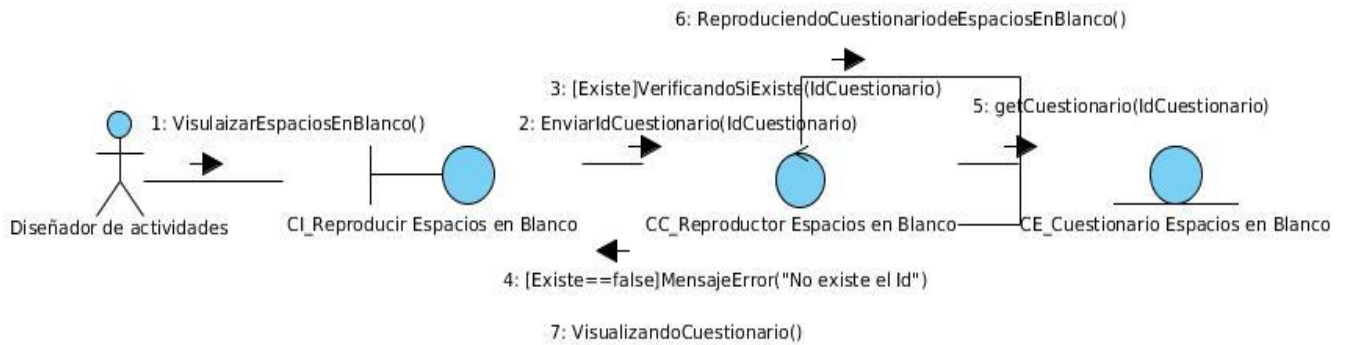


Ilustración 29: Diagrama de Colaboración CU5 Reproducir actividad de espacios en blanco.

Anexo 3: Diagrama de secuencia.

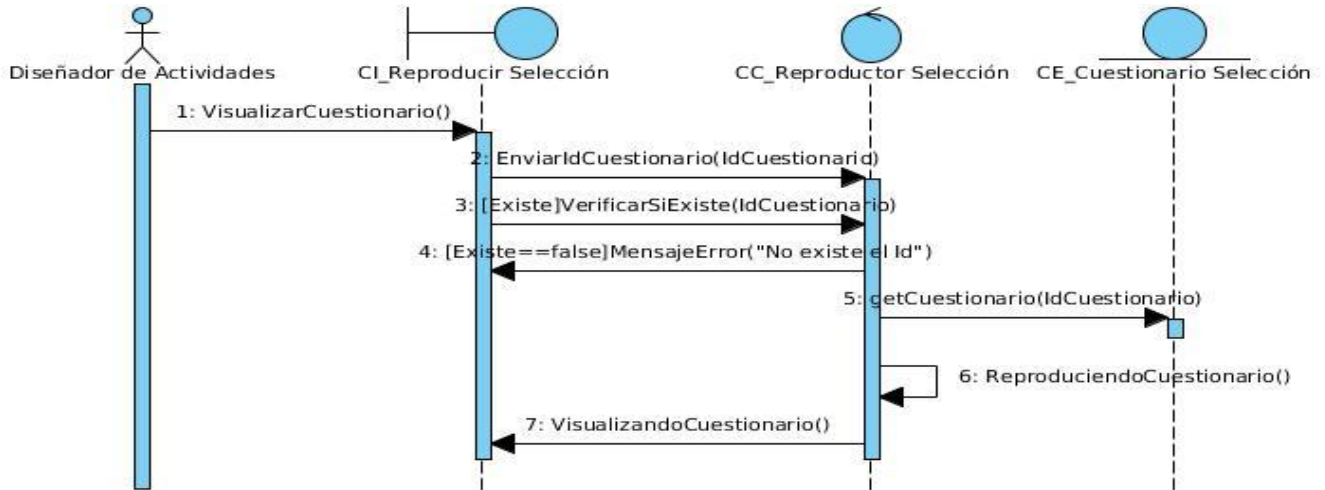


Ilustración 30: Diagrama de Secuencia CU1 Reproducir actividad de selección.

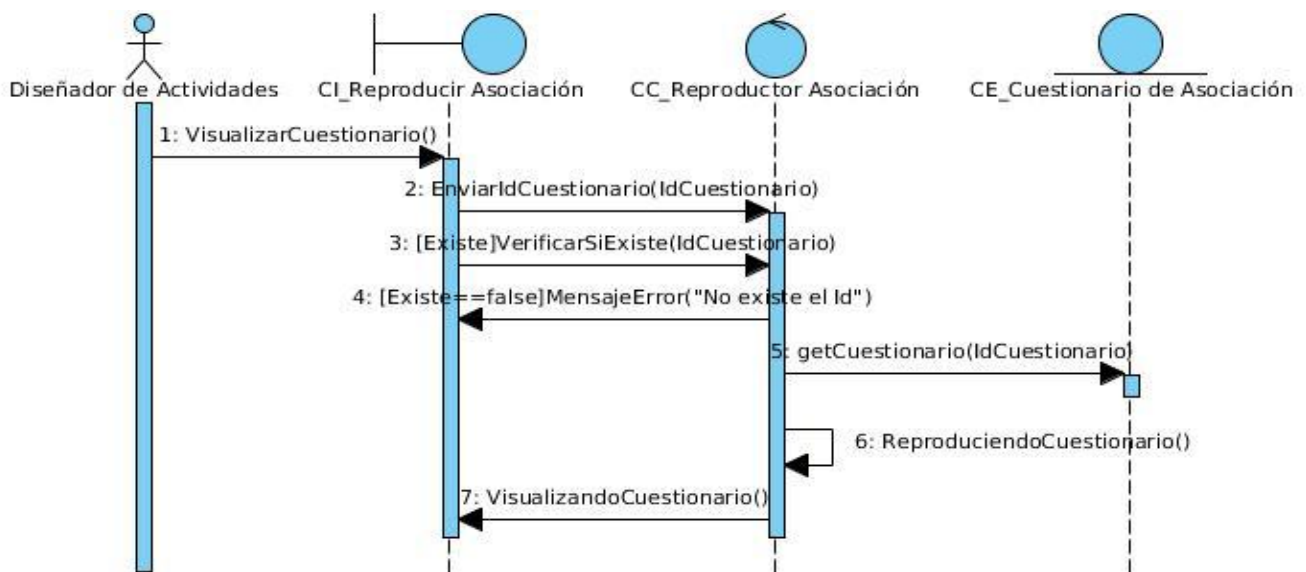


Ilustración 31: Diagrama de Secuencia CU2 Reproducir actividad de asociación.

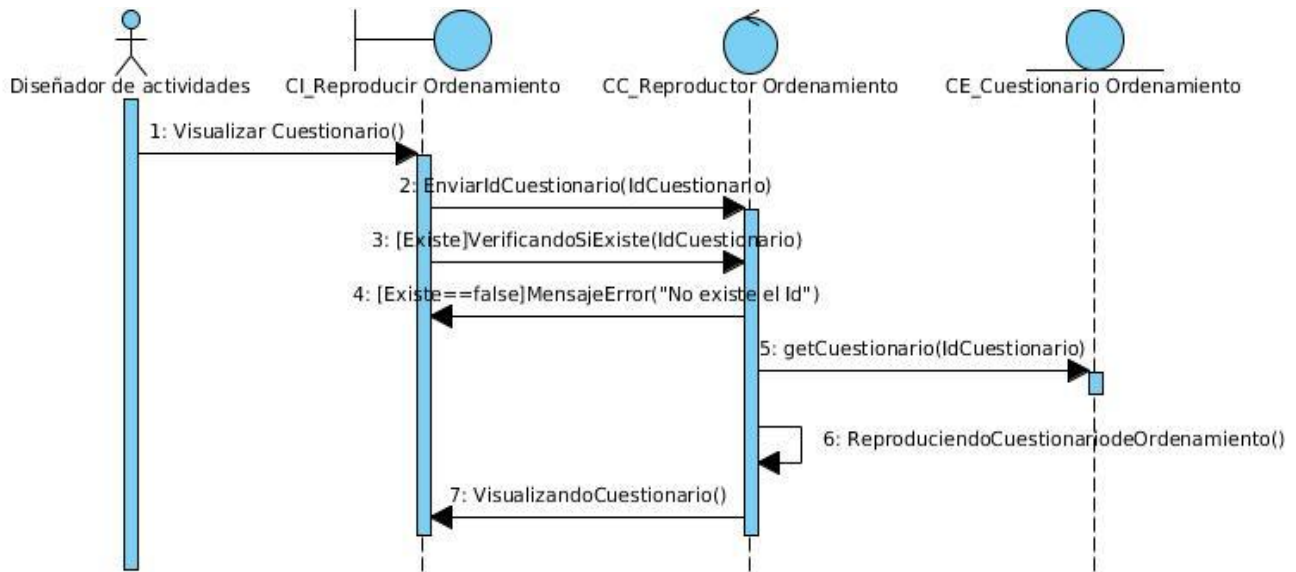


Ilustración 32: Diagrama de Secuencia CU3 Reproducir actividad de ordenamiento.

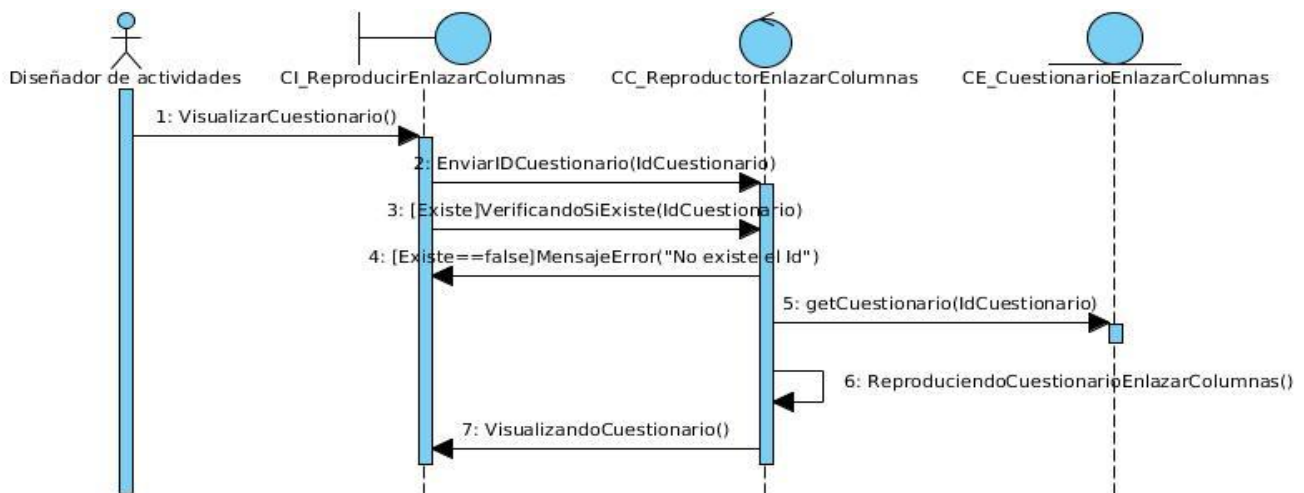


Ilustración 33: Diagrama de Secuencia CU4 Reproducir actividad de enlazar columnas.

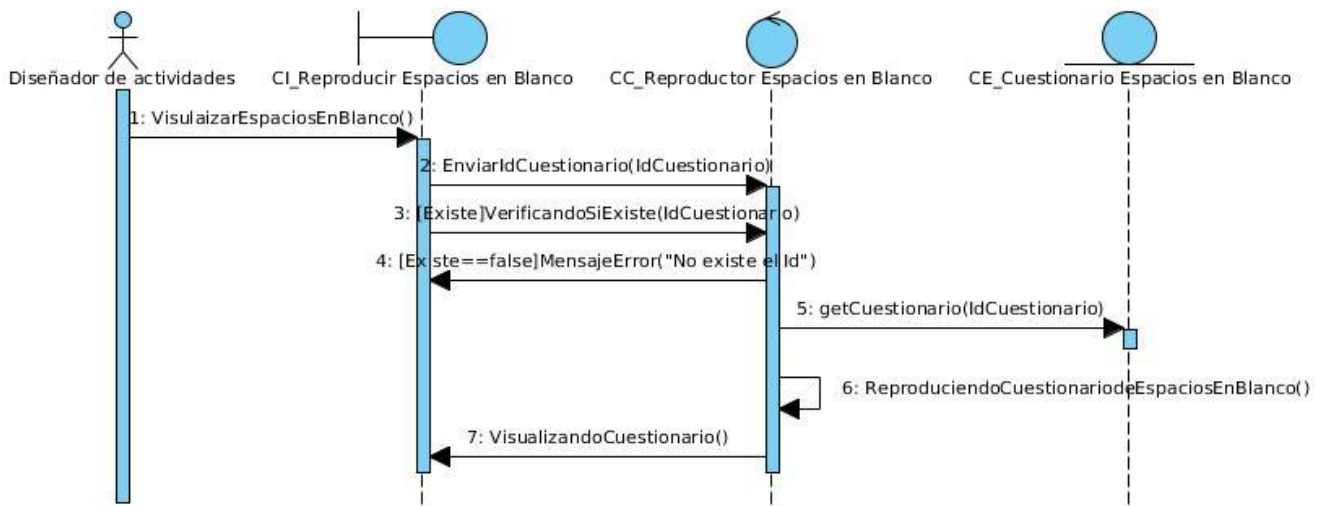


Ilustración 34: Diagrama de Secuencia CU5 Reproducir actividad de espacios blanco.

Anexo 4: Diagrama de clase del diseño.

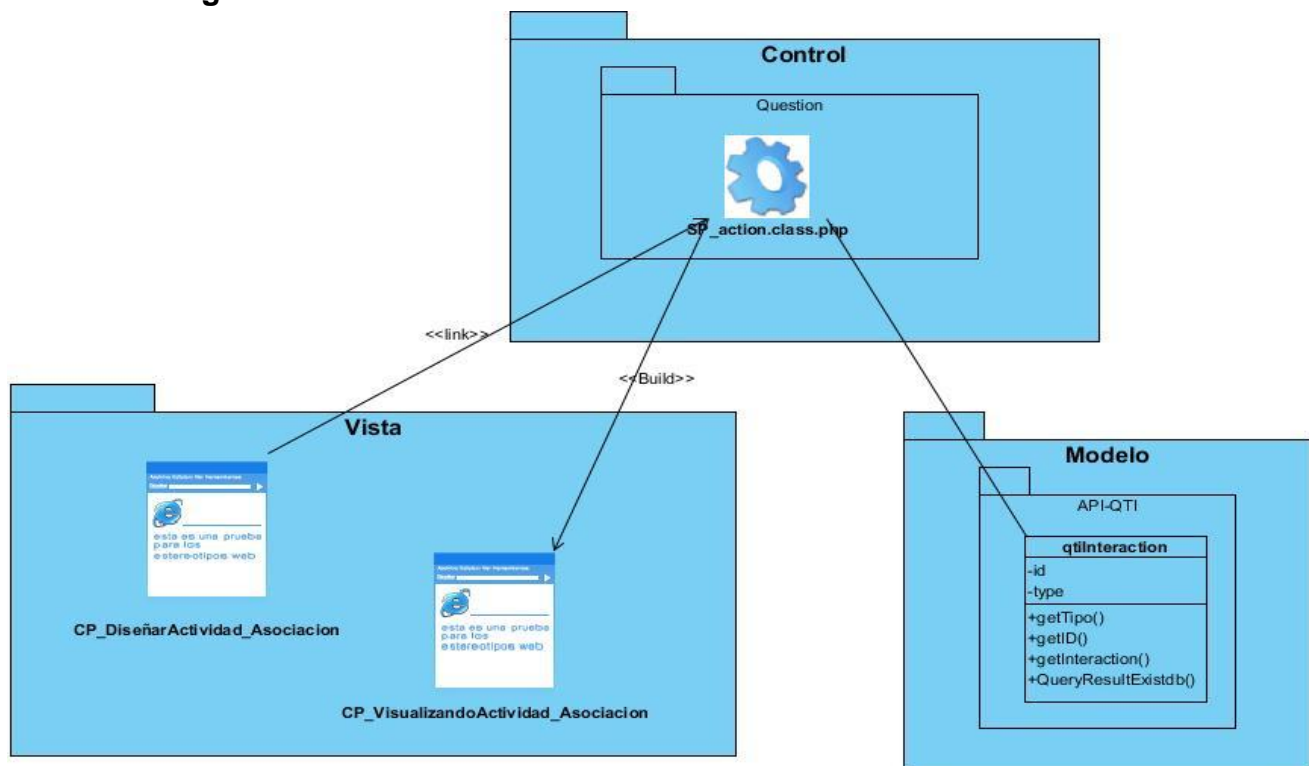


Ilustración 35: Diagrama de clases del diseño CU2 Reproducir actividad de asociación.

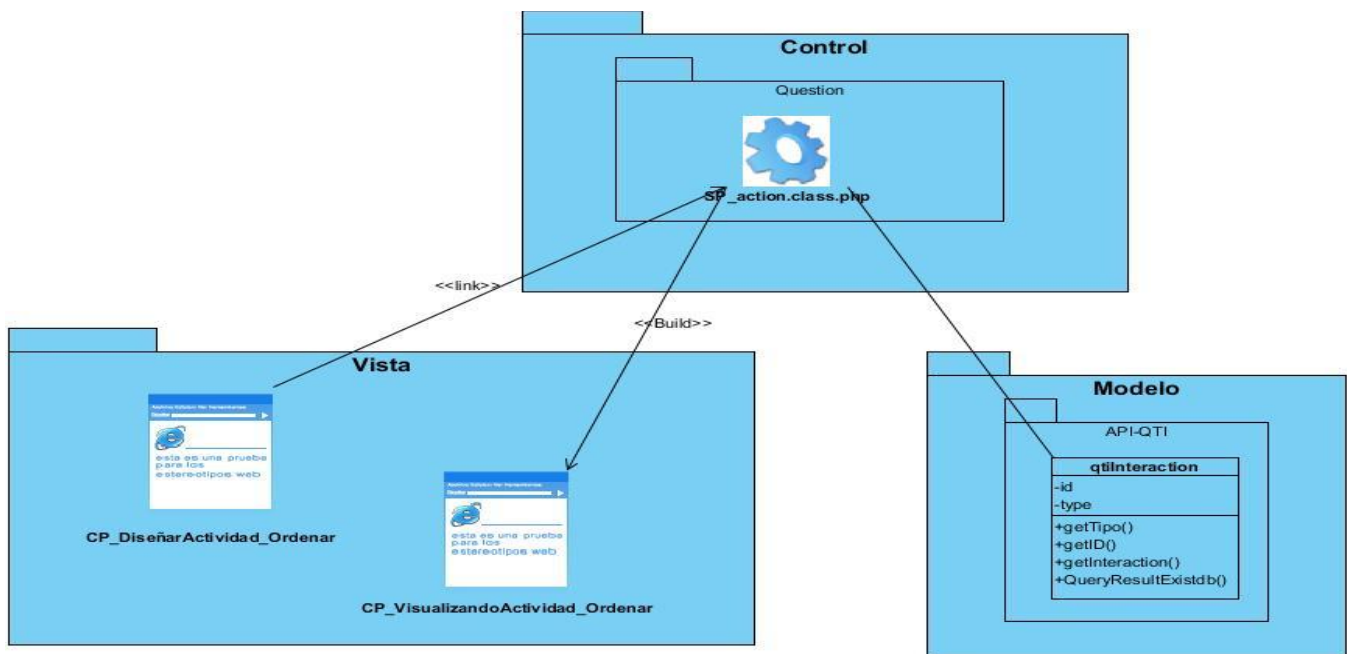


Ilustración 36: Diagrama de clases del diseño CU3 Reproducir actividad de ordenamiento.

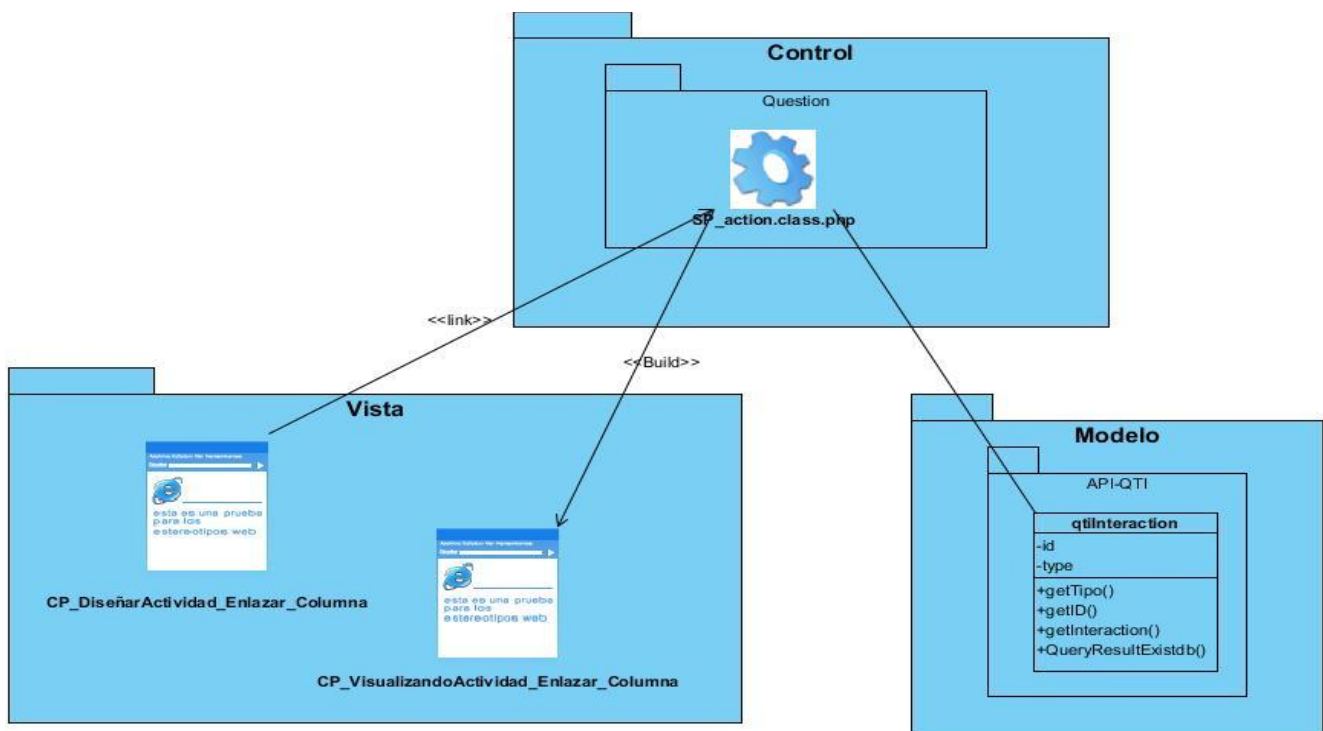


Ilustración 37: Diagrama de clases del diseño CU4 Reproducir actividad de enlazar columnas.

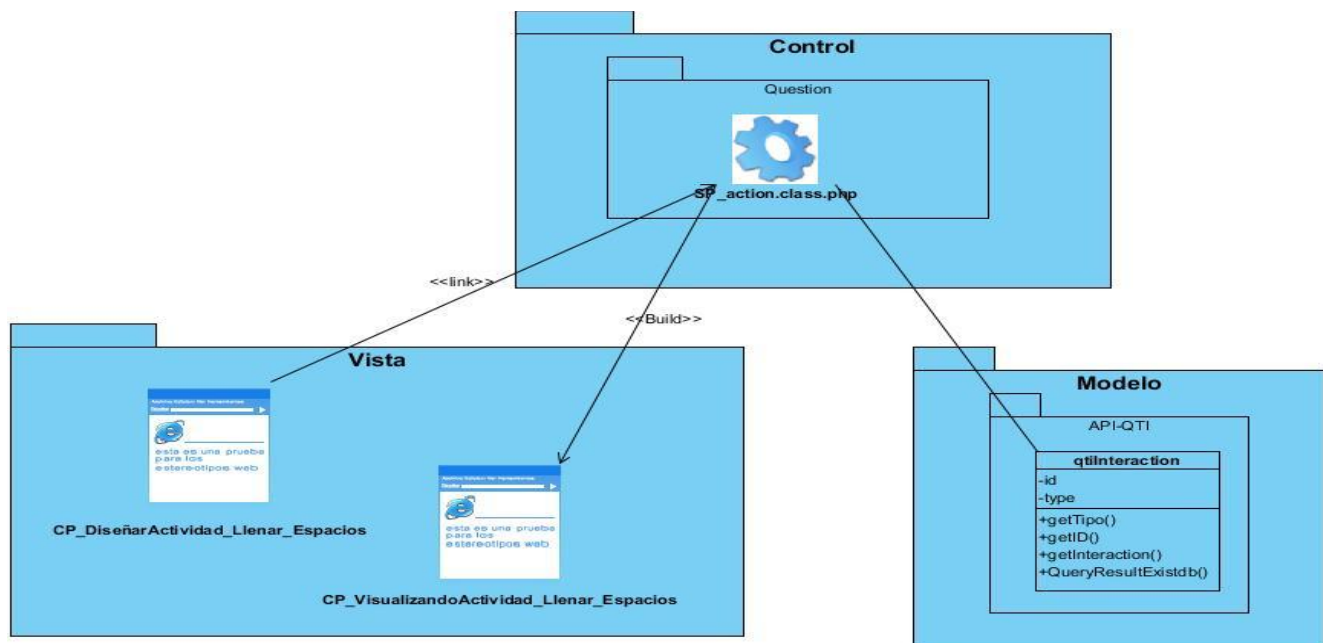


Ilustración 38: Diagrama de clases del diseño CU5 Reproducir actividad de llenar espacios.

Anexo 5: Validación de Requerimientos.

Revisores

R1. Ing. Osvaldo E. Stable Vilches.

R2. Ing. Yisel Quiñones Guerrero.

R3. Ing. Jorge Iturría Pozo.

R4. Ing. Yandris Mata Cabrera.

R5. Ing. Leandro Gonzáles Vallejo.

Los Requerimientos Funcionales y no Funcionales. (Ver Capítulo 2, Acápites 2.3 Requerimientos)

Especificidad

Para la evaluación de la especificidad de los requisitos cada revisor emitió su criterio, lo que arrojó como resultado que los revisores no consideraron específicos 3 de los requisitos

funcionales escritos por lo que fue necesario modificarlos para un mejor entendimiento de los mismos.

Para obtener el resultado final de los restantes indicadores es necesario indicar en la columna Resul, si está correcto (C) o incorrecto (I) el requerimiento de acuerdo al indicador, al menos tres revisores deben coincidir con un criterio.

Tabla Validación de completitud

Compleción

Revisores Requerimientos	R1	R2	R3	R4	R5	Resul
RF1	I	C	I	I	C	I
RF2	C	C	C	C	C	C
RF3	C	I	I	C	C	C
RF4	C	I	I	I	C	I
RF5	C	C	C	C	C	C
RF6	I	C	C	C	C	C
RF7	C	C	C	C	I	C
RF8	C	C	C	C	C	C
RF9	C	C	C	C	C	C
RF10	C	C	I	C	C	C
RF11	I	C	C	C	I	C
RF12	C	C	C	C	C	C
RF13	C	I	C	C	C	C
RF14	C	C	C	C	C	C
RF15	C	C	C	C	C	C
RF16	C	C	C	C	C	C
RF17	C	C	C	C	C	C
RF18	C	C	C	C	C	C
RF19	C	C	C	C	I	C
RF20	C	C	C	C	C	C
RF21	C	I	C	C	C	C

Tabla Validación de corrección

Corrección

Revisores Requerimientos	R1	R2	R3	R4	R5	Resul
RF1	I	I	C	I	C	I
RF2	C	C	C	C	I	C
RF3	C	I	I	I	I	I
RF4	I	C	I	I	I	I
RF5	C	C	C	C	C	C
RF6	C	C	C	C	I	C
RF7	C	C	C	C	C	C
RF8	C	C	C	C	I	C
RF9	C	I	C	C	C	C
RF10	C	C	C	I	C	C
RF11	C	C	C	C	C	C
RF12	C	C	C	C	C	C
RF13	C	C	C	C	I	C
RF14	C	C	C	I	C	C
RF15	C	C	C	C	C	C
RF16	C	C	C	C	C	C
RF17	C	C	I	C	C	C
RF18	C	C	C	C	C	C
RF19	C	C	C	C	I	C
RF20	C	C	C	C	C	C
RF21	C	C	C	C	C	C

RF22	C	C	C	I	C	C
RF23	C	C	C	C	C	C
RF24	C	C	C	C	C	C
RF25	C	C	C	C	C	C
RF26	C	C	C	C	C	C
RF27	C	C	C	C	C	C
RF28	C	C	C	C	C	C
RF29	C	C	C	I	C	C
RF30	C	C	C	C	C	C
RF31	C	C	C	C	C	C
RF32	C	C	I	C	C	C
RF33	C	C	C	C	C	C
RF34	C	I	C	C	C	C
RF35	C	C	C	C	C	C
RFN1	C	I	C	C	C	C
RFN2	I	I	I	I	C	I
RFN3	C	C	C	C	C	C
RFN4	C	C	C	C	C	C

RF22	C	C	C	C	C	C
RF23	C	C	C	C	C	C
RF24	C	C	C	C	C	C
RF25	C	C	C	C	C	C
RF26	C	C	C	C	C	C
RF27	C	C	C	C	C	C
RF28	C	C	C	C	C	C
RF29	C	C	C	C	C	C
RF30	C	C	C	C	C	C
RF31	C	C	C	C	C	C
RF32	C	C	C	C	C	C
RF33	C	C	C	C	C	C
RF34	C	C	C	C	C	C
RF35	C	C	C	C	C	C
RFN1	C	C	C	C	C	C
RFN2	C	C	C	C	C	C
RFN3	C	C	C	C	I	C
RFN4	C	I	C	C	C	C

Tabla Consistencia interna

Consistencia interna

Revisores Requerimientos	R1	R2	R3	R4	R5	Resul
RF1	C	C	C	C	C	C
RF2	C	C	C	C	C	C
RF3	C	C	C	C	C	C
RF4	C	C	C	C	C	C
RF5	C	C	C	C	C	C
RF6	C	C	C	C	C	C
RF7	C	C	C	C	I	C
RF8	C	C	C	C	C	C

Tabla Estabilidad

Estabilidad

Revisores Requerimientos	R1	R2	R3	R4	R5	Resul
RF1	C	C	C	C	C	C
RF2	C	C	C	C	C	C
RF3	C	C	I	C	C	C
RF4	C	C	I	C	C	C
RF5	C	C	C	C	C	C
RF6	C	C	C	C	C	C
RF7	C	C	C	C	I	C
RF8	C	C	C	C	C	C

RF9	C	C	C	C	C	C
RF10	C	C	C	C	C	C
RF11	C	C	C	C	C	C
RF12	C	C	C	C	C	C
RF13	C	C	C	C	C	C
RF14	C	C	C	C	C	C
RF15	C	C	C	C	C	C
RF16	C	C	C	C	C	C
RF17	C	C	C	C	C	C
RF18	C	C	C	C	C	C
RF19	C	C	C	C	C	C
RF20	C	C	C	C	C	C
RF21	C	C	C	C	C	C
RF22	C	C	C	C	C	C
RF23	C	C	C	C	C	C
RF24	C	C	C	C	C	C
RF25	C	C	C	C	C	C
RF26	C	C	C	C	C	C
RF27	C	C	C	C	C	C
RF28	C	C	C	C	C	C
RF29	C	C	C	C	C	C
RF30	C	C	C	C	C	C
RF31	C	C	C	C	C	C
RF32	C	C	C	C	C	C
RF33	C	C	C	C	C	C
RF34	C	C	C	C	C	C
RF35	C	C	C	C	C	C
RFN1	C	C	C	C	C	C
RFN2	C	C	C	C	C	C
RFN3	C	C	C	C	C	C
RFN4	C	C	C	C	C	C

RF9	C	C	C	C	C	C
RF10	C	C	C	C	C	C
RF11	C	C	C	C	C	C
RF12	C	C	C	C	C	C
RF13	C	C	C	C	C	C
RF14	C	C	C	C	C	C
RF15	C	C	C	C	C	C
RF16	C	C	C	C	C	C
RF17	C	C	C	C	C	C
RF18	C	C	C	C	C	C
RF19	C	C	C	C	C	C
RF20	C	C	C	C	C	C
RF21	C	C	C	C	C	C
RF22	C	C	C	C	C	C
RF23	C	C	C	C	C	C
RF24	C	C	C	C	C	C
RF25	C	C	C	C	C	C
RF26	C	C	C	C	C	C
RF27	C	C	C	C	C	C
RF28	C	C	C	C	C	C
RF29	C	C	C	C	C	C
RF30	C	C	C	C	C	C
RF31	C	C	C	C	C	C
RF32	C	C	C	C	C	C
RF33	C	C	C	C	C	C
RF34	C	C	C	C	C	C
RF35	C	C	C	C	C	C
RFN1	C	C	C	C	C	C
RFN2	C	C	C	C	I	C
RFN3	C	C	C	C	C	C
RFN4	C	C	C	C	C	C

Anexo 6: Cuestionario para la aplicación de la técnica de ladov.

1-¿Considera usted factible la utilización de la especificación IMS QTI sin la visualización de los cuestionarios que esta soporta?

Sí No No sé

2-¿Considera usted necesaria la utilización de una API que facilite el acceso a los cuestionarios basados en IMS QTI?

Sí No No sé

3-¿Cuál es su opinión sobre la API desarrollada en CRODA para facilitar el acceso a los cuestionarios basados en IMS QTI?

Me gusta mucho.

Me gusta más de lo que me disgusta.

Me es indiferente.

Me disgusta más de lo que me gusta.

No me gusta en lo absoluto.

No sé.

4-¿Cree usted que con la utilización de una API se facilita el trabajo al desarrollador?

Sí No No sé