



UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

Análisis y diseño de nuevas funcionalidades
para el Módulo Actividades en la
herramienta de autor web CRODA 2.0

TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE INGENIERO EN CIENCIAS
INFORMÁTICAS.

Autor:

YASMANY AGUILERA SÁNCHEZ

Tutores:

MARINÉS ALEMÁN LLANO

JORGE ITURRIA POZO

Cotutor:

OSVALDO ERNESTO STABLE VILCHES

LA HABANA, JUNIO DEL 2012

“AÑO 54 DE LA REVOLUCIÓN”

Declaración de autoría

Por este medio declaro que soy el único autor del trabajo Análisis y diseño de nuevas funcionalidades para el Módulo Actividades en la herramienta de autor web CRODA 2.0 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 el presente a los ____ días del mes de _____ del año_____.

AUTOR:

YASMANY AGUILERA SÁNCHEZ

TUTORES:

MARINÉS ALEMÁN LLANO

JORGE ITURRIA POZO

OSVALDO STABLE VILCHES



"... Si un hombre es perseverante, aunque sea duro de entendimiento se hará inteligente y aunque sea débil se transformará en fuerte."

Leonardo Da Vinci

DEDICATORIA

A mi mamá, por todos estos años de dedicación y desvelos, por hacer de mí todo lo que soy.

A mi hermana, por significar tanto para mí en esta vida; espero que haya logrado ser tu ejemplo.

A todas las personas de mi familia, que vean que no importa la cantidad de obstáculos que se interpongan en el camino, lo importante siempre es la resistencia y los deseos de lograr alcanzar las metas.

AGRADECIMIENTOS



Todo el empeño por lograr la meta de ser un profesional termina este día, donde queda la alegría y el orgullo de haber compartido con personas maravillosas que siempre quedarán en nuestros corazones. A esas personas quisiera agradecerles a continuación:

Quisiera comenzar agradeciendo a Dios por guiar mis pasos en los estudios y ayudarme a que este sueño se hiciera realidad el sueño de convertirme en el orgullo de mi familia y mis padres.

Agradezco a la Revolución, a Fidel y la UCI por ser los principales protagonistas de convertirme en ingeniero.

A todas las personas que de una forma u otra se interponían en mi camino poniendo a diario un obstáculo; pues en vez de frenarme lograban fortalecerme.

Al tribunal y en especial a mi oponente por haberme guiado en cada momento que fue necesario y mostrarme su ayuda cuando más la necesité.

A mis tutores Marinés y Yoandy (Mary: esta tesis es casi más tuya, que mía, a ti mil gracias), Jorge (aunque al final me cambiaste por un viaje a Venezuela, pero bueno no te guardo rencor, muchas gracias por todo tu apoyo) y finalmente y no menos importante a Osvaldo, por toda la ayuda que me brindaron en este periodo para culminar satisfactoriamente mi trabajo de diploma.

A Rosalba Carralero (Rosy), mi trigueña intensa: no tengo palabras para agradecerte todo lo que has hecho por mí a lo largo de estos 6 años, gracias por existir.

A todos los miembros del proyecto CRODA, Yudith, Daniel, Ramiro, Wilson y Milene.

A mi analista destacada Danay, aunque por poco le da algo cuando se enteró que yo tenía que ponerle trabajo, espero que para el año que viene ya hayas alcanzado una buena experiencia.

A todas mis amistades, me llevo un poquito de cada uno, a todas las quiero mucho.

A los que me pararon en cualquier lugar para hacerme la pregunta ¿Cómo va la tesis?

A los que lograron acompañarme en mis momentos de locura de los festivales de artistas aficionados de la FEU y en especial a Nersa, por verme soportado tantos años bailando kizomba. A todos los miembros



del piquete de “los Octavitos” en la competencia Equipo UCI: Leandro, Alberto, Roberto Daniel, Nesty y en especial a Milena, por permitirme bailar con ella en todas las galas y Arcadio, el motor impulsor de todas las danzas de la facultad.

A la mejor secretaria que ha tenido la facultad, Elibetsy (mi cherry). Por ayudarme en todo momento, por tanto apoyo incondicional gracias.

A la planificadora aguerrida de la facultad Mary, gracias por ser tan preocupada y atenta conmigo.

Al vicedecano Antonio por verme tendido una mano amiga cuando más lo necesitaba.

A Yusdel y el profe Basulto por verme dado una segunda oportunidad cuando todo parecía estar perdido.

A Yuliet y Mairelys por permitirme aprender programación repasándolas.

A Claudia, por enseñarme que en la vida las cosas para alcanzarlas hay que lucharlas y pasar mucho trabajo para luego disfrutarlas más, por tu ayuda muchas gracias.

A Amanda por confiar en mí cuando me necesitaste para tu competencia y a Yohanne, espero que ahora si estés convencida que lo que yo prometo lo cumplo, parece que te cuadro cuando de mi “Bachata” no quieres irte.

A la gente con que compartí mi último año de la carrera y en especial a Vilson y el Guille por convertirse en mis hermanos de la UCI cuando me sentía más solo.

A mis compañeros del grupo 8104 con los que compartí mi primer año en la escuela.

A los muchachos del 4504 que cuando entre en su grupo me acogieron como un amigo más

A mi gente de la 5ta graduación por compartir con ellos tantos buenos y no tan buenos momentos, en especial a: Mita, Jesy, Yane, Lisy, Liset, Yadima, Lia, Mily, Anet, Suly, Leo, Ale, Ancel, Yustin, Adrian, Yerandy, Julio, Noslén, Jose y Edisnay (y a Yisell, una compañera más de cuarto). Por permitirme tantas locuras en el apartamento, muchas gracias.

A Ana Margarita Claro, Ana: muchas gracias por tu amistad y por tantas revisiones de mi tesis; casi te conviertes en una tutora más. Espero no ver quedado mal contigo. Por ser tan buena amiga muchas gracias.

A la profe Mayi, por su amistad y dedicación, por tu ejemplo: CUANDO SEA GRANDE QUIERO SER COMO TÚ!

A mi familia de la UCI por permitirme tantas locuras en su compañía: Licett, Iraisy, Yamila, Indy (espero que no se te olviden los besos de la buena suerte, TE QUIERO MI CHIQUITICA).



Lisy, esta tesis también es tuya y espero que pronto me digas ESTAS INVITADO A LA MÍA, ESPERO QUE NO TE OLVIDES DE MÍ.

A mi hermano Tomy, aunque me abandonaste en el último año, no importa estás perdonado.

A Dailyn, por toda su preocupación aun cuando ya no se encontraba trabajando en la escuela. Muchas gracias por tenerme siempre presente.

A una persona que se convirtió en mi compañera, amiga y hermana y nunca me abandonó en los momentos difíciles, por permitirme ser tu hermano y consejero en todo momento un millón de gracias, Arianna Rodríguez (Ary) nunca voy a tener palabras para expresarte lo agradecido que estoy contigo.

A mi bomboncito por hacer de mis años en la universidad los más entretenidos, Yerisleidys Sao (Yery) gracias por permitirme estar contigo en los momentos difíciles.

A Yoly, muchas gracias por verme ayudado en los inicios de mi tesis cuando aún no sabía ni por cual rumbo coger.

A mis abuelos, mis viejitos lindos para los cuales espero ser su orgullo por el resto de la vida.

A mis tías: Olivia, Osmerly, Jisela, Reinoa, Nansy y Griselda, así como a mis tíos Eddy, Julio y Anibal a todos gracias por tanto apoyo.

A dos primas que cuando llegue aquí a La Habana me acogieron en su casa como si fuera un hijo, a Mabel y Yoandra muchas gracias.

A Emilia (mi cuca), que el tiempo no le alcanzó para ver este momento y que hubiera estado muy orgullosa de verme ingeniero.

A mi hermanita linda, mi tita, mi nini: Leyi, espero que a partir de ahora sigas mi ejemplo y nunca te des por vencida en la vida, pues los sueños si uno lo desea se hacen realidad.

Mi mayor agradecimiento a, quienes me enseñaron, mis padres Omar, Maricela y mi padrasto Iraelio que ha sido también un padre para mí. Por darme la oportunidad de nacer junto a ellos. Por los sacrificios que han hecho por mí en tantos años. Por inculcarme principios y valores para empezar el camino de la vida. Por guiar mis estudios, con útiles consejos respecto a mi formación. Por comprenderme y apoyarme en las decisiones que he tomado. Por el ejemplo demostrado en cada situación de la vida. Por las fuerzas que me brindaban para sobreponerme a las dificultades. Por cada momento de tristeza y felicidad, por la unidad y confianza que hemos mantenido y sobre todas las cosas, quisiera agradecerles por el inmenso amor que me han ofrecido, siendo la principal fuente de inspiración en este empeño.

A TODOS, MUCHAS GRACIAS.



INTRODUCCIÓN.....	1
CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA	6
1.1 La enseñanza virtual. Introducción al e-learning.....	6
1.2 Estándares en el e-learning.....	7
1.2.1 Instructional Management Systems (IMS).....	8
1.3 Objetos de aprendizaje.....	10
1.4 Herramientas de autor y sus actividades.....	11
1.4.1 ExeLearning.....	12
1.4.2 Udutu.....	13
1.4.3 CourseLab	14
1.4.4 JClic.....	15
1.5 Importancia del uso de las actividades.....	17
1.6 Tecnologías y herramientas para el desarrollo.....	17
1.6.1 Metodologías de desarrollo de software	18
Selección de la metodología de desarrollo de software.....	21
1.6.2 Herramientas CASE.....	22
Visual Paradigm para UML. Versión 6.4	22
ArgoUML. Versión 0.32	22
1.7 Framework de desarrollo	23
1.8 Lenguajes de desarrollo.....	24
Lenguaje de Marcas Extensibles (XML). Versión 1.0.....	24
Hojas de Estilo en Cascada (CSS). Versión 2.0.....	25
1.9 Sistemas Gestores de Bases de Datos (SGBD).....	26
PostgreSQL. Versión 8.4.....	26



Oracle. Versión 11.2.0.1	27
eXist-db. Versión 1.4	27
1.10 Servidores web.....	28
Apache. Versión 2.2.17	28
Roxen. Versión 3.3.63	28
Conclusiones parciales.....	29
CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA	30
2.1 Modelo de dominio	30
2.2 Requisitos del sistema	33
Requisitos funcionales del sistema	33
Requisitos no funcionales del sistema.....	35
2.3 Definición de actores y casos de uso del sistema	38
2.4 Patrones de casos de uso.....	38
2.5 Diagrama de casos de usos identificados.....	40
2.6 Descripción textual de los casos de uso.....	40
Conclusiones parciales.....	49
CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA	50
3.1 Modelo de Análisis.....	50
3.1.1 Diagrama de clases del análisis.....	50
3.1.2 Diagramas de colaboración.....	52
3.2 Arquitectura propuesta	56
3.2.1 Patrón arquitectónico.....	56
Patrón Modelo-Vista-Controlador.....	57



3.4 Modelo del diseño.....	58
Diagramas de clases del diseño.....	58
Conclusiones parciales.....	61
CONCLUSIONES GENERALES.....	62
RECOMENDACIONES.....	63
REFERENCIAS BIBLIOGRÁFICAS	64
BIBLIOGRAFÍA CONSULTADA	67
GLOSARIO DE TÉRMINOS	¡Error! Marcador no definido.
ANEXOS	¡Error! Marcador no definido.



Imagen 1: Ejemplo de GapMatchInteraction	9
Imagen 2: Ejemplo de extendedTextInteraction.....	10
Imagen 3. Galería de imágenes	12
Imagen 4. Ejercicio de verdadero o falso y área de edición de la galería de imágenes	13
Imagen 5. Uso de archivos multimedia	14
Imagen 6. Área de creación de animaciones	15
Imagen 7. Puzzle y juego de memoria de la herramienta JClic.....	16
Imagen 8: Características de RUP	19
Imagen 9. Flujos de trabajo y fases de la metodología RUP.....	20
Imagen 10. Modelo de dominio del módulo Actividades.....	31
Imagen 11: Patrón concordancia-reusabilidad.....	39
Imagen 12: Patrón concordancia-adición.....	39
Imagen 13: Patrón CRUD-parcial.....	39
Imagen 14. Diagrama de casos de uso del sistema.....	40
Imagen 15: Diagrama de clases del análisis CU Crear sopa de palabras	52
Imagen 16: Diagrama de clases del análisis CU Crear pregunta de redacción guiada	52
Imagen 17: Diagrama de clases del análisis CU Crear asociación simple texto-imagen	52
Imagen 18: Diagrama de colaboración CU Crear sopa de palabras	53
Imagen 19: Diagrama de colaboración CU Gestionar galería de imágenes Sección Eliminar.....	53
Imagen 20: Diagrama de colaboración CU Gestionar galería de imágenes Sección Editar	54
Imagen 21: Patrón Modelo-Vista-Controlador.....	57
Imagen 22: Diagrama de clases del diseño con estereotipos web CU redacción guiada.	59
Imagen 23: Diagrama de clases del diseño con estereotipos web CU Crear asociación simple texto-imagen.....	59
Imagen 24: Diagrama de clases del diseño con estereotipos web CU Crear crucigramas.....	60
Imagen 25: Diagrama de clases del diseño con estereotipos web CU Crear sopa de palabras	60
Imagen 26: Diagrama de clases del diseño con estereotipos web CU Gestionar Galería de imágenes	61

RESUMEN

Con la incorporación de las Tecnologías de la Información y las Comunicaciones (TICs) al proceso de enseñanza-aprendizaje se evidenciaron avances notables, entre ellos el surgimiento del e-learning. A raíz de dicho surgimiento se incrementa de manera gradual la necesidad de compartir recursos y reutilizarlos en cualquier área del conocimiento, surgiendo así los Objetos de Aprendizaje (OA). Con el objetivo de apoyar la teleformación en la Universidad de las Ciencias Informáticas (UCI), se desarrolló en el Centro de Tecnologías para la Formación (FORTES) la herramienta de autor web CRODA, la cual posibilita la creación y edición de Objetos de Aprendizaje de forma flexible haciendo uso de estándares dentro de los que se encuentran: SCORM, LOM e IMS-QTI. Sin embargo, esta herramienta no presenta un suficiente número de actividades que les permita a los profesores desarrollar el proceso de autoevaluación de los estudiantes. La presente investigación asume como idea central, realizar el análisis y diseño de nuevos componentes, que permitan crear las bases para la implementación de nuevas funcionalidades, con el fin de facilitar la creación de ejercicios correspondientes a la sección de autoevaluación del OA, dentro de la herramienta de autor web CRODA.

Palabras clave: actividades, e-learning, herramienta de autor, objetos de aprendizaje, teleformación.

INTRODUCCIÓN

El avance de la ciencia y la tecnología ha provocado en la actualidad grandes cambios en diversas áreas, tal es el caso del ámbito educativo. Han sido muchas las novedades diseñadas, relacionadas con la forma en que los profesores imparten sus clases y las evalúan. Una de las nuevas modalidades de la pedagogía recae en la educación a distancia, la que ha evolucionado a la par de las Tecnologías de la Información y las Comunicaciones (TICs); proceso que se desarrolla de una forma acelerada en las últimas décadas.

Diferentes medios como son: la radio, la televisión, las grabaciones sonoras y más recientes, los servicios de Internet, han logrado que la educación a distancia sea una alternativa viable para dar solución a quienes por limitantes geográficas, físicas u ocupacionales, no puedan asistir a los cursos en las escuelas presenciales tradicionales.

Con este fin, la web se ha introducido rápidamente como un recurso de apoyo a los esquemas tradicionales de la enseñanza presencial o a distancia. La vinculación de las tecnologías en el proceso de enseñanza-aprendizaje ha dado paso al surgimiento del e-learning o “aprendizaje electrónico”, y como tal, puede comprender cualquier actividad educativa que utilice medios electrónicos para realizar el proceso formativo. El e-learning no sustituye el método tradicional de enseñanza, pero si ayuda en gran medida al traer consigo mayor interacción o participación del estudiante en las actividades propuestas.

A raíz de las potencialidades que brinda el surgimiento del e-learning, muchas fueron las instituciones educativas que se interesaron en su investigación. Entre estas se encuentra la Universidad de las Ciencias Informáticas (UCI); la cual aplica un modelo de formación caracterizado por encuentros semi-presenciales, utilizando para ello los recursos didácticos que se encuentran publicados en la misma. Todo esto conlleva a que se eleve la exigencia en la creación de proyectos productivos con fines educativos, logrando que se establezca el Centro de Tecnologías para la Formación (FORTES).

Uno de los proyectos que forma parte de este centro es CRODA, el cual se centra en el desarrollo de una herramienta de autor web, la cual permite la creación de objetos de aprendizaje (OA) reutilizables,



accesibles, duraderos e interoperables, de forma flexible. Para ello hace uso de los estándares Sharable Content Object Reference Model (SCORM) y Learning Object Metadata (LOM). Esta herramienta permitirá a los profesores según su nivel de preparación académica y científico-metodológica, el diseño de OA que permitan atraer la atención de sus estudiantes.

El Módulo Actividades de la herramienta de autor web CRODA en su versión 1.0; permite la creación de cuestionarios que contienen diferentes tipologías de ejercicios. Dicho módulo tenía como limitante la incorporación de elementos multimedia que facilitará la comprensión de los contenidos estudiados, además del insuficiente número de tipologías de ejercicios, que provoca la falta de diversidad en las actividades para cubrir las distintas habilidades de una determinada disciplina, así como para llevar a cabo el proceso de autoevaluación de los conocimientos adquiridos en el uso del objeto de aprendizaje.

En este sentido, se plantea la necesidad de adicionar todos estos elementos a la herramienta de autor web CRODA en su versión 2.0. Un aspecto esencial en el desarrollo de esta versión es la incorporación de nuevas funcionalidades que permitan la interoperabilidad de la sección de autoevaluación en el OA.

De aquí la necesidad de realizar el análisis y diseño de nuevas funcionalidades para el Módulo Actividades de la herramienta de autor web CRODA en su versión 2.0, y de esta forma sentar las bases al proceso de implementación.

Por lo anteriormente expuesto se plantea como **problema de investigación** que no existe la documentación requerida que permita guiar a los desarrolladores en el proceso de implementación del Módulo Actividades, que contribuya a la diversificación de las actividades de autoevaluación que conforman los objetos de aprendizaje.

El problema descrito genera como **objeto de estudio** el proceso de desarrollo de herramientas de autor web.

Para darle solución al problema a resolver mencionado se define como **objetivo general** de la investigación: elaborar el análisis y diseño de las nuevas funcionalidades para el Módulo Actividades en la herramienta de autor web CRODA 2.0.

Definiendo como **campo de acción** el proceso de análisis y diseño de la herramienta de autor web CRODA.

El objetivo general se desglosa en los siguientes **objetivos específicos**:



- Analizar las tendencias y tecnologías existentes relacionadas con las herramientas educativas.
- Realizar el análisis de las nuevas funcionalidades para el Módulo Actividades en la herramienta de autor web CRODA 2.0.
- Diseñar las nuevas funcionalidades para el Módulo Actividades en la herramienta de autor web CRODA 2.0.

La **idea a defender** radica en que, mediante el desarrollo del análisis y diseño de nuevos componentes del módulo Actividades, se crearán las bases para la implementación de un módulo que permita diversificar la autoevaluación del aprendizaje en los OA desarrollados en la herramienta de autor web CRODA 2.0.

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

- Análisis de los conceptos y tecnologías referentes al e-learning para conformar la base teórica metodológica de la investigación.
- Estudio de las tipologías de ejercicios en las herramientas de autor.
- Selección de la herramienta CASE a utilizar.
- Selección de la metodología de desarrollo de software.
- Generación del modelo de dominio, para percibir el entorno real donde se desarrollarán las nuevas funcionalidades.
- Búsqueda y selección de diferentes técnicas de recopilación de información para la captura de requisitos.
- Identificación de los nuevos requisitos funcionales (RF) y no funcionales (RNF).
- Especificación de los actores del sistema.
- Descripción de los casos de uso de las nuevas funcionalidades para el Módulo Actividades en la herramienta de autor web CRODA 2.0.
- Realización de los diagramas de clases del análisis y clases del diseño de las nuevas funcionalidades para el Módulo Actividades en la herramienta de autor web CRODA 2.0.



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

Histórico–Lógico: Para estudiar el origen y evolución de las actividades de auto-evaluación dentro de los OA, así como los estándares y conceptos relacionados.

Analítico–Sintético: Para el análisis de la documentación actual relacionada con el tema, así como las diferentes propuestas y avances existentes en el mundo del e-learning, extrayendo de ello lo necesario para realizar la propuesta del tema en cuestión.

Modelación: Permite la definición y descripción de las funcionalidades que conforman el Módulo Actividades en la herramienta de autor web CRODA 2.0, además la modelación de diseños de interfaz, siendo estas tipologías de ejercicios la base para la integración de las funcionalidades para el módulo propuesto en la herramienta de autor web CRODA 2.0.

Los métodos empíricos empleados en la investigación:

Observación: Permite observar y obtener información sobre cómo se gestionan las actividades en otras herramientas y proyectos existentes, que ayuden a resolver la problemática planteada.

Entrevista: Propició obtener las principales tipologías de ejercicios a incorporar en la nueva versión del Módulo Actividades, así como las diferentes habilidades del aprendizaje de los estudiantes en las cuales enfocar el trabajo.

Estructura capitular

Capítulo 1.- Este capítulo contiene un estudio del estado del arte de las herramientas y tecnologías utilizadas para brindar soluciones a este tipo de problemas, y aborda de forma concreta temas relacionados con el e-learning. Este análisis posibilita la recopilación de información que se utiliza para definir las herramientas y metodologías usadas para el desarrollo del análisis y diseño de las nuevas funcionalidades para el módulo, así como para su implementación en la herramienta de autor web CRODA 2.0.

Capítulo 2.- Este capítulo describe la solución que se propone, realizando primeramente el modelo de dominio para comprender el ámbito donde se desarrolla este módulo. Además se desarrolla el flujo de trabajo Requerimientos de la metodología RUP, exponiendo básicamente los requisitos funcionales y no



funcionales del sistema, llegando así a los casos de uso, donde se realiza una descripción textual de los mismos, y se efectúa un diseño de los prototipos de interfaz de usuario de cada caso de uso identificado.

Capítulo3.- Este capítulo se enfoca en el flujo de trabajo Análisis y diseño, donde a través del desarrollo de los artefactos del análisis se consigue una comprensión precisa de los requisitos, refinándolos y estructurándolos, utilizando el lenguaje de los desarrolladores para proporcionar una visión general del sistema. También se desarrollan los artefactos del diseño, se modela el sistema y la arquitectura que da soporte todos los requisitos, incluyendo los no funcionales y las restricciones que se le imponen.

CAPÍTULO 1.

FUNDAMENTACIÓN TEÓRICA

En este capítulo se realiza la fundamentación teórica de la investigación, abordando temas de relevancia que sustentan la investigación y desarrollo de nuevas funcionalidades que permitan favorecer la diversidad de actividades en el diseño de OA, haciendo uso de la herramienta de autor web CRODA 2.0. Se realiza un análisis del estado del arte del tema a investigar, teniendo en cuenta las herramientas existentes que facilitan la incorporación de diferentes tipologías de ejercicios a los OA. Asimismo, se hace un análisis de las herramientas y metodologías a utilizar para el análisis y diseño de las nuevas funcionalidades para el Módulo Actividades en la herramienta de autor web CRODA 2.0.

1.1 La enseñanza virtual. Introducción al e-learning

Las TIC se emplean cada vez más en la creación de entornos de aprendizaje, ejemplo de ello lo constituyen todos los sectores de la educación. Sin embargo, la tecnología en sí no es suficiente. La creación de contenidos de buena calidad es imprescindible para el desarrollo del e-learning.

Algunos autores lo definen como: “un conjunto de aplicaciones y servicios orientados a facilitar la enseñanza a través de Internet/Intranet, que facilitan el acceso a la información y la comunicación con otros participantes” (1).

Otros plantean que el e-learning es la capacitación no presencial que, a través de plataformas tecnológicas, posibilita y flexibiliza el acceso y el tiempo en el proceso de enseñanza-aprendizaje, adecuándolos a las habilidades, necesidades y disponibilidades de cada discente, además de garantizar ambientes de aprendizaje colaborativos mediante el uso de herramientas de comunicación síncrona y asíncrona, potenciando en suma el proceso de gestión basado en competencias (2).

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA



El autor considera esta última definición como la más acertada, teniendo en cuenta que definen la relación del e-learning con términos que tienen gran importancia en el proceso educativo, tal es el caso de las plataformas tecnológicas y las herramientas de comunicación síncrona y asíncrona.

El e-learning constituye una forma de enseñanza, que presenta varias ventajas entre las que se encuentran las siguientes (3):

- **Se alternan diversos métodos de enseñanza:** Los participantes pueden trabajar individualmente o de manera grupal.
- **Permite flexibilidad horaria:** El alumno accede en el momento que dispone de tiempo.
- **Aumenta el número de destinatarios:** Esta modalidad de formación se puede dirigir a una audiencia más amplia.
- **Favorece la interacción:** Los alumnos pueden comunicarse unos con otros y con el tutor, a través de los recursos on-line disponibles en Internet.
- **Disposición de recursos on-line y multimedia:** Internet proporciona acceso instantáneo e ilimitado a una gran cantidad de recursos, como textos, gráficos, videos y animaciones.

1.2 Estándares en el e-learning

Un estándar se define como un conjunto de elementos, que propone un grupo u organismo reconocido, para describir un determinado dominio o tipo de recursos (4). Estos surgieron con la necesidad de convertir las plataformas del e-learning en plataformas interoperables, es decir, que pudieran comunicarse e integrarse de una forma efectiva aun siendo plataformas diferentes. Diseñar estándares que puedan ser usados en cualquier entorno, es un reto difícil que solo hace unos pocos años comenzó a desarrollarse, sin embargo existen actualmente organizaciones y proyectos que se han enfrascado en el desarrollo de diferentes trabajos, enfocados en la realización de estándares para aprovechar la ventaja que ofrecen las plataformas del e-learning integradas entre sí.

A continuación se explican algunos de estos grupos y proyectos, abordados por la reconocida investigadora en temas referentes al e-learning Clara López Guzmán, en su tesis de doctorado “Los Repositorios de Objetos de Aprendizaje como soporte a un entorno e-learning”.

CAPÍTULO 1.

FUNDAMENTACIÓN TEÓRICA



1.2.1 Instructional Management Systems (IMS)

Las especificaciones IMS (2004) son el resultado de una activa iniciativa que está desarrollando y proponiendo especificaciones basadas en tecnologías abiertas (XML), para facilitar las actividades de aprendizaje sobre tecnología web, principalmente para el intercambio de contenidos y de información sobre los estudiantes. Es una propuesta ambiciosa que cubre, entre otros rubros, accesibilidad y adaptación del estudiante, la definición de competencias, el empaquetamiento de contenidos, información de agentes del proceso educativo, el diseño del aprendizaje a través de un lenguaje para expresar diferentes modelos pedagógicos, así como la formación de repositorios de contenidos digitales. Las especificaciones disponibles son:

- **IMS Content Packaging.** Provee la funcionalidad para describir y empaquetar materiales de aprendizaje, tales como cursos individuales o una colección de cursos, en paquetes interoperables y distribuibles. Esta especificación direcciona la descripción, estructura y ubicación de materiales de aprendizaje en línea, así como la definición de algunos tipos específicos de contenidos. Los proveedores y desarrolladores de contenidos utilizan este formato para asegurar que sus productos serán compatibles e importables/exportables con cualquier herramienta que soporte esta especificación (5).
- **IMS Digital Repositories Interoperability.** Esta especificación provee recomendaciones para la interoperabilidad de las funciones más comunes entre repositorios. En el nivel más general, define los repositorios digitales como colecciones de recursos con acceso a través de una red, sin conocimiento previo de la estructura de la colección. Los repositorios pueden contener los objetos o los metadatos que los describen y no importa si los objetos y los metadatos se encuentran en diferentes repositorios (6).
- **IMS Learning Resources Meta-Data.** Esta especificación hace más eficiente el proceso de búsqueda y uso de los recursos, ya que proporciona una estructura para los elementos (metadatos) que describen o catalogan los recursos de aprendizaje, incluye también cómo los elementos deben ser usados, representados y organizados. La especificación se basa en la aplicación de LOM (7).

CAPÍTULO 1.

FUNDAMENTACIÓN TEÓRICA



- **IMS Question& Test Interoperability.** Propone la descripción de preguntas y test basándose en el lenguaje estándar XML. Está orientada a permitir la interoperabilidad de contenido entre los sistemas de evaluación. Siendo útil para editores, autoridades de certificación, maestros quienes podrán importar y exportar sus datos entre sistemas compatibles.

Las interacciones a incluir como parte de las nuevas funcionalidades se describen a continuación (8):

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 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"/>

Imagen 1: Ejemplo de GapMatchInteraction

- ❖ **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 posibles opciones de respuesta. Hay que destacar que las opciones posibles son compartidas por todos los huecos. Como posibles respuestas es posible utilizar texto o también es posible utilizar imágenes. Además, es posible restringir el número mínimo y máximo de veces que es utilizada cada una de las posibles opciones del conjunto de respuestas.



WRITING A POSTCARD

Read this postcard from your English pen-friend, Sam.

Here is a postcard of my town. Please send me a postcard from your town. What size is your town? What is the nicest part of your town? Where do you go in the evenings?

Sam.

Write Sam a postcard. Answer the questions. Write 25-35 words.

Imagen 2: Ejemplo de extendedTextInteraction

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

Ambas interacciones serán utilizadas en dos de las principales funcionalidades a incorporar en la nueva versión de la herramienta de autor web CRODA 2.0. Se evidencia su uso en las preguntas de redacción guiada y las preguntas de selección de palabras o frases. Estas funcionalidades, además, son generalmente aceptadas como necesarias en cualquier proceso educativo, ya que permiten la identificación de las áreas con mayor nivel de dificultad para el estudiante. La especificación IMS QTI ha sido ampliamente reconocida en la industria y en el mundo académico, pues potencia la interoperabilidad y la reutilización de los recursos educativos.

1.3 Objetos de aprendizaje

A raíz del surgimiento del e-learning son muchos los usuarios que participan de los entornos virtuales de enseñanza, todos con diferentes objetivos e intereses, así como contenidos con características, objetivos y formatos de todo tipo. La tendencia actual es que éstos contenidos sean reutilizables e intercambiables entre aplicaciones dado el origen de los OA.

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA



En la actualidad no existe una definición única sobre OA. Varios autores se han pronunciado al respecto, como la Dra. Clara López Guzmán en el 2005, al plantear que un OA es cualquier recurso con una intención formativa, compuesto de uno o varios elementos digitales, descritos con metadatos, que pueda ser utilizado y reutilizados dentro de un entorno e-learning. Por su parte, el Instituto de Ingenieros Eléctrico y Electrónicos (IEEE) en el 2001 opina que un OA es una entidad, digital o no digital, que puede ser utilizada, reutilizada y referenciada durante el aprendizaje apoyado con tecnología. Se considera que los objetos de aprendizaje se definen como cualquier entidad, digital o no digital, que puede ser utilizada, reutilizada o referenciada durante el aprendizaje apoyado en la tecnología.

Se ofrecen diversos criterios sobre la definición de OA, para esta investigación se tomará como referencia la definición planteada por el departamento de Teleformación de la Universidad de las Ciencias Informáticas (UCI), teniendo en cuenta que, a diferencia de las definiciones antes citadas, esta expresa con claridad las principales características de los OA de los restantes recursos educativos y tecnologías utilizadas en el desarrollo del proceso de enseñanza-aprendizaje. Este concepto 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 reutilizado en diversas situaciones de enseñanza- aprendizaje.

1.4 Herramientas de autor y sus actividades

Las herramientas de autor fueron desplegadas para la creación de materiales educativos y publicación de cursos. Son aplicaciones que permiten un trabajo constructivista, para generar un entorno de aprendizaje dinámico. Dentro de las funcionalidades que este tipo de herramientas presenta cabe destacar la posibilidad de crear actividades o pequeñas aplicaciones desde la misma.

Actualmente existe una diversidad de herramientas de autor que poseen diferentes características en cuanto a funcionalidades, ejemplo de ellas: eXeLearning, Uduu, CourseLab y JClic. Se mencionan a continuación un grupo de características y funcionalidades de las mismas.

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA



1.4.1 ExeLearning

ExeLearning es una herramienta de autor de código abierto para la creación de recursos educativos. Es de gran utilidad para los docentes, ya que permite construir contenido web didáctico sin necesidad de ser experto en la edición y marcado con XML¹ o HTML², permite exportar contenido como páginas web y los recursos pueden exportarse en formatos de paquete de contenido de IMS y SCORM 1.2.

Esta herramienta tiene una interfaz muy amigable y permite la catalogación de metadatos con el estándar DublinCore de una manera muy simple con elementos de ayuda y una interfaz basada solamente en formularios (9).

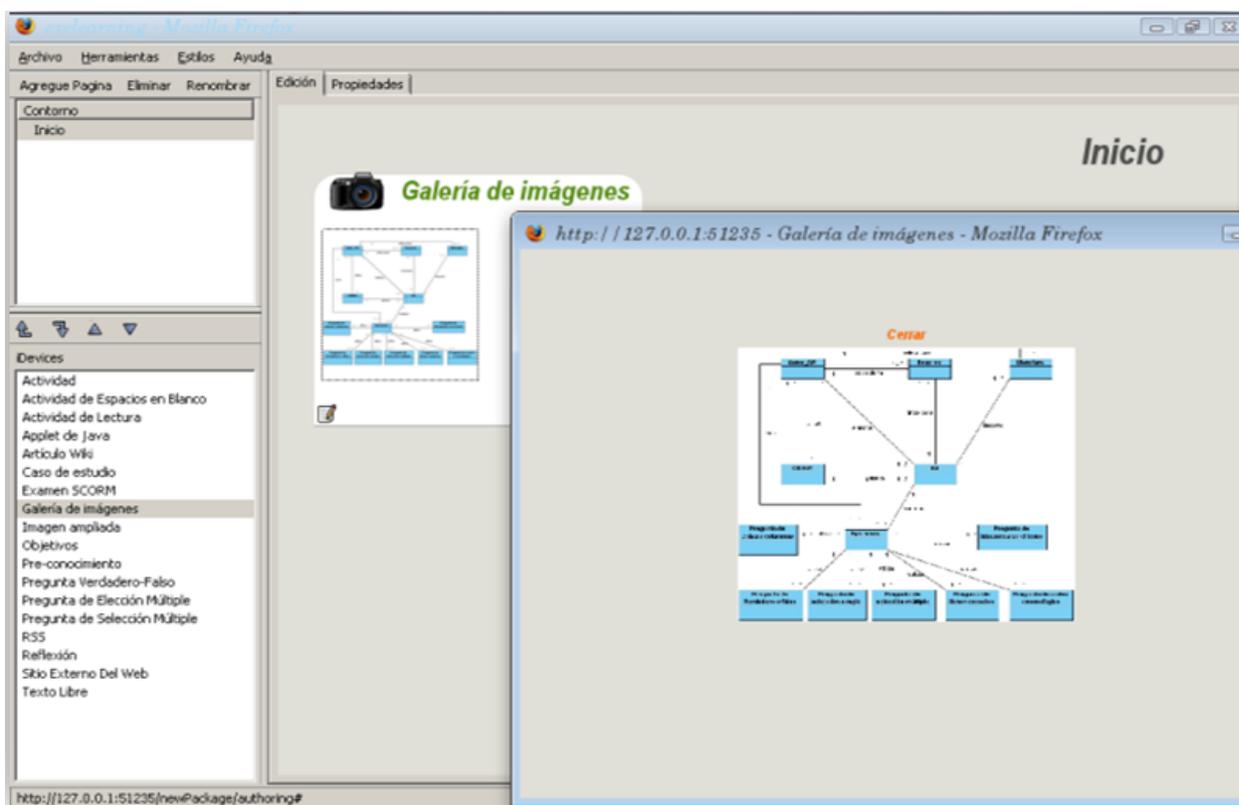


Imagen 3. Galería de imágenes

¹ Lenguaje de marcas extensibles (XML): Es un metalenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium (W3C).

² Protocolo de transferencia de hipertexto (HTTP): Es el protocolo de transferencia de hipertexto usado en cada transacción de la web.

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA



Imagen 4. Ejercicio de verdadero o falso y área de edición de la galería de imágenes

Esta herramienta es muy potente en cuanto a las actividades que le brinda al usuario, entre las que se encuentran las galerías de imágenes. Estas funcionalidades serán incluidas en la nueva versión de la herramienta de autor web CRODA.

1.4.2 Uduu

Uduu es una herramienta de autor web que permite la creación de cursos completamente multimedia, permite añadir contenidos en flash, en audio, en vídeo, documentos, imágenes, ejercicios de autoevaluación y otros contenidos de manera que se pueda hacer uso de ellos en la creación de cursos. Una vez creado el curso, Uduu permite exportarlo a formatos como SCORM 1.2 y SCORM 2004, lo cual lo hace compatible con todos los Learning Management System o Sistemas de Administración del Aprendizaje (LMS). Esta herramienta es libre, permite el trabajo colaborativo y es multiplataforma, sin embargo no permite la descripción de recursos con el empleo de metadatos, siendo esta su principal desventaja, pues los recursos creados, no tendrán descripción por lo que se vería afectado su uso y reutilización (10).

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

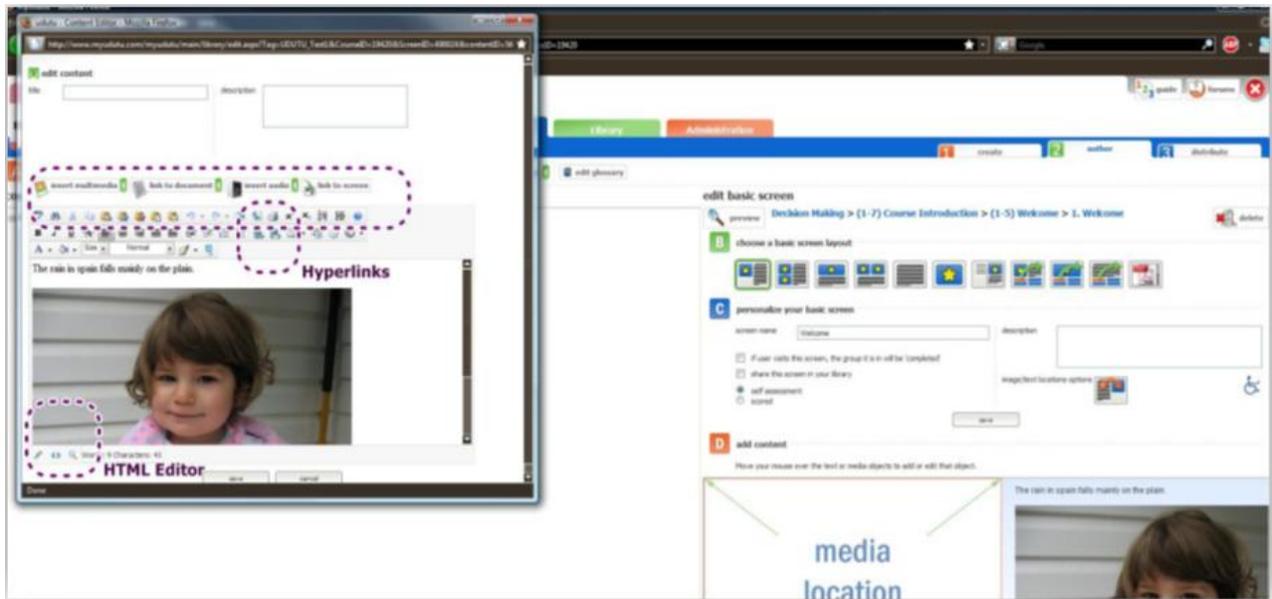


Imagen 5. Uso de archivos multimedia

1.4.3 CourseLab

CourseLab es un programa potente y fácil de usar, orientado a la creación de cursos de e-learning compatible con los estándares internacionales existentes en este ámbito.

CourseLab organiza un curso en módulo de aprendizaje, cada módulo consta de un conjunto de presentaciones que reciben el nombre de diapositivas o slides. A su vez cada diapositiva, consta de un conjunto de planos estáticos denominados frames que se estipulan para inducir el movimiento, la animación que suele incorporar la presentación (11).

Los diferentes objetos que pueden situarse sobre el espacio de la presentación pueden ser sometidos a un conjunto de acciones potentes, pero fáciles de introducir a partir de las cuales son dotados de interactividad. Estos objetos pueden ser texto, imágenes o video, lo que nos permite introducir importantes efectos multimedia.

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

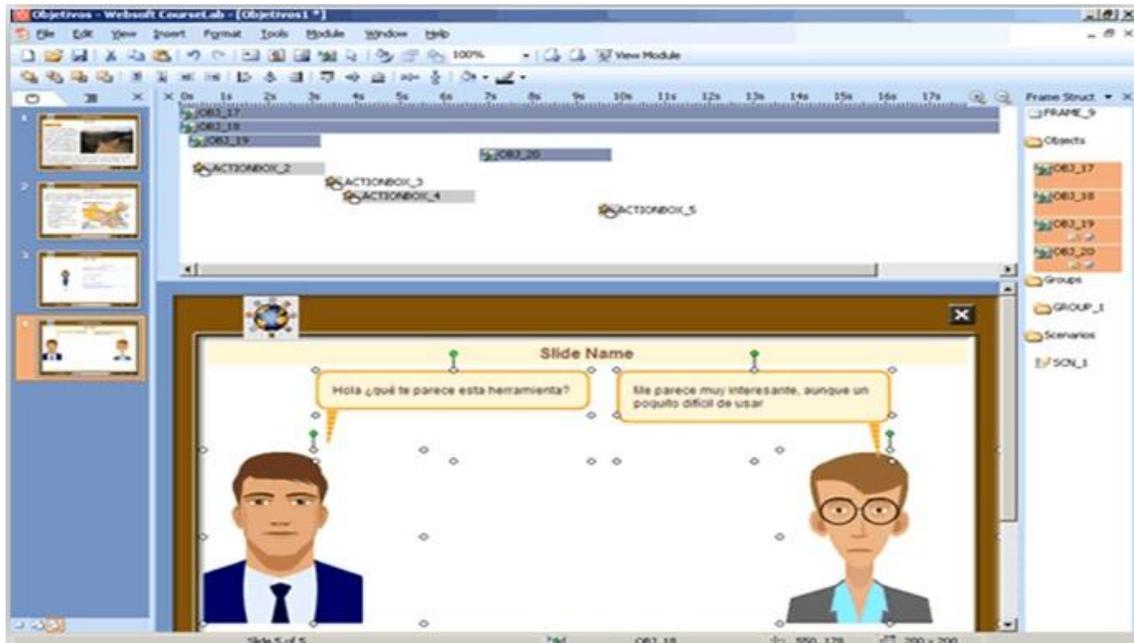


Imagen 6. Área de creación de animaciones

Esta herramienta está orientada a realizar contenidos muy secuenciados, con pantallas de tamaño regular, abundantes animaciones y elementos interactivos lo cual sería conveniente incluirlo en la nueva versión de la herramienta, ya que esto permitiría centrar la atención de los estudiantes y mantenerlos más enfocados en el proceso de autoestudio con el OA.

1.4.4 JCLic

JCLic es una herramienta de autor que permite al profesorado crear con facilidad recursos educativos digitales. Permite crear mayor variedad de actividades, cuenta con nuevas funcionalidades y posibilita crear recursos cuya visualización no está restringida a ningún sistema operativo en particular.

Los objetivos perseguidos por el proyecto JCLic son (12):

- Hacer posible el uso de aplicaciones educativas multimedia "en línea", directamente desde Internet.
- Mantener la compatibilidad con las aplicaciones Clic 3.0 existentes.

CAPÍTULO 1.

FUNDAMENTACIÓN TEÓRICA



- Hacer posible su uso en diversas plataformas y sistemas operativos, como Windows, GNU/Linux y Solaris.
- Utilizar un formato estándar y abierto para el almacenaje de los datos, con el fin de hacerlas transparentes a otras aplicaciones, así como facilitar su integración en bases de datos de recursos.
- Ampliar el ámbito de cooperación e intercambio de materiales entre escuelas y educadores de diferentes países y culturas, facilitando la traducción y adaptación tanto del programa como de las actividades creadas.
- Recoger las sugerencias de mejoras y ampliaciones que los usuarios han ido enviando.
- Hacer posible que el programa pueda ir ampliándose a partir del trabajo cooperativo entre diversos equipos de programación.



Imagen 7. Puzzle y juego de memoria de la herramienta JClic

La herramienta JClic nos ofrece la posibilidad de aumentar el nivel de entretenimiento en las actividades de evaluación, ya que los ejercicios que posee son en forma de puzzle y juegos de memoria, los cuales proporcionan una mayor atracción por parte de los usuarios, pero estos por sí solos no contribuyen al enriquecimiento de la autoevaluación de los estudiantes, por lo que sería conveniente implementarlos en

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA



conjunto de otras tipologías de ejercicios como las que ya están implementadas en la versión 1.0 de la herramienta, y las que se pretenden implementar en la nueva versión.

1.5 Importancia del uso de las actividades

Las actividades en forma de sopas de palabras, crucigramas, puzzles y juegos de memorias contribuyen al desarrollo de las capacidades mentales de los estudiantes. Cada actividad cognitiva exige emplear diferentes tipos de operaciones mentales. A continuación se presentan algunas de las operaciones mentales, a las cuales tributan las actividades antes mencionadas (13).

- **Identificación:** Reconocimiento de la realidad por medio de sus rasgos característicos globales recogidos en un término que la define.
- **Diferenciación:** Reconocimiento de la realidad por sus características, pero distinguiendo las relevantes y las irrelevantes, en cada momento. Se estudian las semejanzas y diferencias entre objetos o hechos. La percepción de los objetos necesita ser clara y estable para poder comparar.
- **Comparación:** Búsqueda de semejanzas y diferencias entre objetos o hechos, de acuerdo con sus características.
- **Clasificación:** Agrupación de objetos de acuerdo con sus atributos comunes. Los criterios de agrupación son variables.
- **Inferencia lógica:** Operación mental que permite realizar deducciones a partir de unas informaciones previas. Es la capacidad para realizar deducciones y crear nueva información a partir de los datos percibidos.
- **Razonamiento lógico:** Todo el desarrollo mental lleva al pensamiento lógico. Este pensamiento lógico formal consiste en la representación de acciones posibles; es el arte del buen pensar; la organización del pensamiento que llega a la verdad lógica, a través de otras formas de pensamiento (inferencial, hipotético, transitivo, silogístico).

1.6 Tecnologías y herramientas para el desarrollo

En el desarrollo del análisis y diseño de las nuevas funcionalidades para el Módulo Actividades presentado, se adoptarán diferentes tecnologías y herramientas. Para la selección de estas, se realizará

CAPÍTULO 1.

FUNDAMENTACIÓN TEÓRICA



un estudio basado en sus características y ventajas. Además, de mantener las políticas de desarrollo del centro FORTES en su actual migración al software libre. Las nuevas funcionalidades que se proponen, serán incorporadas al Módulo Actividades en la herramienta de autor web CRODA 2.0.

1.6.1 Metodologías de desarrollo de software

Las metodologías o procesos de desarrollo de software ofrecen un método para la creación de aplicaciones flexibles y robustas de un modo organizado y disciplinado, facilitando su comprensión. Además, abarcan procedimientos, técnicas, documentación y herramientas que tienen como base fundamental el fortalecimiento de los procesos de desarrollo de un software. Actualmente no existe una metodología universal que le haga frente con éxito a cualquier proyecto de desarrollo de software, ni que se adapte a todo tipo de sistema, debido a que las características y recursos de cada equipo de desarrollo no siempre son las mismas.

1.6.1.1 Metodologías tradicionales

Las metodologías tradicionales imponen una disciplina de trabajo sobre el proceso de desarrollo del software, con el fin de conseguir un software más eficiente. Para ello, se hace énfasis en la planificación total de todo el trabajo a realizar y una vez que está todo detallado, comienza el ciclo de desarrollo del software. Se centran especialmente en el control del proceso, mediante una rigurosa definición de roles, actividades, artefactos, herramientas y notaciones para el modelado y documentación detallada. Además, las metodologías tradicionales no se adaptan adecuadamente a los cambios, por lo que no son métodos adecuados cuando se trabaja en un entorno, donde los requisitos no pueden predecirse o bien pueden variar.

✓ Proceso Unificado de Desarrollo (RUP)

RUP es un proceso de desarrollo de software perteneciente al grupo de metodologías pesadas, como cualquier proceso de desarrollo de software define quién hace, qué, cómo y cuándo. RUP define cuatro elementos: trabajadores (roles), que responden a la pregunta ¿Quién?, las actividades que responden a la pregunta ¿Cómo?, los artefactos (productos), que responden a la pregunta ¿Qué? y los flujos de trabajo de las disciplinas que responde a la pregunta ¿Cuándo? A continuación se detallan estos elementos:

Trabajadores (“quién”): Define el comportamiento y responsabilidades (rol) de un individuo, grupo de individuos, sistema automatizado o máquina, que trabajan en conjunto como un equipo. Ellos realizan las

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA



actividades y son propietarios de elementos.

Actividades (“cómo”): Es una tarea que tiene un propósito claro, es realizada por un trabajador y manipula elementos.

Artefactos (“qué”): Productos tangibles del proyecto que son producidos, modificados y usados por las actividades. Pueden ser modelos, elementos dentro del modelo, código fuente y ejecutables.

Flujo de trabajo (“Cuándo”): Secuencia de actividades realizadas por trabajadores y que produce un resultado de valor observable.

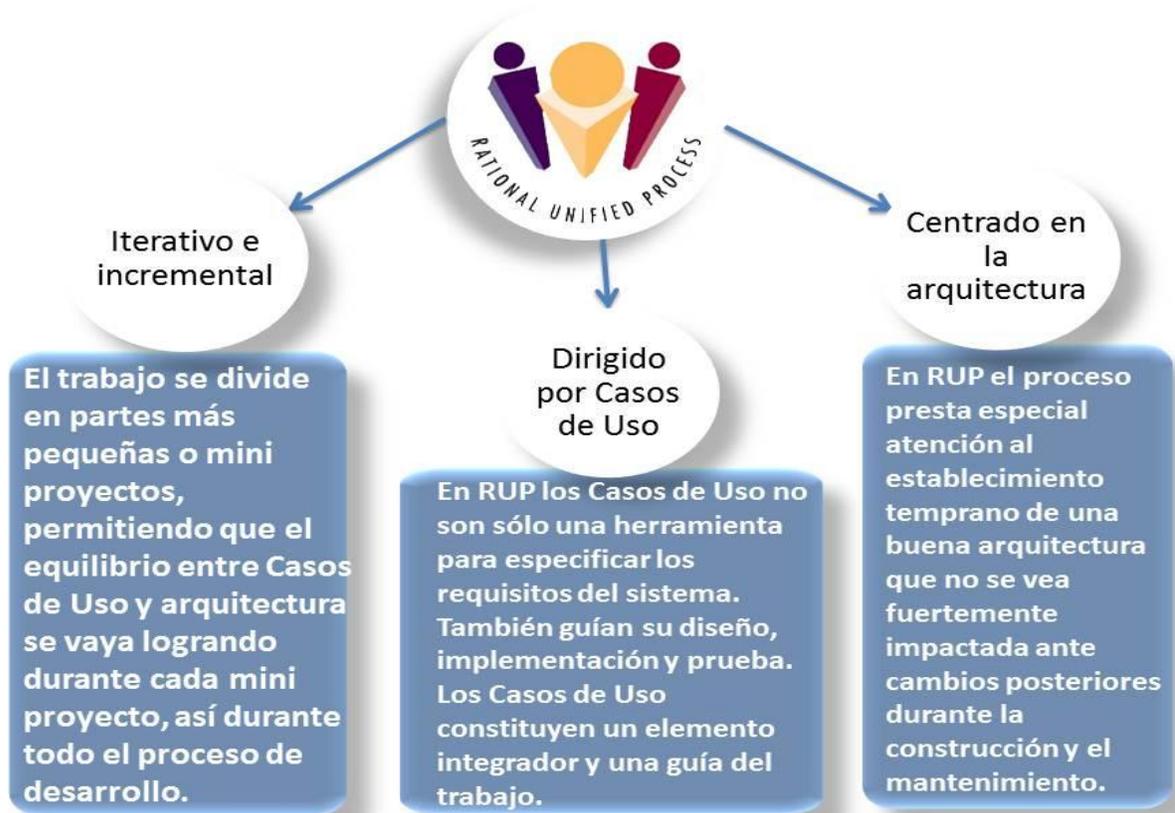


Imagen 8: Características de RUP

RUP divide el proceso en 4 fases, dentro de las cuales se realizan varias iteraciones en número variable según el proyecto y en las que se hace un mayor o menor énfasis en las distintas actividades (14):

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA



- **Inicio:** Se hace un plan de fases, se identifican los principales casos de uso y se identifican los riesgos. Se define el alcance del proyecto.
- **Elaboración:** Se hace un plan de proyecto, se completan los casos de uso y se eliminan los riesgos.
- **Construcción:** Se concentra en la elaboración de un producto totalmente operativo y el manual de usuario.
- **Transición:** Se instala el producto en el cliente y se entrena a los usuarios. Como consecuencia de esto suelen surgir nuevos requisitos a ser analizados.

RUP define 9 flujos de trabajo de los cuales 6 son de ingeniería y los restantes 3 de apoyo, cada flujo está destinado a actividades específicas y están divididos en 4 fases. Estos flujos se desarrollan en paralelo durante todo el ciclo de desarrollo estando presente, unos más que otros en las diferentes iteraciones, en la figura se pueden apreciar los mismos, así como su impacto en cada fase e iteración.

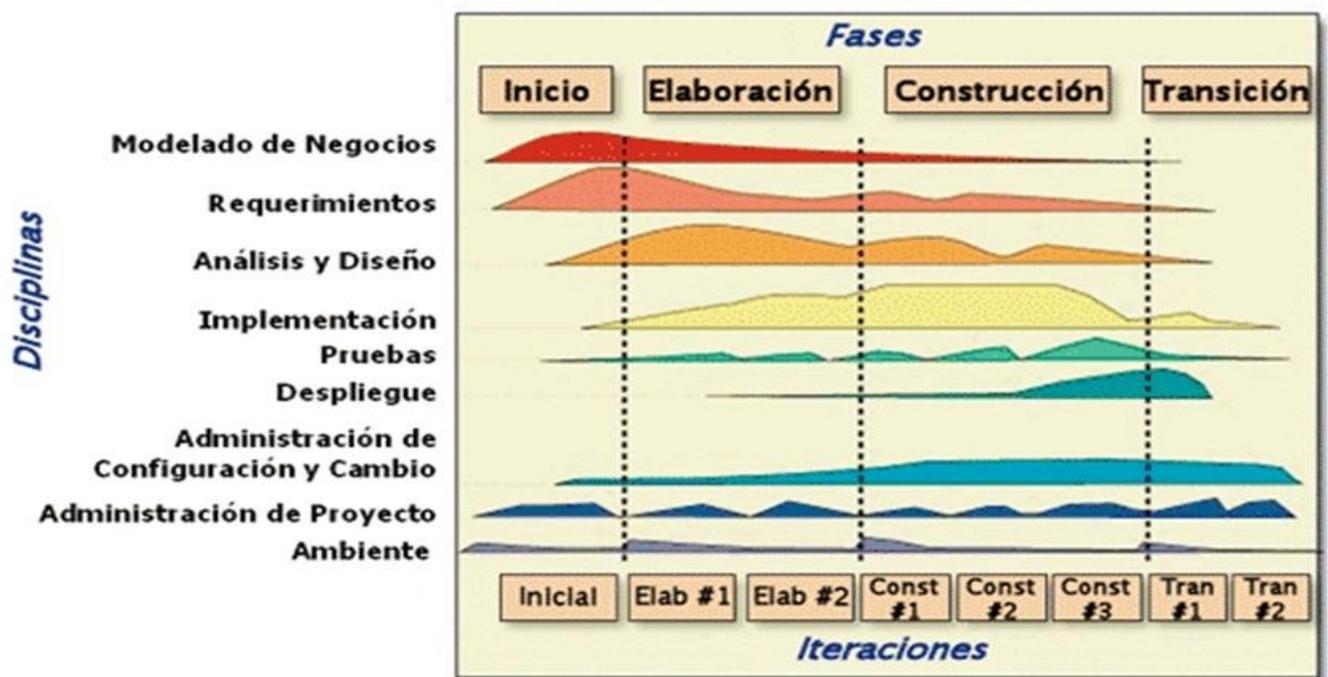


Imagen 9. Flujos de trabajo y fases de la metodología RUP



1.6.1.2 Metodologías ágiles

En febrero de 2001, tras una reunión celebrada en Utah-EEUU, nace el término ágil aplicado al desarrollo de software. En esta reunión participan un grupo de 17 expertos de la industria del software, incluyendo algunos de los creadores o impulsores de metodologías de software. Su objetivo fue esbozar los valores y principios que deberían permitir a los equipos desarrollar software rápidamente y respondiendo a los cambios que puedan surgir a lo largo del proyecto.

✓ Programación Extrema (XP)

XP es una metodología ágil, centrada en potenciar las relaciones interpersonales como clave para el éxito en el desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. Es adecuada para proyectos con requisitos imprecisos y muy cambiantes, donde existe un alto riesgo técnico. Las principales características de XP se definen en cuatro apartados: historias de usuario, roles, proceso y prácticas.

Características de XP:

- **Pruebas Unitarias:** se basa en las pruebas realizadas a los principales procesos, de tal manera que el adelantarse en algo hacia el futuro, se puedan hacer pruebas de las fallas que pudieran ocurrir. Es como si pudieran obtenerse los posibles errores.
- **Refabricación:** se basa en la reutilización de código, para lo cual se crean patrones o modelos estándares, siendo más flexible al cambio.
- **Programación en pares:** una particularidad de esta metodología es que propone la programación en pares, la cual consiste en que dos desarrolladores participen en un proyecto en una misma estación de trabajo. Cada miembro lleva a cabo la acción que el otro no está haciendo en ese momento. Es como el chofer y el copiloto: mientras uno conduce, el otro consulta el mapa (15).

Selección de la metodología de desarrollo de software

XP es una metodología que establece que tanto cliente, como jefe de proyecto y desarrolladores son partes del equipo de trabajo, además se encuentran implicados en el desarrollo del software. En este caso el cliente no forma parte del grupo de desarrollo, por lo que usar esta metodología sería incumplir con un requisito fundamental que esta exige para que el software tenga la calidad requerida. XP está destinada a

CAPÍTULO 1.

FUNDAMENTACIÓN TEÓRICA



proyectos de corto plazo y se basan en iteraciones pequeñas a diferencia de RUP, ya que esta última metodología está concebida para la realización de proyectos y equipos de trabajo grandes, en cuanto a tamaño y duración. En el caso de RUP mantiene al equipo de trabajo enfocado en producir software en tiempo, con las características y calidad requerida. Basa su trabajo principalmente en la documentación del software y expone un conjunto de actividades orientadas a visualizar, especificar, construir y documentar los artefactos necesarios para el desarrollo de aplicaciones. Por todas estas razones se considera la elección de RUP como metodología que guíe el proceso de desarrollo del Módulo Actividades en su versión 2.0.

1.6.2 Herramientas CASE

Las herramientas CASE (Computer Aided Software Engineering, Ingeniería de Software Asistida por Computadora) son aquellas que nos brindan la facilidad de crear los artefactos necesarios siguiendo una metodología en la construcción de un software. Al ser modelados los artefactos con estas herramientas se hace más fácil realizar los cambios que ocurren en el diseño del software, además de permitir desarrollar un software con mucha más calidad y rapidez. A continuación se describen dos de ellas.

- ✓ **Visual Paradigm para UML. Versión 6.4:** Es una herramienta CASE que soporta todo el ciclo de vida del desarrollo de un software: Análisis y diseño, Construcción, Pruebas y Despliegue. Además permite la construcción de todos los diagramas de clases, casos de uso, diagramas de actividades, genera código desde los diagramas, y posibilita el diseño de prototipos de interfaz de usuario. Proporciona además diferentes tutoriales que permiten un mejor entendimiento de la herramienta por parte de los usuarios. A su vez, permite el diseño de software con UML y posibilita la captura de requisitos con sus respectivos diagramas. En general, proporciona un entorno unificado de diseño de software para analizar, diseñar y mantener aplicaciones de software en una disciplina. Es una herramienta que está disponible en múltiples sistemas operativos como Windows, Linux y Unix y es muy fácil su uso, siendo estas sus principales ventajas (16).
- ✓ **ArgoUML. Versión 0.32:** Es una aplicación para la realización de diagramas UML, escrita en Java

CAPÍTULO 1.

FUNDAMENTACIÓN TEÓRICA



y publicada bajo la Licencia BSD³. Entre sus características más atractivas se encuentran que incluye soporte para los diagramas basados en UML 1.4, ingeniería inversa, además de disposición (layout) automática del diagrama de clases, así como la generación de ficheros PNG, GIF, JPG, SVG, EPS desde diagramas, internacionalización y soporte para el lenguaje de generación de código: Java, PHP, Python, C++ y Csharp (C#).

Visual Paradigm constituye un candidato más fuerte para la elección de una herramienta CASE, teniendo en cuenta que soporta la generación de los principales diagramas establecidos, que constituyen los artefactos requeridos por la metodología de desarrollo seleccionada. Es más integrable con el resto de las herramientas y estándares escogidos. Por otra parte, la constante sincronización del modelo de diseño y el código fuente durante todo el ciclo de desarrollo del software, además de su excelente generación de código y soporte para UML 2.1 realzan su potencial. Al mismo tiempo, la Universidad pagó por su licencia y tiene como política que sea la herramienta case a utilizar.

1.7 Framework de desarrollo

En el desarrollo de software, el framework constituye la estructura conceptual y tecnológica que está definida con artefactos o módulos de software concretos, en base a la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, puede incluir soporte de programas, bibliotecas y un lenguaje interpretado, entre otros programas para ayudar a desarrollar y unir los diferentes componentes de un proyecto.

- ✓ **Symfony. Versión 1.4.9:** Es un framework PHP que facilita el desarrollo de las aplicaciones web. Symfony es, además, el framework más documentado del mundo, ya que cuenta con miles de páginas de documentación distribuidas en varios libros gratuitos y decenas de tutoriales. Se diseñó para que se ajustara a los siguientes requisitos (17):
 - Fácil de instalar y configurar en la mayoría de plataformas (y con la garantía de que funciona correctamente en los sistemas Windows y Unix-like (*nix) estándares).
 - Independiente del sistema gestor de bases de datos.

³ Esta licencia permite distribuir el código del programa o el ejecutable, indicando que está amparado por la licencia BSD y que en caso de crear una aplicación derivada de otra con esta licencia, **no se puede usar el nombre del autor original para promover tu creación.**

CAPÍTULO 1.

FUNDAMENTACIÓN TEÓRICA



- Sencillo de usar en la mayoría de casos, pero lo suficientemente flexible como para adaptarse a los casos más complejos.
- Código fácil de leer que incluye comentarios de phpDocumentor⁴ y que permite un mantenimiento muy sencillo.
- Fácil de extender, lo que permite su integración con librerías desarrolladas por terceros.

Symfony tiene numerosas características tales como su publicación bajo una licencia de software libre, la amplia documentación existente, su creación sólo para PHP 5, para obtener el máximo rendimiento de PHP y aprovechar todas sus características.

- ✓ **Zend Framework. Versión 1.11.3:** Se trata de un framework para el desarrollo de aplicaciones y servicios web con Hypertext Pre-processor (PHP), brinda soluciones para construir sitios web modernos, robustos y seguros. Además, implementa el patrón Modelo-Vista-Controlador (MVC), es orientado a objetos y sus componentes tienen un bajo acoplamiento por lo que se puede usar en forma independiente. También este framework brinda un estándar de codificación.

Para llevar a cabo el proceso de desarrollo de las nuevas funcionalidades para el Módulo Actividades de la versión 2.0, se propone el framework Symfony debido a que la herramienta de autor web CRODA lo utiliza para llevar a cabo su proceso de desarrollo, además de esta manera se garantiza una mejor organización y no provocará cambios significativos en la arquitectura de la herramienta.

1.8 Lenguajes de desarrollo

Un lenguaje de programación es una serie de comandos que permiten codificar instrucciones de manera que sean entendidas y ejecutadas por una computadora.

- ✓ **Lenguaje de Marcas Extensibles (XML). Versión 1.0:** Se propone el uso del lenguaje XML, teniendo en cuenta que las interacciones de IMS-QTI se describen empleando este lenguaje, el mismo se define como: 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. Su principal característica es que no posee etiquetas definidas desde un principio, por lo que el propio autor las define como desea.

⁴ **phpDocumentor:** Es un generador de documentación de código abierto escrito en PHP.

CAPÍTULO 1.

FUNDAMENTACIÓN TEÓRICA



La codificación en XML de las iteraciones IMS-QTI es fundamental, ya que proporcionará una mayor interoperabilidad a las actividades entre diferentes LMS (18).

- ✓ **Lenguaje Extensible de Marcado de Hipertexto (XHTML). Versión 1.0:** Según la W3C⁵ —XHTML (Lenguaje de Marcado de Hipertexto Extensible) es una versión más estricta y limpia de HTML, que nace precisamente con el objetivo de reemplazar a HTML ante su limitación de uso con las abundantes herramientas basadas en XML, XHTML extiende HTML 4.0 combinando la sintaxis de HTML, diseñado para mostrar datos, con la de XML, diseñado para describir los datos.
- ✓ **Hojas de Estilo en Cascada (CSS). Versión 2.0:** Es un lenguaje usado para definir la presentación de un documento estructurado escrito en HTML. Algunas ventajas de utilizar CSS son (19):
 - Control centralizado de la presentación de un sitio web completo, con lo que se agiliza de forma considerable la actualización del mismo.
 - Una página puede disponer de diferentes hojas de estilo según el dispositivo que la muestre o, incluso, a elección del usuario.
 - El documento HTML en sí mismo es más claro de entender, y se consigue reducir considerablemente su tamaño.
- ✓ **JavaScript. Versión 1.5:** Según Mozilla Developer Network (MDN), “JavaScript es un lenguaje de script multiplataforma. Es un lenguaje pequeño y ligero; no es útil como un lenguaje independiente, más bien está diseñado para una fácil incrustación en otros productos y aplicaciones, tales como los navegadores web. (...)”
 - **ExtJs. Versión 3.4:** Es una librería desarrollada en JavaScript, que será usada en el desarrollo del módulo propuesto, pues permite el desarrollo de aplicaciones web interactivas empleando tecnologías como Ajax⁶, DHTML y DOM⁷. ExtJs proporciona una

⁵ El World Wide Web Consortium, abreviado W3C, es un consorcio internacional que produce recomendaciones para la World Wide Web.

⁶ Ajax, acrónimo de Asynchronous JavaScript And XML (JavaScript asíncrono y XML), es una técnica de desarrollo web para crear aplicaciones interactivas.

⁷ Modelo de Objetos del Documento: Es esencialmente una interfaz de programación de aplicaciones (API) que proporciona un conjunto estándar de objetos para representar documentos HTML y XML.

CAPÍTULO 1.

FUNDAMENTACIÓN TEÓRICA



interfaz de usuario enriquecida, muy parecida a las que se encuentran en las aplicaciones de escritorio. Esto permite a los desarrolladores web concentrarse en las funcionalidades de las aplicaciones y no en los detalles técnicos. Puede trabajar en conjunto con otras librerías si se definen adaptadores para las mismas, incluye componentes de interfaz de usuario personalizables. Cuenta con un modelo de componentes extensibles, una interfaz de programación de aplicaciones (API) fácil de usar y licencias Open Source (GPL) y comerciales. La librería ExtJS cuenta con un conjunto de componentes para incluir dentro de una aplicación web, como: cuadros y áreas de texto, radio buttons y check boxes, editor HTML, árbol de datos, pestañas y paneles divisibles en secciones (20).

- ✓ **PHP. Versión 5.3.5:** Presenta capacidad de conexión con la mayoría de los motores de base de datos, destacándose principalmente en MySQL y PostgreSQL. Es el empleado actualmente por el equipo de desarrollo de la herramienta de autor web CRODA 2.0. Es un lenguaje multiplataforma, está orientado al desarrollo de aplicaciones web manteniendo acceso a información almacenada en una base de datos, el código fuente escrito en PHP es transparente al navegador y al usuario, haciendo la programación segura y confiable (21).

1.9 Sistemas Gestores de Bases de Datos (SGBD)

Los SGBD son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. Tienen a su vez un propósito general que es el de manejar de manera clara, sencilla y ordenada un conjunto de información. Además, poseen un objetivo definido que es el de brindar a los usuarios una herramienta que les permita de forma general, manejar los datos de una base de datos exentos de conocer la forma en que se almacenan o se accede a los mismos.

- ✓ **PostgreSQL. Versión 8.4:** Es un potente servidor de base de datos relacional, orientado a objetos de uso libre y de código abierto, incluye extensa documentación. Se ejecuta en la mayoría de los sistemas operativos más utilizados en el mundo incluyendo Linux, varias versiones de UNIX y en Windows. A continuación, se hace referencia a otras características del gestor PostgreSQL (22):
 - Cuenta con comunidades muy activas, además muchas de ellas en idioma español.
 - Es altamente adaptable a las necesidades del cliente.

CAPÍTULO 1.

FUNDAMENTACIÓN TEÓRICA



- Presenta un soporte nativo para los lenguajes más populares del medio: PHP, C, Perl, Python.
 - Soporta todas las características de una base de datos profesional (disparadores, procedimientos almacenados, funciones, secuencias, relaciones, reglas, tipos de datos definidos por usuarios).
- ✓ **Oracle. Versión 11.2.0.1:** Fabricado por Oracle Corporation, utiliza la arquitectura cliente/servidor. Ha incorporado en su sistema el modelo objeto-relacional, pero al mismo tiempo garantiza la compatibilidad con el tradicional modelo relacional de datos, así ofrece un servidor de bases de datos híbrido. Es uno de los más conocidos y ha alcanzado un buen nivel de madurez y profesionalidad. Es robusto y muy completo, cuenta con características como su estabilidad y escalabilidad, además de ser multiplataforma (23).

Sin embargo, Oracle no cuenta con una licencia libre, la gran potencia que tiene y su elevado precio hacen que sólo se utilice en empresas muy grandes y multinacionales. Todas estas desventajas llevan a Oracle a no ser muy extendido como otros sistemas gestores de base de datos, y por tanto no se propone para el desarrollo futuro de este módulo. Por todas las características y numerosas ventajas de PostgreSQL, se propone como gestor de base de datos a utilizar en la incorporación de las nuevas funcionalidades al Módulo Actividades de la herramienta de autor web CRODA 2.0.

- ✓ **eXist-db. Versión 1.4:** El gestor de base de datos XML posibilita guardar un XML en su estructura original, y realizarle consultas para la obtención de información o modificación de su contenido. Debido a que las estructuras de un OA están expresadas en ficheros XML, llevarlas a una base de datos relacional traería varios inconvenientes como garantizar que se mantenga la integridad entre el esquema y la estructura física del XML. Es libre y de código abierto. A diferencia de la mayoría de los sistemas de gestión de base de datos relacionales, eXist-db soporta distintos lenguajes de consultas XML. Además, algunas de sus características más destacadas son la indexación de documentos, el soporte para la actualización de datos, y la capacidad de obtener los resultados de las consultas en formato XML. Finalmente, el acceso a eXist-db se realiza mediante una interfaz gráfica en entorno web, que facilita la gestión y comunicación del mismo (24).

CAPÍTULO 1.

FUNDAMENTACIÓN TEÓRICA



1.10 Servidores web

Los servidores web son grandes proveedores de información para todo tipo de usuarios. Estos surgieron por la necesidad que tenían o requerían algunas empresas de compartir información con algún grupo de clientes. Su principal tarea es alojar sitios y/o aplicaciones, las cuales son accedidas por los clientes utilizando un navegador que se comunica con el servidor utilizando el protocolo HyperText Transfer Protocol (HTTP⁸).

- ✓ **Apache. Versión 2.2.17:** Es un servidor web robusto cuya implementación se realiza de forma colaborativa. Entre sus principales características se encuentra el spelling, la cual es una prestación que permite definir una página de error para los recursos no encontrados que sugiera al usuario algunos nombres de recurso parecidos, al que solicitaba para el caso de que hubiese cometido un error al escribir. El status proporciona una página web, generada por el servidor donde éste muestra su estado de funcionamiento y nivel de respuesta. Además, Apache es modular, de código abierto, multi-plataforma y posee facilidades para conseguir ayuda/soporte (25).
- ✓ **Roxen. Versión 3.3.63:** Es un servidor web de licencia GNU, desarrollado por un grupo sueco que después fundaría la empresa Roxen Internet Services. Roxen (que antes se llamó Spider y después, Spinner) destaca por su gran cantidad de funcionalidades. Este servidor, desarrollado en el lenguaje Pike, ofrece cientos de módulos que permiten el desarrollo sencillo de sitios web muy ricos y dinámicos, sin más herramienta que el servidor Roxen. Sus características más destacables son (26):
 - Multiplataforma: puede ejecutarse en Windows, Linux, MAC OS/X y Solaris.
 - Código libre.
 - Interfaz de administración basada en web, completa y fácil de usar.

Ambas alternativas son buenas, aunque se selecciona Apache por su alta capacidad de configuración y robustez, siendo además casi universal al estar en una multitud de sistemas operativos. Resaltar también que, Apache presenta un diseño modular excelente para ampliar las capacidades del servidor web,

⁸ HyperText Transfer Protocol (HTTP): Es el protocolo de transferencia de hipertexto usado en cada transacción de la web.

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA



contando además, con una amplia variedad de módulos que se pueden instalar a medida que se necesiten.

Conclusiones parciales

Como resultado de la investigación y el análisis bibliográfico realizado, a lo largo de este capítulo, han sido expuestos los principales puntos de interés relacionados con los OA, los estándares utilizados durante la creación de las actividades, así como los conceptos y tecnologías en el e-learning. Resultaron seleccionadas la metodología de desarrollo de software RUP y la herramienta CASE Visual Paradigm, de acuerdo a las características estudiadas. Se realizó una propuesta de las tecnologías y herramientas necesarias para su futura implementación de las nuevas funcionalidades.

CAPÍTULO 2.

CARACTERÍSTICAS DEL SISTEMA

Este capítulo constituye la propuesta de la solución para contribuir con la diversificación de las funcionalidades del Módulo Actividades para la herramienta de autor web CRODA 2.0. Para la descripción de la solución, se desarrolla el modelo de dominio, describiendo las clases, así como la relación que existe entre ellas. Se realiza la especificación de requisitos funcionales y no funcionales que debe presentar el sistema, además se identifican y describen los casos de uso del sistema y los actores que los inicializan. De forma general, el capítulo recoge el flujo de trabajo Requerimientos de la metodología RUP analizada en el capítulo anterior.

2.1 Modelo de dominio

El flujo de trabajo Modelamiento del negocio determina el funcionamiento del negocio. Es un modelo real de las personas, grupos o entidades que se benefician de la actividad desarrollada en el negocio (actores del negocio) y la interacción con las personas que realizan estas actividades (trabajadores del negocio). En este flujo se describen las entidades, actividades y procesos (casos de uso del negocio) que se realizan, para entender las implicaciones y ámbito del negocio, que sirva de guía en la realización de los restantes flujos de trabajo.

En el estudio realizado no fue posible identificar procesos del negocio, ya que está altamente centrado en tecnologías informáticas, pues el objetivo de este módulo es la gestión de la descripción de actividades de autoevaluación, haciéndose difícil determinar actores del negocio y procesos de negocio, por lo que se hace necesaria la realización del modelo de dominio.

CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA



El modelo de dominio es una representación visual de los principales conceptos u objetos del mundo real, significativos para un problema o área de interés. Es de gran ayuda para desarrolladores y usuarios, ya que de esta forma utilizan un vocabulario común y pueden entender el contexto en que se enmarca el sistema.

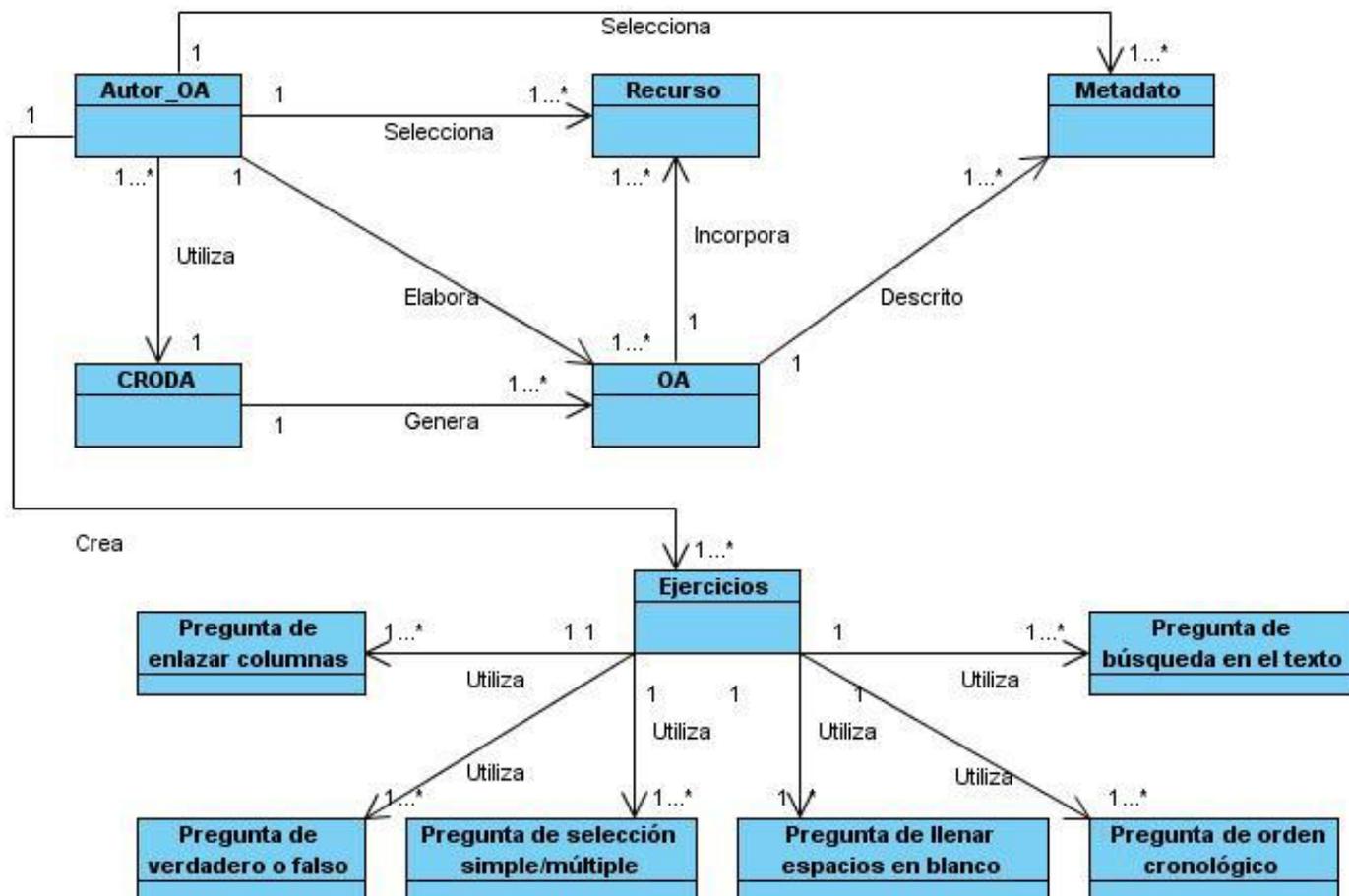


Imagen 10. Modelo de dominio del módulo Actividades

Conceptos de las clases del modelo de dominio

Autor: Es la persona encargada de crear los OA en la herramienta de autor web CRODA.

CRODA: Herramienta de autor web cuyo objetivo principal es permitir la creación de OA, reutilizables, accesibles, duraderos e interoperables, de forma flexible, empleando los estándares SCORM y LOM.

Ejercicios: Conjunto de preguntas de diferentes tipos englobadas en un mismo cuestionario.

CAPÍTULO 2.

CARACTERÍSTICAS DEL SISTEMA



Metadato: Conjunto de información que describe los recursos utilizados dentro del OA, para facilitar su búsqueda y reutilización dentro del mismo.

OA: Recurso digital dirigido a la formación, fácilmente identificado y encontrado a través de sus metadatos, con capacidades de integración en diferentes sistemas e-learning, para ser usados con distintos propósitos educativos y para combinarse con otros objetos formando una sola entidad.

Recurso: Cualquier tipo de archivo de tipo imagen, audio, video, sonido o texto.

Pregunta de enlazar columnas: Tipología de pregunta que permite dado dos conjuntos de opciones crear pares de asociaciones entre ellos.

Pregunta de Verdadero o falso: Tipología de pregunta que permite contestar verdadero o falso dada una idea.

Pregunta de selección simple/múltiple: Tipología de pregunta que permite seleccionar una o varias respuestas correctas entre varias posibilidades.

Pregunta de llenar espacios en blanco: Tipología de pregunta que permite llenar con texto espacios en blanco dentro de un texto dado.

Pregunta de búsqueda en el texto: Tipología de pregunta que permite realizar una búsqueda en un fragmento de texto.

Pregunta de orden cronológico: Tipología de pregunta que permite ordenar un grupo de ideas.

Descripción del modelo de dominio

El autor es un rol autenticado en la herramienta de autor web CRODA, que utiliza la misma para la creación de uno o más OA. Si el autor lo considera, le puede incorporar al OA recursos entre los que se encuentran objetos multimedia; como imágenes, videos o animaciones y a su vez describirlas haciendo uso de metadatos. Cuando se esté confeccionando el OA el autor puede realizar uno o varios ejercicios de autoevaluación. Estos podrán estar conformados por una o varias preguntas de diferentes o una misma tipología como pueden ser: verdadero o falso, enlazar columnas, selección simple o múltiple, llenar espacios en blanco, orden cronológico y búsqueda en el texto.

CAPÍTULO 2.

CARACTERÍSTICAS DEL SISTEMA



2.2 Requisitos del sistema

La ingeniería de requerimientos es un proceso extenso y arduo, ya que desarrollar un sistema o parte de este, cambia el entorno y las relaciones entre las personas, por lo que es importante identificar a todos los implicados, considerar sus necesidades y asegurar que entiendan las implicaciones de los cambios a realizar en el sistema. Se pueden emplear varias técnicas para obtener los requisitos del cliente, tales como: tormentas de ideas, casos de uso, comparación de terminología y estudio de otros sistemas.

Requisitos funcionales del sistema

Los requisitos funcionales son declaraciones de los servicios que debe proporcionar el sistema, de la manera en que este debe reaccionar a entradas particulares y de cómo se debe comportar en situaciones particulares (27).

Los requisitos funcionales con que debe cumplir la aplicación en cuestión serán:

RF1: La aplicación debe permitir crear galería de imágenes.

RF2: La aplicación debe permitir eliminar galería de imágenes.

RF3: La aplicación debe permitir editar galería de imágenes.

RF4: La aplicación debe permitir seleccionar las imágenes a utilizar en la galería de imágenes.

RF5: La aplicación debe permitir exportar a formatos doc, pdf y HTML.

RF6: La aplicación debe permitir visualizar dentro del navegador los archivos de formato pdf.

RF7: La aplicación debe permitir agregar actividades en forma de sopas de palabras.

RF8: La aplicación debe permitir la opción de añadir una nueva palabra a la lista de palabras escondidas de la sopa de palabras.

RF9: La aplicación debe permitir agregar actividades en forma de crucigramas.

RF10: La aplicación debe permitir ubicar las descripciones de las palabras a descubrir en el crucigrama.

RF11: La aplicación debe permitir insertar palabras en el crucigrama dada una descripción.

RF12: La aplicación debe permitir modificar palabras en el crucigrama dada una descripción.

CAPÍTULO 2.

CARACTERÍSTICAS DEL SISTEMA



- RF13:** La aplicación debe permitir crear un puzzle con una imagen.
- RF14:** La aplicación debe permitir seleccionar la imagen a utilizar en el puzzle.
- RF15:** La aplicación debe permitir obtener una vista previa del puzzle.
- RF16:** La aplicación debe permitir cargar una imagen como fondo para el puzzle.
- RF17:** La aplicación debe permitir crear un puzzle de intercambio con contenido textual.
- RF18:** La aplicación debe permitir obtener una vista previa del puzzle de intercambio con contenido textual.
- RF19:** La aplicación debe permitir editar el texto de cada una de las casillas del puzzle de intercambio con contenido textual.
- RF20:** La aplicación debe permitir cargar una imagen como fondo para el puzzle de intercambio con contenido textual.
- RF21:** La aplicación debe permitir crear un juego de memoria con elementos iguales.
- RF22:** La aplicación debe permitir obtener una vista previa del juego de memoria con elementos iguales.
- RF23:** La aplicación debe permitir seleccionar las imágenes a utilizar en el juego de memoria con elementos iguales.
- RF24:** La aplicación debe permitir crear un ejercicio de asociación simple entre imágenes y texto.
- RF25:** La aplicación debe permitir buscar una imagen para el ejercicio de asociación simple entre texto e imagen.
- RF26:** La aplicación debe permitir obtener una vista previa del ejercicio de asociación entre el texto y la imagen.
- RF27:** La aplicación debe permitir crear una actividad de exploración.
- RF28:** La aplicación debe permitir obtener una vista previa de la actividad de exploración.
- RF29:** La aplicación debe permitir seleccionar una imagen para utilizar en la actividad de exploración.
- RF30:** La aplicación debe permitir agregar actividades de selección de palabras o frases dado un conjunto de opciones.

CAPÍTULO 2.

CARACTERÍSTICAS DEL SISTEMA



RF31: La aplicación debe permitir agregar las palabras o frases al ejercicio.

RF32: La aplicación debe permitir agregar actividades que permitan la construcción de textos dada una pregunta específica.

RF33: La aplicación debe permitir agregar las preguntas al ejercicio de construcción de texto.

RF34: La aplicación debe permitir seleccionar el archivo en formato pdf que desea visualizar.

RF35: La aplicación debe permitir buscar el ejercicio que se desea exportar a un formato determinado.

Requisitos no funcionales del sistema

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 (28).

En este trabajo se identificaron los siguientes requisitos no funcionales:

Usabilidad

RNF 1: El sistema podrá ser usado por cualquier persona que posea conocimientos básicos en el manejo de la computadora y de un ambiente web en sentido general.

RNF 2: En la interacción con el software el sistema mostrará mensajes que sean de entendimiento común con el usuario, para de esta forma elevar la comunicación con el mismo.

RNF 3: En los campos que sean complejos se mostrarán notas indicando ejemplos concretos para guiar el proceso.

Accesibilidad

RNF 4: Las interfaces deben tener vínculos entre ellas para facilitar el proceso.

RNF 5: Protección contra acciones no autorizadas que puedan afectar la integridad de los datos.

Disponibilidad

RNF 6: Una vez publicado el sistema, deberá estar disponible para los usuarios en cualquier momento.

Eficiencia

CAPÍTULO 2.

CARACTERÍSTICAS DEL SISTEMA



RNF7: Tiempos de respuestas rápidas como máximo 10 segundos al igual que la velocidad de procesamiento de la información.

RNF 8: El sistema funcionará de manera correcta en los navegadores web Mozilla Firefox, Internet Explorer 6 o superior y Opera.

RNF 9: Debe ser accedido el sistema desde cualquier sistema operativo, debido a las características de las tecnologías propuestas para el desarrollo del módulo.

Seguridad y privacidad

RNF 10: El sistema debe asegurar que los materiales y recursos sean vistos, además de ser analizados únicamente por las personas capacitadas y las establecidas previamente por el administrador de la aplicación, por lo que se hace necesario el uso de roles especificando los privilegios para cada uno en el sistema.

RNF 11: El sistema debe garantizar que datos sensibles no viajen en texto plano por la red.

RNF12: El sistema debe asegurar que no se realicen ataques por inyección sql a la aplicación.

Software

RNF 13: Se utilizará PostgreSQL versión 8.4 como gestor de base de datos.

RNF 14: Se utilizará Exist-db versión 1.4 como base de datos de XML nativo.

Restricciones del diseño

Herramienta CASE de modelado visual

RNF 15: Para el modelado visual se deberá usar Visual Paradigm 6.4 como herramienta CASE, por sus múltiples beneficios en la modelación de los diferentes artefactos que se obtienen en los distintos flujos de trabajo.

Lenguaje de programación

RNF 16: Por el empleo del framework Symfony 1.4.9 el cual es compatible con la mayoría de los gestores de base de datos y está basado en el patrón Modelo-Vista-Controlador, el lenguaje de programación a utilizar será PHP 5.3.3 el cual constituye la base del framework, se utilizarán además JavaScript para

CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA



controlar las ventanas del navegador y el contenido que se mostrará, CCS para dar estilo a documentos HTML y XML logrando separar el contenido de la presentación y HTML para darle estructura a las páginas de la herramienta.

CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA



Entorno de desarrollo integrado IDE

RNF 17: Se hará uso de Netbeans 6.9 como entorno de desarrollo IDE por ser un producto libre, gratuito y sin restricciones de uso.

Interfaz

Interfaces de usuario

RNF 18. Identificación de colores y formatos compatibles con la herramienta de autor CRODA 1.0.

2.3 Definición de actores y casos de uso del sistema

La tabla que se muestra a continuación incluye una breve descripción del actor que interactúa con el presente módulo.

Actor	Descripción
Autor_OA	Es el autor con permiso sobre el OA que establece cada una de las actividades a incorporar en el mismo.

2.4 Patrones de casos de uso

La experiencia en la utilización de casos de uso ha evolucionado en un conjunto de patrones que permiten con precisión reflejar los requisitos reales, haciendo más fácil el trabajo con los sistemas, y mucho más simple su mantenimiento. Dado un contexto y un problema a resolver, estas técnicas han mostrado ser la solución adoptada en la comunidad del desarrollo de software. A continuación, se explican los patrones presentes en el diagrama de casos de uso del sistema.

Concordancia: Extrae una subsecuencia de acciones que aparecen en diferentes lugares del flujo de casos de uso y es expresado por separado.

Reusabilidad: Consta de tres casos de uso. El primero llamado subsecuencia común, modela una secuencia de acciones que aparecerán en múltiples casos de uso en el modelo. Los otros casos de uso modelan el uso del sistema que comparte la subsecuencia común de acciones. De manera que deben existir al menos dos de ellos.



Imagen 11: Patrón concordancia-reusabilidad

Adición: En el caso de este patrón alternativo, la subsecuencia común de casos de uso, extiende los casos de uso compartiendo la subsecuencia de acciones. Los otros casos de uso modelan el flujo que será expandido con la subsecuencia. Este patrón es preferible usarlo cuando otros casos de uso se encuentran propiamente completos, o sea, que no requieren de una subsecuencia común de acciones para modelar los usos completos del sistema.



Imagen 12: Patrón concordancia-adición

CRUD (Crear, Leer, Modificar, Eliminar)

Este patrón se basa en la fusión de casos de uso simples para formar una unidad conceptual.

Parcial: Este patrón alternativo modela una de las vías de los casos de uso como un caso de uso separado. Es preferiblemente utilizado cuando una de las alternativas de los casos de uso es más significativa, larga o más compleja que las otras.

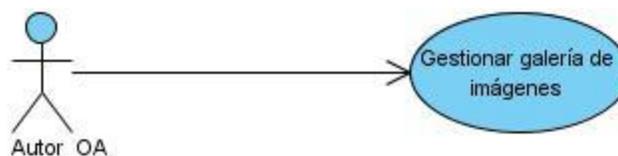


Imagen 13: Patrón CRUD-parcial

CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA



2.5 Diagrama de casos de usos identificados

El diagrama de casos de uso del sistema brinda las funcionalidades que el sistema debe ofrecer para aportar un resultado de valor observable, representando gráficamente los procesos y su interacción con los actores del sistema. (Ver imagen 14)

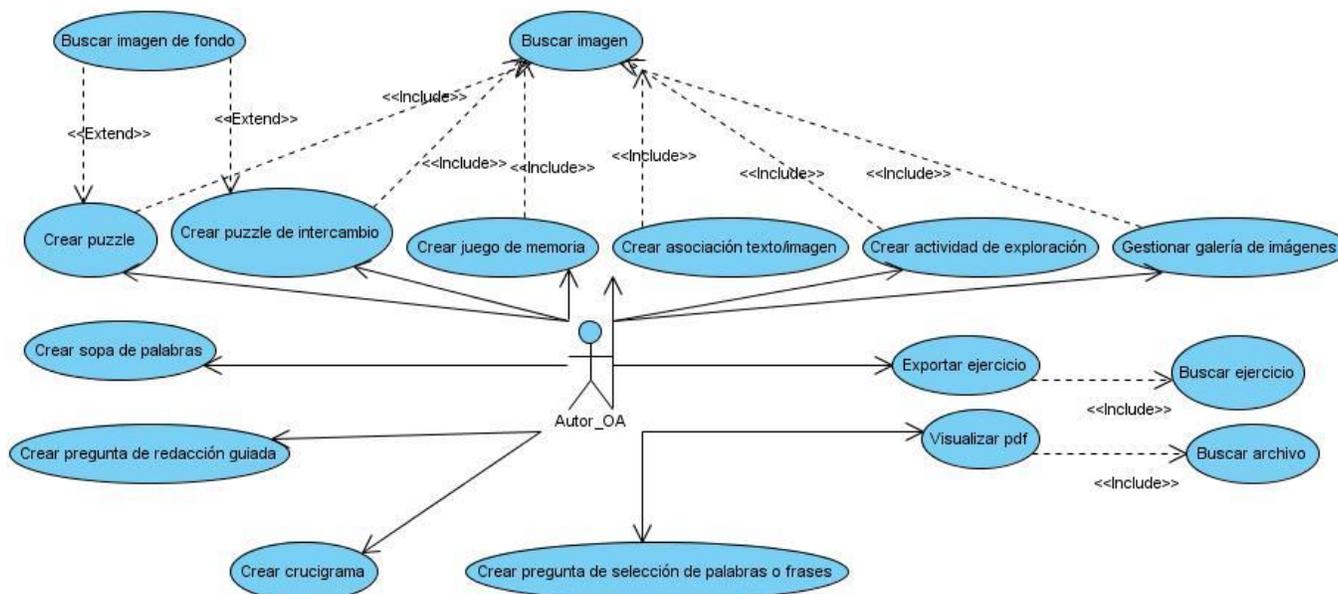


Imagen 14. Diagrama de casos de uso del sistema

2.6 Descripción textual de los casos de uso

A continuación se presentan las descripciones de algunos de los casos de uso críticos del sistema que presenta el módulo (Ver anexos del 4 al 19).

Caso de Uso:	Gestionar galería de imágenes
Actores:	Autor_OA
Resumen:	El caso de uso inicia cuando el actor accede a las opciones de creación de una galería de imágenes, eliminar galería de imágenes, además puede guardar las galerías creadas, finaliza el caso de uso cuando el actor accede a una de las opciones anteriores.

CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA



Precondiciones:	El autor del OA debe estar autenticado en el sistema y ubicado en la sección de autoevaluaciones del OA.
Poscondiciones	Se gestiona la galería de imágenes.
Referencias	RF1,RF2,RF3
Prioridad	Crítico
Flujo Normal de Eventos	
Sección “Crear galería de imágenes”	
Acción del Actor	Respuesta del Sistema
1. El actor selecciona la acción “Crear galería de imágenes”.	2. El sistema muestra los siguientes campos a introducir: <ul style="list-style-type: none"> • Nombre de la galería de imágenes. • Seleccionar las imágenes para la galería. Se muestra las opciones: “Enviar” y “Cancelar”.
3. El autor completa los campos obligatorios para la creación de la galería de imágenes y selecciona la opción “Enviar”.	4. El sistema valida los datos introducidos por el autor de la actividad.
	5. El sistema guarda los datos introducidos y se muestra la galería de imágenes creada en las actividades de autoevaluación del OA.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
3. El autor completa los campos obligatorios para la creación de la galería de imágenes y selecciona la	4. El sistema muestra la sección de autoevaluación del OA.

CAPÍTULO 2.

CARACTERÍSTICAS DEL SISTEMA



opción "Cancelar".	
Flujo Normal de Eventos	
Sección "Eliminar galería de imágenes"	
Acción del Actor	Respuesta del Sistema
1. El autor del OA selecciona la opción "Eliminar galería de imágenes".	2. El sistema muestra el siguiente mensaje: ¿Desea en realidad eliminar su galería de imágenes? Se eliminarán los archivos que están publicados en la misma. Se muestran las opciones "Aceptar" y "Cancelar".
3. El autor selecciona la opción "Aceptar"	4. El sistema muestra el mensaje: "Se ha eliminado correctamente la galería de imágenes".
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
3. El autor selecciona la opción "Cancelar".	4. El sistema muestra la sección de autoevaluación del OA.
Flujo Normal de Eventos	
Sección "Editar galería de imágenes"	
Acción del Actor	Respuesta del Sistema
1. El autor del OA selecciona la opción "Editar galería de imágenes".	2. El sistema muestra la interfaz de edición de la galería de imágenes y las opciones "Aceptar" y "Cancelar".
3. El autor realiza los cambios deseados en la galería de imágenes que se encuentra editando y selecciona la opción "Aceptar"	4. El sistema valida los datos introducidos por el autor del OA.

CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA



	5. El sistema guarda los datos que se han editado y muestra el mensaje: “Se ha editado correctamente la galería de imágenes”.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
3. El autor selecciona la opción “Cancelar”.	4. El sistema muestra la sección de autoevaluación del OA.

Caso de Uso:	Crear sopa de palabras
Actores:	Autor de OA
Resumen:	El caso de uso se inicia cuando el actor accede a la opción “Crear sopa de palabras” y finaliza cuando se crea de forma satisfactoria la actividad.
Precondiciones:	El actor debe estar autenticado en el sistema.
Poscondiciones:	Se crea la sopa de palabras de forma satisfactoria.
Referencias:	RF7, RF8
Prioridad:	Crítico
Flujo Normal de Eventos	
Sección “Crear sopa de palabras”	
Acción del Actor	Respuesta del Sistema
1. El actor selecciona la opción “Crear sopa de palabras”.	2. El sistema muestra los siguientes campos a introducir: <ul style="list-style-type: none"> • Tema. • Enunciado. • Número de filas. • Número de columnas. • Ancho de las celdas.

CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA



	<ul style="list-style-type: none"> • Altura de las celdas. • Lista de palabras escondidas. <p>Se muestran las opciones “Enviar” y “Cancelar”.</p>
3. El actor completa los campos obligatorios para la creación de la sopa de palabras y selecciona la opción “Enviar”.	4. El sistema valida los datos introducidos por el autor de la actividad.
	5. El sistema guarda los datos introducidos y se muestra la sopa de palabras creada en las actividades de autoevaluación del OA.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
3. El actor selecciona la opción “Cancelar”.	4. El sistema muestra la sección de autoevaluación del OA.

Caso de Uso:	Crear crucigramas
Actores:	Autor_ OA
Resumen:	El caso de uso se inicia cuando el actor selecciona la opción “Crear crucigramas” y finaliza cuando se crea de forma satisfactoria la actividad.
Precondiciones:	El actor debe estar autenticado en el sistema.
Poscondiciones	Se crea el crucigrama.
Referencias	RF9,RF10,RF11,RF12
Prioridad	Crítico
Flujo Normal de Eventos	

CAPÍTULO 2.

CARACTERÍSTICAS DEL SISTEMA



Sección “Crear crucigramas”	
Acción del Actor	Respuesta del Sistema
1. El actor selecciona la opción “Crear crucigrama”.	<p>2. El sistema muestra los siguientes campos a introducir:</p> <ul style="list-style-type: none"> • Tema. • Enunciado. • Número de filas. • Número de columnas. • Altura de las celdas. • Ancho de las celdas. • Descripción de las palabras a encontrar. • Palabras según la descripción dada. • Como buscar las palabras vertical o horizontal. <p>Se muestran las opciones “Enviar” y “Cancelar”.</p>
3. El actor completa los campos obligatorios para la creación del crucigrama y selecciona la opción “Enviar”.	4. El sistema valida los datos introducidos por el autor de la actividad.
	5. El sistema guarda los datos introducidos y se muestra la sopa de palabras creada en las actividades de autoevaluación del OA.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema

CAPÍTULO 2.

CARACTERÍSTICAS DEL SISTEMA



3. El actor selecciona la opción “Cancelar”.	4. El sistema muestra la sección de autoevaluación del OA.
--	--

Caso de Uso:	Crear asociación simple texto-imagen.
Actores:	Autor_ OA
Resumen:	El caso de uso inicia cuando el actor accede a las opciones de creación de asociación texto-imageny finaliza cuando se crea de forma satisfactoria el ejercicio.
Precondiciones:	El autor del OA debe estar autenticado en el sistema y ubicado en la sección de autoevaluaciones del OA.
Poscondiciones	Se crea el ejercicio de asociación texto-imagen.
Referencias	RF24,RF25, RF26
Prioridad	Crítico

Flujo Normal de Eventos

Sección “Crear asociación texto-imagen”

Acción del Actor	Respuesta del Sistema
1. El autor selecciona la opción “Crear asociación texto-imagen”.	2. El sistema muestra los siguientes campos a introducir: <ul style="list-style-type: none"> • Color inactivo. • Color activo. • Cantidad de filas panel A. • Cantidad de filas panel B. • Cantidad de columnas panel B. • Cantidad de columnas panel A. • Ancho de la celda. • Alto de la celda. • Distribución. • Insertar imágenes.

CAPÍTULO 2.

CARACTERÍSTICAS DEL SISTEMA



	<ul style="list-style-type: none"> • Insertar palabras • Asociar palabras con imágenes. <p>Se muestran las opciones de “Enviar” y “Cancelar”.</p>
3. El autor completa los campos obligatorios para crear el ejercicio de asociación texto-imagen y presiona la opción “Enviar”.	4. El sistema valida los datos introducidos por el autor y muestra una vista previa de la actividad.
	5. El sistema guarda los datos introducidos y se muestra el ejercicio de asociación texto-imagen creado formando parte de las actividades de autoevaluación del OA.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
3. El autor completa los campos obligatorios para la creación del ejercicio de asociación texto- imagen y presiona la opción “Cancelar”.	4. El sistema muestra la sección de autoevaluación del OA.

Caso de Uso:	Crear pregunta de redacción guiada
Actores:	Autor_OA
Resumen:	El caso de uso inicia cuando el actor accede a las opciones de crear ejercicio de redacción guiada, finaliza el caso de uso cuando el actor accede a la opción crear ejercicio de redacción guiada luego de completar los campos obligatorios y ver concluido uno de los flujos del caso de uso.

CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA



Precondiciones:	El autor del OA debe estar autenticado en el sistema y ubicado en la sección de autoevaluación del OA.	
Poscondiciones:	Se crea el ejercicio de forma exitosa.	
Referencias	RF32,RF33	
Prioridad	Crítico	
Flujo Normal de Eventos		
Sección “Crear redacción guiada”		
Acción del Actor	Respuesta del Sistema	
1. El autor del OA accede a la opción “Crear pregunta de redacción guiada”.	2. El sistema muestra los campos obligatorios a introducir: <ul style="list-style-type: none"> • Tema • Enunciado • Preguntas. Muestra las opciones de “Crear” y “Cancelar”.	
3. El usuario completa los campos obligatorios y selecciona la opción “Crear”.	4. El sistema valida los datos introducidos por el autor de la actividad.	
	5. El sistema guarda los datos y se crea el ejercicio de redacción guiada satisfactoriamente, mostrándolo a la vez formando parte de las Actividades de autoevaluación del OA.	
Flujos Alternos		
Acción del Actor	Respuesta del Sistema	
3. El usuario completa los campos obligatorios y selecciona la opción “Cancelar”.	4. El sistema muestra la sección de autoevaluación del OA.	

CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA



Conclusiones parciales

Se ha realizado un modelo de dominio para ofrecer una visión general del ámbito donde se desarrolla, brindando una base para el desarrollo del flujo de trabajo Requerimientos. Se determinaron los requisitos funcionales y no funcionales que debe tener el sistema, asimismo se desarrolló el modelo de casos de uso del sistema. De esta manera, el presente capítulo resume las consideraciones previas al flujo de trabajo Análisis y diseño.

CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

Como un paso fundamental para la posterior implementación, en el presente capítulo se realizan los artefactos requeridos del flujo de trabajo Análisis y diseño para los casos de uso identificados anteriormente. Tanto el análisis, como el diseño constituyen elementos básicos en el proceso de desarrollo de software, pues guían al desarrollador; mostrándole mediante los artefactos del modelo de diseño un conjunto de elementos que serán posteriormente implementados.

3.1 Modelo de Análisis

Teniendo en cuenta los requisitos capturados en el capítulo anterior, en el Modelo de Análisis se realiza un esbozo preliminar del sistema a implementar. El principal objetivo del análisis, es comprender perfectamente los requisitos del software y no precisar cómo se implementará la solución, por lo que no se tienen en cuenta los lenguajes de programación, herramientas, ni tecnologías a utilizar. Básicamente se enfoca en los requisitos funcionales del sistema.

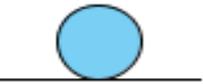
3.1.1 Diagrama de clases del análisis

El primer paso para la elaboración del diagrama de clases del análisis es la identificación de las clases del análisis, las cuales son la representación de los roles de los elementos del modelo. Las clases del análisis se dividen en tres estereotipos que se presentan a continuación:

Estereotipo	Descripción
-------------	-------------

CAPÍTULO 3. ANÁLISIS Y DISEÑO DEL SISTEMA



 <p style="text-align: center;">CI</p>	<p>Clase Interfaz: Se utilizan para modelar la interrelación entre el sistema y sus actores (es decir usuarios y sistemas externos). Las clases interfaz modelan las partes del sistema que dependen de sus actores lo que implica que clarifiquen y reúnan los requisitos en los límites del sistema.</p>
 <p style="text-align: center;">CC</p>	<p>Clase controladora: Representan la coordinación, secuencia, transacciones y control de objetos, se usan con frecuencia para encapsular el control de un caso de uso en concreto.</p>
 <p style="text-align: center;">CE</p>	<p>Clase entidad: Se utilizan para modelar información que posee una vida larga y que es a menudo persistente. En la mayoría de los casos se derivan directamente de una clase entidad del negocio.</p>

A continuación se presentan los diagramas de clases del análisis de algunos de los casos de uso críticos del sistema.

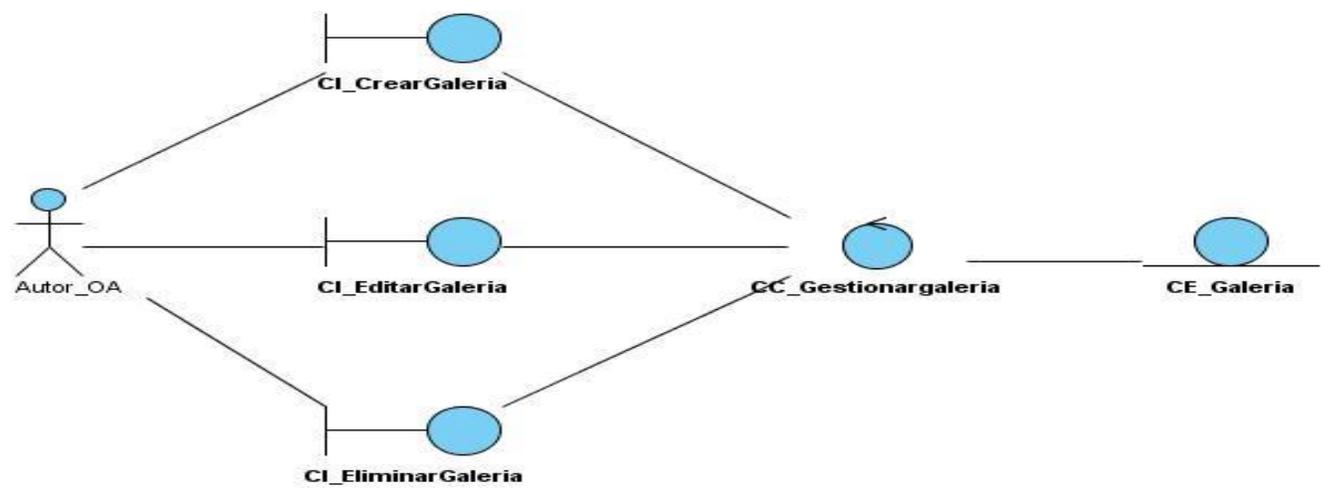


Imagen 9: Diagrama de clases del análisis CU Gestionar galería de imágenes

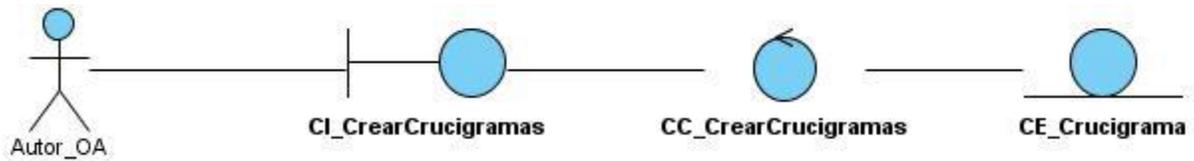


Imagen 10: Diagrama de clases del análisis CU Crear crucigramas

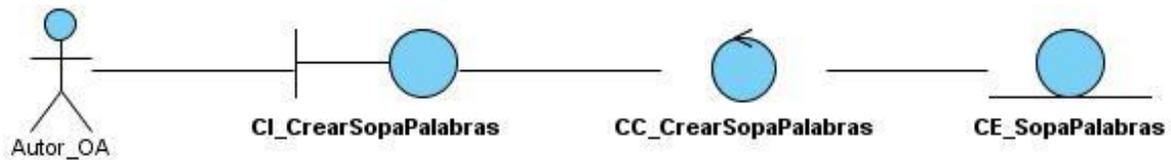


Imagen 15: Diagrama de clases del análisis CU Crear sopa de palabras

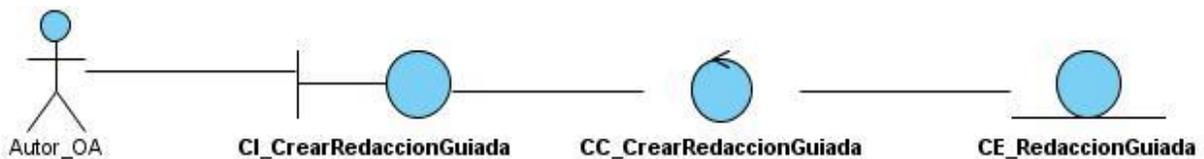


Imagen 16: Diagrama de clases del análisis CU Crear pregunta de redacción guiada



Imagen 17: Diagrama de clases del análisis CU Crear asociación simple texto-imagen

En este 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. (Ver Anexos del 20 al 35).

3.1.2 Diagramas de colaboración

Un diagrama de colaboración es una forma de representar interacción entre objetos, pueden mostrar el contexto de la operación viéndose los objetos que son atributos, los temporales y los ciclos en la ejecución. A continuación se presentan algunos de los diagramas de colaboración de los casos de usos críticos que posee el sistema.

CAPÍTULO 3. ANÁLISIS Y DISEÑO DEL SISTEMA

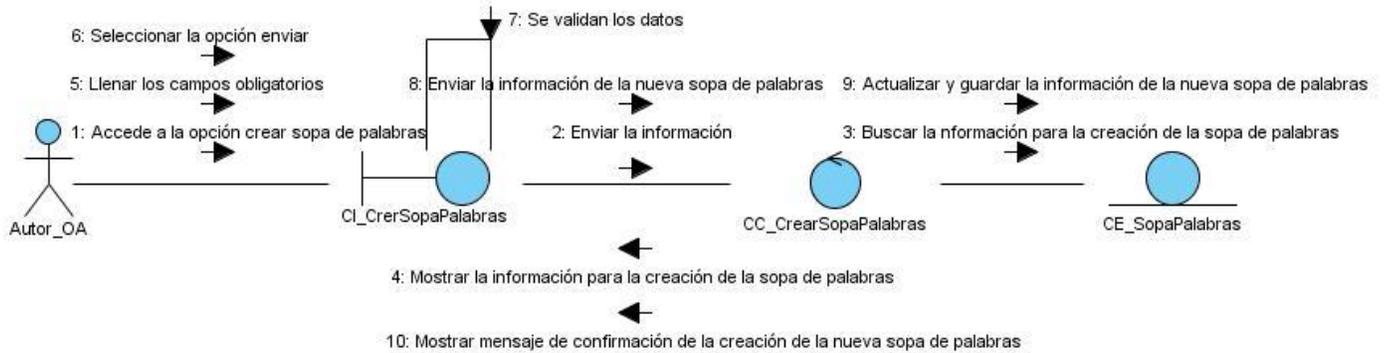


Imagen 18: Diagrama de colaboración CU Crear sopa de palabras



Imagen 19: Diagrama de colaboración CU Gestionar galería de imágenes Sección Eliminar

CAPÍTULO 3. ANÁLISIS Y DISEÑO DEL SISTEMA



Imagen 20: Diagrama de colaboración CU Gestionar galería de imágenes Sección Editar

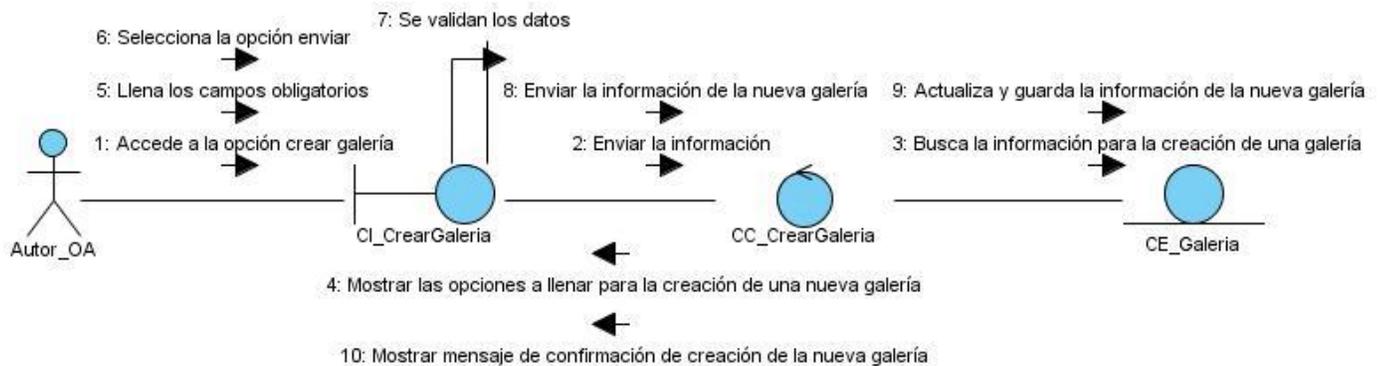


Imagen 17: Diagrama de colaboración CU Gestionar galería de imágenes Sección Crear

CAPÍTULO 3. ANÁLISIS Y DISEÑO DEL SISTEMA

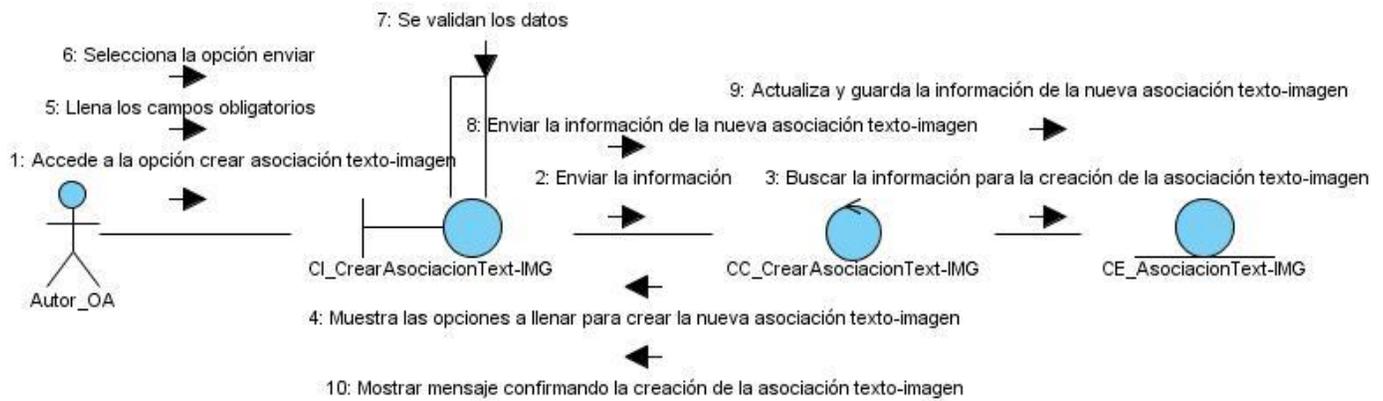


Imagen 18: Diagrama de colaboración CU Crear asociación simple texto-imagen

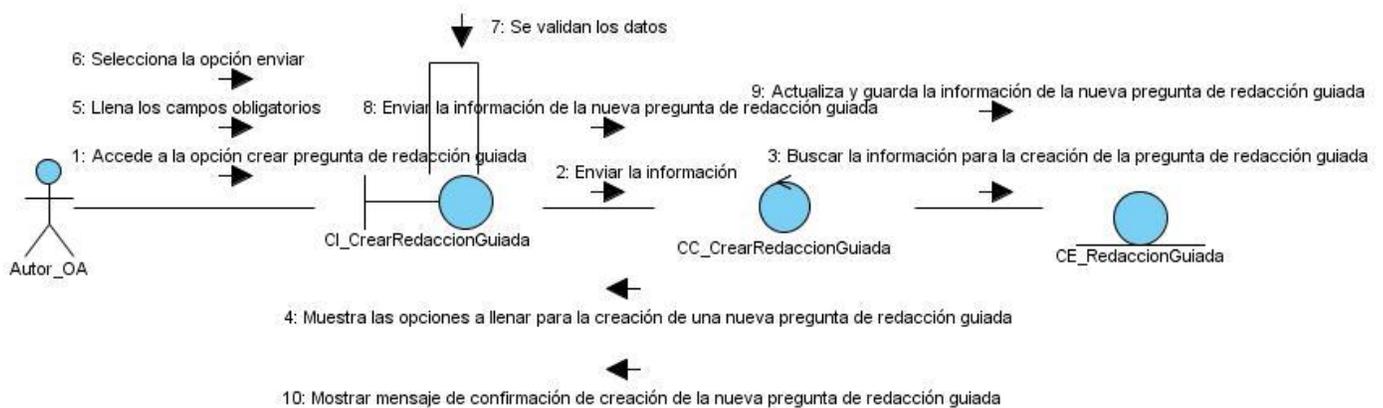


Imagen 19: Diagrama de colaboración CU Crear pregunta de redacción guiada

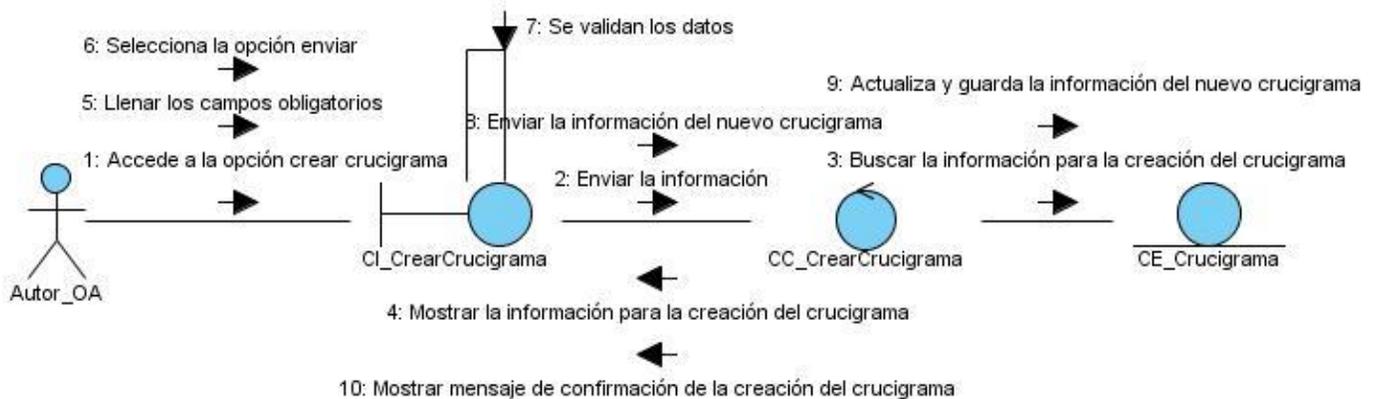


Imagen 20: Diagrama de colaboración CU Crear crucigramas

Los diagramas de colaboración de las nuevas funcionalidades a incluir en el Módulo Actividades se encuentran en los anexos del 36 al 53.

3.2 Arquitectura propuesta

La arquitectura de software define, de manera abstracta, los componentes que llevan a cabo alguna tarea de computación, sus interfaces y la comunicación entre ellos. También tiene que ver con el diseño y la implementación de estructuras de software de alto nivel. Es el resultado de ensamblar un cierto número de elementos arquitectónicos de forma adecuada para satisfacer la mayor funcionalidad y requisitos de desempeño de un sistema, así como requisitos no funcionales, la confiabilidad, escalabilidad, portabilidad y disponibilidad (29).

3.2.1 Patrón arquitectónico

Son la expresión de un esquema estructural fundamental de organización de sistemas de software. Proporciona un conjunto de subsistemas predefinidos, especifica sus responsabilidades, e incluye normas y directrices para la organización de las relaciones entre ellos (30).

Se considera entonces que los patrones arquitectónicos son los que definen la estructura de un sistema software. Los cuales a su vez se componen de subsistemas con sus responsabilidades, también presentan una serie de directivas para organizar sus componentes con el objetivo de facilitar la tarea del diseño.

CAPÍTULO 3. ANÁLISIS Y DISEÑO DEL SISTEMA



Patrón Modelo-Vista-Controlador

El patrón Modelo–Vista–Controlador (MVC) está catalogado como un patrón de diseño de software donde:

Modelo: Administra y maneja todo lo relacionado con los datos del sistema, da respuesta a peticiones de información sobre el estado de la aplicación (normalmente desde la vista), y responde con instrucciones de cambio de estado (usualmente desde el controlador) a la vista.

Vista: Gestiona lo relacionado con mostrar la información al usuario.

Controlador: El controlador interpreta los eventos que son lanzados por la entrada estándar del usuario (normalmente mouse y teclado), informando de los mismos al modelo y/o la vista para que se ejecuten los cambios apropiadamente.

A continuación se muestra una representación gráfica del patrón MVC.

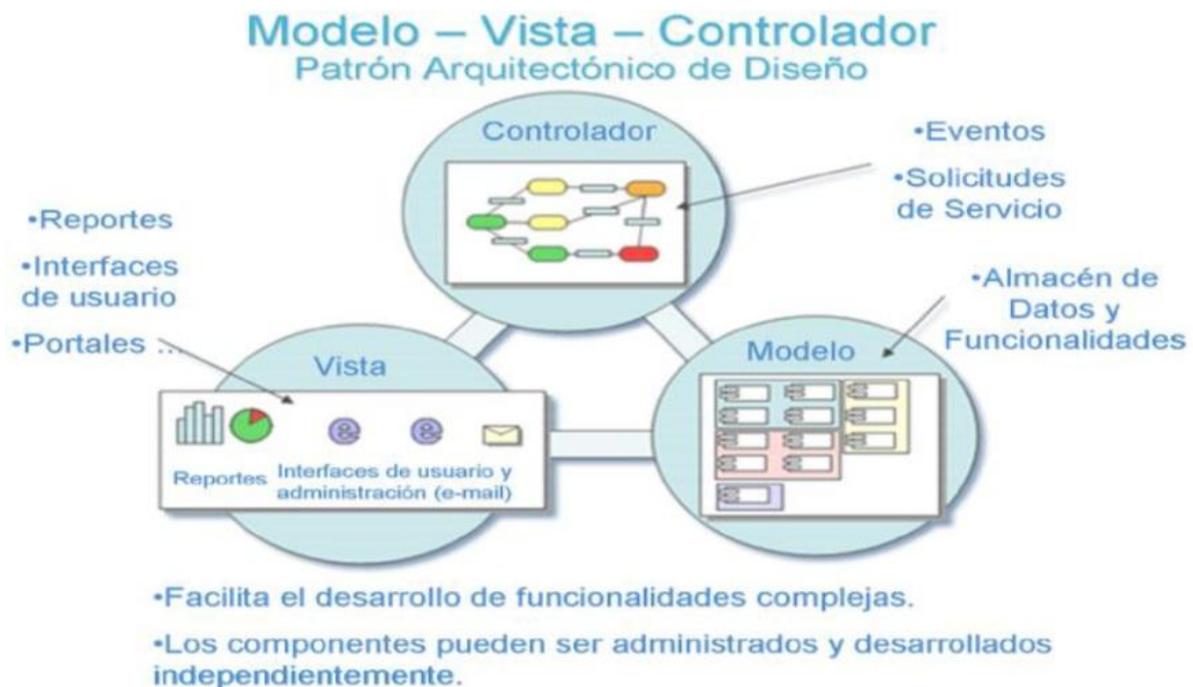


Imagen 21: Patrón Modelo-Vista-Controlador



3.4 Modelo del diseño

El modelo del diseño constituye el refinamiento del análisis. Soporta los requisitos funcionales y no funcionales, así como las restricciones que se suponen. Define cómo cumple el sistema sus objetivos, además impone una estructura del sistema que se debe conservar lo más fielmente posible cuando demos forma al sistema. Constituyendo el refinamiento del análisis y la entrada al flujo de trabajo de implementación. Propósitos que presenta RUP:

- Capturar las interfaces entre los subsistemas antes que el ciclo de vida del software.
- Descomponer los trabajos de implementación en partes más manejables.
- Crear una entrada apropiada y un punto de partida para actividades de implementación.

Diagramas de clases del diseño

Los diagramas de clases del diseño muestran el sistema desde un punto de vista estático, a través de una colección de elementos declarativos, como clases, colaboraciones y sus relaciones. Son clases con un mayor nivel de detalle. Se conciben para satisfacer los requisitos funcionales y no funcionales, teniendo en consideración la tecnología en la cual se implementará el diseño. A continuación se explican los estereotipos utilizados:

Prototipo	Descripción
	<<Server Page>> : Representa la clase que tiene código que se ejecuta en el servidor, la cual se encarga de construir (build) o generar el resultado HTML y/o realizar peticiones a la capa inferior.
	<<Client Page>> : Es la página web de formato HTML, que le brinda al usuario toda la interfaz de la aplicación. Sus atributos son las variables declaradas dentro del script que son accesibles para cualquier función dentro de la página. Cada página cliente es construida por una sola página de servidor.

CAPÍTULO 3. ANÁLISIS Y DISEÑO DEL SISTEMA



<<Form>>: Es una colección de elementos de entrada que están contenidos en la página cliente. Sus atributos son los elementos de entrada del formulario. No tienen operaciones y se comunican con las páginas servidores mediante submit.

A continuación se presentan los diagramas de clases del diseño con estereotipos web de los casos de uso descritos del sistema.

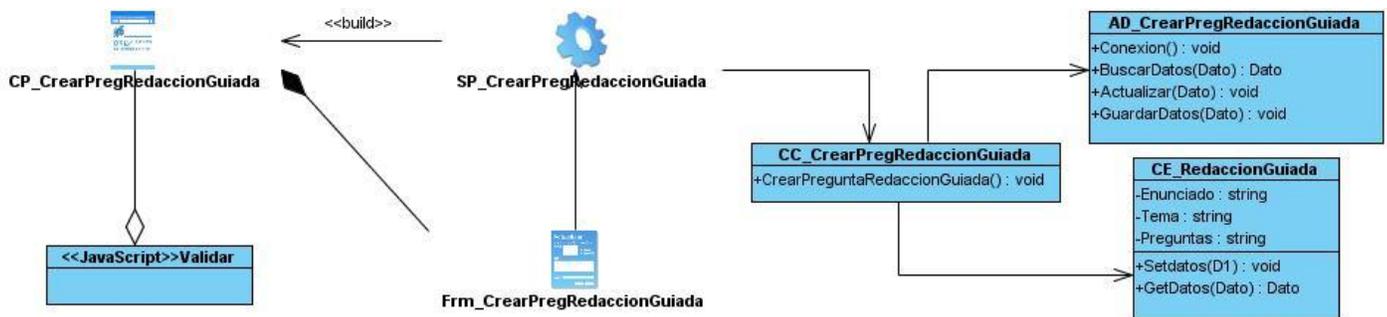


Imagen 22: Diagrama de clases del diseño con estereotipos web CU redacción guiada.

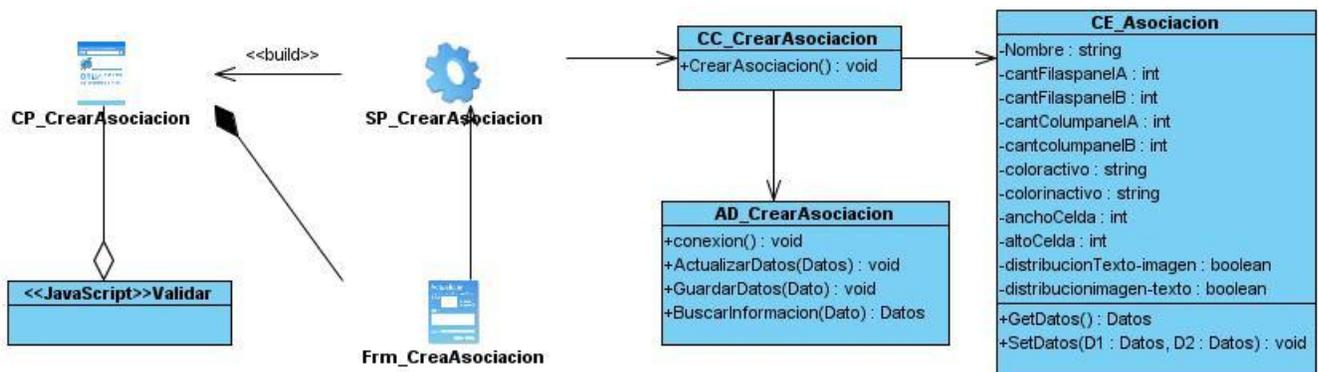


Imagen 23: Diagrama de clases del diseño con estereotipos web CU Crear asociación simple texto-imagen

CAPÍTULO 3. ANÁLISIS Y DISEÑO DEL SISTEMA

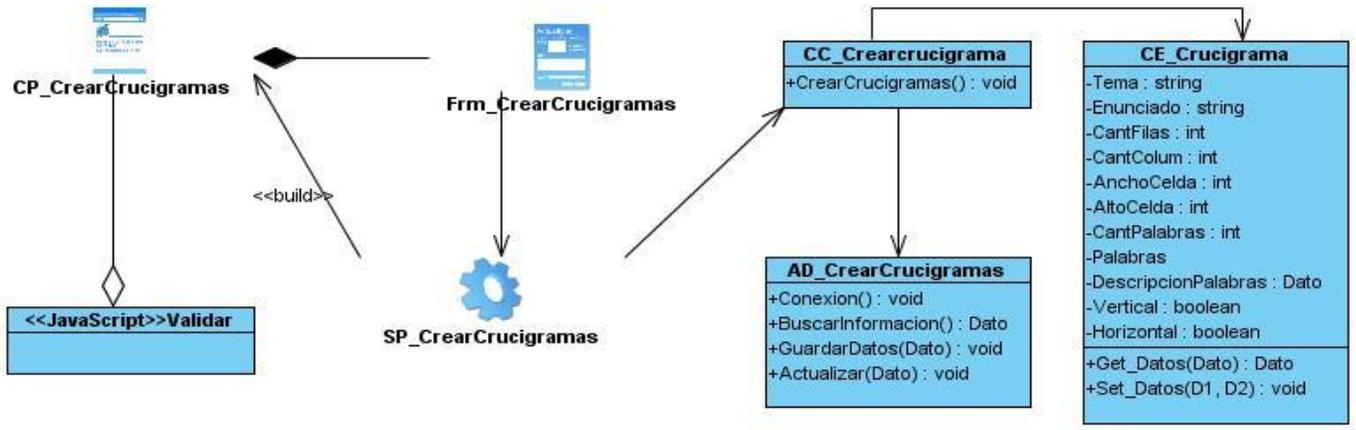


Imagen 24: Diagrama de clases del diseño con estereotipos web CU Crear crucigramas

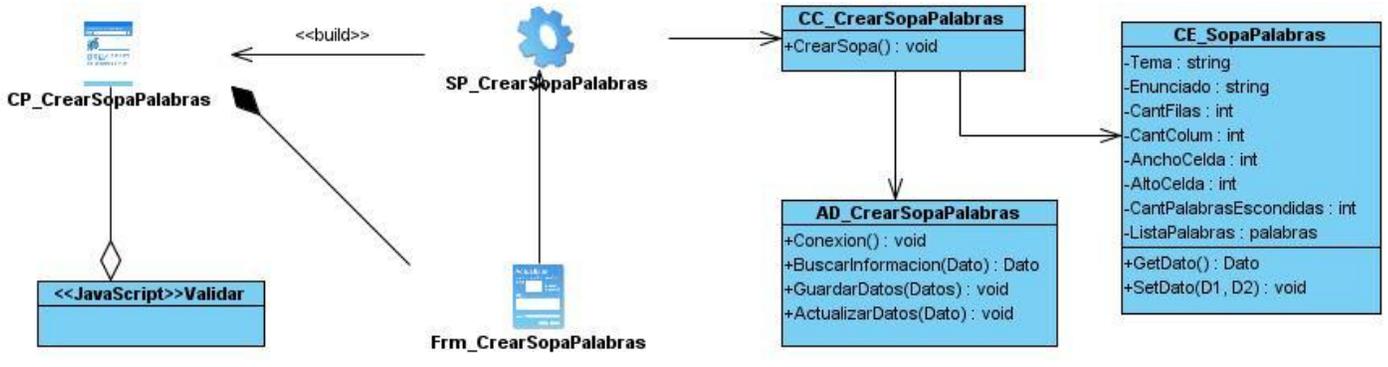


Imagen 25: Diagrama de clases del diseño con estereotipos web CU Crear sopa de palabras

CONCLUSIONES GENERALES

Con el propósito de darle cumplimiento al objetivo general y a la problemática planteada en el presente trabajo, se han llevado a cabo satisfactoriamente cada una de las tareas que fueron trazadas al comienzo del mismo. Al finalizar la presente investigación, se puede afirmar que:

- Con los artefactos generados en los flujos de trabajo Requerimientos y Análisis y diseño de la metodología RUP, se crearon las bases para la futura implementación de las nuevas funcionalidades propuestas para el Módulo Actividades en la herramienta de autor web CRODA 2.0.
- Las herramientas seleccionadas soportaron el análisis y diseño de la propuesta de solución.
- Se demostró lo beneficioso que resulta la utilización de la metodología de desarrollo de software RUP, lo que propiciará una futura implementación de las nuevas funcionalidades.

RECOMENDACIONES

Para un futuro perfeccionamiento del Módulo Actividades en la herramienta de autor web CRODA 2.0 se recomienda:

- Realizar la implementación de las nuevas funcionalidades propuestas.
- Mantener una constante investigación referida a las actividades de autoevaluación más utilizadas, para preservar la herramienta de autor actualizada.
- Elaborar un manual de ayuda para consulta de los profesores en el trabajo con la herramienta de autor web CRODA 2.0.

REFERENCIAS BIBLIOGRÁFICAS

1. **TTnet, Grupo de trabajo de e-Learning de la red.** *La formación sin distancia.* España: Subdirección General de Formación Ocupacional, 2006.
2. **García Peñalvo, Francisco José.** *Estado Actual de los sistemas e-learning.* 2. Salamanca: Ediciones Universidad de Salamanca, 2005. ISSN 1138 9737.
3. Ayuntamiento de Cartagena. [En línea] 2011. [Citado el: 3 de noviembre de 2011.] <http://welearn.cartagena.es/moodle/mod/book/view.php?id=509&chapterid=11>.
4. **Berlanga, Adriana J , y otros.** *Consideraciones para reforzar el Valor de los Metadatos en los Objetos de Aprendizaje.* Salamanca: Editorial de Universidad de Salamanca, 2006.
5. IMS. [En línea] [Citado el: 10 de enero de 2012.] <http://www.imsglobal.org/content/packaging/index.html>.
6. IMS. [En línea] [Citado el: 10 de enero de 2012.] <http://www.imsglobal.org/digitalrepositories/index.html>.
7. IMS. [En línea] [Citado el: 10 de enero de 2012.] <http://www.imsglobal.org/metadata/index.html>.
8. IMS. [En línea] [Citado el: 10 de enero de 2012.] <http://www.imsglobal.org/question/index.html>.
9. eXe. [En línea] [Citado el: 4 de noviembre de 2011.] <http://exelearning.org>.
10. Udutu. [En línea] [Citado el: 4 de noviembre de 2011.] <http://www.udutu.com>.
11. **Angélica, Patricia Peraza.** *Manual Introductorio al Course Lab.* El Salvador: Universidad Modular Abierta Sonsonate, 2008.
12. Introducción a Jclic. [En línea] Telemática Educativa Andalucía. [Citado el: 5 de noviembre de 2011.] <http://www.juntadeandalucia.es/averroes/jclic/agradecimientos.htm>.
13. **Cardozo, Javier Herrera.** *Las operaciones mentales y su desarrollo en el aula.* 15 de junio del 2009.
14. **Gómez Gallejo, Juan Pablo.** *Fundamentos de la Metodología RUP.* Colombia: Universidad Tecnológica de Pereira, 2007.



15. Introducción a Extreme Programming. [En línea] 2002. [Citado el: 23 de octubre de 2011.] <http://www.dsi.uclm.es/asignaturas/42551/trabajosAnteriores/Presentacion-XP.pdf>.
16. Visual Paradigm for UML 6.4 Community Edition. [En línea] [Citado el: 12 de noviembre de 2011.] <http://www.visual-paradigm.com/product/vpuml/editions/community.jsp>.
17. **Potencier, Fabien y Zaninotto.** *Symfony 1.0, la guía definitiva*. Editorial Apress, 2008. ISBN-13: 978-1590597866.
18. **W3C.** Extensible Markup Language (XML) 1.0 (Fifth Edition). [En línea] 26 de noviembre de 2008. [Citado el: 15 de noviembre de 2011.] <http://www.w3.org/TR/2008/REC-xml-20081126/>.
19. Diccionario de siglas. [En línea] 2008. [Citado el: 18 de noviembre de 2011.] <http://www.elcodigo.net/tutoriales/diccionario.html>.
20. Sencha. [En línea] 2006. [Citado el: 18 de noviembre de 2011.] <http://www.sencha.com/products/js/>.
21. PHP-Hispano. [En línea] 31 de agosto de 2011. [Citado el: 12 de enero de 2012.] <http://www.php-hispano.net/>.
22. PostgreSQL. [En línea] 2007. [Citado el: 18 de noviembre de 2011.] <http://www.postgresql.org>.
23. Oracle Corporation. [En línea] 2011. [Citado el: 18 de noviembre de 2011.] <http://www.oracle.com>.
24. eXist-db Open Source Native XML Database. [En línea] 2008. [Citado el: 16 de febrero de 2012.] <http://exist.sourceforge.net/>.
25. Apache. [En línea] Apache Software Foundation, 2011. [Citado el: 1 de diciembre de 2011.] <http://httpd.apache.org/>.
26. Cibernetia. [En línea] Universidad Abierta de Cataluña, 2010. [Citado el: 1 de diciembre de 2011.] http://www.cibernetia.com/manuales/instalacion_servidor_web/3_otros_servidores_web.php.
27. **Sommerville, Ian.** *Ingeniería del Software*. España: PEARSON EDUCACIÓN, SA., 2005. ISBN:84-7829-074-5.
28. **Jacobson, Ivar, Booch, Grady y Rumbaugh, James.** *El proceso unificado de desarrollo de software*. España: Rational Software Corporation, 2000. 87-7829-036-2.



29. Servicio nacional de aprendizaje. [En línea] SENA, 2009. [Citado el: 5 de abril de 2012.] <http://ciclodevidassoftware.wikispaces.com/arquitectura+de+software>.

30. **otros, Buschmann y.** Definition Of Architectural Pattern. [En línea] 2004. [Citado el: 5 de abril de 2012.]

http://www.crystalclearsoftware.com/cgi-bin/arch_patterns/wiki.pl?Definition_Of_Architectural_Pattern.

BIBLIOGRAFÍA CONSULTADA

- [1]. Angélica, Patricia Peraza. Manual Introductorio al Course Lab. El Salvador: Universidad Modular Abierta Sonsunate, 2008.
- [2]. Apache. [En línea] Apache Software Foundation, 2011. [Citado el: 1 de diciembre de 2011.] <http://httpd.apache.org/>.
- [3]. Ayuntamiento de Cartagena. [En línea] 2011. [Citado el: 3 de noviembre de 2011.] <http://welearn.cartagena.es/moodle/mod/book/view.php?id=509&chapterid=11>.
- [4]. Berlanga, Adriana J, y otros. Consideraciones para reforzar el Valor de los Metadatos en los Objetos de Aprendizaje. Salamanca: Editorial de Universidad de Salamanca, 2006.
- [5]. Cardozo, Javier Herrera. Las operaciones mentales y su desarrollo en el aula. 15 de junio del 2009.
- [6]. CARRODEGUAS, RICARDO. 2008. Análisis y Diseño de una herramienta de autor Web Interoperable. 2008.
- [7]. Ciberneta. [En línea] Universidad Abierta de Cataluña, 2010. [Citado el: 1 de diciembre de 2011.] http://www.ciberneta.com/manuales/instalacion_servidor_web/3_otros_servidores_web.php.
- [8]. Diccionario de siglas. [En línea] 2008. [Citado el: 18 de noviembre de 2011.] <http://www.elcodigo.net/tutoriales/diccionario.html>.
- [9]. eXe. [En línea] [Citado el: 4 de noviembre de 2011.] <http://exelearning.org>.
- [10]. exist, Comunidad. 2009. [En línea] 2009. Disponible en: <http://exist.sourceforge.net/>.
- [11]. EXist-db Open Source Native XML Database. [En línea] 2008. [Citado el: 16 de febrero de 2012.] <http://exist.sourceforge.net/>.



- [12]. Etxeberriá, Ana Landeta. 2007. Libro de buenas prácticas de e-learning. [En línea] 2007. UDIMA Universidad a Distancia de Madrid. Disponible en: <http://www.buenaspracticass-elearning.com/capitulo-16-estandares-e-learning.html>.
- [13]. García Peñalvo, Francisco José. Estado Actual de los sistemas e-learning. 2. Salamanca: Ediciones Universidad de Salamanca, 2005. ISSN 1138 9737.
- [14]. Giancarlo Corzo Deja una respuesta. 2008. Desarrollo en Web. ExtJS lo bueno, lo malo y lo feo. [En línea] 22 de Octubre de 2008. Disponible en: <http://blogs.antartec.com/desarrolloweb/2008/10/extjs-lo-bueno-lo-malo-y-lo-feo/>.
- [15]. Gómez Gallejo, Juan Pablo. Fundamentos de la Metodología RUP. Colombia: Universidad Tecnológica de Pereira, 2007.
- [16]. Guzmán, Clara López. 2005. Los Repositorios de Objetos de Aprendizaje como soporte a un entorno e-learning. España: s.n., 2005.
- [17]. Introducción a Extreme Programming. [En línea] 2002. [Citado el: 23 de octubre de 2011.] <http://www.dsi.uclm.es/asignaturas/42551/trabajosAnteriores/Presentacion-XP.pdf>.
- [18]. Introducción a Jclic. [En línea] Telemática Educativa Andalucía. [Citado el: 5 de noviembre de 2011.] <http://www.juntadeandalucia.es/averroes/jclic/agradecimientos.htm>.
- [19]. IMS. [En línea] [Citado el: 10 de enero de 2012.] <http://www.imsglobal.org/content/packaging/index.html>.
- [20]. IMS. [En línea] [Citado el: 10 de enero de 2012.] <http://www.imsglobal.org/digitalrepositories/index.html>.
- [21]. IMS. [En línea] [Citado el: 10 de enero de 2012.] <http://www.imsglobal.org/metadata/index.html>.
- [22]. IMS. [En línea] [Citado el: 10 de enero de 2012.] <http://www.imsglobal.org/question/index.html>.
- [23]. Jacobson, Ivar, Booch, Grady y Rumbaugh, James. El proceso unificado de desarrollo de software. España: Rational Software Corporation, 2000. 87-7829-036-2.