

Universidad de las Ciencias Informáticas

Facultad 4



**IMPLEMENTACIÓN Y PRUEBAS DE UN SISTEMA DE
NOTIFICACIONES DE RHODA 2.1.**

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

Autor: Yaneilis Lara Pérez.

Tutores:

Ing. Orlando Felipe Salvador Broche

Dr. Augusto César Rodríguez Medina

Ciudad de La Habana, Junio 2011.

“Año 53 de la Revolución”.

DECLARACIÓN DE AUTORÍA

Declaro ser autora del presente trabajo de diploma y otorgo a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma.

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

Yaneilis Lara Pérez

Firma del autor.

Ing. Orlando Felipe Salvador Broche

Firma del Tutor

Dr. Augusto César Rodríguez Medina

Firma del Tutor.

PENSAMIENTO

"El futuro de nuestro país tiene que ser necesariamente un futuro de hombres de ciencia, de hombres de pensamiento".

Fidel Castro Ruz



AGRADECIMIENTOS

Agradezco a la Revolución y al comandante Fidel Castro por haber creado esta universidad y por darme la oportunidad de estudiar en ella.

Agradezco sobre todo a mis padres, por confiar en mí, por hacerme quien soy ahora y por estar siempre al tanto y velando de mí, con todo el cariño, amor y respeto que se merecen, gracias.

A mi hermana Yanelis por preocuparse por mí y porque sé que soy un ejemplo para ella y que algún día llegará a ser una universitaria como yo, te quiero mucho mi hermanita.

En especial a mi novio Felipe Álvarez Beyrute por estar siempre a mi lado y ayudarme y cuidarme en todos los momentos, te amo mi vida.

Agradezco también a la familia de mi novio por preocuparse y apoyarme en todos los momentos de mi vida.

A mi tutor Orlando Felipe Salvador Broche y a los profes Javier Soler y Yandris Mata por su apoyo incondicional.

A mis amigos Deolinda, Yosnaidy y Arisbel por ser como mis hermanos y apoyarme en todo, gracias por todo lo que hicieron por mí, los quiero mucho.

A todos mis amigos y compañeros de aula, gracias de verdad.

A todos los que contribuyeron de una forma u otra en la elaboración de este trabajo, les agradezco de todo corazón.

DEDICATORIA

*A mi mamá y a mi papá, por su confianza y dedicación, y por enseñarme
que uno construye su propio futuro.*

*A mi novio por siempre creer en mí y quererme tanto y porque
sé que soy un ejemplo para él y que logrará hacer su sueño.*

RESUMEN

La evolución del uso de las Tecnologías de la Información y las Comunicaciones en la educación, ha potenciado un desarrollo exponencial del e-learning y con ello, la necesidad de una formación continua en niveles superiores de enseñanza. Una herramienta importante para la educación a distancia son los Repositorios de Objetos de Aprendizaje que se encargan del almacenamiento de los Objetos de Aprendizaje. En la Universidad de Ciencias Informáticas se creó un Repositorio de Objetos de Aprendizaje (RHODA) que permite a los usuarios la búsqueda, creación y utilización de los Objetos de Aprendizaje, pero las informaciones relacionadas con los objetos de aprendizaje son actualizadas diariamente, esto trae consigo que los usuarios se encuentren desactualizados en cuanto a esto se refiere o que tengan que visitar el repositorio diariamente y a cada instante. En esta investigación se propone y desarrolla un sistema para mejorar las notificaciones del repositorio RHODA. Este sistema permite enviar correos a los usuarios interesados de la información que deseen conocer y además brinda la posibilidad de suscribirse mediante RSS a diferentes opciones para mantener a los usuarios actualizados sin necesidad de que visiten el repositorio.

PALABRAS CLAVES: e-learning, notificaciones, objetos de aprendizaje, repositorio.

ÍNDICE DE CONTENIDO

Contenido

INTRODUCCIÓN.....	1
CAPÍTULO 1: Fundamentación teórica.....	5
1.4 Repositorio de Objetos de Aprendizaje	5
1.1.1 Los Repositorios de Objetos de Aprendizaje: su función en el e-learning.....	6
1.1.2 Repositorios de Objetos de Aprendizaje más significativos y sus características	8
1.1.3 Repositorio de Objetos de Aprendizaje RHODA.....	10
1.2 Las notificaciones en sistemas informáticos.....	10
1.2.1 Importancia de las notificaciones en diferentes contextos	12
1.2.2 Notificaciones en Repositorios de Objetos de Aprendizaje	14
1.2.3 Tipos de notificaciones	15
1.3 Really Simple Syndication (RSS)	15
1.3.1 Tecnologías que implementan RSS.....	20
1.4 Protocolos utilizados para el envío de correo electrónico	21
1.4.1 Librerías para el envío de notificaciones por correo electrónico.....	24
1.4.2 Envío de correo electrónico en el Framework Symfony	25
1.5 UML	27
1.6 Visual Paradigm for UML	28
1.7 Metodología de desarrollo.....	29
CAPÍTULO 2: Descripción de la solución propuesta	32
2.1 Arquitectura del Repositorio de Objetos de Aprendizaje.....	32
2.2 Modelo de datos.....	33
2.2.1 Modelo de base de datos relacional	33
2.2.2 Modelo de datos nativos XML.....	34

ÍNDICE DE CONTENIDO

2.3	Módulos involucrados en las notificaciones.....	34
2.3.1	Módulo Gestión de OA	35
2.3.2	Módulo Usuario	49
2.3.3	Módulo Administración	53
2.4	Requisitos no funcionales.....	54
2.5	Estándares de codificación.....	55
2.6	Tareas programadas y las notificaciones	57
CAPÍTULO 3: Validación de la solución propuesta		60
3.1	Pruebas de Caja Negra.....	60
3.1.1	Descripción de los casos de prueba	61
CONCLUSIONES GENERALES.....		65
RECOMENDACIONES.....		66
REFERENCIAS BIBLIOGRÁFICAS.....		67
GLOSARIO.....		69

ÍNDICE DE FIGURAS

Figura 1. Concepto de OA.	7
Figura 2 Diagrama de despliegue.	33
Figura 3 Diagrama de componentes	35
Figura 4 Suscribirse a Mis publicaciones.	36
Figura 5 Canal RSS de Mis publicaciones.	37
Figura 6. Suscribirse a Mis favoritos	38
Figura 7. Canal RSS de Favoritos.....	39
Figura 8. Suscribirse a objetos de aprendizaje en edición.....	40
Figura 9. Canal RSS de OA en edición.....	41
Figura 10. Suscribirse a objetos de aprendizaje en revisión.....	42
Figura 11. Canal RSS de OA en revisión	42
Figura 12 Preferencias.....	43
Figura 13. Ejemplo de correo sobre comentarios a sus OA	44
Figura 14. Ejemplo de correo cuando el comentario ha sido publicado.....	44
Figura 15. Ejemplo de correo cuando el comentario ha sido denegado	45
Figura 16. Ejemplo de correo cuando un OA ha sido aceptado.....	45
Figura 17. Ejemplo de correo cuando un OA ha sido rechazado.....	46
Figura 18. Ejemplo de correo cuando un OA ha sido descargado.....	46
Figura 19. Ejemplo de correo cuando un OA ha sido visualizado.....	46
Figura 20 Ejemplo de correo cuando una solicitud es rechazada.....	47
Figura 21 Ejemplo de correo cuando una solicitud es aceptada.....	47
Figura 22 Ejemplo de correo cuando se recibe una nueva solicitud.....	48
Figura 23 Ejemplo de correo cuando cambie de categoría un OA.	48

ÍNDICE DE FIGURAS

Figura 24 Ejemplo de correo con los OA más solicitados.....	49
Figura 25 Ejemplo de correo cuando una cuenta es creada.	50
Figura 26 Ejemplo de correo cuando una cuenta es bloqueada.....	50
Figura 27 Ejemplo de correo cuando una cuenta es desbloqueada.	51
Figura 28 Ejemplo de correo cuando un rol es modificado.....	51
Figura 29 Ejemplo de correo cuando es añadido a una categoría en el rol de Revisor.	52
Figura 30 Ejemplo de correo cuando es añadido a una categoría en el rol de Autor.....	52
Figura 31 Ejemplo de correo cuando es eliminado del rol.	52
Figura 32 Ejemplo de correo cuando es eliminada una categoría.	53
Figura 33 Ejemplo de correo cuando es eliminada una subcategoría.	53
Figura 34 Ejemplo de correo cuando es renombrada una categoría.	54
Figura 35 Explicación de los campos del Cron.....	58
Figura 36 Ejemplo del fichero de Cron.	58

ÍNDICE DE TABLAS

Tabla 1 CU Gestionar categorías.....	61
Tabla 2 Matriz de datos.....	62

INTRODUCCIÓN

La sociedad se ha visto profundamente transformada con la aparición de las nuevas tecnologías de la información y las comunicaciones (TIC), por lo que actualmente se encuentra inmersa en nuevos paradigmas, los mismos surgen por el avance vertiginoso de estas tecnologías.

El desarrollo científico-técnico alcanzado en el mundo en las últimas décadas, ha llevado al hombre a lograr un desarrollo tan elevado, que ya es casi imposible prescindir de la Informática. Esto ha permitido el desarrollo de herramientas y tecnologías convertidas en aplicaciones puestas al servicio de la sociedad, con el fin de hacerlas cada día más productivas y capaces.

La utilización de estas nuevas herramientas y tecnologías en diversas ramas de la sociedad se ha convertido en un factor determinante, ya que protagonizan el éxito de las mismas alcanzando resultados económicos aceptables y beneficiosos.

Actualmente los sistemas educativos de todo el mundo se enfrentan a la utilización de las tecnologías de la información y la comunicación para proveer a sus educandos con las herramientas y conocimientos necesarios.

Al respecto, se señala que en el área educativa, los objetivos estratégicos apuntan a mejorar la calidad de la educación por medio de la diversificación de contenidos y métodos, promover la experimentación, la innovación, la difusión y el uso compartido de información y de buenas prácticas, la formación de comunidades de aprendizaje y estimular un diálogo fluido sobre las políticas a seguir. Con la llegada de las tecnologías, los métodos convencionales empleados en la educación se han visto obligados a sufrir transformaciones, sustituyendo éstos por el aprendizaje electrónico (e-learning).

La creación de aplicaciones mediante la utilización de las TIC, que fomentan la capacitación docente ha tenido un gran aumento, debido a que constituyen un elemento clave para alcanzar novedades educativas profundas y con amplios alcances. Es importante destacar

que estas aplicaciones estimulan a los integrantes de los grupos a intercambiar ideas, a discutir y decidir en común y a razonar la explicación de sus opiniones.

En la Universidad de las Ciencias Informáticas se encuentra el Centro de Tecnologías para la Formación, FORTES, en el cual existe un proyecto denominado RHODA, donde se desarrolla un Repositorio de Objetos de Aprendizaje, el cual tiene como objetivo primordial apoyar los procesos de enseñanza y aprendizaje, tanto en la universidad como fuera de la misma. Para la mejora de estos procesos es necesario mantener a los usuarios actualizados sin necesidad de visitar la plataforma, debido a que las informaciones relacionadas con los objetos de aprendizaje son actualizadas constantemente, para lograr esto es importante la utilización de las notificaciones.

En la actualidad, el uso de las notificaciones o cualquier tipo de aviso, vía correo, RSS u otros canales, se ha convertido en una manera muy útil de mantener informado a los usuarios. RHODA en su versión 2.0 posee un deficiente sistema de notificaciones, ya que no permite el envío de información al correo electrónico de cada usuario, razón por la cual se hace necesario el desarrollo e implementación de un sistema de notificaciones que admita enviar a los usuarios la mayor cantidad de información posible en sus diferentes variantes, como RSS y a través de correo electrónico, permitiéndoles en todo momento la configuración de la misma. Siendo éste, un aspecto muy importante para la optimización del repositorio.

Teniendo en cuenta lo anteriormente expuesto, se plantea como **problema de investigación** ¿Cómo mejorar el sistema de notificaciones en el Repositorio de Objetos de Aprendizaje (RHODA)?

Se ha definido como **objeto de estudio** el proceso de envío de notificaciones en repositorios de objetos de aprendizaje.

La presente investigación persigue como **objetivo general** Desarrollar un sistema para el Repositorio de Objetos de Aprendizaje (RHODA) que permita enviar notificaciones.

Teniendo como **campo de acción** el sistema de notificaciones del Repositorio de Objetos de Aprendizaje (RHODA).

Planteando como **idea a defender** que con la implementación de un sistema de notificaciones, RHODA contará con un amplio sistema de avisos que mantendrá al usuario informado de los procesos más relevantes del sistema.

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

- Implementar un sistema para el envío de notificaciones a los diferentes usuarios del repositorio.
- Validar los requisitos implementados en dicho sistema.

Para dar cumplimiento a los objetivos mencionados se trazaron como **tareas a realizar**:

- Estudio del estado del arte sobre los sistemas de notificaciones por correo y sindicación en sistemas informáticos.
- Análisis de los diferentes sistemas de notificaciones que pueden incluirse en una aplicación Web.
- Identificación de los requisitos del sistema de notificaciones que se van a incluir en el sistema propuesto.
- Implementación del sistema de notificaciones.
- Validación del sistema implementado.

Desde el punto de vista metodológico se emplearon entre los métodos teóricos el **Histórico – Lógico** a través del cual se profundizó en las tendencias, regularidades y cualidades del objeto de estudio, así como los argumentos que antecedieron al problema, el **Analítico – Sintético** permitió descomponer el tema para realizar el análisis de las teorías, documentos, materiales, repositorios de objetos de aprendizaje, técnicas y herramientas a utilizar, para así procesar la información y llegar a conclusiones, la **Modelación** mediante el cual se pudo representar de manera funcional y gráfica la solución propuesta. Dentro de los métodos

empíricos, se utilizó la **Observación** que se emplea para conocer la situación actual que existe referente al envío de notificaciones en sistemas web.

El documento consta de tres capítulos. Cada uno de ellos aborda diferentes temáticas y logran una mejor distribución del contenido y claridad en las ideas. Estos están estructurados de la siguiente manera.

Capítulo 1. Fundamentación Teórica: Describe el marco teórico referencial del trabajo, se realiza el análisis del estado del arte de los sistemas de notificaciones por correo y sindicación en sistemas informáticos. Se ofrece además, una breve panorámica de las características fundamentales que poseen los sistemas empleados para las notificaciones y se realiza un estudio de las tecnologías en las que se apoya.

Capítulo 2. Descripción de la solución propuesta: Se presenta la solución al problema planteado, es decir, se realiza una descripción detallada de todos los aspectos que se toman en cuenta durante la implementación del sistema de notificaciones.

Capítulo 3. Validación de la solución propuesta: Se describe cómo se realiza la validación de la solución propuesta, a través de pruebas de Caja Negra para comprobar las funcionalidades del sistema.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

CAPÍTULO 1: Fundamentación teórica

En el presente capítulo se exponen los conceptos y aspectos teóricos asociados al problema, el estado del arte de los sistemas de notificaciones por correo y sindicación en sistemas informáticos, lo cual permitirá una mejor comprensión del tema. También se presentan las características de las herramientas utilizadas a nivel mundial para el envío de notificaciones por correo. Se fundamenta la selección de una metodología de desarrollo de software, así como las tecnologías a utilizar para la implementación del sistema.

1.4 Repositorio de Objetos de Aprendizaje

La etimología de la palabra **repositorio** proviene del latín *repositorium*, que significa armario. La mayoría de los diccionarios de lengua española lo definen como: *“Almacén o lugar donde se guardan ciertas cosas”* (1). Este concepto se vio enriquecido o simplemente cambiado con el surgimiento de la era digital, algunos actores más actuales sostienen la definición de que es *“un sitio centralizado donde se almacena y mantiene información digital”* (2).

Los repositorios en general han participado activamente en los entornos e-learning, pero no todos se ajustan naturalmente a las necesidades educativas, se hace necesaria una herramienta capaz de poder explotar todas las potencialidades de un Objeto de Aprendizaje (OA), ya que la mayoría los trata de manera muy general. En la búsqueda de una solución específica, surge lo que se conoce como Repositorio de Objetos de Aprendizaje (ROA), que es una herramienta diseñada para almacenar organizadamente OA, permitiendo así su localización, recuperación, catalogación y reutilización.

Los ROAs pueden clasificarse en dos tipos, están los que almacenan recursos con sus metadatos, donde el objeto y su descripción se encuentran en un mismo sitio, y los que solo almacenan metadatos, que son aquellos que contienen solo la descripción de los OA y una vez localizado el recurso, lo referencia a una fuente externa, generalmente otro repositorio.

“Los ROAs más conocidos comúnmente funcionan de forma independiente (stand-alone). Son aplicaciones con una interfaz web, un mecanismo de búsqueda y listados con algún tipo de

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

clasificación. Otra clase de ROAs operan solo como módulos adicionales a otros productos (LMS o LCMS) que utilizan los contenidos de forma exclusiva y sin que el usuario tenga acceso directo al repositorio. Lo deseable es que los ROAs tengan ambas capacidades, tanto ofrecer una interfaz web, para que los usuarios humanos puedan acceder a la colección, así como la capacidad de comunicarse directamente con las plataformas de aprendizaje y hacer posible la interoperabilidad entre sistemas de diferente naturaleza” (2).

Otra forma de clasificación de los ROAs, son aquellas que tienen en cuenta la manera en que guardan los metadatos, por un lado se pueden apreciar los centralizados, que son aquellos que tienen todos los metadatos en un mismo servidor, y por otro lado y menos comunes de encontrar, están los distribuidos, donde existen varios servidores y cada uno posee un grupo de metadatos, que de alguna forma colaboran entre sí.

Los ROAs al ser bibliotecas digitales deben cubrir al menos los elementos básicos de colección, servicios de valor añadido, personalización y ciclo de vida (2).

Entre las principales características que debe poseer un ROA se encuentran las herramientas de búsqueda, mediante estas se considera la búsqueda a través de palabras clave u otros metadatos, la posibilidad de que el usuario pueda realizar exploraciones en listados predefinidos en alguna categorización o clasificación, así como la capacidad del sistema para notificar a los usuarios sobre eventos determinados en el repositorio y la sindicación de los objetos de aprendizaje.

1.1.1 Los Repositorios de Objetos de Aprendizaje: su función en el e-learning

Antes de seguir abordando acerca de los ROAs, es necesario profundizar en los OA, ya que ambas cosas son inherentes y dependen mutuamente para poder existir, es decir, no se concibe un OA sin un sitio donde persistir en tiempo y espacio, como tampoco tiene sentido un Repositorio de Objetos de Aprendizaje, sin la existencia de aquello que debe almacenar.

La concepción que se tiene de un OA es muy variada, algunos criterios hacen alusión a su filosofía Orientada a Objeto (OO), y otros se pronuncian más hacia sus potencialidades para

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

agilizar el aprendizaje. Es por ello, que antes de continuar se necesita mencionar, de forma reiterada el concepto de OA, que define a: “cualquier recurso con una intención formativa, compuesto de uno o varios elementos digitales, descrito con metadatos, que pueda ser utilizado y reutilizado dentro de un entorno e-learning” (figura 1) (2).

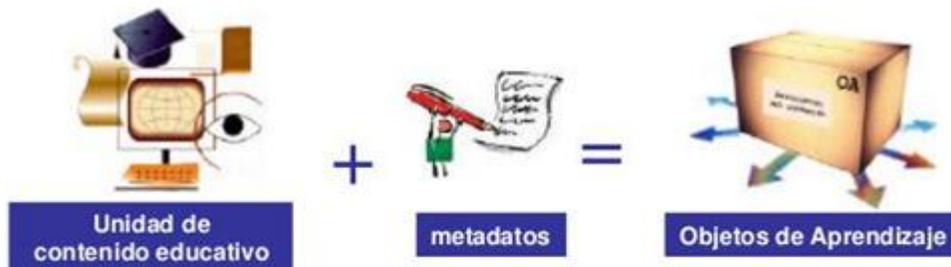


Figura 1. Concepto de OA.

Los OA han ganado seguidores, gracias a sus potencialidades en el proceso de enseñanza-aprendizaje, entre las que se encuentran (2):

- **Reutilización:** el recurso es modular para servir como base o componente de otro recurso. También tiene una tecnología, una estructura y los componentes necesarios para ser incluido en diversas aplicaciones.
- **Accesibilidad:** pueden ser indexados para una localización y recuperación más eficiente, utilizando esquemas estándares de metadatos.
- **Interoperabilidad:** pueden operar entre diferentes plataformas de hardware y software.
- **Portabilidad:** pueden moverse y albergarse en diferentes plataformas de manera transparente, sin cambio alguno en estructura o contenido.
- **Durabilidad:** deben permanecer intactos a las actualizaciones (upgrades) de software y hardware.

Estos atributos dan sentido a los ofrecimientos de los OA como unidades que facilitarán el desarrollo y la expansión global del e-learning. La modularidad que debe caracterizarlos aumenta la versatilidad y la funcionalidad, obteniéndose recursos más disponibles y distribuidos en diferentes sistemas que pueden comunicarse para compartir esfuerzos y reutilizarlos. La

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

creación de OA no es sencilla, pero los esfuerzos y costos de producción se equilibran con las veces que el recurso pueda reutilizarse.

1.1.2 Repositorios de Objetos de Aprendizaje más significativos y sus características

En la Web actualmente existe un gran número de contenidos educativos, que pueden convertirse en OA, sin embargo la evolución del e-learning ha sido acelerada y ha tenido un gran impacto en el ámbito del aprendizaje. Son cada vez más las instituciones de diversas naciones que incorporan estas nuevas tecnologías a sus sistemas cotidianos de enseñanza. No obstante, el hecho de que todavía hay contenidos en la red que no pueden ser localizados para su reutilización, trae como consecuencia que los profesores tengan que elaborar informaciones que ya existen, las publicaciones carecen algunas de una revisión calificada o provienen de fuentes no confiables, algunos contenidos caducan, teniendo aún vigencia y que pueden ser utilizados, todos estos aspectos entorpecen de cierta forma los procesos educativos.

A continuación se mencionan algunos de los ROAs más conocidos y sus características principales.

- **MERLOT (Multimedia Educational Resource for Learning and Online Teaching)**

“MERLOT es sin duda el repositorio más conocido y reconocido, es quizás el que está marcando la pauta para el desarrollo y tendencia de los ROAs. Es un repositorio centralizado que contiene solo los metadatos y apunta a los objetos ubicados en sitios remotos. Es independiente y funciona como un portal de OA. Provee búsquedas y otros servicios como personalización, importación y exportación de objetos. Cualquier usuario puede tener acceso a todos los objetos contenidos en MERLOT y solo los miembros contribuyen agregando objetos, pero para ser miembro no se requiere más que inscribirse y no se adquiere ninguna responsabilidad. La revisión por pares es una actividad que MERLOT utiliza para evaluar la calidad de los objetos agregados” (2).

- **Connexions**

“Esta es una colección que está creciendo muy rápidamente de materiales académicos gratuitos y un poderoso conjunto de herramientas de software para ayudar a los autores a

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

publicar y colaborar, a los profesores a crear y compartir cursos, y a los estudiantes a explorar enlaces entre conceptos, cursos y disciplinas. Los autores están claramente identificados en Connexions. Esta especie de crédito o reconocimiento académico es importante para muchos autores y a menudo es un requisito para ellos el participar. Además, el nombre de autor da autenticidad al trabajo, lo que ayuda más adelante con el control de calidad. Este proyecto está desarrollado por la Rice University” (2). El proyecto de la Rice University es uno de los principales exponentes de los Contenidos Abiertos.

- **CAREO (Campus Alberta Repository of Educational Objects)**

“CAREO es un repositorio centralizado de OA multidisciplinares de profesores de Alberta (Canadá). Es un repositorio independiente que da acceso a objetos remotos y locales a través de los metadatos contenidos en su colección. Cualquier usuario puede tener acceso a los objetos, pero los miembros tienen servicios adicionales, al igual que MERLOT, ser miembro es gratis y abierto a cualquier persona” (2). Desarrollado por las universidades de Alberta, Calgary y Athabasca en cooperación con BELLE (Broadband Enabled Lifelong Learning Environment) y CANARIE (Canadian Network for the Advancement of Research in Industry and Education). Utiliza LOM y es compatible con OAI-PMH. Guarda solo metadatos y está compuesto por dos componentes: la aplicación web Careo y el servidor de administración de metadatos Aloha. Se encuentra programado en Java y es soportado por sistema operativos Unix.

- **SLOR (Simple Learning Object Repository)**

Surge tras la iniciativa de crear un repositorio para plataforma de teleeducación EducAD-UNACH (basada en Moodle). La funcionalidad de SLOR se puede dividir en tres niveles. En el primer nivel se ofrecen objetos compartidos de aprendizaje. A este nivel SLOR tiene la funcionalidad de una biblioteca. Los OA son buscados, recuperados y usados como fueron registrados por el autor original. La estructura de los OA y la arquitectura de SLOR no apoyan su deconstrucción y re-uso con la sola excepción de utilizarlo en plataformas de hardware alternativas (3). La mayoría de los ROAs pertenecen a esta categoría.

1.1.3 Repositorio de Objetos de Aprendizaje RHODA

En la Universidad de Ciencias Informáticas (UCI) se cuenta con el Repositorio de Objetos de Aprendizaje RHODA, del cual se han desarrollado las versiones 1.0 y 2.0 respectivamente, las cuales han sido implantadas en la universidad. Además dicho repositorio cuenta con una personalización en el Ministerio del Poder Popular para la Educación Universitaria en Venezuela (MPPEU), obteniendo una gran aceptación.

RHODA fue creado con el objetivo de apoyar a la comunidad universitaria en la gestión de recursos didácticos, utilizando la tecnología de Objetos de Aprendizaje. Es un lugar para el trabajo metodológico colaborativo, orientado a elevar la calidad de los recursos didácticos. Los objetos de aprendizaje que en él se encuentran son mediados por Revisores, elevando así la calidad de los mismos y del proceso docente educativo en la universidad.

Las versiones desarrolladas cuentan con diferentes opciones que sirven a los usuarios para obtener un recurso con calidad y de interés, y así garantizar la preparación tanto de estudiantes como profesores.

El presente trabajo se basa en la versión 2.1 del producto, la cual se encuentra desarrollando actualmente y permitirá la optimización del repositorio.

1.2 Las notificaciones en sistemas informáticos

Las notificaciones pueden provenir de una gran variedad de orígenes. Un origen común de notificaciones son las credenciales presentadas por un usuario, por ejemplo, como parte de un mensaje enviado a un servicio web. El sistema valida tales notificaciones y se vuelven parte de un conjunto de notificaciones asociado con el usuario. Otros componentes del sistema también pueden ser orígenes de notificaciones, incluidos, aunque no limitados a ellos, el sistema operativo, la pila de red, el entorno de tiempo de ejecución o la aplicación. Además, los servicios remotos también pueden ser un origen de notificaciones.

Por todas estas razones, a lo largo de la historia se han realizado disímiles definiciones, entre las más importantes podemos citar la siguiente:

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

*La etimología de la palabra **notificación** proviene del latín *notificare* y significa comunicar formalmente una resolución o dar una noticia con propósito cierto. También se define como la comunicación oficial que hace una autoridad sobre una conclusión o determinación a la que ha llegado con relación a cierto tema (1).*

La noción de notificación está estrechamente vinculada con una comunicación o un aviso. El envío de una notificación tiene como objetivo dejar asentada determinada resolución o información que se ha tomado o que se tomará en un futuro y que es de vital importancia para esa persona tener conocimiento de la misma.

En el ámbito de la informática, las notificaciones aparecen relacionadas con alertas que emiten ciertos programas o servicios para advertir o actualizar a un determinado usuario interesado en algo específico.

Estas notificaciones han evolucionado en los últimos años, debido al auge de la utilización de las tecnologías de la información y las comunicaciones en conjunto con el aumento de las redes y la necesidad de establecer una permanente comunicación entre los usuarios y las empresas, con el objetivo de mantener informado y actualizado a la persona interesada en un tema específico, constituyendo así una estrecha relación usuario – empresa.

Para la realización de notificaciones electrónicas se requiere el empleo de mecanismos técnicos, tales como servidores de base de datos, servidores de correo electrónico, redes cerradas, como una Intranet o Extranet; y redes abiertas como el Internet (4).

Cuando un sistema envía una notificación por correo electrónico resulta de suma importancia para el usuario al que va destinada, ya que es una manera conveniente de mantenerse actualizado. También es una buena opción para que el administrador de un sitio Web pueda asegurarse de que los cambios que se realizan sean conocidos por sus usuarios.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.2.1 Importancia de las notificaciones en diferentes contextos

En la actualidad, el envío de notificaciones tiene una importancia vital para la ejecución de las distintas estrategias de marketing y fidelización de los clientes, trazadas por las diferentes empresas y organizaciones.

Para obtener una notificación eficaz en su formulación y su entrega se debe tener en cuenta las condiciones de los usuarios a los que van dirigidas tales como etnia, género, etaria, condiciones de discapacidad u otros.

La notificación, por la naturaleza de la materia a que se refiere, debe ser ágil; y así, estar en concordancia con los principios que la fundamentan como inmediación, celeridad, gratuidad, oficiosidad y protección integral (5).

Por lo cual, adquiere particular relevancia realizar la notificación en forma debida garantizándose así la integridad, agilizándose los procedimientos y cumpliéndose con los principios de inmediatez y de celeridad.

Cuando se transmite una notificación falsa se causan graves perjuicios económicos y morales a los involucrados, ya que en esta se definen aspectos personales, como el nombre, domicilio, imagen, correo electrónico, así como derechos de familia tales como hijos, alimentos, matrimonio y aún más derechos patrimoniales como propiedad, posesión, bienes, deudas y herencias, por lo que se hace necesario que este sea un acto seguro, transparente, confiable y eficaz posible.

Las notificaciones resultan de gran importancia en diferentes contextos de la sociedad, entre los cuales se encuentra el comercio, ya que mediante éstas los propietarios de un determinado producto establecen los términos o reglas para comercializar con ellos, encontrando además un medio a través del cual pueden comunicarse con sus clientes sin necesidad de entrevistarse personalmente con ellos.

Desde el punto de vista de la prensa las notificaciones han tenido una gran aceptación ya que por medio de ellas los periodistas pueden mantener un vínculo estrecho entre sus colegas y así

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

estar actualizado de las diferentes noticias y acontecimientos, además de poder contar con las opiniones y comentarios acerca de un tema que deseen tratar o publicar.

En la actualidad las redes sociales tienen más vigencia que nunca si se toman como soportes o sostenes sociales para las personas que conviven en una comunidad dada, constituyen un medio donde mediante acciones organizadas específicas se desarrolla un tejido o red en el que se actúa en pro del bienestar de esa comunidad y por lo tanto de las personas que la integran, actuando como un ámbito donde se resuelven los problemas, convirtiéndose en la respuesta a una necesidad natural de las personas de estar conectadas con otras, por lo cual las notificaciones se han convertido en un punto determinante para establecer dicha conexión entre los diferentes miembros, estrechando así los lazos de amistad entre esas personas y ampliando el contorno de comunicación ya que lo pueden realizar con personas de cualquier parte del mundo.

En el ámbito de la educación se cuenta la educación a distancia, la cual hace referencia a un sistema educativo en el cual los estudiantes y los profesores no se encuentran situados en el mismo lugar y constituye una estrategia para apoyar el proceso de enseñanza - aprendizaje mediante la utilización racional de recursos tecnológicos informáticos y de las telecomunicaciones, con el objetivo de que el proceso docente-educativo y de apropiación del conocimiento resulte más eficaz y eficiente. Su motor impulsor de desarrollo sostenido es la necesidad de diseminar conocimientos y crear habilidades en una población cada vez más ávida y necesitada de ellas, precisamente para satisfacer carencias reales y acceder a un espacio donde la productividad mejore progresivamente en aras de la consecución de un mayor desarrollo social. Para lograr y ampliar estos conocimientos surge la utilización de las notificaciones en estos sistemas ya que mediante éstas los usuarios pueden mantenerse actualizados de las principales actividades o procesos del software sin necesidad de visitarlo. Un ejemplo de estos sistemas son los Repositorios de Objetos de Aprendizaje.

1.2.2 Notificaciones en Repositorios de Objetos de Aprendizaje

La gestión de información ha sido una actividad que durante siglos ha permitido que datos y contenidos sean utilizados para diversas aplicaciones, por diversos grupos e incluso en diferentes lugares geográficos y épocas. Las notificaciones se han convertido en la clave principal para llevar a cabo la gestión de la información en diferentes aplicaciones.

Entre las herramientas más utilizadas para los ambientes o sistemas e-learning están los Sistemas de Administración de Aprendizaje o LMS (Learning Management Systems), también conocidos como plataformas de aprendizaje. *Un LMS es un software basado en un servidor Web que provee módulos para los procesos administrativos y de seguimiento que se requieren para un sistema de enseñanza-aprendizaje, simplificando el control de estas tareas, distinguen además como grupos funcionales de los sistemas de aprendizaje: gestión de cursos, gestión de clases, herramientas de comunicación, herramientas para los estudiantes, gestión del contenido, herramientas de evaluación y gestión de institución educativa (2).*

Los LMS también facilitan el aprendizaje distribuido y colaborativo a partir de actividades y contenidos preelaborados, de forma síncrona o asíncrona, utilizando los servicios de comunicación de Internet como el correo, los foros, las videoconferencias y el chat. (2)

En los ROAs las notificaciones se utilizan para que los usuarios registrados puedan mantenerse actualizados sobre información relacionada a la enseñanza aprendizaje, y obtener así un proceso educativo con mayor calidad tanto para los profesores como para estudiantes.

Estas notificaciones pueden estar relacionadas con información detallada de los objetos de aprendizaje que se encuentran en el repositorio y que pueden servir para ayudar a los estudiantes a adquirir vastos conocimientos sobre un tema de su interés. Además pueden ser utilizadas con la finalidad de obtener opiniones de diferentes usuarios acerca de un tema que se desee tratar.

1.2.3 Tipos de notificaciones

Con el auge de las tecnologías de la información y las comunicaciones, la utilización de las notificaciones en sistemas informáticos se ha convertido en algo imprescindible para el éxito de una aplicación, por lo cual han surgido diferentes maneras de obtener una notificación desde una aplicación Web.

En el caso de las notificaciones por correo ordinario se utilizan para informar a la persona interesada en cualquiera de los casos señalados anteriormente.

En el presente trabajo se tratan el Really Simple Syndication (RSS) y el correo electrónico, debido al auge que han alcanzado en los sistemas informáticos en las últimas décadas, por lo que en los siguientes acápite se abordan aspectos fundamentales relacionados con ellos.

1.3 Really Simple Syndication (RSS)

En la actualidad existen disímiles páginas web que ofrecen la opción de suscribirse a sus canales RSS. Esta es una tecnología que no sólo facilita la lectura de las actualizaciones producidas en los sitios favoritos de los usuarios, sino que además permite mantener a los mismos informados en tiempo real.

Mediante una aplicación en la computadora conectada a Internet, de sencillo manejo y fácil acceso, se pueden obtener todas las novedades de todos los websites favoritos que desee sin necesidad de recurrir al navegador.

Antes de la existencia del RSS la única forma de poder llegar a los nuevos contenidos de las webs más frecuentadas era a través de la visita constante a dichos sitios, un proceso que muchas veces ha sido tedioso a pesar de los atajos conseguidos a través de los marcadores.

Además, luego de llegar a la página elegida, el visitante debía recorrer todo el sitio en búsqueda de las novedades y de los artículos que aún no había leído.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

El Really Simple Syndication surgió de la necesidad de crear un sistema que permitiera a los usuarios conocer en tiempo real las novedades de sus webs favoritas sin necesidad de acudir a otras aplicaciones tales como el navegador.

Debido a la relevancia que ha tomado en los últimos años el RSS, en la actualidad la mayoría de los sitios poseen su propio canal de Feed o fuente Web, algo que los webmaster saben que representa un factor más que importante para lograr una excelente posición en los resultados de las búsquedas.

Una fuente Web (frecuentemente canal web o web feed) es un medio de redifusión de contenido web. Se utiliza para suministrar información actualizada frecuentemente a sus suscriptores. En su jerga, cuando una página web "redifunde" su contenido mediante una fuente web, los internautas pueden "suscribirse" a ella para estar informados de sus novedades. Los interesados pueden usar un programa "agregador" para acceder a sus fuentes suscritas desde un mismo lugar (6).

***RSS** es uno de los principales formatos de una fuente web. Son las siglas de RDF Site Summary o Rich Site Summary, un formato XML para syndicar o compartir contenido en la web. Se utiliza para difundir información actualizada periódicamente a usuarios que se han suscrito a la fuente de contenidos. Este permite distribuir contenidos sin necesidad de un navegador, utilizando un software diseñado para leer estos contenidos RSS. A pesar de eso, es posible utilizar el mismo navegador para ver los contenidos RSS. Las últimas versiones de los principales navegadores permiten leer los RSS sin necesidad de software adicional ya que fue desarrollado específicamente para todo tipo de sitios que se actualicen con frecuencia y por medio del cual se puede compartir la información y usarla en otros sitios web o programas a lo que se le conoce como redifusión web o sindicación web (7).*

Un agregador de noticias es un tipo de software para suscribirse a fuentes de noticias en formatos RSS, Atom y otros derivados de XML/RDF (8).

Este puede reunir las noticias o historias publicadas en los sitios con redifusión web elegidos, y muestra las novedades o modificaciones que se han producido en esas fuentes web; es decir,

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

avisa de cuáles Web han incorporado contenido nuevo desde una última lectura y cuál es ese contenido.

Los formatos de RSS son resultado del empleo explícito de XML, son una especificación genérica para la creación de los formatos de datos. Aunque los formatos de RSS se han desarrollado desde marzo de 1999, el icono de RSS ganó un amplio uso en el período 2005/2006.

Los formatos de RSS fueron precedidos por varias comprobaciones en sindicación, lo cual no alcanzó renombre extenso. La idea básica de reestructurar la información sobre sitios Web va de nuevo desde 1995, cuando Ramanathan V. Guha¹ y otros miembros de Computadora de Apple, el grupo de la tecnología avanzada desarrollaron el Marco contenido del Meta (MCF).

El resumen del sitio de RDF, la primera versión de RSS, fue creado por Guha en Netscape en marzo de 1999 para el uso en el portal de My.Netscape.Com. Esta versión se conocía como RSS 0.9. En julio de ese propio año, se produjo una nueva versión, RSS 0.91, la cual simplificó el formato anterior quitando elementos e incorporando otros. Libby RSS también fue retitulado al resumen rico del sitio y al desarrollo posterior contorneado del formato en un “documento de los futuros”.

Esto se convirtió en una participación pasada de Netscape durante el progreso de RSS en ocho años, debido a que mientras RSS era utilizado por un gran número de editores y portales, la ayuda propuesta por esta compañía cayó en abril de 2001 por consecuencia de la reestructuración de la misma, quitando además, la documentación y las herramientas que apoyaban a dicho formato.

¹ **Ramanathan V. Guha** (1965) es un indio científico de la computación. Se graduó en el Instituto Indio de Tecnología de Madrás. Desde mayo de 2005, ha estado trabajando en Google.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

Winer² publicó luego una versión modificada de la especificación de RSS 0.91 en el sitio Web de UserLand, cubriendo cómo era utilizado en los productos de su compañía, y el copyright demandado al documento. Algunos meses más adelante, UserLand archivó un documento de la marca registrada para RSS, pero la petición del examinador de la marca registrada fue rechazada en diciembre de 2001.

En diciembre de 2000 se publicaron algunas ideas de RSS 0.93 y de RSS 0.94 los cuales fueron retirados posteriormente. En septiembre de 2002, se lanzó una nueva versión importante del formato RSS 2.0, que agregó calidad en las ideas de RSS 0.94 y en la ayuda para los *namespaces*.

En julio de 2003, el software de Winer y de UserLand asignó el copyright de la especificación de RSS 2.0 a Harvard. Al mismo tiempo, lanzó el Tablero consultivo de RSS con Brent Simmons y Jon Udell, un grupo que tenía como propósito mantener y publicar las preguntas de la especificación y las respuestas sobre el formato.

En enero de 2006, Rogers Cadenhead relanzó el Tablero consultivo de RSS, sin Dave Winer, con un deseo indicado de continuar el desarrollo del formato de RSS y de resolver ambigüedades. (9)

Según lo planteado anteriormente, existen diversas versiones de RSS, resumiéndolas en dos ramas importantes (RDF y 2.*).

La rama de RDF, o de RSS 1.* incluye las versiones siguientes (9):

- *RSS 0.90 era la versión original de Netscape RSS. Este RSS fue llamado RDF Resumen del sitio, pero fue basado en un bosquejo de trabajo temprano del estándar de RDF, y no era compatible con la recomendación final de RDF.*

² **Dave Winer** (2 de mayo de 1955 en Brooklyn, Nueva York, EE.UU) es un empresario y escritor. Un pionero en las áreas de los esquemas de gestión de contenidos, XML-RPC, RSS, OPML. Es el autor de Scripting News, una de las más antiguas weblogs, creado en 1997.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

- *RSS 1.0 es un formato abierto por el Grupo de funcionamiento de RSS-DEV, otra vez estando parado para Resumen del sitio de RDF. RSS 1.0 es un formato de RDF como RSS 0.90, pero no totalmente compatible con él, puesto que 1.0 se basa en la recomendación final de RDF 1.0.*
- *RSS 1.1 es también un formato abierto y se piensa poner al día y substituir RSS 1.0. La especificación es un bosquejo independiente no apoyado o no endosado de cualquier manera por el grupo de funcionamiento RSS-Revelador o de cualquier otra organización.*

La rama de RSS 2.* (inicialmente UserLand, ahora Harvard) incluye las versiones siguientes (9):

- *RSS 0.91 es la versión simplificada de RSS lanzada por Netscape, y también el número de la versión simplificada defendida originalmente por Dave Winer del software de Userland. La versión de Netscape ahora fue llamada Resumen rico del sitio; este era un formato de RDF, pero relativamente fácil de utilizar. Sigue siendo la variante más común de RSS.*
- *RSS 0.92 a 0.94 son las extensiones del formato de RSS 0.91, que son sobre todo compatibles con 1.* y con la versión de Winer de RSS 0.91, pero no son compatibles con RSS 0.90. En todas las especificaciones de Userland RSS 0.9x, no era más que siglas.*
- *RSS 2.0.1 tiene la versión número interna 2.0. RSS 2.0.1 fue proclamado “para ser congelado”, pero todavía para ser puesto al día poco después de su lanzamiento sin cambiar el número de versión. RSS ahora estaba parado para sindicación realmente simple. El cambio principal en esta versión es un mecanismo explícito de la extensión usando namespaces de XML.*

Las versiones más recientes en cada rama son compatibles con las anteriores y ambas incluyen mecanismos correctamente documentados de la extensión, usando XML namespaces, o directamente (en la rama 2.*) o con RDF (en la rama 1.*).

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

La especificación de RSS 2.0 fue puesta al día más adelante para incluir ejemplos del HTML entidad-codificado; sin embargo, todos los usos llanos anteriores del texto siguen siendo válidos. (9)

RSS a su vez está compuesto por varios módulos los cuales tienen como objetivo principal ampliar el básico XML, que es el esquema establecido para alcanzar una sindicación más robusta del contenido. Esto exclusivamente permite realizar de diferentes maneras las transacciones de información sin transformar la especificación de la base RSS.

Gracias a los lectores de feeds (programas o sitios que permiten leer fuentes RSS) se pueden obtener resúmenes de todos los sitios que se deseen, desde el escritorio de tu sistema operativo, programas de correo electrónico o por medio de aplicaciones web que funcionan como agregadores, es decir que no es necesario abrir el navegador y visitar las webs.

1.3.1 Tecnologías que implementan RSS

Según lo planteado anteriormente RSS es un formato que basa su contenido específicamente en XML, por lo cual esta es la principal tecnología que lleva a cabo su implementación.

XML es un Lenguaje de Etiquetado Extensible muy simple, pero estricto que juega un papel fundamental en el intercambio de una gran variedad de datos. Es un lenguaje muy similar a HTML pero su función principal es describir datos y no mostrarlos como es el caso de HTML. XML es un formato que permite la lectura de datos a través de diferentes aplicaciones y sirve para estructurar, almacenar e intercambiar información (10).

Es una tecnología definida como sencilla a pesar de estar complementada por otras que la hacen a su vez más grande, obteniendo así amplias posibilidades en la sociedad de la información. Esto le permite jugar un papel muy importante en la actualidad, ya que su principal objetivo es compartir información en disímiles aplicaciones y soportes, siempre de manera segura, fiable y fácil.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

Siendo interesante además, en el campo de la informática, ya que la tendencia actual de las diferentes aplicaciones y sistemas es ser interoperables, permitiendo grandes avances a la hora de trabajar con ellos.

De ahí que el XML posea diferentes ventajas en algunos campos (10):

1. **Comunicación de datos:** Si la información se transfiere en XML, cualquier aplicación podría escribir un documento de texto plano con los datos que estaba manejando en formato XML y otra aplicación recibir esta información y trabajar con ella.
2. **Migración de datos:** Si se tiene que mover los datos de una base de datos a otra sería muy sencillo si las dos trabajasen en formato XML.
3. **Aplicaciones web:** Hasta ahora cada navegador interpreta la información a su manera y los programadores del web tiene que hacer unas cosas u otras en función del navegador del usuario. Con XML se tiene una sola aplicación que maneja los datos y para cada navegador o soporte se puede tener una hoja de estilo o similar para aplicarle el estilo adecuado.

1.4 Protocolos utilizados para el envío de correo electrónico

El correo electrónico es considerado el servicio más utilizado de Internet. En la actualidad existen un gran número de protocolos TCP/IP que ofrecen a su vez una gama de protocolos que permiten una fácil administración del enrutamiento del correo electrónico a través de la red.

Los protocolos de correo electrónico son los que permiten que máquinas en las que se ejecutan sistemas operativos diferentes y que utilizan clientes de correo distintos se comuniquen y transfieran el correo electrónico (11).

Protocolo simple de transferencia de correo

El protocolo SMTP (Protocolo simple de transferencia de correo) es el protocolo estándar que permite la transferencia de correo de un servidor a otro mediante una conexión punto a punto.

Éste es un protocolo que funciona en línea, encapsulado en una trama TCP/IP. El correo se envía directamente al servidor de correo del destinatario. El protocolo SMTP funciona con

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

comandos de textos enviados al servidor con igual nombre. A cada comando enviado por el cliente le sigue una respuesta del servidor compuesta por un número y un mensaje descriptivo (12).

Un punto importante sobre este protocolo es que no requiere autenticación. Esto permite que cualquiera en la Internet pueda enviar correo a cualquiera otra persona o a grandes grupos de personas. Esta característica es lo que hace posible el correo basura o spam. Los servidores SMTP modernos intentan minimizar este comportamiento permitiendo que sólo los hosts conocidos accedan a dicho servidor. Los servidores que no ponen tales restricciones son llamados servidores **open relay** (13).

Protocolo de oficina de correos

El protocolo POP (Protocolo de oficina de correos), como su nombre lo indica, permite recoger el correo electrónico en un servidor remoto llamado servidor POP. Este es necesario para las personas que no están permanentemente conectadas a Internet, ya que así pueden consultar sus correos electrónicos recibidos sin que ellos estén conectados.

Existen dos versiones principales de este protocolo, POP2 y POP3, a los que se le asignan los puertos 109 y 110 respectivamente, y que funcionan utilizando comandos de texto radicalmente diferentes.

Al igual que con el protocolo SMTP, el protocolo POP (POP2 y POP3) funciona con comandos de texto enviados al servidor. Cada uno de estos comandos enviados por el cliente está compuesto por una palabra clave, posiblemente acompañada por uno o varios argumentos, y está seguido por una respuesta del servidor compuesta por un número y un mensaje descriptivo (12).

La ventaja principal que tiene este protocolo es que carpetas, mensajes, etc. se guardan en el ordenador, lo cual permite leer el correo recibido sin estar conectado a la red. Además, al leer los mensajes y bajarlos al ordenador, se libera espacio en el buzón de correo, con lo cual se tiene menos probabilidades que por descuido se llene el mismo y no se pueda recibir más

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

mensajes. Es el más extendido (prácticamente todos los programas de correo lo soportan) y es el ideal para conectarse siempre desde un mismo ordenador.

A diferencia del SMTP, este protocolo requiere autenticación de los clientes usando un nombre de usuario y una contraseña. Por defecto, las contraseñas son pasadas a través de la red de forma encriptada (13).

Cuando se utiliza este protocolo, los mensajes de correo son descargados a través de las aplicaciones de correo cliente. Por defecto, la mayoría de los clientes de correo POP son configurados automáticamente para borrar el mensaje en el servidor de correo después que éste ha sido transferido exitosamente, sin embargo esta configuración se puede cambiar.

Para añadir seguridad, es posible utilizar la encriptación Secure Socket Layer (SSL) para la autenticación del cliente y las sesiones de transferencias de datos (14).

Protocolo de acceso a mensajes de Internet

El protocolo IMAP (Protocolo de acceso a mensajes de Internet) es un protocolo alternativo al de POP3, pero que ofrece más posibilidades (15):

- *Permite administrar diversos accesos de manera simultánea.*
- *Permite administrar diversas bandejas de entrada.*
- *Brinda más criterios que pueden utilizarse para ordenar los correos electrónicos.*

La principal diferencia que tiene respecto al resto de los protocolos es que tanto los mensajes como las carpetas se guardan en el buzón de correo y en el ordenador. Esto es muy útil para conectarse desde ordenadores compartidos, ya que los mensajes no pueden ser leídos por terceras personas, al no quedarse en el PC, además, si no se cuenta con la posibilidad de conectarse siempre del mismo ordenador, se consigue siempre acceder a la totalidad de los mensajes (15).

Cuando se utiliza un servidor de correo IMAP, los mensajes de correo se mantienen en el servidor donde los usuarios pueden leerlos y borrarlos. También permite a las aplicaciones cliente crear, renombrar o borrar directorios en el servidor para organizar y almacenar correo.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

IMAP lo utilizan principalmente los usuarios que acceden a su correo desde varias máquinas. El protocolo es conveniente también para usuarios que se estén conectando al servidor de correo a través de una conexión lenta, porque sólo la información de la cabecera del correo es descargada para los mensajes, hasta que son abiertos, ahorrando de esta forma ancho de banda. El usuario también tiene la habilidad de eliminar mensajes sin verlos o descargarlos (15).

1.4.1 Librerías para el envío de notificaciones por correo electrónico

Para el envío de los correos electrónicos además de los protocolos se utilizan librerías que facilitan el tratamiento de este tipo de notificaciones en las diferentes aplicaciones Web. Entre las más utilizadas podemos encontrar:

JavaMail se trata de una librería desarrollada por SUN, encaminada al envío de correos electrónicos directamente desde una aplicación Java. Esta librería implementa el protocolo SMTP (Simple Mail Transfer Protocol) así como los distintos tipos de conexión con servidores de correo -TLS, SSL, autenticación con usuario y contraseña, y además no se incluye en la JDK ni en la JRE, sino que debe conseguirse como un paquete externo (16).

PHP Mailer es una librería escrita en PHP para envío de correos desde una web utilizando una cuenta válida de correo de un servidor. Incluye diferentes funciones como son (17):

- Envíos a varios remitentes, con CC³, CCO⁴, etc.
- Autenticación por SMTP.
- Envíos con HTML.
- Envíos de correo con adjuntos.
- Inclusión de imágenes en el correo.

³ Campo **CC** (Copia de Carbón): quienes estén en esta lista recibirán también el mensaje, pero verán que no va dirigido a ellos, sino a quien esté puesto en el campo **Para**.

⁴ Campo **CCO** (Copia de Carbón Oculta): una variante del **CC**, que hace que los destinatarios reciban el mensaje sin aparecer en ninguna lista.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

- Posibilidad de **enviar archivos adjuntos**.

Swift Mailer es una potente librería que permite el envío de emails y que entre las características que ofrece está que cuenta con una cabecera de control RFC 2822, internacionalización, balance de carga, soporte para SSL y TLS (para servidores GMail), imágenes embebidas, MIME 1.0 (adjuntos) (18).

Esta es la librería que se utiliza para la implementación del sistema propuesto debido a todas las características mencionadas anteriormente y por estar integrada en el framework de desarrollo Symfony, el cual se emplea para el desarrollo del mismo.

1.4.2 Envío de correo electrónico en el Framework Symfony

La palabra inglesa **framework** define, en términos generales, un conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular, que sirve como referencia para enfrentar y resolver nuevos problemas de índole similar. En el desarrollo de software se define como una estructura conceptual y tecnológica de soporte definida, normalmente con artefactos o módulos de software concretos, con base en 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 (19).

ZendFramework⁵ es un framework de código abierto para PHP desarrollado por Zend, empresa encargada de la mayor parte de las mejoras hechas a PHP, por lo que se podría decir que es el framework "oficial". Implementa el patrón Modelo-Vista-Controlador, es orientado a objetos y sus componentes tienen un bajo acoplamiento por lo que se pueden usar en forma independiente.

CodeIgniter⁶ es un framework de código abierto que permite desarrollar aplicaciones web con

⁵ <http://www.zend.com/en/>

⁶ <http://codeigniter.com/>

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

PHP. Es pequeño, corre muy rápido, está bien documentado y es fácil de aprender siempre que se posean conocimientos de PHP. Utiliza el patrón de diseño de software Modelo-Vista-Controlador, lo que facilita mucho estructurar la aplicación y su mantenimiento, además posee helpers, bibliotecas, que facilitan y agilizan mucho el desarrollo.

Symfony pone a la disposición del desarrollador un conjunto de utilidades para la gestión completa de una aplicación, así como para la automatización de ciertas actividades como: la generación de código SQL, la creación del Modelo de Datos, la creación de módulos, etc., cuenta con diferentes entornos de ejecución siendo los más importantes el entorno de Desarrollo, que cuenta con un sistema de depuración potente, y el entorno de Producción para la ejecución de una aplicación ya en producción, aunque existen otros entornos y se pueden definir nuevos (20).

El framework Symfony siempre ha hecho uso de las mejores ideas y librerías de otros proyectos de software libre, incluso aunque pertenezcan a otros frameworks u otros lenguajes de programación. Además, los creadores de Symfony creen firmemente que antes de programar algo ellos mismos, siempre es mejor comprobar si existe algún proyecto de software libre que resuelva esa necesidad.

La gestión del correo en Symfony es centrada alrededor de un objeto llamado mailer, el cual es obtenido desde la acción con un método nombrado getMailer().

El método compose() toma cuatro argumentos y devuelve un objeto de mensaje de email (21):

- *el remitente (from);*
- *destinatario(s) (to);*
- *el asunto del mensaje;*
- *el cuerpo del mensaje;*

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

El enviar el mensaje es entonces tan simple como llamar a un método `send()` desde la instancia `mailer` y pasarle el mensaje como un argumento, aunque se puede solo escribir y enviar un correo en un método usando `composeAndSend()`.

La configuración `delivery_strategy` le dice a `Symfony` como entregar correos. Por defecto, cuenta con cuatro diferentes estrategias (21):

- 1. `realtime`: Los Mensajes son entregados en tiempo real.*
- 2. `single_address`: Los Mensajes son entregados a una solo dirección.*
- 3. `pool`: Los Mensajes son guardados en cola.*
- 4. `none`: Los Mensajes son simplemente ignorados.*

Por las características planteadas anteriormente y por formar parte de la arquitectura del repositorio RHODA, se seleccionó como framework de desarrollo `Symfony`.

1.5 UML

UML (Unified Modeling Language) conocido en español como el Lenguaje Unificado de Modelado desempeña un papel protagónico en cualquier fase de una metodología de desarrollo. Es un lenguaje que permite modelar artefactos, relaciones, entidades, procesos, métodos de una situación determinada. Cabe recalcar que UML no es un lenguaje de programación, su objetivo es solo modelar situaciones, y se encuentra muy relacionado con la programación Orientada a Objetos, siendo el complemento que permite culminar de una forma directa lo que se modela con UML. En UML existen varios tipos de diagramas los cuales se agrupan en varias categorías por el tipo de ente que modelan (20).

Existen dos clasificaciones básicas basándose en el fin que persiguen los modelos que agrupa (20):

- **Diagramas de estructura:** agrupa seis diagramas dedicados a especificar cómo se modela desde un punto de vista estructural alguna situación del problema a resolver.

- **Diagramas de comportamiento:** agrupa siete diagramas dedicados a modelar el comportamiento, ya sea del sistema o de algún proceso del negocio que se pretende automatizar o servicio a brindar.

1.6 Visual Paradigm for UML

Las herramientas **CASE** (*Computer Aided Software Engineering*, Ingeniería de Software Asistida por Computadora) están definidas como aplicaciones informáticas que pueden ayudar durante el ciclo de vida de desarrollo de un software en aspectos específicos como el proceso de realizar un diseño del proyecto, cálculo de costes, implementación de parte del código automáticamente con el diseño dado, compilación automática, documentación o detección de errores entre otras (22).

Rational Rose es la herramienta CASE que propone la utilización de cuatro tipos de modelo para realizar un diseño del sistema, utilizando una vista estática y otra dinámica de los modelos del sistema, uno lógico y otro físico. Permite crear y refinar estas vistas creando de esta forma un modelo completo que representa el dominio del problema y el sistema de software. Además proporciona mecanismos para realizar la denominada Ingeniería Inversa, es decir, a partir del código de un programa, se puede obtener información sobre su diseño.

Visual Paradigm es otra herramienta CASE profesional, que soporta UML hasta la versión 2.1 así como un grupo de funcionalidades que facilitan el trabajo de cualquier rol dentro de un proyecto, dando además, un fuerte soporte durante todo el desarrollo de un software. Ayuda a una rápida construcción de aplicaciones con calidad y a un menor esfuerzo y permite modelar diagramas de clases, realizar código inverso y es capaz de generar a partir de un proyecto determinado todo un sitio de documentación o código con las descripciones de cada uno de los diagramas diseñados.

Cabe mencionar que una de las razones que hace muy interesante al Visual Paradigm es que es multiplataforma, puede ser usada en una versión de GNU/Linux, obteniendo los mismos

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

resultados que en cualquier otro sistema operativo y a la vez es capaz de integrarse con un gran número de entornos de desarrollo.

Por las características que posee esta herramienta se seleccionó para la modelación de los diferentes artefactos relacionados con el sistema a implementar. Además de formar parte de las herramientas utilizadas en el desarrollo del repositorio.

1.7 Metodología de desarrollo.

Las metodologías de desarrollo de software abarcan todo el ciclo de vida del software, y se definen como “*un conjunto de procedimientos, técnicas, herramientas y un soporte documental que ayuda a los desarrolladores a realizar un nuevo software*” (23), adoptando la misma se obtiene un producto más predecible y permite características deseables como: existencias de reglas bien definidas, verificaciones intermedias, planificación y control, comunicación efectiva y utilización sobre un abanico amplio de proyectos.

XP (Extreme Programming) es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. XP se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico. Los principios y prácticas son de sentido común pero llevadas al extremo, de ahí proviene su nombre (24).

RUP (Rational Undefined Process) es un proceso de Ingeniería de Software (ISW) propuesto por Rational Software Corporation para la construcción completa del ciclo de ISW. Permite la productividad en equipo y la realización de mejores prácticas de software a través de plantillas y herramientas que lo guían en todas las actividades de desarrollo del software. Es un producto que unifica las disciplinas en lo que a desarrollo de software se refiere, incluyendo modelado de negocio, manejo de requerimientos, componentes de desarrollo, ingeniería de datos, manejo y

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

configuración de cambios, y pruebas, cubriendo todo el ciclo de vida de los proyectos basado en la construcción de componentes y maximizando el uso del UML.

Se resume en tres aspectos definitorios: dirigido por casos de uso, iterativo e incremental y centrado en la arquitectura, lo que hace único al proceso. RUP no es un sistema con pasos firmemente establecidos, sino un conjunto de metodologías adaptables al contexto y necesidades de cada organización. Está basado en 5 principios clave que son (25):

- El proceso deberá adaptarse a las características propias del proyecto u organización. El tamaño del mismo, así como su tipo o las regulaciones que lo condicionen, influirán en su diseño específico. También se deberá tener en cuenta el alcance del proyecto.
- Los requerimientos de los diversos participantes presentan diferencias, contradicciones o disputarse recursos limitados. Debe encontrarse un balance que satisfaga los deseos de todos. Debido a este balanceo se corrigen desacuerdos que surgen en el futuro.
- Los proyectos se entregan, aunque sea de un modo interno, en etapas iteradas. En cada iteración se analiza la opinión de los inversores, la estabilidad y calidad del producto, y se refina la dirección del proyecto así como también los riesgos involucrados.
- Este principio dominante motiva el uso de conceptos reutilizables tales como patrón del software, marcos de trabajo (frameworks) por nombrar algunos. Esto evita que los ingenieros de software vayan directamente de los requisitos a la codificación de software a la medida del cliente, sin saber con certeza qué codificar para satisfacer de la mejor manera los requerimientos y sin comenzar desde un principio pensando en la reutilización del código. Un alto nivel de abstracción también permite discusiones sobre diversos niveles y soluciones arquitectónicas. Estas se pueden acompañar por las representaciones visuales de la arquitectura, por ejemplo con el lenguaje UML.
- El control de calidad no debe realizarse al final de cada iteración, sino en todos los aspectos de la producción. El aseguramiento de la calidad forma parte del proceso de desarrollo y no de un grupo independiente.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

Por todas las características mencionadas anteriormente que hacen que RUP sea una de las metodologías más usadas en el mundo y debido a que se utilizó para el desarrollo del repositorio se continúa su uso en la implementación del sistema propuesto.

A través de la investigación realizada en este capítulo se determinó que para el desarrollo del sistema se utilizará el framework de desarrollo Symfony debido a que es utilizado en la implementación del repositorio y además cuenta con potentes librerías para el envío de correos, la herramienta de modelado Visual Paradigm y la metodología de desarrollo RUP por las características analizadas anteriormente.

CAPÍTULO 2: Descripción de la solución propuesta.

CAPÍTULO 2: Descripción de la solución propuesta

En el presente capítulo se exponen los principales elementos arquitectónicos de la aplicación, y se hace un análisis del modelo de datos que presenta la misma. Además se abordan aspectos relacionados con los módulos por los que está compuesta dicha aplicación y su vinculación con las notificaciones.

2.1 Arquitectura del Repositorio de Objetos de Aprendizaje

El Repositorio de Objetos de Aprendizaje es una aplicación web, que se basa en una arquitectura de tipo Cliente-Servidor ya que se hace necesario el uso de las funcionalidades que esta posee para su implementación, además la aplicación se desarrolla usando como framework de desarrollo Symfony. Para el diseño de los diferentes módulos con que cuenta la aplicación se usa el patrón Modelo Vista Controlador, este estilo define el modelo, la vista y la lógica de control en 3 componentes distintos, los cuales interactúan entre sí, dando la salida final al usuario, conociendo cada componente de los demás, solo lo necesario, por lo que una solución basada en este tipo de arquitectura es muy flexible, también utiliza los mecanismos y utilidades existentes en el framework de desarrollo.

Se recomienda que el sistema sea instalado en un servidor de aplicaciones apache usando PHP 5, sobre el sistema operativo Debian, donde se debe usar a su vez un gestor de base de datos PostgreSQL 8.4 y un servidor de base de datos eXist versión 1.4, el sistema es capaz de autenticar usuarios basándose en un servidor LDAP. Para el almacenamiento de los ficheros es recomendable utilizar los servicios de un Storage Area Network (SAN).

Para una mejor comprensión de esta arquitectura a continuación se presenta el modelo de despliegue del mismo.

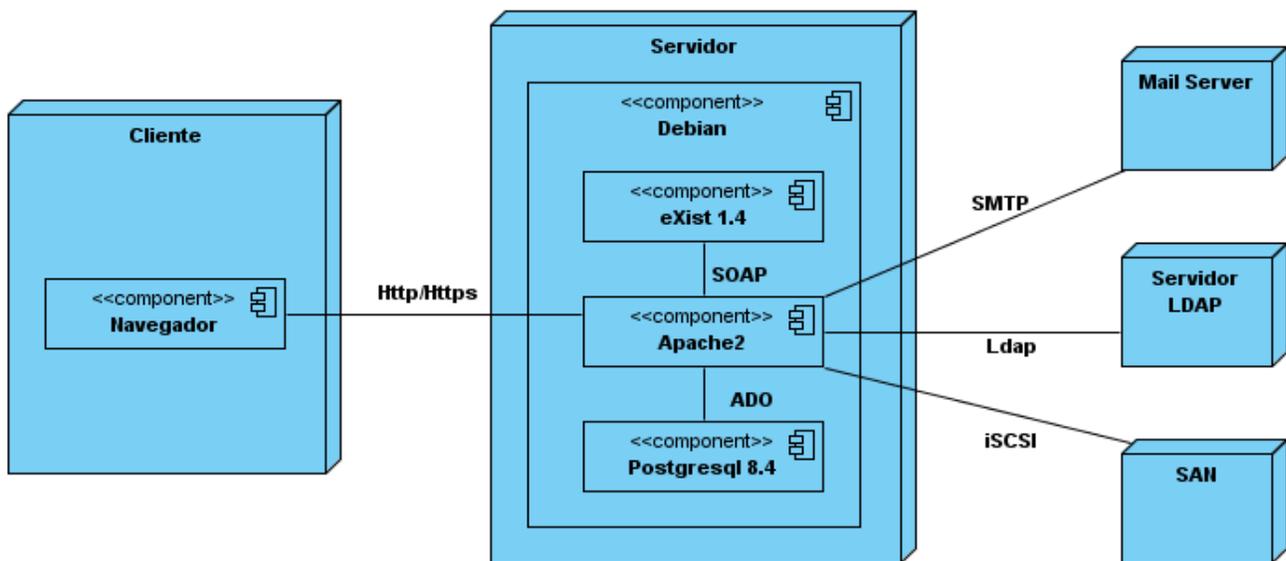


Figura 2 Diagrama de despliegue.

2.2 Modelo de datos

El Repositorio de Objetos de Aprendizaje usa básicamente una base de datos relacional y otra nativa XML, para almacenar documentos XML usados en los objetos de aprendizaje para facilitar su uso, así como la indexación para búsquedas por metadatos. En los acápites siguientes se exponen criterios que se tienen en cuenta en el diseño de estos dos modelos.

2.2.1 Modelo de base de datos relacional

El esquema relacional de la aplicación cuenta con 50 tablas, muchas de las cuales se encuentran interrelacionadas y otras sirven solamente para llevar tabulación de las diferentes relaciones de tipo muchos-a-muchos, las principales relaciones del mismo son:

1. **roa_user**: almacenar datos de los usuarios.
2. **rol**: para llevar los datos de los diferentes roles del sistema.
3. **message**: para temas relativos a la mensajería interna del sistema.

CAPÍTULO 2: Descripción de la solución propuesta.

4. **category:** para tabular las categorías y sub-categorías del sistema.
5. **visits:** lleva las trazas de las visitas de los usuarios del sistema.
6. **log:** el sistema de trazas se guarda en esta relación.
7. **pif:** guarda información relacionada con los diferentes objetos existentes en el repositorio.
8. **revisor_records:** relación que se utiliza para guardar el historial de revisiones.
9. **notify_general:** guarda información relacionada con las notificaciones de los usuarios.

La más importante para la implementación del sistema propuesto es la **notify_general** debido a la información que guarda y a la vez esta se encuentra directamente relacionada con la tabla **roa_user** ya que necesita de los principales datos de los usuarios para el envío de dichas notificaciones.

2.2.2 Modelo de datos nativos XML

Para la gestión de los documentos XML que el sistema maneja, exactamente el `imsmanifest.xml` concerniente a cada objeto de aprendizaje, se utiliza una base de datos nativa XML, siendo eXist el gestor utilizado. Para la recuperación de datos acerca de los mismos se usa XQuery y XUpdate para la actualización de dichos documentos.

El modelo consiste básicamente en una colección, la cual a su vez se encuentran compuesta por 3 subcolecciones: **edition** donde se almacenan los objetos de aprendizaje en edición, **edited** donde se encuentran los objetos de aprendizaje en revisión y **revised** donde se almacenan los objetos de aprendizaje publicados.

2.3 Módulos involucrados en las notificaciones

RHODA cuenta con diferentes módulos en su implementación, algunos de ellos a su vez se encuentran estrechamente vinculados con las notificaciones, tanto por correo electrónico como la sindicación de contenidos mediante RSS. En este acápite se abordan los principales módulos involucrados en las notificaciones, así como las funcionalidades implementadas en los mismos.

CAPÍTULO 2: Descripción de la solución propuesta.

Para esto es necesaria la utilización de la clase **NotifyGeneralPeer**, la cual cuenta en su implementación con las diferentes notificaciones que son enviadas a los usuarios, esta a su vez se encuentra relacionada con las clases **Pif**, **User** y **Config** las cuales gestionan información sobre los objetos de aprendizaje, los usuarios y la parte de administración respectivamente, muy importantes para el envío de las diferentes notificaciones.

Para la mejor comprensión de estas relaciones a continuación se presenta el diagrama de componentes.



Figura 3 Diagrama de componentes

2.3.1 Módulo Gestión de OA

Este módulo es el encargado de todas las acciones referente a los objetos de aprendizaje.

Después de realizar un estudio exhaustivo de la aplicación se comprobó que las opciones más visitadas por los usuarios son: **Mis publicaciones**, **Favoritos**, **Objetos de aprendizaje en edición** y **Objetos de aprendizaje por revisar**, por lo cual se determinó darle la posibilidad a los usuarios de suscribirse mediante RSS a dichos contenidos.

A continuación se explican las principales notificaciones contenidas en este módulo:

Suscribirse a mis publicaciones.

Brinda la posibilidad a los usuarios de mantenerse actualizados de los objetos de aprendizaje de su autoría. Estos pueden suscribirse dando clic como se muestra en la figura siguiente:

CAPÍTULO 2: Descripción de la solución propuesta.

The screenshot shows a web application interface with the following components:

- Búsqueda:** A search bar with a 'Buscar' button and a link to 'Búsqueda avanzada'.
- Menú:** A list of navigation links including 'Mis publicaciones', 'Favoritos', 'Mi perfil', 'Editar perfil', 'Mis mensajes (0)', 'Foros', 'Crear objetos de aprendizaje', 'Importar objetos de aprendizaje', 'Objetos de aprendizaje en edición', 'Objetos de aprendizaje por revisar', 'Revisar comentarios', and 'Historial de revisiones'.
- Administración:** A section with links for 'Usuario', 'Objetos de aprendizaje', and 'Configuración'.
- Mis publicaciones:** The main content area with a search bar, dropdown menus for 'Categoría' and 'Subcategoría', and a 'Buscar' button. Below this is a red circle highlighting the link 'Suscribirse a mis publicaciones.' with an RSS icon.
- Publicación 'matematica':** A highlighted entry with the following details:
 - Autor(es): admin admin (yasmany)
 - Categoría/Subcategoría: dddd/Nueva Subcategoría
 - Fecha de publicación: 09/03/2011
 - Calificación: No se ha evaluado.
 - Palabras claves: matematica
- Ayuda:** A section with links for 'Preguntas frecuentes', '¿Cómo usar el repositorio?', 'Mapa de navegación', 'Acceso por teclado', and 'Contáctenos'.
- Objetos de Aprendizaje:** A section with links for 'Más recientes' and 'Más solicitados'.
- Estadísticas:** A section showing 'Usuarios registrados: 13', 'Objetos de Aprendizaje: Publicados: 1, En edición: 4, En revisión: 0', and a 'Más esta' link.
- Categorías:** A section with links for 'Anónimo', 'aaaa jhnnn', and 'dddd'.

Figura 4 Suscribirse a Mis publicaciones.

Luego de ejecutar esa acción se brinda la posibilidad de suscribirse y de observar las publicaciones con las que cuenta hasta el momento como lo muestra la siguiente figura:

CAPÍTULO 2: Descripción de la solución propuesta.

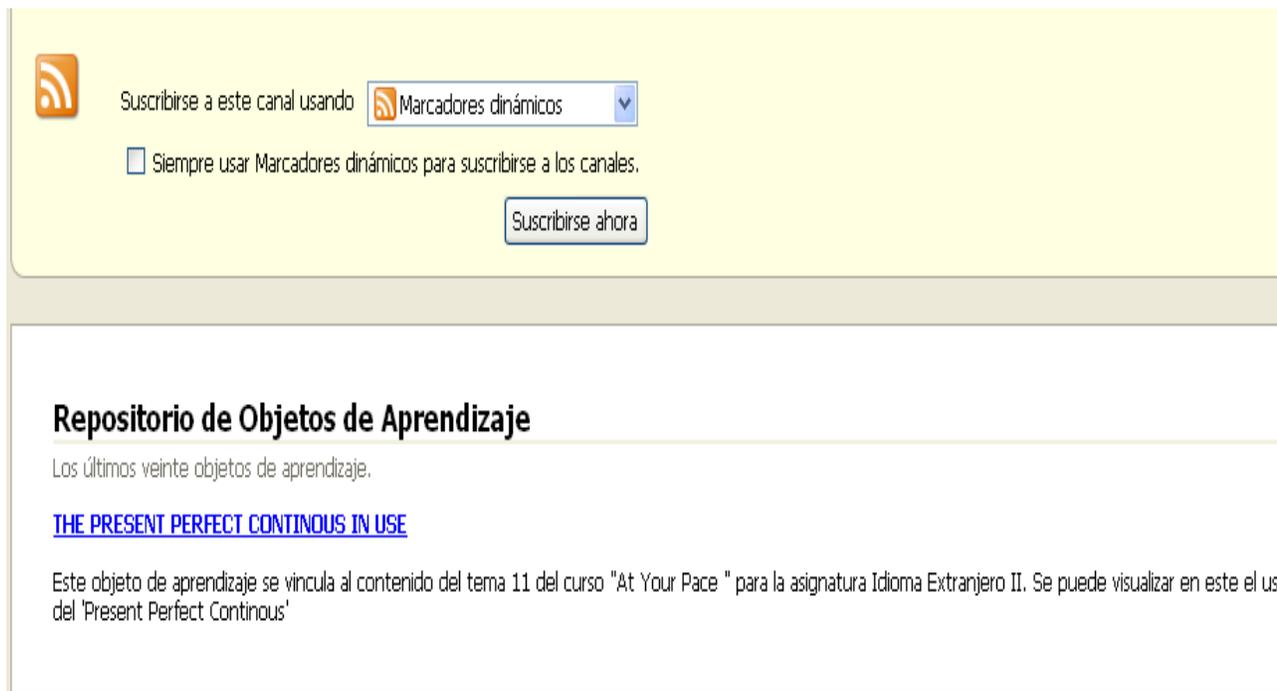


Figura 5 Canal RSS de Mis publicaciones.

Suscribirse a mis favoritos

Brinda la posibilidad a los usuarios de mantenerse actualizados de los objetos de aprendizaje que tiene incluidos en esta categoría. Estos pueden suscribirse dando clic como se muestra en la figura siguiente:

CAPÍTULO 2: Descripción de la solución propuesta.

Figura 6. Suscribirse a Mis favoritos

Luego de ejecutar esa acción se brinda la posibilidad de suscribirse y de observar los objetos de aprendizaje que tiene como favoritos hasta el momento como lo muestra la siguiente figura:

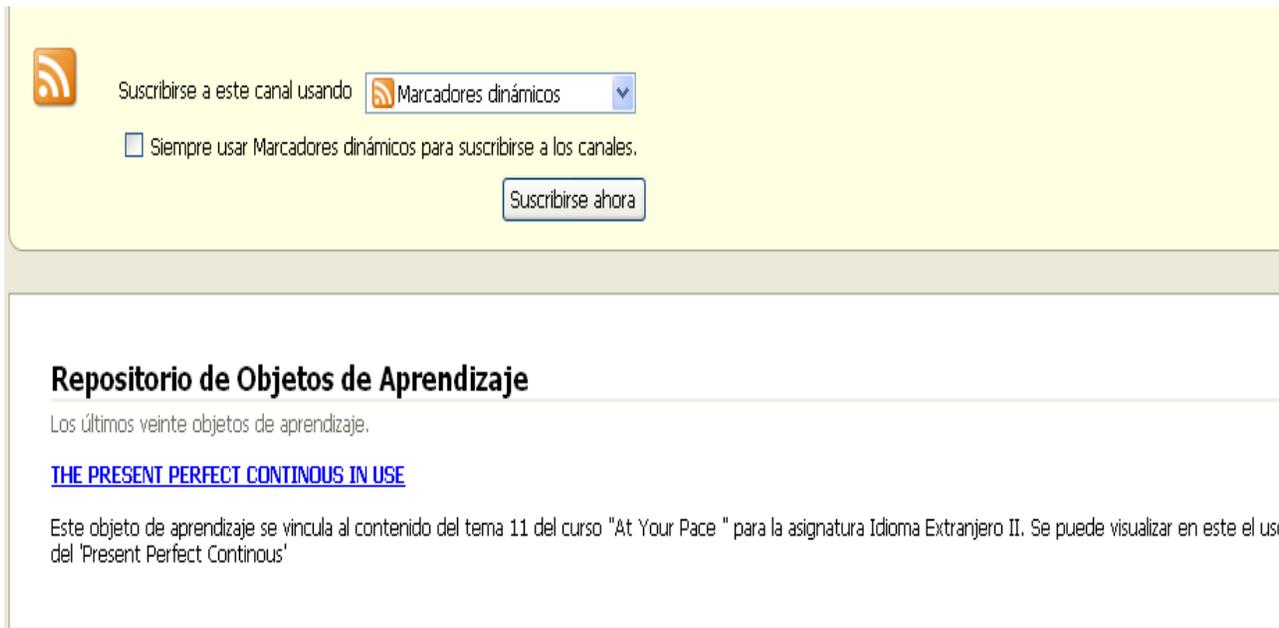


Figura 7. Canal RSS de Favoritos

Suscribirse a objetos de aprendizaje en edición

Brinda la posibilidad a los usuarios de mantenerse actualizados de los objetos de aprendizaje que se encuentran en edición. Estos pueden suscribirse dando clic donde se muestra en la figura siguiente:

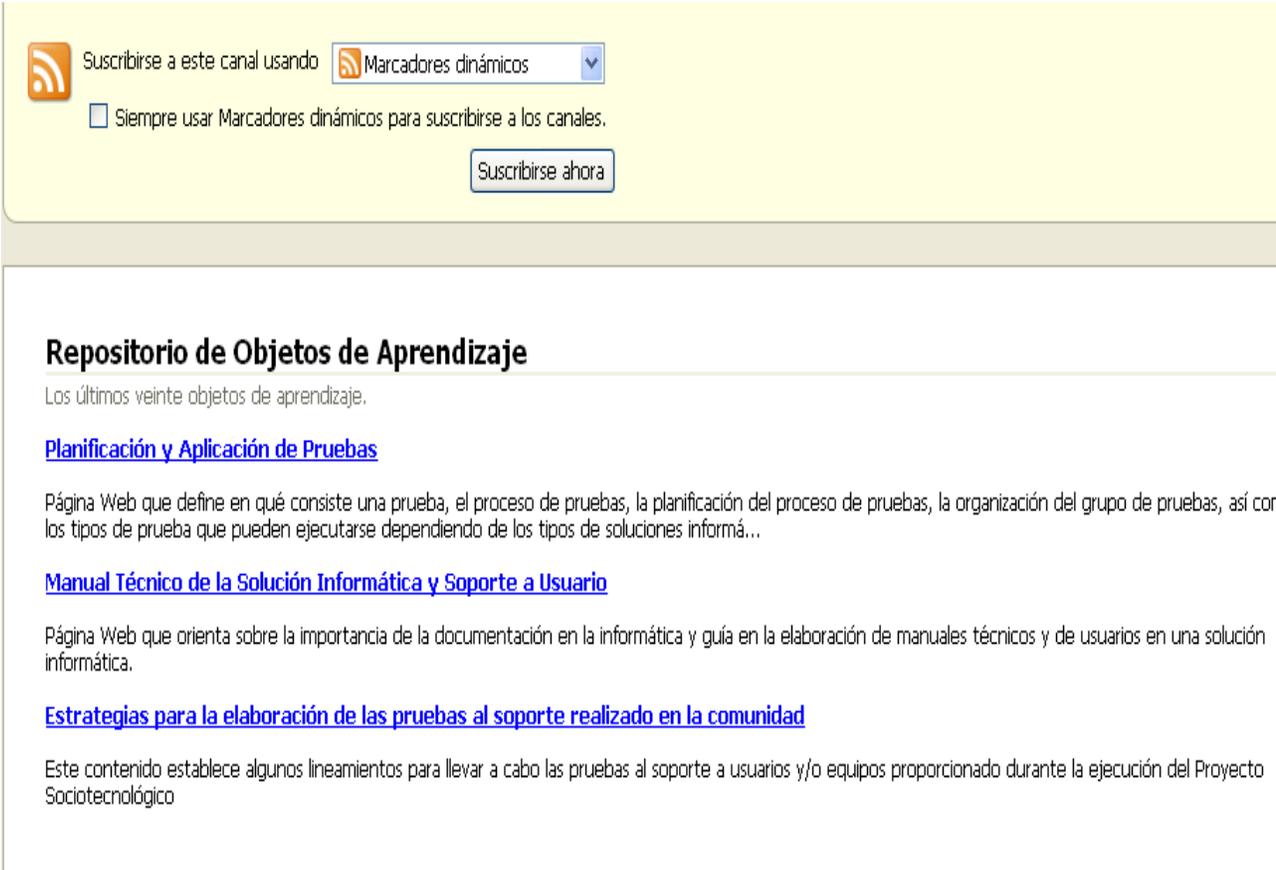
CAPÍTULO 2: Descripción de la solución propuesta.

The screenshot shows a web application interface for learning objects. The top navigation bar includes 'Objetos de aprendizaje en', version '1.3.8', and various utility links like 'config', 'View Layer view', 'Log logs', '16896.0 KB', and '796 ms'. A search bar is located at the top left, and a 'Búsqueda avanzada' link is below it. The main content area features several filters: 'Titulo, Descripción o Palabra clave:', 'Categoría:' (dropdown), 'Subcategoría:' (dropdown), and date filters for 'Fecha de modificación' and 'Fecha de creación', each with 'Desde:' and 'Hasta:' date pickers. A 'Buscar' button is at the bottom of the filter section. A yellow banner with an RSS icon and the text 'Suscribirse a objetos de aprendizaje en edición.' is positioned above the object list. The object list shows 'Objeto5' highlighted, with details: 'Autor(es): admin admin (admin)', 'Categoría/Subcategoría: Anónimo', 'Fecha de creación: 07/03/2011', 'Fecha de última modificación: 07/03/2011', and 'Palabras claves: asd ddd dd'. The right sidebar contains 'Ayuda' links, 'Objetos de Aprendizaje' filters, 'Estadísticas' (13 registered users, 1 published, 4 in edit, 0 in review), and 'Categorías' (Anónimo, aaaa jnnnn, drrrr).

Figura 8. Suscribirse a objetos de aprendizaje en edición.

Luego de ejecutar esa acción se brinda la posibilidad de suscribirse y de observar los objetos de aprendizaje que se encuentran en edición hasta el momento como lo muestra la siguiente figura:

CAPÍTULO 2: Descripción de la solución propuesta.



Suscribirse a este canal usando Siempre usar Marcadores dinámicos para suscribirse a los canales.

Repositorio de Objetos de Aprendizaje

Los últimos veinte objetos de aprendizaje.

[Planificación y Aplicación de Pruebas](#)
Página Web que define en qué consiste una prueba, el proceso de pruebas, la planificación del proceso de pruebas, la organización del grupo de pruebas, así como los tipos de prueba que pueden ejecutarse dependiendo de los tipos de soluciones informá...

[Manual Técnico de la Solución Informática y Soporte a Usuario](#)
Página Web que orienta sobre la importancia de la documentación en la informática y guía en la elaboración de manuales técnicos y de usuarios en una solución informática.

[Estrategias para la elaboración de las pruebas al soporte realizado en la comunidad](#)
Este contenido establece algunos lineamientos para llevar a cabo las pruebas al soporte a usuarios y/o equipos proporcionado durante la ejecución del Proyecto Sociotecnológico

Figura 9. Canal RSS de OA en edición

Suscribirse a objetos de aprendizaje en revisión

Brinda la posibilidad a los usuarios de mantenerse actualizados de los objetos de aprendizaje que se encuentran en revisión. Estos pueden suscribirse dando clic como se muestra en la figura siguiente:

CAPÍTULO 2: Descripción de la solución propuesta.

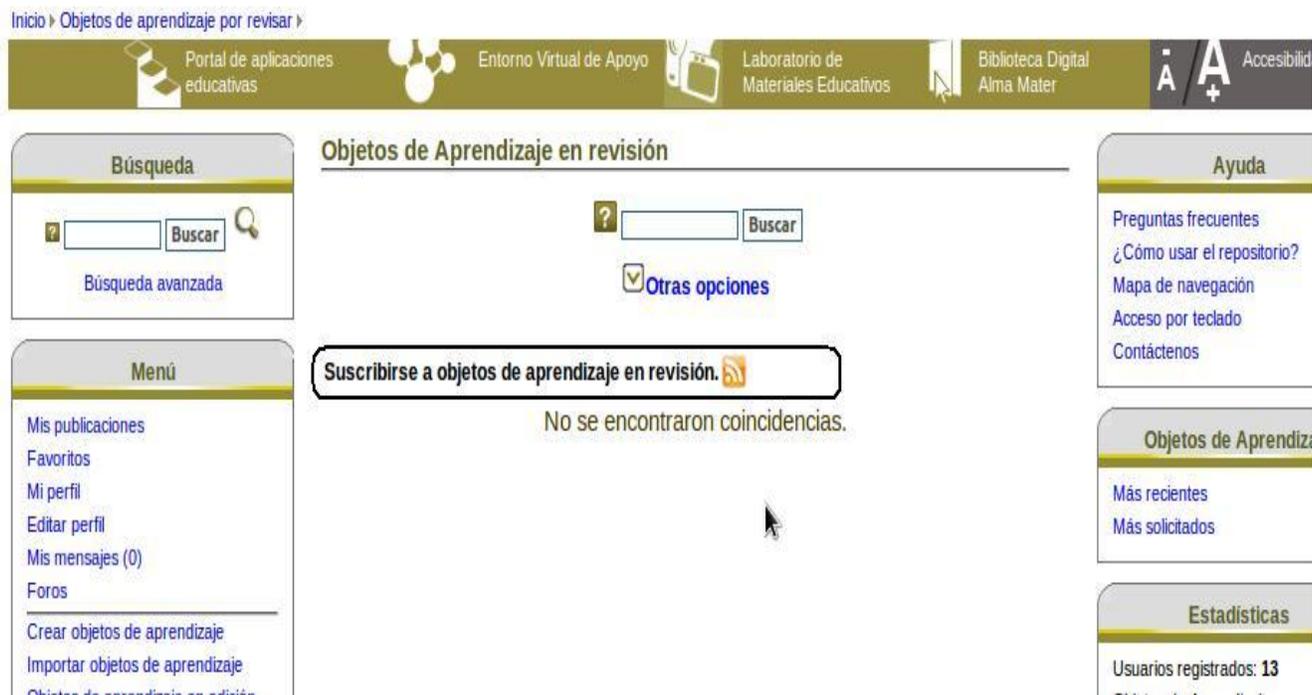


Figura 10. Suscribirse a objetos de aprendizaje en revisión.

Luego de ejecutar esa acción se brinda la posibilidad de suscribirse y de observar los objetos de aprendizaje que se encuentran en revisión hasta el momento como lo muestra la siguiente figura:



Figura 11. Canal RSS de OA en revisión

CAPÍTULO 2: Descripción de la solución propuesta.

Elegir notificaciones y el tiempo de envío de las notificaciones.

En este módulo también se tuvieron presente las notificaciones al correo electrónico personal de los usuarios debido a que es una aplicación que revisan diariamente y así se mantienen informados día a día, para esto se obtuvieron las preferencias editadas en el perfil de cada uno de los usuarios, donde tienen la posibilidad de seleccionar aquellas notificaciones que desean recibir en sus correos como se muestra en la siguiente figura:

Editar mi perfil

[Información básica](#) [Perfil académico](#) [Preferencias](#) [Solicitudes](#)

Preferencias:

Página de inicio:

Búsquedas ?

Notificaciones periódicas:

- Objetos de aprendizaje más recientes.
- Objetos de aprendizaje más solicitados.

Frecuencia de recibo de notificaciones:

Diariamente ?

Notificaciones instantáneas:

- Cuando se realicen comentarios a un objetos de aprendizaje de su autoría.
- Cuando sea rechazado un objeto de aprendizaje de su autoría.
- Cuando sea aceptado un objeto de aprendizaje de su autoría.
- Cuando sea publicado un objeto de aprendizaje de su autoría.
- Cuando sea regresado a edición un comentario de su autoría.
- Cuando sea eliminado un comentario de su autoría.
- Cuando sea publicado un comentario de su autoría.
- Cuando algún usuario descargue su objeto de aprendizaje.
- Cuando algún usuario visualice su objeto de aprendizaje.
- Cuando sea aceptada una solicitud.
- Cuando sea rechazada una solicitud.
- Cuando reciba una solicitud.

Figura 12 Preferencias

Luego de seleccionar cada una de las notificaciones que desean recibir, estos valores se guardan en la base de datos en la tabla correspondiente y cuando son efectuados estos eventos en el repositorio se verifica si el usuario seleccionó dicha opción y se envía el correo relacionado con dicho evento.

CAPÍTULO 2: Descripción de la solución propuesta.

Informar sobre comentarios a objetos de aprendizaje.

Por ejemplo, cuando se realizan comentarios a un objeto de aprendizaje, el autor de dicho objeto recibe una notificación indicándole el nombre del objeto de aprendizaje y el contenido del comentario realizado como se muestra en la figura siguiente:

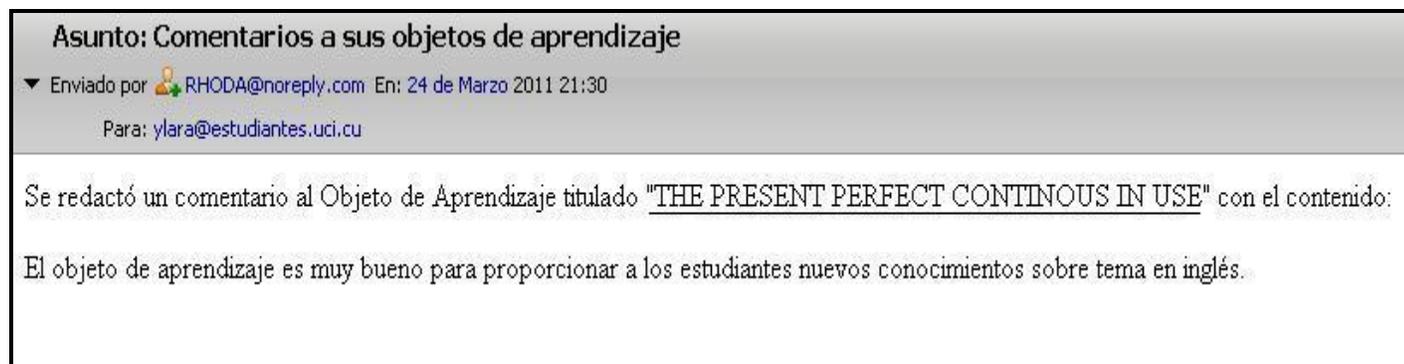


Figura 13. Ejemplo de correo sobre comentarios a sus OA

Publicar y Denegar comentario.

Los diferentes usuarios pueden realizar infinidad de comentarios a los objetos de aprendizaje que deseen, pero esto son enviados a revisión, de ahí que se brinda la posibilidad de recibir una notificación por correo electrónico cuando dicho comentario sea denegado o cuando sea publicado como se muestran en las figuras siguientes:

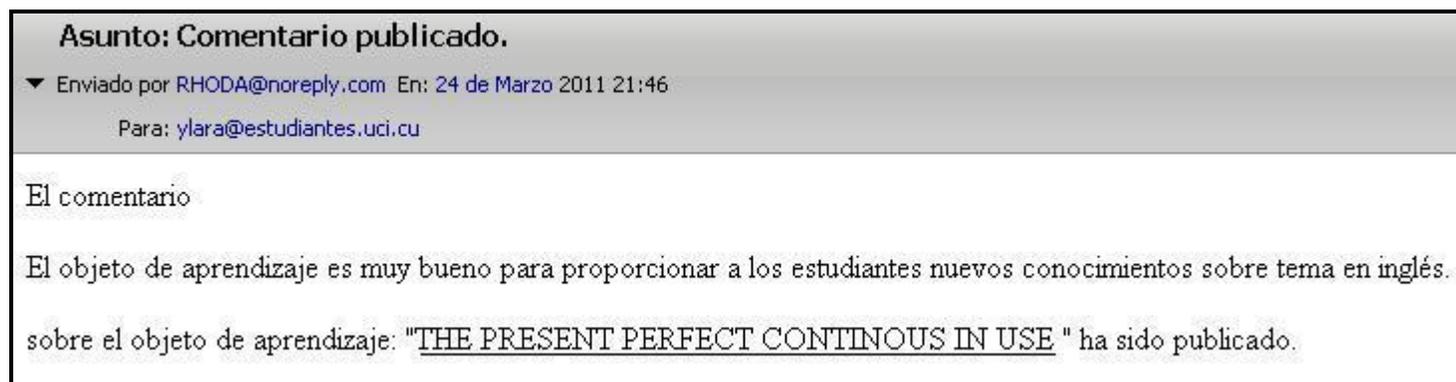


Figura 14. Ejemplo de correo cuando el comentario ha sido publicado

CAPÍTULO 2: Descripción de la solución propuesta.



Figura 15. Ejemplo de correo cuando el comentario ha sido denegado

Informar cuando sea aceptado un objeto de aprendizaje.

Los usuarios del repositorio también pueden crear los objetos de aprendizaje que deseen, pero estos son enviados primeramente a edición y luego pasan a la parte de revisión desde donde se acepta o rechaza, de ahí que se brinda la posibilidad de recibir una notificación por correo electrónico cuando sea aceptado, como se muestra en la figura siguiente:

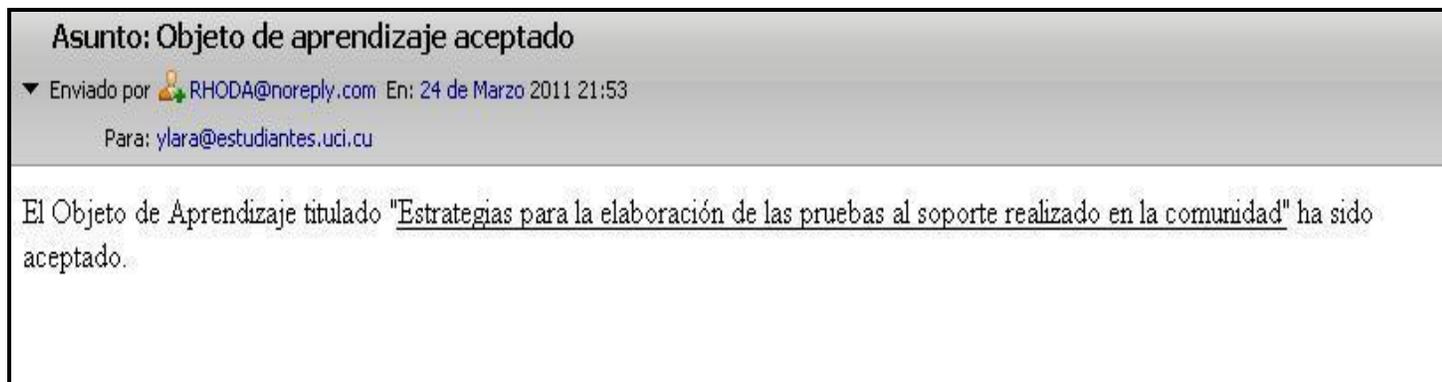


Figura 16. Ejemplo de correo cuando un OA ha sido aceptado

Informar cuando sea rechazado un objeto de aprendizaje.

Los usuarios del repositorio cuando crean los objetos de aprendizaje que desean, estos son enviados primeramente a edición y desde ahí puede ser rechazado directamente el objeto de

CAPÍTULO 2: Descripción de la solución propuesta.

aprendizaje, de ahí que se brinda la posibilidad de recibir una notificación por correo electrónico cuando esto suceda, como se muestra en la figura siguiente:

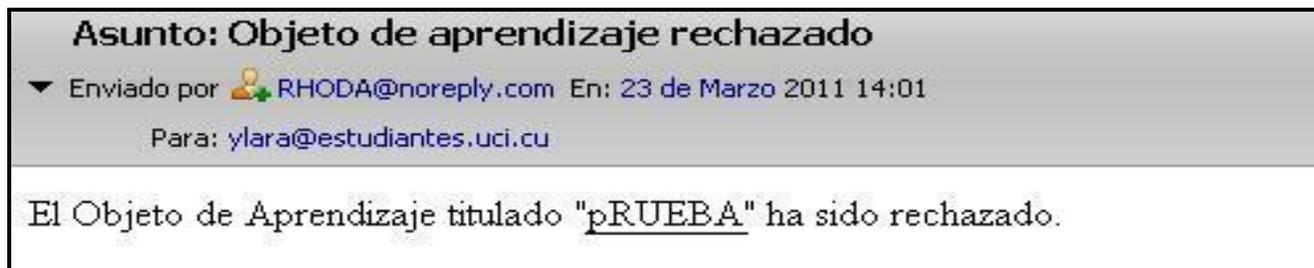


Figura 17. Ejemplo de correo cuando un OA ha sido rechazado

Descargar objeto de aprendizaje para ser reutilizado y Visualizar objeto de aprendizaje.

Además todos los usuarios del repositorio cuentan con la opción de visualizar y descargar los objetos de aprendizaje de otros autores, razón por la cual se brinda la posibilidad de recibir una notificación por correo electrónico cuando algún usuario visualice o descargue un objeto de aprendizaje, como se muestran en las figuras siguientes:



Figura 18. Ejemplo de correo cuando un OA ha sido descargado

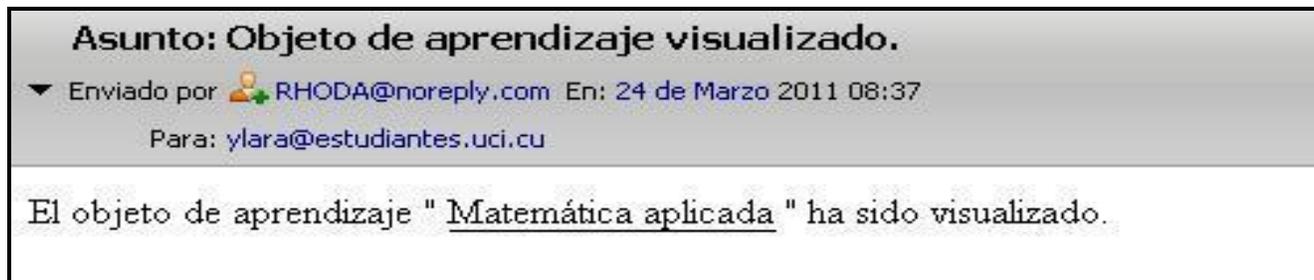


Figura 19. Ejemplo de correo cuando un OA ha sido visualizado

CAPÍTULO 2: Descripción de la solución propuesta.

Informar cuando aceptada o rechazada la solicitud.

También los usuarios tienen la opción de enviar solicitudes al administrador para realizar un determinado cambio, por lo cual se le brinda la posibilidad de recibir una notificación a su correo electrónico cuando dichas solicitudes sean aceptadas o rechazadas como se muestran en las figuras siguientes:

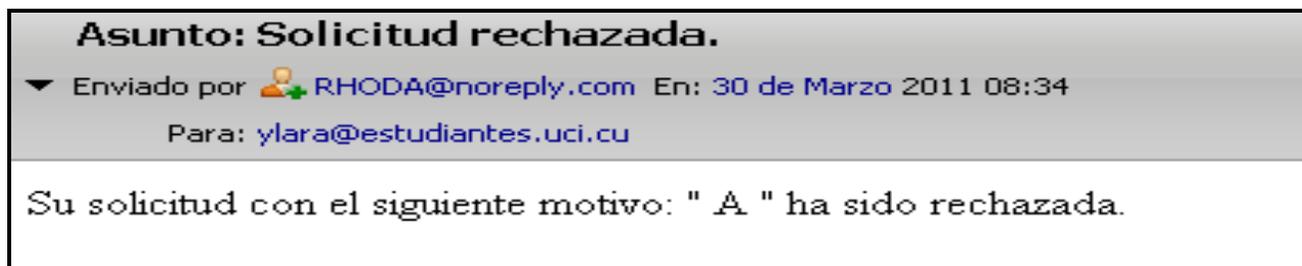


Figura 20 Ejemplo de correo cuando una solicitud es rechazada.

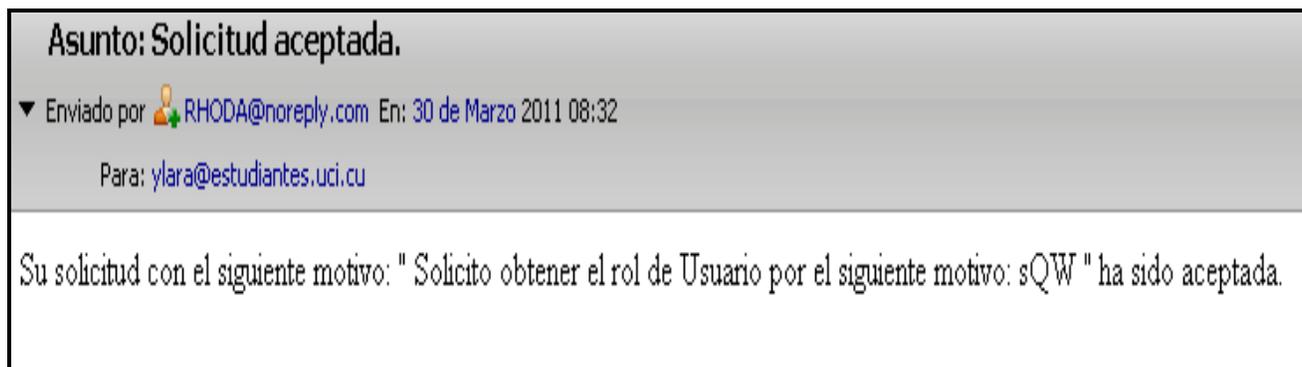


Figura 21 Ejemplo de correo cuando una solicitud es aceptada.

Informar sobre envío de solicitud.

A su vez a los usuarios que cuentan con el rol de Administrador se les brinda la posibilidad de recibir una notificación a su correo electrónico cuando reciba alguna solicitud y así estar al tanto de los cambios que desean realizar los diferentes usuarios y poder darles una respuesta en el menor tiempo posible, como se muestra en la figura siguiente:

CAPÍTULO 2: Descripción de la solución propuesta.



Figura 22 Ejemplo de correo cuando se recibe una nueva solicitud.

Informar cuando cambie de categoría un OA.

El administrador del repositorio puede cambiar a un objeto de aprendizaje determinado para una categoría específica, razón por la cual los usuarios cuentan con la opción de recibir un correo cuando esto ocurra, como se muestra en la figura siguiente:

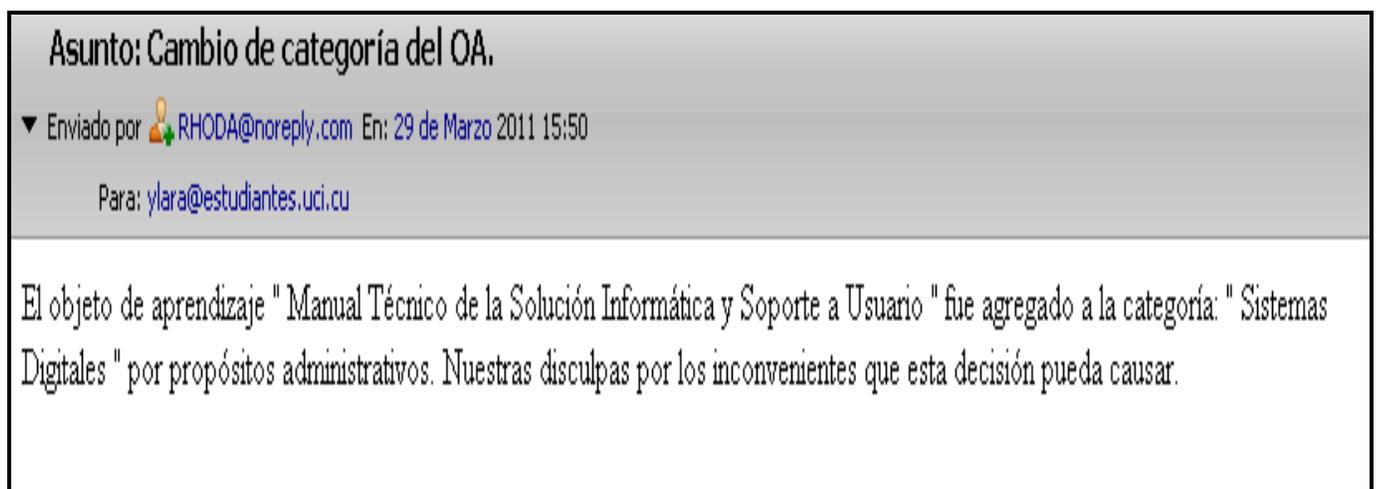


Figura 23 Ejemplo de correo cuando cambie de categoría un OA.

Informar los objetos de aprendizaje más recientes y más solicitados.

Los usuarios cuentan además con las opciones de conocer los objetos de aprendizaje más recientes y los más solicitados, por lo cual se les brinda la posibilidad de recibir una notificación a sus correos electrónicos cada un tiempo especificado por ellos mismos para informales los

CAPÍTULO 2: Descripción de la solución propuesta.

objetos de aprendizaje que se encuentran en esas categorías y así mantenerse informados en este aspecto sin necesidad de visitar el repositorio, como se muestra en la figura siguiente:



Figura 24 Ejemplo de correo con los OA más solicitados.

Las últimas notificaciones mencionadas son controladas por una tarea programada que a su vez utiliza un programa llamado **Cron**, el cual será explicado en un acápite posteriormente, y esta utiliza en su implementación la frecuencia de envío especificada por el usuario.

2.3.2 Módulo Usuario

Este módulo es el encargado de gestionar toda la información relacionada con los usuarios del repositorio. Durante el estudio del mismo se detectó que existen ciertas informaciones que el usuario debe conocer en el momento que desee, por lo cual se implementaron las siguientes funcionalidades.

Informar cuando se cree una cuenta de usuario.

Los usuarios tienen la posibilidad de recibir un correo cuando se cree una cuenta en el repositorio con su nombre y dirección de correo electrónico, informándoles el usuario y la contraseña con la que pueden acceder al mismo a partir de ese momento, como se muestra en la figura siguiente:

CAPÍTULO 2: Descripción de la solución propuesta.

De: RHODA@noreply.com
Para: ypacho@estudiantes.uci.cu
Enviados: Miércoles, 30 de Marzo 2011 15:19:41 GMT -04:00 Georgetown
Asunto: Creación de cuenta en el Repositorio RHODÀ.

Se ha creado una cuenta bajo su nombre y dirección de email en el Repositorio de objetos de aprendizaje RHODÀ. Desde estos momentos puede autenticarse en el mismo con el usuario:yeni y contraseña:yeniEsperamos que disfrute de nuestros servicios.

Figura 25 Ejemplo de correo cuando una cuenta es creada.

Informar cuando sea bloqueada o desbloqueada la cuenta de usuario.

Además tienen la posibilidad de recibir un correo cuando su cuenta en el repositorio sea bloqueada o desbloqueada por razones administrativas, como se muestran en las figuras siguientes:

Asunto: Bloqueada su cuenta de usuario.

▼ Enviado por  RHODA@noreply.com En: 28 de Marzo 2011 13:48

Para: ylara@estudiantes.uci.cu

Estimado usuario, el administrador del sistema ha bloqueado su cuenta de usuario. Pedimos disculpas por los inconvenientes ocasionados.

Figura 26 Ejemplo de correo cuando una cuenta es bloqueada.

CAPÍTULO 2: Descripción de la solución propuesta.

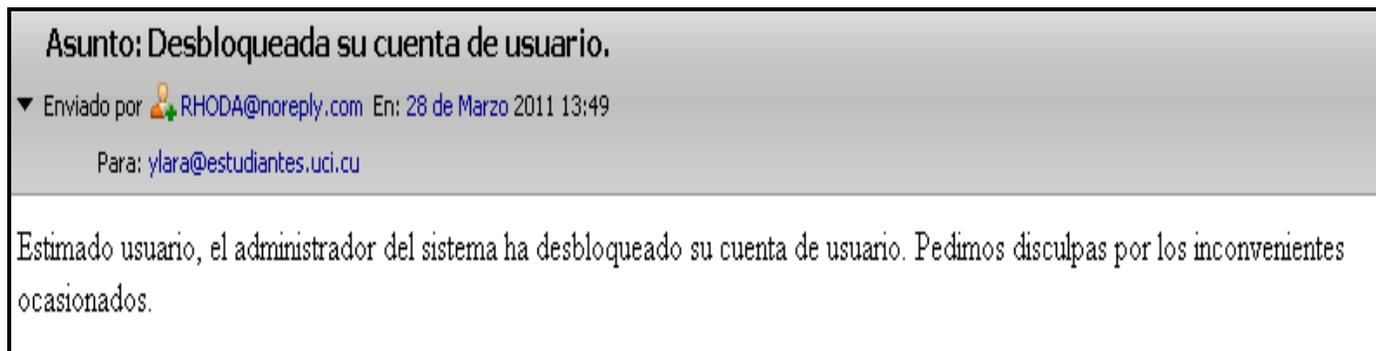


Figura 27 Ejemplo de correo cuando una cuenta es desbloqueada.

Informar cuando sea modificado el rol.

Los usuarios cuentan con la opción de recibir un correo cuando es modificado el rol que desempeñaba en el repositorio por razones administrativas, y así tener conocimiento del rol que posee, como se muestra en la figura siguiente:

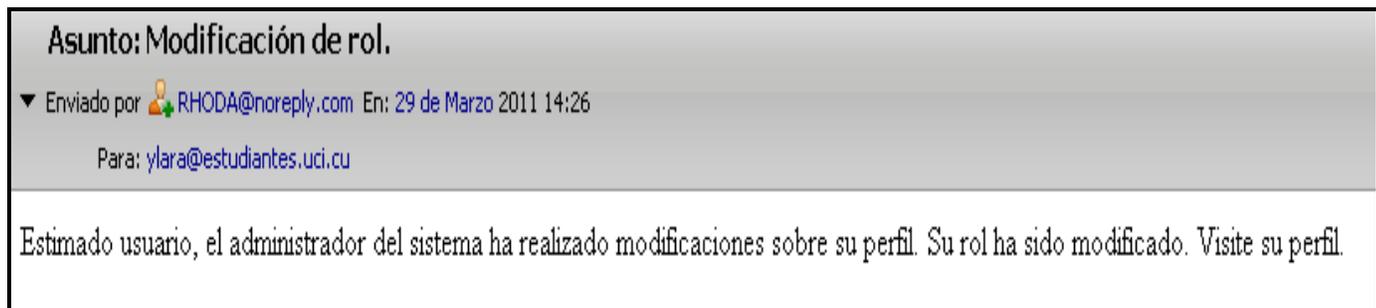


Figura 28 Ejemplo de correo cuando un rol es modificado.

Informar cuando sea añadido a una categoría con un rol determinado.

Los usuarios también tienen la posibilidad de recibir un correo cuando sea añadido a una determinada categoría y con un rol específico, el cual puede ser Autor o Revisor, como se muestran en las figuras siguientes:

CAPÍTULO 2: Descripción de la solución propuesta.

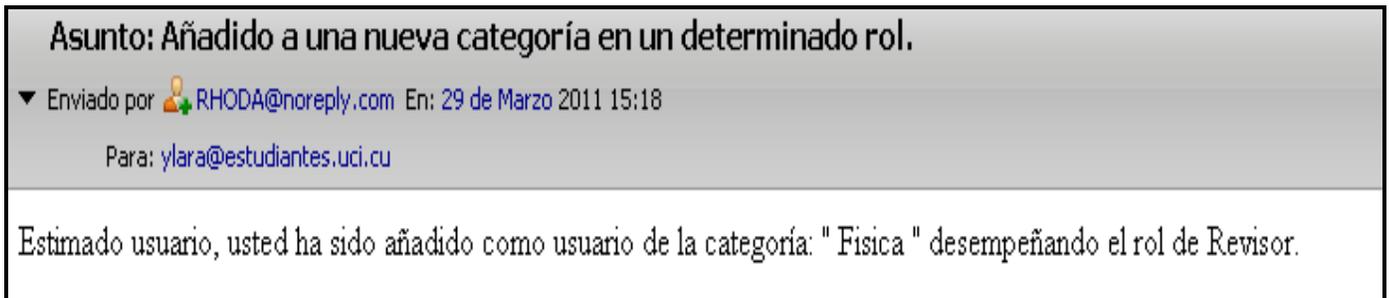


Figura 29 Ejemplo de correo cuando es añadido a una categoría en el rol de Revisor.

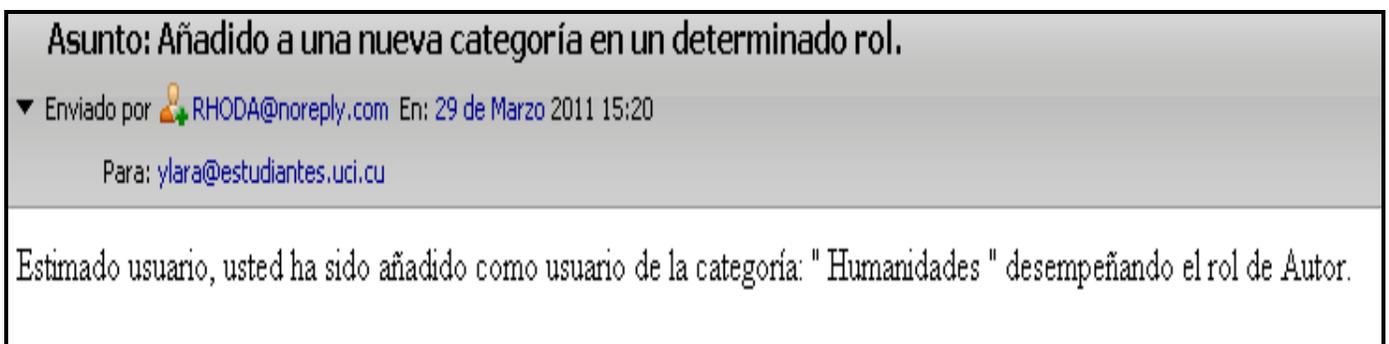


Figura 30 Ejemplo de correo cuando es añadido a una categoría en el rol de Autor.

Informar cuando sea eliminado su usuario del rol que ocupaba en la categoría.

Además estos usuarios cuentan con la opción de recibir un correo cuando sea eliminado del rol que desempeñaba en una categoría, como se muestra en la figura siguiente:

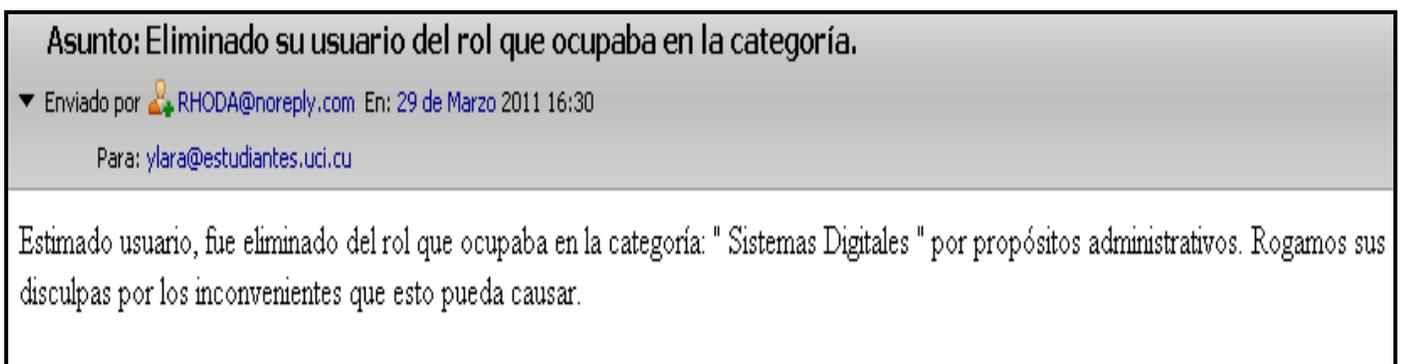


Figura 31 Ejemplo de correo cuando es eliminado del rol.

CAPÍTULO 2: Descripción de la solución propuesta.

2.3.3 Módulo Administración

Este módulo es el responsable de las configuraciones del sistema, permite cambiar todo lo que puede ser configurable. Durante el estudio del mismo se comprobó que era muy importante para los usuarios tener conocimiento cuando fuera eliminada una categoría o subcategoría y además cuando una categoría fuera renombrada, razón por la cual se implementaron las siguientes funcionalidades:

Informar cuando sea eliminada una categoría.

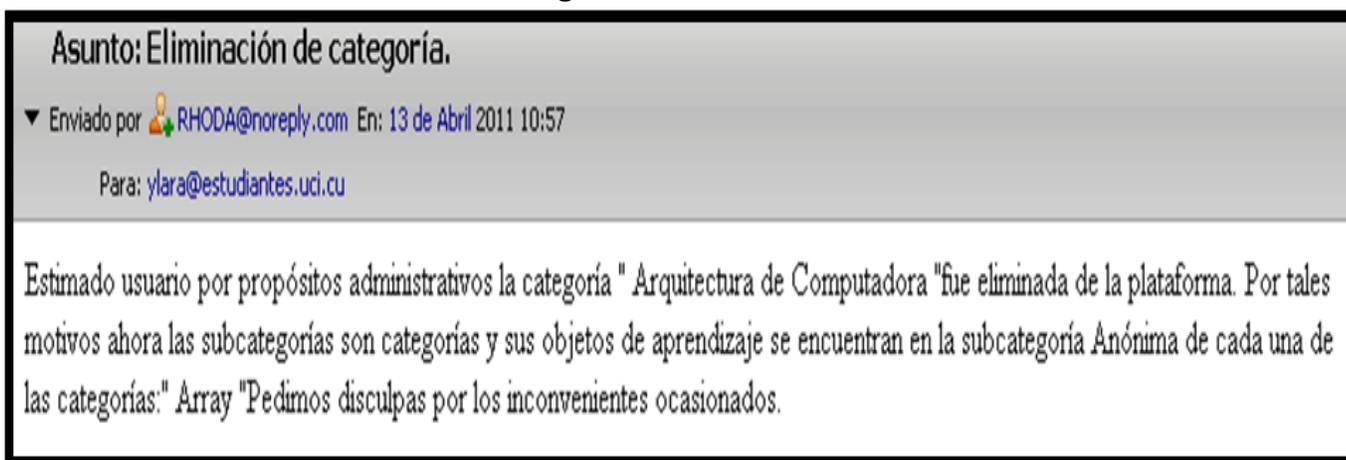


Figura 32 Ejemplo de correo cuando es eliminada una categoría.

Informar cuando sea eliminada una subcategoría.

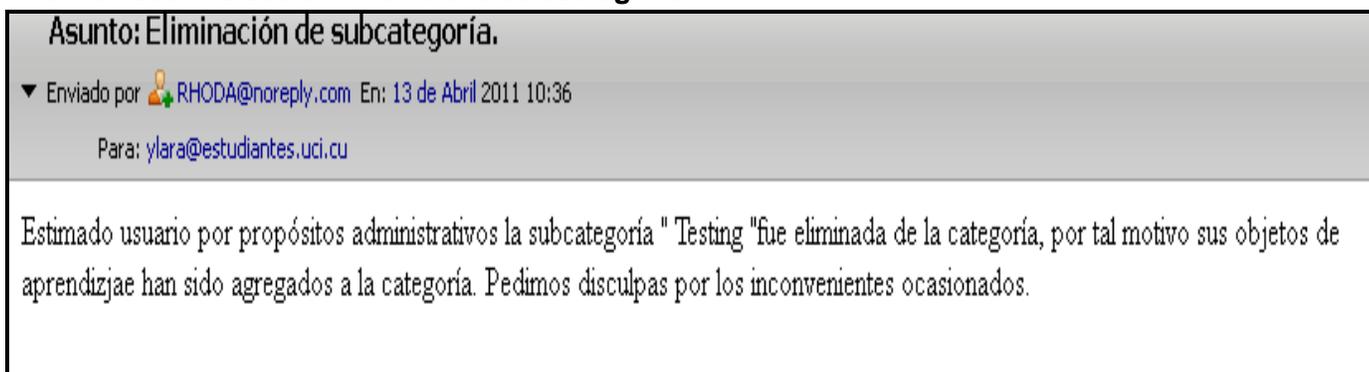


Figura 33 Ejemplo de correo cuando es eliminada una subcategoría.

CAPÍTULO 2: Descripción de la solución propuesta.

Informar cuando se modifique el nombre de la categoría.

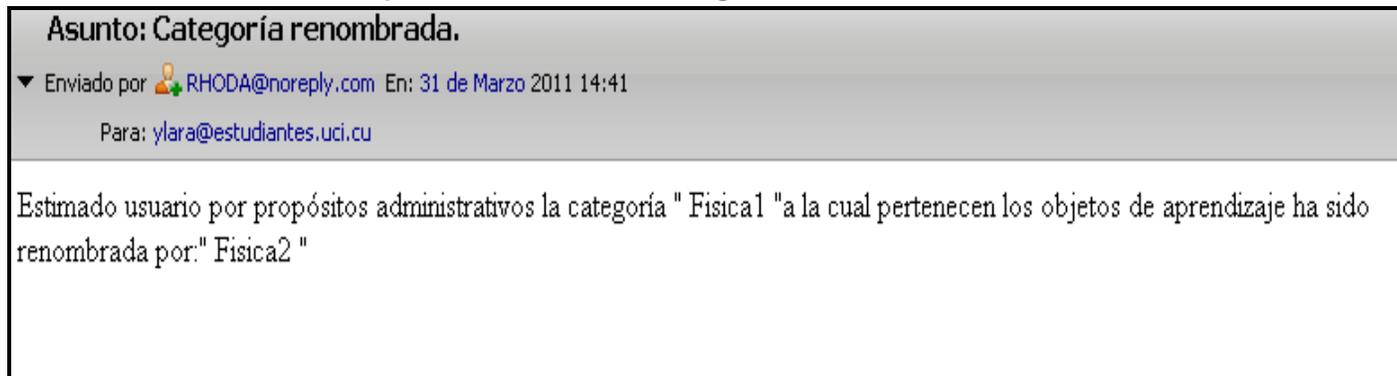


Figura 34 Ejemplo de correo cuando es renombrada una categoría.

2.4 Requisitos no funcionales.

Un **requisito no funcional** es, en la ingeniería de sistemas y la ingeniería de software, un requisito que especifica criterios que pueden usarse para juzgar la operación de un sistema en lugar de sus comportamientos específicos, hacen relación a las características del sistema que aplican de manera general como un todo, más que a rasgos particulares del mismo. Estos requerimientos son adicionales a los requerimientos funcionales que debe cumplir el sistema.

Requerimiento 1: Todas las imágenes (banners, íconos, etc.) y botones (Enviar, Aceptar, Cancelar, etc.) del sistema, al pasar el cursor por encima, mostrarán un texto sugerente.

Requerimiento 2: En los formularios que contengan campos requeridos (obligatorios), estos se especificarán mediante un asterisco y, de no ser vistos, se indicarán con un texto en rojo, manteniendo el resto de la información que se introdujo correctamente.

Requerimiento 3: Las estadísticas se mostrarán en varios tipos de gráficos. Se podrá cambiar el tamaño de los mismos.

Requerimiento 4: En las páginas en que se pueda aplicar, se mostrarán sólo los bloques que contengan información de interés para esa página.

CAPÍTULO 2: Descripción de la solución propuesta.

Requerimiento 5: A cada página, asociada a las nuevas funcionalidades a incluir, se le agregará un título, mostrando la sección correspondiente.

Requerimiento 6: Verificar que todos los enlaces posean un texto breve y sugerente.

Requerimiento 7: Actualización de los rastros creados por el sistema.

Requerimiento 8: El sistema chequeará todos los campos de entrada de datos, con el objetivo de garantizar que no se introduzcan datos erróneos, en cuyo caso se muestra al usuario dónde ocurrió, cuál fue el motivo y cómo solucionarlo.

Requerimiento 9: Se agregarán etiquetas a cada campo con una descripción breve de cuál es el tipo de dato correcto a introducir. Consecuentemente, se actualizará la Arquitectura de la Información del sistema.

2.5 Estándares de codificación

Con el objetivo de lograr que el equipo de desarrollo comprenda el código de los demás programadores se hace necesario el establecimiento de un estándar de codificación, propiciando así un producto final con mayor calidad. También existen ciertas reglas definidas por el framework, relacionada con la nomenclatura que hay que respetar para lograr un correcto funcionamiento que tributan al estándar de codificación a utilizar.

Para el desarrollo del Repositorio de Objetos de Aprendizaje se definieron un conjunto de pautas a seguir en cuanto a la forma de escribir código fuente, teniendo en cuenta los siguientes aspectos:

Idioma

Para nombrar variables, funciones, clases y demás se usarán nombres descriptivos y en idioma inglés.

Identificadores

CAPÍTULO 2: Descripción de la solución propuesta.

Los identificadores se comienzan a escribir con minúscula y en el caso de componerse de más de una palabra se escriben una a continuación de la otra, comenzando con mayúscula a partir de la segunda.

Como se puede apreciar en el fragmento de código que describe a continuación:

```
class Item{
private $str ;
private $icon ;
private $actionPerform ;
}
```

Documentación del código fuente

Una de las mejores prácticas para generar la documentación de los módulos es que esté autocontenida en los mismos, para ello en este trabajo, debido a que se utiliza PHP como lenguaje de programación, se utilizó el estándar phpDoc, debido a que la mayoría de los Entornos de Desarrollo Integrado que dan soporte a PHP lo interpretan y utilizan para ayudar al programador, además de que ofrece la gran ventaja de poder generar páginas HTML con documentación del código fuente.

```
/**
 * This method return an object with the SCORM schema for
 * the XPath passed by parameter
 *
 * @param string $path
 * @return stdClass
 */
private static function getItems( $path ){
```

Identación.

CAPÍTULO 2: Descripción de la solución propuesta.

Para indentar cualquier fragmento de código se utilizarán espacios en blanco y nunca tabulaciones, las estructuras anidadas usarán 4 espacios para la indentación, además los lexemas que se usan para comenzar y cerrar bloques “{” y “}” no poseerán indentación alguna.

```
public function preExecute()  
  
    {  
  
        $id=$this->getRequestParameter('id');  
  
        if ($id == 1)  
  
            {  
  
                $this->redirect404();  
  
            }  
  
    }
```

Comentarios

Para facilitar la comprensión del código se utilizan los comentarios en línea, los cuales pueden ser con fin documental o bien como ayuda. Hay dos tipos de comentarios: Comentarios de implementación y comentarios de documentación. Los comentarios de implementación son del tipo: // ... o /* ... */. Los comentarios de documentación son del tipo: /** ... */.

2.6 Tareas programadas y las notificaciones

Para la realización de las notificaciones es necesaria la utilización de algún mecanismo para controlar el horario en que serán efectuadas las mismas, convirtiéndose en tareas programadas de la aplicación. Uno de los mecanismos más usados actualmente es el Cron.

Cron es el nombre del programa que permite a usuarios Linux/Unix ejecutar automáticamente comandos o scripts (grupos de comandos) a una hora o fecha específica. Es usado normalmente para comandos de tareas administrativas, como respaldos, pero puede ser usado

CAPÍTULO 2: Descripción de la solución propuesta.

para ejecutar cualquier cosa. Como se define en las páginas del manual de cron es un demonio que ejecuta programas agendados.

En prácticamente todas las distribuciones de Linux se usa la versión Vixie Cron, la cual debe su nombre a su autor Paul Vixie.

Este programa cuenta con siete campos principales los cuales forman las líneas que lo componen de la siguiente manera:

Campo	Descripción
Minuto	Controla el minuto de la hora en que el comando será ejecutado, este valor debe de estar entre 0 y 59.
Hora	Controla la hora en que el comando será ejecutado, se especifica en un formato de 24 horas, los valores deben estar entre 0 y 23, 0 es medianoche.
Día del Mes	Día del mes en que se quiere ejecutar el comando. Por ejemplo se indicaría 20, para ejecutar el comando el día 20 del mes.
Mes	Mes en que el comando se ejecutará, puede ser indicado numéricamente (1-12), o por el nombre del mes en inglés, solo las tres primeras letras.
Día de la semana	Día en la semana en que se ejecutará el comando, puede ser numérico (0-7) o por el nombre del día en inglés, solo las tres primeras letras. (0 y 7 = domingo)
Usuario	Usuario que ejecuta el comando.
Comando	Comando, script o programa que se desea ejecutar. Este campo puede contener múltiples palabras y espacios.

Figura 35 Explicación de los campos del Cron.

Así un asterisco * como valor en los primeros cinco campos, indicará inicio-fin del campo, es decir todo. Un * en el campo de minuto indicará todos los minutos y así sucesivamente.

Para la implementación de este sistema se creó una tarea como un fichero PHP llamada **RoanotifyTask** la cual será la encargada de ejecutar cada uno de los métodos especificados y el Cron se programó como se muestra en la siguiente figura:

```
# m h dom mon dow  command
* 2 * * * cd home/yani/Escritorio/Roa/ && /usr/bin/php5 ./symfony Roa:notify
```

Figura 36 Ejemplo del fichero de Cron.

CAPÍTULO 2: Descripción de la solución propuesta.

Después de realizar una profunda investigación se puede concluir que durante el desarrollo de este capítulo se definió la arquitectura del Repositorio de Objetos de Aprendizaje (RHODA) y se explicaron los modelos de datos que utiliza para guardar toda la información referente a los objetos de aprendizaje y al trabajo que se realiza con ellos. Además se explicaron las principales funcionalidades con las que cuenta el sistema implementado, mostrando las vistas correspondientes, así como los estándares de codificación y la aplicación de las tareas programadas dentro de las notificaciones.

CAPÍTULO 3: Validación de la solución propuesta.

CAPÍTULO 3: Validación de la solución propuesta

El proceso de pruebas es un instrumento adecuado para determinar el estado de la calidad del producto. En este proceso se ejecutan pruebas dirigidas a los componentes del software, con el objetivo de medir el grado en que se cumplen los requerimientos.

Las pruebas son una actividad en la cual un sistema o componente es ejecutado bajo condiciones o requerimientos especificados, los resultados son observados y registrados, y se hace una evaluación de algún aspecto del sistema o componente. En muchas ocasiones el éxito de un producto de software puede venir dado por la eficacia con la que se realicen sus pruebas, de ahí la importancia de que el proceso sea llevado a cabo con la calidad requerida.

En este capítulo se aborda acerca del método conocido como Caja Negra utilizado en el flujo de trabajo Pruebas de la metodología RUP.

3.1 Pruebas de Caja Negra

Las pruebas de Caja Negra se centran principalmente en los requisitos funcionales del software, se llevan a cabo sobre la interfaz del software. Estas pruebas permiten obtener un conjunto de condiciones de entrada y salida que ejerciten completamente todos los requisitos funcionales de un programa. Por tanto, en estas pruebas deben estar muy bien definidas sus entradas y salidas, es decir, su interfaz; en cambio, no se precisa definir ni conocer los detalles internos de su funcionamiento.

Las pruebas de Caja Negra están especialmente indicadas en aquellos módulos que van a ser interfaz con el usuario como por ejemplo teclado, pantalla, ficheros, canales de comunicaciones, entre otros.

CAPÍTULO 3: Validación de la solución propuesta.

3.1.1 Descripción de los casos de prueba

La descripción de los casos de pruebas establece una guía para el probador, por lo cual deben ser precisos y abarcar todos los posibles errores, ya que de ahí dependerá el éxito de la prueba. Están compuestos por posibles entradas, condiciones de ejecución y las posibles salidas que pueda generar, para lograr un resultado esperado.

Para la validación del sistema se diseña un caso de prueba por cada caso de uso, en este caso las funcionalidades implementadas forman parte de un caso de uso específico, por lo cual se hizo un estudio detallado de las descripciones textuales de cada uno, con el fin de identificar los escenarios de pruebas más significativos como se aprecia en la tabla siguiente.

Tabla 1 CU Gestionar categorías

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo central
1 Gestionar categorías	EC 1.1 Gestionar categorías	Selecciona en la pantalla principal la opción: Administrar categorías.	RHODA inicio/Administrar categorías
2 Agregar categoría	EC 2.1 Agregar categoría	Selecciona la opción: Agregar categoría de la interfaz Administrar categorías, y se introducen el nombre de la categoría, luego se selecciona la opción: Agregar para enviar los datos al sistema	RHODA inicio/Administrar categorías/Agregar categoría
	EC 2.2: Cancela acción	Decide cancelar la agregación de la categoría y selecciona la opción: Cancelar.	RHODA inicio/Administrar categorías/Agregar categoría/Cancelar

CAPÍTULO 3: Validación de la solución propuesta.

3 Editar categoría	EC 3.1 Editar categoría	Selecciona la opción: Renombrar, correspondiente a una categoría. Modifica el nombre mostrado en el campo de texto y selecciona la opción: Aceptar para terminar la acción. Envía notificación a los usuarios cuyos objetos de aprendizaje se encontraban en esa categoría.	RHODA inicio/Administrar categorías/Editar categoría
	EC 3.2: Cancela acción	Decide cancelar la agregación de la categoría y selecciona la opción: Cancelar.	RHODA inicio/Administrar categorías/Agregar categoría/Cancelar

Tabla 2 Matriz de datos

ID del escenario	Escenario	Nombre de la categoría	Respuesta del sistema	Resultado de la prueba
3.1	Editar categoría	V	3.1 Muestra una ventana donde es posible modificar el nombre de la categoría seleccionada.	Satisfactorio
		V(Matemática 1)	-Nombre categoría (Formato: Alfanumérico, Estado: Obligatorio) 3.2 Muestra las opciones de aceptación y cancelación de la acción: Agregar y Cancelar respectivamente. 3.3 Valida el dato entrado por el actor. 3.4 Modifica el nombre anterior de la categoría por el entrado. 3.5 Actualiza la información mostrada al	

CAPÍTULO 3: Validación de la solución propuesta.

			<p>actor y envía notificación a los usuarios cuyos objetos de aprendizaje se encontraban en esa categoría:</p> <p>-Dirigido a: <i>(Autores de los OA de la categoría)</i></p> <p>-Asunto: <i>Categoría renombrada</i></p> <p>-Mensaje: <i>“Estimado usuario, por propósitos administrativos la categoría (Nombre de la categoría antes de renombrar; ha sido renombrada, ahora se llama (Nombre de la categoría después de renombrar). Pedimos disculpas por los inconvenientes que esta decisión pueda ocasionar.”</i></p> <p>3.6 Actualiza en los OA de la categoría renombrada el valor del campo: Categoría en el archivo de metadatos.</p> <p>3.7 Actualiza en el foro el hilo de discusión relacionado a la categoría modificada, realizando en él, la misma modificación.</p> <p>3.8 Termina el DCP</p>	
3.2	Cancela acción	NA	<p>3.2 Termina la ejecución de la acción: Editar categoría.</p> <p>3.3 Regresa a la interfaz de Administrar categorías. Termina el DCP</p>	Satisfactorio

CAPÍTULO 3: Validación de la solución propuesta.

Una vez terminados todos los casos de prueba se procede a la revisión de los mismos, con el fin de eliminar redundancia o corroborar que no falte ninguno.

Durante la validación de las funcionalidades implementadas se realizaron dos iteraciones, en la primera de ellas algunas funcionalidades resultaron insatisfactorias, realizando así una segunda iteración donde todas las funcionalidades obtuvieron resultados satisfactorios y alcanzando un sistema factible.

En este capítulo se abordó uno de los métodos de pruebas utilizados para la validación de la solución propuesta. Se explicó de modo general las pruebas de Caja Negra, las cuales se realizaron mediante la descripción de casos de prueba, para lo cual se mostró un ejemplo que evidencia uno de ellos. Además se mencionaron los resultados que se obtuvieron de la realización de esas validaciones al sistema implementado.

CONCLUSIONES GENERALES

CONCLUSIONES GENERALES

Con el desarrollo de este trabajo se cumplió satisfactoriamente el objetivo general propuesto. De esta forma se concluye que:

- Se continuó con la línea arquitectónica utilizada en el proyecto RHODA.
- Se utilizaron para la implementación como sistemas de notificaciones el RSS y el correo electrónico.
- Se implementó el sistema de notificaciones para RHODA, el cual brinda información relevante a los usuarios sin necesidad de visitar el mismo.
- Se validó el sistema de notificaciones implementado mediante las pruebas de Caja Negra, obteniendo resultados satisfactorios.

RECOMENDACIONES

RECOMENDACIONES

A partir del presente trabajo se proponen las siguientes recomendaciones:

1. Realizar un estudio con respecto a las notificaciones en los módulos que se encuentran desarrollando en la versión 2.1 del sistema.
2. Continuar con el estudio de los sistemas de notificaciones en aplicaciones Web para mantener así la actualización del repositorio.

BIBLIOGRAFIA CONSULTADA

REFERENCIAS BIBLIOGRÁFICAS

1. **RAE.** Real Academia Española. [En línea] 2009. <http://www.rae.es/rae.html>.
2. **López Guzmán, Clara.** Los Repositorios de Objetos de Aprendizaje como soporte para entornos e-learning. Tesis de grado, Universidad de Salamanca, España, 2005.
3. **Maza Yamasaki, Leonardo y Espinosa Franco, Carolina.** SLOR: Simple Learning Object Repository, diseñando un repositorio de objetos de aprendizaje para la plataforma de teleeducación EducAD-UNACH (basada en MOODLE). 2002.
4. **Nuevastecnologiasyprotecciondedatos.** [Online] 2008. <http://nuevastecnologiasyprotecciondedatos.blogspot.com/2008/06/el-documento-electrnico-en-la.html>
5. **ministeriopublico.poder-judicial.** [Online] 2009. <http://ministeriopublico.poder-judicial.go.cr/biblioteca/libros/juzgado%20violencia%20domestica/09-Notificacion.pdf>
6. **2005ocubom's blog.** [Online] 2009. <http://ocubom.wordpress.com/category/informatica/web/>
7. **XTREMOS.** [Online] 2010. <http://www.xtremos.net/posts/xtremos/24649/Lo-que-faltaba-en-Xtremos-Acceso-RSS.html>
8. **nicaraguaeduca.** [Online] 2010. <http://nicaraguaeduca.edu.ni/.../1640-agregadores-de-noticias>
9. **wordlingo.** [Online] 2009. <http://www.worldlingo.com/ma/enwiki/es/RSS>
10. **W3C.** [Online] 2008. <http://www.w3c.es/divulgacion/guiasbreves/tecnologiasxml#uso>
11. **xuletas.** [Online] 2010. <http://www.xuletas.es/ficha/protocolos-de-correo-electronico-1/>
12. **kioskea.net.** [Online] 2009. <http://es.kioskea.net/contents/internet/smtp.php3>
13. **Blog de Tecnología.** [Online] 2009. <http://blogxdextecnologia.blogspot.com/2009/08/el-servidor-sendmail.html>

BIBLIOGRAFIA CONSULTADA

14. **Apunts informatics.** [Online] 2009.
http://pcyous.blogspot.com/2009_06_01_archive.html
15. **buenastareas.** [Online] 2010. <http://www.buenastareas.com/ensayos/Protocolos-E-Mail-Imap-Y-Pop3/1142815.html>
16. **periospino.blogspot.** [Online] 2010.
<http://periospino.blogspot.com/2010/06/usando-javamail-api-de-envio-de-correos.html>
17. **taringa.** [Online] 2010. <http://www.taringa.net/posts/ebooks-tutoriales/7114524/Enviar-e-mails-con-PHPMailer.html>
18. **sentidoweb.** [Online] 2008. <http://sentidoweb.com/2008/07/30/swift-mailer-libreria-php-para-envio-de-emails.php>
19. **Frameworks PHP.** [Online] 2001.
<http://www.taringa.net/posts/linux/6937435/Frameworks-PHP-y-CSS.html>
20. **Salvador Broche, Orlando Felipe y Martín Soler, Javier.** Implementación de un Repositorio de Objetos de Aprendizaje para la Universidad de las Ciencias Informáticas. Trabajo de Diploma en opción al título de Ingeniero en Ciencias Informáticas. La Habana. 2010.
21. **symfony.** [Online] 2001. http://www.symfonyproject.org/jobeeet/1_4/Propel/es/16
22. **ELEMENTOS HARDWARE** [Online] 2011. <http://elementohardware.blogspot.com/>
23. **Scrib.** [Online] 2008. <http://es.scribd.com/doc/2050925/metodologias-de-desarrollo-software>
24. **Letelier & Penadés.** Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP), Universidad Politécnica de Valencia
25. **Gómez Gallego, Juan Pablo.** FUNDAMENTOS DE LA METODOLOGIA RUP RATIONAL UNIFIED PROCESS, UNIVERSIDAD TECNOLÓGICA DE PEREIRA, 2007.
26. **OpenArchives.** [Online] 2002.
www.openarchives.org/.../openarchivesprotocol.html

GLOSARIO

E-learning: un sistema de aprendizaje electrónico en el que participan el uso de las Tecnologías de la Información y las Comunicaciones y elementos pedagógicos, para juntos facilitar proceso de enseñanza- aprendizaje (2).

Framework: en el desarrollo de software es una estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros software para ayudar a desarrollar y unir los diferentes componentes de un proyecto (22).

Objeto de Aprendizaje: cualquier recurso con una intención formativa, compuesto de uno o varios elementos digitales, descrito con metadatos, que pueda ser utilizado y reutilizado dentro de un entorno e-Learning (2).

Repositorio: un lugar donde la información digital es almacenada, para ser localizada y distribuida a través de la red (2).

Repositorio de Objetos de Aprendizaje: lugar donde los objetos de aprendizaje son almacenados, para ser localizados y distribuidos a través de la red (2).

RUP: Es un proceso en el que se han unificado técnicas de desarrollo y metodologías, está orientado a objetos y preparado para desarrollar grandes y complejos proyectos (25).

LOM: Es un modelo de datos para describir OA y recursos digitales, usualmente usado en XML, el cual está certificado por la IEEE, lo cual significa que es un estándar de jure (20).

OAI-PMH: Es una herramienta de interoperabilidad independiente de la aplicación que permite realizar el intercambio de información para que desde puntos (proveedores de servicio), se puedan hacer búsquedas que abarquen la información recopilada en distintos repositorios asociados (proveedores de datos) (26).