

Universidad de las Ciencias Informáticas

Facultad 4



“Implementación y prueba del módulo Administración para RHODA 2.1”.

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

Autor

Roberto Castro Ferrer

Tutor

Ing. Yandris Mata Cabrera

Co-Tutor

DrC Ing. Augusto César Rodríguez Medina

Declaración de autoría

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

Para que así conste firmamos la presente a los _____ días del mes de _____ del 2011.

Roberto Castro Ferrer

Firma del Autor

Ing. Yandris Mata Cabrera

Firma del Tutor

DrC Ing. Augusto César Rodríguez Medina

Firma del Co-Tutor

Resumen

La evolución y desarrollo de las Tecnologías de la Información y las Comunicaciones junto con la aparición y uso de Internet propiciaron el surgimiento del e-learning. La gestión de contenidos dentro del e-learning ha evolucionado y se están construyendo los contenidos educativos como Objetos de Aprendizaje, que se contienen en Repositorios de Objetos de Aprendizaje. En la Universidad de las Ciencias Informáticas debido a la necesidad de gestionar contenidos educativos como Objetos de Aprendizaje y almacenarlos, se desarrolla el repositorio RHODA. En el presente trabajo se propone como objetivo principal mejorar el módulo de Administración de la versión 2.1 del repositorio RHODA, agregando nuevas funcionalidades que aumenten y faciliten el trabajo del administrador y proporcionen más control del repositorio. Se muestra un estudio de sistemas web y repositorios en cuanto a la gestión de la administración. Se analizan y resumen las principales características de las herramientas y tecnologías utilizadas en el desarrollo de las nuevas funcionalidades y se describen los procesos aplicados durante el flujo de prueba.

Palabras claves: Repositorio, Objeto de Aprendizaje, e-learning, Metadatos.

Agradecimientos

- A mis padres Nancy y José por sacrificar la vida por mi en cada momento.
- A mi hermana Yaneisy por ser mi apoyo y mi mejor amiga.
- A Mami Ramona y Papi Felo por ser mis segundos padres y darme todo el amor del mundo.
- A mis sobrinos Jonathan y Angelina por ser la luz de mis ojos.
- A Migue por ser parte de mi familia y apoyarme.
- A mis tías del alma Dalia y Maira por ser cómplices en todas mis locuras.
- A mis amigas y amigos incondicionales Yuleisy, Yusel, Suramis, Emilio, Rolando y Lesyanis por estar junto a mí en los momentos buenos y malos.
- A mis amigas Arianna, Yamila y Mirian por soportarme y quererme.
- A Fidel, Yorjanis y Danny por todo el apoyo ofrecido en este trabajo.
- A Yasmany y Bleidys por toda la ayuda en el desarrollo de la tesis, sin ellos hubiese sido imposible.
- A mi tutor Yandri y a mi cotutor Augusto por todo el apoyo y guía en el desarrollo de la tesis.
- A el equipo de desarrollo de RHODA en especial a Yailín por ser tan comprensiva y ayudarme.
- Al tribunal de tesis por guiarme, enseñarme y ayudar a superarme a lo largo de esta investigación.
- A los profesores del proyecto RHODA Yailen, Orlando y Javier por su apoyo y ayuda.
- A Roxana por enseñarme tanto a lo largo de 4to y 5to año de la carrera.
- A mis profesores que a lo largo de estos 5 años me apoyaron y transmitieron su conocimiento.
- A los miembros de mi grupo 8104 Germana, Anabel, Ernesto, Carlos, Dahymi, Roberto Carlos, Marlys y Yaismel por ser especiales.
- A los miembros del grupo 8305 Liliana, Yordanka, Adrianet, Leanet, Oksana, Mayliuvis, Daymi, Ailen, Eisel y Jorjito por estar presentes en esta importante etapa.
- A los miembros del 4504 Lizandra, Danay, Gretell, Arlenis, Mirurgia, Naydeline, Anailkis y Mario por ser buenos compañeros.
- A todos los miembros del comité primario Cartaya, Olia, Yunior, Yusdel, Yudislandry, Karolkis y Daylen junto a los cuales pasé muchos buenos y malos momentos.
- A el Comité por la Liberación de los 5 Héroeos por ser un motivo de lucha en mi vida, en especial a Bencomo y Suset por su apoyo y dedicación a la causa.
- A los muchachos de Octavitos por creer en mi y en mis ideas sin importar cuán arriesgadas fueran, además de mantenerme siempre informado.

Dedicatoria

- A mis padres Nancy y José por ser mi corazón y mi razón de ser.
- A Mami Ramona y Papi Felo por ser mi fuerza y la ternura de mi vida.
- A mi hermana Yaneisy por su apoyo y comprensión en todo momento.
- A mis sobrinos Jonathan y Anyelina por ser mi futuro.
- A mi Abuela Toña por darme una madre tan buena.
- A mis tías Dalia y Miara por su apoyo y dedicación.
- A toda mi familia por estar presente en mi vida y hacer que los ame y los quiera.
- A todos mis amigos que siempre estarán en mi corazón.

Índice

Introducción.....	8
Capítulo 1: Fundamentación teórica.....	12
1.1 Tipos de Repositorios.....	12
1.2 Administración.....	15
1.2.1 Sistemas web.....	15
1.2.2 Repositorios.....	17
1.3 Framework de desarrollo Symfony.....	22
1.4 Tecnologías.....	24
1.4.1 JavaScript.....	24
1.4.2 HTML.....	24
1.4.3 CSS.....	25
1.4.4 eXist XML.....	25
1.4.5 PostgreSQL.....	25
1.4.6 NetBeans IDE.....	26
1.4.7 UML.....	26
1.4.8 Servidor Apache.....	26
1.4.9 Librería ExtJS.....	26
1.5 Visual Paradigm for UML.....	26
1.6 Modelo CMMI y Metodología para el desarrollo: RUP.....	27
1.6.1 Modelo CMMI.....	27
1.6.2 RUP.....	29
1.6.3 CMMI-RUP.....	29
Capítulo 2. Implementación y prueba.....	31
2.1 Funcionalidades.....	31
2.1.1 Administrador de OA.....	31
2.1.2 Consultar historial de revisor.....	32
2.1.3 Editar configuración de la conexión a la base de datos postgresql.....	33
2.1.4 Editar configuración de la conexión a la base de datos nativa XML eXist.....	33
2.1.5 Configurar página de mantenimiento.....	34
2.1.6 Configuración de correo electrónico SMTP.....	34
2.1.7 Configurar tipo de autenticación.....	35

2.1.8 Configurar tamaño máximo de los OA.....	36
2.1.9 Listar comprobaciones del servidor.....	37
2.2 Requisitos no funcionales.	37
2.2.1 Accesibilidad.....	37
2.3 Diagrama de casos de uso.....	38
2.4 Diagrama de Componentes.....	39
2.5 Diagrama de despliegue.....	39
2.6 Validación de la solución propuesta.....	40
2.6.1 Pruebas de Caja Blanca.....	40
2.6.2 Pruebas de Caja Negra.....	44
Conclusiones generales.....	50
Recomendaciones.....	51
Referencias bibliográficas.....	52
Anexos.....	56
Glosario de términos.....	73

Introducción

En la actualidad los sistemas educativos del mundo se enfrentan al desafío de utilizar las Tecnologías de la Información y las Comunicaciones para proveer a las personas con las herramientas y los conocimientos necesarios que se requieren en el siglo XXI. Estas tecnologías por su parte cumplen hoy un rol esencial en los procesos de enseñanza-aprendizaje, convirtiéndose en medios y plataformas que facilitan y enriquecen el trabajo de alumnos y profesores. Actualmente se cuenta con métodos que mejoran el proceso de enseñanza-aprendizaje, haciendo a un lado las limitaciones del método de aprendizaje tradicional.

Con el surgimiento de Internet estos métodos pasan a ser un importante factor, tomando ventaja de las facilidades que provee la red mundial, dando lugar así al surgimiento del Aprendizaje Electrónico (e-learning) que es definido como: *“la capacitación no presencial que, a través de plataformas tecnológicas, posibilita y flexibiliza el acceso y el tiempo en el proceso de enseñanza-aprendizaje, adecuándolos a las habilidades, necesidades y disponibilidades de cada docente, además de garantizar ambientes de aprendizaje colaborativos mediante el uso de herramientas de comunicación síncrona y asíncrona, potenciando en suma el proceso de gestión basado en competencias.”* (García, 2009). Con el surgimiento del e-learning se abren las puertas a nuevas áreas y formas de interactuar con el contenido docente educativo.

La gestión de contenidos en los entornos e-learning está generando cambios en diversos sentidos dentro de los métodos de enseñanza-aprendizaje. Una nueva tendencia propiciada por estos cambios es que los contenidos educativos se están construyendo como Objeto de Aprendizaje (OA), que se define como: *“Recurso digital con una granularidad apropiada y una marcada intención formativa, compuesto por uno o varios objetos de información, con un único objetivo, descrito con metadatos y con un comportamiento secuenciado que asegure el correcto enlace entre los elementos de su estructura didáctica y que puede ser reutilizado en entornos e-learning”.* (Rodríguez, 2011).

En el mundo se utilizan repositorios para el almacenamiento de la información de diferentes instituciones o disciplinas generada en su labor investigativa. Estos Repositorios de Información se clasifican en Temáticos e Institucionales, que a su vez se agrupan dentro de las categorías de Repositorio Abierto y Repositorio de Objetos de Aprendizaje (ROA). Los ROA son clasificados teniendo en cuenta aspectos como *“...la forma en la que se concentran los recursos y por la forma en la que los*

catálogos de metadatos se organizan” (López, 2006). Los repositorios de acuerdo a la organización de los catálogos se clasifican en *“centralizados y distribuidos”* (Downes, 2004).

Los OA se recopilan en ROA definidos como: *“un sistema software que almacena recursos educativos y sus metadatos (o solamente estos últimos), y que proporciona algún tipo de interfaz de búsqueda de los mismos, bien para interacción con humanos o con otros sistemas software“* (ANCED, 2007). Los repositorios los organizan y los hacen accesibles para diferentes aplicaciones y perfiles de usuarios.

En el mundo existen innumerables ROA publicados en Internet, entre ellos se encuentran: MERLOT¹ (Multimedia Educational Resource for Learning and Online Teaching), CAREO² (Campus Alberta Repository of Educational Objects), SMETE³ (Science, Mathematics, Engineering and Technology Education), SLOR (Simple Learning Object Repository), Agrega⁴, Morea⁵ y Conexión⁶.

La administración de un repositorio juega un papel fundamental en el manejo y protección de los OA y toda la información contenida en él, facilitando que se realicen cambios autorizados en el OA, se valore la legitimidad del mismo y se impida un manejo inadecuado por personal no autorizado. La mayoría de los sistemas antes mencionados proponen un sistema basado en roles de usuarios que van desde un invitado (usuario con menor privilegio) hasta un administrador (usuario de mayor privilegio), asegurando así que los usuarios tengan solamente acceso a la información y a las funcionalidades técnicas del sistema de acuerdo al nivel de jerarquía de su rol.

Cuba lucha por no mantenerse al margen de la evolución tecnológica y aunque en el país ya existen repositorios como “La universidad de la salud. Cuba” perteneciente a INFOMED, y los llamados “Repositorios de la Comunidad Cubana” destinados a brindar todo lo necesario en cuanto a software libre en Cuba, la Universidad de las Ciencias Informáticas (UCI) juega un papel fundamental en este aspecto. La UCI comenzó en el año 2008 el desarrollo de un Repositorio de Objetos de Aprendizaje llamado RHODA⁷; en la primera versión este destaca por contener importantes funcionalidades como la creación, almacenamiento y búsqueda de OA.

Debido a la necesidad de la UCI de crear y almacenar contenidos educativos como OA se crea la versión 2.0 de RHODA. Esta contiene nuevas funcionalidades que añaden grado de madurez al repositorio, entre las cuales destacan el trabajo con foros, estadísticas y la utilización del estándar

1 Disponible en: <http://www.merlot.org/>

2 Disponible en: <http://www.careo.org/>

3 Disponible en: <http://www.smete.org/smete/>

4 Disponible en: <http://redes.agrega.indra.es/>

5 Disponible en: <http://morea.usc.es/home/home.php>

6 Disponible en: <http://cnx.org/>

7 Disponible en: <http://roa.uci.cu/roa.php>

SCORM definido como: “...un conjunto de estándares técnicos que permiten a los sistemas de aprendizaje basados en web, el encontrar, importar, compartir, reusar y exportar contenidos de formación de manera normalizada.” (Rouyet y Martín, 2006) en su versión 2004 para la creación y empaquetamiento de los OA.

Aunque es notable el desarrollo de la versión 2.0 de RHODA se debe señalar que el repositorio en la gestión de la administración no posibilita un manejo adecuado de los datos de usuario, así como la inscripción de los mismos carece de opciones y funcionalidades necesarias para un óptimo funcionamiento.

Los reportes dentro de la gestión de la administración son pobres en cuanto a las opciones que ofrecen, porque las funcionalidades implementadas no explotan al máximo las informaciones registradas para hacer una mayor variedad de reportes. El sistema actual no ofrece la opción de agrupar los reportes de acuerdo a criterios más específicos de los usuarios.

Teniendo en cuenta los aspectos expuestos anteriormente sobre las limitaciones que presenta RHODA se plantea el siguiente **problema de investigación**: ¿Cómo mejorar las funcionalidades del módulo Administración en la versión 2.1 de RHODA para lograr una mejor gestión de los procesos?

Se define como **objeto de estudio**: el proceso de gestión de la administración dentro de los sistemas web y como **campo de acción**: los procesos de administración en los Repositorios de Objetos de Aprendizaje.

Se define como **objetivo general**: Desarrollar las nuevas funcionalidades que permiten mejorar el módulo de Administración dentro de la versión 2.1 del Repositorio de Objetos de Aprendizaje (RHODA) para mejorar los procesos existentes en dicha herramienta.

Como **idea a defender** se plantea que con el mejoramiento de las funcionalidades del módulo Administración se tendrá un mayor control de las acciones ejecutadas dentro del Repositorio de Objetos de Aprendizaje RHODA 2.1.

Objetivos específicos:

- Analizar el estado de los Repositorios de Objetos de Aprendizaje para elaborar el marco teórico conceptual.
- Implementar el módulo de Administración.

- Realizar las pruebas del módulo de Administración para verificar que cumplan con los requisitos especificados.

Para dar cumplimiento a las tareas de investigación se emplean los **métodos teóricos**:

Análisis Histórico-Lógico: Se utiliza con el fin de realizar el análisis de los antecedentes y tendencias de los principales Repositorios de Objetos de Aprendizaje existentes en el mundo y en Cuba, lo que facilitó apreciar su evolución y sistematizar los fundamentos teóricos y prácticos de los mismos; definiendo así sus características, ventajas y desventajas.

Analítico-Sintético: Permite el estudio y análisis de la bibliografía referente al tema, así como el estudio de los conceptos, definiciones y avances acerca de la situación descrita anteriormente, lo que permitirá ofrecer una propuesta adecuada que solucione el problema planteado.

Modelación: Se utiliza para representar la información generada durante el flujo de implementación y prueba.

Capítulo 1: En este capítulo se abordan las características, funciones y tipologías de los Repositorios de Objetos de Aprendizaje, identificando las herramientas y repositorios más utilizados a nivel mundial. Se estudiarán sitios web y Repositorios de Objetos de Aprendizaje en cuanto a su administración. Se analizan las principales actualizaciones de las herramientas y la metodología a utilizar.

Capítulo 2: Se describe la solución propuesta al problema planteado, partiendo de los análisis que se han realizado anteriormente. Se hace una descripción minuciosa de los elementos valorados a la hora de mejorar el módulo de Administración. Se recogen los artefactos generados durante la implementación. Se exponen todos los resultados obtenidos durante la validación de las soluciones a través de todas las pruebas realizadas.

Capítulo 1: Fundamentación teórica

Introducción.

En este capítulo se abordan temas referentes a los Repositorios de Objetos de Aprendizaje, se enfatiza en sus definiciones, características, tipologías y se sintetizan los principales aspectos de algunos de los repositorios más importantes en el mundo. Se recogen características de administración dentro de aplicaciones web y ROA.

Se explican el modelo CMMI (Capability Maturity Model Integration) y RUP (Rational Undefined Process) como metodología para el desarrollo, centrándose en el flujo de implementación y prueba. Se abordarán los puntos de convergencia al aplicar CMMI y RUP. Se referencian los principales elementos de las diferentes tecnologías utilizadas en el desarrollo de la herramienta.

1.1 Tipos de Repositorios.

Las universidades desde sus inicios hasta la actualidad han constituido uno de los polos investigativos con mayor producción de conocimiento e investigaciones que crece cada día más con el avance de la ciencia y la tecnología. Se puede decir que el nivel de madurez de un centro académico se puede evaluar por la cantidad de conocimiento desarrollado y archivado, para su posterior reutilización. Así surge la necesidad de tener un lugar donde almacenar los conocimientos generados a partir de la investigación. La Real Academia Española (RAE) define la palabra **repositorio** como: “*Armarío o alacena, lugar donde se guarda algo*” (RAE, 2010). Con el desarrollo de las tecnologías y la digitalización de casi toda la información se hizo necesario transpolar este concepto y adquirió un nuevo significado dentro del ámbito digital donde se define como: “*un sitio centralizado donde se almacena y mantiene información digital*” (López, 2005).

Los repositorios en el mundo son utilizados para el almacenamiento de la información digital de diferentes instituciones o disciplinas generada en su labor investigativa. Estos Repositorios de Información se clasifican en “*Temáticos e Institucionales*” (López, 2005), que a su vez se agrupan dentro de las categorías de Repositorio Abierto y Repositorios de Objetos de Aprendizaje.

Repositorios Institucionales

Los Repositorios Institucionales (RI) se definen como: “*...un servicio para organizar, gestionar, preservar, difundir y ofrecer acceso libre a la producción científica y académica en soporte digital, generada por los miembros de la institución*” (SBDI⁸, 2010). Es necesario añadir a este concepto que

⁸ Sistema de Bibliotecas, Documentación e Información de la Universidad de Costa Rica.

no solamente es un servicio sino también un lugar donde almacenar los materiales generados. Los RI surgen por la necesidad de almacenar toda la actividad investigativa de una institución dada y facilitar con ello que se pueda acceder de manera directa y rápida a la información.

Repositorios Temáticos

Los Repositorios Temáticos se definen como: *“Archivos que recolectan y gestionan objetos digitales académicos relativos a una o más áreas temáticas.”* (BDBF⁹, 2009). Las universidades de todo el mundo han sentido la necesidad de fortalecer sus archivos investigativos generados dentro de su institución porque con el desarrollo de Internet se han abiertas nuevas puertas a la colaboración entre personas de todo el mundo con el objetivo de desarrollar proyectos investigativos de una rama específica.

Repositorios Abiertos

En la actualidad las publicaciones científicas son privativas por lo que el acceso a ellas es bastante costoso y no todos los usuarios pueden pagarlas. Como respuesta a este problema de escala mundial surge el movimiento de “Open Access”¹⁰ (Acceso Abierto) que tiene como objetivo la socialización de la información sin ningún interés económico. A partir de las especificidades del Open Access se puede definir que un Repositorio Abierto (RA) es un lugar donde se van a almacenar los recursos que se generen a partir de la actividad científica, y que permitirá el acceso abierto a la información contenida en él, sin ningún costo monetario de uso y publicación.

Repositorios de Objetos de Aprendizaje

Debido a la necesidad de almacenar los OA surgen los llamados Repositorios de Objetos de Aprendizaje (ROA) que se definen como: *“...un sistema software que almacena recursos educativos y sus metadatos (o solamente estos últimos), y que proporciona algún tipo de interfaz de búsqueda de los mismos, bien para interacción con humanos o con otros sistemas software”* (ANCED, 2007).

Los ROA son clasificados teniendo en cuenta aspectos como *“...la forma en la que se concentran los recursos y por la forma en la que los catálogos de metadatos se organizan”* (López, 2006). Los repositorios de acuerdo a la organización de los catálogos se clasifican en *“...centralizados y distribuidos”* (Downes, 2004).

Clasificación por concentración de recursos

⁹ Biblioteca Digital de Bioquímica y Farmacia.

¹⁰ Nombre del Movimiento para el Acceso Abierto a la Información.

En la clasificación de los ROA teniendo en cuenta la concentración de los recursos se identifican dos tipos (Downes, 2004):

a) los que contienen los objetos de aprendizaje y sus metadatos, en éstos los objetos y sus descriptores se encuentran dentro de un mismo sistema e incluso dentro de un mismo servidor (Figura 1a).

b) los que contienen sólo los metadatos, en este caso el repositorio contiene sólo los descriptores y se accede al objeto a través de una referencia a su ubicación física que se encuentra en otro sistema o repositorio de objetos (Figura 1b).

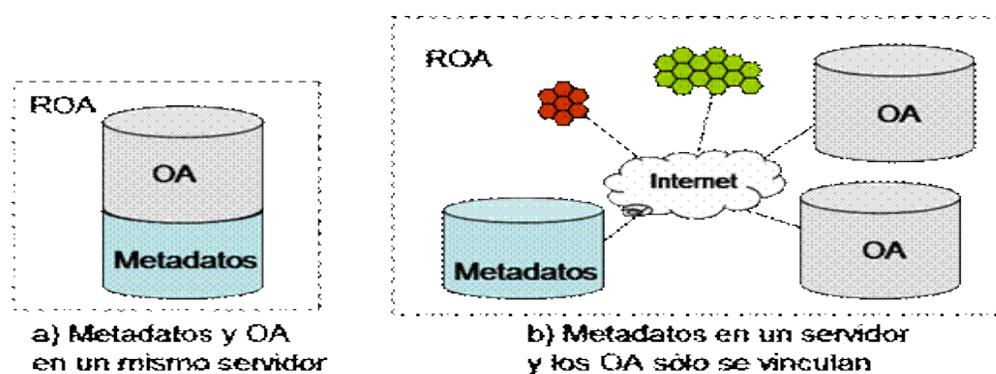


Figura 1. Tipos de repositorios según concentración de recursos (López, 2006).

Clasificación por organización de catálogos

Teniendo en cuenta la organización de los catálogos, los repositorios se clasifican en “centralizados y distribuidos” (Downes, 2004).

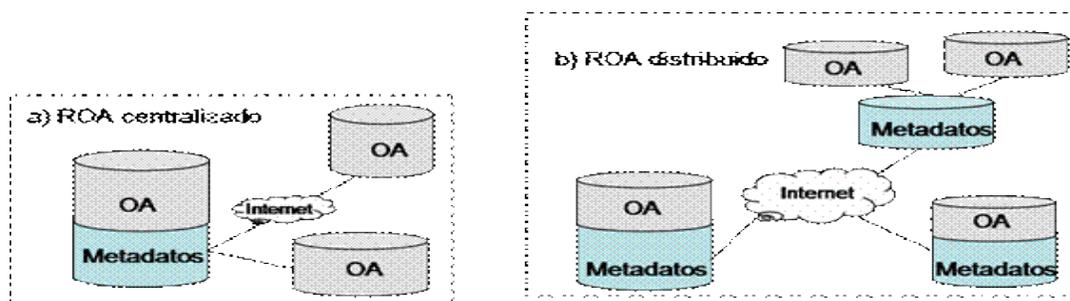


Figura 2. Tipos de ROA por la distribución de los metadatos (López, 2006).

Son centralizados cuando los metadatos se encuentran contenidos dentro del mismo repositorio y/o servidor sin importar que los recursos se encuentren almacenados en otro lugar como se muestra en la (figura a). Los repositorios centralizados son más utilizados actualmente que los distribuidos, estos

últimos consisten en la conexión entre varios repositorios que contienen diferentes metadatos y se conectan entre sí para referenciarlos y compartirlos (*Figura b*).

1.2 Administración.

La administración es tan antigua como la misma civilización, desde los inicios el hombre tuvo la necesidad de organizarse. Esta palabra etimológicamente viene del latín “**Administratione**” que se define como: “*acción de administrar y el término de administrar está compuesto por dos vocablos: **ad** y **ministrare**, que significa conjuntamente “**servir**”, llevando en forma implícita en su sentido que es una **actividad cooperativa** que tiene el propósito de servir”.* (Thompson, 2008). Actualmente la acción de administración se ha convertido en todo un proceso por lo que ha adquirido nuevas perspectivas y se define como: “*El proceso de planificar, organizar, dirigir y controlar el uso de los recursos y las actividades de trabajo con el propósito de lograr los objetivos o metas de la organización de manera eficiente y eficaz.*” (Thompson, 2008).

Con el surgimiento y evolución de las tecnologías surgió la necesidad de administrar las aplicaciones que se fueron creando. Muchas de estas aplicaciones necesitan constantes actualizaciones, sobre todo las aplicaciones web.

1.2.1 Sistemas web.

Los sistemas web son muy utilizados a nivel mundial para diversos usos dentro de cualquier rama, tanto económica como social. Todos estos sistemas tienen la necesidad de gestionar sus contenidos y administrar sus configuraciones para mantenerse en buen funcionamiento y en constante actualización. Las tareas de administración son específicas para cada sistema en dependencia de los servicios que brinde y la tecnología que utilice. Aún en el caso de los Sistema de Gestión de Contenidos que brindan una plataforma estándar para el desarrollo de sitios web, la administración es adaptable a las necesidades de los desarrolladores.

Joomla:¹¹ es un Sistema de Gestión de Contenidos (CMS) que permite la creación de sitios web y otras aplicaciones en línea, con un fuerte sistema para la administración de sus contenidos. Este sistema incluye un portal web para la administración al cual se accede de forma independiente al sitio web que se tenga publicado. Para acceder a este sitio existen 3 roles:

11 Disponible en: www.joomlaspanish.org/

Súper administrador: tiene control total de todo el sitio y sus configuraciones.

Administrador: tiene limitaciones en cuanto a eliminar o editar usuarios con el rol de súper administradores, ni accede a la configuración global del sistema.

Manager: su acceso al admin¹² es limitado porque no puede acceder a la administración de componentes, ni editar los usuarios ni ningún elemento de configuración.

Su panel de administración está excelentemente estructurado, el mismo cuenta con 4 áreas: el menú principal que es indispensable dentro de la navegación, la barra de herramientas que permite la manipulación de los contenidos, la barra de información que ofrece reportes sobre el sistema y el área de trabajo donde se muestran las opciones de configuración elegidas en los menús.



Figura 3. Panel de administración de Joomla (Joomla, 2011).

Dentro de la estructura que presenta JOOMLA, es muy interesante la forma en que separa la administración de la presentación de los contenidos. Esto ofrece la ventaja de trabajar en la administración de la aplicación sin necesidad de interactuar de forma directa con la información, lo cual hace menos engorroso el trabajo. Además facilita tener un panel de administración más ameno, visual y funcionalmente más completo, evitando así la preocupación del tamaño del panel con respecto a las dimensiones de la aplicación.

Colegio de Ingenieros Eléctricos y Electrónicos del Azuay (CIEELA):¹³ es un sitio web destinado a elevar el nivel de los ingenieros asociados y organizar la información del área administrativa. El acceso a este portal se realiza mediante un usuario y contraseña proporcionado por CIEELA, que está asociado a dos roles fundamentales, usuario y administrador. El administrador es un usuario sin

¹² Nombre del panel de administración de Joomla.

¹³ Disponible en: <http://www.cieela.org>

restricciones que tiene acceso a todas las funcionalidades del sistema. Al registrarse en CIEELA en modo de administrador se tiene acceso a las siguientes acciones dentro del panel de administración: Actualización de afiliación, Listado de socios, Maletín web, Nueva afiliación, Nueva directiva, Nuevo pago, Vídeo Streaming, Reportes.



Figura 4. Panel de administración de CIEELA (CIEELA, 2009).

CIEELA no presenta características muy relevantes en cuanto a la gestión de la administración, debido a que las funcionalidades identificadas en ella responden de manera específica al motivo por el que fue creado. Es necesario señalar que dentro del panel de administración se puede observar que tienen un sistema de reportes que permite obtener información de cada una de las acciones que se realizan dentro de la aplicación. Los reportes son un aspecto fundamental a tener en cuenta dentro de la gestión de administración porque permiten al administrador monitorear y evaluar el estado de la aplicación.

1.2.2 Repositorios.

Los ROA se encuentran en línea y están en constante cambio al tener que publicar materiales de diferentes temáticas. La administración no solo se limita a este proceso que es el más visible para todos los usuarios, sino que existen procesos que realiza el administrador de manera anónima pero que son de vital importancia para mantener el servicio.

Por lo general los repositorios a nivel mundial mantienen su sistema y panel de administración de forma secreta por una razón de seguridad, y esto hace muy difícil marcar un estándar de panel o sistema de administración. Existen diferentes herramientas para el desarrollo de repositorios y

repositorios en sí que se pueden analizar en cuanto a la administración, porque son de uso libre o tienen publicados materiales a cerca de estos aspectos.

Dspace:¹⁴ es un repositorio que tiene estructurada su información de forma jerárquica por lo que es necesario establecer roles para controlar el acceso y administración de las funcionalidades y recursos que ofrece. Se definen dos roles dentro de este repositorio que son el administrador, que consta con todos los privilegios dentro de la aplicación y otros usuarios, sus actividades estarán definidas por los privilegios que le adjudique el administrador. El administrador podrá crear, organizar, distribuir y eliminar los OA teniendo en cuenta las comunidades, subcomunidades, colecciones e ítems que se permiten crear en Dspace. Se encarga además de la gestión de usuarios que consiste en la incorporación de nuevos usuarios y edición de sus perfiles. Estos usuarios pueden ser contenidos en grupos donde se definen las tareas de cada usuario y la política de funcionamiento de dicho conjunto, que solo puede ser variada por el administrador. En el panel de administración se encuentran las siguientes opciones: Comunidades/Colecciones, Usuario, Grupos, Ítems, Registro Dublin Core, Registro de formato Bitstream, Flujo de trabajo, Autorización, Editar noticias, Supervisores, Estadísticas, Ayuda y Salir.



Figura 5. Panel de administración de DSpace (Solano, 2007).

El repositorio Dspace presenta como características novedosas la forma de gestionar los usuarios para definir políticas de funcionamiento, tanto para un usuario específico como para un grupo. Esto facilita el control de los usuarios y sus acciones dentro del repositorio. La definición de restricciones y políticas de funcionamiento es un buen aspecto a tener en cuenta para fortalecer el sistema de administración enfocado a la gestión de usuarios.

14 Disponible en: <http://www.dspace.org> se puede descargar en: <http://sourceforge.net/projects/dspace/>

Agrega:¹⁵ es un repositorio que tiene estructurada su red de forma muy peculiar, está compuesta por un grupo de nodos que están asociados a un nodo central AGE (Administración General del Estado). Estos nodos, tanto el central como los nodos autónomos de la red tienen una estructura que permite separar las funcionalidades en dos bloques, uno que es el referente a los usuarios del repositorio y otro dedicado a las tareas de administración; este último es accedido solo por el administrador que tiene todos los privilegios para administrar y configurar el nodo. La arquitectura del portal de administración va a estar compuesta por tres áreas fundamentales:

Publicaciones de servicios web (interfaz de interoperabilidad): contiene a Presentar/Almacenar, Solicitar/Entregar, Buscar/Mostrar y Servicios adicionales.

Módulos funcionales: engloba a los módulos de Autenticación, Gestión de roles, Autorización, Localización, Gestión de licencias y Auditoría de acceso, Indexador, Planificador, Monitorización, Configuración, Agregador RSS y Logs.

Capa de acceso a datos: recoge a los Repositorios de Contenidos y Metadatos, Usuarios, Datos Relacionales y Configuración de Logs.

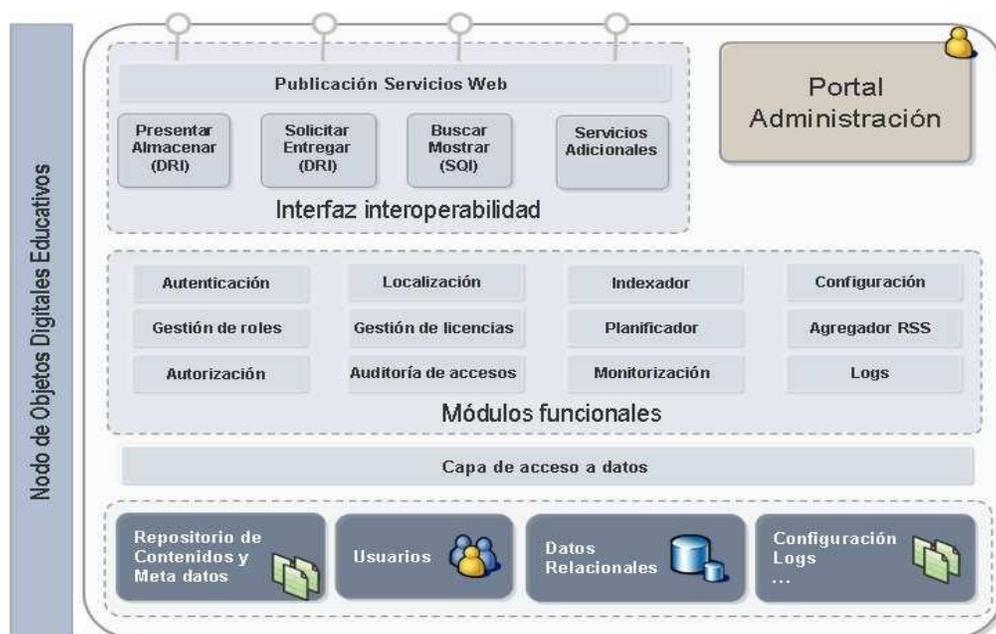


Figura 6. Portal de administración de Agrega (Canabal y Sarasa, 2007).

15 Disponible en: <http://redes.agrega.indra.es/>

Agrega presenta una excelente estructura dentro del portal de administración perteneciente a cada nodo. La organización de las funcionalidades en las tres capas antes señaladas permite un mejor control y fácil acceso para localizar las funcionalidades. El sistema de reporte está muy bien estructurado, la configuración y trabajo con los logs desde la aplicación son también características a tener en cuenta debido a su importancia en el control de la seguridad del repositorio por parte del administrador.

Riboa:¹⁶ es un ROA implementado para la Universidad de Oriente que tiene su gestión de usuarios basada en 3 roles fundamentales: usuarios simples, usuarios creadores y administradores. El rol de administrador de esta aplicación tiene asociado el acceso a todas las funcionalidades existentes sin restricción y se encarga de administrar los OA y las configuraciones de la herramienta. Un administrador dentro de su panel de administración y en el repositorio puede realizar las siguientes acciones: *“Agregar Usuario, Agregar Categoría de OA, Agregar OA, Eliminar Usuario, Eliminar Categoría de OA, Eliminar OA, Modificar Usuario, Modificar Categoría de OA, Modificar OA, Cargar Usuario, Auditar comportamiento de un Usuario, Auditar estadísticas de uso de un OA, Cargar Categoría de OA, Buscar OA, Visualizar el contenido de un OA, Agregar Configuración, Modificar Configuración y Cargar Configuración”* (Peláez y Reina, 2011).

El repositorio Riboa presenta características importantes aunque se puede observar que solamente contiene funcionalidades básicas en cuanto al manejo y gestión tanto de OA como de usuarios. Es válido señalar que ofrece reportes sobre usuarios y objetos de aprendizaje que permiten monitorear el estado de trabajo de los mismos. Además permite realizar un trabajo sobre las configuraciones de la aplicación, aspecto fundamental a asegurar dentro de la administración de los repositorios.

En la UCI existe el Repositorio de Objetos de Aprendizaje RHODA actualmente desarrollado hasta su versión 2.0 que cuenta con un sistema de administración que recoge dentro de un panel las funcionalidades específicas del administrador, estas se encuentran organizadas en 4 grupos:

Gestión de usuarios: se encarga de todas las acciones correspondientes al manejo de usuarios pertenecientes al sistema. Contiene a las funcionalidades Crear usuario, Gestionar usuario, Gestionar cuentas bloqueadas y Gestionar roles.

Gestión de objetos de aprendizaje: permite el correcto manejo y monitorización de los OA que son almacenados asegurando la fiabilidad, integridad y veracidad de sus contenidos. Ofrece la opción

¹⁶ Repositorio de objetos de aprendizaje de la Universidad de Oriente.

Gestión de categoría/subcategoría, Definir metadatos para la búsqueda y Tamaño de los objetos de aprendizaje.

Gestión de la configuración: agrupa a todas las acciones que permiten la configuración del sistema, permitiendo el control sobre innumerables aspectos definidos por la aplicación y que son esenciales para su buen funcionamiento. Engloba a Editar página de inicio, Captcha, Expiración de la sesión, Expiración de la contraseña, Tipo de autenticación y Protocolo seguro HTTPs.

Gestión de la seguridad: asegura que la aplicación no sea atacada mediante virus u otro tipo de ataque, además se encarga de toda la recopilación y monitorización de las informaciones generadas a partir de la acción de los usuarios dentro del repositorio. Permite conocer los Usuarios en línea, Reportes, Usuarios inactivos y realizar Copias de seguridad.

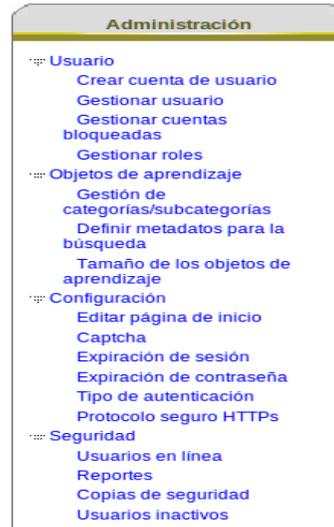


Figura 7. Panel de administración de RHODA 2.0.

Este panel de administración ofrece funcionalidades muy importantes para la gestión de todas las esferas, pero aún es muy pobre en cuanto al control y la configuración. Es necesario potenciar y ampliar las funcionalidades de manejo de usuarios, de la gestión de OA, del control que tiene sobre la aplicación para tener una mejor seguridad y sobre todo ampliar la gama de configuraciones, que permitan al administrador del sistema definir qué hacer o no y cómo se hace.

Los tipos de gestión de la administración y estructuras de los paneles son muy diversos. Tener un panel de administración independiente o que sea visualmente cómodo favorece y facilita el trabajo del administrador. Los reportes dentro de la administración son sumamente importantes para el control y monitoreo dentro de la aplicación, además ofrecen estadísticas y datos necesarios para saber cómo

está siendo utilizado el repositorio o sistema web. Incluir dentro de la gestión de usuarios para administrar la aplicación las restricciones específicas para un usuario o grupo de ellos, ofrece la oportunidad de definir el comportamiento acorde a las necesidades e importancia de los usuarios para la gestión de OA. La configuración de la aplicación, el acceso y definición de los logs desde el panel de administración es muy importante porque facilita el trabajo y logra ofrecer al administrador la posibilidad de configurar todos los aspectos necesarios sin acceder directamente al código fuente.

1.3 Framework de desarrollo Symfony.

Un framework web es un: *“conjunto de componentes (por ejemplo clases en java y descriptores y archivos de configuración en XML) que componen un diseño reutilizable que facilita y agiliza el desarrollo de sistemas Web.”* (Gutiérrez, 2008). Entre los frameworks más utilizados en el mundo para el desarrollo de aplicaciones web se encuentran: CakePhp¹⁷, CodeIgniter¹⁸, Prado¹⁹, Seagull²⁰, ZendFramework²¹, Symfony²².

Características de Symfony:

El framework Symfony para el lenguaje PHP hace del desarrollo web una tarea de pura creación. Sus principales características son (Eguiluz, 2011):

- 1.Fácil de instalar y configurar en sistemas Windows, Mac y Linux.
- 2.Funciona con todas las bases de datos comunes (MySQL, PostgreSQL, SQLite, Oracle, MS SQL Server).
- 3.Compatible solamente con PHP 5 desde hace años, para asegurar el mayor rendimiento y acceso a las características más avanzadas de PHP.
- 4.Basado en la premisa de "convenir en vez de configurar", en la que el desarrollador solo debe configurar aquello que no es convencional.
- 5.Preparado para aplicaciones empresariales, ya que se puede adaptar con facilidad a las políticas y arquitecturas propias de cada empresa u organización.
- 6.Flexible hasta cualquier límite y extensible mediante un completo mecanismo de plugins.

17 Disponible en: <http://cakephp.org/>

18 Disponible en: <http://codeigniter.com/>

19 Disponible en: <http://www.xisc.com/>

20 Disponible en: <http://seagullproject.org/>

21 Disponible en: <http://framework.zend.com/>

22 Disponible en: <http://www.symfony-project.org/>

7.Publicado bajo licencia MIT de software libre y apoyado por una empresa comprometida con su desarrollo.

8.Traducido a más de 40 idiomas y fácilmente traducible a cualquier otro idioma.

9.Utiliza el patrón Modelo-Vista-Controlador (MVC).

Patrón Modelo-Vista-Controlador

Symfony se basa fundamentalmente en este patrón para definir sus tres niveles (Potencier y Zaninotto, 2008):

El Modelo: representa la información con la que trabaja la aplicación, es decir, su lógica de negocio.

La Vista: transforma el modelo en una página web que permite al usuario interactuar con ella.

El Controlador: se encarga de procesar las interacciones del usuario y realiza los cambios apropiados en el modelo o en la vista.

Al aplicar en Symfony el patrón MVC se agrupan en los niveles definidos (**Ver anexo 1**) los componentes: Abstracción de la base de datos, Acceso a los datos, Vista, Plantilla, Layout, Controlador frontal, Acción. Estos componentes son necesarios para el desarrollo de una aplicación, quedando esta última estructurada de la siguiente forma:

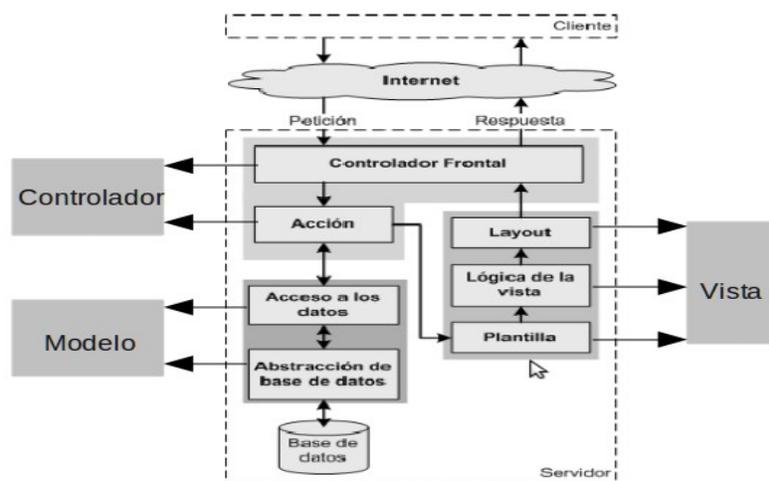


Figura 8. Esquema interno de una aplicación en Symfony (Potencier y Zaninotto, 2008).

1.4 Tecnologías.

1.4.1 JavaScript.

En la construcción de páginas web se utiliza el lenguaje de programación JavaScript, éste permite interactuar con los navegadores de manera dinámica y eficaz. El mismo se define como *“un lenguaje de tipo script compacto, basado en objetos y guiado por eventos, diseñado específicamente para el desarrollo de aplicaciones cliente-servidor dentro del ámbito de Internet”*. (Cárdenas y García, 2003).

Existen diferentes versiones de JavaScript, las cuales han sido integradas en un estándar denominado ECMAScript-262. Dicho estándar fue adoptado y realizado en junio de 1997 por la organización ECMA²³ (European Computer Manufacturers Association), la misma se dedica a la estandarización de información y sistemas de comunicación. Este lenguaje se utiliza principalmente del lado del cliente y es implementado en un navegador web, permitiendo mejoras en la interfaz de usuario y en páginas web dinámicas, aunque también existe una forma de JavaScript del lado del servidor. Los programas escritos con dicho lenguaje se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios.

1.4.2 HTML.

HTML ²⁴(**HyperText Markup Language**): *“... es un lenguaje muy sencillo que permite describir hipertexto, es decir, texto presentado de forma estructurada y agradable, con enlaces (hyperlinks) que conducen a otros documentos o fuentes de información relacionadas, y con inserciones multimedia (gráficos, sonido...)”* (Martínez, 1995). Este lenguaje especifica la estructura lógica en el texto del contenido, como títulos, párrafos de texto, enumeraciones, definiciones, citas, etc; además de los diferentes efectos que se quieren dar, por lo que se especifican los lugares del documento donde se debe poner cursiva, negrita, o un gráfico determinado; finalmente la presentación de dicho hipertexto se realiza por un programa especializado.

HTML se ha convertido en uno de los lenguajes más utilizados para el desarrollo de documentos y es uno de los más sencillos de aprender. Existen además para la realización de sitios web o edición de código HTML programas conocidos como editores WYSIWYG²⁵. Este lenguaje utiliza estilos CSS para la presentación de un documento.

23 Disponible en: <http://www.ecma-international.org/>

24 Disponible en: <http://www-app.etsit.upm.es> se puede descargar en: <http://download.html.it/>

25 Disponible en: <http://www.softonic.com/s/editor-wysiwyg>

1.4.3 CSS.

CSS (Cascading Style Sheets) se define como “...un lenguaje formal usado para definir la presentación de un documento estructurado escrito en HTML.” (Sensagent²⁶, 1989). Básicamente CSS separa la estructura de un documento de su presentación. Al utilizarlo es posible controlar el texto completo de un sitio web y agilizar de forma considerable su actualización. Estos estilos aumentan considerablemente la accesibilidad, pues es favorable a personas con poca visibilidad, así se puede configurar la hoja de estilo para aumentar el tamaño del texto.

1.4.4 eXist XML.

El eXist²⁷ “...es un sistema de gestión de bases de datos libre y de código abierto que almacena datos XML²⁸ de acuerdo a un modelo de datos XML y funciones de procesamiento eficiente, basado en índices de XQuery²⁹”. (eXist, 2011). Entre los aspectos que identifican a dicho sistema se puede mencionar la ejecución del soporte para lenguajes de consultas XML como XQuery, la indexación de documentos y el soporte para la actualización de los datos. En la actualidad cumple en un 99.4% el estándar XQuery. La base de datos eXist proporciona un entorno para la implementación de aplicaciones web, estas se pueden escribir en XQuery, XHTML, CSS y Javascript.

1.4.5 PostgreSQL.

PostgreSQL³⁰ “...es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD³¹ y con su código fuente disponible libremente. Es el sistema de gestión de bases de datos de código abierto más potente del mercado y en sus últimas versiones no tiene nada que envidiarle a otras bases de datos comerciales”. (Martínez, 2010).

Dicho sistema utiliza un modelo cliente/servidor y multiprocesos para asegurar su estabilidad. De este modo, si ocurre algún fallo en uno de los procesos no se afectará al resto y el sistema continuará funcionando. PostgreSQL es considerado como una de las bases de datos más potentes y robustas del mercado debido a que posee características únicas como el tiempo, la estabilidad, la potencia, la robustez y la facilidad de administración e implementación de estándares.

26 Diccionario y traductor en línea. Disponible en: <http://diccionario.sensagent.com/>

27 Disponible en: <http://exist.sourceforge.net/>

28 Lenguaje de Etiquetado Extensible.

29 Lenguaje para consultar datos XML.

30 Disponible en: <http://www.postgresql.org/> se puede descargar en: <http://www.postgresql.org/download/>

31 Distribución de software Berkeley.

1.4.6 NetBeans IDE.

NetBeans IDE³² “...es una aplicación de código abierto pensado para escribir, compilar, depurar y ejecutar programas para aplicaciones programadas...” (Dacosta, 2011). NetBeans permite el desarrollo de aplicaciones multiplataforma de forma muy sencilla. Permite además la creación y el soporte de script PHP. Es un producto libre y gratuito sin restricciones de uso.

1.4.7 UML.

Unified Modeling Language (UML) que en español significa Lenguaje Unificado de Modelado es definido como: “un estándar Object Management Group (OMG³³) diseñado para visualizar, especificar, construir y documentar software orientado a objetos” (Booch, Rumbaugh y Jacobson, 2000). UML define 16 tipos de diagramas (Ver anexo 2) que se utilizan para modelar sistemas de software, lo que permite visualizar y documentar todo el desarrollo de software. Además decir que tiene una estrecha relación con Visual Paradigm que utiliza UML como lenguaje de modelado.

1.4.8 Servidor Apache.

Apache es un servidor web HTTP muy utilizado en el mundo debido a que es sumamente configurable, permite instalar un servidor HTTP de forma rápida y sencilla y mostrar páginas web dinámicas. Es de código abierto y funciona perfectamente en plataformas como Windows y Linux. Es un servidor que cuenta con mucha documentación acerca de su configuración y funcionamiento.

1.4.9 Librería ExtJS.

Es una librería Javascript que permite: “... crear interfaces de usuario en aplicaciones web.” (Zammeti, 2009). Ofrece la oportunidad de imitar las interfaces de programas de escritorios, así como la mayoría de las funcionalidades que ofrecen las aplicaciones en general. Además permite la creación de aplicaciones de internet sofisticadas (RIA³⁴).

1.5 Visual Paradigm for UML.

La herramienta Visual Paradigm está categorizada como una herramienta Computer Aided Software Engineering (CASE) que utiliza UML como lenguaje de modelación. La utilización de esta herramienta ofrece muchas ventajas, lo que provoca que se desarrolle la aplicación de forma rápida y con mucha

32 Disponible en: <http://www.dacostabalboa.com> se puede descargar en: <http://netbeans.org/>

33 Object Management Group. Consorcio dedicado al cuidado y el establecimiento de diversos estándares.

34 Rich Internet Applications.

calidad porque permite diseñar todos los diagramas de clases necesarios. Permite obtener mediante sus diagramas el código correspondiente. Genera además informes para la obtención de documentación de forma automática.

Esta herramienta ofrece soporte del lenguaje UML hasta su versión 2.1 y mantiene una estrecha relación con la metodología de desarrollo Rational Unified Process (RUP). Funciona en óptimas condiciones en sistemas operativos como: Windows, Mac OS X y Linux.

1.6 Modelo CMMI y Metodología para el desarrollo: RUP.

1.6.1 Modelo CMMI.

Capability Maturity Model Integration (CMMI) que traducido al español significan Integración de Modelos de Madurez de Capacidades es un modelo que se implementa en las empresas u organizaciones con el fin de elevar la calidad del software. Dicho modelo está compuesto por 5 niveles de madurez que se le otorgan a las organizaciones en dependencia del grado de madurez que tienen los procesos que utilizan para desarrollar software.

Niveles de Madurez

Cada nivel de madurez está compuesto por diferentes áreas de trabajo que agrupan a buenas prácticas. Los niveles de madurez definidos por CMMI son (García, J, 2005):

Inicial o Nivel 1 CMMI: este es el nivel donde están todas las empresas que no tienen procesos.

Repetible o Nivel 2 CMMI: quiere decir que el éxito de los resultados obtenidos se pueden repetir. El desarrollo no es opaco y se puede saber el estado del proyecto en todo momento.

Definido o Nivel 3 CMMI: alcanzar este nivel significa que la forma de desarrollar proyectos (gestión e ingeniería) está definida.

Cuantitativamente Gestionado o Nivel 4 CMMI: los proyectos usan objetivos medibles para alcanzar las necesidades de los clientes y la organización. Se usan métricas para gestionar la organización.

Optimizado o Nivel 5 CMMI: los procesos de los proyectos y de la organización están orientados a la mejora de las actividades. Mejoras incrementales e innovadoras de los procesos que mediante métricas son identificadas, evaluadas y puestas en práctica.

La UCI se encuentra enfrascada en el inicio de la implementación del modelo CMMI por lo que se encuentra ubicada en el nivel 1, se están implementando nuevas direcciones de trabajo con el fin de

alcanzar los objetivos trazados por el nivel 2 de madurez y alcanzar esta nueva categoría, lo que la catalogaría como la primera organización en Cuba con este nivel de CMMI.

Nivel 2 de madurez

El nivel dos de madurez definido por CMMI es el más difícil de alcanzar por las organizaciones que implementan este modelo. Para alcanzar el nivel dos de madurez se tiene que implementar un nuevo estilo de trabajo basado en la organización de todas las áreas y procesos.

El nivel dos de madurez tiene 7 áreas de procesos definidas como (Axentia, 2006):

Administración de Requerimientos (REQM): esta área de proceso tiene como propósito mantener bajo control los requerimientos que el producto a desarrollar deberá satisfacer. Las prácticas incluidas aquí apuntan a que los requerimientos no solo estén claramente identificados, sino también que todos los involucrados en el proyecto (el cliente, el equipo de proyecto, etc.) estén de acuerdo en su significado.

Planificación del Proyecto (PP): esta área de proceso tiene como propósito establecer y mantener el plan que será empleado para ejecutar y monitorear el proyecto. El plan se desarrolla sobre la base de los requerimientos administrados por el área REQM.

Monitoreo y Control del Proyecto (PMC): esta área de proceso es complementaria y una consecuencia de Planificación del Proyecto (PP). Su propósito es monitorear la ejecución del proyecto empleando para ello el plan y gestionar acciones correctivas en el caso de detectarse desvíos.

Medición y Análisis (MA): esta área de proceso apunta, justamente, a desarrollar y mantener capacidades de medición que permitan satisfacer las necesidades de información de la organización.

Aseguramiento de la Calidad de Productos y Procesos (PPQA): una vez establecidos procesos y estándares será necesario evaluar su aplicación. El objetivo de esta área es justamente ese: proveer una evaluación objetiva de los procesos y de los artefactos producidos.

Administración de la Configuración (CM): esta área de proceso tiene como propósito mantener la integridad de todos los artefactos (entregables o no) producidos por el proyecto.

Administración de Acuerdos con Proveedores (SAM): esta área de proceso apunta a resolver otro de los problemas habituales en muchas organizaciones: el de la tercerización.

1.6.2 RUP

Rational Unified Process (RUP) es definido como: *“...un proceso para el desarrollo de un proyecto de un software que define claramente quién, cómo, cuándo y qué debe hacerse en el proyecto”* (Gómez y Galves, 2007). RUP persigue que se logre a lo largo del ciclo de vida del software que los procesos de este se adapten a las distintas características de la organización teniendo en cuenta el tamaño de esta, la capacidad del personal, el alcance del proyecto y las condiciones en que se desarrollaran las tareas. RUP está estructurado en 4 fases que pueden ser iteradas en dependencia de la necesidad que se defina. Las fases de RUP son (Gómez y Galves, 2007): Inicio, Elaboración, Construcción, Transición (**Ver anexo 3**).

La fase de construcción abarca los flujos de implementación y prueba. Durante la implementación se tienen como objetivos fundamentales implementar todas las funcionalidades del software, con el fin de tener en su culminación un sistema ejecutable que posteriormente será revisado durante la realización de las pruebas. Estas son las responsables de asegurar que el producto cumpla con todos los requisitos identificados en fases anteriores, esto asegura que el cliente quedará satisfecho.

1.6.3 CMMI-RUP.

Para hablar de la vinculación de CMMI con RUP es necesario definir en qué consiste cada uno de ellos. RUP no es más que una metodología para el desarrollo de software, mientras que CMMI es un modelo que va a clasificar a las organizaciones en 5 niveles de madurez. CMMI tiene como característica que define las metodologías, porque tiene incluidas las prácticas necesarias para el proceso de desarrollo de software. RUP provee las herramientas necesarias para la implementación del modelo CMMI, lo que asegura la correcta implementación de este modelo. CMMI describe a lo largo de sus 5 niveles objetivos y prácticas tanto genéricas como específicas que están enfocadas en el **“qué”** hacer, mientras que RUP define el **“cómo”** hacerlo.

De las 7 áreas de procesos que plantea CMMI en el nivel dos de madurez solo están definidas claramente dentro de la metodología RUP las áreas **REQM, PP, PMC, CM**, mientras que **SAM** no es considerado dentro de RUP, **MA** no se describe de forma concisa dentro de esta metodología y en **PPQA** se define el control de la calidad del producto pero se obvia el control de la calidad del proceso. CMMI y RUP tienen muchas características que son comunes, por lo que se hace más fácil su integración y los elementos diferentes que tienen asociados hacen factible la implementación de ambos en una organización.

Conclusiones.

En este capítulo se abordaron temas relacionados con los Repositorios de Objetos de Aprendizaje. Se realizó un estudio de algunos sistemas web y repositorios, que arrojó como resultado características novedosas para la administración del repositorio RHODA. Entre los principales elementos sobresalen las políticas restrictivas para los usuarios, la necesidad de una buena estructura del panel de administración, la configuración desde el ROA y el sistema de reportes como herramienta de control de las acciones realizadas en el repositorio. Se analizaron las herramientas definidas para el desarrollo de la nueva versión de RHODA, donde se determinó que cumplen con los parámetros necesarios para satisfacer las necesidades del proyecto. Se definió que RUP como metodología no cubre todos las áreas de procesos existentes en el desarrollo de la aplicación, por lo que se decidió solaparla con el modelo CMMI que cubre completamente aspectos que esta metodología no describe; como por ejemplo el control de la calidad del proceso.

Capítulo 2. Implementación y prueba.

Introducción.

En este capítulo se abordarán los flujos de trabajo de implementación y prueba. RUP plantea que el flujo de implementación tiene como principal objetivo el desarrollo de las funcionalidades de la aplicación que posteriormente serán revisadas durante el flujo de prueba. Se generarán los artefactos de los flujos de implementación y prueba definidos por RUP, regidos estrictamente por las características obtenidas del flujo de análisis y diseño.

2.1 Funcionalidades.

Se detallan los requisitos funcionales añadidos al módulo de Administración enfatizando en sus principales características, que definen sus objetivos y alcance dentro de la aplicación.

2.1.1 Administrador de OA.

El administrador de OA es la unión de 6 funcionalidades. Estas funcionalidades se muestran en una interfaz común (*figura 9*) donde ocupa toda la página y no se muestran los módulos laterales. Esta vinculación le permite al administrador tener en sus manos una herramienta que le posibilita:

- Filtrar OA por sus atributos:** consiste en el filtrado de OA teniendo en cuenta 6 criterios de búsqueda como son: por fecha de creación, fecha de la última modificación, título del OA, autor, categoría en la que se encuentra ubicado, subcategoría asociada a él y el estado en que se encuentra, ya sea en edición, revisión o publicado.
- Listar OA para administrar:** al acceder al administrador de OA se muestra de forma automática una lista de datos de los OA que se encuentran en el sistema. Dicha lista se actualiza en dependencia de los parámetros que se seleccionen al filtrar los OA por sus atributos y se disponga a buscar en el sistema. Los datos que se muestran en el listar son: título del OA, autor, categoría en la que se encuentra, subcategoría, y estado del mismo.
- Ordenar los OA por sus atributos:** permite ordenar los OA listados teniendo en cuenta sus atributos, mencionados en la funcionalidad anterior. Se permite ordenarlos de forma ascendente y descendente, lo que le facilita al administrador el trabajo con los mismos.

•**Administrar propiedades de OA:** le permite al administrador manejar el OA de acuerdo a 5 opciones: editar, propiedades, eliminar, visualizar y descargar, estas opciones se muestran de forma independiente para cada OA listado para administrar.

•**Eliminar OA:** ofrece al administrador la opción de manejo de la eliminación de múltiples OA seleccionados.

•**Editar OA:** otorga al administrador la opción de editar de forma múltiple los OA seleccionados de la lista, solo es posible cambiar en esta edición la categoría, subcategoría y el estado.

Administrador de objeto de aprendizaje

Fecha de creación

Desde: ... ?

Hasta: ... ?

Fecha de modificación

Desde: ... ?

Hasta: ... ?

Título: ?

Autor: ?

Categoría: ?

Subcategoría: ?

Estado: ?

<input type="checkbox"/>	Título del objeto de aprendizaje	Autor	Categoría/subcategoría	Estado	Editar	Propiedades	Eliminar	Visualizar	Descargar
<input type="checkbox"/>	Sin título.	Yanelis Lara	Anónimo	Edición	Editar	Propiedades	Eliminar	Visualizar	Descargar
<input type="checkbox"/>	Elementos Básicos del Modelo Entidad Relación	Yanelis Lara	Anónimo	Revisión	Editar	Propiedades	Eliminar	Visualizar	Descargar

? ?

Figura 10. Administrador de objetos de aprendizaje.

2.1.2 Consultar historial de revisor.

Permite al administrador consultar todas las revisiones de OA que han sido realizadas por los revisores. Cuando se accede a esta funcionalidad se listan de manera automática todas las revisiones existentes en el sistema, mostrando: usuario que realizó la revisión, título del OA revisado, autor del OA, fecha de revisión, categoría y subcategoría a la que está asociado y la acción que fue ejecutada sobre él. Todos estos campos pueden ser ordenados atendiendo a dos criterios: ascendente y descendente. Una vez listadas todas las revisiones el administrador puede filtrar los resultados atendiendo a 3 criterios: usuario que realizó la revisión, categoría y subcategoría a la que está asociada la revisión. (figura 10).

Historial de revisor

Usuario:

Categoría:

Subcategoría:

4 revisiones

Usuario	Título del objeto de aprendizaje	Autor	Fecha de revisión	Categoría/Subcategoría	Acción
yaneilis	Estrategias para la elaboración de las pruebas al soporte realizado en la comunidad	Yanelis Lara	24/03/2011 21:53:00	Anónimo	Aprobado

Figura 11. Historial de revisor.

2.1.3 Editar configuración de la conexión a la base de datos postgresql.

Ofrece al administrador la opción de configurar la conexión a la base de datos postgresql (figura 11). Al acceder el administrador a la aplicación se le muestran 4 campos para la nueva configuración: servidor en el que se encuentra la base de datos, nombre de la base de datos a utilizar, usuario de acceso, y la contraseña correspondiente al usuario especificado. Al llenar estos datos puede escoger la opción de guardar la nueva configuración o cancelar la misma.

Configurar conexión de base de datos Postgres

Servidor:* ?

Base de datos:* ?

Usuario:* ?

Contraseña:* ?

(*)Campos obligatorios

Figura 12. Configurar conexión de base de datos postgresql.

2.1.4 Editar configuración de la conexión a la base de datos nativa XML eXist.

El administrador tiene la opción de configurar la conexión a la base de datos nativa XML eXist (figura 12). Al acceder a la aplicación se le muestran 4 campos que permiten la nueva configuración: servidor en el que se encuentra la base de datos, nombre de la base de datos a utilizar, usuario de acceso, y la contraseña correspondiente al usuario especificado. Al llenar estos datos puede escoger la opción de guardar la configuración o cancelar la misma. La nueva configuración se guardará en el archivo app.yml para poder efectuar la conexión a la base de datos.

Configurar conexión de base de datos xml exist

Servidor:* ?

Base de datos:* ?

Usuario:* ?

Contraseña:* ?

(*)Campos obligatorios

Figura 13. Configurar conexión de base de datos nativa XML eXist.

2.1.5 Configurar página de mantenimiento.

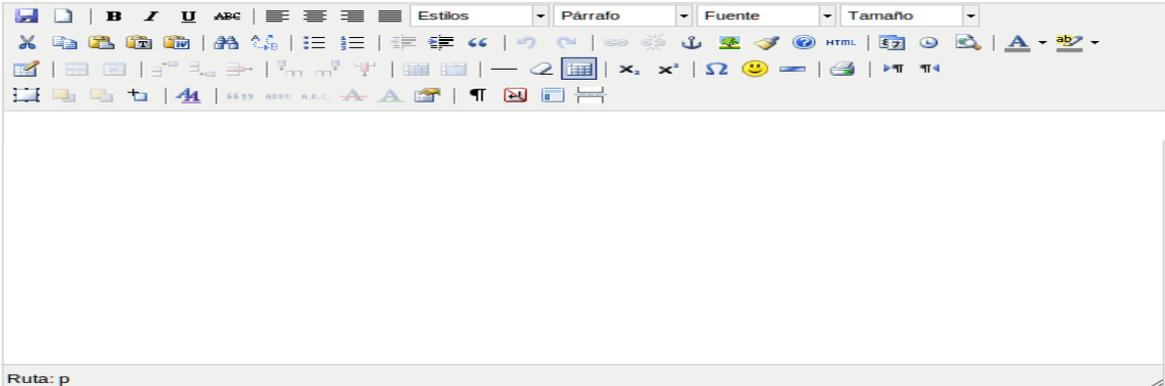
Permite al administrador mediante un editor HTML configurar la página que se mostrará cuando el repositorio se encuentre en mantenimiento por alguna razón. En este editor HTML se puede cargar todo tipo de media. Ofrece la opción de generar la página, o sea ponerla en funcionamiento y la opción de deshabilitarla cuando terminen las labores de mantenimiento en la aplicación.

Configurar página de mantenimiento

Rango de fechas:

Desde: ... ?

Hasta: ... ?



Ruta: p

Figura 14. Configurar página de mantenimiento.

2.1.6 Configuración de correo electrónico SMTP.

El administrador configura el servidor de correo electrónico SMTP introduciendo los datos necesarios para establecer la conexión. Los campos de datos a llenar son: el servidor de correo SMTP, el puerto que va a utilizar para dicha conexión, el usuario y contraseña válida y el protocolo que se utilizará durante el intercambio; este último dato es el único con carácter opcional.

Configurar correo SMTP

Servidor SMTP:* ?

Puerto:* ?

Usuario:* ?

Contraseña:* ?

Protocolo a utilizar: ?

(*)Campos obligatorios

Figura 15. Configurar correo electrónico SMTP.

2.1.7 Configurar tipo de autenticación.

El administrador al configurar el tipo de registro puede seleccionar opciones como: el registro manual que permite al invitado registrarse sin ningún tipo de verificación y el registro por correo electrónico que permite al invitado registrarse, una vez que lo ha hecho el sistema envía un correo al mismo para que confirme su registro. Las opciones de registro manual y registro por correo electrónico son excluyentes porque solo se puede activar uno de estos dos servicios, nunca los dos al mismo tiempo. Independientemente de la opción antes seleccionada se puede utilizar un servicio LDAP que permite al usuario registrarse mediante el usuario y clave de un directorio activo existente.

Configurar tipo de autenticación

Autoregistro.
Permite al invitado registrarse sin necesidad de alguna verificación. No se recomienda este método por razones de seguridad.

Autenticación basada en email.
Permite al invitado registrarse y una vez que lo ha hecho el sistema envía un correo al mismo para que confirme su registro. Este método limita en gran medida la cantidad de usuarios que pueda tener una misma persona en el sistema.

Usar servidor LDAP.
Permite al usuario registrarse mediante el usuario y clave de un directorio activo existente. Este método garantiza que cada usuario de un dominio determinado tenga un solo usuario en el sistema.

Figura 16. Seleccionar tipo de autenticación.

La configuración de los ajustes del servidor para consumir un servicio LDAP se realiza en la aplicación teniendo en cuenta 11 parámetros: URL del host, Servidor, Versión, Codificación LDAP, Ocultar contraseña, Usuario, Contraseña, Contexto, Buscar en subcontextos, Atributos y Datos.

Usar servidor LDAP.
Permite al usuario registrarse mediante el usuario y clave de un directorio activo existente. Este método garantiza que cada usuario de un dominio determinado tenga un solo usuario en el sistema.

Ajustes del servidor.

URL del host:* ?

Servidor:* ?

Versión: ?

Codificación LDAP:* ?

Ocultar contraseña: ?

Usuario:* ?

Contraseña:* ?

Contexto:* ?

Buscar en subcontextos: ?

Atributos:* ?

Datos:* ?

(*)Campos obligatorios

Figura 17. Configuración de ajustes del servidor para servicio LDAP.

2.1.8 Configurar tamaño máximo de los OA.

Se le ofrece al administrador la opción de definir el tamaño máximo que tendrán todos los usuarios por defecto para crear los OA en el sistema, además de limitar la capacidad total que se destinará al almacenamiento de los OA en el servidor. Se pueden configurar además cuotas que definen el tamaño para un usuario específico.

Configurar tamaño máximo de los objetos de aprendizaje.

La capacidad del servidor destinada a los objetos de aprendizaje es de: 100MB

El tamaño máximo actual para un objeto de aprendizaje es de: 1MB

Nueva capacidad de almacenamiento: KB 

Nuevo tamaño de OA: KB 



Figura 18. Configuración de tamaño y almacenamiento de OA.

Las cuotas serán establecidas por el administrador para los usuarios que presenten algún tipo de inestabilidad o necesiten un mayor tamaño para los OA teniendo en cuenta la necesidad de su trabajo. Para ello se introduce el usuario y la cantidad de KB, MB o GB que se le deseen otorgar. Estas cuotas podrán ser modificadas y/o eliminadas independiente y múltiplemente.

Administrador de cuotas de objeto de aprendizaje

 [Adicionar nueva cuota](#)

Usuario:*  Tamaño:* KB 

(*)Campos obligatorios

<input type="checkbox"/>	Usuario 	Nombre 	Tamaño 	Editar	Eliminar
<input type="checkbox"/>	sbello	admin admin	1 MB	Editar	Eliminar

Figura 19. Administración de cuotas de tamaño de OA.

2.1.9 Listar comprobaciones del servidor.

Se le muestra al administrador de forma gráfica los datos de rendimiento del servidor. Los datos ofrecidos son: el porcentaje de uso del CPU, de memoria utilizada, de almacenamiento destinado a los OA utilizado y la concurrencia, todos estos datos se dan en porcentaje en una escala de 0 a 100.

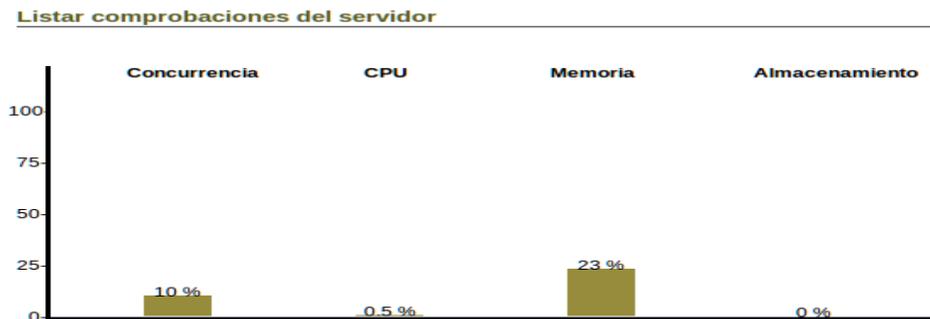


Figura 20. Listar comprobaciones del servidor.

2.2 Requisitos no funcionales.

Los requisitos no funcionales son un eslabón fundamental a tener en cuenta porque aseguran parámetros que son de vital importancia para la interacción de la herramienta con el medio y del usuario final con la aplicación.

2.2.1 Accesibilidad.

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.

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.

2.3 Diagrama de casos de uso.

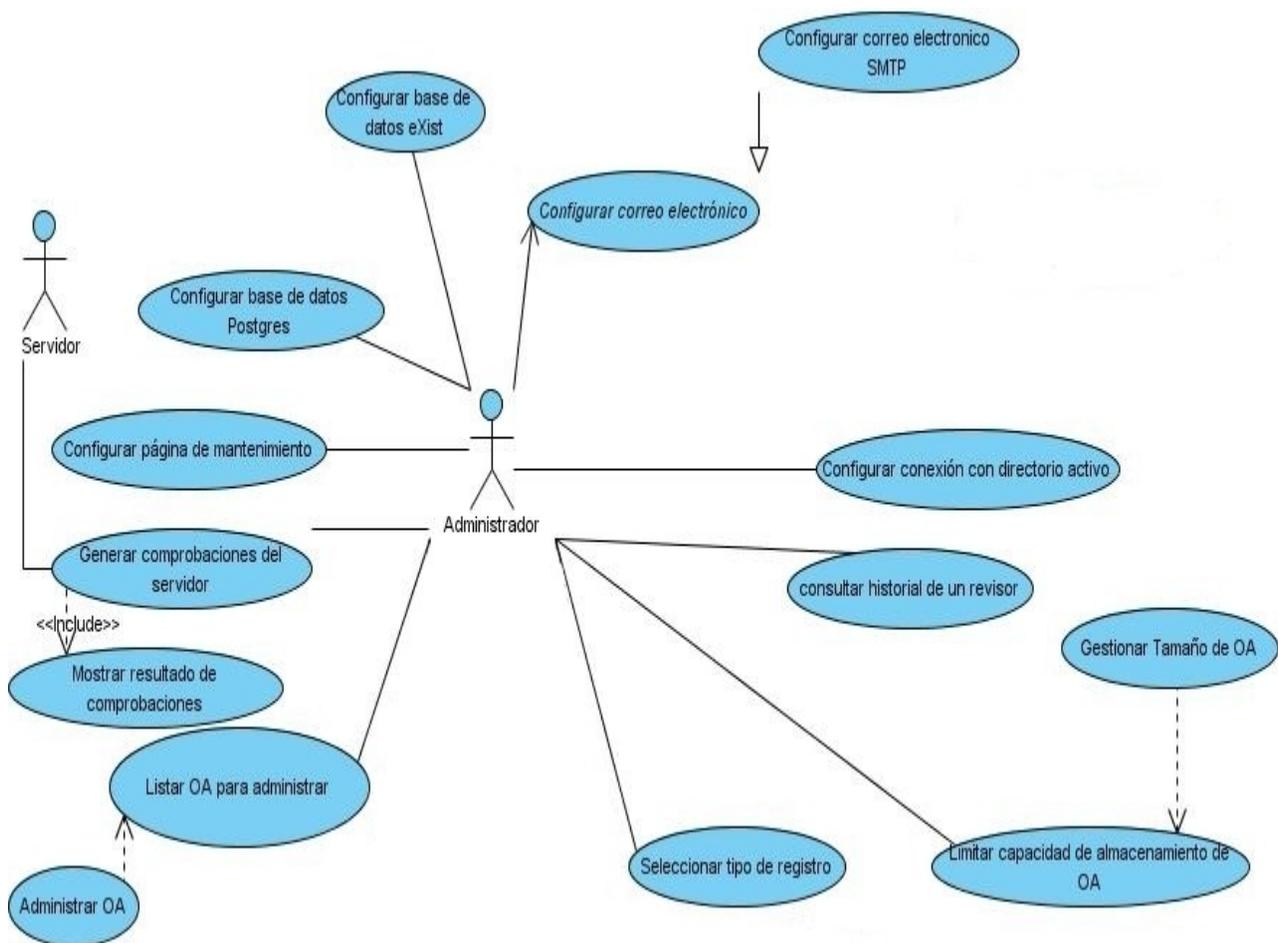


Figura 9. Diagrama de casos de uso (Rodríguez, 2011).

2.4 Diagrama de Componentes.

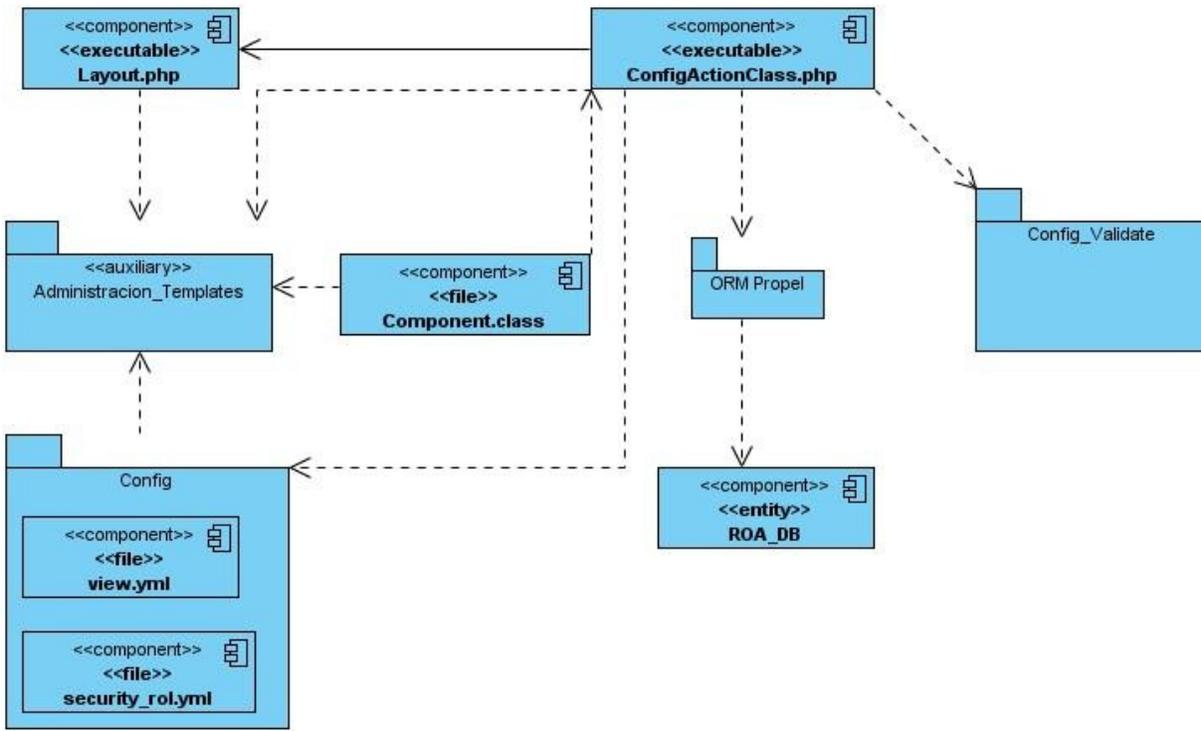


Figura 10. Diagrama de componentes.

2.5 Diagrama de despliegue.

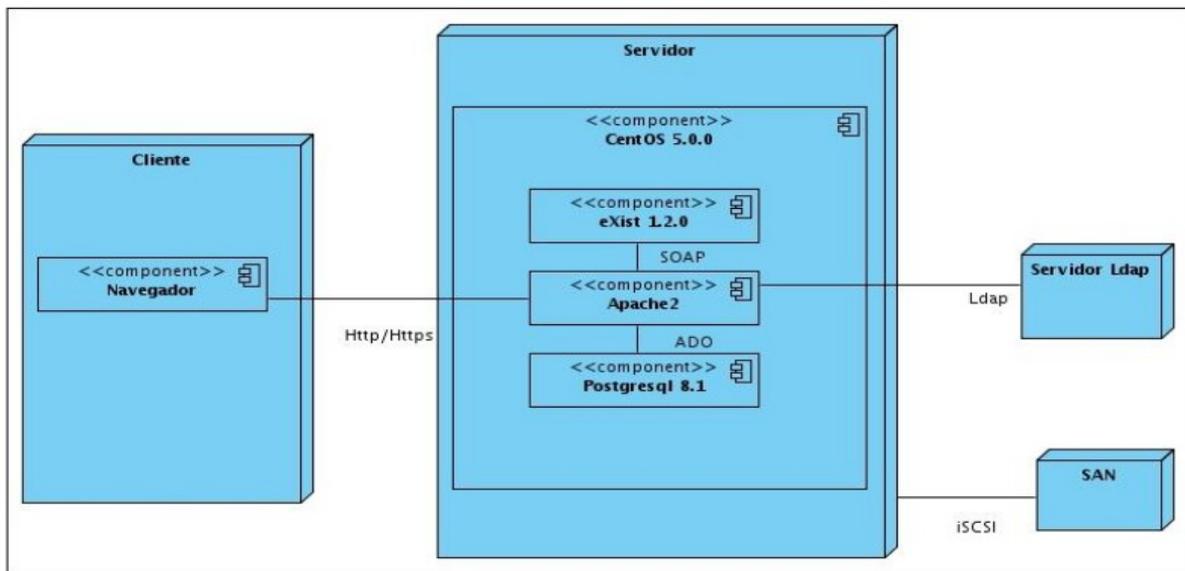


Figura 11. Diagrama de despliegue.

2.6 Validación de la solución propuesta.

La palabra probar tiene su origen del latín “*probāre*”, es muy utilizada en el mundo a la hora de tener que medir las cualidades de algo o alguien. La acción y efecto de probar se conoce como prueba, que se define como la: “*Razón, argumento, instrumento u otro medio con que se pretende mostrar y hacer patente la verdad o falsedad de algo. Ensayo o experimento que se hace de algo, para saber cómo resultará en su forma definitiva.*” (RAE, 2010).

Con el avance de las tecnologías y el surgimiento de la informática y las comunicaciones se hizo necesario asegurar la calidad de todos los productos que se generan a partir de aplicar el conocimiento adquirido. Para asegurar la calidad de un programa surgen las pruebas de software definidas como: “*Un elemento crítico para la garantía de la calidad del software y representa una revisión final de las especificaciones, del diseño y de la codificación*” (Pressman, 2002).

Las pruebas se pueden realizar a lo largo de todo el ciclo de vida del software con lo que se garantiza que todos los artefactos generados cumplan con todas las especificaciones pertinentes. Las pruebas realizadas no pueden estar enfocadas en demostrar que el software carece de errores, todo lo contrario deben sacar a la luz todo los errores posibles que tenga la aplicación; por estas razones se considera que una prueba de software es exitosa cuando se encuentran mediante ella errores inesperados. La metodología RUP dentro de sus flujos de trabajo cuenta con el flujo de prueba que tiene como principal objetivo asegurar la validez y calidad del producto. Durante el proceso de probar el software se somete al mismo a métodos como el de Caja Blanca y Caja Negra asegurando así cubrir todos los errores posibles.

2.6.1 Pruebas de Caja Blanca.

Las pruebas de Caja Blanca también conocidas como pruebas de Caja Transparentes o Cristal se definen como: “*...un método de diseño de casos de prueba que usa la estructura de control del diseño procedimental para obtener los casos de prueba.*” (Pressman, 2002). Estas pruebas son muy importantes para determinar errores internos en la estructura y lógica del código generados a partir de la programación. Al diseñar o programar un código se tienen en cuenta las condiciones de ejecución habituales, no siendo así para los casos excepcionales. Al teclear las líneas de código se cometen errores al escribir las sintaxis o las palabras en general, por lo que es muy importante la revisión del código segmento a segmento. Para la realización de estas pruebas se utilizan métodos como (Pressman, 2002):

•**Método de prueba del camino básico:** permite al diseñador de casos de prueba obtener una medida de la complejidad lógica de un diseño procedimental y usar esa medida como guía para la definición de un conjunto básico de caminos de ejecución.

•**Método de prueba de la estructura de control:** utiliza el método de prueba del camino básico que en conjunto con otras variantes de estructuras de control amplían la cobertura de la prueba y mejoran las pruebas de Caja Blanca. Entre estas variantes se encuentran las Pruebas de condición, Pruebas de flujo de datos y las Pruebas de bucles.

La utilización de estos métodos convencionales propician que las pruebas de Caja Blanca sean muy lentas y por lo tanto se demore mucho este proceso. Con el objetivo de reducir el tiempo de prueba y realizar concurrentemente como sea posible la ejecución de las mismas, se utilizan herramientas que aceleren, hagan más factible este proceso y ofrezcan datos que serían imposible de obtener por los métodos convencionales, como por ejemplo el tiempo de respuesta de una petición, el tiempo de ejecución etc.

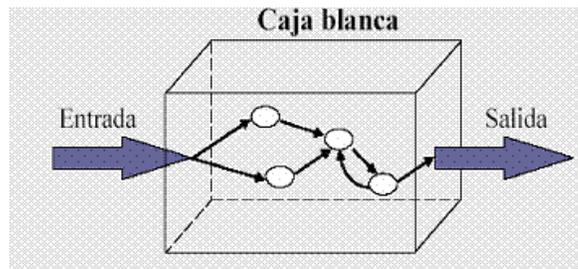


Figura 21. Pruebas de Caja Blanca (Potencier y Zaninotto, 2008).

2.6.1.1 Utilizando Symfony en la Prueba de Caja Blanca.

El framework de desarrollo Symfony incluye el framework Lime para pruebas, este está basado en la librería Test::More de Perl. Este framework se encarga de realizar las pruebas unitarias y las hace más fácil de crear. Lime utiliza para la confección de las pruebas muchos tipos de métodos que ofrecen la seguridad de encontrar errores y mostrar mensajes más específicos sobre estos, además de hacer más legibles las pruebas confeccionadas. Las principales ventajas de su uso son (Potencier y Zaninotto, 2008):

•Ejecuta los archivos de prueba en un entorno independiente para evitar interferencias entre las diferentes pruebas. No todos los frameworks de pruebas garantizan un entorno de ejecución "limpio" para cada prueba.

- Las pruebas de Lime son fáciles de leer y sus resultados también lo son. En los sistemas operativos que lo soportan, los resultados de Lime utilizan diferentes colores para mostrar de forma clara la información más importante.

- Symfony utiliza Lime para sus propias pruebas y su "regression testing", por lo que el código fuente de Symfony incluye muchos ejemplos reales de pruebas unitarias y funcionales.

- El núcleo de Lime se valida mediante pruebas unitarias.

- Está escrito con PHP, es muy rápido y está bien diseñado internamente. Consta únicamente de un archivo, llamado lime.php, y no tiene ninguna dependencia.

2.6.1.2 Pruebas unitarias realizadas.

Las pruebas unitarias utilizando el framework Lime se le realizaron a un total de 10 métodos, estos métodos son los principales que se incorporan al módulo de Administración en la versión 2.1. Se validó que cumplieran con los requisitos especificados, los tipos de datos correctos y se obtuvieran valores satisfactorios para cada posible respuesta a validar. Los métodos probados son:

- ConfigAction

- ConfigAutentification(\$register, \$ldap):Array

- ConfigPif(\$name, \$mb, size, \$mb2):Boolean

- ConfigShare(\$user, \$size, \$mb):Boolean

- ManagerOa():Array

- ConfigMaintenance(\$date1, \$date2, \$hora1, \$hora2):Boolean

- ConfigPostgres(\$server, \$database, \$user, \$password):Boolean

- ConfigExist(\$server, \$database, \$user, \$password):Boolean

- ManagerProperty(\$category, \$subcategory, \$state):Boolean

- ConfigSMTP(\$server, \$port, \$user, \$password):Boolean

- HistoryRevisor():Array

En la realización de las pruebas de estos métodos se confeccionaron las pruebas en los archivos métodoTest.php que se incluyeron en las carpetas /test/unit. En el caso del método ConfigSMTP se determinaron los 3 juegos de datos o pruebas a realizarle y se incluyeron en el archivo configTest.php y se ejecutaron las pruebas desde la consola mediante el comando php symfony test:unit config.

```

root@PC7-Rhoda: /home/roberto/Escritorio/Roa
Archivo Editar Ver Terminal Ayuda
roberto@PC7-Rhoda:~$ sudo su
root@PC7-Rhoda:/home/roberto# cd Escritorio/Roa
root@PC7-Rhoda:/home/roberto/Escritorio/Roa# php symfony test:unit config
1..3
# config::configSMTP
ok 1 - Este metodo Guarda la configuracion
ok 2 - Esto comprueba que el puerto sea numerico
not ok 3 - Esto comprueba que los campos no esten vacios
# Failed test (./lib/symfony/vendor/limelime/limelime.php at line 80)
# got: 3
# expected: '4'
# Looks like you failed 1 tests of 3.
root@PC7-Rhoda:/home/roberto/Escritorio/Roa#

```

Figura 22. Ejecución de pruebas unitarias con Lime.

2.6.1.3 Resultados pruebas unitarias.

Métodos	Iteraciones	Pruebas	Pruebas Correcta(S)	Pruebas incorrecta(S)
ConfigAutentification	1ra	3	3	0
	2da	3	3	0
ConfigPif	1ra	4	3	1
	2da	4	4	0
ConfigShare	1ra	6	4	2
	2da	6	6	0
ManagerOa	1ra	7	7	0
	2da	7	7	0
ConfigMaintenance	1ra	2	2	0
	2da	2	2	0

ConfigPostgres	1ra	2	0	2
	2da	2	2	0
ConfigExist	1ra	2	2	0
	2da	2	2	0
ManagerProperty	1ra	5	4	1
	2da	5	5	0
ConfigSMTP	1ra	3	2	1
	2da	3	3	0
HistoryRevisor	1ra	6	5	1
	2da	6	6	0

Durante el proceso de prueba a los métodos seleccionados se le realizaron 2 iteraciones. En la primera iteración se detectaron un total de 10 errores en 6 métodos para un total de 40 pruebas o juegos de datos. Se corrigieron los errores encontrados durante el proceso de prueba y se realizó una segunda iteración con un resultado de 40 juegos de datos correctos y ninguna prueba inválida. Durante las iteraciones los casos de prueba permanecieron sin variación para asegurar la erradicación de los errores.

2.6.2 Pruebas de Caja Negra.

Al realizar las pruebas de Caja Negra sobre las aplicaciones no se tiene en cuenta como está estructurado el código fuente, ni se tiene conocimiento de su funcionamiento interno. Están basadas específicamente sobre los requerimientos establecidos y en las funcionalidades del sistema, de allí que su objetivo fundamental sea demostrar la operatividad de dichas funciones. Estas pruebas se realizan directamente sobre la interfaz del software entrando una serie de datos para comprobar el correcto manejo de los mismos y que se obtenga un resultado válido y satisfactorio. La prueba de Caja Negra: *“...examina algunos aspectos del modelo fundamental del sistema sin tener mucho en cuenta la estructura lógica interna del software”* (Pressman, 2002).

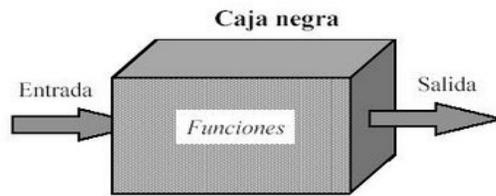


Figura 22. Pruebas de Caja Negra (Montero, 2010).

Para ejecutar las pruebas de Caja Negra o Caja Opaca como también se conoce es necesario utilizar diferentes métodos para obtener los mejores resultados posibles (Pressman, 2002):

- **Método de partición equivalente:** es un método que divide el campo de entrada de un programa en clases de datos de los que se pueden derivar casos de pruebas.

- **Método de análisis de valores límites:** los errores tienden a darse más en los límites del campo de entrada que en el centro. Por ello, se ha desarrollado el análisis de valores límites (AVL) como técnica de prueba. Esta técnica se encarga de evaluar el manejo de los datos en su estado límite por parte del programa.

- **Método de Prueba de la tabla ortogonal:** este método se aplica a problemas en que el dominio de entrada es relativamente pequeño. Es particularmente útil al encontrar errores asociados a fallos localizados.

A lo largo de estas pruebas se intentan encontrar errores que puedan provocar que la aplicación colapse siendo operada desde la interfaz y que se encuentre entre las siguientes categorías (Pressman, 2002):

- Errores de interfaz.
- Errores en estructuras de datos o en accesos a bases de datos externas.
- Errores de rendimiento.
- Errores de inicialización y de terminación.

Durante el flujo de pruebas se aplicó el método de partición equivalente. Para la selección de este método se tuvo en cuenta que está orientado a los requisitos funcionales del software permitiendo que se creen juegos de datos para lograr probar a profundidad todas las funcionalidades del software. Además hay que señalar que son pruebas que gozan de gran aceptación entre los probadores y se encuentran registradas en innumerables bibliografías tanto en formato duro como en Internet. Como se

habló anteriormente, de este método se pueden derivar casos de pruebas que aseguren que se detecten errores genéricos de forma inmediata, que en otras circunstancias serían detectados después de la ejecución de varios casos de prueba. Para el desarrollo de este método se identifican dos aspectos fundamentales:

- Las clases de equivalencia.
- Los diseños de casos de pruebas.

2.6.2.1 Clases de equivalencia.

Las clases de equivalencia van a identificarse como dos conjuntos de datos de entradas, uno de datos válidos y otro de datos inválidos. Estos conjuntos se definen como clases de equivalencia válidas y clases de equivalencia inválidas respectivamente. Para definir estas clases se utilizan las siguientes reglas (Pressman, 2002):

- 1.Si una condición de entrada especifica un rango, se define una clase de equivalencia válida y dos no válidas.
- 2.Si una condición de entrada requiere un valor específico, se define una clase de equivalencia válida y dos no válidas.
- 3.Si una condición de entrada especifica un miembro de un conjunto, se define una clase de equivalencia válida y una no válida.
- 4.Si una condición de entrada es lógica, se define una clase de equivalencia válida y una no válida.

Aplicando estas reglas se determinaron las clases de equivalencia del caso de uso “Configurar correo SMTP”:

Condición externa	Clase de equivalencia válida	Clase de equivalencia no válida
Campo obligatorio	$0 < \text{valor}$	Valor < 0 Valor = 0
Campo de tipo numérico	Valor = 1...9	Valor = A..Z

Tabla 1. Clases de equivalencia del caso de uso “Configurar correo SMTP”.

2.6.2.2 Casos de prueba

Una vez culminadas la determinación de las clases de equivalencia se procede a realizar las descripciones de los casos de pruebas que constituyen la brújula del probador, porque de él depende la eficacia y efectividad de las pruebas. Por esta razón es muy importante que las descripciones sean directas, eficientes y redactadas en un lenguaje claro. Existe una estrecha relación entre los casos de uso y los casos de pruebas, por lo que se tuvo en cuenta las descripciones detalladas de los casos de uso para conformar un caso de prueba para cada uno de ellos. Se analizó en cada caso de uso las posibles entradas de datos y resultados posibles a obtener a partir de las acciones del usuario y teniendo en cuenta las clases de equivalencia obtenidas. Se confeccionó cada caso de prueba con la identificación de los principales escenarios con sus posibles entradas (**Tabla 2**). A partir de lo determinado se procede a confeccionar la matriz de los datos con sus posibles variantes, además de determinar la respuesta del sistema y la incorporación de la respuesta obtenida a partir de la prueba.

Nombre de la sección	Escenario de la sección	Descripción de la funcionalidad
SC 1 Configurar Correo SMTP.	EC 1.1 Configurar correo SMTP	Se muestra una interfaz que brinda la posibilidad de realizar la configuración. Se muestran además los botones "Guardar" y "Cancelar". Se pulsa el botón "Guardar" y se almacenan los datos en el sistema.
	EC 1.2 Datos incompletos	Se muestra un mensaje al lado de cada campo que se encuentre vacío y se señalará en rojo. Se permite que vuelvan a entrar los datos solicitados.
	EC 1.3 Datos incorrectos	Al introducir los datos de forma incorrecta se muestra al lado del campo un mensaje señalando el tipo de error y el campo señalado en rojo.
	EC 1.4 Cancelar la acción	Se presiona el botón Cancelar para evitar guardar los cambios efectuados.

Tabla 2. Escenarios para el caso de uso "Configurar correo SMTP".

Escenario	Servidor SMTP	Puerto	Usuario	Contraseña	Protocolo	Respuesta del Sistema	Resultado de la Prueba
EC 1.1 Configurar correo SMTP	V	V	V	V	V	Se muestra una interfaz que brinda la posibilidad de realizar la configuración. Se muestran además los botones "Guardar" y "Cancelar". Se pulsa el botón "Guardar" y se almacenan los datos en el sistema.	Satisfactorio.
EC 1.2 Datos incompletos	NA	NA	NA	N/A	N/A	Se muestra un mensaje indicando que los campos son obligatorios. Se señala el campo en rojo. Se permite volver a introducir los datos.	Satisfactorio.
	V	N/A	N/A	N/A	N/A		
	N/A	V	N/A	N/A	N/A		
	N/A	N/A	V	N/A	N/A		
	N/A	N/A	N/A	V	N/A		
	N/A	N/A	N/A	N/A	V		
EC 1.3 Datos incorrectos.	N/A	I	N/A	N/A	N/A	Al introducir los datos de forma incorrecta se muestra al lado del campo un mensaje señalando el tipo de error y el campo señalado en rojo.	Satisfactorio.
	N/A	V	N/A	N/A	N/A		
	N/A	I	N/A	N/A	N/A		
EC 1.4 Cancelar la acción.	N/A	N/A	N/A	N/A	N/A	Se presiona el botón Cancelar para evitar guardar los cambios efectuados.	Satisfactorio.

Tabla 3. Escenarios para el caso de uso "Configurar correo SMTP".

Se realizaron las pruebas a todos los casos de uso y arrojaron errores que se documentaron en los casos de prueba y luego se incorporaron al documento de no conformidades. Se realizaron tres iteraciones donde se fueron detectando diferentes tipos de errores y se le fueron dando solución hasta erradicarlos completamente (**Tabla 4**).

Iteración	No conformidades	Cerrada(S)	No procede(N)
1ra	4	4	0
2da	2	2	0
3ra	0	-	-

Tabla 4. Resultados de las iteraciones.

Conclusiones.

En este capítulo se abordaron aspectos de la solución propuesta como el diagrama de caso de uso, el diagrama de componentes y el diagrama de despliegue. Se realizó una detallada descripción de las funcionalidades implementadas dentro del módulo de Administración. Estas funcionalidades tributan directamente a mejorar los procesos existentes en RHODA porque permiten la configuración de elementos como bases de datos, correo y página de mantenimiento que se realizaban de forma manual; además de obtener reportes de los OA y las revisiones de estos realizadas en el sistema. Se realizaron las pruebas pertinentes a la aplicación y se corrigieron las deficiencias encontradas, obteniendo así un producto mucho más seguro y robusto.

Conclusiones generales

Una vez concluida la investigación se determinó que se cumplieron los objetivos trazados, arribando como resultado del cumplimiento de los mismos a las siguientes conclusiones:

- Se determinaron características esenciales como: la importancia de mantener una organizada estructura del panel de administración, la necesidad de incorporar políticas restrictivas, nuevos reportes a incluir en la versión 2.1 de RHODA y las nuevas configuraciones de la plataforma.
- El desarrollo del flujo de implementación y prueba permitió agregar al repositorio funcionalidades de configuración y de obtención de reportes que mejoran los procesos existentes en la herramienta libres de errores.
- Con el desarrollo de este trabajo se aumentó la operatividad de las configuraciones de RHODA desde el panel de administración, permitiendo un mayor control y fácil trabajo en el repositorio.

Recomendaciones

Para dar continuidad al siguiente trabajo luego de cumplirse todos los objetivos propuestos, se recomienda para futuras versiones:

- Implementar nuevas funcionalidades que tributen a un mayor control de la administración del repositorio mediante reportes y configuraciones.
- Incorporar la gestión de temas que permitan la configuración de la apariencia del repositorio.
- Incorporar la configuración del idioma que tribute al módulo de internacionalización.
- Implementar el módulo de Administración de forma independiente de la interfaz de información.

Referencias bibliográficas

ANCED, Asociación Nacional de Centros de e-Learning y Distancia, s.a. Libro de buenas practicas de E-Learning.[En línea].2007.[Citado el: 2 de diciembre de 2010].

<http://www.buenaspracticas-elearning.com/capitulo-16-estandares-e-learning.html>

Axentia. Una introducción a CMMI.2006.[Consultado el: 13 de enero de 2011].

BDBF(Biblioteca Digital de Bioquímica y Farmacia).Proyecto de Mejoramiento de la enseñanza en Farmacia y Bioquímica (PROMFyB). [En línea].10 de septiembre de 2009.[Citado el 6 de diciembre de 2010].

<http://www.bdbf.edu.ar/>

Chiappe, A., Segovia, Y., & Rincon, H. Y. Toward an instructional design model based on learning objects. Educational Technology Research and Development, 55, 671-681. 2007. [Citado el: 2 de diciembre de 2010].

Chiappe Laverde, Andrés. Evolución conceptual de los Objetos de Aprendizajes: Antes de comenzar... un poco de historia. [En línea]. Proceso de publicación.[Citado el: 2 de diciembre de 2010].

<http://www.cudi.edu.mx/diplomadoOA/>

Colectivo de Autores. [En línea].14 de febrero de 2002.[Citado el: 7 de diciembre de 2010].Budapest Open Access Initiative.

<http://www.soros.org/openaccess/read.shtml>

De Volder, Carolina. Los Repositorios de Acceso Abierto en la Argentina SITUACIÓN ACTUAL .[En línea].4 de noviembre de 2008.[Consultado el: 7 de diciembre de 2010].

<http://www.scielo.org.ar/pdf/ics/n19/n19a05.pdf>

Domínguez, José Alberto. UML: Lenguaje Unificado de Modelado.[En línea]. 14 de marzo de 2009. [Citado el: 16 de enero de 2011].

<http://www.que-informatica.com/index.php/software/uml-lenguaje-unificado-de-modelado/>

Downes, Stephen. The Learning Marketplace. Meaning, Metadata and Content Syndication in the Learning Object Economy.[En línea]. 2004. [Citado el: 2 de diciembre de 2010].

<http://www.downes.ca/files/book3.htm>

Eguiluz, Javier. ¿Qué es Symfony?.[En línea]. 2011.[Citado el: 11 de enero de 2011].

<http://www.symfony.es/que-es-symfony/>

Flores Cuesta, Gilda y Sánchez Tarragó, Nancy .Los repositorios institucionales: análisis de la situación internacional y principios generales para Cuba.[En línea].2007.[consultado el: 6 de diciembre de 2010].

http://bvs.sld.cu/revistas/aci/vol16_6_07/aci061207.htm

Galeana de la O, Lourdes. Objetos de Aprendizaje .[En línea]2005.[Citado el: 6 de diciembre de 2010].

<http://www.uoc.edu/symposia/spdece05/ppt/ID16.ppt>

Garcia, Joaquin. CMM-CMMI.[En línea].14 de agosto de 2005.[Citado el: 13 de enero de 2011].

<http://www.ingenierossoftware.com/calidad/cmm-cmmi.php>

García Peñalvo, Francisco José .Estado actual de los sistemas e-learning.[En línea].2009.[Citado el: 29 de noviembre de 2010].

Giraldo, Luis y Zapata, Yuliana. Herramientas de Desarrollo de Ingeniería de SW Para Linux. [En línea]. 24 de Septiembre de 2005 .[Consultado el: 16 de enero de 2011].

http://campus.usal.es/~teoriaeducacion/rev_numero_06_2/n6_02_art_garcia_penalvo.htm

Gómez Gallego, Juan Pablo y Galves, Jorge. Fundamentos de la metodología RUP: Rational Unified Process.[En línea]. 2007.[Citado el: 15 de enero de 2011].

<http://www.monografias.com/trabajos-pdf4/ensayo...rup/ensayo-sobrite-rup.pdf>

Gutiérrez, Javier J. ¿Qué es un framework web? .[En línea]. 2008. [Citado el: 10 de enero de 2011]

http://www.lsi.us.es/~javierj/investigacion_ficheros/Framework.pdf

Inostroza, Patricio. OA: Objetos de Aprendizaje. [En línea].27 de Diciembre de 2006.[Consultado el: 6 de diciembre de 2010].

<http://www.dcc.uchile.cl/~pinostro/OA/ObjetosDeAprendizaje.html>

Jacobson, Ivar. Booch, Grady. Rumbaugh, James. El proceso Unificado de Desarrollo de Software. [En línea]. 2000.[Citado el: 16 de enero de 2011].

<http://elvex.ugr.es/decsai/java/pdf/3E-UML.pdf>

López Guzmán, Clara y García Peñalvo, Francisco J. Repositorios de objetos de aprendizaje: bibliotecas para compartir y reutilizar recursos en los entornos e-learning .[En línea]. 2006. [Citado el: 2 de diciembre de 2010].

<http://redalyc.uaemex.mx/pdf/285/28590202.pdf>

Méndez, Eva y Senso, José A. Unidad de Autoformación: Introducción a los metadatos: Estándares y Aplicación.[En línea]. SEDIC 2004.[Citado el: 3 de diciembre de 2010].

<http://www.sedic.es/autoformacion/metadatos/programa.htm>

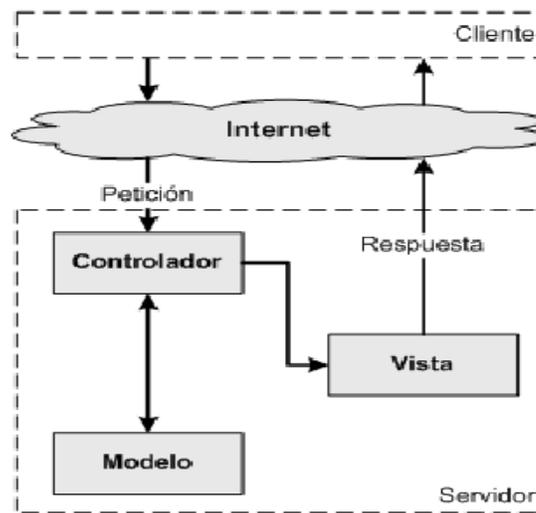
- Mendoza, Jorge A.** e-Learning, el futuro de la educación a distancia. [En línea]. 10 de Junio del 2003. [Citado el: 29 de noviembre de 2010].
<http://www.informaticamilenium.com.mx/paginas/mn/articulo78.html>
- Montero, Daniel Fernando.** Pruebas de Software-Fundamentos y objetivos. [En línea]. 2010.[Citado el: 19 de mayo de 2011].
<http://adsig7.blogspot.com/>
- Objetos vitruales de aprendizaje.** Listado de repositorios. [En línea].[Consultado el: 2 de diciembre de 2010].
<http://sites.google.com/site/jowaloje/repositorios>
- Open Society Institute.**A guide to institutional repository software. [En línea].2004.[Consultado el: 7 de diciembre de 2010].
http://www.soros.org/openaccess/software/OSI_Guide_to_Institutional_Repository_Software_v3.htm
- Potencier, Fabien y Zaninotto, François.** Symfony la guía definitiva .[En línea]. 2008.[Citado el: 11 de enero de 2011].
- Pressman, Roger S.** Ingeniería de Software: Un enfoque práctico. 2002.[Citado el: 18 de mayo 2011].
http://www.librosweb.es/symfony_1_2
- RAE.** Real Academia Española.[En línea].2010.[Citado el: 3 de diciembre de 2010].
<http://www.rae.es>
- Rodríguez Gairín, Josep Manuel y Sulé Duesa, Andreu.** DSpace: un manual específico para gestores de la información y la documentación. [En línea]. 2009. [Citado el: 18 de diciembre de 2010.]
<http://www.ub.es/bid/20rodri2.htm>.
- Rouyet, Juan I. y Martín, Victor.** A comparative study of the metadata in SCORM and Dublin Core . 2006.[Citado el: 2 de junio de 2011].
- Salvador Broche, Orlando Felipe y Soler Martín , Javier.** Implementación de un Repositorio de Objetos de Aprendizaje para la Universidad de las Ciencias Informáticas. Mayo de 2010.[Consultado el: 3 de diciembre de 2010].
- Sierra, María.** Trabajando con Visual Paradigm for UML.[En línea].[Consultado el: 16 de enero de 2011].
<http://personales.unican.es/ruizfr/is1/doc/lab/01/is1-p01-trans.pdf>
- Thompson, Ivan.** Definición de Administración.[En línea]. Enero de 2008.[Citado el: 17 de enero de 2010].
<http://www.promonegocios.net/administracion/definicion-administracion.html>

Uttangi, Roshan V. La vía rápida a la aplicación de CMMI: Integración de los CMMI y los marcos de proceso RUP.[En línea].[Consultado el: 16 de enero de 2011].

<http://www.ibm.com/developerworks/rational/library/oct07/uttangi.../index.html>

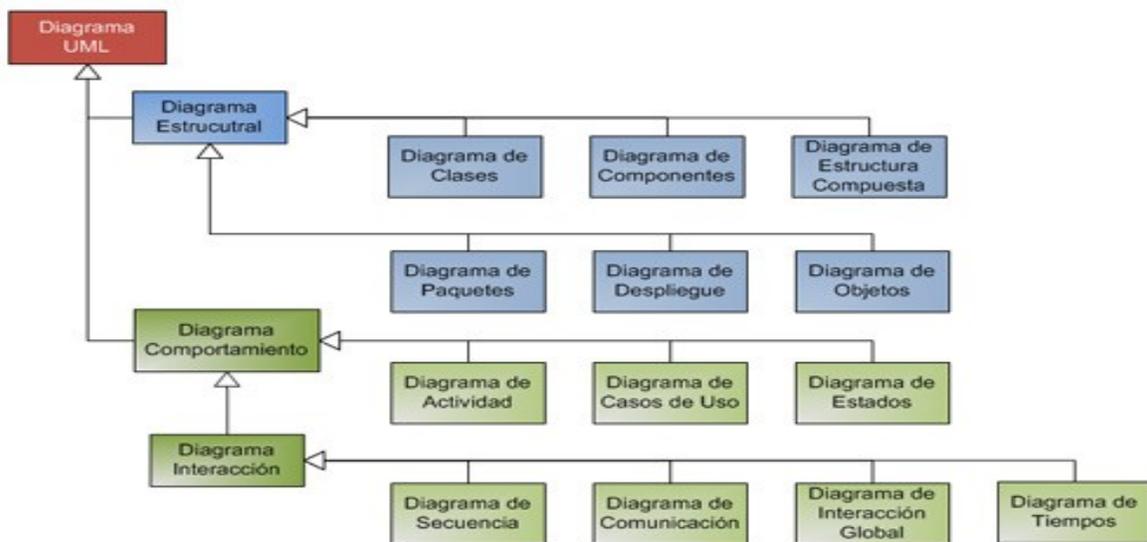
Zammeti, Frank W. Practical Ext JS Projects with Gears. 2009. [Consultado: 1 de junio de 2011].

Anexo 1. Niveles del patrón MVC.



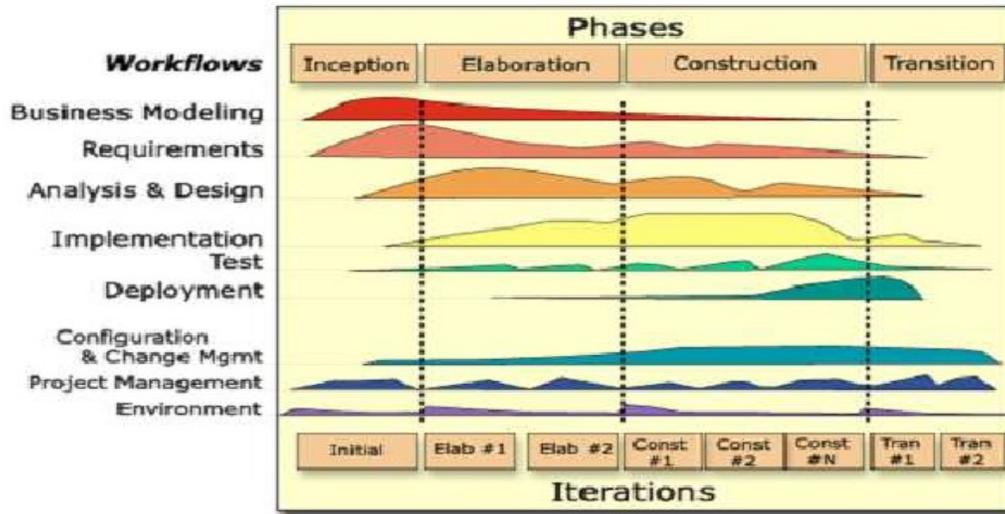
(Potencier y Zaninotto, 2008)

Anexo 2. Taxonomía de UML.



(Domínguez, 2009)

Anexo 3. Faces del ciclo de vida de RUP.



(Gómez y Galves, 2007)

Anexo 4. Diseño de caso de prueba “Configurar Base de datos Postgres”.

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad
1. Base de datos Postgres.	EC 1.1: Seleccionar Configurar conexión de la base de datos Postgres.	Se muestran los campos para configurar la conexión y los botones de confirmación de acción: Aceptar y Cancelar. Se pulsa el botón “Guardar” y se almacenan los datos en el sistema.
	EC 1.2: Datos incompletos.	Se muestra un mensaje al lado de cada campo que se encuentre vacío señalando que es de tipo obligatorio. El campo se señalará en rojo.
	EC 1.3: Datos incorrectos.	Se muestra al lado del campo un mensaje señalando el tipo de error y el campo señalado en rojo.
	EC 1.4: Cancelar la acción.	Se presiona el botón Cancelar para evitar guardar los cambios efectuados.

Escenario	Servidor	Base de datos	Usuario	Contraseña	Respuesta del Sistema	Resultado de la Prueba
1.1 Configurar base conexión de datos Postgres	V	V	V	V	Se muestra una interfaz que brinda la posibilidad de realizar la configuración. Se muestra además un botón "Guardar" y "Cancelar". Se pulsa el botón "Guardar" y se almacenan los datos en el sistema.	Satisfactorio.
EC 1.2 Datos incompletos	N/A	N/A	N/A	N/A	Se muestra un mensaje al lado de cada campo que se encuentre vacío y se señalará en rojo. Se permite que se vuelvan a entrar los datos solicitados.	Satisfactorio.
	V	N/A	N/A	N/A		
	N/A	V	N/A	N/A		
	N/A	N/A	V	N/A		
	N/A	N/A	N/A	V		
1.3 Datos incorrectos.	I	I	I	I	Al introducir los datos de forma incorrecta se muestra al lado del campo un mensaje señalando el tipo de error y el campo señalado en rojo.	Satisfactorio.
	I	V	V	V		
	V	I	V	V		
	V	V	I	V		
	V	V	V	I		
EC 1.4 Cancelar la acción.	N/A	N/A	N/A	N/A	Se presiona el botón Cancelar para evitar guardar los cambios efectuados.	Satisfactorio.

Anexo 5. Diseño de caso de prueba “Página de mantenimiento”.

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad
1. Página de mantenimiento.	EC 1.1: Seleccionar y configurar página de mantenimiento.	Se muestra una interfaz donde se va a diseñar la página de mantenimiento que se quiere mostrar. Permite además que se establezcan el rango de fecha y hora en que se quiere que se active y se desactive la página. Se ofrece la opción de habilitar y deshabilitar la página de mantenimiento. Se pulsa el botón habilitar y se guarda la configuración en el sistema.
	EC 1.2: Datos incorrectos.	Cuando se introduce un rango de fecha incorrecto se muestra un mensaje alertando del error.
	EC 1.3: Deshabilitar la acción.	Se presiona el botón “Deshabilitar” y se elimina la configuración que se encuentra vigente. Se restablece la página de inicio del sistema.

Escenario	Fecha de inicio	Fecha de Fin	Hora de inicio	Hora de fin	Editor	Respuesta del Sistema	Resultado de la Prueba
EC 1.1 Página mantenimiento	V	V	V	V	N/A	Se muestra una interfaz para diseñar la página de mantenimiento. Permite además que se establezcan el rango de fecha y hora en que se quiere que se active y se desactive la página. Se ofrece la opción de habilitar y deshabilitar la página de mantenimiento. Se pulsa el botón habilitar y se guarda la configuración en el sistema.	Satisfactorio.

Escenario	Fecha de inicio	Fecha de Fin	Hora de inicio	Hora de fin	Editor	Respuesta del Sistema	Resultado de la Prueba
EC 1.2 Datos incorrectos	I	I	N/A	N/A	N/A	Cuando se introduce un rango de fecha incorrecto se muestra un mensaje alertando del error.	Satisfactorio.
	V	I	N/A	N/A	N/A		
	I	V	N/A	N/A	N/A		
EC 1.3 Deshabilitar	N/A	N/A	N/A	N/A	N/A	Se presiona el botón Deshabilitar para eliminar la configuración que se encuentra vigente. Se restablece la página de inicio del sistema.	Satisfactorio.

No	No conformidad	Etapas de detección	Significativa	Estado
1	No redirecciona bien la aplicación al ser deshabilitada la página de mantenimiento.	1ra iteración	X	16/05/11-PD: Pendiente 20/5/11-RA: Resuelta

Anexo 6. Diseño de caso de prueba “Administrar OA”.

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad
1. Administrar OA	EC 1.1: Administrar OA.	Muestra el listado de los OA para administrar, sobre los que se puede realizar funciones como: Editar, Editar múltiple, Eliminar, Eliminar múltiple, Visualizar, Descargar.
2. Eliminar OA	EC 2.1: Eliminar.	Muestra una ventana con el mensaje

		<p>“Está seguro que desea eliminar” y los botones de acción: Aceptar y Cancelar.</p> <p>Se pulsa el botón “Aceptar” y se elimina el OA del sistema.</p>
	EC 2.2: Cancelar.	Evita que se elimine el OA del sistema.
3. Editar OA	EC 3.1: Editar OA.	Muestra la interfaz dedicada a la edición de OA. Ver caso de Prueba editar OA.
4. Visualizar OA	EC 4.1: Visualizar OA.	Visualiza el OA correspondiente en una nueva ventana.
5. Eliminar selección	EC 5.1: Eliminar selección.	<p>Muestra una pantalla de confirmación de la acción con el mensaje: “Está seguro que desea eliminar los OA” y los botones de acción: Aceptar y Cancelar.</p> <p>Se pulsa el botón “Aceptar” y se eliminan los OA seleccionados del sistema.</p>
	EC 5.2: Cancelar.	Evita que los OA seleccionados sean eliminados del sistema.
6. Editar selección	EC 6.1: Editar OA seleccionados.	<p>Muestra una pantalla de confirmación de la acción con el mensaje: “Está seguro que desea editar los OA” y los botones de acción: Aceptar y Cancelar.</p> <p>Se pulsa el botón “Aceptar” y se muestra la interfaz dedicada a la edición múltiple de OA. Donde se podrá editar el OA atendiendo a tres criterios:</p> <p>Categoría, Subcategoría, Estado.</p>
	EC 6.2: Cancelar.	Evita que los OA seleccionados sean editados.

Escenario	Título	Usuario	Categoría	Subcategoría	Estado	Respuesta del Sistema	Resultado de la Prueba
EC 1.1 Administrar OA.	NA	NA	NA	NA	NA	Muestra el listado de los OA para administrar, sobre los que se puede realizar funciones como: Editar, Editar múltiple, Eliminar, Eliminar múltiple, Visualizar, Descargar.	Satisfactorio.

Escenario	Título	Usuario	Categoría	Subcategoría	Estado	Respuesta del Sistema	Resultado de la Prueba
EC 2.1 Eliminar OA.	NA	NA	NA	NA	NA	Muestra una ventana con el mensaje “Está seguro que desea eliminar” y los botones de acción: Aceptar y Cancelar. Se pulsa el botón “Aceptar” y se elimina el OA del sistema.	Satisfactorio.
EC 2.1 Cancelar.	NA	NA	NA	NA	NA	Evita que se elimine el OA del sistema.	Satisfactorio.

Escenario	Título	Usuario	Categoría	Subcategoría	Estado	Respuesta del Sistema	Resultado de la Prueba
EC 3.1 Editar OA.	NA	NA	NA	NA	NA	Muestra la interfaz dedicada a la edición de OA. Ver caso de Prueba editar OA.	Satisfactorio.

Escenario	Título	Usuario	Categoría	Subcategoría	Estado	Respuesta del Sistema	Resultado de la Prueba
EC 4.1 Visualizar OA.	NA	NA	NA	NA	NA	Visualiza la información correspondiente al OA.	Satisfactorio.

Escenario	Título	Usuario	Categoría	Subcategoría	Estado	Respuesta del Sistema	Resultado de la Prueba
EC 5.1 Eliminar selección	NA	NA	NA	NA	NA	Muestra una pantalla de confirmación de la acción con el mensaje: “Está seguro que desea eliminar los OA” y los botones de acción: Aceptar y Cancelar. Se pulsa el botón “Aceptar” y se eliminan los OA seleccionados del sistema.	Satisfactorio.

Escenario	Título	Usuario	Categoría	Subcategoría	Estado	Respuesta del Sistema	Resultado de la Prueba
EC 6.1 Editar selección.	NA	NA	NA	NA	NA	<p>Muestra una pantalla de confirmación de la acción con el mensaje: “Está seguro que desea editar los OA” y los botones de acción: Aceptar y Cancelar.</p> <p>Se pulsa el botón “Aceptar” y se muestra la interfaz dedicada a la edición múltiple de OA. Donde se podrá editar el OA atendiendo a tres criterios:</p> <p>Categoría, Subcategoría, Estado.</p>	Satisfactorio.
EC 6.2 Cancelar.	NA	NA	NA	NA	NA	Evita que se editen los OA seleccionados del sistema.	Satisfactorio.

No	No conformidad	Etapas de detección	Significativa	Estado
1	Solo edita el primero de la selección.	1ra iteración	X	16/05/11-PD: Pendiente. 2/5/11-RA: Resuelta.

Anexo 7. Diseño de caso de prueba “Configurar base de datos XML Exist”.

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad
1. Base de datos XML Exist	EC 1.1: Configurar conexión de la base de datos XML Exist.	Se muestran los campos para configurar la conexión:

		<p>Servidor (Obligatorio)</p> <p>Base de datos (Obligatorio)</p> <p>Usuario (Obligatorio)</p> <p>Contraseña (Obligatorio)</p> <p>Muestra los botones: Aceptar y Cancelar. Se pulsa el botón "Guardar" y se almacenan los datos en el sistema.</p>
	EC 1.2: Datos incompletos.	<p>Se muestra un mensaje al lado de cada campo que se encuentre vacío señalando que es de tipo obligatorio. El campo se señalará en rojo.</p> <p>Se permite que vuelvan a entrar los datos solicitados.</p>
	EC 1.3: Datos incorrectos.	<p>Al introducir los datos de forma incorrecta y pulsar el botón de guardar se muestra un mensaje señalando el tipo de error, el campo se señala en rojo.</p> <p>Se permite activar nuevamente la opción de introducir nuevos datos y se da paso al escenario 1.</p>
	EC 1.4: Cancelar.	<p>Se presiona el botón Cancelar para evitar guardar los cambios efectuados.</p>

Escenario	Servidor	Base de datos	Usuario	Contraseña	Respuesta del Sistema	Resultado de la Prueba
EC 1.1 Configurar base de datos XML Exist	V	V	V	V	Se muestran los campos para configurar la conexión: Servidor (Obligatorio) Base de datos (Obligatorio) Usuario (Obligatorio) Contraseña (Obligatorio) Muestra los botones de confirmación de acción: Aceptar y Cancelar. Se pulsa el botón "Guardar" y se almacenan los datos en el sistema.	Satisfactorio.
EC 1.2 Datos incompletos.	I	I	I	I	Se muestra un mensaje al lado de cada campo que esté vacío señalando que es de tipo obligatorio. El campo se señalará en rojo.	Satisfactorio.
	V	I	I	I		
	I	V	I	I	Se permite que se vuelvan a entrar los datos solicitados.	
	I	I	I	I		
	I	I	I	V		
EC 1.3 Datos incorrectos.	I	I	I	I	Al introducir los datos de forma incorrecta y pulsar el botón de guardar se muestra al lado de los campos un mensaje señalando el tipo de error, se señalan los campos en rojo.	Satisfactorio.
	I	V	V	V		
	V	I	V	V		
	V	V	I	V	Se permite activar nuevamente la opción de introducir nuevos datos y se da paso al escenario 1.	
	V	V	V	I		

Escenario	Servidor	Base de datos	Usuario	Contraseña	Respuesta del Sistema	Resultado de la Prueba
EC 1.4 Cancelar.	N/A	N/A	N/A	N/A	Se presiona el botón Cancelar para evitar guardar los cambios efectuados.	Satisfactorio.

Anexo 8. Diseño de caso de prueba “Configurar conexión con un Directorio Activo”.

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad
1. Configurar DA	EC 1.1: Configurar DA.	<p>Se muestra se activan los campos:</p> <p>URL host (Obligatorio)</p> <p>Servidor (Obligatorio)</p> <p>Usuario (Obligatorio)</p> <p>Contraseña (Obligatorio)</p> <p>Contexto (Obligatorio)</p> <p>Atributos (Obligatorio)</p> <p>Datos (Obligatorio)</p> <p>Se muestran los botones “Guardar” y “Cancelar”. Se pulsa el botón “Guardar” y se almacena la nueva configuración en el sistema.</p>
	EC 1.2: Datos incorrectos.	<p>Al introducir los datos incorrectos y pulsar el botón “Guardar” se muestra un mensaje al lado de los campos señalando el tipo de error. Los campos donde se han introducido datos inválidos se señala en rojo.</p>

	EC 1.3: Cancelar.	Se presiona el botón Cancelar para evitar guardar los cambios efectuados. Se sale de la funcionalidad.
--	-------------------	--

Escenario	URL del host	Servidor	Versión	Codificación LDAP	Ocultar contraseña	Usuario	Contraseña	Contexto	Buscar en subcontextos	Atributos	Datos	Respuesta del Sistema	Resultado de la Prueba
EC 1.1 Configurar DA.	V	V	V	V	V	V	V	V	V	V	V	Se activan los campos: URL host (Obligatorio) Servidor (Obligatorio) Usuario (Obligatorio) Contraseña (Obligatorio) Contexto (Obligatorio) Atributos (Obligatorio) Datos (Obligatorio) Se muestran los botones “Guardar” y “Cancelar”. Se pulsa el botón “Guardar” y se almacena la nueva configuración en el sistema.	Satisfactorio.
	I	I	I	I	I	I	I	I	I	I	I	Al introducir los datos	Satisfactorio.
	I	V	V	V	V	V	V	V	V	V	V	incorrectos y pulsar el	

Escenario	URL del host	Servidor	Versión	Codificación LDAP	Ocultar contraseña	Usuario	Contraseña	Contexto	Buscar en subcontextos	Atributos	Datos	Respuesta del Sistema	Resultado de la Prueba
EC 1.2 Datos incorrectos.	V	I	V	V	V	V	V	V	V	V	V	botón “Guardar” se muestra un mensaje al lado de los campos señalando el tipo de error. Los campos donde se han introducido datos inválidos se señala en rojo. Se permite que se llenen nuevamente y se ejecuta el escenario 1.1.	
	V	V	I	V	V	V	V	V	V	V	V		
	V	V	V	I	V	V	V	V	V	V	V		
	V	V	V	V	I	V	V	V	V	V	V		
	V	V	V	V	V	I	V	V	V	V	V		
	V	V	V	V	V	V	I	V	V	V	V		
	V	V	V	V	V	V	V	V	I	V	V		
	V	V	V	V	V	V	V	V	V	I	V		
	V	V	V	V	V	V	V	V	V	V	I		
	V	V	V	V	V	V	V	V	V	V	I		
EC 1.3 Datos incompletos.	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	Se presiona el botón Cancelar para evitar guardar los cambios efectuados. Se sale de la funcionalidad.	Satisfactorio.

Anexo 9. Diseño de caso de prueba “Historial de revisor”.

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad
1. Historial de revisor	EC 1.1: Consultar Historial de Revisor.	Muestra una interfaz con el listado de todos los revisores del sistema y las revisiones que efectuaron. Permite filtrar por:

		<p>usuario, categoría, subcategoría.</p> <p>Permite ordenar los campos de forma ascendente y descendente.</p> <p>Se ejecuta la búsqueda al pulsar el botón “Buscar”.</p>
	EC 1.2: Filtra por atributos.	Se seleccionan los filtros por los que se pretende ejecutar la búsqueda. Los filtros se pueden combinar para lograr un resultado más efectivo.

Escenario	Usuario	Categoría	Subcategoría	Respuesta del Sistema	Resultado de la Prueba
EC 1.1 Consultar Historial de revisor.	I	I	I	<p>Se muestra una interfaz con el listado de todos los revisores del sistema y las revisiones que efectuaron.</p> <p>Permite filtrar por:</p> <p>usuario, categoría, subcategoría.</p> <p>Permite ordenar los campos de forma ascendente y descendente.</p> <p>Se ejecuta la búsqueda al pulsar el botón “Buscar”.</p>	Satisfactorio.
EC 1.2 Filtrar por atributos.	NA	NA	NA	Se seleccionan los filtros por los que se pretende ejecutar la búsqueda. Los filtros se pueden combinar para lograr un resultado más efectivo.	Satisfactorio.
	V	NA	NA		
	NA	V	NA		
	NA	NA	V		
	V	V	NA		
	V	NA	V		
	NA	V	V		

Anexo 10. Diseño de caso de prueba “Limitar capacidad de almacenamiento”.

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad
1. Limitar capacidad de almacenamiento	EC 1.1: Tamaño de OA.	Se muestra una pantalla donde aparecen los campos para limitar la capacidad de almacenamiento y el tamaño de los objetos de aprendizaje. Ofrece además las opciones de “Guardar” y “Cancelar” los cambios efectuados. Además de dar la opción de establecer las cuotas (ver CU: Gestionar tamaño de OA). Se presiona el botón “Guardar” para almacenar los datos en el sistema.
	EC 1.2: Datos incorrectos.	Al introducir los datos incorrectos y pulsar el botón “Guardar” se muestra un mensaje al lado del campo señalando el tipo de error. Los campos donde se han introducido datos inválidos se señala en rojo.
	EC 1.3: Cancelar la acción.	Se presiona el botón Cancelar para evitar guardar los cambios efectuados.

Escenario	Capacidad de almacenamiento	Tamaño de OA	Respuesta del Sistema	Resultado de la Prueba
EC 1.1 Tamaño de OA.	V	N/A	<p>Se muestra una pantalla donde aparecen los campos para limitar la capacidad de almacenamiento y el tamaño de los objetos de aprendizaje.</p> <p>Ofrece además las opciones de “Guardar” y “Cancelar” los cambios efectuados. Además de dar la opción de establecer las cuotas (ver CU: Gestionar tamaño de OA).</p> <p>Se presiona el botón “Guardar” para almacenar los datos en el sistema.</p>	Satisfactorio.
	N/A	V		
EC 1.2 Datos incorrectos.	I	N/A	<p>Al introducir los datos incorrectos y pulsar el botón “Guardar” se muestra un mensaje al lado de los campos señalando el tipo de error. Los campos donde se han introducido datos inválidos se señala en rojo. Se permite que se llenen nuevamente y se ejecuta el escenario 1.1.</p>	Satisfactorio.
	N/A	I		
	I	I		
	V	I		
	I	V		
EC 1.3 Cancelar.	N/A	N/A	<p>Se presiona el botón Cancelar para evitar guardar los cambios efectuados.</p>	Satisfactorio.

Glosario de términos

- **Administración web:** control de las configuraciones y flujo de trabajo que se realiza dentro de una aplicación web.
- **Aplicación:** programa que cuando se ejecuta permite la interacción con funcionalidades de la misma en la computadora.
- **E-learning:** *“la capacitación no presencial que, a través de plataformas tecnológicas, posibilita y flexibiliza el acceso y el tiempo en el proceso de enseñanza-aprendizaje, adecuándolos a las habilidades, necesidades y disponibilidades de cada docente, además de garantizar ambientes de aprendizaje colaborativos mediante el uso de herramientas de comunicación síncrona y asíncrona, potenciando en suma el proceso de gestión basado en competencias.”* (García, 2009).
- **Frameworks:** es un marco de trabajo que contiene un grupo de componentes que ofrecen la opción de reutilizarlo para facilitar el desarrollo de una aplicación. Pueden estar compuestas por bibliotecas y en una mayor medida de componentes de software.
- **Metadatos:** conjunto de atributos que conforman la descripción de los objetos de aprendizaje.
- **Objeto de aprendizaje:** *“recurso digital con una granularidad apropiada y una marcada intención formativa, compuesto por uno o varios objetos de información, con un único objetivo, descrito con metadatos y con un comportamiento secuenciado que asegure el correcto enlace entre los elementos de su estructura didáctica y que puede ser reutilizado en entornos e-learning”* (Rodríguez, 2011).
- **PHP:** lenguaje de programación web creado con el fin de construir sitios web dinámicos. Suele utilizarse del lado del servidor aunque también se puede utilizar desde una interfaz de líneas de comandos.
- **Repositorio:** lugar o almacén digital utilizado para contener la información digital con el fin de ser utilizados través de la red.
- **Repositorio de Objetos de Aprendizaje:** lugar o almacén digital utilizado para contener los objetos de aprendizaje con el fin de ser utilizados en el proceso de enseñanza-aprendizaje a través de la red.
- **TIC:** Tecnologías de la Informática y las comunicaciones.