

Universidad de las Ciencias Informáticas
Facultad 1



“Repositorio de Componentes para el Departamento de Servicios Informáticos para Internet”

Trabajo de diploma para optar por el título de Ingeniero en Ciencias Informáticas

Autor:

Maikel Maldonado del Toro

Tutores:

Ing. Yuneldis Reyes Velázquez

Ing. Serguey Gonzalez Garay

La Habana, Junio 2015



Declaración de autoría

Declaro que soy el único autor de este trabajo y autorizo al centro de Ideo-informática de la Universidad de las Ciencias Informáticas; así como a dicha institución para que hagan el uso que estimen pertinente con este trabajo.

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

Maikel Maldonado del Toro

Firma del Autor

Ing. Serguey Gonzalez Garay

Firma del Tutor

Ing. Yuneldis Reyes Velázquez

Firma del Tutor

Agradecimientos

A mi familia, a mi mamá, mis papás (Edecio y Alexei), mis hermanas, por apoyarme en todo momento cuando más lo necesite y ser esa fuente inspiración de luchar cada día.

A ti mi abuelita querida: Bernis por correr y ayudarme en todas mis locuras y estar siempre preocupada por mi bienestar y salud.

A mis amigos de manera general, los colegas del grupo, los muchachos del secretariado de la FEU y los arrecostados, por apoyarme en estos 5 Años.

A Zaidée por su cariño y apoyo en toda la carrera.

A mi tutor y tribunal por guiarme durante el proceso de desarrollo de la tesis en la redacción y presentación de la misma.

A las personas que no están pero que de una forma u otra contribuyeron en mi formación.



Dedicatoria

En especial a mi abuelo Edecio, que dios lo tenga en la gloria, este título es para ti que tanto querías que obtuviera y que me formara como el hombre que soy.

A mis padres, a mi mamá por ayudarme a convertirme en esta persona de hoy en día, por estar siempre ahí aportándome su granito de vida e inspiración.

Resumen

El presente trabajo de diploma describe el desarrollo de un repositorio de componentes para el Departamento de Servicios Informáticos para Internet del centro de Ideo-informática. Esta herramienta permitirá recolectar, documentar y proveer desde una ubicación central los componentes necesarios para ser reutilizados en la construcción de portales web. Se estudiaron repositorios a nivel nacional e internacional para determinar las características y servicios que brindan, así como los principales conceptos relacionados al proceso de reutilización. Para la implementación se utilizó el CMS Drupal en su versión 7.34 y PostgreSQL 9.1 como Sistema Gestor de Bases de Datos (SGBD). Para guiar todo el proceso de desarrollo de la aplicación se seleccionó la metodología SXP. Se empleó el servidor web Apache 2.4, PHP 5, HTML 5, CSS 3, JavaScript 1.8, jQuery 1.4 y Firebug 2.0. Como herramienta gráfica para la administración del SGBD se utilizó el PgAdminIII 1.14, Apache JMeter 2.9 para las pruebas de carga y estrés y como IDE de desarrollo NetBeans 7.3. El repositorio contará con funcionalidades como la gestión de componentes y avisos, así como permitirá exportar requisitos a Word y a Excel, además de contar con un foro de discusión. Posee un sistema de notificación por correo al efectuarse una nueva publicación y una búsqueda avanzada con varios criterios de filtrado. Como resultado se obtiene un producto funcional con una documentación que sirve de base para futuras investigaciones o modificaciones a la solución. Las pruebas realizadas validaron el funcionamiento de la aplicación desarrollada según los requisitos definidos.

Palabras clave: componentes, factoría de software, reutilización, repositorio, repositorio de componentes.

Índice de Contenido

Introducción	1
Capítulo 1: Fundamentos teóricos de los repositorios de componentes, funcionalidades y tecnologías para su desarrollo	8
1.1 Estudio de repositorios homólogos	8
1.1.1 Resultados obtenidos del estudio	14
1.2 Análisis de las tecnologías, herramientas y lenguajes a utilizar	16
1.2.2 Lenguajes de programación.....	19
1.2.3 Framework.....	21
1.2.4 Sistemas gestores de base de datos (SGBD).....	22
1.2.5 Servidor web	24
1.2.6 Herramientas de desarrollo	25
1.2.7 Metodología de desarrollo de software.....	28
1.3 Conclusiones parciales.....	32
Capítulo 2. Características del repositorio de componentes.....	34
2.1 Propuesta del Repositorio de componentes	34
2.2 Modelo de dominio	35
2.3.1 Descripción de las clases y roles.....	35
2.4 Levantamiento de requisitos.....	36
2.4.1 Lista de Reserva del Producto (LRP)	36
2.4.2 Historias de Usuario.....	42
2.5 Arquitectura	44
2.6 Diseño	47

2.6.1 Diagrama de Clases de Diseño.....	47
2.7 Modelo de despliegue	50
2.8 Conclusiones parciales.....	51
Capítulo 3: Implementación y pruebas al repositorio de componentes.....	53
3.1 Diagrama de componentes.....	53
3.2 Estándares de codificación.....	56
3.3 Pruebas.....	58
3.3.1 Funcionales	58
3.3.2 Seguridad	64
3.3.3 Carga y Estrés	65
3.4 Conclusiones parciales.....	66
Conclusiones Generales.....	68
Recomendaciones	69
Referencias.....	70
Bibliografía Consultada	75
Glosario de términos.....	79

Introducción

En la actualidad existe un acelerado desarrollo de aplicaciones informáticas en la web relacionada con aspectos comerciales, laborales, científicos y educativos de gran impacto para el avance de cualquier país. Esto conjuntamente con la evolución de las Tecnologías de la Informática y las Comunicaciones (TIC) ha traído consigo que la industria del software se encuentre inmersa en grandes cambios, pues se requieren software ya sean de sistemas, programación o aplicaciones, cada vez más grandes y complejos con mayor calidad y en menor tiempo de desarrollo.

Se deriva de lo anterior que tanto a nivel nacional como internacional, la construcción de aplicaciones web se torne un proceso importante y a la vez necesario, teniendo en cuenta los avances en el desarrollo tecnológico y las necesidades sociales que son cada vez más exigentes. Este aspecto ha conllevado a que en los últimos tiempos se evidencie una proyección en este sentido, centrado en el trabajo en equipo, centralización de la información y reutilización del conocimiento, experiencias, ideas, artefactos, componentes y otros, en aras de no duplicar esfuerzos.

Cuando se habla de reutilización en el marco de la producción de software, no solo se está haciendo referencia a la reutilización de código en el desarrollo de proyectos, ni líneas de código compiladas en bibliotecas o marcos de trabajo, sino a todo aquel artefacto generado en dicho ciclo de proyecto ya sea documentación, manuales, planes de prueba, procedimientos de instalación y hasta el propio conocimiento de cada trabajador expresado en investigaciones realizadas, artículos o en webs interactivas creadas al efecto.

Según plantean (PELECHANO, 2005), (TERREROS, 2015) y (MONTILVA, 2003) al reutilizar se construyen sistemas y soluciones cada vez más grandes y robustas, con bases firmes y de calidad. Se logra en este sentido automatizar el desarrollo y mantenimiento de diversos productos de software, a través de su adaptación, ensamblaje y configuración de componentes.

En Cuba existe una industria creciente en el desarrollo de software, pues varias son las instituciones y universidades implicadas en ello. Se destaca entre estas la Universidad de las Ciencias Informáticas (UCI), que tiene entre sus pilares la producción de aplicaciones y servicios informáticos orientados a diversos sectores de la economía y los servicios, dentro y fuera de Cuba (UCI, 2012). En esta universidad,

específicamente en los centros de desarrollo de software distribuidos por cada una de sus facultades se ha reconocido la importancia y puesta en práctica de un desarrollo basado en componentes que está siendo utilizado en la actualidad. Este tipo de desarrollo permite que la estructura organizacional de algunos centros se transforme con el fin de ser competitivos, logrando un enfoque industrializado de sus procesos. Este hecho conduce a la necesidad de implantar nuevas metodologías y modelos, así como la reformulación de nuevos conceptos para optimizar los procesos de desarrollo de software según (VÁSQUEZ, 2013), (CASAÑOLA, 2007) y (VIGNAGA,2003).

De acuerdo con las investigaciones de (TERREROS, 2015), (COMPONENTSOURCE, 2014), (SZYPERSKI, 2002) y (MONTILVA, 2003) se define para los efectos de esta investigación como componente aun un paquete coherente de artefactos de software que ha de poder ser desarrollado, adquirido, incorporado al sistema y compuesto con otros componentes de forma independiente para construir algo mucho más grande, entendiéndose también otros elementos de software, tales como algoritmos, diseños, arquitecturas de software, documentación y especificaciones de requerimientos. Estos son de vital importancia a la hora de desarrollar aplicaciones web por parte de algunos de los centros de desarrollo de la UCI. La reutilización de componentes se ha convertido en uno de sus propósitos de manera que no se pase dos veces por un mismo proceso para resolver el mismo problema, sino que se disponga de mecanismos y herramientas centralizadas y de fácil localización, a partir de información almacenada en situaciones precedentes.

Dentro de los centros de desarrollo de software de la UCI se encuentra el centro de Ideo-Infomática (CIDI), que cuenta con el departamento de Servicios Informáticos para Internet (SENIT) donde se trabaja en el desarrollo web a partir de proyectos Innovación + Desarrollo (I+D). Este departamento por sus peculiaridades y experiencias en la producción de portales web y en los servicios que brinda, está adoptando un nuevo modelo de desarrollo: la factoría de software. Esta factoría se define como una línea de productos de software que configura herramientas extensibles, procesos y contenido para automatizar el desarrollo y mantenimiento de variantes de un producto mediante la adaptación, ensamblaje y configuración de componentes basados en marco de trabajo (PELECHANO, 2005).

Este enfoque de desarrollo de software, se extiende desde el 'hacer a mano' hacia la fabricación o manufactura, proporcionando un alto grado de especialización hacia el Desarrollo de Software Basado en Componentes (DSBC). Este último constituye una aproximación al desarrollo de software que describe,

construye y emplea técnicas de software para elaborar sistemas abiertos y distribuidos, mediante el ensamblaje de partes de software reutilizables (MARTÍNEZ, 2003). De esta manera se reducen considerablemente los costos y el tiempo de desarrollo, ayudando así a elevar la productividad y la calidad del trabajo.

El modelo de Factoría de Software que se implementa en el departamento de SENIT con la denominación de Fábrica de Portales, permitirá la organización del proceso de desarrollo de software. En general contribuirá al desarrollo de productos más fortalecidos y con altos porcentajes de reutilización a través de los componentes que se desarrollan a diario en el centro. Una de las entidades organizacionales de este modelo de desarrollo de software lo constituye el repositorio de componentes existente en el departamento desde el 2014 (FERNÁNDEZ, 2014). Dicho repositorio facilita la reutilización y descarga de componentes para la construcción de portales web, ya sean módulos, herramientas, arquitecturas de información, códigos, documentación y temas que se gestionan en el sistema. Se permite la autenticación de usuarios otorgándose uno de dos roles: usuario autenticado o administrador, además de dar acceso al foro de discusión.

Con el fin de recopilar información para el análisis del repositorio se utilizó la entrevista (véase Anexo 1) a diferentes miembros del departamento SENIT, cuyos resultados arrojaron, entre otros elementos, que los entrevistados nunca visitaban el repositorio y en algunos casos tenían desconocimiento de su existencia. Durante el análisis realizado se detectaron insuficiencias que generan insatisfacción en el departamento y que frenan la utilización de los componentes del repositorio.

Una de estas insuficiencias radica en que el proceso de publicación de componentes solo puede ser llevado a cabo por los administradores del sitio. Lo anterior implica que un usuario del departamento, al querer publicar o actualizar un componente tenga que previamente contactar con un administrador para que este lo publique, convirtiéndose en un proceso tedioso. La publicación y actualización de los diversos componentes del repositorio según (FERNÁNDEZ, 2014) hasta el 25 de mayo del 2015 era de un total de 275 componentes durante casi un año y medio de funcionamiento, siendo el 78% de los componentes publicados en enero de 2014 y el último en septiembre de ese mismo año, lo que evidencia la carencia y falta de sistematicidad en la inclusión de componentes.

Además, no se permite la gestión de componentes necesarios para el departamento como lo son: pautas, estándares y recursos de diseño, requisitos funcionales, instaladores y expedientes tipo, provocando la no preservación de los mismos en el tiempo y que no se brinde el acceso abierto a usuarios externos al departamento que necesiten de estos. Por otra parte, localizar los componentes dentro del repositorio consume bastante tiempo y es trabajoso debido a que no posee una búsqueda avanzada que le proporcione al usuario diferentes criterios de filtrado para una mejor localización. El diseño del repositorio no es factible para que el usuario tenga acceso a informaciones importantes para el mismo, como por ejemplo no se muestra un bloque con los últimos componentes publicados en la portada provocando desconocimiento y descontento entre los usuarios. De igual manera no se cuenta con un sistema de notificación por correo al efectuarse una nueva publicación, ni con bloques de suscripción a boletín, contenidos más descargados, votados o populares, afectando la percepción de la importancia de los componentes cuando un usuario intenta buscar una solución adecuada a sus necesidades.

Otro aspecto a considerar es el período de tiempo que se utiliza en el departamento SENIT para la construcción de un portal. Este puede extenderse como promedio hasta 4,5 meses (ÁLVAREZ, 2015) y si esta condición persiste podría provocar que posibles clientes busquen otras alternativas para satisfacer sus necesidades. En materia de desarrollo es usual que se implementen reiteradamente funcionalidades o mejoras en forma de módulos o parches para la plataforma de desarrollo utilizada. Sin embargo, debido a las existentes deficiencias a la hora de publicar un componente en el repositorio, es posible que estos no puedan ser reutilizados y se reproduzca el proceso de investigación, implementación y prueba de la misma solución por los equipos de desarrollo, con el consumo de tiempo, esfuerzo y dedicación que este proceso implica.

Todas estas insatisfacciones propician que el proceso de reutilización de componentes por parte de los especialistas del departamento no se efectúe adecuadamente y con la sistematicidad requerida. Lo anterior conlleva a que en algunos casos no se atienda de manera rápida y eficiente, la gran demanda de solicitudes de desarrollo de portales web que instituciones y organizaciones realizan.

Teniendo en cuenta lo anteriormente expuesto se plantea el siguiente **problema de investigación**: ¿Cómo contribuir a la recolección, documentación y almacenamiento de componentes reutilizables para la

construcción de portales web en el Departamento de Servicios Informáticos para Internet del centro de Ideo-informática?

Se define como **objeto de estudio** el proceso de recolección, documentación y almacenamiento de componentes de software y constituye el **objetivo general** de la investigación desarrollar un repositorio de componentes que permita la recolección, documentación y almacenamiento de componentes reutilizables para la construcción de portales web en el Departamento de Servicios Informáticos para Internet del centro de Ideo-informática.

Se plantea como **campo de acción** el proceso de recolección, documentación y almacenamiento de componentes de software vinculado al desarrollo de portales web del Departamento de Servicios Informáticos para Internet del centro de Ideo-informática.

Objetivos específicos que conducirán al logro del objetivo general:

1. Caracterizar los fundamentos teóricos y tendencias en el desarrollo de repositorios de componentes.
2. Diseñar las funcionalidades del repositorio de componentes para la recolección y gestión de elementos reutilizables en la construcción de portales web.
3. Implementar las funcionalidades del repositorio de componentes para la recolección y gestión de elementos reutilizables en la construcción de portales web.
4. Validar la solución desarrollada mediante la realización de pruebas de software.

La presente investigación **defiende la siguiente idea**: con el desarrollo del repositorio de componentes para el Departamento de Servicios Informáticos para Internet se contribuye a la recolección, documentación y almacenamiento de componentes reutilizables para la construcción de portales web en el mencionado departamento del centro de Ideo-informática.

Para dar cumplimiento a los objetivos trazados anteriormente se efectuarán las siguientes **tareas de investigación**:

1. Realización de un estudio sobre los fundamentos teóricos y las tendencias del desarrollo web de repositorios de componentes.

2. Identificación de requisitos funcionales y no funcionales para el desarrollo del repositorio de componentes.
3. Selección de las tecnologías, herramientas, estándares y patrones que permitirán desarrollar la solución.
4. Selección de la metodología de desarrollo de software.
5. Implementación de las funcionalidades del repositorio de componentes a desarrollar.
6. Aplicación de las pruebas de funcionalidad, carga y estrés y seguridad a la solución desarrollada.

Los métodos de investigación utilizados en la presente investigación son:

Métodos teóricos:

El Analítico-Sintético se utilizó en la búsqueda y análisis de los elementos más importantes asociados a los repositorios de componentes según la bibliografía consultada en libros, páginas web y artículos, realizando una síntesis de los elementos más importantes de la misma. Se identificó con la utilización de este método las características, tecnologías, pautas y fundamentos teóricos que vinculan a los repositorios.

El Histórico-Lógico permitió una mayor comprensión del estado y tendencias actuales de los repositorios de componentes, así como las tecnologías existentes y utilizadas que permitiera la selección de las más idóneas para el desarrollo del sistema.

Modelación permitió la modelación de diagramas de componentes, de paquetes, entre otros, para la implementación de la propuesta de solución.

Métodos empíricos:

La Observación se tuvo en cuenta a la hora de valorar la propuesta más adecuada de las tecnologías y soluciones existentes, además de permitir la observación del proceso de negocio de desarrollo de portales web en el departamento SENIT y relacionados con la investigación para conocer su funcionamiento, organización y estructura de la información que contiene.

La Entrevista se empleó en encuentros con profesores, especialistas y el jefe de departamento de Servicios Informáticos para Internet del centro CIDI con el objetivo de entender mejor el proceso del negocio y definir las principales funcionalidades del sistema.

La presente investigación está estructurada en 3 capítulos de la siguiente manera:

Capítulo 1: “Fundamentos teóricos de los repositorios de componentes, funcionalidades y tecnologías para su desarrollo”: en este capítulo se abarcará el estado de los repositorios de componentes a nivel nacional e internacional, para posteriormente realizar un balance y estudio de un conjunto de herramientas, tecnologías y metodologías a seleccionar, para el desarrollo de la propuesta de solución.

Capítulo 2. “Características del repositorio de componentes”: en este capítulo se expone la propuesta y características del repositorio de componentes de manera detallada, describiéndose los requisitos funcionales y no funcionales, diagramas de clases de diseño, diagrama de despliegue, así como todo lo referente a la arquitectura del sistema y a los patrones de diseño utilizados.

Capítulo 3: “Implementación y pruebas al repositorio de componentes”: en este capítulo se detalla la construcción y validación de la solución al problema planteado. Además, se realizan y describen las pruebas de software a la aplicación.

Como **posible resultado** se espera obtener un repositorio de componentes para el departamento SENIT que permita la recolección, documentación y almacenamiento de componentes reutilizables para la construcción de portales web. Este sistema estará desarrollado en forma de una aplicación web y permitirá la gestión del contenido en forma de flujo de aprobación de los elementos que se publican en el repositorio. De igual manera como valor agregado el mismo dispondrá de un documento con todo lo referente a la documentación de la presente investigación, estando al alcance de los especialistas y trabajadores del departamento SENIT.

Capítulo 1: Fundamentos teóricos de los repositorios de componentes, funcionalidades y tecnologías para su desarrollo

En el presente capítulo se precisan un conjunto de elementos teóricos que van a posibilitar un mejor entendimiento de la presente investigación. Se realiza un estudio del estado del arte relacionado con los repositorios de componentes a nivel nacional e internacional con el objetivo de conocer las características y servicios que brindan. A partir de los resultados se analizan y seleccionan los diferentes lenguajes y herramientas para el desarrollo de la propuesta de solución, así como la metodología para guiar el proceso de desarrollo de software.

1.1 Estudio de repositorios homólogos

Con el objetivo de recopilar información acerca de repositorios existentes en el mundo, se realizó una búsqueda en directorios de repositorios como BuscaRepositorios¹, OpenDOAR², ROAR³, ROAD⁴ y en el sitio de Ranking Web de Repositorios⁵ encontrándose agrupados por distintas zonas del mundo, temáticas y por el tipo de herramienta que fue desarrollado. Se seleccionó para el estudio aquellos repositorios afines al tipo de temática relacionada con la propuesta de solución y al tipo de información que se almacena, partiendo del análisis a nivel nacional e internacional.

Repositorios Nacionales

- Repositorio de tesis doctorales

El repositorio almacena tesis de doctorado en Ciencias Biomédicas y de la Salud y de otras ramas afines que estén relacionadas o respalden las temáticas del Sistema Nacional de Salud de Cuba dirigido a

¹ Repositorios Institucionales Españoles de Acceso Abierto, <http://www.accesoabierto.net/repositorios/>

² Repositorios académicos de acceso abierto, <http://www.opendoar.org/>

³ Registro de repositorios en acceso abierto, <http://roar.eprints.org/>

⁴ Directorio escolar de recursos de acceso abierto, <http://road.issn.org/en>

⁵ <http://repositories.webometrics.info/es>

preservar y difundir las tesis de doctorado y otros tipos de categoría científicas. El mismo funciona con la plataforma EPrints 3, programa de código abierto y software libre GNU EPrints. Presenta una interfaz amigable e intuitiva al usuario, permitiéndole realizar búsquedas simples o avanzadas a partir de la selección de materia, tipo de tesis, provincia, año, número de resolución, institución, tutores y otros. Permite suscribirse al servicio de alerta para recibir por correo electrónico el aviso de nuevas tesis incorporadas.

En su página principal se visualiza funcionalidades de autenticación la cual debe ser aprobada por el administrador, menú de navegación ya sea por año, materia, especialidad doctoral, tipo de tesis o todos los campos, menús de sitios de interés, suscripción a RSS⁶ y un bloque con las últimas tesis publicadas. Contiene un menú principal con los vínculos de Inicio, Acerca de, Políticas y Ayuda.

- Repositorio Institucional de la Escuela de Hotelería y Turismo de Camagüey (EHTC)

El Repositorio Institucional de la Escuela de Hotelería y Turismo de Camagüey tiene como finalidad ofrecer acceso abierto a la producción científica y académica de la institución y asegurar su preservación digital. Presenta un diseño sencillo e información estructurada en comunidades, encontrándose en su inicio un bloque que permite elegir una comunidad EHTC para visualizar colecciones de artículos e investigaciones. Dentro de estas comunidades se puede localizar los envíos más recientes en el tema de la comunidad. Ofrece funcionalidades como publicación de comentarios, bloques de información concerniente a los contenidos publicados, información para investigadores de la EHTC y suscripción a RSS. Permite realizar búsquedas simples y avanzadas, listar artículos científicos por comunidades y colecciones, fecha de publicación, autor, título y materia; y un menú que contiene servicios de alertas, autenticación solo para usuarios autorizados, edición del perfil, ayuda y sobre EHTC.

- Repositorio Institucional de la UCI

El Repositorio Institucional (RI) de la UCI desarrollado con el software de código libre DSpace se utiliza con el objetivo de atesorar, divulgar y conservar la memoria científica de la institución. Se pueden encontrar artículos y ponencias científicas, fondo antiguo digitalizado y otros documentos en distintos formatos

⁶ *Really Simple Syndication*, un formato XML para syndicar o compartir contenido en la web. Se utiliza para difundir información actualizada frecuentemente a usuarios que se han suscrito a la fuente de contenidos

digitales. El contenido del RI se organiza alrededor de comunidades, que corresponden a las áreas temáticas de la universidad (Artículos, Eventos, Premios, Revistas, Tesis). Presenta varios menús de navegación, una guía para el autoarchivo, suscripción a RSS y búsqueda tanto simple como avanzada por diferentes criterios de filtrado como comunidades, fecha de publicación, autor, título y materia. Para la publicación y revisión de un trabajo es necesario autenticarse accediendo al vínculo Mi cuenta, además se permite suscribirse al sistema para recibir diariamente correos electrónicos de alerta sobre nuevos artículos publicados en el RI. El repositorio también cuenta en su lateral izquierdo con el panel de Ayuda el cual contiene vínculos a Licencia de uso y depósito, Políticas, Ayuda y Sobre DSpace.

- Repositorio de componentes del departamento SENIT

El repositorio de componentes para el departamento SENIT está desarrollado en el CMS Drupal y dedicado a salvaguardar componentes como módulos, códigos, arquitecturas de información, documentación, herramientas y temas para su posterior reutilización en la Fábrica de Portales. Su diseño es muy sencillo a la vista del usuario brindándose acceso a sus principales secciones: plantillas, módulos, documentación, herramientas y arquitectura de información. Presenta funcionalidades como foro de discusión, buscador simple para la localización de contenidos dentro del repositorio y servicio de autenticación, basado este último en los siguientes roles: administrador y usuario autenticado. El proceso de publicación y actualización de componentes es tedioso debido a que el administrador es el único que tiene los permisos para hacerlo, lo que obliga a los usuarios con rol autenticado contactar personalmente con este para su publicación. El repositorio no contiene una sección de contactos, así como no ofrece la posibilidad de inscribirse a boletín.

La publicación de componentes y comentarios en el foro es casi nula, en el momento en que se realizó el estudio fueron mostrados solamente 275 componentes durante los casi año y medio de funcionamiento de este. No presenta enlaces de interés a otros sitios, ni ofrece la posibilidad de conocer los últimos componentes publicados ni los más descargados. El sistema está desarrollado en la versión 7.23 del CMS Drupal, identificándose una desactualización, teniendo en cuenta las versiones más recientes que incorporan mejoras y corrigen vulnerabilidades de seguridad. Además no se cuenta con trazas de navegación que permitan al usuario conocer en que parte o sección del repositorio se encuentra posicionado.

Repositorios Internacionales

- Apache UIMA (*Unstructured Information Management Architecture* en su definición en inglés)

El proyecto Apache UIMA es una aplicación con licencia de Apache de código abierto que analizan grandes volúmenes de información no estructurada como texto, audio, video y otros, con el fin de descubrir que conocimiento es relevante para un usuario final. Se provee acceso a diferentes repositorios que comparten especificaciones, conversiones de código y herramientas. El sitio está conformado por una interfaz sencilla y una búsqueda simple, un bloque con las últimas noticias y múltiples menús y secciones para el acceso a la información como: *general, marcos de trabajo, componentes y herramientas, comunidad, desarrollo, eventos y conferencias y ASF* (en su definición en inglés *Apache Software Foundation*). Apache UIMA facilita información para iniciarse en el desarrollo a partir de la reutilización de componentes UIMA: descargas e instalación, convenciones de códigos fuente como Java y C++, especificaciones UIMA y guía para el uso de los repositorios. Se permite acceder a disímiles componentes para el análisis de información no estructurada, tales como herramientas y servidores, complementos y entornos limitados o recursos externos que están distribuidos en diferentes repositorios UIMA, siendo espacios propicios para aquellos usuarios que quieran aportar código y deseen unirse a la comunidad.

Para la publicación de componentes, se hace necesario empaquetarlos, por medio de la herramienta PEAR⁷ (*Processing Engine Archive*) que forma parte de UIMA SDK (*Software Development Kit*). PEAR emplea el formato estándar de compresión *zip* y permite reusabilidad, modularidad y desarrollo de aplicaciones basado en componentes.

- SourceCodeOnline

SourceCodeOnline es un repositorio web donde se comparte componentes de software reutilizables con fines de lucro, codificados en diferentes lenguajes de programación; ASP (por sus siglas en inglés *Active Server Pages*), HTML (por sus siglas en inglés *HyperText Markup Language*), Java, PHP (*Hypertext Preprocessor*), SQL (lenguaje de consulta estructurado o por sus siglas en inglés *Structured Query*

⁷ Estándar de UIMA para empaquetar automáticamente componentes desarrollados, tales como motores de análisis, consumidores CAS, y recursos.

Language), C/C++, además de artículos, bibliotecas, manuales, programas y código. Estos pueden ser accedidos de diversas formas desde la página principal del repositorio, estructurando la información por lenguajes de programación, ranking de códigos más vistos, nuevos componentes subidos, listado de nuevos códigos con sus descripciones y componentes más votados. Una característica de este repositorio es que no es necesario registrarse para el intercambio ni descarga de componentes, debido a que cuenta con la funcionalidad Subir Código, que muestra un formulario que describe el componente a subir, lenguaje, categoría, nombre, su versión, una breve descripción, licencia y nombre del autor, solo por citar algunas.

De manera general el repositorio presenta una estructura en forma de jerarquía de componentes, que de manera fácil y rápida permite localizarlos en el repositorio a pesar de solo contar con una búsqueda simple al ingresar una palabra clave. Al efectuarse una búsqueda se muestra un listado de coincidencias que pueden estar disponibles en el repositorio o en otros, siendo en tales casos la inclusión de un vínculo para su acceso.

- Software-artifact Infrastructure Repository (SIR)

El Repositorio de Infraestructura de Software-artefacto es un repositorio que está siendo mejorado y expandido a través de la incorporación de nuevos temas. SIR contiene artefactos relacionados con el software, como Java, C++ y C, en múltiples versiones, junto a artefactos tales como bancos de pruebas, datos de fallas y scripts. El repositorio también incluye documentación sobre el uso de los lenguajes antes mencionados, así como herramientas de apoyo. Los componentes de este repositorio cuentan con un manual, el cual describe y proporciona información sobre cómo fueron creados y su uso. Permite acceder a un listado con todos los componentes del repositorio, estando enumerados y ordenados por nombre, tipo de prueba, versión, última actualización y cantidad de descargas por mencionar algunas. Además al entrar en esta opción aparece en la parte superior una búsqueda simple y avanzada de componentes, eligiendo el lenguaje, nombre del componente y otros campos para una mayor exactitud en la búsqueda. En búsqueda avanzada, se filtra por valores máximos y mínimos de versión, tamaño, clase y matriz de fallas.

Para poder disponer de la descarga y acceso a los componentes, previamente hay que registrarse en el sistema. Dichos registros son procesados y revisados manualmente por el administrador del sitio, el cual envía una notificación al correo para que la cuenta sea activada satisfactoriamente. Al registrarse, el sistema

ofrece suscripción a RSS si lo desea, sobre noticias y actualizaciones de los componentes, dándole a escoger al usuario el tema del boletín a recibir, ya sea SIR objetos, SIR herramientas o documentación. Por otro lado, la publicación de componentes no se realiza de manera directa, si no que esta interacción se formaliza a través de correo o por medio de la opción Reporte de Problemas.

- VCLComponents

VCLComponents es un repositorio con fines de lucro, con una interfaz amigable y gran variedad de componentes, desde scripts y código fuente implementado en diferentes lenguajes de programación hasta programas y documentación referentes a los lenguajes de programación. Estos componentes pueden ser *freeware* (gratis) o *shareware* (versiones de prueba). En su página principal cuenta un menú principal que ofrece diferentes maneras de acceder y de realizar una búsqueda rápida hacia los componentes, como es: inicio, componentes nuevos, componentes populares, componentes más votados, componentes por autores, opciones de contacto, subir componente y búsqueda.

Permite que los componentes estén organizados en un directorio de lenguajes de programación como por ejemplo: Java, HTML, PHP, SQL, C#, entre otros, permitiendo comentarios en sus descripciones. Dentro de cada lenguaje es posible identificar otras categorías propias de cada lenguaje existiendo una jerarquía de componentes, por ejemplo, para el lenguaje PHP, existen las subcategorías documentación, desarrollo de herramientas, juegos, contadores, calculadora y libros, entre otros. Presenta en su página principal un listado de últimos componentes subidos, los componentes más populares y los componentes más buscados. La búsqueda se puede realizar por tipo de componente o por el lenguaje de programación. Para la publicación y recuperación de un componente en específico no es necesario estar registrado, siendo solamente necesario completar un formulario.

- ComponentSource

ComponentSource es un repositorio con fines de lucro, se dedica a la distribución de componentes, herramientas y complementos de desarrollo de software a nivel internacional. Realiza sus ventas en diferentes idiomas, lo que hace que su alcance en el mercado sea mayor. La búsqueda de los componentes tanto simple como avanzada pueden realizarse por categorías, autores, o por orden alfabético. Organiza

sus productos por tipo, plataforma o categorías. Presenta varios bloques informativos entre los que se destacan los relacionados con los productos más vendidos, más descargados y más revisados, las últimas publicaciones de componentes, aplicaciones y complementos, así como un bloque centrado en las nuevas versiones disponibles. Como principal desventaja presenta que es privativo, los componentes presentan un alto costo de adquisición y se limita su gestión solo a componentes desarrollados en Microsoft .NET/COM y Java/J2EE.

1.1.1 Resultados obtenidos del estudio

Como resultado del estudio, se concluye que los repositorios visitados presentan similitudes en cuanto a sus funcionalidades pero no satisfacen las necesidades actuales del departamento SENIT. La mayoría de estos repositorios estudiados se centran en almacenar componentes que no son característicos con los desarrollados en el departamento y en lenguajes de programación que no son utilizados para la construcción de portales web, como C++, C#, ASP.NET, Java, Delphi y Pascal. Tampoco contienen funcionalidades y características propias del departamento como exportar requisitos a Excel y Word, gestión de avisos y componentes como módulos, expedientes de proyectos, estándares de diseño, arquitecturas y temas. Por otra parte algunos de estos repositorios estudiados contemplan componentes con un alto costo de adquisición con fines de lucro.

Lo anterior evidencia la necesidad de desarrollar una nueva solución que además de contemplar las especificaciones necesarias para satisfacer las necesidades del usuario final, incluya un conjunto de funcionalidades resultado del análisis realizado a los repositorios, que a pesar de sus diferentes concepciones a la hora de estructurar los componentes y artefactos para la reutilización (Ver Anexo 2 Tabla resumen obtenida en el análisis a los repositorios web homólogos), son útiles para la propuesta de solución a desarrollar. A continuación se presentan dichas funcionalidades:

Buscador: tanto simple como avanzada es la principal funcionalidad con la que cuentan los repositorios, por lo que demanda una estricta eficiencia en la recuperación de componentes. Este servicio permite la búsqueda dentro del sitio, tanto por letras, palabras o frases, materia o año de publicación, siendo en algunos casos más profundas y estructuradas que en otros.

Autenticación: esta característica se hace necesaria e indispensable a la hora de subir algún componente en algunos repositorios, la cual se aprobará por el administrador. Existen otros casos que no es necesario este servicio para el intercambio debido a que la publicación se realiza de manera directa al llenar un formulario. La utilización o no de la autenticación, está en la manera que decida la propia institución de controlar las publicaciones de componentes su repositorio.

Bloques informativos: permiten un rápido acceso a la información de los componentes del repositorio y un grado de conocimiento de estadísticas como: componentes más votados o populares, más descargado, los más vistos, los más buscados, las últimas publicaciones.

Boletín: este servicio permite a los usuarios por medio de notificaciones al correo, estar al tanto de las últimas publicaciones y noticias subidas al repositorio. Esta suscripción solo se realiza a solicitud del usuario.

Información estructurada: atendiendo a la manera idónea y fácil de localizar la información por el usuario, esta se puede encontrar estructurada de diferentes maneras, ya sea por materia, autores, comunidades, directorio de lenguajes y componentes. Dentro de cada una de estas categorías se encuentran subcategorías de filtrado lo que evidencia una jerarquía de elementos.

Enlaces de Interés: pueden ser encontrados en el pie de página del sitio o en cualquier menú, dirigidos muchas veces a sitios que se relacionan con el repositorio visitado.

Comentar: se manifiesta al permitir dejar un comentario de manera general o específicamente en la descripción del componente en el que esté situado. En los repositorios de componentes visitados se evidencia la tendencia de utilizar este tipo de servicio con el fin de un mayor intercambio entre los usuarios y de que exista una retroalimentación.

Contacto: se presenta en forma de un formulario en el menú principal, permitiendo enviar un mensaje de contacto a los administradores del repositorio aunque también puede contener solamente las direcciones de correo de estos.

1.2 Análisis de las tecnologías, herramientas y lenguajes a utilizar

A partir del estudio de los repositorios de componentes para conocer sus características, funcionalidades y servicios que brindan se hace necesario evaluar y seleccionar las tecnologías y herramientas más idóneas así como la metodología de desarrollo a seguir para el desarrollo de la solución propuesta.

Sistema de gestión de contenidos (CMS)

Las comunicaciones y las tecnologías van evolucionando en la medida de las necesidades de informatización de las empresas e instituciones siendo necesario utilizar herramientas de software que automaticen un desarrollo web ágil, intuitivo, funcional y sencillo en su administración. De manera abarcadora estas exigencias las cumplen los Sistemas Gestores de Contenidos (CMS).

En este sentido (BILIB, 2012) y (LEÓN, 2011) definen los Sistema de Gestión de Contenidos o CMS (*Content Management System*) como herramientas informáticas estructuradas sobre la base de módulos que permiten con distintos niveles de gestión y acceso la creación, el almacenamiento, actualización, recuperación y visualización de los contenidos en una página web, así como cambiar su diseño.

Los CMS se pueden evidenciar en dos tipos, los de código abierto y código privativo. Con un auge las soluciones libres por encima de las privativas, que tienen la desventaja del sobre costo de adquisición y mantenimiento de sus licencias. A partir de lo planteado anteriormente se realiza una selección y estudio de algunos CMS de código libre utilizados y representativos a nivel mundial según (W3Techs, 2015) y en la UCI, encontrándose entre otros⁸ Drupal, Joomla y WordPress.

Drupal

Drupal⁹ es un sistema de gestión de contenido modular y configurable. Programa de código abierto programado en PHP y licencia GNU/GPL¹⁰, desarrollado y mantenido por una activa comunidad de usuarios.

⁸ Plone, Liferay, eZ Publish.

⁹ <http://drupal.org/>.

¹⁰ Licencia Pública General GNU.

Destacado por la calidad de su código y de las páginas generadas, el respeto de los estándares de la web, y un énfasis especial en la usabilidad y consistencia de todo el sistema (HISPANO, 2014). Permite trabajar con MySQL¹¹ como gestor de base de datos, aunque también puede funcionar con PostgreSQL o SQLite, y Apache o Microsoft IIS como servidor Web. Se puede encontrar en diferentes series desde la 5.x (última versión 5.23), 6.x (última versión 6.35), 7.x (última versión estable 7.37), hasta su última versión 8.0 en fase de desarrollo.

Entre sus características más destacadas se pueden encontrar según (BILIB, 2012):

- Dispone de un entorno de personalización robusto, tanto el contenido como la presentación pueden ser tratados de forma individual de acuerdo a unas preferencias definidas por el usuario.
- La gestión de contenido se realiza como objetos independientes, de forma que puede realizarse un tratamiento individualizado de la información, facilitando su inclusión en cualquier página o permitiendo comentarios específicos sobre cada uno de ellos.
- Los mecanismos de actualización de contenidos son realmente sencillos, permite editar la mayor parte de los contenidos tanto desde la interfaz de usuario como de administrador.
- Ofrece la posibilidad de gestionar las taxonomías y la estructuración de contenidos.
- Desde el punto de vista de la seguridad, la gestión de permisos destaca por encima de cualquier otra característica; ofrece un sistema muy avanzado y personalizable a nivel de rol y páginas.
- Mecanismos de control de gestión configurable para habilitar o deshabilitar módulos.

Joomla

(OPENLABS, 2014) y (BARRIOCANAL, 2009) concluyen que Joomla¹² es un CMS que permite crear sitios web dinámicos e interactivos de forma simple. Es de código abierto y gratuito, siendo programado en lenguaje PHP bajo licencia GPL que utiliza una base de datos MySQL para almacenar el contenido y los parámetros de configuración del sitio. Ofrece una estructura sólida, ordenada y sencilla a la vista de los

¹¹ Sistema gestor de base de datos relacional, multi-hilo y multi-usuario.

¹² <http://www.joomla.org>

administradores, organizando los contenidos del sitio en secciones y categorías, lo que facilita la navegabilidad para los usuarios.

Otra de las características de Joomla es la versatilidad que ofrece el sistema a través de plantillas, extensiones y adaptaciones. Existen cientos de módulos, componentes y *plugins* que extienden la funcionalidad original del CMS: gestión de archivos, gestión de contactos, sistema de búsqueda, tiendas online, bolsas de trabajo, integración con redes sociales, gestión de noticias y *newsletter*, sistemas de encuestas (BILIB, 2012).

Autores como (WEBDESIGNCUBA, 2014) y (SOMOZA, 2013) señalan como desventajas la dificultad de crear y modificar los diseños de este CMS, así como su programación también suele ser difícil y compleja. Problemas para etiquetar y crear estructuras de contenido y multilinguaje evidencian desventajas a tener en cuenta para su selección a la hora de desarrollar una aplicación web.

WordPress

WordPress¹³ es una plataforma de publicación personal, desarrollado en PHP y MySQL como servidor de base datos, distribuido bajo la licencia estándar GPL (WORDPRESS, 2014).

La hallazgos de (HOSTING, 2014) aborda algunas de las características de WordPress:

- Puede actuar como gestor de contenidos, como blog o como ambos simultáneamente, lo que permite crear sitios web empresariales con blogs corporativos, todo ello gestionado de forma sencilla.
- Sus páginas y artículos se generan dinámicamente a medida que se publica, por lo que la actualización es fácil y rápida.
- Al tener una base internacional, permite tener blogs y sitios web en casi cualquier idioma.
- Permite crear diferentes usuarios con distintos derechos de administración y edición de contenidos (administrador, editor, autor, colaborador, suscriptor).

¹³ <http://wordpress.org/>

WordPress es uno de los CMS más conocidos y utilizados en el mundo según (W3Techs, 2015), debido a su sencillez de uso e implantación hacia los blog personales. Posee diversas comunidades donde se puede obtener ayuda y asistencia a través de abundante documentación y foros.

Al examinar y tener en cuenta los resultados del estudio de los CMS se puede concluir que los tres presentan gran reputación y confiabilidad en el mercado según (W3Techs, 2015), estableciéndose funcionalidades similares entre sí. Atendiendo a las necesidades de la forma de gestión de contenido, la robustez y la finalidad a la cual se pretende llegar con el desarrollo de un repositorio de componentes se selecciona el CMS Drupal en su versión 7.34 teniendo en cuenta sus mejoras en cuanto a flexibilidad, seguridad y rendimiento.

1.2.2 Lenguajes de programación

El uso del CMS Drupal va a restringir el conjunto de lenguajes de programación, debido a que basa su funcionamiento en el lenguaje PHP.

PHP5

Es un lenguaje de programación de código abierto muy popular y adecuado para el desarrollo web. Permite la creación de aplicaciones web, se conecta a servidores de bases de datos tales como Oracle, MySQL y PostgreSQL y es multiplataforma¹⁴ (SANTOS, 2001). PHP está soportado por una gran comunidad de desarrolladores, caracterizándose por ser un lenguaje rápido y de alto nivel, totalmente compatible con los modernos métodos orientados a objetos, prácticas y principios.

HTML

Lenguaje de Marcado de Hipertexto o HTML. Lenguaje compuesto por cientos de etiquetas o marcas que permite definir el contenido y la apariencia de una página web en gran medida, además puede incluir script como JavaScript el cual afecta el comportamiento de los navegadores y otros procesadores de HTML, define

¹⁴ Sistemas operativos: Unix, Windows, Linux y Mac OS.

una estructura de documento jerárquica, con elementos y componentes interconectados (MORA, 2002). Se utiliza en su versión 5, incorporando nuevas características y etiquetas para el desarrollo web.

CSS

CSS (Hojas de estilo en cascada o *Cascading Style Sheets*) desarrollada por la W3C (*World Wide Web Consortium*), es un lenguaje de hojas de estilos creado para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML y XHTML que separa los contenidos y su presentación, lo que es imprescindible para crear páginas web complejas (EGUÍLUZ, 2009). Permite definir la apariencia de cada elemento de la aplicación web: colores, fondos, márgenes, bordes, tipos de letra, modificando la apariencia y posición de cada elemento dentro de la página, lo que permite controlar el estilo y formato de los documentos. Se hace uso en la propuesta de solución en su versión 3.

Ofrece además:

- Fácil creación de las plantillas al mantener la misma imagen en todas las páginas del sitio.
- Mayor accesibilidad y limpieza del código fuente.
- Código HTML más legible.
- Logra que los documentos se vean igual en todos los navegadores.
- Optimización de los tiempos de carga y el tráfico del servidor.

JavaScript

Los siguientes autores (MORA, 2002) y (ZAKAS, 2009) constatan que es un lenguaje interpretado que no necesita ser compilado, solo es necesario un navegador para su interpretación por lo que se ejecuta del lado del cliente. Creado con el objetivo de hacer páginas web dinámicas, es orientado a objeto y guiado por eventos que produce el mismo usuario, haciendo amena la interactividad con las páginas mediante efectos tales como: cambio de color de algunos elementos de la página, crear páginas interactivas con programas como calculadoras, agendas, tablas y calendarios.

Es un lenguaje multiplataforma siendo interpretado por la mayoría de los navegadores modernos. Es sencillo, rápido y fácil de aprender por personas de poca experiencia. Permite reducir la carga en el servidor,

ya que los datos incorrectos se validan en el cliente y no se envían al servidor. Se utiliza la versión 1.8 en el desarrollo de la propuesta de solución.

jQuery

jQuery es una biblioteca de JavaScript rápida y concisa que simplifica el trabajo con documentos HTML y XHTML, manejo de eventos, animación y las interacciones Ajax para el desarrollo web rápido. Está diseñado para cambiar la forma en que escribes JavaScript. Es multiplataforma y para comenzar a utilizarlo simplemente se debe incluir un script que contiene el código de jQuery (FOUNDATION, 2014). Para el desarrollo del repositorio se utiliza la versión 1.4.

UML

El Lenguaje de Modelado Unificado o UML por sus siglas en inglés (*Unified Modeling Language*) es un lenguaje estándar y gráfico para la modelación de sistemas de software. Está respaldado por la OMG (*Object Management Group*) y permite describir métodos o procesos, visualizar, especificar construir y documentar un sistema (STEVENS, 2007). UML es gratuito, accesible a todos, y conforma la colección de las mejores técnicas de ingeniería en el desarrollo de software, se utiliza en su versión 2.4.1.

1.2.3 Framework

Bootstrap

Es el *framework* de twitter que permite crear interfaces web con CSS y Javascript que adaptan la interfaz dependiendo del tamaño del dispositivo en el que se visualice de forma nativa, es decir, automáticamente se adapta al tamaño de un ordenador o de una Tablet sin que el usuario tenga que hacer nada, esto se denomina diseño adaptativo o *Responsive Design* (SANCHEZ, 2013). El *framework* permitió personalizar el diseño web de la propuesta de solución.

Atendiendo al autor (CUERVO, 2013) resalta características distintivas en Bootstrap como son:

- Desarrollo de aplicaciones *Responsive Design*.
- Soporte HTML5 y CSS3.

- Permite el desarrollo de aplicaciones para escritorio y móvil.
- Soporte multi-navegador, incluido Internet Explorer.
- Permite definir diferentes diseños.
- Amplio conjunto de componentes para el desarrollo.
- Posibilidad de compilar el *framework* con diferentes valores.

1.2.4 Sistemas gestores de base de datos (SGBD).

Los sistema de gestión de base de datos o en inglés *Database management system* (DBMS), según (ALVAREZ, 2007) se define como el conjunto de programas que administran y gestionan la información contenida en una base de datos, ayudando a realizar las siguientes acciones:

- Definición de los datos
- Mantenimiento de la integridad de los datos dentro de la base de datos
- Control de la seguridad y privacidad de los datos
- Manipulación de los datos

Algunos ejemplos de DBMS son Oracle, DB2, PostgreSQL, MySQL, y MS SQL Server, siendo entre ellos los más reconocidos y populares dentro del software libre MySQL y PostgreSQL (Engines, 2015). Estos dos gestores de base datos se integran al CMS Drupal haciendo de esta una excelente herramienta para el desarrollo web.

MySQL

MySQL es un sistema gestor de bases de datos muy conocido y ampliamente usado por su simplicidad y notable rendimiento. Aunque carece de algunas características avanzadas disponibles en otros SGBD del mercado, es una opción atractiva tanto para aplicaciones comerciales, como de entretenimiento precisamente por su facilidad de uso y tiempo reducido de puesta en marcha. Esto y su libre distribución en Internet bajo licencia GPL le otorgan como beneficios adicionales (no menos importantes) contar con un alto grado de estabilidad y un rápido desarrollo (CASILLAS, 2014).

Está disponible para múltiples plataformas tanto libre como privativas, es multi-hilo y multi-usuario, utilizándose como cliente-servidor o incrustado en aplicaciones. Facilita la integración con programas desarrollados en C y C++ pues está desarrollado en este lenguaje.

PostgreSQL

PostgreSQL es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD (Distribución de Software Berkeley o por sus siglas en inglés *Berkeley Software Distribution*) y con su código fuente disponible libremente. Es uno de los sistemas de gestión de bases de datos de código abierto más usado en el mercado (Engines, 2015). Utiliza un modelo cliente/servidor y usa multiprocesos en vez de multi-hilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando (MARTINEZ, 2010).

Es compatible con el almacenamiento de objetos binarios de gran tamaño, como imágenes, sonidos o vídeo. Cuenta con interfaces de programación nativas para C / C ++, Java, NET, Perl, Python, Ruby, Tcl, ODBC, entre otros. Presenta características como la replicación asincrónica y sincrónica, transacciones anidadas(puntos de retornos), “*tablespaces*”, copias de seguridad en caliente, múltiples métodos de autenticación, MVCC (Control de Concurrencia Multi-Versión) el cual permite que se ejecuten sobre una tabla varias transacciones a la vez y un sofisticado planificador consulta/optimizador (POSTGRESQL, 2014).

Permite el funcionamiento con grandes cantidades de datos y una alta concurrencia de usuarios accediendo a la misma vez al sistema. Es un sistema multiplataforma disponible para Linux y Windows y presenta una extensa y gran variedad de documentación.

Selección de Sistema gestor de BD

Dadas las características y ventajas que ambos gestores ofrecen en accesibilidad y manejo de los datos en cuanto a aplicaciones web se refiere, se concluye con la selección de PostgreSQL en su versión 9.1 como sistema gestor de base datos, por su nivel de estabilidad y rendimiento del sistema ante grandes cargas de trabajo según (PostgreSQL, 2015).

1.2.5 Servidor web

Los servidores web son aquellos cuya tarea es alojar sitios y/o aplicaciones, las cuales son accedidas por los clientes utilizando un navegador que se comunica con el servidor por el protocolo HTTP (*Hypertext Transfer Protocol*), este se mantiene a la espera de peticiones y le responde con el contenido según sea solicitado (MORALES, 2009).

Nginx

Nginx ("engine x") de código abierto, es un servidor web HTTP y proxy inverso de alto rendimiento. Aunque se encuentra en una etapa beta, es conocido por su estabilidad, configuración simple y bajo consumo de recursos, siendo liberado bajo licencia BSD; su última versión estable es la 1.8 (Nginx, 2015). Este servidor web desarrollado casi en su totalidad en lenguaje C, lo que le proporciona un alto rendimiento aprovechando al máximo los recursos del sistema. Incluye también servicios de correo electrónico con acceso al *Internet Message Protocol* (IMAP) y al servidor *Post Office Protocol* (POP). Se utiliza para balancear la carga entre los servidores *back-end*, o para facilitar almacenamiento en su caché. Entre sus características mencionar la tolerancia a fallos, el soporte de HTTP sobre SSL (acrónimo de *Secure Sockets Layer* - Protocolo de Capa de Conexión Segura) y autenticación (SERVIDORESADMIN, 2015).

Apache

Servidor web de distribución libre y de código abierto, siendo uno de los servidores más populares y utilizados en el mundo según (W3techs, 2015), permitiéndose ejecutar en múltiples sistemas operativos. Es desarrollado y mantenido por una comunidad abierta de desarrolladores bajo el auspicio de la Apache Software Foundation.

Características según (OPENSUSE, 2011):

- Es un servidor web flexible, rápido y eficiente, continuamente actualizado y adaptado a los nuevos protocolos HTTP.
- Multiplataforma.

- Es modular, lo que permite ser adaptado a diferentes entornos y necesidades, con los diferentes módulos de apoyo que proporciona, y con la API de programación de módulos, para el desarrollo de módulos específicos.
- Es extensible, teniendo en cuenta que es modular se propicia el desarrollo de diversas extensiones entre las que destaca PHP, un lenguaje de programación del lado del servidor.

Los servidores web permitirán alojar y desplegar la propuesta de solución, dándole mayor seguridad y disponibilidad en todo momento al repositorio. Finalmente se selecciona apache en su versión 2.4 debido que permite ampliar sus funcionalidades y ser adaptado a diferentes entornos y necesidades, encontrándose contemplado en la base tecnológica del centro. La popularidad de este software libre se debe a que puede ser ejecutado en una multitud de sistemas operativos, lo que lo hace prácticamente universal. Apache permite personalizar la respuesta ante los posibles errores que se puedan dar en el servidor. Es posible configurarlo para que ejecute un determinado conjunto de instrucciones cuando ocurra un error en concreto (CIBERAULA, 2014).

1.2.6 Herramientas de desarrollo

NetBeans IDE

NetBeans ¹⁵ es un Entorno de Desarrollo Integrado (IDE), modular, de base estándar (normalizado) y escrito en el lenguaje de programación Java. El proyecto NetBeans consiste en un IDE de código abierto y una plataforma de aplicación, que pueden ser usadas como una estructura de soporte general (*framework*) para compilar cualquier tipo de aplicación (ORACLE, 2013). El IDE soporta varios lenguajes de Java, C / C ++, XML y HTML, PHP, Groovy, Javadoc, JavaScript y JSP. Puede instalarse en varios sistemas operativos: Windows, Linux o Mac OS.

Ofrece diferentes vistas de los datos, desde varias ventanas de proyectos a herramientas útiles para la creación de aplicaciones y su gestión de manera eficiente, lo que le permite desglosar sus datos de forma rápida y sencilla. También proporciona plantillas de código, consejos de codificación y herramientas de refactorización. Posee APIs (del inglés *Application Programming Interface*) bien documentadas y una

¹⁵ <https://netbeans.org>.

comunidad extensa y activa de colaboradores (COMMUNITY, 2013). En la presente solución de investigación se utiliza esta en su versión 7.3.

Visual Paradigm para UML

Visual Paradigm es una herramienta CASE¹⁶ profesional para el desarrollo de aplicaciones utilizando modelado UML, utilizada por ingenieros de software, analistas de sistemas y arquitectos de sistemas interesados en construir sistemas de software fiables y estables en el desarrollo orientado a objetos.

Emplea las últimas notaciones de UML, ingeniería inversa, generación de código, importación de *Rational Rose*, exportación e importación XML. Además, brinda una gran gama de posibilidades para desarrolladores y analistas tales como: diseñar la documentación del sistema con plantillas de diseño y estimar las consecuencias de los cambios con los diagramas de análisis de impacto, como la matriz y el diagrama de análisis, respectivamente. También soporta aplicaciones web, exporta en formato HTML, está disponible en varios idiomas y es fácil de instalar y de actualizar (LARMAN, 1999). La misma se utiliza en su versión 8.0.

PgAdminIII

PgAdmin III es una aplicación gráfica con licencia Open Source para gestionar el gestor de bases de datos PostgreSQL. Está programada en C++ usando la biblioteca gráfica multiplataforma wxWidgets, lo que permite que se pueda usar en Linux, FreeBSD, Solaris, Mac OS X y Windows. Está diseñado para responder a las necesidades de todos los usuarios, desde escribir consultas SQL simples hasta desarrollar bases de datos complejas. El interfaz gráfico soporta todas las características de PostgreSQL y facilita enormemente la administración. La aplicación también incluye un editor SQL con resaltado de sintaxis, un editor de código de la parte del servidor y un agente para lanzar scripts programados. La conexión al servidor puede hacerse mediante conexión TCP/IP (Protocolo de control de transmisión/Protocolo de Internet) y puede encriptarse mediante SSL para mayor seguridad (UBUNTU, 2008).

¹⁶ Ingeniería de Software Asistida por Computador o en inglés Computer Aided Software Engineering.

Presenta una amplia documentación en diferentes idiomas, fue desarrollada por una comunidad de especialistas de la base de datos en todo el mundo. Se utiliza en la propuesta de solución la versión 1.14.

Firebug

Firebug es una extensión de Firefox creada y diseñada especialmente para desarrolladores y programadores web. Es un paquete de utilidades con el que se puede analizar (revisar velocidad de carga, estructura DOM), editar, monitorizar y depurar el código fuente, CSS, HTML y JavaScript de una página web de manera instantánea e online. Su atractiva e intuitiva interfaz, con solapas específicas para el análisis de cada tipo de elemento (consola, HTML, CSS, Script, DOM y red), permite al usuario un manejo fácil y rápido. Firebug está encapsulado en forma de *plugin* o complemento de Mozilla Firefox, es Open Source, libre y de distribución gratuita (CCESOPERU, 2010).

Es una herramienta aceptada por los desarrolladores debido a sus potencialidades en el desarrollo web. Destaca el hecho de editar y depurar el código HTML o CSS de una página y comprobar el resultado al instante, sin tener que recargar o ir al fichero que las contiene. Es utilizada en la propuesta de solución en su versión 2.0.

JMeter

JMeter es un proyecto de Apache que puede ser utilizado como una herramienta de pruebas unitarias de carga para analizar y medir el desempeño de una variedad de servicios, con énfasis en aplicaciones web. Puede ser utilizado para probar el rendimiento tanto en recursos estáticos y dinámicos (Servicios Web (SOAP/REST), lenguajes dinámicos Web-PHP, Java, ASP.NET, archivos, objetos de Java, bases de datos y consultas, FTP Servidores y más). Permite simular una carga pesada en un servidor, grupo de servidores, la red o en un objeto para poner a prueba su resistencia o para analizar el rendimiento general bajo diferentes tipos de carga. Es utilizado para hacer un análisis gráfico de rendimiento o para probar el comportamiento de scripts y objetos de servidor bajo carga pesada concurrente (JMETER, 2015). Se utiliza para la realización de las pruebas de software en su versión 2.9.

Websecurify

Websecurify es una herramienta gratuita que es utilizada para entornos de pruebas de seguridad en aplicaciones web de gran alcance, permitiendo identificar con rapidez los problemas y vulnerabilidad de seguridad del sistema. Entre las principales vulnerabilidades que puede de detectar según (Security, 2015) son: inyección de comandos, lenguaje de expresión y SQL, *Cross-site Scripting*, problemas de divulgación de información, la divulgación de ruta, servicio de SOAP descubierto y otros. Está disponible para todos los principales sistemas operativos (Windows, Mac OS, Linux) incluyendo los dispositivos móviles (iPhone, Android). Otra característica importante que es presenta una interfaz de fácil uso para los usuarios. Se utiliza en su versión 0.9 para realizar las pruebas de seguridad.

1.2.7 Metodología de desarrollo de software

La metodología de desarrollo del software constituye un proceso donde se definen técnicas y procedimientos para llevar a cabo el desarrollo de software. No existe hasta el momento una metodología que sea utilizada de forma universal. Existen diversas metodologías y cada una con sus diferencias en el grado de detalle en la manera en que gestionan con precisión los artefactos, roles y actividades del proyecto. Pero de manera general se pueden encontrar en dos grandes grupos: metodologías tradicionales (o también denominadas Metodologías Pesadas, o Peso Pesado) y metodologías ágiles.

Las tradicionales establecen durante todo el proceso de desarrollo un mayor énfasis en la planificación y control del proyecto, y en la especificación precisa de los requisitos y el modelado. Por otro lado las ágiles están más orientadas a la generación de código con ciclos muy cortos de desarrollo, enfocándose a equipos de desarrollo pequeños, haciendo especial hincapié en aspectos humanos asociados al trabajo en equipo e involucrando activamente al cliente en el proceso (PROCESOSDESFTWARE, 2014).

Teniendo en cuenta lo anterior y la necesidad de adaptarse a las exigencias y condiciones del cliente en cuanto al ahorro de tiempo y de costos en el ciclo de desarrollo y entrega de la solución con calidad, se profundizará más en el estudio de las metodologías ágiles por encima de las robustas.

OpenUp

Open Unified Process (OpenUp) es un proceso ágil y unificado, que contiene el conjunto mínimo de prácticas que ayudan a los equipos a ser más eficaces en el desarrollo de software. OpenUp abraza una filosofía pragmática y ágil que se centra en la naturaleza colaborativa de desarrollo de software. Es un proceso iterativo que es mínimo, completo y extensible que puede utilizarse tal cual o ampliarse para tratar una amplia variedad de tipos de proyecto. Se caracteriza por ser iterativo e incremental, estar centrado en la arquitectura y guiado por los casos de uso. Está organizada dentro de cuatro áreas principales de contenido: Comunicación y Colaboración, Intención, Solución y Administración.

El OpenUp está organizado en dos dimensiones diferentes pero interrelacionadas: el método y el proceso. El contenido del método es donde los elementos del método (roles, tareas, artefactos y lineamientos) son definidos, sin tener en cuenta como son utilizados en el ciclo de vida del proyecto. El proceso es donde los elementos del método son aplicados de forma ordenada en el tiempo. Muchos ciclos de vida para diferentes proyectos pueden ser creados a partir del mismo conjunto de elementos del método (HERNÁNDEZ, 2012).

Fases que propone OpenUp:

1. Inicio o Concepción: El propósito en esta fase aparte de lograr concurrencia entre todos los *stakeholders* sobre los objetivos del ciclo de vida para el proyecto, además se determina la factibilidad de la arquitectura e identificar los riesgos.
2. Elaboración: El propósito de esta fase es establecer la línea base de la arquitectura del sistema y proporcionar una base estable para el gran esfuerzo de desarrollo de la siguiente fase.
3. Construcción: Esta fase está enfocada en completar el desarrollo del sistema basado en la arquitectura definida. Se efectúan iteraciones de administración y planeación.
4. Transición: En esta fase el sistema es desplegado para los usuarios finales. Se realizan pruebas de aceptación del producto y actividades de afinación tales como la depuración de errores, mejoras en el desempeño y la usabilidad. Se genera documentación necesaria para futuras mejoras y futuros proyectos.

De forma general Open UP ofrece una metodología ágil y flexible, que se puede acoplar a la mayoría de proyectos, cubriendo aspectos como la seguridad y contratación de personal.

Metodología Programación Extrema (XP)

XP¹⁷ es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en el desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. XP se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. Esta se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico (PENADÉS, 2006).

El ciclo de vida ideal de XP consiste de las siguientes fases:

1. Exploración: En esta fase, los clientes plantean a grandes rasgos las historias de usuario. Se prueba y se familiariza el equipo de desarrollo con las herramientas, tecnologías y prácticas que se utilizarán en el proyecto.
2. Planificación de la Entrega: Se establece la prioridad de cada historia de usuario dada por el cliente, y correspondientemente, los programadores realizan una estimación del esfuerzo necesario de cada una de ellas, para posteriormente determinar un cronograma en conjunto con el cliente.
3. Iteraciones: Esta fase incluye varias iteraciones sobre el sistema antes de ser entregado elaborándose el Plan de la Iteración. Al final de la última iteración el sistema estará listo para entrar en producción.
4. Producción: Se llevan a cabo pruebas adicionales y revisiones de rendimiento antes de que el sistema sea trasladado al entorno del cliente.
5. Mantenimiento: Se realizan tareas de soporte para el cliente.
6. Muerte del Proyecto: Es la fase final del ciclo de desarrollo donde el cliente no tiene más historias de usuario a incluir en el sistema, generándose la documentación final del sistema y no permitiéndose más cambios en la arquitectura del mismo. Esta fase puede ocurrir cuando no se genera los beneficios esperados o por escasos de presupuesto.

¹⁷ www.extremeprogramming.org, www.xprogramming.com

Como se ha evidenciado esta metodología tiene como base la simplicidad y como objetivo principal la satisfacción del cliente, estableciéndose un trabajo mutuo entre este y el equipo de desarrollo. El cliente debe tener cierta comprensión de cómo alcanzar el objetivo y debe ser un actor más en el proceso de desarrollo (GRIFOL, 2012).

Metodología SXP

SXP es un híbrido, compuesta por las metodologías ágiles SCRUM y XP que ofrece una estrategia tecnológica, a partir de la introducción de procedimientos ágiles que permitan actualizar los procesos de software para el mejoramiento de la actividad productiva. Además, permite fomentar el desarrollo de la creatividad, y fortalecer el trabajo en equipo aumentando en estos el nivel de preocupación y responsabilidad, ayudando al líder del proyecto a tener un mejor control del mismo.

Esta metodología consta teniendo en cuenta a (Romero, 2011) de 4 fases principales: Planificación-Definición, Desarrollo, Entrega y Mantenimiento, cada una desglosada en flujos y actividades que generan artefactos, para la documentación del expediente de proyecto.

1. Planificación-Definición donde se establece la visión, se fijan las expectativas y se realiza el aseguramiento del financiamiento del proyecto.
2. Desarrollo, es donde se realiza la implementación del sistema hasta que esté listo para ser entregado.
3. Entrega, puesta en marcha.
4. Mantenimiento, donde se realiza el soporte para el cliente.

De cada una de estas fases se realizan numerosas actividades tales como el levantamiento de requisitos, la priorización de la lista de reserva del producto, definición de las historias de usuario, diseño, implementación, pruebas, entre otras; de donde se generan artefactos para documentar todo el proceso. Las entregas son frecuentes, y existe una refactorización continua, lo que nos permite mejorar el diseño cada vez que se le añade una nueva funcionalidad.

Es una metodología concebida para proyectos pequeños, principalmente para equipos de proyecto pequeños, vinculando al cliente como parte del equipo de trabajo. Se presta interés en el desarrollo y no en

la documentación a llenar, priorizando el tiempo a emplear por el equipo en otras tareas de importancia, tales así como la atención a las tareas docentes, de estudio independiente entre otras. Con la utilización de SCRUM para la gestión, se logra una planificación y organización inigualable; mientras que XP respalda con sus prácticas todo el proceso de desarrollo, obteniéndose de esta forma un proceso de software completo. La generación de los artefactos necesarios e imprescindibles respalda la documentación de cada uno de los sistemas, logrando así que no queden sistemas sin ser analizados y documentados. (PEÑALVER, 2010)

Selección de la metodología

Es posible concluir, a partir de las metodologías analizadas, que para guiar el proceso de desarrollo es conveniente utilizar SXP, debido a su proyección apropiada hacia proyectos pequeños y hacia el trabajo en equipo, centrada en el cliente y buenas prácticas que abarcan el ciclo completo de desarrollo del proyecto. En fundamentación de dicha selección se tuvo en cuenta las características del sistema en desarrollo y que toda metodología debe ser adaptada al contexto del proyecto.

1.3 Conclusiones parciales

Después de ser investigados los fundamentos teóricos de los repositorios de componentes, funcionalidades y tecnologías para su desarrollo se concluye lo siguiente:

1. El estudio de los repositorios tanto en el ámbito nacional como en el internacional permitió determinar las características y servicios a tener en cuenta para el desarrollo de un repositorio de componentes para el departamento SENIT.
2. Se realizó la selección del CMS Drupal para el desarrollo del sistema y la utilización de SXP como metodología para guiar el proceso de desarrollo de software.
3. La identificación de las funcionalidades y características que distinguen a los repositorios, permitieron seleccionar en conjunto con el criterio del cliente las más idóneas para la presente propuesta de solución.

4. Las tecnologías libres son idóneas para el desarrollo de aplicaciones web gracias a su flexibilidad en la libertad del código y amplia comunidad de desarrollo enfocada a la solución de errores.

Capítulo 2. Características del repositorio de componentes

En el presente capítulo se exponen las principales características del repositorio a desarrollar teniendo como precedente el estudio de repositorios homólogos y las entrevistas realizadas al cliente y a los especialistas del departamento SENIT, estableciéndose como guía para el ciclo de desarrollo del repositorio la metodología SXP. Se define el modelo de dominio y se identifican los requisitos, tanto funcionales como no funcionales con los que debe cumplir el repositorio de componentes. Además, se explica la arquitectura y los patrones de diseño utilizados para el desarrollo.

2.1 Propuesta del Repositorio de componentes

A partir del estudio de los repositorios web se determinó que estos poseen características comunes en cuanto a la manera en que estructuran y organizan la información para una mejor búsqueda a la hora de reutilizarla. Esto hizo posible que la presente solución contara con varios menús mediante los cuales se podrá acceder de manera rápida a los componentes y a los contenidos promocionados. Se mostrarán contenidos relevantes, más populares, más descargados y más votados en forma de bloques a los laterales. Para un mayor conocimiento por parte de los usuarios de los contenidos publicados en el repositorio, se contará con un sistema de notificación por correo al efectuarse una nueva publicación.

Para el desarrollo de la solución haciendo uso del CMS Drupal, se permitirá el acceso de forma sencilla para los usuarios, además de facilitar el trabajo de publicación de contenido y la gestión del mismo, proyectando una interfaz amigable y sencilla. Se contará con un foro de debate que permitirá estimular las discusiones u opiniones en línea sobre el desarrollo web entre los usuarios. La búsqueda se desarrollará de manera intuitiva, ya que será el eslabón fundamental del sistema debido al tipo de solución que se desarrolla, contando con varios criterios de filtrado que permitirán una mejor localización y reutilización de los componentes dentro del repositorio.

La propuesta contará con las características antes descritas además de posibilitar la utilización de componentes elaborados con anterioridad en el Departamento de Servicios Informáticos para Internet. Esto propicia que se agilice el desarrollo de portales y satisfacer la necesidad del departamento ante las deficiencias obtenidas en las entrevistas realizadas.

2.2 Modelo de dominio

Para la modelación de los procesos que intervienen en el sistema se utiliza el modelo de dominio, en el cual se representan los conceptos más importantes y significativos en el desarrollo de la propuesta de solución, siendo el punto de partida para el diseño de este. Constituye el medio para comprender el sector de negocios al cual el sistema va a servir. Además, permite identificar las interrelaciones entre las entidades o clases comprendidas en el ámbito del dominio del problema, al igual que sus atributos y restricciones.

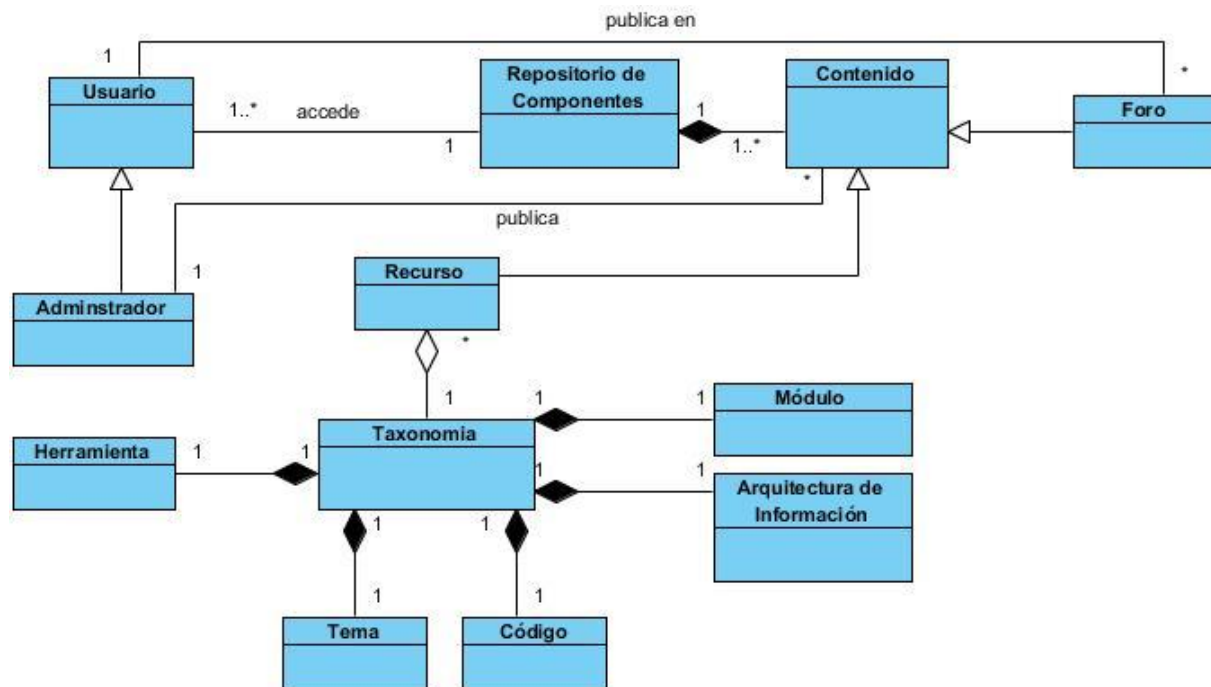


Figura 1. Modelo de Dominio

2.3.1 Descripción de las clases y roles

A continuación se explica en qué consiste cada una de las clases que conforman el modelo de dominio.

Repositorio de componentes: Sistema que llevará acabo toda la gestión de contenidos como foros y recursos, que estos últimos a su vez pueden ser módulos, herramientas, arquitectura de información, tema y código.

Recurso: Componente reutilizable para el desarrollo web, el cual puede ser descargado por los usuarios.

Foro: Contenido que posibilita la discusión entre los usuarios sobre diferentes temas de interés.

Usuario: Persona genérica que puede acceder al sistema, subir comentarios al foro y descargar componentes de su interés.

Contenido: Es cualquier tema del foro o recurso del repositorio (módulo, herramienta, arquitectura de información, tema y código) publicado por el usuario.

Taxonomía: Vocabulario para clasificar el contenido recurso en los términos módulos, herramientas, arquitectura de información, tema y código.

2.4 Levantamiento de requisitos

Una vez analizado el modelo de dominio, para un mejor entendimiento del funcionamiento del sistema se hace necesario especificar la Lista de Reserva del Producto (LRP) la cual especifica los requisitos de software que debe cumplir el sistema, partiendo de las características y cualidades que el mismo debe poseer.

A continuación se muestran los requisitos identificados, clasificándose en funcionales y no funcionales. Los funcionales permiten describir lo que el sistema debe hacer o cumplir, mientras que los no funcionales especifican cualidades sobre el propio sistema.

2.4.1 Lista de Reserva del Producto (LRP)

Tabla 1. Lista de Reserva del Producto

Código	Descripción (Requisitos Funcionales)	Prioridad
	Aviso	
RF1	Crear Aviso	Alta
RF2	Editar Aviso	Media
RF3	Eliminar Aviso	Media

RF4	Mostrar Aviso	Alta
	Requisitos	
RF6	Crear Requisito	Alta
RF6	Editar Requisito	Media
RF7	Eliminar Requisito	Media
RF8	Mostrar Requisito	Alta
	Arquitectura Tipo	
RF9	Crear Arquitectura Tipo	Alta
RF10	Editar Arquitectura Tipo	Media
RF11	Mostrar Arquitectura Tipo	Alta
RF12	Eliminar Arquitectura Tipo	Media
	Pauta del Diseño	
RF13	Crear Pauta del Diseño	Alta
RF14	Editar Pauta del Diseño	Media
RF15	Mostrar Pauta del Diseño	Alta
RF16	Eliminar Pauta del Diseño	Media
	Recurso	
RF17	Crear Recurso	Alta
RF18	Editar Recurso	Media
RF19	Eliminar Recurso	Media
RF20	Mostrar Recurso	Alta
	Contenido Populares	

RF21	Mostrar Contenidos Populares	Alta
Documentación		
RF22	Crear Documentación	Alta
RF23	Editar Documentación	Media
RF24	Eliminar Documentación	Media
RF25	Mostrar Documentación	Alta
Usuario		
RF26	Autenticar Usuario	Alta
RF27	Crear Usuario	Alta
RF28	Editar Usuario	Media
RF29	Eliminar Usuario	Media
RF30	Listar usuarios	Baja
Herramienta		
RF31	Crear Herramienta	Alta
RF32	Editar Herramienta	Media
RF33	Eliminar Herramienta	Media
RF34	Mostrar Herramientas	Alta
Módulo		
RF35	Crear Módulo	Alta
RF36	Editar Módulo	Media
RF37	Eliminar Módulo	Media
RF38	Mostrar Módulos	Alta

	Temas	
RF39	Crear Tema	Alta
RF40	Editar Tema	Media
RF41	Eliminar Tema	Media
RF42	Mostrar Temas	Alta
	Instalador	
RF43	Crear Instalador	Alta
RF44	Editar Instalador	Media
RF45	Eliminar Instalador	Media
RF46	Mostrar Instaladores	Alta
	Estándar	
RF47	Crear Estándar	Alta
RF48	Editar Estándar	Media
RF49	Eliminar Estándar	Media
RF50	Mostrar Estándares	Alta
	Expediente Tipo	
RF51	Crear Expediente Tipo	Alta
RF52	Editar Expediente Tipo	Media
RF53	Eliminar Expediente Tipo	Media
RF54	Mostrar Expedientes Tipo	Alta
	Foro	
RF55	Crear Foro	Alta

RF56	Editar Foro	Media
RF57	Eliminar Foro	Media
RF58	Mostrar Foro	Alta
Búsqueda		
RF59	Realizar Búsqueda Simple	Alta
RF60	Realizar Búsqueda Avanzada	Alta
Generales		
RF61	Mostrar enlaces a sitios de interés	Baja
RF62	Mostrar componentes más descargados	Media
RF63	Mostrar últimas publicaciones	Alta
RF64	Exportar a Word y Excel contenido requisito	Alta
Boletín		
RF65	Crear Boletín	Media
RF66	Suscribir a Boletín	Alta
RF67	Enviar Boletín	Media
RF68	Eliminar Boletín	Baja
Comentarios		
RF69	Insertar comentario	Media
RF70	Editar comentario	Baja
RF71	Eliminar comentario	Baja

RF72	Mostrar comentario	Media
RNF (Requisitos no funcionales)		
Usabilidad		
RnF-1	El sistema debe garantizar el uso de trazas de navegación, indicándole al usuario la organización de los elementos accedidos en el sistema.	
Confiabilidad		
RnF-2	Se asignarán los permisos de acceso, escritura, lectura en dependencia del rol que desempeñe cada usuario del sistema.	
RnF-3	Llevar un registro de sucesos donde se archiven los eventos del sistema incluyendo los eventos de error, inicio de sesión, cierre de sesión y modificación de la información.	
Eficiencia		
RnF-4	El sistema debe permitir la navegación de varios usuarios simultáneamente (máximo 100 usuarios del departamento SENIT) sin que influya en el rendimiento del sitio.	
Restricciones del Diseño		
RnF-5	Las páginas generadas deben cumplir con los estándares para XHTML 2.0 y CSS 3.0.	
RnF-6	El software, bibliotecas o componentes empleados deben estar regidos en la medida de lo posible por licencias y patentes de software libre.	
Interfaz		
RnF-7	La aplicación debe ser visible en navegadores como Mozilla Firefox versión superior a la 30, Chrome versión superior a la 31 y Opera 11 o superior.	
RnF-8	La comunicación entre el cliente y el servidor web será realizada a través del protocolo HTTPS (<i>Hypertext Transfer Protocol Secure</i> , siendo en español: Protocolo seguro de transferencia de hipertexto).	
Requisitos de licencia		
RnF-9	Uso de la licencia PHP License.	

RnF-10	Uso de la licencia BSD de PostgreSQL.
RnF-11	Uso de la licencia GNU/GPL para el CMS Drupal.
Estándares aplicables	
RnF-12	Se siguen el estándar de programación sugerido por los desarrolladores del CMS Drupal.
Seguridad	
RnF-13	Se garantizará la integridad y confidencialidad de la información mediante mecanismos de control de accesos no autorizados utilizando: usuario, contraseña y niveles de accesos para cada usuario, de manera que cada uno pueda tener disponible solamente las opciones relacionadas con su actividad y el acceso a datos propios.
RnF-14	Las contraseñas en la base datos estarán encriptadas.

2.4.2 Historias de Usuario

Las Historias de Usuario, es la técnica usada que permite detallar el funcionamiento o comportamiento de una funcionalidad determinada, es decir los requisitos del software. A continuación para una mejor comprensión se describe uno de prioridad para el cliente, el resto se muestran en el expediente de proyecto en el artefacto CIDI Repositorio de Componentes Descripción Historias de Usuario.

Tabla 2. Descripción de HU Crear Módulo

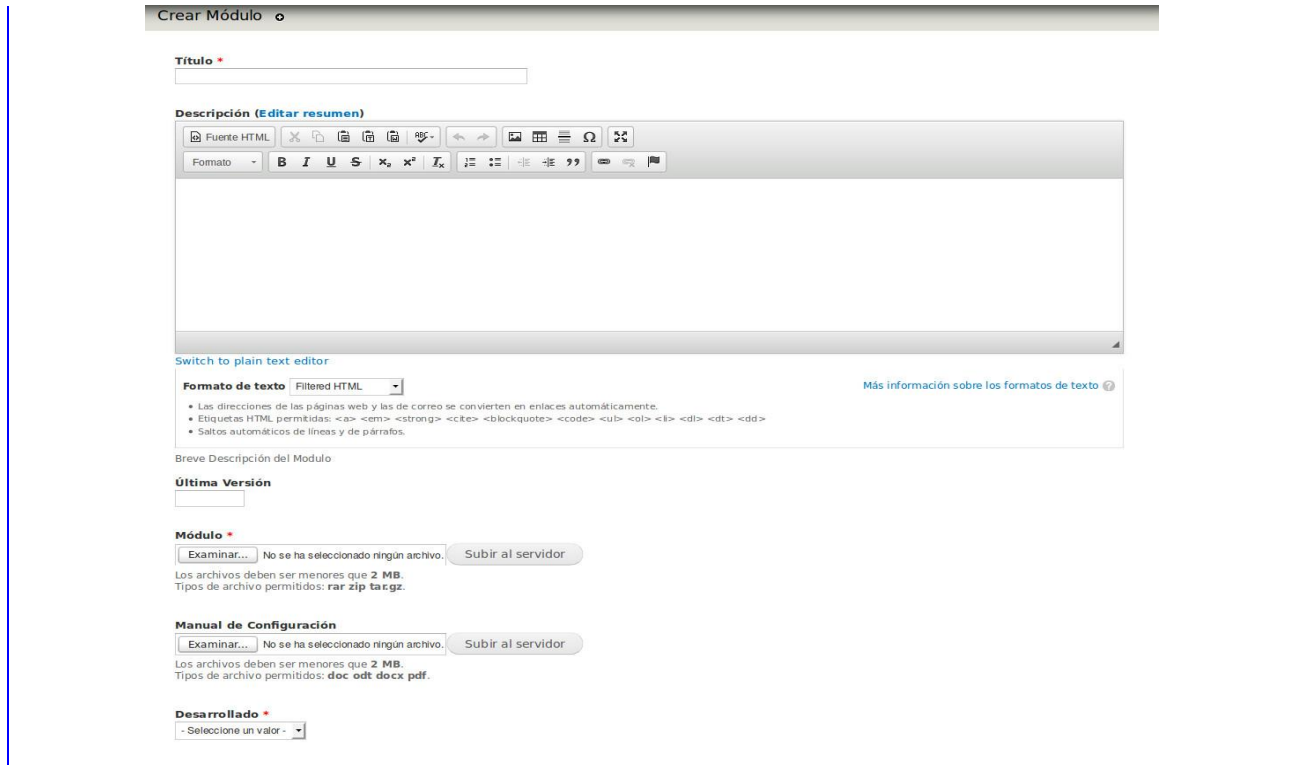
Número: 35	Nombre Historia de Usuario: Crear Módulo
Programador: Maikel Maldonado del Toro	Iteración Asignada:
Prioridad: Media	Tiempo Estimado: ¼ días
Riesgo en Desarrollo:	Tiempo Real: ¼ días
Descripción: Los usuarios autenticados pueden crear nuevos contenidos de tipo módulo en el	

sistema, para ello deben llenar los siguientes campos:

- **Nombre** (Obligatorio. Campo de texto. Longitud máxima 255 caracteres. Se admiten los caracteres: a-z, A-Z, 0-9 y caracteres especiales, no se admiten cadenas compuestas solamente por caracteres especiales).
- **Descripción** (Obligatorio. Campo de texto. Se admiten los caracteres: a-z, A-Z, 0-9 y caracteres especiales, no se admiten cadenas compuestas solamente por caracteres especiales).
- **Versión** (Obligatorio. Campo de texto. Longitud máxima 5 caracteres. Se admiten los caracteres: a-z, A-Z, 0-9 y caracteres especiales, no se admiten cadenas compuestas solamente por caracteres especiales).
- **Módulo** (Obligatorio. Campo archivo. Se admiten archivos con las extensiones rar, zip. El tamaño máximo de subida de los archivos es de 2 MB).
- **Manual de configuración** (Opcional. Campo archivo. Se admiten archivos con las extensiones doc, odt, pdf. El tamaño máximo de subida de los archivos es de 2 MB).
- **Desarrollado** (Opcional. Campo de selección. Valores (Departamento, Comunidad)).

Observaciones:

Prototipo de interfaz:



2.5 Arquitectura

El sistema implementado sobre el CMS Drupal hereda la arquitectura de este. Su arquitectura se encuentra actualmente sometida a polémica ya que su estructura no se ajusta a un único patrón de los conocidos. La presente propuesta de solución presenta una arquitectura en capas (Figura 2. Arquitectura de Drupal), identificándose en el repositorio cinco capas que son descritas a continuación:

Temas (*theme*): Esta capa establece la apariencia gráfica que se le muestra al usuario. Esta separación entre información y los estilos permite cambiar la apariencia del portal web sin necesidad de modificar los contenidos y funcionalidades implementadas. Los temas también pueden ser regulados sobre la base de permisos de usuario.

Vistas (views): Es la capa encargada de mostrar en los temas los cambios realizados a través de los módulos. Se complementan con las entidades y campos ampliando las posibilidades de presentación de contenidos dinámicos.

Entidades (entities): Representa las entidades, que engloban los nodos, los usuarios, las taxonomías y los comentarios. Esta nueva estructura permite que sea posible añadirle campos a todo aquello que sea una entidad.

Módulos (modules): Engloba los elementos, *plugins*, que operan sobre los nodos, usuarios, menús y administración, fusionándose al núcleo de Drupal para la creación de nuevas funcionalidades al sistema. Permite incrementar sus capacidades o adaptarlas a las necesidades de cada aplicación web. Ejemplo de estos módulos son: *views*, *download_count* (desarrollado por el autor de la investigación para gestionar las descargas), *datatables*, *quicktabs* y *panels*.

Base de Datos (database): Esta capa es la encargada de gestionar el acceso a la información almacenada referente al funcionamiento del sistema y a los contenidos que serán mostrados a través del tema activo.

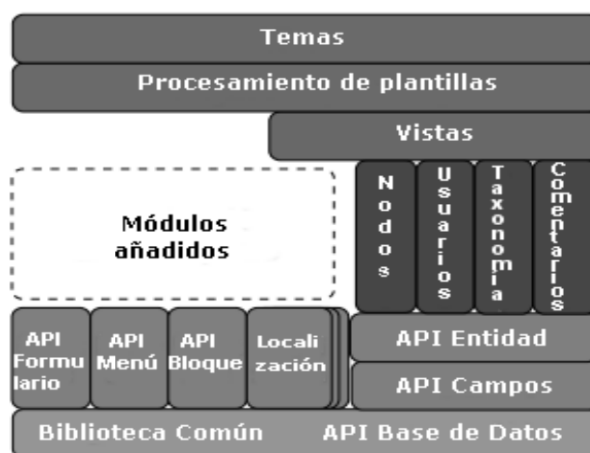


Figura 2. Arquitectura de Drupal. Estructura en 5 capas.

Patrones de Diseño

En el desarrollo de software el uso de patrones de diseño constituye una buena práctica, proporcionando una organización estructurada entendible por cualquier programador. El uso de patrones permite construir software que sean más fáciles de mantener y extender, representando un ahorro de tiempo (SHALLOWAY, 2013). A continuación se describen los principales patrones empleados en el desarrollo de la propuesta de solución:

Observer (Observador): El patrón observador es generalizado en Drupal, evidenciándose en los módulos, que implementan *hooks* (ganchos) determinados para eventos de inserción o actualización de una determinada entidad, siendo declarados como observadores de las entidades con las que interactúan.

Este patrón se evidencia en el módulo *taxonomy* cuando una modificación es realizada sobre un vocabulario en el sistema de taxonomía del repositorio, el *hook taxonomy* es llamado en todos los módulos que lo implementan, actualizando sus estados. Otro ejemplo de la evidencia de este patrón es en las vistas, cuando se modifica algún tipo de contenido de los que dispone el repositorio, el sistema informa a todas sus dependencias de la modificación realizada.

Chain of Responsibility (Cadena de Responsabilidad): El sistema de menú del repositorio de componentes sigue el patrón Cadena de Responsabilidad. En cada solicitud de la página, el menú del sistema determina si hay un módulo para gestionar la solicitud, si el usuario tiene acceso a los recursos solicitados, y que función se llama para ejecutar la petición. Si se puede ejecutar la acción, se devuelve la información solicitada, si no, pasa a la siguiente comprobación de la solicitud hasta que un módulo atienda la petición o hasta que un módulo deniegue el acceso.

Bridge (puente): La capa de abstracción de datos de Drupal se encuentra implementada siguiendo el patrón puente. Los módulos necesitan ser programados independientes del gestor de bases de datos que se esté usando. Se utiliza en el *hook simplenews_node_insert ()* que permite conectarse a la BD y realizar consultas abstrayéndose del SGBD que se emplee para manejar los datos almacenados.



2.6 Diseño

2.6.1 Diagrama de Clases de Diseño

El Diagrama de Clases es el diagrama principal de diseño y análisis para un sistema. En él, la estructura de clases del sistema se especifica, con relaciones entre clases y estructuras de herencia. Durante el análisis del sistema, el diagrama se desarrolla buscando una solución ideal. Durante el diseño, se usa el mismo diagrama, y se modifica para satisfacer los detalles de las implementaciones. (IBIBLIO, 2002)

Con el objetivo de alcanzar un mayor grado de comprensión de la aplicación a desarrollar, es indispensable tener en cuenta el funcionamiento del CMS Drupal, específicamente los paquetes que este genera. En el siguiente diagrama se muestran los mismos, con su descripción correspondiente:

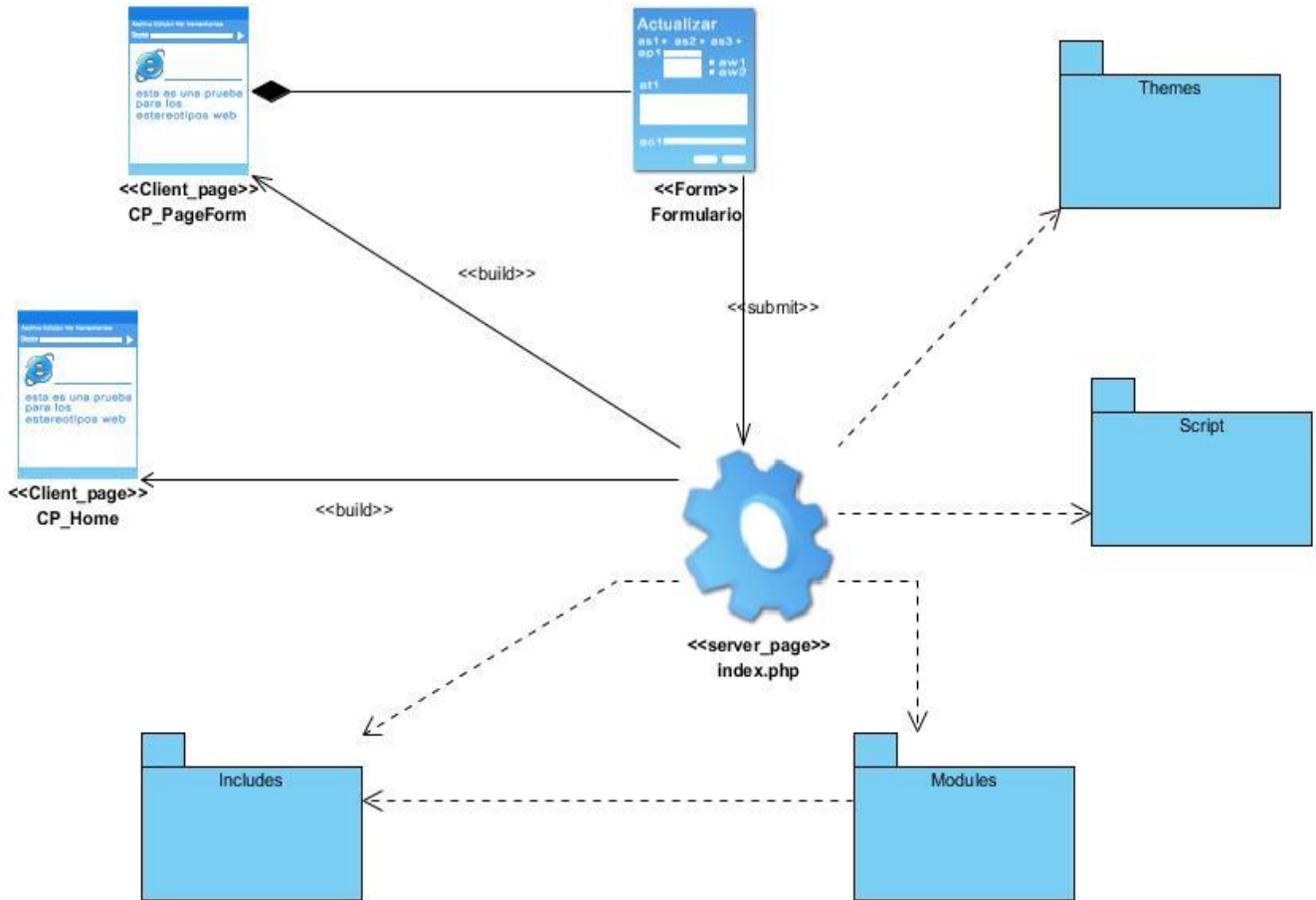


Figura 3. Diagrama de Clases de Diseño

“Includes”: En este paquete se encuentran los ficheros de conexión a la base de datos del sistema.

“Modules”: Espacio en el que se encuentran todos los módulos que forman parte del funcionamiento del CMS.

“Themes”: Agrupa las plantillas que forman parte del tema de la interfaz del sistema. Cuando se desee incluir un nuevo diseño, se copia la plantilla dentro de esta carpeta.

“Scripts”: Almacena un grupo de ficheros que contienen código para el correcto funcionamiento del sistema, permitiendo que se visualicen los datos correctamente.

Una vez analizado el diagrama de clases de diseño de la plataforma, se modelan los diagramas de clases de diseño para cada escenario como se muestra a continuación. Los restantes diagramas se ubican en el Anexo 3.

Paquete Gestionar Módulo

Entre las principales funcionalidades que debe tener implementada la solución se encuentra la de gestionar el contenido módulo. Como parte de esta gestión se incluyen las funciones de insertar un módulo nuevo, modificar un módulo existente, eliminar un módulo y mostrar un módulo. A continuación se describen los distintos escenarios:

En el caso de insertar un nuevo módulo la página servidora *SP_GestionarModulo* construye la página cliente *CP_InsertarModulo*, la cual se compone por el formulario *frm_InsertarModulo* que permite el ingreso de los datos requeridos para la creación del nuevo módulo y que va a ser ejecutado por la página servidora. Para modificar un módulo existente la página servidora crea la página cliente *CP_ModificarModulo* que contiene el formulario *frm_ModificarModulo* donde se van a modificar los datos y ser ejecutado por la página servidora; para acceder a los datos se hace mediante la clase controladora *CC_GestionarModulo*. Para eliminar un módulo se accede al formulario *frm_EliminarModulo* contenido en la página cliente *CP_EliminarModulo* la cual ha sido construida por la página servidora. Por último para mostrar los módulos la página servidora construye la página cliente *CP_MostrarModulo* que permite su visualización, teniendo en cuenta que la misma se compone por el formulario *frm_MostrarModulo*.

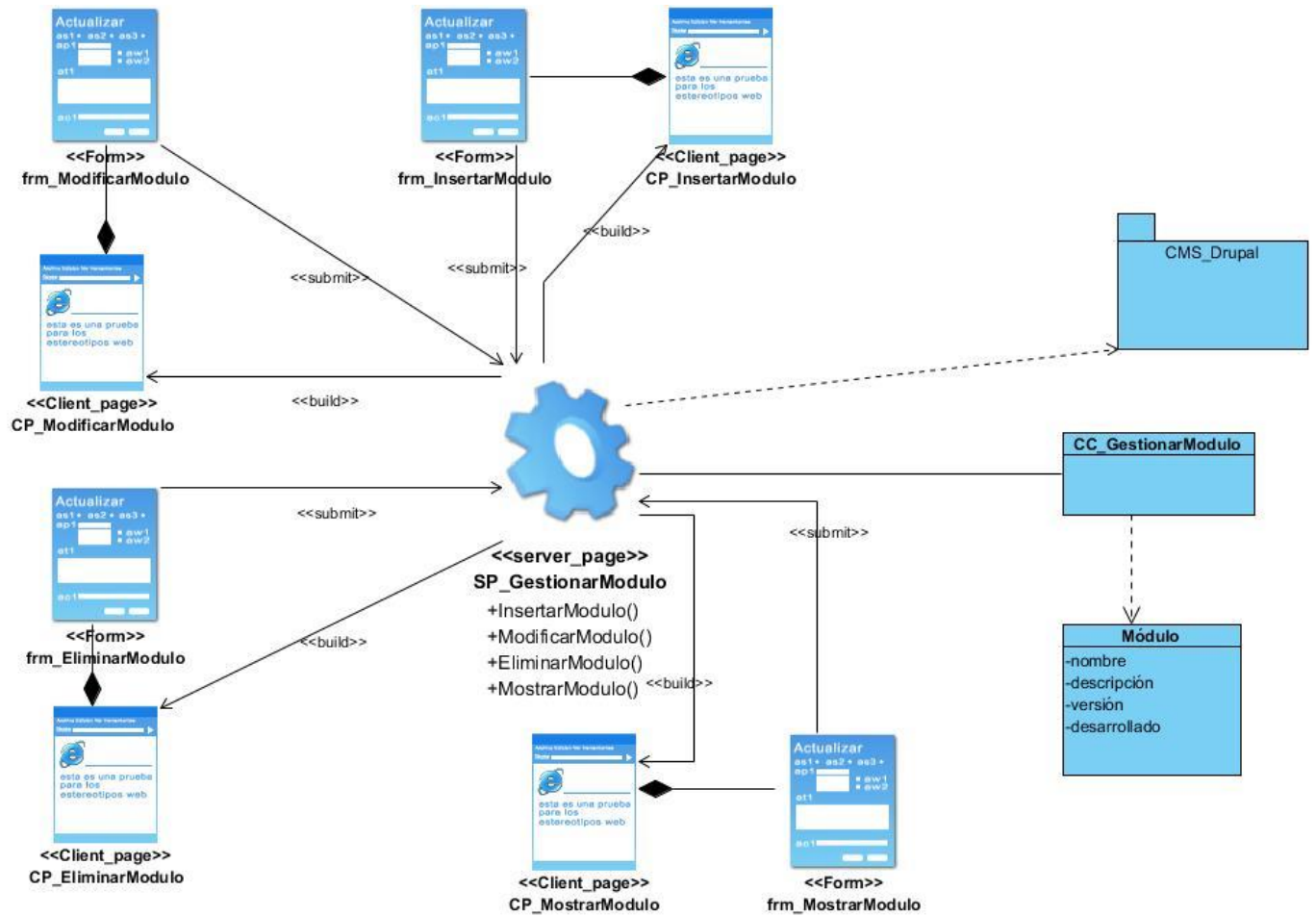


Figura 4. Diagrama de clases de diseño. Paquete Gestionar Módulo

2.7 Modelo de despliegue

El diagrama de despliegue, es utilizado para representar de forma visual las relaciones físicas que existen entre los componentes de software y hardware en el sistema. El mismo representa nodos y las relaciones entre ellos, los nodos son conectados por enlaces de red, conexiones TCP/IP, HTTP, HTTPS, entre otras.

En la figura se muestra el diagrama de despliegue correspondiente a la presente solución. El nodo “PC cliente” representa las computadoras de los usuarios que se conectan al sistema a través de un navegador

web, las cuales realizan peticiones al Servidor Web (nodo “**Servidor Web**”) mediante el protocolo HTTPS donde estará alojado el sistema. Este servidor mantendrá una conexión mediante el protocolo TCP/IP al servidor de bases de datos PostgreSQL representado por el nodo “**SGBD**” y mediante el protocolo SMTP con el servidor de correo nodo “**Servidor de Correo**” para el envío de boletines.

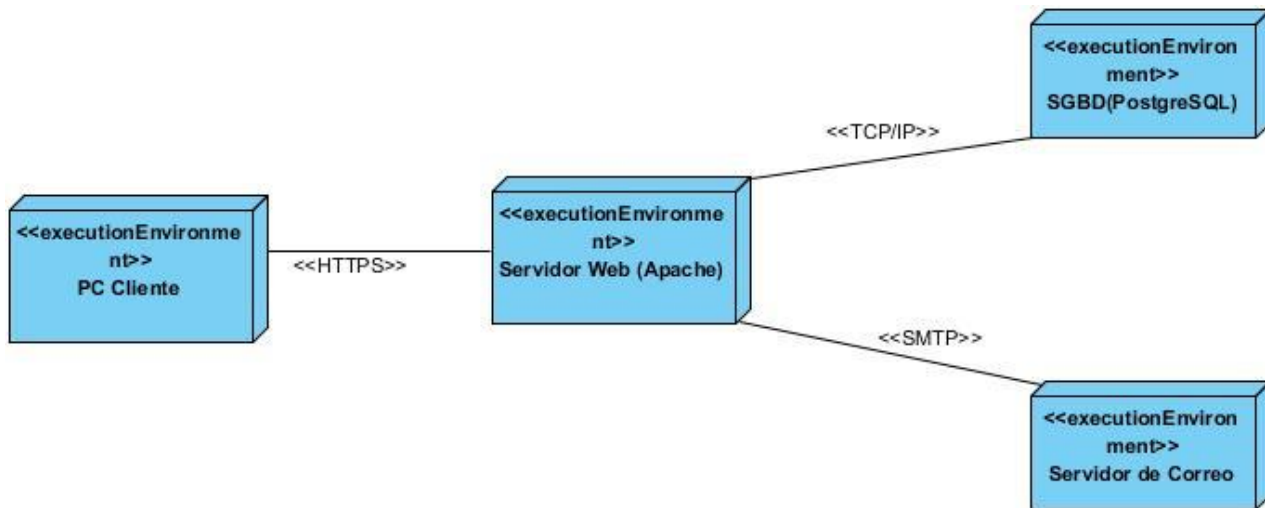


Figura 5. Diagrama de Despliegue

2.8 Conclusiones parciales

En el presente capítulo se ha podido evidenciar las principales características y elementos significativos de la arquitectura de software y de información del repositorio, así como los diferentes requisitos funcionales y no funcionales. Por lo que se puede constatar como resultado:

- La obtención de los requisitos funcionales y no funcionales obtenidos a partir del proceso de identificación de los requisitos, garantizó identificar las funcionalidades así como las características y condiciones que debe cumplir el sistema, respondiendo a las necesidades del cliente y de los usuarios finales.
- Los artefactos generados constituyeron una guía fundamental para el desarrollo del repositorio de componentes.

- El diagrama de despliegue permitió comprender las relaciones entre los componentes de hardware de la propuesta de solución.



Capítulo 3: Implementación y pruebas al repositorio de componentes

El presente capítulo describe la implementación del repositorio de componentes reutilizables para el Departamento de Servicios Informáticos para Internet, para ello se presenta el modelo de componentes y los estándares de codificación empleados. Además, los casos de prueba a los que fueron sometidas las funcionalidades del sistema y se describen las pruebas realizadas a los casos de pruebas y a la aplicación en funcionamiento.

3.1 Diagrama de componentes

Los diagramas de componentes son utilizados para estructurar el modelo de la implementación. Permiten modelar una vista estática del sistema, muestran la organización y las dependencias lógicas entre un conjunto de componentes del software, que pueden ser bibliotecas, binarios, ejecutables y códigos fuentes. A continuación se representa dicho diagrama para la solución que se desarrolla:

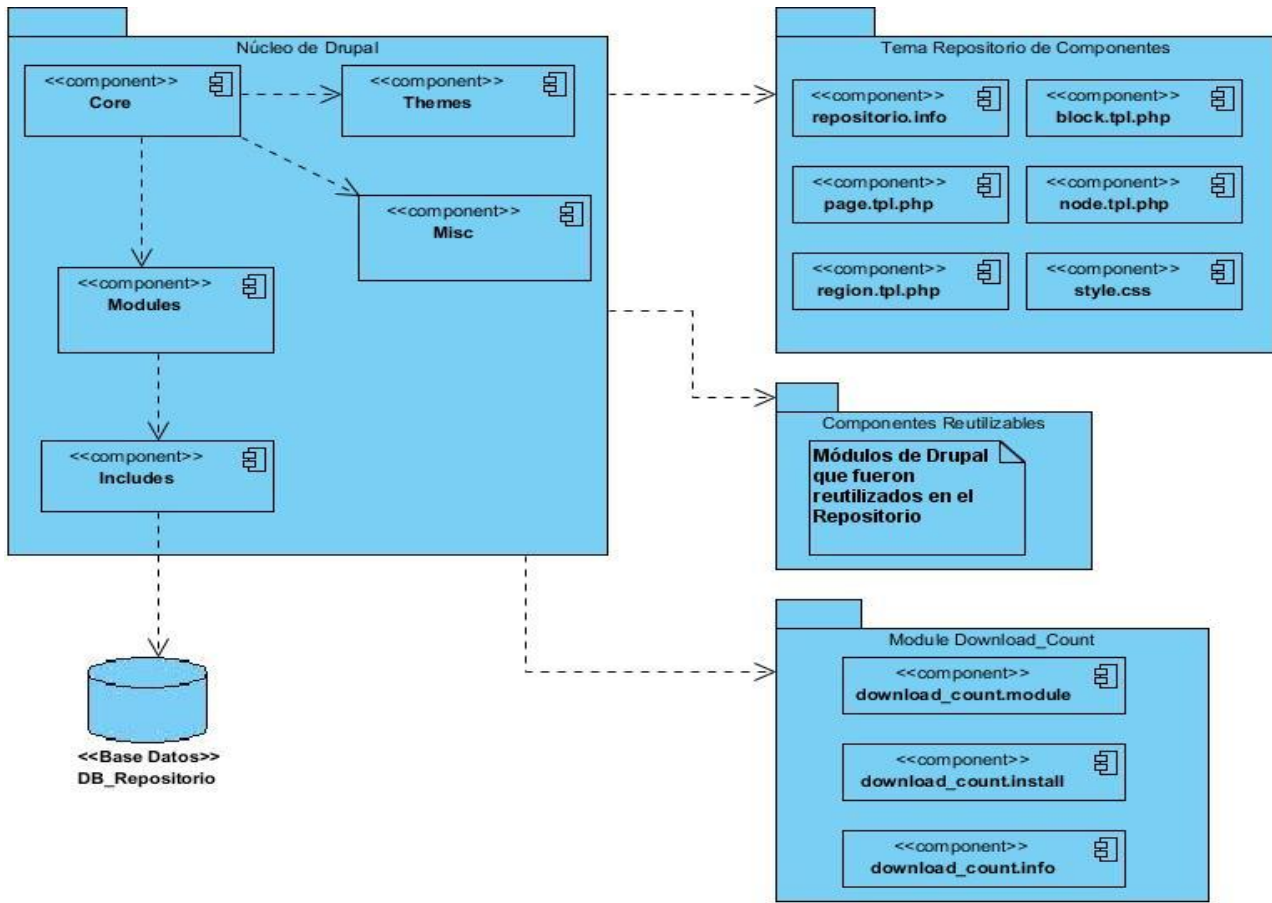


Figura 6. Diagrama de Componentes

La tabla que se muestra a continuación describe cada uno de los componentes representados en el diagrama.

Tabla 3. Descripción de componentes

COMPONENTES	DESCRIPCIÓN
<i>Themes</i>	Se encuentra toda la capa de presentación y diseño.
Tema Repositorio de Componentes	Nuevo tema creado para el repositorio.

<i>repositorio.info</i>	Este archivo guarda información sobre las regiones disponibles, los archivos CSS y JavaScript.
<i>block.tpl.php</i>	Es un archivo de plantilla que contiene código HTML y variables PHP, que serán sustituidas por sus valores correspondientes al presentar el contenido de la página, siendo en este caso la estructura de los bloques del sitio.
<i>page.tpl.php</i>	Es un archivo de plantilla que contiene código HTML y variables PHP, que serán sustituidas por sus valores correspondientes al presentar el contenido de la página, encargándose de definir la estructura HTML de cualquier página del sitio.
<i>node.tpl.php</i>	Es un archivo de plantilla que contienen código HTML y variables PHP, utilizadas para definir la estructura de un tipo de contenido.
<i>style.css</i>	Archivo principal que contiene los estilos del tema.
<i>Core</i>	Es el punto de inicio del portal, a partir de esta entrada se invocan los diferentes módulos del CMS.
<i>Modules</i>	Directorio donde se encuentran los módulos de Drupal.
<i>download_count.module</i>	Incluye el código del módulo, en forma de funciones PHP. Es en este archivo donde se realizarán las llamadas a los <i>hooks</i> (ganchos), creando las funciones que los implementan.

<i>download_count.install</i>	Archivo de instalación del módulo.
<i>download_count.info</i>	Contiene información básica sobre el módulo (nombre, descripción, versión de Drupal, etc.), requisitos mínimos (por ejemplo, una versión de PHP determinada), posibles dependencias con otros módulos de Drupal, paquete al que pertenece el módulo y ficheros que incluye el módulo.
<i>Include</i>	Este componente contiene un conjunto de ficheros indispensables para el funcionamiento de Drupal, tales como funciones y scripts necesarios para el funcionamiento de los módulos y del sistema.
<i>Misc</i>	Incluye elementos que tienen que ver con el diseño, y funcionamiento (imágenes, ficheros js, etc).
Módulos Desarrollados	Contiene los módulos desarrollados para el repositorio de componentes.
BD Repositorio	Es la base de datos del repositorio.

3.2 Estándares de codificación

Los estándares de codificación constituyen el elemento principal por el cual se rigen los desarrolladores para asegurar que el código sea legible para cualquier equipo de trabajo, lo que permite que las tareas realizadas sean más efectivas y contengan menos errores (W3C, 2015).

La solución desarrollada utiliza como estándares para la codificación en los lenguajes utilizados, aquellos que son propuestos por los desarrolladores o comunidades internacionales que crearon estos lenguajes.

Como ejemplo se puede citar que para la codificación en HTML y CCS se utilizaron los estándares¹⁸ sugeridos por la W3C¹⁹. Aunque es importante resaltar que para el desarrollo en Drupal se sigue el estándar de programación sugerido por el sitio oficial del CMS²⁰. Este estándar destaca elementos como los siguientes:

- Se sigue el estándar de programación sugerido por Drupal.
- Cuando se esté escribiendo en PHP, siempre se debe utilizar las etiquetas `<?php` y `?>`, y en ningún caso la versión corta `<? y ?>`.
- Cada sentencia condicional debe llevar sus respectivas llaves.
- En los ficheros `.module` y `.inc` se omite la etiqueta de cierre de PHP (`?>`).
- En los archivos de plantilla `.tpl.php`, cada fragmento de PHP debe llevar sus correspondientes etiquetas de apertura y cierre, para diferenciarlo del código HTML.
- Se pueden usar tanto las comillas simples ('cadena') como las comillas dobles ("cadena") para delimitar las cadenas de caracteres.
- Los nombres de las funciones deben estar escritos en minúsculas y las palabras separadas por guión bajo.
- El nombre de un módulo nunca debería incluir guiones bajos, aunque se componga de varias palabras.
- Todas las clases definidas deben tener el siguiente formato: `.clase-definida { } #clase-definida { }`.

¹⁸ <http://www.w3.org/standards/>

¹⁹ El Consorcio *World Wide Web* (W3C) es una comunidad internacional para el desarrollo de estándares web.

²⁰ <https://www.drupal.org/coding-standards>

3.3 Pruebas

Las pruebas de software son aquellos procedimientos que se realizan para verificar la calidad de un producto de software y pueden ser aplicadas periódicamente. Estas tienen como objetivo fundamental la identificación de posibles errores que se presenten en la aplicación, además de mostrar hasta qué punto las funciones del software operan de acuerdo con las especificaciones y requisitos del cliente (ORÉ, 2009). Entre las pruebas realizadas al sistema se pueden mencionar:

Pruebas de Caja Negra

Las Pruebas de Caja Negra se desarrollan sobre la interfaz visual del software y se encarga de verificar que las funciones que debe desempeñar el sistema son operativas. Se centran en los requisitos funcionales del software, sin internarse en el funcionamiento interno de la misma. Mediante la realización de estas pruebas se pueden encontrar errores de interfaz, funciones incorrectas, errores de salida y problemas con el acceso a datos. Estas pruebas permiten obtener un conjunto de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un programa (ORÉ, 2009).

Pruebas de Sistema

Tienen como objetivo fundamental verificar el sistema de software para ver si cumple con sus requisitos. Dentro de esta fase pueden desarrollarse distintos tipos de pruebas, integración, pruebas unitarias, pruebas de rendimiento, pruebas de seguridad, entre otras. Se centra en comprobar que los sistemas desarrollados funcionan acorde a las especificaciones funcionales y requisitos del cliente. En sí, son pruebas específicas, concretas y exhaustivas para probar y validar que el software hace lo que debe y sobre todo, lo que se ha especificado (HERNAN, 2005).

A continuación se presentan los resultados obtenidos después de aplicar las pruebas al sistema.

3.3.1 Funcionales

Para la realización de estas pruebas se utilizó la técnica de caja negra las cuales son realizadas desde el punto de vista de la interfaz de usuario, lo que permitió valorar el funcionamiento de la aplicación según la interacción del usuario con el portal.

A continuación se describen los casos de prueba para algunos requisitos con prioridad Alta:

Condiciones de ejecución: El usuario debe estar autenticado.

Tabla 4. Caso de pruebas. Escenario Crear Módulo

Escenario	Descripción	Nombre	Descripción	Versión	Desarrollado	Módulo	Manual de configuración	Respuesta del sistema	Flujo central
EC 3.1 Crear Módulo correctamente	El usuario introduce los datos correctos del módulo.	V	V	V	V	V	V	El sistema verifica que los datos estén correctos, de ser así, almacena el módulo en la base de datos.	El usuario selecciona la opción "Añadir Módulo" en la sección Módulos de la vista Desarrollo. El sistema muestra una interfaz para insertar el
		"Voting Rules"	"Reglas de votación que ofrece integración de Reglas para el módulo VotingAPI...."	"7.x-1.0"	"Comunidad"	"voting.rar" De tamaño 2 MB	"config_voting.pdf" De tamaño 2 MB		
EC 3.2 Crear Módulo incorrectamente.	El usuario introduce los datos del módulo	I	I	I	I	I	I	El sistema verifica que los datos estén correctos, de no ser así, el	
		"Voting45*& Rules"	(vacío)	"versión 7.x-1.*"	"Departamento 43\$*"	"voting.tar" De tamaño 2 MB	"config_voting.rtf" De tamaño 2 MB		



	incorrectos.							sistema muestra un mensaje indicando el error.	módulo. El usuario introduce los datos y selecciona el botón "Guardar".
--	--------------	--	--	--	--	--	--	--	---

Tabla 5. Caso de pruebas. Escenario Editar Módulo

Escenario	Descripción	Nombre	Descripción	Versión	Desarrollado	Módulo	Manual de configuración	Respuesta del sistema	Flujo central
EC 3.1 Editar Módulo correctamente	El administrador modifica los datos correctos del módulo.	V "Voting Rules"	V "Reglas de votación que ofrece integración de Reglas para el	V "7.x-1.0"	V "Comunidad"	V "voting.rar" De tamaño 2 MB	V "config_voting.pdf" De tamaño 2 MB	El sistema verifica que los datos estén correctos, de ser así, almacena el módulo en	El usuario selecciona la opción "Editar Módulo". El sistema muestra una interfaz para Editar el módulo. El usuario

			módulo VotingA PI...."					la base de datos.	introduce los datos y selecciona el botón "Guardar".
EC 3.2 Editar módulo incorrectamente.	El administrador modifica los datos del módulo incorrectos.	"Voting45*&Rules"	N/A	"versión 7.x-1.*\$\$"	(vacío)	(vacío)	"config_voting.rtf" De tamaño 2 MB	El sistema verifica que los datos estén correctos, de no ser así, se muestra un mensaje indicando el error.	

Tabla 6. Caso de Prueba. Escenario Eliminar Módulo

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 3.1 Eliminar Módulo correctamente	En la funcionalidad de eliminar no existen variables solo se deben seguir los	El sistema muestra un mensaje de confirmación de la acción: "¿Seguro que desea eliminar el módulo Voting Rules?". Esta acción no se puede deshacer. El usuario	El administrador, el cual tiene los permisos pertinentes accede a la página principal y en la Vista de Administración se muestran los módulos existentes en el sistema. El

	pasos del flujo central.	selecciona la opción “eliminar” y el sistema elimina el módulo de la base de datos.	usuario selecciona la opción “eliminar”.
EC 3.2 Cancelar la operación.	En la funcionalidad de cancelar no existen variables solo se deben seguir los pasos del flujo central.	El sistema deshace la operación y muestra el listado de los módulos existentes en el sistema.	El administrador el cual tiene los permisos pertinentes accede a la página principal y en la Vista de Administración se muestran los módulos existentes en el sistema. El sistema le permite eliminar uno o varios módulos y cancelar la operación. El usuario selecciona la opción “Cancelar”.

Tabla 7. Casos de prueba. Variables Escenario Gestionar Módulo

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	Nombre	Campo de texto	No	(Obligatorio. Campo de texto. Longitud máxima 255 caracteres. Se admiten los caracteres: a-z, A-Z, 0-9 y caracteres especiales, no se admiten cadenas compuestas solamente por caracteres especiales).
2	Descripción	Campo de texto	No	(Obligatorio. Campo de texto. Se admiten los caracteres: a-z, A-Z, 0-9 y caracteres

				especiales, no se admiten cadenas compuestas solamente por caracteres especiales).
3	Versión	Campo de texto	No	(Obligatorio. Campo de texto. Longitud máxima 5 caracteres. Se admiten los caracteres: a-z, A-Z, 0-9 y caracteres especiales, no se admiten cadenas compuestas solamente por caracteres especiales).
4	Desarrollado	Campo de selección	No	(Campo de selección. Valores (Departamento, Comunidad)).
5	Manual de Configuración	Fichero	Si	(Opcional. Campo archivo. Se admiten archivos con las extensiones doc, odt, pdf. El tamaño máximo de subida de los archivos es de 2 MB)
6	Módulo	Fichero	No	(Obligatorio. Campo archivo. Se admiten archivos con las extensiones rar, zip. El tamaño máximo de subida de los archivos es de 2 MB)

Resultados de las pruebas funcionales

Se probaron todas las funcionalidades que responden a los casos de pruebas del repositorio de componentes y se obtuvieron un total de 27 no conformidades las cuales fueron todas resueltas, siendo los principales errores detectados en errores ortográficos, textos en idioma inglés y campos no validados.

La siguiente gráfica desglosa las no conformidades detectadas en las dos iteraciones realizadas. En una primera iteración se recoge un total de 18 no conformidades de las cuales se resolvieron las 18 y en una segunda iteración se detectaron 9 no conformidades, resolviéndose en su totalidad las 9.

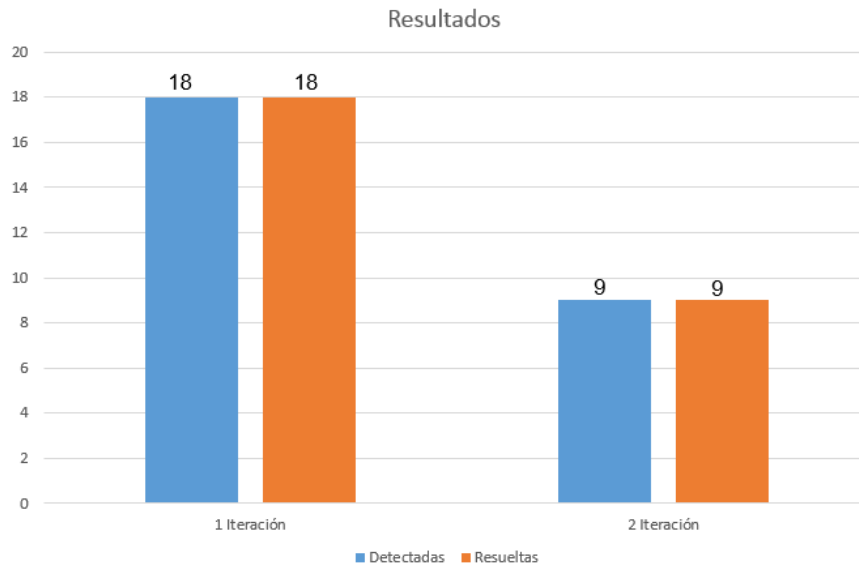


Figura 7. Resultados de las pruebas funcionales

3.3.2 Seguridad

La seguridad es un aspecto importante a tenerse en cuenta en todo sistema, debido que protege sus intereses y beneficios para que no sufra de ataques de un intruso informático, por eso es de suma importancia saber cuan seguro se encuentra un software. Teniendo en cuenta lo planteado anteriormente, en el desarrollo de todo sistema se aplican pruebas de seguridad que garantizan que los usuarios estén restringidos a funcionalidades específicas o su acceso esté limitado únicamente a los datos que están autorizados a acceder.

Las pruebas de seguridad aplicadas a la aplicación fueron realizadas con la herramienta Websecurify, la cual es un entorno integrado para pruebas de seguridad web. También se realizó un chequeo de elementos de seguridad mediante la lista de chequeo para Prueba de Seguridad diseñadas por el Laboratorio Industrial de Pruebas de Software perteneciente a la empresa CALISOFT.

Resultados de las pruebas de seguridad

El resultado de estas pruebas indicó un buen nivel de seguridad del sistema pues la herramienta Websecurify solo arrojó 1 error (Ver Anexo 4 Reporte de Websecurify), el cual fue resuelto al modificar en el fichero `/etc/apache2/conf-enabled/security.conf` la variable `ServerSignature` en `Off` y la variable `ServerTokens` en `Prod`. Como se muestra en la Figura 8. Resultados de las pruebas de seguridad, 3 de 4 indicadores evaluados poseen un 100% de seguridad y 1 indicador posee un 85.71% de seguridad debido a que no se cumple el 6to aspecto de los 7 evaluados en este indicador (Ver Anexo 5 Listas de Chequeo para Pruebas de Seguridad). Su solución estuvo dada en la instalación del módulo *Secure Login*.

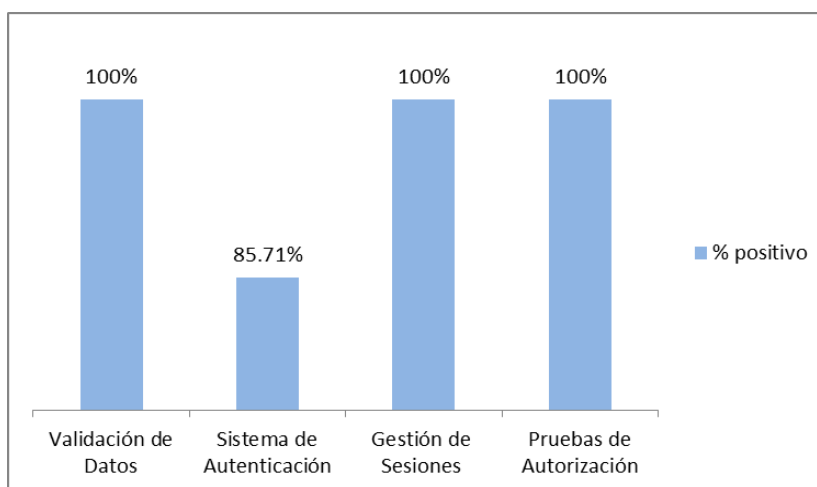


Figura 8. Resultados de las pruebas de seguridad

3.3.3 Carga y Estrés

Prueba de carga: Las pruebas de carga miden la capacidad del sistema para continuar funcionando apropiadamente bajo diferentes condiciones de carga. Su meta es determinar y asegurar que el sistema funciona apropiadamente aún más allá de la carga de trabajo máxima esperada (HERNAN, 2005). En otros términos esta prueba determina y valida la respuesta de la aplicación cuando es sometida a una carga de usuarios y/o transacciones que se espera en el ambiente donde se desplegará el sistema.

Prueba de estrés: Enfocada a evaluar cómo el sistema responde bajo condiciones extremas. (Extrema sobrecarga, insuficiente memoria, servicios y hardware no disponible, recursos compartidos no disponible)

(HERNAN, 2005). El objetivo de este tipo de pruebas es conocer el límite de carga de trabajo al que podemos llevar el sistema, lo que hace posible que podamos tomar medidas antes de llegar a ese límite.

Resultados de las pruebas de carga y estrés

Las pruebas de rendimiento fueron realizadas en la herramienta JMeter, diseñada para pruebas de carga de comportamientos funcionales y la medición del rendimiento. Prueba la resistencia y analiza el rendimiento en diferentes tipos de carga. Las pruebas se desarrollaron en un entorno que cumple con las siguientes características: 1 servidor con un procesador Intel Core Duo y 3GB de RAM.

Teniendo en cuenta que los usuarios pueden realizar diferentes acciones al conectarse al repositorio se diseñó el plan de pruebas de rendimiento. Para una muestra de 30, 50 y 100 usuarios conectados concurrentemente, con un período de subida de 1 segundo (tiempo que espera cada usuario para realizar una petición) la aplicación generó un 0 % de error de peticiones realizadas al sistema en los tres casos, con un tiempo promedio de acceso a una página de 308, 544 y 1140 milisegundos respectivamente, realizándose un total de 7620, 1280 y 2550 peticiones respectivos al servidor con un rendimiento de 82.5/sec, 81.5/sec y 81.1/sec peticiones por segundo en cada una de las muestras siendo satisfactorias las pruebas. En el Anexo 6 se encuentra el reporte resumen generado por la herramienta JMeter en las tres muestras.

3.4 Conclusiones parciales

Durante el proceso de implementación y realización de pruebas al repositorio de componentes se concluye lo siguiente:

- El diagrama de componentes definido permite apreciar de forma visual la organización y las dependencias lógicas entre componentes de software del sistema.
- Los resultados obtenidos para el entorno en que fueron aplicadas las pruebas de software se consideran satisfactorios.

- Las pruebas realizadas validan el funcionamiento de la aplicación desarrollada según los requisitos definidos.

Conclusiones Generales

- 1) El estudio y caracterización de repositorios homólogos propició la necesidad de desarrollar una nueva propuesta de solución que integre las funcionalidades comunes en cada uno de los repositorios estudiados con el propósito de desarrollar una herramienta que satisfaga las necesidades del departamento SENIT.
- 2) La definición de las tecnologías, las herramientas y la metodología para la implementación del repositorio de componentes, permite afirmar que el uso de herramientas y tecnologías libres, y de metodologías ágiles: como es SXP, aseguran la independencia tecnológica del sistema desarrollado, evitan la elaboración de documentación innecesaria y disminuyen el tiempo de desarrollo de la solución en cuestión.
- 3) Se obtuvo un repositorio web, desarrollado con tecnologías libres, con funcionalidades de gestión y búsqueda de componentes por medio de un entorno amigable que propicia la reutilización y preservación de los componentes en el departamento SENIT.
- 4) La propuesta de solución resultante de esta investigación contribuye a fortalecer el modelo organizativo: Fábrica de Portales del centro CIDI y a elevar la productividad de los procesos de desarrollo de portales web.
- 5) Las pruebas de software efectuadas al producto permitieron identificar y corregir las no conformidades contribuyendo así la calidad del repositorio desarrollado.

Recomendaciones

1. Continuar trabajando en el desarrollo de nuevas funcionalidades de la presente solución a partir de nuevas necesidades que surjan.
2. Trazar una política de actualizaciones periódicas utilizando el módulo de actualización del CMS Drupal, con el objetivo de evitar fallas de seguridad.
3. Desarrollar la variante del repositorio para dispositivos móviles.

Referencias

SHALLOWAY, A. y TROTT, J. *Design Patterns Explained. A new perspective on object oriented design.* Addison Wesley, 2013. 258 p.

Álvarez Gómez, Yasmani. *Experiencias en el desarrollo de portales web y mercadotecnia en el centro CIDI .Comunicación Personal (Asesor de Mercadotecnia) 2015.* Centro de Ideo-Infornática. Universidad de las Ciencias Informáticas, La Habana Cuba.

ALVAREZ, S. *Sistemas gestores de bases de datos* [en línea]. [Consultado el: 19 Octubre 2014]. Disponible en: <http://www.desarrolloweb.com/articulos/sistemas-gestores-bases-datos.html>.

BARRIOCANAL, L. *Edujoomla.es - ¿Qué es joomla?* [en línea]. [Consultado el: 9 Octubre 2014]. Disponible en: <http://www.edujoomla.es/que-es-joomla>.

BILIB, F. P. C. Y. T. D. A. Estudio de los sistemas de gestión de contenidos web. Análisis de las mejores soluciones del mercado. *Junta de Comunidades de Castilla La Mancha*, 13 Julio 2012 2012, nº p. 107.

CASAÑOLA, Y. T. *Modelo de Factoría aplicando Inteligencia.* Master, Universidad de las Ciencias Informáticas, 2007.

CCESOPERU. *Firebug* [en línea]. [Consultado el: 20 Octubre 2014]. Disponible en: <http://s3.accesoperu.com/wp6/wp6.php?p=13350>.

CIBERAULA. Una Introducción a Apache [en línea]. [Consultado el: 28 de mayo del 2015]. Disponible en: http://linux.ciberaula.com/articulo/linux_apache_introl/.

COMMUNITY, N. *NetBeans IDE Features* [en línea]. [Consultado el: 20 Octubre 2014]. Disponible en: <https://netbeans.org/features/index.html>.

COMPONENTSOURCE. *What are components?* [en línea]. [Consultado el: 20 Mayo 2014]. Disponible en: <http://www.componentsource.com/services/about-us/components-es.html>.

CUERVO, V. *Hola Mundo con Bootstrap.* [en línea]. 2013, [Consultado el: 16 Octubre 2014]. Disponible en: <http://lineadecodigo.com/bootstrap/hola-mundo-con-bootstrap/>.

EGUÍLUZ, J. *Introducción a CSS* [en línea]. [Consultado el: 15 Octubre 2014]. Disponible en: <http://librosweb.es/css/index.html>.

ENGINES. DB-Engines Ranking - popularity ranking of database management systems [en línea]. [Consultado el: 29 de mayo del 2015]. Disponible en: <http://db-engines.com/en/ranking>.

- FOUNDATION, J.** *jQuery Project* [en línea]. [Consultado el: 15 Octubre 2014]. Disponible en: <https://jquery.org/>.
- FERNÁNDEZ, M.** SENIT-Repositorio de Componentes Reutilizables [en línea]. [Consultado el: 28 de mayo del 2015]. Disponible en: <http://10.53.30.219/>.
- GRIFOL, D.** *Metodologías de desarrollo ágil: Programación Extrema* [en línea]. [Consultado el: 2014-10-21]. Disponible en: <http://danielgrifol.es/metodologias-de-desarrollo-agil-programacion-extrema>.
- HERNÁNDEZ, Y. A.** *CONFIGURACIÓN DE LA METODOLOGÍA OPENUP*. Universidad de las Ciencias Informáticas, Ciudad de la Habana, 2012,
- HERNAN, J.** Ingeniería de Software: TIPOS DE PRUEBAS DE SOFTWARE [en línea]. [Consultado el: 29 de mayo del 2015]. Disponible en: <http://ing-sw.blogspot.com/2005/04/tipos-de-pruebas-de-software.html>.
- HISPANO, D.** *Sobre Drupal | Drupal Hispano* [en línea]. [Consultado el: 9 Octubre 2014 de 2014]. Disponible en: <http://drupal.org/es/drupal>.
- HOSTING, E.** *Características del gestor de contenidos WordPress | Esfera Hosting* [en línea]. [Consultado el: 10 Octubre 2014]. Disponible en: <http://www.esferahosting.com/wordpress>.
- IBIBLIO.** *Modelado de Sistemas con UML - Capítulo 4. Un estudio a fondo de UML* [en línea]. [Consultado el: 11 Octubre 2014]. Disponible en: <http://www.ibiblio.org/pub/linux/docs/LuCaS/Tutoriales/doc-modelado-sistemas-UML/multiple-html/x219.html>
- MARTÍNEZ, I. y LUIS, F.** *Un Modelo de Mediación para el Desarrollo de Software basado en Componentes COTS*. Tesis Doctoral. Universidad de Málaga. España, 2003.
- JMETER, A.** Apache JMeter - Apache JMeter™ Apache software foundation [en línea]. [Consultado el: 25 de mayo del 2015]. Disponible en: <http://jmeter.apache.org/>
- LARMAN, C.** *UML y patrones: introducción al análisis y diseño orientado a objetos*. Editado por: Educación, P. PHH, 1999. 507 p. ISBN 9789701702611.
- LEÓN, Y. R.** Sistemas gestores de contenidos: una mirada desde las ciencias de la información. *Revista Cubana de Información en Ciencias de la Salud*, 2011/07/06 2011, vol. 22, nº 1, [Consultado el: 8 Octubre 2014]. Disponible en: <http://www.acimed.sld.cu/index.php/acimed/article/view/153>. ISSN 2307-2113.
- SANTILLÁN, L. A. C.; GINESTÀ, M. G., et al.** Bases de datos en MySQL. 2014, nº p. 67. [Consultado el: 19 Octubre 2014]. Disponible en: http://ocw.uoc.edu/computer-science-technology-and-multimedia/bases-de-datos/bases-de-datos/P06_M2109_02151.pdf. ISSN 02151.

- MARTINEZ, R.** *Sobre PostgreSQL* [en línea]. [Consultado el: 19 Octubre 2014]. Disponible en: http://www.postgresql.org.es/sobre_postgresql.
- MONTILVA, J.; ARAPÉ, N., et al.** Desarrollo de software basado en componentes. Actas del IV. Congreso de Automatización y Control. Mérida, Venezuela, 2003, nº [Consultado el: 28 de mayo del 2015]. Disponible en: <http://webdelprofesor.ula.ve/ingenieria/jonas/Productos/Publicaciones/Congresos/CAC03%20Desarrollo%20de%20componentes.pdf>.
- MORA, S. L. RUA:** *Programación de aplicaciones web: historia, principios básicos y clientes web* [en línea]. Universidad de Alicante. Departamento de Lenguajes y Sistemas Informáticos: Editorial Club Universitario, [Consultado el: 15 Octubre 2014]. Disponible en: <http://rua.ua.es/dspace/handle/10045/16995>.
- MORALES, P. A. A.** *Servidores Web* [en línea]. [Consultado el: 20 Octubre 2014]. Disponible en: <http://www.monografias.com/trabajos75/servidores-web/servidores-web.shtml>.
- NGINX.** Nginx Community [en línea]. [Consultado el: 28 de mayo del 2015]. Disponible en: <http://wiki.nginx.org/Main>.
- OPENLABS.** *Características del CMS Joomla* [en línea]. [Consultado el: 9 Octubre 2014]. Disponible en: <http://www.openlabs.com.mx/portal/servicios-adicionales/desarrollo-web/caracteristicas-de-cms-joomla>.
- OPENSUSE.** *Apache - openSUSE* [en línea]. [Consultado el: 20 Octubre 2014]. Disponible en: <https://es.opensuse.org/Apache>.
- ORACLE.** *Información NetBeans IDE 6.1* [en línea]. [Consultado el: 20 Octubre 2014]. Disponible en: https://netbeans.org/community/releases/61/index_es.html.
- ORÉ, A.** *Introducción al Software Testing and QA* [en línea]. [Consultado el: 28 de mayo del 2015]. Disponible en: http://www.calidadyssoftware.com/testing/introduccion_software_testing.php.
- ORÉ, A.** *Pruebas Funcionales - Software Testing and QA* [en línea]. [Consultado el: 29 de mayo del 2015]. Disponible en: http://www.calidadyssoftware.com/testing/pruebas_funcionales.php.
- PAZ, W.** *Reingeniería* [en línea]. [Consultado el: 26 Octubre 2014]. Disponible en: <http://prezi.com/3eonr9nvanhn/reingenieria/>.
- PELECHANO, J. M. V.** *MDA vs Factorías de Software. Actas del II Taller sobre Desarrollo de Software Dirigido por Modelos, MDA y Aplicaciones (DSDM 2005). Granada (Spain), 2005, vol. 13, nº p. 1.*

- PENADÉS, P. L. Y. M. C.** *Métodologías ágiles para el desarrollo de software: eXtreme Programming (XP)*. Laboratorio de Sistemas de Información. Departamento de Sistemas Informáticos y Computación. Facultad de Informática. Universidad Politécnica de Valencia: 2006.
- PEÑALVER, R.; MENESES, A., et al.** SXP, METODOLOGÍA ÁGIL PARA EL DESARROLLO DE SOFTWARE 1er Congreso Iberoamericano de Ingeniería de Proyectos, Chile, 2010, nº [Consultado el: 27 de Febrero del 2015]. Disponible en: <http://usbvirtual.usbcali.edu.co/ijpm/images/stories/documentos/v1n2/009.pdf>.
- STEVENS, P. y POOLEY, R.** *Utilización de UML*. 2007. vol. 2ª ED,
- POSTGRESQL.** *PostgreSQL: About* [en línea]. [Consultado el: 20 Octubre 2014]. Disponible en: <http://www.postgresql.org/about/>.
- POSTGRESQL.** PostgreSQL: PostgreSQL 9.1 Press Kit [en línea]. [Consultado el: 28 de mayo del 2015]. Disponible en: <http://www.postgresql.org/about/press/presskit91/es/>.
- PROCESOSDESOFTWARE.** *ProcesosdeSoftware - METODOLOGIAS PARA DESARROLLO DE SOFTWARE* [en línea]. [Consultado el: 20 Octubre 2014]. Disponible en: <http://procesosdesoftware.wikispaces.com/METODOLOGIAS+PARA+DESARROLLO+DE+SOFTWARE>.
- RAE.** *Real Academia Española* [en línea]. Real Academia Española. In, [Consultado el: 21 Octubre 2014 de vol. 2010]. Disponible en: <http://www.rae.es/rae.html>.
- ROMERO, G. M. P.; PUENTE, S. J. G. D. L., et al.** *SXP, metodología de desarrollo de software. Serie Científica de la Universidad de las Ciencias Informáticas, 2011, vol. 4, nº 11, [Consultado el: 26 de Febrero del 2015].* Disponible en: <http://publicaciones.uci.cu/index.php/SC/article/view/429/482>
- SANCHEZ, A. F.** *¿Que es Bootstrap?* [en línea]. Open Web CMS, [Consultado el: 16 Octubre 2014]. Disponible en: <http://openwebcms.es/2013/que-es-bootstrap/>.
- SANTOS, H. H.** *¿Qué es el PHP?* [en línea]. [Consultado el: 15 Octubre 2014]. Disponible en: <http://www.maestrosdelweb.com/phpintro/>.
- SECURITY, W.** *Web Security Vulnerabilities* [en línea]. [Consultado el: 25 de mayo del 2015]. Disponible en: <http://www.websecrify.com/vulnerabilities.html>.
- SERVIDORESADMIN.** *Nginx Servidor Web Open Source - Servidores Dedicados Administrados* [en línea]. Disponible en: <http://www.servidoresadmin.com/nginx/>.
- SOMOZA, J. M.** *Ventajas y desventajas de los CMS (I) - Joomla* [en línea]. [Consultado el: 9 Octubre 2014]. Disponible en: <http://www.pymenetworks.es/blog/ventajas-y-desventajas-de-los-cms-i-joomla>.

- SZYPERSKI, C.** *Component Software: Beyond Object Oriented Programming*. 2da Edición. Addison-Wesley Edition, 2002.
- TERREROS, J. C.** Desarrollo de Software basado en Componentes [en línea]. [Consultado el: 28 de mayo del 2015]. Disponible en: <https://msdn.microsoft.com/es-es/library/bb972268.aspx>.
- UBUNTU, G.** *PgAdmin III - Guía Ubuntu* [en línea]. [Consultado el: 20 Octubre 2014]. Disponible en: http://www.guia-ubuntu.com/index.php/PgAdmin_III.
- UCI.** Portal de la Universidad de las Ciencias Informáticas [en línea]. [Consultado el: 28 de mayo del 2015]. Disponible en: <http://www.uci.cu/>.
- VÁZQUEZ, S. R. y ALGUACIL, L. S.** Arquitectura organizacional para soluciones empresariales de software. Revista Cubana de Ciencias Informáticas, 09/2013 2013, vol. 7, nº 3, p. 1-13. [Consultado el: 26 Octubre 2014]. Disponible en: http://scielo.sld.cu/scielo.php?script=sci_abstract&pid=S2227-18992013000300001&lng=es&nrm=iso&tlng=es. ISSN 2227-1899.
- VIGNAGA, A. y PEROVICH, D.** Enfoque Metodológico para el Desarrollo Basado en Componentes Instituto de Computación Facultad de Ingeniería Universidad de la República, 2003, vol. 21, nº p. 25. [Consultado el: 28 de mayo del 2015]. Disponible en: <http://www.fing.edu.uy/inco/lab/dotnet/material/relacionado/vp03.pdf>.
- WEBDESIGNCUBA.** *Ventajas y Desventajas de Joomla* [en línea]. [Consultado el: 9 Octubre 2014]. Disponible en: <http://www.webdesigncuba.net/ventajas-y-desventajas-de-joomla>.
- WORDPRESS.** *es:Características de WordPress* « *WordPress Codex* [en línea]. [Consultado el: 10 Octubre 2014]. Disponible en: http://codex.wordpress.org/es:Caracter%C3%ADsticas_de_WordPress.
- W3TECHS.** W3Techs - extensive and reliable web technology surveys [en línea]. [Consultado el: 28 de mayo del 2015]. Disponible en: <http://w3techs.com/>.
- W3C.** World Wide Web Consortium (W3C) [en línea]. [Consultado el: 15 de marzo del 2015]. Disponible en: <http://www.w3.org/>.
- ZAKAS, N. C.** *Professional JavaScript for Web Developers*. Editado por: Indianápolis, D. E. Wiley Publishing, 2009.

Bibliografía Consultada

ALEGSA, L. ¿Cuál es la definición de SGBD? [en línea]. [Consultado el: 19 Octubre 2014]. Disponible en: <http://www.alegsa.com.ar/Dic/sqbd.php>.

ALICANTE, U. D. Repositorios [en línea]. CI2 Universidad de Alicante, [Consultado el: 24 Octubre 2014]. Disponible en: https://moodle2012-13.ua.es/moodle/pluginfile.php/53300/mod_resource/content/1/Fuentes%20de%20informacio%CC%81n%20en%20Internet/pagina_03.htm.

ARTIGOS, C. D. NetBeans, Historia, NetBeans Platform, NetBeans IDE, Módulos integrados, NetBeans IDE Descargar Bundles, Localización [en línea]. [Consultado el: 20 Octubre 2014]. Disponible en: http://centrodeartigo.com/articulos-informativos/article_69622.html.

BARRADAS, I. Metodología Ágil OpenUP. [en línea]. 2010, [Consultado el: 20 Octubre 2014]. Disponible en: <http://yuliethbarradas.blogspot.com/2010/04/metodologia-agil-openup.html>.

BOLIVARIANA, U. U. XP - Extreme Programing Ingeniería de Software [en línea]. Universidad Unión Bolivariana: [Consultado el: 21 Octubre 2014]. Disponible en: http://ingenieriadesoftware.mex.tl/52753_XP--Extreme-Programing.html.

BOOTSTRAP. BOOTSTRAP [en línea]. [Consultado el: 16 Octubre 2014]. Disponible en: <http://ingsw.ccbas.uaa.mx/sitio/images/material/bootstrap.htm>.

COMMONS, C. Apuntes. Ingeniería del software. Sistemas Informáticos. Nivel de madurez software. Informática Aplicada a la Gestión Pública. 2005/06-2. Universidad de Murcia. Rafael Barzanallana [en línea]. [Consultado el: 20 de Octubre del 2014]. Disponible en: <http://www.um.es/docencia/barzana/LAGP/lagp2.html>.

CUERDA, X. G. Mosaic | Introducción a los Sistemas de Gestión de Contenidos (CMS) de código abierto [en línea]. [Consultado el: 9 Octubre 2014]. Disponible en: <http://mosaic.uoc.edu/2004/11/29/introduccion-a-los-sistemas-de-gestion-de-contenidos-cms-de-codigo-abierto/>. ISBN 1696-3296.

DEFINICIONABC. Definición de MySQL » Concepto en Definición ABC [en línea]. [Consultado el: 19 Octubre 2014]. Disponible en: <http://www.definicionabc.com/tecnologia/mysql.php>.

DEV, G. NetBeans IDE entorno de desarrollo para lenguajes como Java PHP C/C++ Groovy [en línea]. [Consultado el: 20 Octubre 2014]. Disponible en: <http://www.genbetadev.com/herramientas/netbeans-1>.

GONZALEZ, J. 6 Características Que Hacen Tan Popular A Bootstrap. [en línea]. 2014, [Consultado el: 16 Octubre 2014]. Disponible en: <http://jagonzalez.org/6-caracteristicas-que-hacen-tan-popular-a-bootstrap/>.

GROUP, T. P. G. D. PostgreSQL 9.3.0 Documentation. Regents of the University of California: PostgreSQL Global Development Group, 2013.

GUTIÉRREZ, J. J. *Qué es un framework web?* [en línea]. Escuela Técnica Superior de Ingeniería Informática, Universidad de Sevilla, [Consultado el: 16 Octubre 2014]. Disponible en: www.lsi.us.es/javierj/investigacion_ficheros/Framework.pdf.

GUZMÁN, C. L. Los Repositorios de Objetos de Aprendizaje como soporte para entornos e-learning. Doctoral, In. España: Universidad de Salamanca, 2005.

HISPANO, D. Sobre Drupal [en línea]. [Consultado el: 9 Octubre 2014]. Disponible en: <http://drupal.org.es/caracteristicas>

IDICT. Inteligencia Empresarial qué y cómo. La Habana: Instituto de Información Científica y Tecnológica, 2009. ISBN 978-959-234-070-1.

JUAREZ, E. A. O. Open UP: Metodología Open UP. [en línea]. 2013, [Consultado el: 20 Octubre 2014]. Disponible en: <http://openupeaojmp.blogspot.com/2013/09/metodologia-open-up.html>.

LEGRÁ, J. L. H. Repositorio de Componentes para el Departamento de Señales Digitales Trabajo de Diploma, Universidad de las Ciencias Informáticas, 2011.

LOTURA. 10 razones por las que no utilizar Joomla para desarrollar una web. [en línea]. 2012, [Consultado el: 10 Octubre 2014]. Disponible en: <http://blog.lotura.com/10-razones-por-las-que-no-utilizar-joomla-para-desarrollar-una-web/>.

MASTERMAGAZINE. Definición de MySQL - Significado y definición de MySQL [en línea]. [Consultado el: 19 Octubre 2014]. Disponible en: <http://www.mastermagazine.info/termino/6051.php>.

PERDITA STEVENS, R. P. Utilización de UML. 2007. vol. 2ª ED,

PGADMIN. pgAdmin: Feature [en línea]. [Consultado el: 20 Octubre 2014]. Disponible en: <http://www.pgadmin.org/features.php>.

PHILIPPE KRUTCHEN , P. K. Rational Unified Process. 2da Edición. Addison –Wesley Edition, 2002.

POSTGRESQL. PostgreSQL: Características, limitaciones y ventajas. [en línea]. 2014, [Consultado el: 20 Octubre 2014]. Disponible en: <http://postgresql-dbms.blogspot.com/p/limitaciones-puntos-de-recuperacion.html>.

PROGRAMMING4. PgAdmin III 1.10.0 [en línea]. [Consultado el: 20 Octubre 2014]. Disponible en: <http://mscerts.programming4.us/es/16904.aspx>.

SÁNCHEZ, I. R. A. y VERA, I. A. Bibliografía Biomédica. Editorial Ciencias Médicas, 2008. ISBN 978-959-212-295-6.

SUÁREZ, M. V. Repositorio de componentes para la línea de Soluciones para Intranets y Portales Trabajo de Diploma, Universidad de las Ciencias Informáticas, 2012.

TECLEA. ¿Qué sistema de Gestor de Contenidos (CMS) utilizar? Agencia de marketing en Madrid y Logroño [en línea]. [Consultado el: 9 Octubre 2014]. Disponible en: <http://www.teclea.com/tecnologia/que-sistema-de-gestor-de-contenidos-cms-utilizar/>.

UNIVERSO. UniVerso: El Periódico de los Universitarios - No. 486 [en línea]. Dirección de Comunicación Universitaria Departamento de Prensa. Universidad Veracruzana, [Consultado el: 21 Octubre 2014]. Disponible en: http://www.uv.mx/universo/486/infgral/infgral_15.html.

UPTODOWNN. Firebug - Descargar [en línea]. [Consultado el: 20 Octubre 2014]. Disponible en: <http://firebug.uptodown.com/>.

V., J. Lo nuevo en Firebug 1.12 [en línea]. [Consultado el: 20 Octubre 2014]. Disponible en: <http://www.mozilla-hispano.org/lo-nuevo-en-firebug-1-12/>.

WEBVIGO. Introducción a Bootstrap: front-end framework - Diseño Web Vigo. [en línea]. 2012, [Consultado el: 16 Octubre 2014]. vol. 2014, Disponible en: <http://www.webvigo.com/blog/introduccion-a-bootstrap-front-end-framework/>.

WIIG, K. M. Knowledge management: where did it come from and where will it go? Expert systems with applications, 1997, vol. 13, nº 1, p. 1-14.

WIIG, K. M.; HOOG, R. D., et al. Supporting knowledge management: a selection of methods and techniques. Expert systems with applications. 1997. vol. 13,

WHATIS. What is Web server? - Definition from WhatIs.com [en línea]. [Consultado el: 20 Octubre 2014]. Disponible en: <http://whatis.techtarget.com/definition/Web-server>.

YUKILEMA, F.; TORRES, P., et al. DRUPAL - características [en línea]. [Consultado el: 15 Octubre 2014]. Disponible en: <http://www.slideshare.net/pabloto/drupal-caracteristicas>.

ZAKAS, N. C. Professional JavaScript® for Web Developers. Editado por: Indianápolis, D. E. Wiley Publishing, 2009.

Glosario de términos

Ajax: Acrónimo de JavaScript asíncrono y XML, es una técnica de desarrollo web para crear aplicaciones interactivas.

GNU: Licencia Pública General de GNU o más conocida por su nombre en inglés *GNU General Public License* (o simplemente sus siglas del inglés GNU GPL) fue un proyecto iniciado por Richard Stallman con el objetivo de crear un sistema operativo libre.

Marco de trabajo: patrón arquitectónico que proporciona una plantilla extensible para aplicaciones dentro de un dominio.

Módulo: En informática un módulo es una parte autónoma de un programa de ordenador.

Repositorio: La palabra repositorio proviene del latín *repositorium*, que significa armario. La mayoría de los diccionarios de lengua española lo definen como: “Almacén o lugar donde se guardan ciertas cosas” (RAE, 2009). Aunque actualmente ha sido enriquecido con el desarrollo de la era digital, sosteniéndose por algunos autores la definición de que es “un sitio centralizado donde se almacena y mantiene información digital” (GUZMÁN, 2005).

Repositorio de componentes: Los repositorios de componentes como entidades, constituyen la base de componentes reutilizables con que cuenta la factoría del software con el objetivo de dar soporte al proceso de reutilización que se lleve a cabo en el proyecto e influir positivamente en el desarrollo de software.

Reutilización de software: “Se define la reutilización de software como el uso de cualquier tipo de artefacto (también llamado activo), o parte del mismo, creado con anterioridad, en un nuevo proyecto (PAZ, 2012).”

Stakeholders: *stakeholders* sería parte interesada (del inglés *stake*, apuesta, y *holder*, poseedor). Se puede definir como cualquier persona o entidad que es afectada o concernida por las actividades o la marcha de una organización; por ejemplo, los trabajadores de esa organización, sus accionistas, las asociaciones de vecinos afectadas o ligadas, los sindicatos, las organizaciones civiles y gubernamentales que se encuentren vinculadas.

URL: Acrónimo de *Uniform Resource Locator* (Localizador de Recursos Uniforme), es una secuencia de caracteres, de acuerdo a un formato modélico y estándar, que se usa para nombrar recursos en Internet para su localización o identificación, como por ejemplo documentos textuales, imágenes, vídeos, presentaciones digitales, dirección de una página web, etc.