

*Universidad de las Ciencias Informáticas*

*Facultad # 1*



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

*Título: Sistema de gestión de productos de software de la  
Red Social de la Universidad de las  
Ciencias Informáticas*

*Autor: Luis Alfredo Borrero Torres*

*Tutores: Ing. Yusliel García Vázquez  
Ing. Damaris Cano López*

*La Habana, Junio de 2012.  
"Año 54 de la Revolución."*



*“Si la inspiración no viene a mí salgo a su encuentro, a la mitad del camino.”*

*Sigmund Freud*

## **Declaración de autoría**

Declaro ser autor de la presente tesis y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los \_\_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

Luis Alfredo Borrero Torres

---

Firma del Autor

Ing. Yusliel García Vázquez

---

Firma del Tutor

Ing. Damarys Cano López

---

Firma del Tutor

### *Agradecimientos*

*Resulta muy difícil en poco más de una simple cuartilla agradecer a todo aquel que de una forma u otra brindó su granito de arena para hacer posible esta difícil meta. A todas aquellas personas que me brindaron su apoyo amor y cariño gracias de todo corazón.*

*Agradezco a mi familia por todo su apoyo y en especial a mi mamá que siempre ha estado a mi lado en cualquier situación dando todo de ella sin importar lo que sea para que yo sea feliz y pueda lograr todo lo que me proponga. A mi niñita Soilet que trajo mucha felicidad a mi vida y cada día al despertar siento la sensación de que una personita te quiere y te dan deseos de dar el alma y la vida por ella. A mi abuelo Oscar por todo lo que me enseñó, todo su cariño y ser la persona que más influyó en mi carácter, donde quieras que estés gracias siempre te llevo conmigo. A mi novia Brenda que es la principal responsable de que haya podido realizar esta tesis, siendo mi inspiración y mi luz gracias por estar a mi lado en cada momento de trabajo y traer tanta felicidad a mi vida.*

*A mis hermanas que siempre llevo conmigo preocupándome por sus cosas y tratando de ser un ejemplo para ellas. A mi papá que me quiere mucho aunque a veces hay palabras que son difíciles de mencionar. Gracias a mis abuelas Nardis y Pilar por ser tan quisquillosas y guiarme por el buen camino. A mis amigos que ocupan un plano muy alto en mi vida gracias por estar conmigo en todo momento, en especial a Rafael que aunque nos vemos poco por estudiar en lugares diferente la amistad ha sido la misma o mayor. A mis amigos de Tacajó que es muy difícil de mencionarlos a todos aunque los tengo en mi corazón Carlos, Lojo, Yunior, Jose, Samir y Alejandro. A mis amigos que han estado a mi lado desde la vocacional compartiendo momentos buenos y malos, aconsejándonos unos a los otros, Yordanis, David, Joel y Julito. A todos mis amigos que se fundaron a lo largo de la carrera en especial a Odeinis que llegando a la escuela fuimos los mejores amigos del mundo en cualquier condición, a mi gente del apartamento que cada día compartimos la vida juntos, a Lachi, Hilario, Dayan, Luison, Angelito, Jose, Santi, Carlos, Rjvero.*

*Agradezco a mis primos que jugábamos juntos desde chiquito y compartimos tan buenos momentos juntos. A lui gracias por la ayuda que me brindaste cuando estabas aquí y todo lo que estás haciendo con tu niño, a Pepe, Alexander, Pombo, que tantos juegos de pelota hemos jugado y tantos consejos me han dado, a Leydis, Renier, Martica y Milena por todo su cariño.*

*A todos los compañeros del proyecto que día a día trabajábamos por cumplir con nuestra tarea dándonos aliento y ayuda unos a los otros, a Karelia, Miguel, Yoennys, Yannier, Yenier, a los profesores Jorjito, Miguel y Yunieski, un agradecimiento muy especial para Jose por toda su ayuda y apoyo cuando más lo necesitaba.*

*Muchas gracias a mis tutores Yusliel y Damarys por todo su apoyo y ayuda, gracias Damarys por soportar todos los dolores de cabeza que le di y toda su preocupación.*

*Dedicatoria.*

*A mi mamá por darme todo de sí desde mi primer momento de vida.*

*A mi niñita Soilet por ser mi estrella y mi vida.*

*A mi abuelo Oscar por todo su amor y ser la persona determinante en mi carácter y educación.*

*A mi novia Brenda por ser mi magia, mi luz y mi inspiración.*

## **Resumen**

Actualmente la gestión de *software* ocupa un espacio cada vez mayor en el mundo de la informática. Una persona haciendo uso de programas informáticos puede reducir en gran medida el tiempo de realización de un trabajo y mejorar la calidad del mismo. Los procesos académicos y productivos de la Universidad de las Ciencias Informáticas demandan la continua interacción del personal universitario con programas imprescindibles destinados a este fin. Con el presente trabajo, se creó una aplicación con tecnología Drupal que se encuentra integrada a la Red Social de la Universidad de las Ciencias Informáticas, la cual permite gestionar productos de *software* previamente catalogados brindando una mejor accesibilidad e interacción con los mismos. En este trabajo se analizó el estado del arte de los sistemas de gestión de *software*, desde su creación hasta la actualidad. Se estudiaron y se especificaron las herramientas y tecnologías que se utilizaron, eligiendo al CMS Drupal como base para el desarrollo de la aplicación, además, se siguieron los pasos que brinda el proceso de desarrollo ágil orientado al nivel 2 de CMMI, la cual guió el análisis, diseño, implementación y las pruebas al sistema.

**Palabras clave:** catálogo, catálogo digital, *software*.

## ÍNDICE DE CONTENIDO

<b>INTRODUCCIÓN</b> .....	<b>1</b>
<b>Capítulo I: Sistema de gestión de productos de software de la Red Social de la Universidad de las Ciencias Informáticas</b> .....	<b>5</b>
1.1 Marco conceptual.....	5
1.2 Antecedentes de los catálogos de software .....	6
1.3 Ventajas de los sistemas de catalogación de software .....	6
1.4 Análisis de soluciones similares .....	7
1.4.1 Soluciones internacionales.....	7
1.4.2 Soluciones nacionales.....	8
1.4.3 Sistemas en la UCI.....	8
1.5 Resultado del análisis de los sistemas existentes .....	9
1.6 Proceso de desarrollo, lenguajes, herramientas y tecnologías a utilizar .....	9
1.6.1 Proceso de desarrollo con enfoque ágil orientado al segundo nivel de CMMI .....	9
1.6.2 Herramienta de modelado de software .....	11
1.6.3 Lenguajes del lado del cliente .....	11
1.6.4 Lenguajes del lado del servidor .....	12
1.6.5 Sistemas de Gestión de Contenidos (CMS) .....	13
1.6.5.1 Fundamentación del CMS Drupal.....	16
1.6.6 Entorno de Desarrollo Integrado.....	16
1.6.6.1 Fundamentación del Entorno de Desarrollo Integrado .....	17
1.6.7 Sistema de Gestión de Bases de Datos .....	17
1.6.7.1 Fundamentación del servidor de base de datos.....	18
1.6.8 Administración de base de datos.....	18
1.6.9 Servidor web .....	19
1.6.10 Herramienta de prueba.....	19
<b>Capítulo II: Caracterización del Sistema de gestión de productos de software de la Red Social de la Universidad de las Ciencias Informáticas</b> .....	<b>20</b>
2.1 Problemática .....	20
2.2 Propuesta del sistema.....	20
2.2.1 Categorización .....	21
2.2.2 Integración con la Red Social Universitaria .....	22



2.3 Modelo de dominio .....	23
2.4 Especificación de los requisitos de software .....	24
2.4.1 Requisitos funcionales.....	24
2.4.2 Requisitos no funcionales.....	32
2.5 Arquitectura y patrones .....	34
2.5.1 La pila tecnológica de Drupal .....	35
2.5.2 Principales módulos utilizados en el sistema .....	36
2.5.3 Patrones de diseño .....	36
<b>Capítulo III: Implementación y validación del Sistema de gestión de productos de software de la Red Social de la Universidad de las Ciencias Informáticas .....</b>	<b>39</b>
3.1 Estándares de codificación definidos por el CENIA .....	39
3.2 Diagrama de despliegue .....	39
3.3 Pruebas de caja negra .....	41
3.3.1 Casos de prueba .....	42
3.4 Resultados de las pruebas .....	51
3.5 Prueba de carga y estrés .....	51
3.5.1 Análisis e interpretación de los resultados.....	53
<b>Conclusiones generales .....</b>	<b>55</b>
<b>Recomendaciones .....</b>	<b>56</b>
<b>Bibliografía referenciada .....</b>	<b>57</b>
<b>Bibliografía consultada .....</b>	<b>60</b>

## ÍNDICE DE FIGURAS

Figura 1: Modelo de dominio.....	23
Figura 2: Pila tecnológica de Drupal.....	35
Figura 3: Modelo de despliegue.....	40
Figura 4: Cantidad de no conformidades por iteraciones. ....	51

## ÍNDICE DE TABLAS

Tabla 1: Requisitos funcionales. ....	25
Tabla 2: Descripción del requisito funcional Adicionar <i>software</i> . ....	28
Tabla 3: Descripción del requisito funcional Búsqueda básica. ....	30
Tabla 4: Descripción del requisito funcional Descargar <i>software</i> .....	31
Tabla 5: Descripción del requisito funcional Aprobar propuesta. ....	32
Tabla 6: Descripción Requisitos no funcionales. ....	34
Tabla 7: Caso de prueba Adicionar <i>software</i> .....	44
Tabla 8: Descripción de los campos Caso de prueba Adicionar <i>software</i> .....	44
Tabla 9: Descripción caso de prueba Editar <i>software</i> . ....	46
Tabla 10: Descripción campos caso de prueba Editar <i>software</i> . ....	46
Tabla 11: Descripción caso de prueba Búsqueda básica. ....	47
Tabla 12: Descripción campos caso de prueba Búsqueda básica.....	47
Tabla 13: Descripción caso de prueba Descargar <i>software</i> . ....	48
Tabla 14: Descripción caso de prueba Proponer <i>software</i> . ....	49
Tabla 15: Descripción campos caso de prueba Proponer <i>software</i> .....	50
Tabla 16: Características del hardware de prueba. ....	52
Tabla 17: Características del <i>software</i> de prueba. ....	52
Tabla 18: Carga de trabajo y resultados esperados de los escenarios.....	53
Tabla 19: Resultados EC1. Acceder al sistema.....	53
Tabla 20: Resultados EC2: Acceder a un <i>software</i> . ....	54
Tabla 21: Resultados EC3: Descargar un <i>software</i> .....	54

## **INTRODUCCIÓN**

Unido al auge y desarrollo alcanzado por la computación surgen en el mundo un conjunto de necesidades, entre las que se encuentra la comunicación entre las personas para compartir recursos e información. En la década de 1980 del siglo pasado nace Internet, dándole solución al protocolo de comunicación máquina a máquina. De esta forma se oficializa la creación de un medio de propagación mundial que fue evolucionando, desde la comunicación entre paquetes y diálogo entre sistemas de cómputo, hasta los protocolos web que existen en la actualidad.

Uno de los logros más grandes del mundo moderno es intentar llevar la informatización a todas las ramas de la vida cotidiana. Una persona haciendo uso de programas informáticos puede reducir en gran medida el tiempo de realización de un trabajo y mejorar la calidad del mismo. El auge de las Tecnologías de la Información y las Comunicaciones (TIC), ha hecho posible varios de los avances tecnológicos que se producen a diario. El buen uso de las mismas ha contribuido a elevar el nivel cognoscitivo de la sociedad humana, donde a través del uso de la computadora y el Internet se ha evolucionado social y culturalmente. En la actualidad la mayor parte de las empresas en el mundo hacen uso de las TIC para lograr un buen desempeño laboral de sus trabajadores.

Debido al auge creciente del uso de productos de *software* en la industria de la informática, urge la necesidad de tener acceso rápido y permanente a los productos y soluciones informáticas que vienen a suplir las necesidades de la sociedad actual. Un medio a través del cual se le ha dado respuesta a la problemática de cómo y dónde obtener los productos de *software* es la Web, en la que se usan recursos como los catálogos donde se almacenan y se brinda a los usuarios de la red las aplicaciones requeridas para mantener su entorno de trabajo funcional. Entre los hechos más trascendentales de la Web actual se encuentra su diseño centrado en el usuario, la interoperabilidad, entre otras propiedades que están asociadas a un fenómeno social, las cuales, en el entorno de Internet, se han definido como Web 2.0.

La Web 2.0 es la transición que se ha dado de aplicaciones tradicionales hacia aplicaciones que funcionan a través de la Web enfocadas al usuario final y propiciando la interacción directa con estos, la colaboración y servicios que reemplacen las aplicaciones de escritorio. Este nuevo paradigma de la Web, permite la creación de comunidades virtuales, servicios, blogs, redes sociales, alojamiento de videos, etc.

La Universidad de las Ciencias Informáticas (UCI) no ha estado ajena a estas novedades, la misma cuenta con un Centro de Informatización Universitaria (CENIA) que entre los principales objetivos se encuentra el desarrollo de una red social que integrará varios servicios dentro de un escritorio, que podrá ser personalizado por el usuario.

Actualmente los usuarios de la UCI tienen disponible un repositorio de *software* llamado UCIstore. Este no es más que un servidor FTP (Protocolo de Transferencia de Archivos) con una organización de carpetas según cierta clasificación de *software*.

UCIstore brinda a los usuarios de la UCI actualizaciones y aplicaciones de diferentes *software* para el desarrollo profesional o académico de la universidad. Posee una amplia gama de cursos y tutoriales tanto de las asignaturas impartidas en la universidad como otras de perfil profesional, conjuntamente de las herramientas necesarias para el desarrollo de la producción.

Independientemente de las facilidades que brinda, resulta muy difícil la búsqueda y localización de algún programa de *software* debido a la carencia de información que brindan las aplicaciones expuestas. En muchos casos existen productos de *software* en manos de los usuarios de la universidad que resulta imposible compartirlo con toda la comunidad universitaria. Al ser un servidor FTP no existen mecanismos para promover los nuevos productos que a él se integran conjuntamente de no conocer las opiniones y sugerencias de los usuarios, necesario para el continuo desarrollo de la Universidad.

A partir de la problemática expuesta con anterioridad se plantea el siguiente **problema a resolver**: ¿Cómo potenciar la organización y el conocimiento de los productos de *software* que se desarrollan en la Universidad de las Ciencias Informáticas a través de la Red Social Universitaria?

Se plantea como **objeto de estudio** la catalogación de *software* y el **campo de acción** los sistemas informáticos de catálogo de *software*.

Dadas las condiciones mencionadas con anterioridad se propone como **objetivo general**: Desarrollar un sistema que permita la organización y promoción de los productos informáticos en la Red Social de la Universidad de las Ciencias Informáticas mediante la utilización de un sistema que permita catalogar *software*, potenciando el intercambio entre los usuarios de la misma.

Del objetivo general, se derivan los siguientes **objetivos específicos**:

- Caracterizar aspectos teóricos conceptuales sobre sistemas existentes para la gestión de *software*.
- Implementar el sistema que permite la gestión de *software* en la Universidad de las Ciencias Informáticas.

- Validar la solución desarrollada mediante la aplicación de pruebas funcionales.

Como **idea a defender** se propone: Con el desarrollo del sistema de gestión de productos de *software* de la Red Social de la Universidad de las Ciencias Informáticas se obtendrá un sistema capaz de catalogar el *software* en un repositorio organizado y estructurado, erradicando de esta forma los problemas existentes en cuanto a la búsqueda y localización de los programas.

Con el fin de dar cumplimiento al objetivo general se han propuesto las siguientes **tareas a resolver**:

- Estudio de diferentes sistemas informáticos existentes a nivel nacional e internacional dedicados a la gestión y catalogación del *software*.
- Investigación sobre la metodología y las herramientas a utilizar para el modelado del sistema informático, así como el marco de trabajo propuesto por el grupo de arquitectura del proyecto.
- Definición y descripción de las funcionalidades del sistema.
- Implementación del sistema.
- Validación del sistema mediante la aplicación de las pruebas.

Los métodos científicos teóricos que se emplearon para darle solución a las tareas propuestas fueron:

- Histórico-lógico: se estudió como ha sido la trayectoria y la evolución de sistemas de gestión de programas de *software* para tener una visión de las funcionalidades que la propuesta debe cumplir.
- Analítico-sintético: se utilizó en el estudio de la literatura especializada relacionada con el tema y la exploración de resultados de investigaciones afines, lo que permitió adoptar posiciones teóricas relacionadas con el objeto de investigación.

Los métodos científicos empíricos que se emplearon para darle solución a las tareas propuestas fueron:

- Observación: se usa porque permite investigar los procesos externamente sin tener que llegar a la esencia de los mismos, lo que ayudó al planteamiento del problema científico, además de permitir conocer bien el proceso delimitado como objeto de estudio, lo cual ayuda a tener un conocimiento más detallado de lo que se quiere, lo que hace falta hacer y cómo hay que hacerlo.
- Entrevista: mediante entrevistas informales realizadas a los especialistas del centro se pudo determinar cuáles eran las expectativas que se deben cumplir, herramientas a utilizar, además de esclarecer dudas sobre funcionalidades que debería tener la propuesta a implementar.
- Encuesta: con la realización de encuestas a usuarios de la universidad se logró llegar a una correcta categorización de acuerdo con sus necesidades y se definieron los prototipos de interfaz de la propuesta a implementar.

Al concluir la investigación se **espera obtener** el Sistema de gestión de productos de *software* de la Red Social de la Universidad de las Ciencias Informáticas, así como la obtención de todos los artefactos que se generan como parte del desarrollo de dicho sistema.

A continuación se describen los capítulos por los que está formada la presente investigación.

**Capítulo I. Sistema de gestión de productos de software de la Red Social de la Universidad de las Ciencias Informáticas.** En este capítulo se plantean las bases teóricas para dar solución al problema planteado. Además se realizó un estudio de los sistemas de gestión de información.

**Capítulo II. Caracterización del Sistema de gestión de productos de software de la Red Social de la Universidad de las Ciencias Informáticas.** Presenta una descripción general del funcionamiento y las herramientas del sistema, además de los requerimientos de la propuesta solución según las necesidades de los usuarios.

**Capítulo III. Implementación y pruebas del Sistema de gestión de productos de software de la Red Social de la Universidad de las Ciencias informáticas.** Presenta la documentación de las diferentes etapas del desarrollo y los artefactos relacionados con la implementación, además de la descripción de las pruebas realizadas al sistema para su posterior validación

**Capítulo I: Sistema de gestión de productos de software de la Red Social de la Universidad de las Ciencias Informáticas**



**Introducción**

En el presente capítulo se recogen los principales conceptos relacionados con los catálogos de *software*. Se realiza un análisis detallado de diferentes herramientas que permiten gestionar y catalogar programas de *software* a nivel internacional y nacional. Se describe el Proceso de desarrollo con enfoque ágil orientado al segundo nivel de CMMI, las herramientas y las tecnologías utilizadas en el diseño y desarrollo de la aplicación.

**1.1 Marco conceptual**

Con el objetivo de lograr una mayor comprensión de la solución propuesta a continuación se exponen los principales conceptos.

- **Catálogo:** un catálogo no es más que una lista ordenada o clasificada que se hará sobre objetos. Se conoce como aquella publicación empresarial cuyo principio es la promoción de aquellos productos o servicios que una empresa ofrece y sobre la cual en definitiva versará el catálogo. El catálogo constituye la forma ordenada que tiene una empresa de presentar al mundo los productos que fabrica o comercializa. Está compuesto principalmente por imágenes de los productos o servicios que se ofrecen en la empresa y que pueden ir acompañadas de breves descripciones, como por ejemplo el precio o algunos de los principales beneficios del producto. (LAPUENTE, 2011a)
- **Catálogo digital:** en sistema de catalogación electrónica realiza la gestión de cualquier tipo de información que pueda ser organizada de manera jerárquica. Cuentan en su gran mayoría con un repositorio centralizado con la información de todos sus productos, además de funcionalidades adicionales que ayudan a la personalización de la aplicación. Estos pueden realizar búsquedas por determinadas categorías, lo que los hacen muy precisos. El seguimiento de las acciones del usuario, ayuda a mantener un estudio estadístico de la aceptación de determinados productos, la suscripción del usuario a los productos para la futura recepción de noticias acerca de este, entre otras. (MEJÍA, 2010)
- **Software:** el *software* es un conjunto de programas elaborados por el hombre, que controlan la actuación de la computadora, haciendo que éste siga en sus acciones una serie de esquemas lógicos predeterminados. Tal característica lógica o inteligente del *software* es lo que hace que

se le defina también como la parte inmaterial de la informática, ya que aunque los programas que constituyen el *software* residan en un soporte físico, como la memoria principal o los disquetes (o cualquier dispositivo rígido de almacenamiento), la función de los programas en un computador es semejante a la del pensamiento en un ser humano.

El *software* es el nexo de unión entre el hardware y el hombre. La computadora, por sí sola, no puede comunicarse con el hombre y viceversa, ya que lo separa la barrera del lenguaje. El *software* trata de acortar esa barrera, estableciendo procedimientos de comunicación entre el hombre y la máquina; es decir, el *software* obra como un intermediario entre el hardware y el hombre. (VERGARA, 2007)

### **1.2 Antecedentes de los catálogos de software**

El término catálogo puede parecer fácilmente definible, encierra en su componente más importante una de las más antiguas y discutidas definiciones de la historia de las palabras. Etimológicamente “catálogo” proviene, a través del vocablo latino *cátalo gus*, de la frase griega *kata logos*. *Kata* significa por o de acuerdo con. La palabra “*logos*” en ocasiones es interpretada como “palabra” o “verbo”, otras veces como orden y en otras ocasiones como razón.

Distintos autores han analizado la historia de la catalogación y aunque el término historia sugiere fundamentalmente un desarrollo gradual y progresivo, algunos expertos en el tema nos demuestran que no ocurre así con la catalogación, ya que los antiguos bibliotecarios eran tan expertos en el arte de construir catálogos como puede serlo cualquier catalogador hoy en día, lo que difieren son sus métodos.

Hoy se puede afirmar que el catálogo es un instrumento de comunicación e información. Si el catálogo es un instrumento de comunicación, entonces es un instrumento determinado históricamente: ligado sobre todo a las necesidades de quien lo utiliza, por una parte y al modo en que se organiza y se manifiesta el proceso informativo, por otra. (ALFONSO FUENTES, 2010)

### **1.3 Ventajas de los sistemas de catalogación de software**

Entre los beneficios que ofrece los sistemas de catálogos de *software* se pueden mencionar:

- Ofrecen la información actualizada en tiempo real permitiendo de esta forma la satisfacción del usuario con mayor rapidez.
- El mantenimiento al cual se someten estas aplicaciones es rápido y sencillo.
- Ofrecen simplicidad en su uso para los usuarios.
- Pueden ofrecer funcionalidades personalizadas y dinámicas, de acuerdo a las preferencias e intereses del usuario.



- Fácil actualización de los productos de interés sin costos adicionales.
- Información detallada de los productos ofrecidos.

#### **1.4 Análisis de soluciones similares**

##### **1.4.1 Soluciones internacionales**

###### ➤ **Open PYME**

Open PYME constituye un catálogo de *software* libre donde se recopilan de forma categorizada productos que pueden incorporarse en cualquier ámbito productivo de una empresa, mejorando así su gestión y competitividad gracias a la inclusión de herramientas TIC. Dispone de una amplia gama de aplicaciones. Las mismas se encuentran categorizadas y ordenadas de manera que encontrarlas resulta bastante fácil. Junto a cada aplicación aparece una pequeña reseña de su funcionalidad. A la vez indica los requisitos técnicos para su instalación.(MARTÍN, RAYCO ABAD, 2012)

###### ➤ **Softonic**

Softonic ofrece más de 100.000 programas catalogados. De cada programa de *software* presenta una pequeña explicación en español de sus principales funcionalidades, lo que posibilita que los usuarios no avanzados comprendan en qué consiste esencialmente el *software*. Apuntando a este tipo de usuarios, Softonic ofrece la descarga desde la página del autor o desde su propio servidor. Además, también cuentan con una sección orientada a *software* para profesionales. Es válida para buscar programas de Windows, Mac, Linux, etc. e incluye la posibilidad de que los usuarios puedan escribir sus opiniones, luego de que estos descarguen algún programa, facilitando así la labor de encontrar los programas más adecuados a las necesidades de las personas. No es necesario registrarse para poder descargar los contenidos de esta Web.(ESTEBAN, 2011)

###### ➤ **Cdlibre**

Cdlibre se llevó a cabo con la finalidad de recopilar *software* libre que pudiera ser utilizado sobre el sistema operativo Windows. En la misma se pueden encontrar una gran cantidad de *software* libre. Cuenta con un catálogo de gran diversidad como *software* de audio, base de datos, navegadores, clientes de FTP, *software* educativo, juegos, programación y otros. Dicho sistema posibilita el envío de correos a los usuarios acerca de la información manejada por el sistema. Para cada aplicación de *software* presentada por dicho sistema, se muestra una breve descripción de la misma, idioma, plataforma donde se ejecuta, licencia, página de descarga, entre otros datos. Además de esto, en la misma Web se pueden encontrar varios DVDs/CDs con recopilaciones de las aplicaciones publicadas.(MARCO, 2010 )

➤ **Uptodown**

Uptodown presenta un diseño sencillo, claro y adaptado a las necesidades de los usuarios. Incluye además descargas convencionales y un foro donde los usuarios pueden exponer sus dudas sobre programas. Una de sus características es la categoría 'Educativa' que recoge programas vinculados con materias educativas, como las matemáticas, la mecanografía, idiomas o diccionarios. Además, cuenta con un 'top 10', una lista con los diez programas más descargados durante la semana en las distintas categorías.(RODRÍGUEZ, 2011)

**1.4.2 Soluciones nacionales**

➤ **Grupo de usuarios de tecnologías libres**

El Grupo de Usuarios de Tecnologías Libres, organizado en Cuba a partir de la evolución de las comunidades de usuarios de GNU/Linux de Cuba, presenta como objetivo principal la migración hacia tecnologías y conocimientos libres.(LIBRES, 2010) El mismo cuenta con una zona de descarga donde se encuentra gran variedad de *software* principalmente para Linux. Estos productos están organizados por categoría y no cuentan con una descripción, sino que solo se muestra la fecha de actualización. La página web contiene un buscador que permite a los usuarios encontrar un *software* determinado en poco tiempo.

**1.4.3 Sistemas en la UCI**

➤ **Biblioteca UCI**

La UCI cuenta con una biblioteca virtual, la cual brinda una extensa documentación de artículos, revistas, tesis y libros. En esta biblioteca se puede realizar búsqueda básica y avanzada mediante la entrada de parámetros o criterios por los que algún material puede ser encontrado. Brinda además una ayuda y un manual muy útil para los usuarios inexpertos. Aunque el sistema resulta ser realmente de utilidad, su funcionalidad de catálogo es para la organización y búsqueda de documentación.

➤ **UCIstore**

UCIstore es el repositorio más conocido y utilizado en la UCI por la gran variedad de productos de *software* que brinda. A diario es visitado por la mayoría de los usuarios de la comunidad universitaria, pero como se ha explicado anteriormente la búsqueda se dificulta por la ausencia de un sistema que ayude a la catalogación y organización de los programas que en él se encuentran. Además, este consiste en un servidor FTP, incumpliendo con el objetivo de la creación de un sistema basado en tecnología web.

## **1.5 Resultado del análisis de los sistemas existentes**

Luego de analizar los principales catálogos de *software* existentes en el mundo se concluye que estos sistemas son privativos, no permiten la posibilidad de ser descargados y usados además de que ninguno cumple con las características necesarias para darle solución a la problemática planteada, ya que estos han sido implementados para satisfacer las necesidades específicas de una institución. En el caso de Cdlibre está enfocado al *software* libre para Windows, además que el intercambio de información entre los usuarios es limitado o bajo condiciones como es el caso de Softonic que para poder comentar el *software* se tiene que haber descargado alguno con anterioridad. Otra de las razones por la que no se utilizan los sistemas analizados es porque el objetivo que se persigue es la integración con la Red Social de la universidad donde el flujo de intercambio de los usuarios con el sistema será controlado por dicha Red Social. Basándose en estas limitaciones se propone el desarrollo de un sistema de gestión de *software* que cumpla con todas las necesidades existentes en la universidad y a su vez se integre a la Red Social Universitaria.

## **1.6 Proceso de desarrollo, lenguajes, herramientas y tecnologías a utilizar**

### **1.6.1 Proceso de desarrollo con enfoque ágil orientado al segundo nivel de CMMI**

Un proceso de desarrollo de *software* tiene como propósito la producción eficaz y eficiente de un *software* que reúna los requisitos del cliente. Un proceso es ágil cuando el desarrollo de *software* es incremental (entregas pequeñas de *software*, con ciclos rápidos), cooperativo (clientes y desarrolladores trabajan juntos constantemente con una cercana comunicación), sencillo (el método en sí mismo es fácil de aprender y modificar, bien documentado), y adaptable (permite realizar cambios de último momento). (BAHIT, 2011) El proceso de desarrollo con enfoque ágil orientado al segundo nivel de CMMI es el definido por el centro CENIA.

### **Enfoque ágil**

Este enfoque ágil de las metodologías de desarrollo de *software* surge como alternativa a las metodologías formales que se consideran excesivamente pesadas y rígidas por su carácter normativo y fuerte dependencia de planificaciones detalladas previas al desarrollo. Dicho enfoque supera los defectos de estas, ya que se basan:

- En la estrecha colaboración con los usuarios para ofrecer varias versiones del *software* en ráfagas cortas (sprints). En cada versión, los requisitos para la próxima versión son refinados a partir de los comentarios de los usuarios.
- Valorar más el *software* que funciona que la documentación excesiva. Genera la documentación necesaria del desarrollo del producto.

- Mejores soluciones, menos costo y tiempo. Asegura que cada versión incorpore sólo los requerimientos de alto valor para el negocio. El enfoque ágil permite que el *software* sea implementado rápidamente.
- Valorar más la respuesta al cambio que el seguimiento de un plan. por lo que es ideal para proyectos donde los requisitos son inestables.

### **Nivel 2 de CMMI**

CMMI es un modelo de calidad del *software* que provee lineamientos para las empresas, organizaciones o áreas que deseen una mejora continua y efectiva en sus procesos de desarrollo. Para esto propone 25 áreas de proceso las cuales están agrupadas en 5 niveles (iniciado, gestionado, definido, gestionado cuantitativamente, en optimización) de madurez para clasificar a las empresas en función de en qué áreas de proceso consiguen sus objetivos y se gestionan con principios de ingeniería.

El nivel 2 de CMMI es el gestionado, se enfoca en la gestión de procesos y define siete áreas:

- **Gestión de requisito:** mantiene los requerimientos y describe las actividades para obtener y controlar los cambios a los requerimientos y asegurar que otros planes y datos relevantes se mantengan actualizados.
- **Planificación del proyecto:** incluye el desarrollo del plan de proyecto, la involucración de las partes interesadas de forma apropiada, la obtención de compromisos con el plan y el mantenimiento del plan.
- **Seguimiento y control del proyecto:** incluye las actividades de monitorización y toma de acciones correctivas. Estas acciones pueden incluir la re-planificación del proyecto.
- **Gestión de acuerdos con proveedores:** trata la necesidad del proyecto de adquirir aquellas partes del trabajo que son producidas por proveedores. Las fuentes de productos que podrían ser usadas para satisfacer los requerimientos del proyecto se identifican de forma proactiva.
- **Medición y análisis:** da soporte a todas las áreas de proceso, proporcionando prácticas específicas que guían a los proyectos y a las organizaciones durante la alineación de las necesidades y objetivos de medición con una forma de medir que proporcionará resultados objetivos.
- **Aseguramiento de calidad de procesos y productos:** da soporte a todas las áreas de proceso, proporcionando prácticas específicas para evaluar objetivamente los procesos, los productos de trabajo y los servicios realizados frente a las descripciones aplicables de procesos, estándares y procedimientos.

- **Gestión de configuración:** da soporte a todas las áreas de proceso, estableciendo y manteniendo la integridad de los productos de trabajo usando la identificación de la configuración, el control de la configuración, los informes de estado de la configuración y las auditorías de la configuración. (ALELÍ SÁNCHEZ MÉNDEZ, 2012)

A partir de lo explicado anterior podemos decir que el proceso de desarrollo con enfoque ágil orientado al segundo nivel de CMMI reúne:

- Las mejores prácticas del modelo traducidas en metas que indican qué hacer y que guíen la gestión del proceso.
- Las buenas prácticas del proceso de desarrollo ágil, traducidas en tareas de ingeniería enfocadas en cómo hacer las cosas.

### **1.6.2 Herramienta de modelado de software**

#### ➤ **Visual Paradigm 8.0**

Visual Paradigm para UML es la herramienta definida por el centro; soporta el ciclo de vida completo del desarrollo de *software*: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El *software* de modelado UML ayuda a una más rápida construcción de aplicaciones de calidad y de menor costo. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. Visual Paradigm ofrece:

- Entorno de creación de diagramas para UML.
- Diseño centrado en casos de uso y enfocado al negocio que generan un *software* de mayor calidad.
- Uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación.
- Capacidades de ingeniería directa (versión profesional) e inversa.
- Modelo y código que permanece sincronizado en todo el ciclo de desarrollo.
- Disponibilidad de múltiples versiones, para cada necesidad.
- Disponibilidad de integrarse en los principales IDE.
- Disponibilidad en múltiples plataformas. (VISUAL-PARADIGM.COM, 2012)

### **1.6.3 Lenguajes del lado del cliente**

#### ➤ **HTML**

HTML (*HyperText Markup Language* o en español Lenguaje de Marcado de Hipertexto), es el lenguaje con el que se definen las páginas web. Se trata de un conjunto de etiquetas que sirven para definir la forma en la que presentar el texto y otros elementos de la página.

Este lenguaje se escribe en un documento de texto, por eso se necesita un editor de textos para escribir una página web. El archivo donde está contenido el código HTML es un archivo de texto, con una peculiaridad, que tiene extensión .html o .htm.(LAPUENTE, 2011b)

➤ **CSS**

El CSS (*Cascading Style Sheets* o en español Lenguaje de Hojas de Estilos), es un lenguaje de hojas de estilos creado para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML y XHTML (*Extensible HyperText Markup Language* o en español Lenguaje de Marcado de Hipertexto Extensible). CSS es la mejor forma de separar los contenidos de su presentación y es imprescindible para crear páginas web complejas.

Separar la definición de los contenidos de su aspecto presenta numerosas ventajas, ya que obliga a crear documentos HTML/XHTML bien definidos, además, mejora la accesibilidad del documento, reduce la complejidad de su mantenimiento y permite visualizar el mismo documento en infinidad de dispositivos diferentes. Al crear una página web, se utiliza en primer lugar el lenguaje HTML/XHTML para marcar los contenidos, es decir, para designar la función de cada elemento dentro de la página: párrafo, titular, texto destacado y tabla, lista de elementos. Una vez creados los contenidos, se utiliza el lenguaje CSS para definir el aspecto de cada elemento: color, tamaño y tipo de letra del texto, separación horizontal y vertical entre elementos, posición de cada elemento dentro de la página, etc.(HTML.NET, 2010)

➤ **JavaScript**

JavaScript es un lenguaje de programación creado principalmente para hacer páginas web dinámicas. Este se basa en las tecnologías del lado del cliente, ya que es el navegador el que soporta la carga de procesamiento. Dentro de sus características se encuentra brindar interactividad en las páginas con el cliente. Es un lenguaje con muchas posibilidades dentro de la programación web que permite la programación de pequeños *scripts*, pero también de programas grandes, orientados a objetos, con funciones y estructuras de datos complejas. JavaScript permite controlar todo lo que ocurre en la página cuando la está visualizando el cliente. Gracias a su compatibilidad con la mayoría de los navegadores modernos, es el lenguaje de programación del lado del cliente más utilizado.(ALVAREZ, MIGUEL ANGEL, 2011)

#### **1.6.4 Lenguajes del lado del servidor**

➤ **PHP 5**

PHP (*Hypertext Preprocessor* o en español Preprocesador de Hipertexto) es un lenguaje *script* de alto nivel interpretado del lado del servidor, por eso permite acceder a los recursos con los que cuenta el

servidor o a otros externos. No es necesario ser soportado por el navegador, no obstante, para que funcione debe ser soportado por el servidor. Debido a su código abierto, es utilizado por una gran cantidad de desarrolladores, garantizando así que los fallos sean resueltos con rapidez, por lo tanto, el código es continuamente mejorado. Puede interactuar con muchos motores de bases de datos como: MySQL, PostgreSQL, Oracle, entre otros. Está integrado a varias bibliotecas externas para el manejo de los datos, la generación de documentos en PDF (documentos de Acrobat Reader), calendarios, XML. Permite técnicas de programación orientada a objetos.(PHP.NET, 2012)

### **1.6.5 Sistemas de Gestión de Contenidos (CMS)**

Para el desarrollo del *software* se propone la utilización de un Sistema de Gestión de Contenido, debido a la gran cantidad de funcionalidades que facilita para la gestión en la Web. Una de sus principales características es que separan el contenido del diseño, esto permite que en cualquier momento se pueda cambiar el diseño de la página web sin modificar el contenido, además facilitan una serie de funcionalidades como el manejo de contenido, alta interactividad, eficiencia y seguridad en la administración de roles.

Un CMS presenta una interfaz que controla una o varias bases de datos donde se aloja el contenido del sitio. El sistema permite manejar de manera independiente el contenido por una parte y el diseño por otra, así, es posible manejar el contenido y darle en cualquier momento un diseño distinto al sitio sin tener que darle formato al contenido de nuevo, además de permitir la fácil y controlada publicación en el sitio a varios editores.

Un CMS consta de dos elementos: la aplicación para el manejo de contenido y la aplicación para la entrega de contenido. La aplicación de administración de contenido le permite a la persona encargada de administrar el contenido o al autor, quien puede no conocer el lenguaje HTML con el que se hacen las páginas, administrar, crear, modificar y retirar contenido de un sitio web sin necesitar la ayuda de un administrador. El elemento de despliegue de contenido utiliza y compila la información del CMS para actualizar el sitio web.(TELL, 2011)

#### **➤ Drupal**

Drupal es un programa de código abierto, con licencia GNU/GPL, escrito en PHP, desarrollado y mantenido por una activa comunidad de usuarios. Destaca 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. El diseño de Drupal es especialmente idóneo para construir y gestionar comunidades en Internet, no obstante, su flexibilidad y adaptabilidad, así como la gran

cantidad de módulos adicionales disponibles, hace que sea adecuado para realizar varios tipos de sitio web. A continuación se describen las principales características del CMS Drupal:

- **Ayuda en línea:** presenta un robusto sistema de ayuda en línea y páginas de ayuda para los módulos del núcleo, tanto para usuarios como para administradores.
- **Búsqueda:** todo el contenido en Drupal es totalmente indexado en tiempo real y se puede consultar en cualquier momento.
- **Código abierto:** el código fuente de Drupal está libremente disponible bajo los términos de la licencia GNU/GPL.
- **Módulos:** la comunidad de Drupal ha contribuido muchos módulos que proporcionan funcionalidades como página de categorías, autenticación mediante jabber, mensajes privados, etc.
- **Personalización:** un robusto entorno de personalización está implementado en el núcleo de Drupal. Tanto el contenido como la presentación pueden ser individualizados de acuerdo a las preferencias definidas por el usuario.
- **Multiplataforma:** Drupal ha sido diseñado desde el principio para ser multi-plataforma. Puede funcionar con Apache o Microsoft IIS como servidor web y en sistemas como Linux, BSD, Solaris, Windows y Mac OS X. Por otro lado, al estar implementado en PHP, es totalmente portable.
- **Independencia de la base de datos:** aunque la mayor parte de las instalaciones de Drupal utilizan MySQL, existen otras opciones. Drupal incorpora una capa de abstracción de base de datos que actualmente está implementada y mantenida para MySQL y PostgreSQL, aunque permite incorporar fácilmente soporte para otras bases de datos.
- **Objetos de Contenido (Nodos):** el contenido creado en Drupal es, funcionalmente, un objeto (nodo). Esto permite un tratamiento uniforme de la información como una misma cola de moderación para envíos de diferentes tipos, promocionar cualquiera de estos objetos a la página principal o permitir comentarios o no sobre cada objeto. (TOMLINSON, 2010)

➤ **Joomla**

Joomla es un sistema de gestión de contenidos de código abierto, programado haciendo uso del lenguaje PHP y SQL (*Structure Query Language* o en español Estructura de Lenguaje de Consulta) y consecuentemente utiliza la base de datos relacionales MySQL para almacenar el contenido y los parámetros de configuración del sitio. Además de libre, Joomla es gratuito. Entre sus principales virtudes está la de permitir editar el contenido de un sitio web de manera sencilla. Una de las mayores



potencialidades que presenta este CMS es la gran cantidad de extensiones existentes programadas por su comunidad de usuarios, aumentado las posibilidades de Joomla con nuevas características que se integran fácilmente en él. Existen cientos de extensiones disponibles y con diversas funcionalidades como por ejemplo:

- Generadores de formularios dinámicos
- Directorios de empresas u organizaciones
- Gestores de documentos
- Galerías de imágenes multimedia
- Motores de comercio y venta electrónica
- *Software* de foros y *chats*
- Calendarios
- *Software* para blogs
- Servicios de directorio
- Boletines de noticias
- Herramientas de registro de datos
- Sistemas de publicación de anuncios
- Servicios de suscripción.(BARRIOCANAL, 2009)

#### ➤ **WordPress**

Es un CMS, orientado fundamentalmente a la construcción de blogs. Es un sistema de publicación bien diseñado, programado en PHP y MySQL. Sus raíces y desarrollo comenzaron en 2001 y actualmente es un producto totalmente estable. Se distribuye conforme a la licencia estándar GPL y para que funcione es necesario poseer un servidor que tenga soporte para PHP y MySQL. Presenta una fácil administración de usuarios con sus propios perfiles, una fácil instalación y actualización a nuevas versiones. Todas las páginas son generadas de forma dinámica, es decir, que la actualización del contenido o diseño se realiza de la forma más rápida posible. Dispone de una gran cantidad de idiomas. Posee una gran comunidad activa. Sus principales funcionalidades se describen a continuación:

- Fácil instalación, actualización y personalización.
- Múltiples autores o usuarios, junto con sus roles o perfiles que establecen distintos niveles de permisos.
- Múltiples blogs o bitácoras.

- Permite ordenar artículos y páginas estáticas en categorías, sub-categorías y etiquetas.
- Cuatro estados para una entrada (Publicado, Borrador, Esperando Revisión y Privado).
- Permite comentarios y herramientas de comunicación entre blogs.
- Subida y gestión de adjuntos y archivos multimedia.
- Búsqueda integrada.(WORDPRESS, 2012)

#### **1.6.5.1 Fundamentación del CMS Drupal**

Luego de realizar un análisis de diferentes CMS, se concluye que Drupal en su versión 7.14 será el utilizado en la construcción del sistema. Drupal permite realizar sistemas complejos, tiene en cuenta la gestión de usuarios y permisos ya que contiene una serie de grupos con permisos de acceso que pueden ser personalizados o adaptados a las necesidades concretas, superando al resto de los sistemas de gestión de contenido. El sistema tendrá que soportar varias descargas de *software* al mismo tiempo por lo que es fundamental su rendimiento y el tiempo de respuesta de las páginas, la replicación de base de datos y el balance de carga son factores de gran importancia que hacen que Drupal sea el idóneo para este caso.

#### **1.6.6 Entorno de Desarrollo Integrado**

Un Entorno Integrado de Desarrollo (IDE traducido del inglés *Integrated Development Environment*) es un programa informático compuesto por un conjunto de herramientas de programación. Es la unión de un editor de código, un compilador, un depurador y un constructor de interfaz gráfica, suministrando un marco de trabajo para la mayoría de los lenguajes de programación y en ocasiones puede funcionar también como un sistema en tiempo de ejecución, utilizando el lenguaje de programación de forma interactiva.(LUCIANO, 2008)

##### **➤ NetBeans**

Es un IDE de libre uso y de código abierto que permite la extensión de aplicaciones hechas en este entorno dada su característica de que los módulos pueden ser desarrollados independientemente. NetBeans provee una estructura para los proyectos que se pueden crear junto a este IDE, propone un esqueleto para organizar el código fuente, el editor conjuntamente integra los lenguajes como HTML, JavaScript y CSS. Además NetBeans posee un sistema para examinar todo los directorios de cada proyecto, haciendo reconocimiento y carga de clases, métodos y objetos, para acelerar la programación.(NETBEANS.ORG, 2012)

##### **➤ Eclipse**

Es un IDE de código abierto y multiplataforma para desarrollar aplicaciones robustas. Eclipse emplea

módulos para proporcionar toda su funcionalidad al frente de la plataforma, a diferencia de otros entornos monolíticos donde las funcionalidades están todas incluidas, las necesite el usuario o no. Le proporciona al desarrollador marcos de trabajo para desarrollar aplicaciones gráficas, definición y manipulación de modelos de *software* y aplicaciones web.(GUIA-UBUNTU.ORG, 2011)

➤ **Zend Studio**

Entorno de Desarrollo Integrado que sirve de editor de texto para páginas PHP que proporciona un buen número de ayudas desde la creación y gestión de proyectos hasta la depuración del código. Divide sus funcionalidades de la parte del cliente y las de la parte del servidor, instalándose cada una de ellas por separado.(GREHAN, 2010)

**1.6.6.1 Fundamentación del Entorno de Desarrollo Integrado**

Como IDE de desarrollo se decidió utilizar el NetBeans que es un producto de código abierto incluyendo estándares de la industria de *software* ampliamente aceptados, con soporte para autocompletado de código. La versión utilizada NetBeans 7.0.1 es para el desarrollo de aplicaciones web PHP que abarca una variedad de secuencias de comandos y lenguajes de marcado.

**1.6.7 Sistema de Gestión de Bases de Datos**

Los sistemas para gestionar datos constituyen un conjunto de programas no visibles al usuario final que se encargan de la privacidad, la integridad, la seguridad de los datos y la interacción con el sistema operativo. Proporciona una interfaz entre los datos, los programas que los manejan y los usuarios finales. Cualquier operación que el usuario hace contra la base de datos está controlada por el gestor. (ALVAREZ, SARA, 2007)

➤ **PostgreSQL**

Es un Sistema de Gestión de Bases de Datos objeto-relacional, distribuido bajo licencia BSD (*Berkeley Software Distribution* o en español Distribución de *Software* de Berkeley) y con su código fuente disponible libremente. Funciona con grandes cantidades de datos y una alta concurrencia de usuarios accediendo a la vez al sistema. Las principales características de PostgreSQL se describen a continuación:

- Soporta una gran variedad de tipos de datos.
- Permite la gestión de diferentes usuarios, como también los permisos asignados a cada uno de ellos.
- Posee una gran escalabilidad. Es capaz de ajustarse al número de CPU (*Central Processing Unit* o en español Unidad Central de Procesamiento) y a la cantidad de memoria que posee el

sistema de forma óptima, capaz de soportar una mayor cantidad de peticiones simultáneas de manera correcta.

- Tiene la capacidad de comprobar la integridad referencial, así como también la de almacenar procedimientos en la propia base de datos, equiparándolo con los gestores de bases de datos de alto nivel.(POSTGRESQL.ORG, 2009)

➤ **MySQL**

Es un sistema gestor de base de datos relacional, bajo la licencia GPL de la GNU. Es conocido y usado por su rendimiento, rapidez, facilidad para ser aprendido y sencillez al ser utilizado, siendo frecuente usado en aplicaciones web y plataformas. MySQL es un programa de intercambio que permite conectarse a servidores MySQL, para la realización de consultas y obtención de los resultados. Suele ser combinado para trabajar con PHP, siendo esta combinación bastante segura. Las principales características de MySQL se describen a continuación:

- Bajo costo en requerimientos para la elaboración de bases de datos, ya que debido a su bajo consumo puede ser ejecutado en una máquina con escasos recursos sin ningún problema.
- Facilidad de configuración e instalación.
- Gestión de usuarios y contraseñas, manteniendo un muy buen nivel de seguridad en los datos.(MYSQL.COM, 2012)

#### **1.6.7.1 Fundamentación del servidor de base de datos**

De acuerdo con los requerimientos del sistema el servidor de base de datos que se decidió utilizar es el PostgreSQL v8.4, ya que brinda gran estabilidad y seguridad de los datos. PostgreSQL es el idóneo para sistemas con alta concurrencia, donde se realice un gran número de consultas en cortos espacios de tiempo, que manejen gran cantidad de datos como es el caso del sistema en cuestión y su código fuente es disponible libremente.

#### **1.6.8 Administración de base de datos**

➤ **PgAdmin III**

Es una aplicación gráfica para gestionar el Gestor de Bases de Datos PostgreSQL, con licencia Open Source. Está escrita en C++ usando la librería gráfica multiplataforma wxWidgets, lo que permite que se pueda usar en Linux, FreeBSD, Mac OS X y Windows. Es capaz de gestionar versiones a partir de la PostgreSQL 7.3 ejecutándose en cualquier plataforma, así como versiones comerciales de PostgreSQL como Pervasive Postgres, EnterpriseDB, Mammoth Replicator y SRA PowerGres.

PgAdmin III está diseñado para responder a las necesidades los usuarios, desde escribir consultas SQL simples hasta desarrollar bases de datos complejas. La interfaz gráfica soporta todas las características de PostgreSQL y facilita la administración. La aplicación también incluye un editor SQL, un editor de código de la parte del servidor y un agente para lanzar *scripts* programados.(UBUNTU, 2010)

#### **1.6.9 Servidor web**

##### **➤ Apache 2.0**

Es un servidor HTTP de código abierto. Presenta entre otras características mensajes de error altamente configurables, bases de datos de autenticación y negociado de contenido. Su costo es gratuito, presenta gran fiabilidad y extensibilidad convirtiéndose así en una herramienta potente y muy fiable. Es un *software* que está estructurado en módulos. La configuración de cada módulo se hace mediante las directivas que están contenidas dentro del módulo y gracias a ser modular se han desarrollado diversas extensiones entre la que se destaca PHP.(FOUNDATION, 2011)

#### **1.6.10 Herramienta de prueba**

##### **➤ JMeter**

JMeter es una herramienta de escritorio que permite la realización de pruebas de *software*. Se puede utilizar para probar el rendimiento tanto en recursos estáticos como dinámicos (archivos, *scripts* de Perl, objetos Java, bases de datos, consultas y servidores FTP). Además permite probar la concurrencia de los usuarios y velocidad de carga en un sitio, mostrando los resultados de dichas pruebas mediante informes. Cuenta con un conjunto de funcionalidades que le aportan extensibilidad ya que posee un número ilimitado para la realización de pruebas.(JMETER, 2011)

#### **Conclusiones parciales**

Con el estudio de las diferentes funcionalidades de los sistemas que permiten la gestión de *software*, se definió el marco teórico conceptual de la investigación, que permitió identificar la problemática e identificar las bases para analizar, diseñar e implementar el Sistema de gestión de productos de *software* para la Red Social de la Universidad de las Ciencias Informáticas. Mediante el estudio y caracterización de las principales herramientas y tecnologías para la implementación del sistema, quedaron definidas las herramientas para la solución.

**Capítulo II: Caracterización del Sistema de gestión de productos de software de la Red Social de la Universidad de las Ciencias Informáticas**



**Introducción**

En el presente capítulo se describen las principales características del Sistema de gestión de productos de *software* de la Red Social de la Universidad de las Ciencias Informáticas, así como los requisitos funcionales y no funcionales. Se describe la arquitectura y los patrones utilizados.

**2.1 Problemática**

La Universidad de las Ciencias Informática tiene entre sus principales objetivos la producción de aplicaciones para la venta a otros países y de consumo nacional. Por sus características, la necesidad de tener a disposición de la comunidad universitaria herramientas digitales de uso diario va en ascenso. Los usuarios de la Universidad tienen disponible un repositorio de *software* llamado UCIstore. Este no es más que un servidor FTP con una organización de carpetas según cierta clasificación de *software*.

UCIstore brinda a los usuario de la UCI actualizaciones y aplicaciones de diferentes *software* para el desarrollo profesional y académico de la Universidad. Posee una amplia gama de cursos y tutoriales tanto de las asignaturas impartidas en la Universidad como otras de perfil profesional, conjuntamente de las herramientas de desarrollo necesarias para la producción. Independientemente de las facilidades que ofrece, resulta muy difícil la búsqueda y localización de la aplicación deseada debido a la carencia de información que brindan las aplicaciones expuestas. La categorización de las carpetas está basada en el sistema operativo Windows, solo cuenta con una carpeta Linux y Mac, con una pobre gama de productos. En muchos casos los usuarios de la Universidad poseen programas de *software* que en ocasiones no se encuentran dentro del servidor de UCIstore y se hace muy difícil que estos lleguen a manos de todo el que lo necesita. Al ser un FTP no existen mecanismos para promover los nuevos productos que a él se integran, además de no dar a conocer las opiniones y sugerencias de *software* de los usuarios para el continuo desarrollo de la Universidad.

**2.2 Propuesta del sistema**

La solución propuesta constituye un *software* elaborado con la tecnología Drupal. Dicho sistema incluye productos de *software*. Los usuarios pueden acceder a este sistema desde cualquier sitio, teniendo acceso rápido y visualización óptima del contenido que presenta. Los usuarios tienen la posibilidad de descargar archivos de forma rápida y segura. Cada producto de *software* cuenta con una breve descripción de sus principales funcionalidades, nombre, tamaño última actualización y una

foto, facilitando así la búsqueda. El sistema además cuenta con un bloque que muestra los *software* más descargados, evitando que tengan que ser buscados de forma muy seguida por los usuarios.

De cada usuario del sistema se conocen una serie de detalles como son IP de la máquina de donde está conectado, el nombre de los *software* a los que accedió y la fecha de acceso, de esta forma se conoce cuáles son las aplicaciones que más visita, para cuando se lleven a cabo cambios o se incluya algún *software* de ese tipo le llegue un mensaje de notificación a través de la RSU. Los usuarios tienen la opción de votar por el *software* de su preferencia. El sistema está integrado a la RSU, los usuarios pueden compartir cualquier *software* con sus amigos de la RSU, el sistema envía un correo electrónico con los datos del usuario que comparte el producto y el URL donde se encuentra.

La búsqueda podrá realizarse manualmente, a partir de la selección de categorías específicas o tecleando el nombre del *software* o caracteres aproximados a este, mostrando imágenes y descripciones del programa seleccionado. Los usuarios tienen la posibilidad también de proponer algún *software* para su inclusión dentro de la aplicación, caso que se llevará a análisis por parte de los administradores del sistema.

Para evitar la pérdida de información y logrando una mayor seguridad de los usuarios que acceden al sistema, los mismos tienen asignado un rol que determina cuáles son las limitaciones o permisos dentro de la aplicación. A continuación se describen estos roles y sus respectivos permisos.

- **Usuario invitado:** usuario que solo accede al sitio en busca de información referente a las noticias publicadas.
- **Usuario autenticado:** usuario que se registra en el sistema. Este tendrá la oportunidad de acceder a otras funcionalidades de la aplicación como la de descarga y subida de *software*. Tendrá libre acceso en el sistema excepto a la sección de administración.
- **Administrador:** gestiona toda la información que almacena el sistema además de ser el encargado de brindar soporte y mantenimiento.

### **2.2.1 Categorización**

Cada institución tiene formas diferentes de categorizar sus repositorios teniendo en cuenta sus prioridades. En el caso de UCIstore, es un repositorio que está orientado al sistema operativo Windows. Debido al auge e implantación del sistema operativo Linux en la Universidad se hace necesario una nueva catalogación de las aplicaciones expuestas. Teniendo en cuenta el estudio de aplicaciones similares a nivel nacional e internacional que más se asemejan al contexto universitario y encuestas realizadas a usuarios de la Universidad se definió la siguiente categorización.

- Internet: Gestores de descarga, Navegadores web, Correo electrónico, Mensajería instantánea.

- Seguridad: Contraseñas, Copias de seguridad, Recuperación de archivos, Antivirus, Cortafuegos, Anti espía, Cifrado.
- Administración: Impresión, Monitorización, Actualizadores de logs, Administración de usuarios.
- Video: Codec de video, Editores de video, Reproductores de video, Otros programas de video.
- Sistemas Operativos.
- Personaliza tu PC: Entornos de escritorio, Iconos, Temas y fondos, Utilidades de escritorio.
- Servidores: Servidores de archivos, Servidores de base de datos, Servidores de correo, Servidores de proxy, Servidores DNS, Servidores FTP, Servidores Web.
- Programación: Entornos de desarrollo, Programación Web, Utilidades de programación, Lenguajes de programación.
- Sonido: Conversores de sonido, Reproductores de sonido, Administradores de sonido, Otros programas de sonido.
- Imagen: Editores gráficos, Otras aplicaciones de imagen, Capturadores de pantalla, Retoque y optimización, Diseño 3D.
- Redes: Gestores de conexión, Análisis y administración, Control de acceso, Redes inalámbricas.
- Educación y ciencia: Física, Historia, Matemática.
- Utilidades: Texto, Gestores de paquetes, Análisis y optimización, Arranque y apagado, Archivos, Particiones, Emuladores, Compresores.

### **2.2.2 Integración con la Red Social Universitaria**

Para lograr la integración fue necesario confeccionar un módulo llamado *rsintegration* que posibilita a los usuarios del sistema compartir un *software* con sus amigos de la RSU utilizando los servicios que se describen a continuación:

- **Servicio de perfil:** plataforma que permite recuperar toda la información asociada a los usuarios, además posibilita la comunicación entre ellos y la interacción con el resto de los servicios brindados en la RSU. Se utilizó este servicio para obtener la lista de amigos del usuario que está haciendo uso del sistema.
- **Integración de los principales servicios telemáticos con la Red Social Universitaria:** la integración de estos servicios con la Red Social se realiza mediante una Interfaz de Programación de Aplicaciones (API) que permite mediante la Transferencia de Estado Representacional (REST), interactuar con las funcionalidades que brindan el servicio de correo electrónico, la mensajería instantánea y el servicios de cuotas de navegación. Esta API



posibilitó el envío de una notificación por correo electrónico a los amigos seleccionados para compartir un *software* haciendo uso del servicio de perfil.

### 2.3 Modelo de dominio

Durante el análisis de los procesos surgieron varios conceptos relacionados entre sí, estos definen el modelo del dominio, que permite a los usuarios, clientes, desarrolladores e interesados, utilizar un vocabulario común para poder entender el contexto en que se ubica el mismo, para capturar correctamente los requisitos y lograr una solución adecuada.

El objetivo del modelo del dominio es especificar y conocer las clases y objetos más significativos dentro del entorno del problema, lo que ayuda a determinar los procesos y roles más relevante para el sistema a desarrollar. (ORTIZ, 2010)

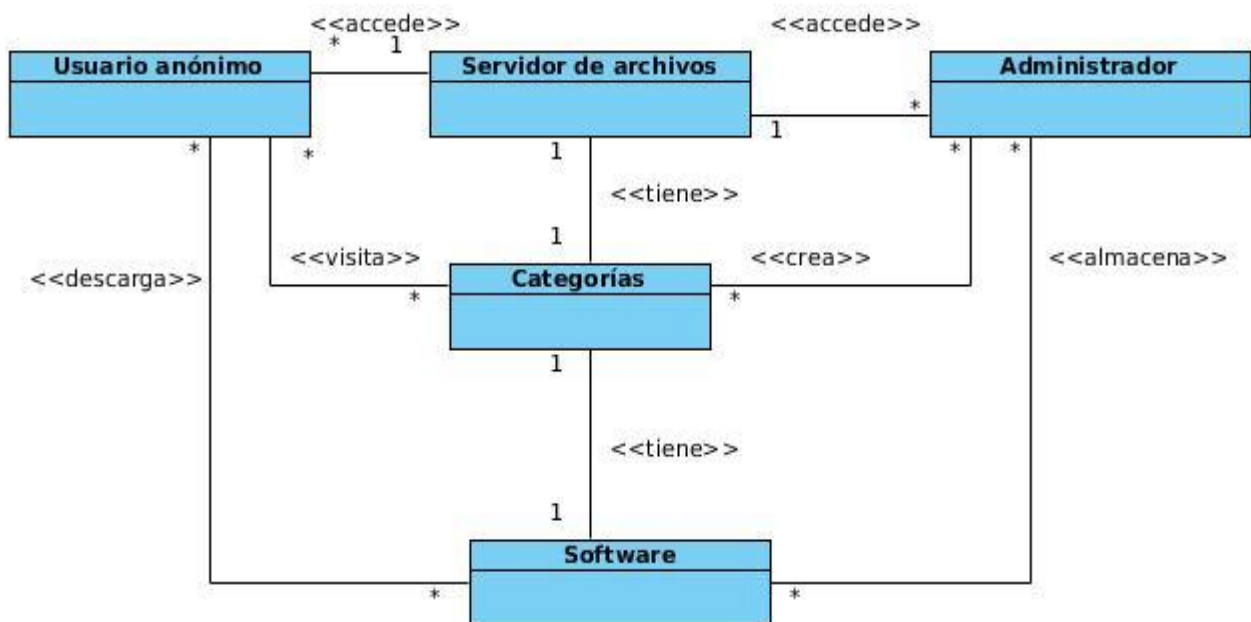


Figura 1: Modelo de dominio.

#### Conceptos principales del modelo de dominio

- **Servidor de archivos:** servidor FTP, el mismo brinda a los usuario de la UCI actualizaciones y aplicaciones de diferentes *software* para el desarrollo profesional o académico de la universidad.
- **Usuario anónimo:** usuario que accede a la aplicación de forma anónima en busca de productos de *software*.
- **Administrador:** usuario con el rol de administrador encargado de crear las categorías y almacenar el *software*.

- **Software:** es un conjunto de programas elaborados por el hombre, que controlan la actuación de la computadora, haciendo que éste siga en sus acciones una serie de esquemas lógicos predeterminados.
- **Categorías:** categoría a la cual pertenecen los productos de *software* dentro del repositorio.

## **2.4 Especificación de los requisitos de software**

Los requisitos son condiciones o necesidades de un usuario para resolver un determinado problema o alcanzar un objetivo. Son declaraciones de los servicios que provee el sistema, definiendo la manera en que este reacciona a entradas particulares. Estos pueden dividirse en requisitos funcionales y requisitos no funcionales. (ROMERO, 2011)

Para la propuesta de solución se definieron los requisitos funcionales y no funcionales que se listan a continuación:

### **2.4.1 Requisitos funcionales**

Una vez conocidos todos los conceptos que rodean al objeto de estudio, se puede analizar qué debe hacer el sistema para que se cumplan los objetivos planteados al inicio de este trabajo. Para ello se enumeran a través de requerimientos funcionales las prestaciones que el sistema será capaz de brindar. Dentro de ellas se incluyen las acciones que podrán ser ejecutadas por el usuario, las acciones ocultas que debe realizar el sistema y las condiciones extremas a determinar por el sistema. Teniendo en cuenta los objetivos planteados, el sistema debe ser capaz de cumplir con los siguientes requisitos.

<b>No</b>	<b>Requisito</b>
1	Autenticar usuario
2	Adicionar <i>software</i>
3	Editar <i>software</i>
4	Eliminar <i>software</i>
5	Búsqueda básica
6	Búsqueda avanzada
7	Descargar <i>software</i>
8	Proponer <i>software</i>
9	Editar propuesta.
10	Aprobar propuesta
11	Eliminar propuesta

12	Establecer permisos de usuario
13	Votar
14	Adicionar categoría
15	Editar categoría
16	Eliminar categoría
17	Mostrar categorías existentes
18	Mostrar los más descargados
19	Comentar
20	Responder comentario
21	Editar comentario
22	Aprobar comentario
23	Eliminar comentario

**Tabla 1:** Requisitos funcionales.

A continuación se describen los requisitos funcionales más importantes dentro de la aplicación.

No.	Nombre	Descripción	Complejidad	Prioridad para el cliente
RF_2	Adicionar <i>software</i>	Para acceder a esta funcionalidad el usuario va al menú de atajos que se encuentra en la parte superior de la ventana y elige la opción Agregar contenido (IU: Opción agregar contenidos). Posteriormente selecciona la opción <i>software</i> (IU: Opción <i>software</i> ) mostrando un formulario en pantalla (IU: formulario para crear un <i>software</i> ), el cual debe de ser llenado para poder subir el <i>software</i> . Al usuario escribir el título, seguidamente se sube la imagen que identifica al <i>software</i> a subir, se describen las principales funcionalidades del	Media	Alta

software, se añade el fichero el cual debe de estar compactado y se selecciona la categoría a la que pertenece el software a adicionar.

Finalmente se guarda el formulario, adicionando el software al sistema. Se muestra un mensaje de información diciendo que el software se ha creado.

### Prototipo de Interfaz



IU: Opción "Agregar contenido".



IU: Opción "Software".

The screenshot shows a web interface for creating software. At the top, there is a navigation bar with links like 'Contenido', 'Estructura', 'Apariencia', etc., and a user profile 'Bienvenido, laborrero'. Below the navigation, the page title is 'Crear Software'. The form consists of several sections:
 

- Título \***: A text input field.
- IMAGEN**: A section for adding a new image. It includes a file input field, an 'Examinar...' button, and a 'Subir al servidor' button. A note states: 'Los archivos deben ser menores que 128 MB. Tipos de archivo permitidos: png gif jpg jpeg.'
- Descripción**: A large text area for entering a description.
- FICHERO**: A section for adding a new file. It includes a file input field, an 'Examinar...' button, and a 'Subir al servidor' button. A note states: 'Los archivos deben ser menores que 128 MB. Tipos de archivo permitidos: zip rar tar.'
- Categoría \***: A dropdown menu with the option '- Seleccione un valor -'.
- Información de la revisión**: A section with a 'Sin revisión' status, a 'Crear nueva revisión' checkbox, and a 'Mensaje del historial de revisiones' text area. The message area has a note: 'Proporcionar una explicación de los cambios realizados para que otros autores comprendan las razones.'
- Opciones de comentarios**: A button labeled 'Abrir'.
- Información de autoría**: A section showing 'Por laborrero'.
- Opciones de publicación**: A section showing 'Publicado'.

 At the bottom of the form is a 'Guardar' button.

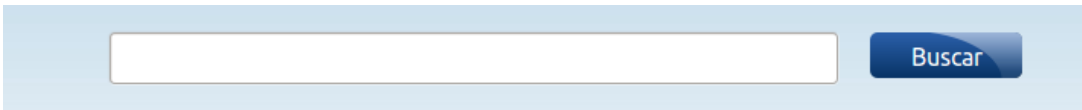

IU: Opción "Crear software".

Campos	Tipo de Datos	Reglas o Restricciones
Título	<i>varchar</i>	-Cualquier combinación de letras y números.

			-Campo obligatorio.
	Imagen	<i>varchar</i>	-Admitidos solamente los siguientes tipos de formatos: jpg, png, gif y jpeg -Campo opcional.
	Descripción	<i>varchar</i>	-Cualquier combinación de letras y números. -Campo opcional.
	Subir archivo	<i>varchar</i>	-Admitidos solamente los siguientes tipos de formatos: rar, zip, tar. -Campo obligatorio.
	Categorías	<i>varchar</i>	-Campo seleccionado. -Campo obligatorio.
	<b>Observaciones</b>	<p>El usuario para poder acceder a esta funcionalidad debe de estar autenticado en el sistema y poseer los permisos correspondientes para realizar esta opción.</p> <p>Después de subido el <i>software</i> se muestra una ventana con la información del nuevo <i>software</i> agregado pudiendo el usuario hacer votaciones y comentarios del mismo, así como compartirlos con los amigos pertenecientes a la Red Social.</p>	


**Tabla 2:** Descripción del requisito funcional Adicionar *software*.

No.	Nombre	Descripción	Complejidad	Prioridad para el cliente
RF_5	Búsqueda básica	Para acceder a esta funcionalidad el usuario va al buscador que se encuentra en la parte superior de la página principal y teclea el nombre del <i>software</i> que desea y pulsa la opción Buscar (IU: Opción buscar). Posteriormente el sistema muestra los resultados	Baja	Alta

	<p>de la búsqueda (IU: Opción resultados de la búsqueda). Finalmente el usuario selecciona el <i>software</i> buscado.</p>		
<b>Prototipo de Interfaz</b>			
<div style="text-align: center;">  <p><b>IU: Buscar.</b></p>  <p><b>IU: Resultados de la búsqueda.</b></p> </div>			
<b>Campos</b>	<b>Tipo de Datos</b>	<b>Reglas o Restricciones</b>	
Buscar	<i>String</i>	-Cualquier combinación de letras y números	
<b>Observaciones</b>	<p>El usuario puede acceder a esta funcionalidad sin estar autenticado. Después de seleccionar el <i>software</i> se muestra una ventana con la información del <i>software</i> pudiendo el usuario descargarlo y si está autenticado puede realizar votaciones y comentarios del mismo así</p>		

		como compartarlos con los demás amigos pertenecientes a la red social.
--	--	--


Tabla 3: Descripción del requisito funcional Búsqueda básica.

No.	Nombre	Descripción	Complejidad	Prioridad para el cliente
RF_7	Descargar software	Para acceder a esta funcionalidad el usuario va al software que desea descargar dando clic sobre él y le muestra la interfaz descargar el software (IU: Opción Descargar software).	Baja	Alta
<b>Prototipo de Interfaz</b>				
				
<b>IU: Descargar software.</b>				
<b>Observaciones</b>	El usuario puede acceder a esta funcionalidad sin estar autenticado.			



		Después de seleccionar el <i>software</i> se muestra una ventana con la información del <i>software</i> pudiendo el usuario si está autenticado realizar votaciones y comentarios del mismo así como compartirlos con los demás amigos pertenecientes a la red social.
--	--	--

**Tabla 4:** Descripción del requisito funcional Descargar *software*.

No.	Nombre	Descripción	Complejidad	Prioridad para el cliente
RF_10	Aprobar propuesta	Para acceder a esta funcionalidad el usuario va al menú Propuestas que se encuentra en la parte superior de la ventana (IU: Opción Propuestas). A continuación se le muestra una vista con las propuestas existentes (IU: Propuestas existentes), selecciona las propuestas que desea incluir en el sistema y selecciona la opción aprobar seleccionadas. Finalmente la propuesta pasa a ser un <i>software</i> del sistema y muestra un mensaje diciendo la propuesta aprobada.	Media	Alta
<b>Prototipo de Interfaz</b>				
				
<b>IU: Propuestas.</b>				

	 <p style="text-align: center;"><b>IU: Propuestas existentes.</b></p>
<p><b>Observaciones</b></p>	<p>El usuario para poder acceder a esta funcionalidad debe de estar autenticado en el sistema y poseer los permisos correspondientes para realizar esta opción. Después de crear la propuesta se muestra una ventana con las propuestas existentes.</p>

Tabla 5: Descripción del requisito funcional Aprobar propuesta.

### 2.4.2 Requisitos no funcionales

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable.

RNF1. Usabilidad	
No	Requisito
RNF1.1	La aplicación podrá ser utilizada por todos los usuarios de la Red Social de la Universidad.
RNF1.2	Para su uso se necesitan conocimientos básicos sobre aplicaciones web y navegadores en general.
RNF2. Seguridad	
No	Requisito
RNF2.1	Identificar al usuario antes de que pueda realizar cualquier acción de entrada de datos en el sistema.

## Capítulo II: Caracterización del Sistema de gestión de productos de software de la Red Social de la Universidad de las Ciencias Informáticas

RNF2.2	Proteger la información manejada por el sistema de accesos no autorizados.
RNF2.3	Garantizar que las funcionalidades del sistema se muestren de acuerdo al nivel del usuario que esté activo.
RNF2.4	Proteger la información manejada por el sistema contra la corrupción de ficheros o estados inconsistentes.
<b>RNF3. Portabilidad</b>	
<b>No</b>	<b>Requisito</b>
RNF3.1	Compatible con sistemas operativos de la familia Windows o Linux.
<b>RNF4. Soporte</b>	
<b>No</b>	<b>Requisito</b>
RNF4.1	EL Sistema de Base de Datos debe tener capacidad para soportar grandes cantidades de información, rápido procesamiento de datos y respuestas del sistema ante cualquier petición.
<b>RNF5. Restricciones de diseño</b>	
<b>No</b>	<b>Requisito</b>
RNF5.1	El sistema gestor de base de datos es PostgreSQL 8.4.
RNF5.2	El servidor web es el apache 2.0.
RNF5.3	Lenguaje de programación PHP5.
<b>RNF6. Interfaz hardware</b>	
<b>No</b>	<b>Requisito</b>
RNF6.1	Debe existir una red de área local para que los usuarios de la Universidad puedan acceder a la aplicación.
RNF6.2	Para el desarrollo PC Pentium 4 o superior, CPU 1.6 GHZ o superior, 1 GB RAM y 40 GB HDD mínimo.
RNF6.3	Para explotación del cliente PC Pentium 2 o superior, CPU 166 MHZ o superior, 256 RAM mínimo.
RNF6.4	Para explotación del servidor CPU Dual Core 2.0 GHZ o superior, 4 GB RAM o superior y 70 GB HDD mínimo.
RNF6.5	Para explotación del servidor de sistema de archivos CPU Dual Core 2.0 GHZ o

	superior, 4 GB RAM o superior y 660 GB HDD mínimo.
<b>RNF7. Interfaz externa</b>	
<b>No</b>	<b>Requisito</b>
RNF7.1	Diseño gráfico con navegación sencilla, donde el usuario pueda usar cómodamente la aplicación.
RNF7.2	No debe tener animaciones ni imágenes pesadas que obstaculicen la rapidez de las transiciones del ancho de banda.
RNF7.3	Diseño gráfico orientado a cumplir con las pautas de diseño de la Universidad y el Centro de Informatización Universitaria (CENIA).
<b>RNF8. Eficiencia</b>	
<b>No</b>	<b>Requisito</b>
RNF8.1	El sistema debe soportar un tiempo de respuesta menor de 5 segundos.

**Tabla 6:** Descripción Requisitos no funcionales.

## 2.5 Arquitectura y patrones

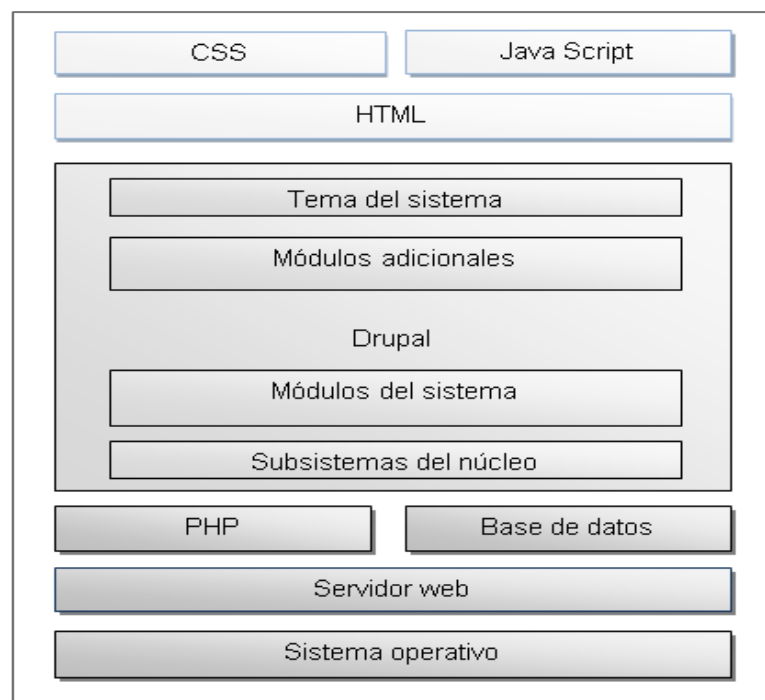
El sistema presenta una arquitectura en capas que se describen a continuación.

- **Capa 1 (Datos y Nodos):** la base del sistema es la colección de nodos (la piscina de datos). Antes de visualizar algo en el sitio, deben ser introducidos los datos.
- **Capa 2 (Módulos):** en esta capa se encuentran los módulos. Los módulos son complementos funcionales que son parte del núcleo de Drupal. Los módulos se basan en la funcionalidad básica de Drupal, que le permite personalizar los elementos de datos (campos) en sus tipos de nodos.
- **Capa 3 (Bloques y Menús):** en esta capa se encuentran los bloques y menús. Los bloques proporcionan a menudo la salida de un módulo o se pueden crear para mostrar lo que se desee y luego se puede colocar en varios puntos de la plantilla (*theme*) de diseño. Los bloques pueden ser configurados para la producción de diversas formas, así como sólo mostrando en ciertas páginas definidas, o sólo para ciertos usuarios definidos.
- **Capa 4 (Permisos de Usuario):** en esta capa se muestran los permisos de usuario. Aquí es donde se configuran los permisos de los diferentes tipos de usuarios. Los permisos se definen para varios papeles y a cada usuario se le asigna un papel.

- **Capa 5 (Tema):** en la última capa se encuentra el tema que se compone principalmente de XHTML y CSS, con algunas variables de PHP entremezcladas, por lo que el contenido generado por Drupal puede ir en los lugares apropiados. También se incluye con cada tema un conjunto de funciones que pueden ser utilizadas para reemplazar las funciones estándar en los módulos a fin de proporcionar un control completo sobre cómo los módulos generan su marcado en el momento de salida. Las plantillas también pueden ser asignados en la marcha sobre la base de los permisos de usuario.(HUNTER, 2008)

### 2.5.1 La pila tecnológica de Drupal

La pila de Drupal, que aparece en la siguiente figura, es como una capa intermedia entre Internet y la interfaz.



**Figura 2:** Pila tecnológica de Drupal.

En las capas inferiores de la pila se encuentra el sistema operativo, el servidor web, la base de datos y PHP ejecutándolo. El sistema operativo se encarga de las tareas que mantienen al sitio web en funcionamiento, el servidor haciendo que sea accesible, la base de datos almacenando la información del sitio y PHP para procesar la información de la base de datos.

Drupal se compone de capas, su capa más baja proporciona funcionalidad adicional encima de PHP, añadiendo subsistemas como manejo de sesión de usuario, autenticación, filtro de seguridad e interpretación de plantillas. Esta se encuentra encima de la capa construida con los elementos de

funcionalidad de Drupal, los módulos, encargados de generar los contenidos de cualquier página. Antes de mostrar una página a un usuario, esta es ejecutada a través del sistema de temas. La página de salida habitualmente XHTML, emplea CSS para controlar el diseño y los colores de la página y JavaScript para los elementos dinámicos de la misma, como campos de formularios o campos plegables.(HURTADO, 2010)

### **2.5.2 Principales módulos utilizados en el sistema**

Para el desarrollo del Catálogo de *software* de la Universidad, es necesaria la utilización de varios módulos del CMS Drupal que permiten un mejor rendimiento y funcionalidad del sistema:

- **Comment:** permite a los usuarios discutir y comentar contenido publicado.
- **Menú:** permite a los administradores personalizar el menú de navegación del sitio.
- **Views:** este módulo proporciona un método flexible para los diseñadores de sitios en Drupal para controlar cómo las listas de contenido (nodos) se presentan. Esta herramienta es esencialmente un constructor de consultas inteligentes. Puede construir la adecuada consulta, ejecutarlo y mostrar los resultados. Dispone de cuatro modos, además de un modo especial y ofrece gran cantidad de funcionalidades.
- **Taxonomy:** activa la categorización del contenido.
- **Search:** permite la recuperación y visualización de la información referida a contenidos, usuarios y otros criterios de búsqueda.
- **UCIAuthentication:** se integra con el dominio de la Universidad. Se usa para controlar el acceso y autenticación al sistema y los niveles de permisos.
- **Votingapi:** permite a los usuarios votar por el *software* publicado.
- **Visitors:** se utiliza para mostrar una información de los visitantes.
- **RSintegration:** este módulo permite integrarse a la Red Social, mostrando al usuario su lista de amigos de la Red Social permitiendo compartir el *software*, los amigos son notificados mediante un correo electrónico con la URL del *software*.

### **2.5.3 Patrones de diseño**

Los patrones de diseño representan de forma esquemática la estructura del diseño de un sistema. Pueden además ser la solución a problemas específicos.

Los patrones de diseño ayudan a los diseñadores a reutilizar con éxito diseños para obtener nuevos. El objetivo de los patrones es guardar la experiencia en diseños de programas orientados a objetos haciendo a su vez más fácil reutilizar con éxito los diseños y arquitecturas, ayudan a elegir entre

diseños alternativos, hacen a un sistema reutilizable y evitan alternativas que comprometen la reutilización.

El CMS Drupal funciona con la ayuda de patrones de diseño de tipo GOF (Gang of four) lo cual determina aplicaciones con un diseño bien definido y estructuralmente robusto.

Los patrones GOF pueden estar clasificados en tres categorías:

- **Patrones de creación:** muestran la guía de cómo crear objetos cuando sus creaciones requieren tomar decisiones. Estas decisiones normalmente serán resueltas dinámicamente mostrando que clases instanciar o sobre que objetos un objeto delegará responsabilidades.
- **Patrones estructurales:** describen la forma en que diferentes tipos de objetos pueden ser organizados para trabajar unos con otros.
- **Patrones de comportamiento:** se utilizan para organizar, manejar y combinar comportamientos. (MARTÍN, LIBER MATOS, 2010)

Patrones utilizados dentro de cada una de las clasificaciones seguidas por Drupal:

### Creación:

- **Singleton (Única instancia):** tomando los módulos y los temas como objetos, y siguiendo el patrón Singleton, estos no encapsulan datos. Lo que separa a un módulo de otro es el conjunto de funciones que contiene, por lo que debe ser pensado como una clase con un ejemplo aislado. Dicho patrón está presente en el tema y todos los módulos implementados en el sistema propuesto.

### Estructurales:

- **Decorator (Decorador):** esta característica permite una amplia variedad de comportamientos que se añaden a los nodos. Una necesidad común es para los archivos cargados y que se adjunta a un nodo, por lo que se podría diseñar un nuevo tipo de nodo con la posibilidad de adjuntar archivos. El principal objetivo del sistema es la descarga de *software* donde tienen que estar previamente adjuntos al sistema, acompañado de imágenes y descripciones satisfaciendo esta necesidad la concesión de cada nodo que le solicite la posibilidad de tener archivos adjuntos.

### Comportamiento

- **Observer (Observador):** el patrón *Observer* es generalizado en Drupal. Cuando una modificación es hecha a un vocabulario en el sistema de taxonomía de Drupal, el gancho de taxonomía es llamado en todos los módulos que lo implementan. Mediante la aplicación del gancho, se han registrado como observadores del objeto vocabulario.

- **Chain of responsibility (Cadena de responsabilidad):** el sistema de menús de Drupal sigue el patrón *Chain of Responsibility*. 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 la función se llama para hacer el trabajo. Para ello, el mensaje se pasa a la opción del menú correspondiente a la vía de la solicitud. Si el elemento de menú no puede manejar la petición, se pasa de la cadena. Esto continúa hasta que un módulo se encarga de la petición, un módulo niega el acceso para el usuario, o la cadena se ha agotado.

### **Conclusiones parciales**

La definición de los requisitos funcionales y no funcionales permitió obtener el diseño y la realización de la propuesta de solución a la problemática planteada. Con el análisis y diseño realizado se logró un mejor entendimiento de las principales funcionalidades del Sistema de gestión de productos de *software* para la Red Social de la Universidad de las Ciencias Informáticas.



**Capítulo III: Implementación y validación del Sistema de gestión de productos de software de la Red Social de la Universidad de las Ciencias Informáticas**



**Introducción**

Para lograr una estandarización en el código del Sistema para la gestión de *software* de la Red Social de la Universidad de las Ciencias Informáticas se definen los estándares de código. Las pruebas de *software* tienen como objetivo principal detectar posibles errores que puedan ocurrir durante el flujo de trabajo de la aplicación. Las pruebas constituyen un instrumento para determinar la calidad de un producto *software*. El proceso de prueba puede estar destinado a efectuarse sobre componentes de un sistema o sobre el sistema en sí.

**3.1 Estándares de codificación definidos por el CENIA**

Un estándar de codificación completo comprende todos los aspectos de la generación de código. Un código fuente completo debe reflejar un estilo armonioso, como si un único programador hubiera escrito todo el código de una sola vez. Usar técnicas de codificación sólidas y realizar buenas prácticas de programación con vistas a generar un código de alta calidad es de gran importancia para la calidad del *software* y para obtener un buen rendimiento. Los estándares de código utilizados en el desarrollo del sistema son los definidos por el CENIA, estos se encuentran en el expediente de proyecto.

**3.2 Diagrama de despliegue**

El diagrama de despliegue permite apreciar de forma visual como se encuentran relacionados físicamente los componentes de la aplicación. (AVILA, 2009) En este caso la aplicación se encuentra hospedada en un servidor web, la misma se comunica con el sistema de gestión de base de datos PostgreSQL que se encuentra en otro servidor. Se emplea como protocolo de comunicación entre el Cliente y el Servidor web el protocolo HTTP debido a que está basado en el modelo cliente-servidor.

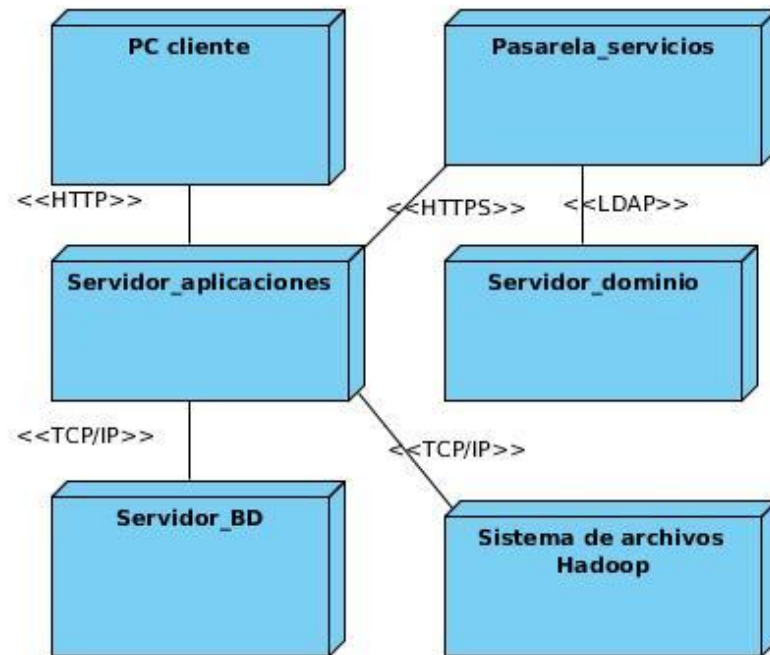


Figura 3: Modelo de despliegue.

- **PC cliente:** la PC-cliente ejecuta la aplicación con un navegador web que puede ser: Internet Explorer, Mozilla Firefox, Opera, Safari, Chrome entre otros. Se comunican con el servidor web mediante el protocolo HTTP.
- **Servidor-Aplicaciones:** no es más que un servidor web que hace uso extensivo del contenido dinámico y tiene una fuerte integración con las bases de datos, ya que maneja la mayoría de las transacciones relacionadas con la lógica y el acceso a los datos de la aplicación, utilizando para ello el protocolo TCP/IP. El servidor de aplicaciones es el que permite que las PC-cliente interactúen y tengan acceso a la aplicación. Brinda servicios web que ayudan a la autenticación y la detección de intrusos en el sistema.
- **Servidor-BD:** los servidores de bases de datos son programas que permiten organizar datos en una o más tablas relacionadas. Se utilizan para cualquier aplicación que necesite almacenamiento, acceso y análisis de datos estructurados. El servidor de base datos utilizado en la aplicación es PostgreSQL 8.4
- **Servidor-Sistema-Archivos:** el *Hadoop Distributed File System* o en español Sistema de Archivo Distribuido Hadoop (HDFS), permite el almacenamiento y transferencias de archivos en un ambiente distribuido. Hadoop es altamente tolerante a fallos y está diseñado para ser implementado en *hardware* de bajo costo, proporciona acceso de alto rendimiento de datos de aplicación y es adecuado para aplicaciones que tienen grandes conjuntos de datos.

- **Pasarela-servicios:** la pasarela de autenticación es un servicio diseñado con el objetivo de garantizar seguridad y confiabilidad a la hora de interactuar con el servidor de dominio. A través de ella se realiza el proceso de autenticación de los usuarios en las diferentes aplicaciones que usan el servicio sin necesidad de que las mismas interactúen de forma directa con el servidor que contiene el directorio activo.

### **3.3 Pruebas de caja negra**

Pruebas que se llevan a cabo sobre la interfaz del *software*. El objetivo es demostrar que las funciones del *software* son operativas, que las entradas se aceptan de forma adecuada y se produce un resultado correcto, y que la integridad de la información externa se mantiene. Estas pruebas comprueban si el sistema cumple con los requisitos funcionales, ignorando la estructura lógica interna del *software*. Se centran principalmente en los requisitos funcionales del *software* que permiten encontrar: Funciones incorrectas o ausentes, errores de interfaz, errores en estructuras de datos o en accesos a las bases de datos externas, errores de rendimiento, errores de inicialización y terminación. (MUSTELIER, 2011)

### 3.3.1 Casos de prueba

#### Caso de prueba: Adicionar software.

**Descripción:** para acceder a esta funcionalidad el usuario va a el menú de atajos, luego elige la opción “Agregar contenido”, posteriormente selecciona la opción *software* donde aparece un formulario para llenar los datos del *software*.

**Condiciones de ejecución:** el usuario debe estar autenticado en el sistema con el rol de administrador.

Escenario	Descripción	Título	Imagen	Descripción	Fichero	Categoría	Respuesta del sistema	Flujo central
EC 1.1 Adicionar <i>software</i> de forma correcta.	El sistema adiciona un <i>software</i> .	V (Pandion).	V (icono del <i>software</i> ).	V (una breve descripción de sus principales funcionalidades).	V (se adjunta el <i>software</i> ).	V (se selecciona la categoría a la cual pertenece).	El sistema guarda los datos y muestra un mensaje “El <i>software</i> Pandion se ha creado”.	<ol style="list-style-type: none"> <li>1. El usuario accede a la aplicación.</li> <li>2. Selecciona la opción “menú de atajos”.</li> <li>3. Selecciona la opción “Agregar contenido”.</li> <li>4. Selecciona la opción “<i>software</i>”.</li> <li>5. El usuario llena el formulario con los datos del <i>software</i>.</li> <li>6. Selecciona la opción “Guardar”.</li> </ol>

								7. El sistema muestra el mensaje “El <i>software</i> Pandion se ha creado”.
EC 1.2 Adicionar <i>software</i> dejando campos vacíos.	Autenticar al usuario para que pueda acceder a la aplicación.	V (vacío).	V (vacío).	V (vacío).	V (vacío).	V (vacío).	El sistema verifica los datos y muestra los mensajes “el campo título es obligatorio”, “el campo categoría es obligatorio”, “el campo fichero es obligatorio”.	<ol style="list-style-type: none"> <li>1. El usuario accede a la aplicación.</li> <li>2. Selecciona la opción “menú de atajos”.</li> <li>3. Selecciona la opción “Agregar contenido”.</li> <li>4. Selecciona la opción “<i>software</i>”.</li> <li>5. El usuario llena el formulario con los datos del <i>software</i> dejando campos vacíos.</li> <li>6. Selecciona la opción “Guardar”.</li> <li>7. El sistema muestra los mensajes “el campo título es obligatorio”, “el campo categoría es obligatorio”, “el campo fichero es obligatorio”.</li> </ol>

**Tabla 7:** Caso de prueba Adicionar *software*.

No	Nombre del campo	Clasificación	Valor Nulo	Descripción
1.	Título	Campo de texto	No	El nombre del <i>software</i> que se agregará.
2.	Imagen	Campo de subir archivo	Si	Se adjunta el icono del <i>software</i> .
3.	Descripción	Campo de texto	Si	Breve descripción de sus principales funcionalidades.
4.	Fichero	Campo de subir archivo	No	Se adjunta el <i>software</i> .
5.	Categoría	Lista de selección	No	Se selecciona la categoría a la cual pertenece el <i>software</i> .

**Tabla 8:** Descripción de los campos Caso de prueba Adicionar *software*.

**Caso de prueba: Editar *software*.**

**Descripción:** para acceder a esta funcionalidad el usuario va al *software* que desea editar y lo selecciona, luego selecciona la opción “editar”.

**Condiciones de ejecución:** el usuario debe estar autenticado en el sistema con el rol de administrador.

Escenario	Descripción	Título	Imagen	Descripción	Fichero	Categoría	Respuesta del sistema	Flujo central
EC 1.1	El sistema	V	V (icono	V (una	V (se	V (se	El sistema guarda los	1. El usuario accede a la aplicación.

<p>Editar <i>software</i> de forma correcta.</p>	<p>adiciona un <i>software</i>.</p>	<p>(Pandion).</p>	<p>del <i>software</i>).</p>	<p>breve descripción de sus principales funcionalidades).</p>	<p>adjunta el <i>software</i>).</p>	<p>selecciona la categoría a la cual pertenece).</p>	<p>datos y muestra un mensaje “El <i>software</i> Pandion se ha creado”.</p>	<ol style="list-style-type: none"> <li>2. Selecciona el <i>software</i> que desea editar.</li> <li>3. Selecciona la opción “Editar”.</li> <li>4. El usuario edita el contenido del formulario con los datos del <i>software</i>.</li> <li>5. Selecciona la opción “Guardar”.</li> <li>6. El sistema muestra el mensaje “<i>software</i> (nombre) ha sido actualizado”.</li> </ol>
<p>EC 1.2 Editar <i>software</i> dejando campos vacíos.</p>	<p>Autenticar al usuario para que pueda acceder a la aplicación.</p>	<p>V (Pandion).</p>	<p>V (vacío).</p>	<p>V (vacío).</p>	<p>V (vacío).</p>	<p>V (vacío).</p>	<p>El sistema verifica los datos y muestra los mensajes “el campo categoría es obligatorio”, “el campo fichero es obligatorio”.</p>	<ol style="list-style-type: none"> <li>1. El usuario accede a la aplicación.</li> <li>2. Selecciona la opción “menú de atajos”.</li> <li>3. Selecciona la opción “Agregar contenido”.</li> <li>4. Selecciona la opción “<i>software</i>”.</li> <li>5. El usuario llena el formulario con los datos del <i>software</i> dejando campos vacíos.</li> </ol>

								6. Selecciona la opción "Guardar". 7. El sistema muestra los mensajes "el campo categoría es obligatorio", "el campo fichero es obligatorio".
--	--	--	--	--	--	--	--	--

**Tabla 9:** Descripción caso de prueba Editar *software*.

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
1.	Título	Campo de texto	No	El nombre del <i>software</i> que se agregará.
2.	Imagen	Campo de subir archivo	Si	Se adjunta el icono del <i>software</i> .
3.	Descripción	Campo de texto	Si	Breve descripción de sus principales funcionalidades.
4.	Fichero	Campo de subir archivo	No	Se adjunta el <i>software</i> .
5.	Categoría	Lista de selección	No	Se selecciona la categoría a la cual pertenece el <i>software</i> .

**Tabla 10:** Descripción campos caso de prueba Editar *software*.

**Caso de prueba: Búsqueda básica.**

**Descripción:** el usuario accede a la aplicación y en la parte superior de la aplicación aparece el buscador y escribe el nombre del *software* que desea buscar.



Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Buscar <i>software</i> obteniendo resultado.	Busca todo el <i>software</i> que contenga la palabra escrita en el buscador.	Muestra los productos de <i>software</i> que contengan la palabra escrita en el buscador.	<ol style="list-style-type: none"> <li>1. Acceder a la aplicación.</li> <li>2. Escribir la palabra en el buscador.</li> <li>3. Muestra una lista con los <i>software</i> que contenga la palabra escrita en el buscador.</li> </ol>
EC 1.2 Buscar <i>software</i> sin obtener resultados.	Busca todo el <i>software</i> que contenga la palabra escrita en el buscador.	Muestra un mensaje “su búsqueda no produjo resultados”.	<ol style="list-style-type: none"> <li>1. Acceder a la aplicación.</li> <li>2. Escribir la palabra en el buscador.</li> <li>3. Muestra un mensaje “su búsqueda no produjo resultados”.</li> </ol>

**Tabla 11:** Descripción caso de prueba Búsqueda básica.

No	Nombre del campo	Clasificación	Valor Nulo	Descripción
1	Buscador	Campo de texto	Si	Palabra o frase que tenga relación con lo que le interesa al usuario buscar en la aplicación.

**Tabla 12:** Descripción campos caso de prueba Búsqueda básica.

**Caso de prueba: Descargar *software*.**

**Descripción:** El usuario accede a la aplicación, busca el *software* que desea descargar lo selecciona y se le muestra una vista para guardar el *software* en el lugar que usuario desee.

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Descargar el <i>software</i> de forma correcta.	El usuario busca el <i>software</i> que quiere y lo descargar.	Muestra una interfaz dándole la opción de escoger el lugar a descargar el <i>software</i> .	<ol style="list-style-type: none"> <li>1. Acceder a la aplicación.</li> <li>2. Selecciona el <i>software</i> a descargar.</li> <li>3. Guarda en donde desee el <i>software</i>.</li> </ol>

**Tabla 13:** Descripción caso de prueba Descargar *software*.

**Caso de prueba: Proponer *software*.**

**Descripción:** para acceder a esta funcionalidad el usuario va al menú Proponer, el cual debe de ser llenado para poder subir el *software*. Finalmente se guardan el formulario adicionando el *software* al sistema el cual muestra un mensaje de información diciendo que la propuesta se ha creado.

**Condiciones de ejecución:** el usuario debe estar autenticado en el sistema.

Escenario	Descripción	Nombre	Imagen	Descripción	Fichero	Categoría	Respuesta del sistema	Flujo central
EC 1.1 Proponer <i>software</i> de forma correcta.	El sistema adiciona un <i>software</i> a la lista de	V (Firefox).	V (icono del <i>software</i> ).	V (una breve descripción de sus	V (se adjunta el <i>software</i> ).	V (se selecciona la categoría a la	El sistema guarda los datos y muestra un	<ol style="list-style-type: none"> <li>1. El usuario accede a la aplicación.</li> <li>2. Selecciona la opción "Proponer".</li> <li>3. El usuario llena el formulario con los datos</li> </ol>

	propuestas.			principales funcionalidades).		cual pertenece).	mensaje "Propuesta Firefox se ha creado".	del software. 4. Selecciona la opción "Guardar". 5. El sistema muestra el mensaje "Propuesta Firefox se ha creado".
EC 1.2 Editar software dejando campos vacíos.	El sistema adiciona un software a la lista de propuestas.	V (vacío).	V (vacío).	V (vacío).	V (vacío).	V (vacío).	El sistema verifica los datos y muestra los mensajes "el campo categoría es obligatorio", "el campo fichero es obligatorio".	1. El usuario accede a la aplicación. 2. Selecciona la opción "menú de atajos". 3. Selecciona la opción "Agregar contenido". 4. Selecciona la opción "software". 5. El usuario llena el formulario con los datos del software dejando campos vacíos. 6. Selecciona la opción "Guardar". 7. El sistema muestra los mensajes "el campo nombre es obligatorio".

**Tabla 14:** Descripción caso de prueba Proponer software.

No	Nombre del campo	Clasificación	Valor Nulo	Descripción
1.	Nombre	Campo de texto	No	El nombre del <i>software</i> que se agregará.
2.	Imagen	Campo de subir archivo	Si	Se adjunta el icono del <i>software</i> .
3.	Descripción	Campo de texto	Si	Breve descripción de sus principales funcionalidades.
4.	Fichero	Campo de subir archivo	No	Se adjunta el <i>software</i> .
5.	Categoría	Lista de selección	No	Se selecciona la categoría a la cual pertenece el <i>software</i> .

**Tabla 15:** Descripción campos caso de prueba Proponer *software*.

### 3.4 Resultados de las pruebas

El gráfico que se muestra a continuación establece una relación entre las iteraciones de pruebas realizadas y el número de no conformidades detectadas en las mismas.



**Figura 4:** Cantidad de no conformidades por iteraciones.

En la figura mostrada anteriormente se puede observar cómo en la primera iteración se detectaron 9 no conformidades, número que se fue reduciendo hasta la tercera revisión donde no se obtuvieron no conformidades. Cada una de las no conformidades detectadas fueron resueltas inmediatamente por el equipo de desarrollo. La solución de estas no conformidades ayudó a que la aplicación alcanzara una mejor calidad y cumpliera con los requisitos planteados.

### 3.5 Prueba de carga y estrés

Esta prueba puede mostrar los tiempos de respuesta de todas las transacciones importantes de la aplicación. Si la base de datos, el servidor de aplicaciones y otros componentes sufren afectaciones durante la carga, la carga se diseña lo más real posible y mide anticipadamente el rendimiento que proporcionará la aplicación. Identifica los puntos de ruptura de la aplicación por debajo de la demanda máxima. (MUSTELIER, 2011)

#### Caso de prueba

Para realizar las pruebas de carga y estrés se definen los escenarios: EC1, EC2 y EC3.

Características del *hardware* de prueba.

Hardware	Datos
Microprocesador	1.6 GHz
Memoria RAM	1 GB
Ancho de banda	10/100 Mbps

**Tabla 16:** Características del hardware de prueba.

Características del *software* de prueba.

Hardware	Datos
Sistema Operativo	Ubuntu 11.4
Gestor de Base de Datos	PostgreSQL 8.4.
Antivirus	Kaspersky Workstation 6.0
<i>Software</i> de prueba	Jmeter Versión 2.3.4

**Tabla 17:** Características del *software* de prueba.

Carga de trabajo y resultados esperados de los escenarios.

Identificador del escenario	Escenarios de la sección	Carga de trabajo	Descripción	Resultados esperados	Resultado obtenidos de la prueba
EC1	Acceder al sistema.	20	Acceder al sistema dirección: <a href="http://10.53.7.234/ucisoftware">http://10.53.7.234/ucisoftware</a>	La aplicación debe responder en un tiempo menor de 2.50 segundos.	La aplicación responde en un tiempo (parámetro de rendimiento de Jmeter) de 2.29 segundos.
EC2	Acceder a un <i>software</i> .	20	Acceder a un <i>software</i> : <a href="http://10.53.7.234/ucisoftware/node/4">http://10.53.7.234/ucisoftware/node/4</a>	La aplicación debe responder en un tiempo menor de 4	La aplicación responde en un tiempo (parámetro de

				segundos.	rendimiento de Jmeter) de 3.8 segundos.
EC3	Descargar un software.	20	Entrar al sistema dirección: <a href="http://10.53.7.234/ucissoftware/system/files/repositorio/firefox">http://10.53.7.234/ucissoftware/system/files/repositorio/firefox</a>	La aplicación debe responder en un tiempo menor de 1.5 segundos.	La aplicación responde en un tiempo (parámetro de rendimiento de Jmeter) de 1.3 segundos.

**Tabla 18:** Carga de trabajo y resultados esperados de los escenarios.

### 3.5.1 Análisis e interpretación de los resultados

Con respecto al rendimiento de la aplicación para los EC1, EC2 y EC3 no se encontraron ninguna NC, esto se evidencia en las siguientes tablas que contienen los resultados generados por el informe agregado que se escoge como plan de prueba de la herramienta Jmeter para mostrar los valores de las pruebas.

Resultados EC1: Acceder al sistema.

Petición	# Muestra	Media	Mediana	Línea de 90%	Min	Max	%Error	Rendimiento	KB/S
total	1040	3093	41	732.1	0.0	126389	0,0	2,29 segundos	15,6

**Tabla 19:** Resultados EC1. Acceder al sistema.

Resultados EC2: Acceder a un software.

Petición	# Muestra	Media	Mediana	Línea de 90%	Min	Max	%Error	Rendimiento	KB/S
----------	-----------	-------	---------	--------------	-----	-----	--------	-------------	------

total	740	4760	45	210	1	186683	0,0	3,8 segundos	23,3
-------	-----	------	----	-----	---	--------	-----	--------------	------

**Tabla 20:** Resultados EC2: Acceder a un *software*.

Resultados EC3: Descargar un *software*.

Petición	# Muestra	Media	Mediana	Línea de 90%	Min	Max	%Error	Rendimiento	KB/S
total	20	15370	15150	16798	1	16841	0,0	1,13 segundos	25,273

**Tabla 21:** Resultados EC3: Descargar un *software*.

### Significado de los elementos del informe agregado

- **# Muestra:** cantidad de páginas (hilos) que simulan la cantidad de usuarios, que están interactuando con el sistema desde la misma URL.
- **Media:** media de páginas que se cargaron satisfactoriamente.
- **Mediana:** tiempo promedio que ha tardado en cargarse las páginas.
- **Línea de 90 %:** el 90 % de las páginas se cargaron de forma satisfactoria.
- **Min:** tiempo mínimo en que ha demorado en cargarse una página.
- **Max:** tiempo máximo en que ha tardado en cargarse una página.
- **% Error:** por ciento de error de las páginas que no se cargaron de forma satisfactoria.
- **Rendimiento:** tiempo total que demoró en realizarse la cantidad de peticiones (páginas hilos) de la prueba.
- **KB/S:** tiempo total que demoró en realizarse la cantidad de peticiones (páginas hilos) de la prueba en Kbyte por segundo.

### Conclusiones parciales

La utilización del sistema implementado permite gestionar los programas de *software* de la Red Social de la Universidad de las Ciencias Informáticas todos los recursos publicados en la red. Aplicadas las técnicas de validación a la propuesta de solución, se comprobó el buen funcionamiento del sistema de acuerdo a los requisitos planteados.



## **Conclusiones generales**

La presente investigación tiene un papel fundamental en la definición e implementación del Sistema de gestión de productos de *software* para la Red Social de la Universidad de las Ciencias Informáticas

- Con el estudio de las diferentes funcionalidades de los sistemas que permiten la gestión de *software*, se definió el marco teórico conceptual de la investigación, que permitió identificar la problemática e identificar las bases para analizar, diseñar e implementar el Sistema de gestión de productos de *software* para la Red Social de la Universidad de las Ciencias Informáticas.
- La definición de los requisitos funcionales y no funcionales permitió obtener el diseño y la realización de la propuesta de solución a la problemática planteada.
- Con el análisis y diseño realizado se logró un mejor entendimiento de las principales funcionalidades del Sistema de gestión de productos de *software* para la Red Social de la Universidad de las Ciencias Informáticas.
- La utilización del sistema implementado permite gestionar los programas de *software* de la Red Social de la Universidad de las Ciencias Informáticas.
- Aplicadas las técnicas de validación a la propuesta de solución, se comprobó el buen funcionamiento del sistema de acuerdo a los requisitos planteados.

Al finalizar se obtiene una investigación sustentada en elementos teóricos, todos los artefactos definidos dentro de la Proceso de desarrollo de *software* seleccionado y un sistema que permite gestionar los productos de *software* de la Red Social Universitaria por lo que se puede decir que se cumplió con el objetivo principal del trabajo, obteniéndose los resultados esperados.

## **Recomendaciones**

Existen varias formas de catalogar y categorizar un *software* de acuerdo con las necesidades de información y su propósito. Como resultado de la investigación y del proceso de desarrollo de la aplicación, han surgido ideas que serían recomendables para futuras versiones del sistema y el perfeccionamiento del mismo.

- Agregarle nuevas funcionalidades de acuerdo con las expectativas que puedan surgir tanto en la comunidad universitaria como necesidades de la Red Social a la que está integrado.
- Agregar nuevas categorías o modificar las existentes a medida que crezca el número de *software* publicados o sean sujetas a análisis teniendo en cuenta las opiniones y sugerencias de los usuarios.

## **Bibliografía referenciada**

1. ALELÍ SÁNCHEZ MÉNDEZ, Y. R. V., ANAY DÍAZ ESTRADA *PROCEDIMIENTO PARA EL DESARROLLO DE SOFTWARE CON UN ENFOQUE ÁGIL Y CMMI NIVEL 2* 2012, Disponible en: <http://uciencia.uci.cu/es/node/1310>.
2. ALFONSO FUENTES, Y. *Análisis, diseño e implementación de un catálogo comercial digital*. 2010.
3. ALVAREZ, M. A. *Una introducción meramente conceptual al potente lenguaje de script del lado del cliente*. [Consultado el: 3 de diciembre de 2011]. Disponible en: <http://www.desarrolloweb.com/articulos/25.php>.
4. ALVAREZ, S. *Sistemas gestores de bases de datos* [Consultado el: 8 de diciembre de 2011]. Disponible en: <http://www.desarrolloweb.com/articulos/sistemas-gestores-bases-datos.html>.
5. AVILA, E. A. *Diagramas de implementación* [Consultado el: 15 de abril de 2012]. Disponible en: <http://www.slideshare.net/techmi/curso-uml-25-diagramas-de-implementacin>.
6. BAHIT, E. *Introduciéndonos en el Desarrollo Ágil de Software* [Consultado el: 13 de diciembre de 2011]. Disponible en: <http://www.desarrolloweb.com/articulos/intro-desarrollo-agil-software.html>.
7. BARRIOCANAL, L. *¿Qué es Joomla?* [Consultado el: 5 de diciembre de 2011]. Disponible en: <http://edujoomla.es/que-es-joomla>.
8. ESTEBAN, A. B. *Qué es Softonic* [Consultado el: 30 de noviembre de 2011]. Disponible en: <http://support.softonic.com/index.php?/Knowledgebase/Article/View/15/9/que-es-softonic>.
9. FOUNDATION, T. A. S. *Visión general de las nuevas funcionalidades de Apache 2.0* [Consultado el: 1 de diciembre de 2011]. Disponible en: [http://httpd.apache.org/docs/2.0/es/new\\_features\\_2\\_0.html](http://httpd.apache.org/docs/2.0/es/new_features_2_0.html).
10. GREHAN, R. *Zend Studio review* [Consultado el: 10 de diciembre de 2011]. Disponible en: <http://review.techworld.com/applications/3212820/zend-studio-review/>.
11. GUIA-UBUNTU.ORG. *Eclipse* [Consultado el: 3 de diciembre de 2011]. Disponible en: <http://www.guia-ubuntu.org/index.php?title=Eclipse>.
12. HTML.NET. *Lección 1: ¿Qué es CSS?* [Consultado el: 1 de diciembre de 2011]. Disponible en: <http://es.html.net/tutorials/css/lesson1.php>.
13. HUNTER, L. *La visión general de Drupal* [Consultado el: 21 de noviembre de 2011]. Disponible en: <http://drupal.org/getting-started/before/overview>.

14. HURTADO, B. H. *Desarrollo web con Drupal*. 2010, Disponible en:  
[http://eficiencia.urjc.es/bitstream/10115/4232/1/Memoria\\_PFC\\_Benjamin\\_Herrero\\_Hurtado.pdf](http://eficiencia.urjc.es/bitstream/10115/4232/1/Memoria_PFC_Benjamin_Herrero_Hurtado.pdf).
15. JMETER, A. *Apache JMeter* de 2 de febrero]. Disponible en: <http://jmeter.apache.org/>.
16. LAPUENTE, M. J. L. *Del catálogo tradicional al catálogo en línea* [Consultado el: 2 de diciembre de 2011]. Disponible en: <http://www.hipertexto.info/documentos/catalogo.htm>.
17. LAPUENTE, M. J. L. *Lenguajes hipertextuales*. 2011b, Disponible en:  
[http://www.hipertexto.info/documentos/lenguajes\\_h.htm](http://www.hipertexto.info/documentos/lenguajes_h.htm).
18. LIBRES, G. D. U. D. T. *¿Quiénes somos?* [Consultado el: 13 de noviembre de 2011]. Disponible en: <http://gutl.jovenclub.cu/quienes-somos/>.
19. LUCIANO. *Entornos de Desarrollo Integrado* [Consultado el: 13 de diciembre de 2011].  
Disponible en: <http://luauf.com/2008/05/13/entornos-de-desarrollo-integrado-para-java/>.
20. MARCO, B. S. *Recopilaciones de Software Libre* [Consultado el: 26 de noviembre de 2011].  
Disponible en: <http://www.cdlibre.org/otros/faq.html>.
21. MARTÍN, L. M. *POO y patrones de diseño en DRUPAL*. 2010, vol. 2012, Disponible en:  
<http://jitcode.blogspot.com/2010/06/poo-y-patrones-de-diseno-en-drupal.html>.
22. MARTÍN, R. A. *Open pyme* [Consultado el: 20 de febrero de 2012]. Disponible en:  
<http://www.slideshare.net/oswc2012/open-pyme-selibreoswc2012-11001183>.
23. MEJÍA, J. *Catalogación electrónica: Aplicación de Nuevas Tecnologías con beneficios reales* [Consultado el: 9 diciembre de 2011]. Disponible en:  
[http://www.articulo.org/articulo/23887/catalogacion\\_electronica\\_aplicacion\\_de\\_nuevas\\_tecnologias\\_con\\_beneficios\\_reales.html](http://www.articulo.org/articulo/23887/catalogacion_electronica_aplicacion_de_nuevas_tecnologias_con_beneficios_reales.html).
24. MUSTELIER, S. C. *Propuesta de Procedimiento y Herramienta para el desarrollo de Pruebas de Rendimiento de Carga y Estrés en el Grupo de Calidad de FORTES*. 2011.
25. MYSQL.COM. *MySQL 5.6 Manual* [Consultado el: 5 de febrero de 2012]. Disponible en:  
<http://dev.mysql.com/doc/refman/5.6/en/introduction.html>.
26. NETBEANS.ORG. *NetBeans IDE 7.1 Características* [Consultado el: 2 de febrero de 2012].  
Disponible en: <http://netbeans.org>.
27. ORTIZ, K. H. *Descripción del Modelo de Dominio* [Consultado el: 3 de febrero de 2012].  
Disponible en:  
<http://www.eumed.net/libros/2009c/583/Representacion%20del%20Modelo%20de%20Objetos%20de%20Dominio.htm>.
28. PHP.NET. *PHP Manual* [Consultado el: 16 enero de 2012]. Disponible en:  
<http://www.php.net/manual/es/index.php>.

29. POSTGRESQL.ORG. *PostgreSQL 8.4* [Consultado el: 6 diciembre de 2011]. Disponible en: <http://www.postgresql.org/about/news/1108/>.
30. RODRÍGUEZ, F. *UpToDown.com: Bajar programas gratis* [Consultado el: 3 de diciembre de 2011]. Disponible en: <http://www.iwebsi.com/t/uptodown-com>.
31. ROMERO, M. *Especificación de requerimientos, Ingeniería de Software* [Consultado el: 15 de enero de 2012]. Disponible en: <http://www.slideshare.net/marfonline/especificacin-de-requerimientos-ingenieria-de-software>.
32. TELL, M. *¿Qué es un CMS?* [Consultado el: 3 de diciembre de 2011]. Disponible en: <http://www.joomladay.es/guia-joomla-2/%C2%BFque-es-un-cms/>.
33. TOMLINSON, T. *Pro Drupal 7 Development*. 2010.
34. UBUNTU, G. D. P. *PgAdmin III* [Consultado el: 10 de diciembre de 2011]. Disponible en: [http://guia-ubuntu.org/index.php?title=PgAdmin\\_III](http://guia-ubuntu.org/index.php?title=PgAdmin_III).
35. VERGARA, K. *Concepto y tipos de software*. 2007.
36. VISUAL-PARADIGM.COM. *Visual Paradigm for UML* [Consultado el: 15 marzo de 2012]. Disponible en: <http://www.visual-paradigm.com/product/vpum/>.
37. WORDPRESS, A. *¿Que es WordPress?* [Consultado el: 5 de abril de 2012]. Disponible en: <http://ayudawordpress.com/%C2%BFque-es-wordpress/>.

## Bibliografía consultada

1. ABREU, D. J. *Personalización del Sistema de Gestión de Contenidos Drupal para la implementación de Revistas Científicas Electrónicas*. 2010.
2. ACUÑA, F. E. P. *Catálogo de software de la Universidad de las Ciencias Informáticas*. 2011.
3. ALELÍ SÁNCHEZ MÉNDEZ, Y. R. V., ANAY DÍAZ ESTRADA *PROCEDIMIENTO PARA EL DESARROLLO DE SOFTWARE CON UN ENFOQUE ÁGIL Y CMMI NIVEL 2* 2012, Disponible en: <http://uciencia.uci.cu/es/node/1310>.
4. ALFARO, F. M. *Herramientas Case*. 2008.
5. ALFONSO FUENTES, Y. *Análisis, diseño e implementación de un catálogo comercial digital* 2010.
6. ALVAREZ, M. A. *Una introducción meramente conceptual al potente lenguaje de script del lado del cliente*.  
Disponible en: <http://www.desarrolloweb.com/articulos/25.php>.
7. APACHE.ORG. *Apache* Disponible en: <http://www.apache.org>.
8. BARRIOCANAL, L. *¿Qué es Joomla?* . 2009, nº Disponible en: <http://edujoomla.es/que-es-joomla>.
9. BATISTA, A. L. S. *Personalización de Drupal para teléfonos móviles*. 2010.
10. CDLIBRE.ORG. *Cdlibre* Disponible en: <http://cdlibre.org>.
11. DIEPPA, E. R. Á. *Tienda virtual, Portal Promocional del Catálogo de Productos del Polo de SWL*. 2010.
12. DRUPAL.ORG. *drupal.org* Disponible en: <http://drupal.org>.
13. DUROCHER, C. *Display description text in file listing table* Disponible en:  
<http://drupal.org/node/289988>.
14. FERNÁNDEZ, T. C. *CATÁLOGO DE PRODUCTOS DE LA FACULTAD 8*. 2007.
15. FORMANTÍN, Y. P. *GUÍA PARA EL TRABAJO CON EL CMS DRUPAL*. 2007.
16. GONZÁLEZ, E. C. *Catalogación de técnicas de reducción de los efectos negativos de la latencia en simulaciones y juegos virtuales*. 2008.
17. GRAF, H. *Guía para Joomla 1.7 Español*. 2012, Disponible en:  
<http://sunshine.prod.uci.cu/gridfs/sunshine/books/j17es.pdf>.
18. HERNÁNDEZ, R. R. *PROPUESTA DE METODOLOGÍA WEB BASADA EN EL CMS DRUPAL PARA EL DESARROLLO DEL PORTAL WEB DEL CICPC*. 2008.

19. IBM. *Rational Rose Enterprise* Disponible en:  
<http://www.rational.com.ar/herramientas/roseenterprise.html>.
20. JEFF FRENTZEN, H. S. *Superutilidades para Java Script*. 2005, Disponible en:  
[http://sunshine.prod.uci.cu/gridfs/sunshine/books/Superutilidades\\_para\\_JavaScript.pdf](http://sunshine.prod.uci.cu/gridfs/sunshine/books/Superutilidades_para_JavaScript.pdf).
21. JOHN K. VANDYK, T. T. *Pro Drupal 7 Development*. 2010, Disponible en:  
[http://sunshine.prod.uci.cu/gridfs/sunshine/books/Pro\\_Drupal\\_7\\_Development\\_Third\\_Edition.pdf](http://sunshine.prod.uci.cu/gridfs/sunshine/books/Pro_Drupal_7_Development_Third_Edition.pdf).
22. JOOMLA.CL. *Características de Joomla* Disponible en: <http://www.joomla.cl>.
23. KABIR, M. J. *La Biblia Server Apache*. 2004, Disponible en:  
[http://sunshine.prod.uci.cu/gridfs/sunshine/books/La\\_Biblia\\_Server\\_Apache.pdf](http://sunshine.prod.uci.cu/gridfs/sunshine/books/La_Biblia_Server_Apache.pdf).
24. LAPUENTE, M. J. L. *Lenguajes hipertextuales*. 2011, Disponible en:  
[http://www.hipertexto.info/documentos/lenguajes\\_h.htm](http://www.hipertexto.info/documentos/lenguajes_h.htm).
25. LEMES, J. J. O. *Catálogo de archivos contenedores de datos provenientes de dispositivos GPS*. 2008.
26. MARTÍN, L. M. *POO y patrones de diseño en DRUPAL*. 2010, Disponible en:  
<http://jitcode.blogspot.com/2010/06/poo-y-patrones-de-diseno-en-drupal.html>.
27. MARTINEZ, R. *Sobre PostgreSQL* Disponible en:  
[http://www.postgresql.org.es/sobre\\_postgresql](http://www.postgresql.org.es/sobre_postgresql).
28. MUSTELIER, S. C. *Propuesta de Procedimiento y Herramienta para el desarrollo de Pruebas de Rendimiento de Carga y Estrés en el Grupo de Calidad de FORTES*. 2011.
29. NETBEANS.ORG. *NetBeans IDE 7.1* Disponible en:  
<http://netbeans.org/community/releases/71/>.
30. ORTEGA, I. S. *Curso de php 5*. 2005, Disponible en:  
[http://sunshine.prod.uci.cu/gridfs/sunshine/books/Curso\\_de\\_php5.pdf](http://sunshine.prod.uci.cu/gridfs/sunshine/books/Curso_de_php5.pdf).
31. PADILLA, R. M. *Rendimiento entre Sistemas Gestores de Contenido (CMS)* Disponible en:  
<http://www.maestrosdelweb.com/category/editorial/desarrollo-web>.
32. PÉREZ, J. E. *CSS avanzado*. 2009a, Disponible en:  
[http://sunshine.prod.uci.cu/gridfs/sunshine/books/css\\_avanzado.pdf](http://sunshine.prod.uci.cu/gridfs/sunshine/books/css_avanzado.pdf).
33. PÉREZ, J. E. *Introducción a JavaScript*. 2009b, Disponible en:  
[http://sunshine.prod.uci.cu/gridfs/sunshine/books/introduccion\\_javascript.pdf](http://sunshine.prod.uci.cu/gridfs/sunshine/books/introduccion_javascript.pdf).
34. PÉREZ, J. E. *Introducción a CSS*. 2008, nº Disponible en:  
[http://sunshine.prod.uci.cu/gridfs/sunshine/books/introduccion\\_css.pdf](http://sunshine.prod.uci.cu/gridfs/sunshine/books/introduccion_css.pdf).
35. PÉREZ, M. T. *Sistema para el Control de Asistencia integrado al CMS Drupal*. 2010.

36. PHP.NET. *PHP Manual* Disponible en: <http://www.php.net/manual/en/index.php>.
37. POSTGRESQL.ORG. *postgresql* Disponible en: <http://www.postgresql.org/about/news/1108/>.
38. PUBLISHERS, B. *El surgimiento de la sociedad de redes* Disponible en:  
<http://www.comunicacionymedios.com/cuerpo.htm>.
39. RODRÍGUEZ, A. E. G. *Catálogo automatizado de métricas de calidad para evaluar los productos en las pruebas*. 2011.
40. RODRÍGUEZ, I. B. *Agente Recuperador de Información en Tiempo Real para el Sistema Gestor de Contenidos Drupal*. 2010.
41. S.PRESSMAN, R. *Pressman*. 2005, Disponible en:  
<http://sunshine.prod.uci.cu/gridfs/sunshine/books/Pressman.pdf>.
42. SHREVES, R. *Drupal 7 Themes*. 2011, Disponible en:  
[http://sunshine.prod.uci.cu/gridfs/sunshine/books/Drupal\\_7\\_Themes.pdf](http://sunshine.prod.uci.cu/gridfs/sunshine/books/Drupal_7_Themes.pdf).
43. SOFTONIC.COM. *Softonic* Disponible en: <http://softonic.com>.
44. VERGARA, K. *Concepto y tipos de software*. 2007.
45. VISUAL-PARADIGM.COM. *Visual paradigm* Disponible en: [www.visual-paradigm.com](http://www.visual-paradigm.com).
46. ZELDMAN, J. *Diseño con estándares Web*. 2004, Disponible en:  
[http://sunshine.prod.uci.cu/gridfs/sunshine/books/Diseno\\_con\\_estandares\\_web.PDF](http://sunshine.prod.uci.cu/gridfs/sunshine/books/Diseno_con_estandares_web.PDF).



## Glosario de términos

**Aplicaciones web:** son aquellas aplicaciones que los usuarios pueden utilizar accediendo a un servidor web a través de Internet o de una intranet mediante un navegador. Es una aplicación *software* que se codifica en un lenguaje soportado por los navegadores web en la que se confía la ejecución al navegador.

**Artefactos:** cualquier elemento que resulte del proceso de desarrollo de *software*; por ejemplo: documentos de requisitos, especificaciones, diseños de interfaces, caso de pruebas, etc.

**CENIA:** (Centro de Informatización Universitaria) está encaminado para la informatización y automatización de procesos llevados a cabo en la universidad.

**Código abierto:** es una tendencia internacional del desarrollo de *software* que profesa la distribución del código junto a las aplicaciones, se rigen por licencias tales como GNU/GPL.

**GPL:** es una licencia pública general de GNU o más conocida por su nombre en inglés *General Public License*. Está orientada principalmente a proteger la libre distribución, modificación y uso de *software*. Su propósito es declarar que el *software* cubierto por esta licencia es *software* libre y protegerlo de intentos de apropiación que restrinjan esas libertades a los usuarios.

**Hook:** del inglés gancho, palabra usada en la comunidad de Drupal para nombrar los métodos de la interfaz que deben implementar los módulos desarrollados para este CMS.

**IDE:** *Integrated Development Enviroment* (Ambiente Integrado de Desarrollo).

**Servidor:** Es una computadora que, formando parte de una red, provee servicios a otras computadoras llamadas clientes.

**TIC:** se definen como un conjunto de tecnologías que permiten el acceso a la información y la comunicación de una forma rápida y eficaz, involucran Internet, Correo Electrónico, Televisión Interactiva, Aprendizaje a través de medios electrónicos, entre otros.

**Validación:** comprobación de que se está construyendo el producto correcto.

## Anexos

### Anexo 1: Casos de pruebas

#### Caso de prueba: Votar.

**Descripción:** para acceder a esta funcionalidad el usuario va al *software* que desea votar. Luego de seleccionar el *software* el usuario puede votar dando una puntuación de 1 a 5 pinchando sobre la estrella.

#### Condiciones de ejecución:

El usuario debe estar autenticado en el sistema.

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Votar.	El usuario le da una puntuación al <i>software</i> según su opinión.	El sistema actualiza la información de los votos y muestra el promedio.	<ol style="list-style-type: none"> <li>1. El usuario accede a la aplicación.</li> <li>2. Selecciona el <i>software</i> que dese votar.</li> <li>3. Selecciona la cantidad de estrellas según su voto de 1 a 5.</li> <li>4. El sistema agrega la votación y realiza los cálculos para actualizar el promedio.</li> </ol>

**Tabla 22:** Descripción caso de prueba Votar.

#### Caso de prueba: Autenticar usuario.

**Descripción:** para acceder a esta funcionalidad el usuario va al formulario de autenticación que se encuentra en la parte superior del sistema. Finalmente el sistema valida los datos introducidos por el usuario y el rol que cumple dentro del sistema para asignar permisos.

Escenario	Descripción	Usuario	Contraseña	Sede	Respuesta del sistema	Flujo central
EC 1.1 Autenticar usuario de forma correcta.	Autenticar al usuario para que pueda acceder a la aplicación.	V (el usuario de la persona que se quiere)	V (la contraseña de la persona que se quiere autenticar).	V (se selecciona la sede a la cual pertenece el usuario)	El sistema verifica los datos y accede a la aplicación con los	<ol style="list-style-type: none"> <li>1. El usuario accede a la aplicación.</li> <li>2. Introduce los datos.</li> <li>3. El sistema verifica</li> </ol>

		autenticar ).			permisos que posee.	los datos. 4. Accede a la aplicación.
EC 1.2 Autenticar usuario dejando campos vacíos.	Autenticar al usuario para que pueda acceder a la aplicación.	V (migonzal ez).	V (vacío).	V (sede central).	El sistema verifica los datos y muestra el mensaje “el campo contraseña es obligatorio”.	1. El usuario accede a la aplicación. 2. Introduce los datos dejando campos vacíos. 3. El sistema verifica los datos y muestra el mensaje “el campo contraseña es obligatorio”.
EC 1.3 Autenticar usuario con informa- ción incorrecta.	El usuario introduce los datos per el sistema muestra un mensaje de error.	V (migonzal ez).	V (contraseña).	V(sede central).	El sistema verifica los datos y muestra el mensaje “el usuario migonzalez o la contraseña son incorrectos”.	1. El usuario accede a la aplicación. 2. Introduce los datos de forma incorrecta. 3. El sistema verifica los datos y muestra el mensaje “el usuario migonzalez o la contraseña son incorrectos”.

Tabla 22: Descripción caso de prueba Votar.

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
----	-----------------	---------------	------------	-------------

1.	usuario	Campo de texto	No	El usuario de la persona que se quiere registrar en el sistema.
2.	contraseña	Campo de texto	No	La contraseña del usuario.
3.	sede	Lista de selección	No	La sede a la cual pertenece.

**Tabla 23:** Descripción campos caso de prueba Autenticar usuario.

### Caso de prueba: Eliminar propuesta.

**Descripción:** para acceder a esta funcionalidad el usuario va al menú Propuestas y selecciona las propuestas que desea eliminar, luego debe confirmar mediante la opción “Eliminar seleccionadas” y el botón “confirmar”.

### Condiciones de ejecución:

El usuario debe estar autenticado con el rol de administrador.

Escenario	Descripción	Propuesta	Respuesta del sistema	Flujo central
EC 1.1 Eliminar propuestas.	El usuario selecciona las propuestas a eliminar.	V (se selecciona la o las propuestas a eliminar).	El sistema muestra un mensaje “Propuesta eliminar (y la cantidad de elementos eliminados)”.	<ol style="list-style-type: none"> <li>1. Acceder a la aplicación.</li> <li>2. Selecciona la opción propuesta.</li> <li>3. Selecciona la o las propuestas a eliminar.</li> <li>4. Confirma marcando la opción “Confirmar”.</li> <li>5. El sistema muestra un mensaje “Propuesta eliminar (y la cantidad de elementos eliminados)”.</li> </ol>
EC 1.2 Eliminar	El usuario selecciona las	V (vacíos).	El sistema muestra el	<ol style="list-style-type: none"> <li>1. Acceder a la aplicación.</li> <li>2. Selecciona la opción menú</li> </ol>

propuestas dejando campos vacíos.	propuestas a eliminar.		mensaje "Por favor seleccione al menos un elemento".	portada. 3. No selecciona ninguna propuesta. 4. El sistema muestra el mensaje "Por favor seleccione al menos un elemento".
-----------------------------------	------------------------	--	--	--

Tabla 24: Descripción caso de prueba Eliminar propuesta.

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	Propuestas	Lista de selección	No	Contiene todas las propuestas que los usuarios han subido.

Tabla 25: Descripción campos caso de prueba Eliminar propuesta.

## Anexo 2: Interfaces del sistema



Figura 5: Compartir software.

Contenido Estructura Apariencia Personas Módulos Configuración Informes Bienvenido, **laborrero** Cerrar sesión

Agregar contenido Editar atajos

UCI Universidad de las Ciencias Informáticas

Nota informativa de la Defensa Civil sobre intensas lluvias  
Intranet Aviso 1 de 6

laborrero Cerrar

# UCISoftware

Inicio Proponer Propuestas

Buscar

## Portada

 <b>Administración</b>	 <b>Educación y ciencia</b>	 <b>Imágen</b>	 <b>Internet</b>
 <b>Personaliza tu pc</b>	 <b>Programación</b>	 <b>Redes</b>	 <b>Seguridad</b>
 <b>Servidores</b>	 <b>Sistemas operativos</b>	 <b>Sonido</b>	 <b>Utilidades</b>
 <b>Video</b>			

### Visitantes

Total de visitas: 4277  
 Usuarios registrados: 16  
 Último usuario en registrarse: jmochoa  
 Software publicados: 8  
 Conectado desde el IP: 10.53.7.234

### Usuarios nuevos

jmochoa  
 jaeger  
 Isimon  
 ymatias  
 migonzalez

### Quién está conectado

Hay actualmente 1 usuario conectado.  
 laborrero

### Más descargados

Mozilla Firefox (148)  
 Leech Get (83)  
 Internet Explorer (30)  
 Flash Get (14)  
 Pidgin (2)  
 Psi (2)  
 Pandion (1)

UCISoftware

© 2012 CENIA | Terminos  
 Universidad de las Ciencias Informáticas

Figura 6: Portada.


# Mozilla Firefox

Enviado por admin en Mié, 04/04/2012 - 09:12

**Categoría:** Navegadores Web



Es un navegador web rápido, seguro y fácil de usar. Ofrece multitud de ventajas, destacando la capacidad de instalar extensiones para personalizar su uso.

Adjunto	Tamaño
 Mozilla Firefox versión 4.0	22.32 KB
 Mozilla Firefox versión 10.4	22.32 KB
 Mozilla Firefox versión 11.1	22.32 KB

Voto: ★★★★★

Average: 3.3 (4 votes)

Compartir

## Añadir nuevo comentario

**Su nombre**

jmochoa

**Asunto**

**Comentario \***

Guardar

Vista previa

Figura 7: Software seleccionado.

# Propuestas

▼ Operaciones

Eliminar seleccionados

Aprobar seleccionadas

<input type="checkbox"/>	Nombre	Categoría	Enlace	Enviada por	Fecha de la propuesta
<input type="checkbox"/>	Psi	Mensajería instantánea	vista	laborrero	06/10/2012 - 17:53
<input type="checkbox"/>	Pandion	Mensajería instantánea	vista	laborrero	06/10/2012 - 17:50

**Figura 8:** Propuestas.