



# Universidad de las Ciencias Informáticas

---

## Facultad 1

**Título:** “Sistema para la gestión de las imágenes que se utilizan en la publicación de noticias en el departamento de Comunicación Audiovisual de la Universidad de las Ciencias Informáticas”

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

**Autor:** Addiel Alejandro Téllez Escalona

**Tutores:** MSc. William Santana Méndez

Ing. Damarys Cano López

**Co-tutor:** Ing. Jorge Jesús Pérez



---

**DECLARACIÓN DE AUTORÍA**

Declaro por este medio ser autor del presente trabajo de diploma y autorizo a la Universidad de las Ciencias Informáticas a hacer uso de la misma en su beneficio así como los derechos patrimoniales de la misma con carácter exclusivo. Para que así conste firmo la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

---

**Firma del autor****Addiel Alejandro Téllez  
Escalona**

---

**Firma del Tutor****MSc. William Santana Méndez**

---

**Firma del Tutor****Ing. Damarys Cano López**



---

## *DATOS DE CONTACTO*

**MSc. William Santana Méndez:** graduado de Ingeniero Informático en el Instituto Superior Politécnico José Antonio Echeverría, también conocido como La Ciudad Universitaria José Antonio Echeverría (CUJAE). Categoría docente: Profesor asistente, categoría científica: Máster, años de experiencia en el tema: 5, años de graduado: 6. Dirección de correo: [wsantana@uci.cu](mailto:wsantana@uci.cu)

**Ing. Damarys Cano López:** graduada de Ingeniero en Ciencias Informáticas en la Universidad de las Ciencias Informáticas, 2 años de experiencia laboral. Actualmente se desempeña como líder del proyecto Redacta. Pertenece al departamento Universidad Digital del Centro de Informatización Universitaria, CENIA. Dirección de correo: [dcano@uci.cu](mailto:dcano@uci.cu).

---

## DEDICATORIA

*A mi madre, por traerme al mundo, darme el amor y la educación necesaria para vivir.*

*A mi padre por darme su confianza, su fuerza y ser el guía de mi vida.*

*A todos mis familiares por su apoyo incondicional en todo momento.*

*A todos mis amigos, pues sin ellos no hubiera podido cumplir este sueño.*

---

## AGRADECIMIENTOS

*Agradezco:*

*A mi papá por ser mi guía y mi ejemplo a seguir a lo largo de mi vida.*

*A mi mamá por apoyarme en cada momento y estar siempre pendiente de mí durante los 5 años.*

*A mi hermana por su apoyo incondicional.*

*A mi BB Hanny Valdés Hernández por hacerme el hombre más feliz del mundo y estar a mi lado cuando más lo necesité. “GRACIAS MI VIDA”.*

*A mis abuelos por estar siempre pendientes de mi “TEXIS” y por sus oraciones incasables.*

*A mis tutores: William y Damarys por su ayuda, guía y preocupación.*

*A mi co-tutor Jorge por su gran ayuda y preocupación.*

*A los compañeros de clase del grupo 7 que están conmigo desde primer año.*

*A mis tíos por estar siempre pendientes de mí.*

*A los compañeros del proyecto por compartir este año juntos.*

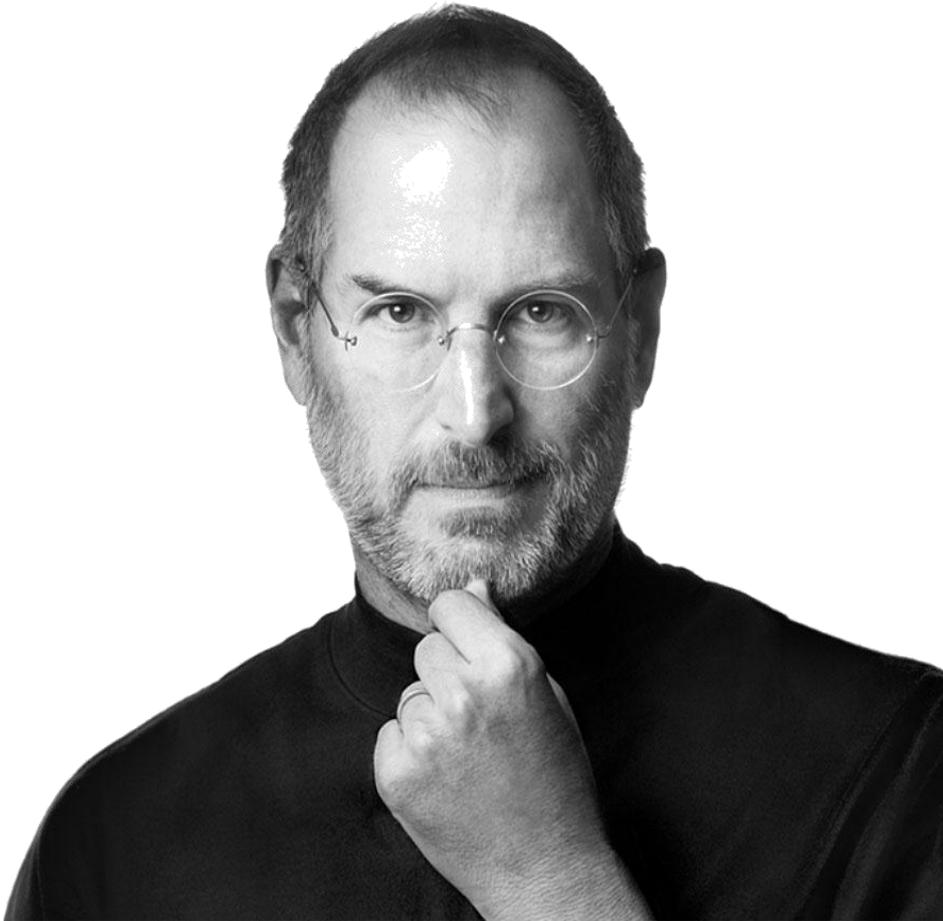
*A la Revolución y en especial a la UCI por formarme como profesional.*

*A la FEU por ser parte de mi vida universitaria y permitirme ser partícipe del trabajo de la mejor facultad.*

*A Dios por estar siempre a la escucha de su hijo en todo momento.*

*A todas las personas que de una forma u otra colaboraron para que mis sueños se hicieran realidad.*

GRACIAS



*"No tenemos la oportunidad de hacer muchas cosas, pero todo el mundo debe ser excelente porque se trata de nuestra vida"*

*Steve Jobs*

## RESUMEN

Actualmente la gestión de imágenes digitales ocupa un espacio cada vez mayor en el mundo de la informática. Un usuario haciendo uso de programas informáticos puede reducir en gran medida el tiempo de búsqueda y localización de una imagen. Actualmente en la Universidad de las Ciencias Informáticas (UCI) existe el departamento de Comunicación Audiovisual que demanda la continua interacción con las mismas. Con el presente trabajo, se creó una aplicación web a través del CMS Drupal, la cual permite gestionar dichas imágenes previamente identificadas con metadatos, brindando una mejor accesibilidad e interacción. Se analizó el estado del arte de los sistemas que permiten la gestión de imágenes a nivel nacional e internacional. Se estudiaron y especificaron las herramientas y tecnologías que se utilizaron en el desarrollo del sistema, se siguieron los pasos que brinda el proceso de desarrollo ágil orientado al nivel 2 de CMMI en el análisis, diseño, implementación y pruebas al sistema. Se exponen además, los elementos fundamentales de la arquitectura, patrones de diseño y estándares de código. La propuesta de solución viabiliza la creación de espacios para almacenar y obtener imágenes que por consiguiente contribuye al intercambio de información.

**Palabras clave:** aplicación web, drupal, imagen digital.



---

*ABSTRACT*

Currently digital image management occupies a growing space in the computer world. A user using computer software can greatly reduce the search time and location of an image. Currently at the University of Informatics Sciences (UCI) exists Communication Studies department that demands continuous interaction with them. In the present work, we created a web application through the Drupal CMS, which allows you to manage the images previously identified with metadata, providing better accessibility and interaction. We analyzed the state of the art systems that allow image management nationally and internationally. We studied and specified the tools and technologies that were used in developing the system, we followed the steps provided by the agile development process oriented CMMI level 2 in the analysis, design, implementation and testing the system. Also discussed are fundamental elements of architecture, design patterns and code standards. The proposed solution makes possible the creation of spaces to store and retrieve images which consequently contributes to the exchange of information.

**Keywords:** digital image, drupal, web application.

## ÍNDICE DE CONTENIDO

<b>Introducción .....</b>	<b>1</b>
<b>Capítulo #1: Fundamentación teórica.....</b>	<b>5</b>
1.1 Introducción .....	5
1.2 Conceptos asociados al objeto de estudio .....	5
1.3 Sistemas homólogos.....	7
1.4 Herramientas y tecnologías.....	9
1.5 Conclusiones parciales .....	16
<b>Capítulo #2. Descripción de la propuesta de solución .....</b>	<b>18</b>
2.1 Introducción .....	18
2.2 Situación problemática.....	18
2.3 Propuesta de solución.....	18
2.4 Modelo de dominio.....	19
2.5 Técnicas de obtención de requisitos .....	19
2.6 Especificación de los requerimientos de software .....	20
2.7 Modelado de datos.....	25
2.8 Patrones de diseño en Drupal.....	29
2.9 Arquitectura de Drupal .....	30
2.10 Conclusiones parciales .....	32
<b>Capítulo #3. Implementación y prueba .....</b>	<b>33</b>
3.1 Introducción .....	33
3.2 Estándares de codificación .....	33
3.3 Diagrama de componentes .....	37
3.4 Diagrama de despliegue .....	38
3.5 Prueba de software .....	39
3.6 Pruebas de rendimiento .....	51
3.7 Conclusiones parciales .....	52
<b>Conclusiones generales .....</b>	<b>53</b>
<b>Recomendaciones .....</b>	<b>54</b>
<b>Referencias bibliográficas.....</b>	<b>55</b>
<b>Bibliografía consultada.....</b>	<b>58</b>
<b>Glosario de términos .....</b>	<b>60</b>
<b>Anexos .....</b>	<b>62</b>
Anexo #1: Descripción de los requisitos funcionales.....	62
Anexo #2: Representación del registro por días. ....	69
Anexo #3: Representación del menú de contacto.....	70



---

Anexo #4: Representación del banco de imágenes. .... 71



---

**ÍDICE DE ILUSTRACIONES**

Ilustración 1: Modelo de dominio.....	19
Ilustración 2: Modelo físico de datos. ....	26
Ilustración 3: Arquitectura de Drupal.....	30
Ilustración 4: Diagrama de componentes. ....	37
Ilustración 5: Diagrama de despliegue. ....	38
Ilustración 6: Representación de prueba de Caja Negra. ....	39
Ilustración 7: Cantidad de no conformidades por iteraciones.....	50
Ilustración 8: Prueba con 15 usuarios. ....	52
Ilustración 9: Registro de acciones.....	69
Ilustración 10: Menú "Contáctenos".....	70
Ilustración 11: Ver banco de imágenes.....	71



## ÍNDICE DE TABLAS

Tabla 1: Clasificación de los metadatos.....	6
Tabla 2: Personas que intervienen en el sistema.....	18
Tabla 3: Requisitos funcionales. ....	20
Tabla 4: Descripción del requisito funcional crítico autenticar usuario. ....	21
Tabla 5: Descripción del requisito funcional crítico adicionar imagen.....	21
Tabla 6: Descripción del requisito funcional crítico adicionar galería. ....	22
Tabla 7: Usuario. ....	26
Tabla 8: Rol.....	27
Tabla 9: Usuario_Rol.....	27
Tabla 10: Permission.....	27
Tabla 11: Imagen.....	27
Tabla 12: Galería.....	28
Tabla 13: Comment.....	28
Tabla 14: Node.....	28
Tabla 15: Node_revision.....	29
Tabla 16: Caso de prueba "Autenticar usuario". ....	40
Tabla 17: Caso de prueba "Adicionar imagen".....	43
Tabla 18: Caso de prueba "Adicionar galería". ....	47
Tabla 19: Caso de Prueba "Eliminar Galería de imágenes".....	50
Tabla 20: Descripción del requisito funcional adicionar categorías a las imágenes. ....	62
Tabla 21: Descripción del requisito funcional editar imagen.....	62
Tabla 22: Descripción del requisito funcional editar galería de imágenes. ....	63
Tabla 23: Descripción del requisito funcional mostrar imagen. ....	64
Tabla 24: Descripción del requisito funcional mostrar galería de imágenes. ....	65
Tabla 25: Descripción del requisito funcional eliminar galería de imágenes.....	66
Tabla 26: Descripción del requisito funcional eliminar imagen. ....	67

## Introducción

Los medios de comunicación poseen gran importancia ya que juegan un papel fundamental en el desarrollo de la sociedad, la tecnología y la información. A través de su uso la sociedad puede mantenerse informada de acontecimientos que tienen lugar en las diferentes esferas de la sociedad, dígase cultura, deporte, economía, avances científicos-tecnológicos, las cuales pueden ser en el ámbito nacional e internacional. Dichos medios tienen como fin informar, investigar, analizar, trazar horizontes y desarrollar sus objetivos conscientes de la función social que desempeñan.

En el mundo la inmediatez de la publicación es una de las necesidades que provocó llevar a cabo la informatización de los medios de comunicación. El primer cambio efectuado en los centros de comunicación es la digitalización de la información. En la actualidad estos centros cuentan con aplicaciones web que permiten la publicación de noticias de interés tanto para una organización como para varios usuarios. Dichas publicaciones pueden ser identificadas por los diferentes campos que componen una noticia, como lo son el título, el cuerpo y la imagen digital que generalmente la identifica.

En Cuba uno de los centros más destacados en el desarrollo de sistemas y servicios informáticos, es la Universidad de las Ciencias Informáticas (UCI). Su misión es formar profesionales comprometidos con su Patria y altamente calificados en la rama de la Informática, producir aplicaciones y servicios informáticos, a partir de la vinculación estudio – trabajo como modelo de formación y servir de soporte a la industria cubana de la informática [1].

En la UCI existen diferentes medios de comunicación en los cuales se manejan disímiles tipos de información para la comunidad universitaria. Con el fin de gestionar dicha información la universidad cuenta con el departamento de Comunicación Audiovisual que tiene como una de sus principales tareas realizar la publicación de noticias para la prensa digital de la universidad. Para realizar estas publicaciones se hace necesaria la gestión (inserción, actualización y eliminación) de un cúmulo de imágenes digitales. Sin embargo, el departamento de Comunicación Audiovisual, presenta como principal contratiempo, que las imágenes digitales no se encuentran en un lugar centralizado, sino que se almacenan en distintos dispositivos de almacenamiento, lo que contribuye a que exista duplicidad, ya que una misma imagen puede poseer diferentes nombres, además permite que el proceso de búsqueda y localización sea engorroso y provoque pérdida de tiempo a las personas que las utilizan. Esto trae como consecuencia que al no existir una aplicación que permita adicionarle metadatos para realizar los filtros de búsqueda, que catalogue y agrupe las imágenes digitales, dichas imágenes no se encuentran disponibles para ser utilizadas en una posterior publicación.



A partir de la **problemática** presentada con anterioridad, se enuncia el siguiente **problema de investigación**: ¿Cómo mejorar la gestión de las imágenes que se utilizan para la publicación de noticias en el departamento de Comunicación Audiovisual de la Universidad de las Ciencias Informáticas?

El **objeto de estudio** de la presente investigación lo constituyen los sistemas de gestión de imágenes digitales y el **campo de acción** los sistemas de gestión de imágenes digitales para la prensa digital.

La investigación estará encaminada a cumplir el **objetivo general** de desarrollar en un Sistema Gestor de Contenidos un sistema web para la gestión de las imágenes que se utilizan en la publicación de noticias en el departamento de Comunicación Audiovisual de la Universidad de las Ciencias Informáticas. A su vez, dicho objetivo estará desglosado en los siguientes **objetivos específicos**:

- Construir el marco teórico de los antecedentes y estado de los sistemas que permiten la gestión de imágenes.
- Desarrollar un sistema web para la gestión de las imágenes que se utilizan en la publicación de noticias en el departamento de Comunicación Audiovisual de la Universidad de las Ciencias Informáticas.
- Seleccionar y poner en práctica un método para la evaluación técnica del sistema desarrollado.

Para guiar la investigación se plantea la siguiente **idea a defender**: con el desarrollo de un sistema de gestión de las imágenes que se utilizan para la publicación de noticias, se garantizaría que el departamento de Comunicación Audiovisual cuente con una herramienta que le permita gestionar de manera automatizada y eficiente dichas imágenes.

Como vía para alcanzar los objetivos planteados, se definieron las siguientes **tareas de investigación**:

1. Análisis comparativo de las herramientas y tecnologías que se utilizan en el desarrollo de sistemas web para la gestión de imágenes utilizados en Cuba y en el mundo.
2. Definición de los requerimientos del Sistema para la gestión de las imágenes que se utilizan en la publicación de noticias en el departamento de Comunicación Audiovisual de la Universidad de las Ciencias Informáticas.
3. Diseño del Sistema para la gestión de las imágenes que se utilizan en la publicación de noticias en el departamento de Comunicación Audiovisual de la Universidad de las Ciencias Informáticas.
4. Implementación del Sistema para la gestión de las imágenes que se utilizan en la publicación de noticias en el departamento de Comunicación Audiovisual de la Universidad de las Ciencias Informáticas.
5. Realización de pruebas al sistema para validar su correcto funcionamiento.



Para dar cumplimiento a estas tareas se utilizaron los siguientes **métodos científicos de investigación**:

**Métodos teóricos utilizados:**

- Histórico-lógico: permitió estudiar cómo han evolucionado los sistemas que posibilitan la gestión de imágenes en departamento de Comunicación Audiovisual.
- Analítico-sintético: este método permitió estudiar la información referente a los sistemas que permiten la gestión de imágenes, sus principales características, ventajas y desventajas.

**Métodos empíricos:**

- Entrevista: se realizaron entrevistas para definir metadatos en imágenes, precisar cómo funcionan en el departamento de Comunicación Audiovisual de la Universidad de las Ciencias Informáticas los procesos de publicación de imágenes e identificar las necesidades actuales.

**Justificación de la Investigación**

La presente investigación propone como aportes prácticos la elaboración de una herramienta que les permitirá a los administradores del sistema gestionar (insertar, actualizar y eliminar) las imágenes que se utilizan para la publicación de noticias en el departamento de Comunicación Audiovisual de la Universidad de las Ciencias Informáticas. El sistema permitirá adicionar metadatos a las imágenes almacenadas, posibilitando así el uso de filtros para la búsqueda de las mismas. Los usuarios del sistema contarán con un repositorio del que podrán obtener las imágenes que necesiten facilitando así la rápida localización de las mismas.

El documento de investigación se encuentra estructurado en tres capítulos, distribuidos de la siguiente forma:

**Capítulo #1. Fundamentación teórica:** en este capítulo se realiza una breve explicación del estudio realizado sobre los metadatos de las imágenes, además de los sistemas existentes que permiten la gestión de imágenes. Se describen las herramientas y tecnologías a utilizar en el desarrollo de la solución.

**Capítulo #2. Descripción de la propuesta de solución:** en este capítulo se describe la propuesta de solución. Se realiza el modelo de dominio. Se especifican los requisitos funcionales y no funcionales del sistema. Se define además el modelo físico, se describen las tablas de la base de datos, se seleccionan los patrones de diseño en Drupal y se describe la arquitectura.

**Capítulo #3. Implementación y prueba:** en este capítulo se aborda lo relacionado con la implementación del sistema, se seleccionan los estándares de codificación, se define el diagrama de componentes y el diagrama de despliegue. Se realizan las pruebas de caja negra y las pruebas de carga y estrés para validar el correcto funcionamiento del sistema.



---

El presente trabajo contiene además **Conclusiones, Recomendaciones, Referencias, Glosario de términos y Anexos.**

## Capítulo #1: Fundamentación teórica

### 1.1 Introducción

Con el desarrollo del presente capítulo se pretende alcanzar conocimientos sobre los metadatos de las imágenes digitales y los conceptos asociados a estos, que permitirán desempeñar la parte práctica con la calidad requerida. Se ofrece una panorámica general relacionada con la existencia de programas utilizados para el almacenamiento y gestión de imágenes digitales, con el objetivo de realizar un estudio que permita conocer las funcionalidades que sirvan de guía en la solución. Además, se describen las herramientas y tecnologías a utilizar para el desarrollo de la nueva propuesta.

### 1.2 Conceptos asociados al objeto de estudio

En el presente epígrafe se realiza una explicación de los conceptos fundamentales asociados al objeto de estudio y al campo de acción de la investigación, con el fin de obtener una mayor comprensión del tema, logrando una solución que cumpla con los objetivos trazados.

#### 1.2.1 Imagen Digital

Las imágenes digitales son fotos electrónicas tomadas de una escena o escaneadas de documentos, fotografías, manuscritos, textos impresos e ilustraciones. Se realiza una muestra de la imagen digital y se confecciona un mapa de ella en forma de cuadrícula de puntos o elementos de la figura (píxeles). A cada píxel se le asigna un valor tonal (negro, blanco, matices de gris o color), el cual está representado en un código binario (ceros y unos). Los dígitos binarios (bits) para cada píxel son almacenados por una computadora en una secuencia y con frecuencia se los reduce a una representación matemática (comprimida). Luego la computadora interpreta y lee los bits para producir una versión analógica para su visualización o impresión [2].

Abordado este concepto se puede definir como imagen digital a toda imagen que sea generada por un ordenador o creada a través de un medio electrónico, como una cámara o escáner.

#### 1.2.2 Gestión de imágenes en la prensa digital

La gestión de imágenes y fotografías es otra de las áreas privilegiadas del software libre. Ya lo era antes del nacimiento de la fotografía digital, pero esta última ha incrementado aún más la cantidad de programas que ofrecen excelentes prestaciones a costo cero. Grandes empresas de software, como Adobe y Corel han ofrecido varias versiones gratuitas de sus programas de gestión de imágenes, con prestaciones recortadas.

A continuación se describen las funciones que debe presentar un sistema gestor de imágenes:

- Organizar y facilitar el acceso a colecciones de imágenes mediante un sistema de categorización y búsqueda.



- Proporcionar sistemas de visualización: vistas en grupo, individuales, creación de presentaciones, vista de metadatos, entre otros.
- Proporcionar opciones básicas de edición: color, dimensión, transformaciones, rotaciones, filtros, ajustes, entre otros.
- Facilitar operaciones de mantenimiento: cambio y exportación de formatos, creación de miniaturas (para páginas web), creación de hojas de contacto, realización de cambios globales, entre otros [3].

### 1.2.3 Metadatos

La Organización Nacional de Estándares de Información define metadatos como información estructurada que describe, explica, localiza, o de otra manera, hace que sea más fácil de recuperar, utilizar o manejar una fuente de información. La organización destaca que los metadatos son “la clave para asegurar de que los recursos sobrevivan y continúen siendo accesibles en el futuro” [4].

### 1.2.4 Clasificación de los metadatos

Para facilitar la comprensión global de los metadatos existentes es necesario clasificarlos. La clasificación sugerida se realiza mediante grupos o categorías de acuerdo a los propósitos generales de cada marco de metadatos [5].

**Tabla 1:** Clasificación de los metadatos.

Tipo	Definición	Ejemplos
Administrativos	Metadatos utilizados para el manejo y administración de los recursos de información.	<ul style="list-style-type: none"> <li>• Registros sobre la adquisición.</li> <li>• Requerimientos de acceso legal.</li> <li>• Información de los recursos a administrar.</li> <li>• Control de versión.</li> <li>• Seguimiento de los usuarios.</li> <li>• Reutilización de contenidos.</li> <li>• Condición física de un recurso.</li> </ul>
Descriptivos y Descubrimiento	Metadatos utilizados para describir, descubrir o identificar recursos de información.	<ul style="list-style-type: none"> <li>• Metadatos de apoyo para la recuperación de la información.</li> <li>• Índices especializados.</li> <li>• Información sobre el esquema de metadatos utilizado.</li> <li>• Taxonomías.</li> </ul>
Técnicos, modelos	Metadatos relacionados con el funcionamiento de un sistema o de cómo se interrelacionan sus componentes.	<ul style="list-style-type: none"> <li>• Metadatos que describen el funcionamiento de hardware y software.</li> <li>• Estructura de formatos digitales.</li> <li>• Modelos de intercambio de metadatos.</li> </ul>



### 1.2.5 Beneficios de los metadatos

Los beneficios de utilizar metadatos son diversos y dependen del área en que se utilicen. En términos generales los metadatos:

- Adhieren contenido, contexto y estructura a los objetos de información, asistiendo de esta forma al proceso de recuperación de conocimiento desde colecciones de objetos.
- Permiten generar distintos puntos de vista conceptuales para sus usuarios o sistemas, y liberan a estos últimos de tener conocimientos avanzados sobre la existencia o características del objeto que describen. Estos puntos de vista conceptuales pueden depender del sistema o usuario que los utiliza.
- Permiten el intercambio de la información sin la necesidad de involucrar el intercambio de los mismos recursos. Esta particularidad facilita entre otras cosas las búsquedas sobre colecciones distribuidas. Además los metadatos permiten una descripción precisa y discreta de los recursos permitiendo la creación de colecciones virtuales de descripciones donde agrupan los objetos de información para satisfacer requerimientos específicos.
- En cada proceso productivo, o en cada etapa del ciclo de vida de un objeto de información, se van generando metadatos para describirlos, y metadatos para describir dichos metadatos (manual o automáticamente). Estos generan valor agregado al sistema de metadatos y objetos que describen, generando datos desde los cuales es posible extraer conocimiento relevante al sistema mismo y sus procesos.
- Permiten un acceso a los recursos en forma controlada ya que se conoce con precisión el objeto descrito. Es posible entonces establecer sistemas de filtrado y permiten generar bases para una autenticación y mecanismos para definir grados de confianza sobre las fuentes de información.
- Permiten preservar los objetos de información permitiendo migrar (gracias a la información estructural) sucesivamente los mismos para el posible uso por parte de las futuras generaciones. La información semántica de los objetos se mantiene, disminuyendo así la pérdida del conocimiento.
- Los metadatos son esenciales para sostener un crecimiento de una Web a mayor escala, permitiendo búsquedas e integración del conocimiento desde un mayor número de fuentes heterogéneas [5].

### 1.3 Sistemas homólogos

En la búsqueda de respuestas a las necesidades que presenta el departamento de Comunicación Audiovisual de la Universidad de las Ciencias Informáticas, se estudiaron diferentes sistemas para la

gestión de imágenes digitales. A través de este análisis se pretende conocer su funcionamiento y las características que pudieran servir de aporte para el desarrollo de la aplicación.

A continuación se ofrece una panorámica general de los sistemas que permiten el almacenamiento y obtención de imágenes digitales.

### 1.3.1 Ámbito internacional

**Flickr:** es un sitio web que permite almacenar, ordenar, buscar y compartir fotografías en línea. Ofrece la posibilidad de crear álbumes de fotos. Su tarea consiste en recopilar imágenes de pinturas de autores organizándolas como una galería virtual de pinturas [6].

**Picasa:** es un sistema que permite organizar, editar y subir fotos de forma rápida y sencilla a través de la Web. Organiza las fotos en carpetas y álbumes. Se puede modificar el nombre, la fecha en la que se produjo, lugar donde se tomaron las fotos y activar y desactivar distintas funciones. Existen versiones del mismo para la mayoría de los sistemas operativos existentes [7].

**Dr. Upload:** es un sistema de almacenamiento de imágenes digitales gratuito. Permite almacenar imágenes de hasta 10 MB, los formatos que soporta son JPG<sup>1</sup>, GIF<sup>2</sup> y PNG<sup>3</sup>. Su uso es sencillo y simple, solo se debe seleccionar el archivo, subirlo y luego el sistema genera una serie de enlaces para compartirla [8].

**Arkiva:** es un servicio de almacenamiento en línea gratuito que ofrece 1 Gb para almacenar y compartir cualquier tipo de fichero digital. Permite la edición en línea de documentos e impresión de fotos [9].

### 1.3.2 Ámbito Nacional

**Fototeca:** módulo desarrollado en el año 2011 en la UCI utilizando el gestor de contenido empresarial Alfresco. Dicho módulo permite organizar la gestión de los recursos fotográficos en los medios de prensa, además agrupa toda la estructura de carpetas, grupos de usuarios, roles, permisos y tipos de contenidos [10].

### 1.3.3 Marco Teórico-Conceptual de la Investigación

Después de realizar un estudio de los sistemas que permiten la gestión de imágenes digitales en el ámbito nacional e internacional, se pudo apreciar que a pesar de las posibilidades que brindan en cuanto a su trabajo con las imágenes digitales, en su generalidad fueron diseñados según sus propias particularidades.

---

<sup>1</sup> JPG: Joint Photographic Experts Group (Grupo Conjunto de Expertos en Fotografía)

<sup>2</sup> GIF: Graphics Interchange Format (Formato de Compresión de Imagen)

<sup>3</sup> PNG: Portable Network Graphics (Gráficos de Red Portátiles)



En el caso de los sistemas Flickr, Picasa, Dr. Upload y Arkiva, se puede señalar que no cuentan con funcionalidades que permitan efectuar la captura e inserción de los metadatos reales que contienen las imágenes digitales. Otro como el módulo Fototeca que su gestión de imágenes se asemeja a la de la nueva propuesta, presenta problemas de portabilidad, pues para su uso necesita tener instalado el sistema de gestión de contenidos empresariales Alfresco con toda la serie de requerimientos que su instalación conlleva. Dicho sistema también presenta como uno de sus mayores inconvenientes que su interfaz de usuario no es de fácil usabilidad y requiere de un conocimiento medio y avanzado del usuario final. Además tampoco permite la agrupación de las imágenes por galerías, característica necesaria para el cliente, ya que le permite agrupar las imágenes por los eventos realizados en la universidad. Concluyendo que, ninguno de los sistemas analizados presentan las características ideales para ser utilizados como guía en la solución. Por lo que se decide realizar la implementación de un nuevo producto que responda a las necesidades del departamento de Comunicación Audiovisual de la Universidad de las Ciencias Informáticas.

## **1.4 Herramientas y tecnologías**

Definir las herramientas, el lenguaje y la metodología a seguir para la creación de un sistema informático, es el primer paso que se realiza y sin duda el más importante para llevar a cabo la implementación de un sistema de forma coherente y sistemática.

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

El proceso de desarrollo de software es aquel que guía cómo realizar un software correctamente y tiene una implicación total en sus resultados, debido a que influyen en la calidad del mismo. Estos procesos proponen un grupo de actividades a realizar y éstas a su vez generan un conjunto de artefactos que tributan a la correcta realización del producto.

La Universidad de las Ciencias Informáticas llevó a cabo a partir del 2008 un proceso de mejora encaminado a alcanzar el nivel 2 del modelo CMMI<sup>4</sup> mediante la gestión ágil de proyectos, adoptando como guía en el desarrollo del software la integración de las metodologías ágiles Scrum y Programación Extrema (XP). Dichas metodologías tienen como objetivo dar garantías a las cuatro demandas principales de la industria en la que se ha generado: valor, reducción del tiempo de desarrollo, agilidad y fiabilidad; y CMMI, modelo de referencia para el crecimiento de capacidades y madurez, que se enfoca tanto en procesos de Administración como de Ingeniería de Sistemas y Software.

El nivel 2 de CMMI posee 7 áreas de procesos, las cuales son:

- Planificación de proyectos
- Seguimiento y control del proyecto

---

<sup>4</sup> CMMI: Capability Maturity Model Integration (Integración de Modelos de Madurez de Capacidades)



- Gestión de acuerdos con proveedores
- Medición y análisis
- Aseguramiento de la calidad del producto y el proceso
- Gestión de la configuración
- Administración de requisitos [11].

El proceso de mejora tiene dos representaciones, la continua y la escalonada. Dichas representaciones son equivalentes, y cada empresa puede optar por la que se adapte a sus características y prioridades de mejora.

- La representación continua permite que una organización seleccione un área específica para hacerle una mejora. Utiliza niveles de capacidad para caracterizar una mejora relativa a un área de proceso individual.
- La representación por niveles o escalonada usa un conjunto predefinido de áreas de procesos para definir un camino para la mejora de una organización [11].

La representación que se utiliza en la UCI es la escalonada, la cual presenta cinco niveles de madurez que permiten ir comprobando el nivel correcto de desarrollo en que se encuentra la organización, los cuales son: el nivel inicial, el administrado, el definido, el administrado cuantitativamente y el optimizado.

Para determinar la correspondencia entre la aplicación de CMMI y las metodologías ágiles XP y Scrum se realizó un análisis donde se evaluó la posibilidad de la implementación de las buenas prácticas de cada una de estas metodologías, permitiendo generar varios artefactos de los proyectos de desarrollo con enfoque ágil. Dichas prácticas presentes en el área de proceso de Administración de Requisitos del nivel 2 de CMMI son:

- Obtener el entendimiento de los requisitos.
- Obtener el compromiso a los requisitos.
- Administrar los cambios a los requisitos.
- Mantener la trazabilidad bidireccional de los requisitos.
- Identificar inconsistencias entre el trabajo del proyecto y los requisitos.

#### **1.4.2 Lenguaje de modelado**

En el proceso de desarrollo de un sistema se debe tener en cuenta el modelo visual de la aplicación. En la presente investigación se utilizará el Lenguaje Unificado de Modelado (en inglés *Unified Modeling Language* (UML)) el cual prescribe un conjunto de notaciones y diagramas estándar para modelar sistemas orientados a objetos. Es capaz de describir la semántica general de los significados de los símbolos utilizados.



UML desarrolla un lenguaje gráfico para visualizar, especificar y documentar las variadas partes que incluye el desarrollo de software.

Entre las características que presenta UML se pueden mencionar:

- Lenguaje de modelado orientación a objetos.
- Viabilidad en la corrección de errores [12].

### **1.4.3 Herramienta CASE de Modelado**

Las aplicaciones informáticas que facilitan el trabajo dentro del ciclo de desarrollo del software son conocidas como herramientas CASE (Ingeniería de Software Asistida por Computadora) y se emplean para aumentar la productividad del desarrollo del software, disminuyendo los tiempos de construcción y el costo de los mismos. La tecnología CASE supone la automatización del desarrollo del software, contribuyendo a mejorar la calidad en el desarrollo de sistemas de información.

Para el desarrollo del ciclo de software se utilizará como herramientas CASE Visual Paradigm por las características que se manifiestan a continuación.

#### **Visual Paradigm v 8.0**

Visual Paradigm para UML es una herramienta profesional que 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, mejores y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. La herramienta UML CASE también proporciona abundantes tutoriales, demostraciones interactivas y proyectos [13].

Presenta varias ventajas entre las que se destacan:

- Poderosa herramienta de generación de .pdf/.html a partir de diagramas UML.
- Sincronización entre el código fuente y el modelo en tiempo real.
- Soporte para toda la notación UML.

### **1.4.4 Sistema de gestión de contenidos (CMS)**

Un Sistema de Gestión de Contenidos (CMS, por sus siglas en inglés) permite la creación y administración de contenidos principalmente en páginas web. Los CMS presentan una interfaz que controla una o varias bases de datos donde se almacena el contenido del sitio a través de una interfaz, su diseño es centrado al usuario. Maneja de forma independiente el contenido y el diseño posibilitando que en cualquier momento se le pueda dar un nuevo diseño al sitio sin necesidad de darle formato al contenido nuevo.

Un CMS tiene dos funciones fundamentales:

- Facilitar la creación de contenidos.

Para el cumplimiento de esta función el CMS provee una serie de herramientas que posibilitan que publicar un contenido no se torne tan complejo y que exista una sola fuente para todos ellos.

- Facilitar la presentación de contenidos.

Esta función permite la publicación de contenidos en múltiples formatos partiendo de una sola fuente, añadiendo metadatos a los mismos para facilitar la navegación en varias facetas [13].

### **Ventajas de los CMS:**

- Disminución de costo y esfuerzo.
- Permiten tener una apariencia y navegación uniforme en todo el sitio.
- Actualizan y gestionan el contenido fácilmente.
- Permiten diseñar una plantilla personalizada.
- Aumento de la productividad del trabajo.

### **CMS a utilizar en el sistema**

Para el desarrollo del sistema se escogió el CMS Drupal por las características que se describen a continuación.

#### **Drupal v 7.18**

Es un sistema de administración de contenidos, dinámico, lo que posibilita que los contenidos textuales y las configuraciones sean almacenados en una base de datos. Es totalmente configurable, de forma tal que el administrador puede activar y desactivar diferentes características y establecer configuraciones que cambian el aspecto del sitio web. Es reconocido por los niveles de seguridad que alcanzan las aplicaciones web configuradas en Drupal, con alta velocidad de respuesta a los usuarios, posee independencia entre la persistencia de datos y la presentación del contenido. Cuenta con una amplia comunidad de desarrolladores tanto a nivel internacional como en la propia Universidad permitiendo así el fácil acceso y consulta a la documentación y utilidades referente al mismo. Está diseñado para ser fácilmente extensible a través de módulos, algunos de estos se encuentran en todas las instalaciones de Drupal, denominados comúnmente como módulos del núcleo, mientras que otras pueden ser instaladas y descargadas individualmente del sitio de Drupal, estos son los denominados módulos contribuidos. El aspecto básico de un sitio de Drupal puede cambiar usando diferentes temas, que al igual que los módulos pueden ser del núcleo y contribuidos [14].

### **Ventajas del CMS Drupal frente a otros CMS:**

- Calidad del código.



Los desarrolladores y la comunidad de Drupal cuentan con herramientas para valorar la estabilidad y la calidad del software, posibilitando que el software de Drupal sea fiable y estable.

➤ **Arquitectura potente, flexible y accesible.**

El sistema Drupal está concebido de base con una arquitectura que le otorga una gran flexibilidad: manejo análogo de diferentes tipos de contenido, categorización del contenido por taxonomías, código HTML generado de gran limpieza y accesibilidad.

➤ **Capacidad para extender las funcionalidades.**

Drupal cuenta con numerosos módulos que extienden las funcionalidades básicas del sistema. En esto puede ser comparable a otros CMS exitosos, la diferencia con Drupal es la capacidad de integración entre todos los módulos, para construir soluciones sólidas y escalables.

➤ **Comunidad y modelo de desarrollo.**

Cada vez más usuarios, desarrolladores y empresas eligen Drupal como su CMS, la comunidad de colaboradores que aportan su esfuerzo en mejorar la solución es enorme y muy bien organizada, con gran espíritu de pertenencia al grupo, y compromiso con el proyecto. Todo ello es importante porque garantiza que el sistema va a seguir creciendo y adaptándose a las nuevas necesidades que vayan apareciendo [15].

### **1.4.5 Lenguajes de programación**

Un lenguaje de programación es la notación para la descripción precisa de algoritmos o programas informáticos. Son el conjunto de instrucciones que permiten al programador pensar de forma clara sobre la complejidad del problema a resolver, de manera que pueda ordenarlas para la creación de un programa ejecutable por la computadora

Siendo consecuente con la elección del CMS Drupal, se restringe la utilización de los lenguajes que se tratan a continuación.

#### **Lenguajes de programación del lado del cliente**

Un lenguaje del lado cliente es independiente del navegador, lo que permite que la página pueda ser albergada en cualquier sitio, necesita tener instalados los *plugin* adecuados en la computadora cliente para obtener una mejor interfaz.

Existen varios lenguajes del lado cliente entre los que se encuentran HTML, CSS, Java Script, Applets de Java, Visual Basic Script, Flash. A continuación se describen y fundamentan los elegidos para el desarrollo del sistema por las características y ventajas que lo destacan de acuerdo al entorno en el que se está trabajando.



### **Lenguaje de Marcado de Hipertexto (HTML v 5.0)**

HTML es el lenguaje de Marcado de Hipertexto que se utiliza para crear los documentos a los que se accede a través de navegadores web. Está compuesto por etiquetas que definen la estructura y el formato que verá el usuario a través de la web. Permite que se creen enlaces entre distintas partes del mismo documento o entre distintas fuentes de información, a través de hiperenlaces o hipervínculos, e incluso insertar otros elementos como imágenes, sonidos y videos, brinda facilidades para publicar documentos en línea con encabezados, listas, o para diseñar formularios, incluir hojas de cálculo y otras aplicaciones directamente en su contenido [16].

### **Hojas de Estilo en Cascada (CSS v 3.0)**

Tecnología usada para el diseño web, permite dar formato a elementos HTML y XML, es de gran utilidad para el programador, posibilitándole darle formato y estilo a varias páginas web al mismo tiempo [15].

### **Lenguaje JavaScript v 1.5**

Es un lenguaje interpretado orientado a objetos, está diseñado para una fácil incrustación en otros productos y aplicaciones, tales como los navegadores web.

Posibilita la reducción de carga del servidor, se hace cargo de la mayoría de las funcionalidades del cliente de las cuales se encargaba el servidor. Puede interactuar con el contenido HTML, permitiendo a los programadores web utilizar contenido dinámico. Es *Open Source* (código abierto), por lo que cualquier persona puede usarlo sin comprar licencia [16].

### **Lenguajes de programación del lado del servidor**

Son aquellos lenguajes que son reconocidos, interpretados y ejecutados por el mismo servidor y enviados al cliente en un formato comprensible para él.

Siendo consecuente con la selección del CMS, se selecciona como lenguaje de programación el **Preprocesador de Hipertexto (PHP v 5.3.6)** por las siguientes características.

- Gratuito e independiente de la plataforma.
- Rápido, con una gran librería de funciones y mucha documentación.
- Desarrollado en política de código abierto.
- Compatible con todos los navegadores.
- Fácil de aprender.
- Capacidad de conexión con la mayoría de los manejadores de base de datos: MySQL, PostgreSQL, Oracle, MS SQL Server, entre otras.
- Capacidad de expandir su potencial utilizando módulos.



- Está diseñado específicamente para ser un lenguaje más seguro para escribir programas CGI<sup>5</sup> que Perl o C y con la selección correcta de opciones de configuración en tiempos de compilación y ejecución, siguiendo algunas prácticas correctas de programación.

#### 1.4.6 Servidor web

Para el desarrollo del sistema se utiliza el servidor web **apache v 2.2.20**, por la cantidad de beneficios que trae consigo, así como las características que lo hace el servidor más popular, a continuación se reflejan estas características y ventajas:

- Flexible, rápido y eficiente.
- Corre en una multitud de sistemas operativos.
- Continuamente actualizado y adaptado a los nuevos protocolos HTTP<sup>6</sup>.
- Multiplataforma.
- Modular, puede ser adaptado a diferentes entornos y necesidades, con los diferentes módulos de apoyo que proporciona, y con la API de programación de módulos, para el desarrollo de módulos específicos.
- Tecnología gratuita, de código fuente abierto.
- Permite personalizar la respuesta ante los posibles errores que se den en el servidor.
- Permite la creación de ficheros de archivos de registro, de este modo se puede tener un mayor control sobre lo que sucede en el servidor [17].

#### 1.4.7 Gestores de Base de Datos

Un Gestor de Base de Datos o Sistema de Gestión de Base de Datos (SGBD), es un software que permite introducir, organizar y recuperar la información de las bases de datos. Proporciona un control de la seguridad y privacidad de los datos, así como la mejora de su manipulación, facilita el proceso de diseño de aplicaciones, posibilita que los tratamientos sean más eficientes y rápidos, dando la mayor flexibilidad posible a los usuarios [18].

Sus objetivos fundamentales son:

- Independencia de los datos y los programas de aplicación.
- Minimización de la redundancia.
- Integración y sincronización de las bases de datos.
- Integridad de los datos.
- Seguridad y protección de los datos.
- Facilidad de manipulación de la información.
- Control centralizado.

---

5 CGI: Common Gateway Interface: tecnología que permite a un cliente (navegador web) solicitar datos de un programa ejecutado en un servidor web.

6 HTTP: HyperText Transfer Protocol: método utilizado para transferir ficheros de hipertexto por Internet.



Para el desarrollo del sistema se utilizará **PostgreSQL v 9.1**, ya que es uno de los gestores de bases de datos relacional orientado a objetos de software libre más utilizado en el mundo. Es capaz de manejar una enorme cantidad de datos, permitiendo el acceso simultáneo de un conjunto de usuarios, además brinda seguridad y estabilidad a los mismos y facilita el trabajo con procedimientos almacenados y consultas.

#### **1.4.8 Entorno de desarrollo integrado (IDE)**

El entorno de desarrollo es el banco de trabajo del programador, es el conjunto de herramientas y procesos de programación utilizados para crear el programa o producto de software.

Es aquel en el que los procesos y las herramientas se coordinan para ofrecer a los desarrolladores una interfaz ordenada y de cómoda visión del proceso de desarrollo, o al menos los procesos de la escritura de código, probándolo y empaquetándolo para su uso.

#### **Netbeans v 7.3**

Es una herramienta que permite que los desarrolladores puedan escribir, depurar, compilar y ejecutar programas. Está escrito en Java pero soporta otros lenguajes de programación como C, C++, Python, CSS, PHP, HTML y Ruby. Es libre y gratuito sin restricciones de uso, tiene una interfaz amigable e intuitiva, posee todas las herramientas para crear aplicaciones profesionales ya sean de escritorio, empresariales, web y móviles. Es multiplataforma y posee una creciente comunidad de usuarios, permite la depuración y ejecución de programas escritos. Posee integración con bases de datos, permitiendo escribir instrucciones de eliminar y actualizar. Sus funciones están previstas por módulos. Por las características que se describen se selecciona esta herramienta para el desarrollo de del sistema [19].

### **1.5 Conclusiones parciales**

Luego de realizar un análisis de los sistemas existentes a nivel nacional e internacional que permiten la gestión de imágenes, se determina que los mismos no cumplen con los requisitos necesarios para el desarrollo del sistema, por lo que se deriva la necesidad de realizar el análisis, diseño e implementación de un sistema que permita que los usuarios del departamento de Comunicación Audiovisual de la Universidad de las Ciencias Informáticas almacenar y obtener en un espacio centralizado todas sus imágenes. Se realizó un análisis de las tecnologías y herramientas a utilizar para el desarrollo de la herramienta, donde queda definido el proceso de desarrollo con enfoque ágil orientado al segundo nivel de CMMI. Se seleccionó Visual Paradigm v 8.0 como herramienta CASE para el modelado UML. Se determina utilizar Drupal v 7.18 como Sistema Gestor de Contenidos, condicionando el uso de los lenguajes HTML v 5.0, CSS v 3.0, JavaScript v 1.5 y PHP v 5.3.10. Como sistema para la gestión de la base de datos se decide utilizar PostgreSQL v 9.1, ya que brinda una gran estabilidad y seguridad de los datos. Como servidor web se decide utilizar Apache v 2.2.20 y como IDE de desarrollo NetBeans v 7.3. Todas estas herramientas tienen como punto común que son



---

de código abierto y poseen una amplia documentación, por lo que pueden ser usadas para el desarrollo de la aplicación.

## Capítulo #2. Descripción de la propuesta de solución

### 2.1 Introducción

El presente capítulo se realiza un análisis de la propuesta de solución aplicando el proceso de desarrollo con enfoque ágil orientado al segundo nivel de CMMI. Se define el modelo de dominio, se especifican los requisitos funcionales y no funcionales del sistema. Se define además el modelo físico, se describen las tablas de la base de datos, se seleccionan los patrones de diseño en Drupal y se describe la arquitectura.

### 2.2 Situación problemática

En la Universidad de las Ciencias Informáticas actualmente existen numerosos sitios que trabajan con imágenes en sus aplicaciones. Uno de los centros con que cuenta dicha universidad y que no está exento de su utilización es el departamento de Comunicación Audiovisual, pues realiza un trabajo minucioso con las imágenes digitales, ya sea con fines informativos mediante la prensa digital o con fines de edición en cuanto a materiales audiovisuales. En este departamento se gestionan estas imágenes de manera independiente y las utilizan de acuerdo a sus principales necesidades, pero no cuentan con un lugar centralizado donde puedan almacenarlas, permitiendo adherirles metadatos para evitar que el proceso de búsqueda y localización sea engorroso.

### 2.3 Propuesta de solución

Luego de haber realizado un estudio sobre los problemas existentes en el departamento de Comunicación de la Universidad de las Ciencias Informáticas, se propone el desarrollo de un sistema web mediante el cual sea posible el almacenamiento y obtención de las imágenes digitales en un lugar centralizado. Este sistema permitirá a los usuarios la posibilidad de adicionar y eliminar las imágenes, así como realizar una búsqueda filtrada a través de los metadatos que el sistema es capaz de adherirle a dichas imágenes como el título, el autor, la descripción, la fecha y las categorías. El usuario podrá acceder a aquellas opciones a las que está autorizado y realizar en las interfaces sólo los cambios que le sean permitidos dependiendo de los niveles de acceso que posea. El sistema permitirá al usuario que tenga como rol administrador configurar las cuentas de almacenamiento y tener la posibilidad de conocer la fecha real de la toma de la imagen.

### Personas involucradas en la solución

**Tabla 2:** Personas que intervienen en el sistema.

Personas relacionadas con el sistema	Justificación
Editor	Es aquel que tiene acceso limitado en la aplicación ya que solamente podrá adicionar y obtener las imágenes.



<b>Administrador</b>	Es el encargado de restringir los permisos a los usuarios dentro del sistema, así como asignar cuentas de almacenamiento, eliminar y adicionar las imágenes.
----------------------	--

## 2.4 Modelo de dominio

En el modelo de dominio se muestra de manera visual los principales conceptos que se manejan, ayudando a los usuarios, desarrolladores e interesados a utilizar un vocabulario común para poder entender el contexto en que se desarrolla el sistema. Además contribuirá a identificar personas, eventos, transacciones y objetos involucrados en el sistema [20].

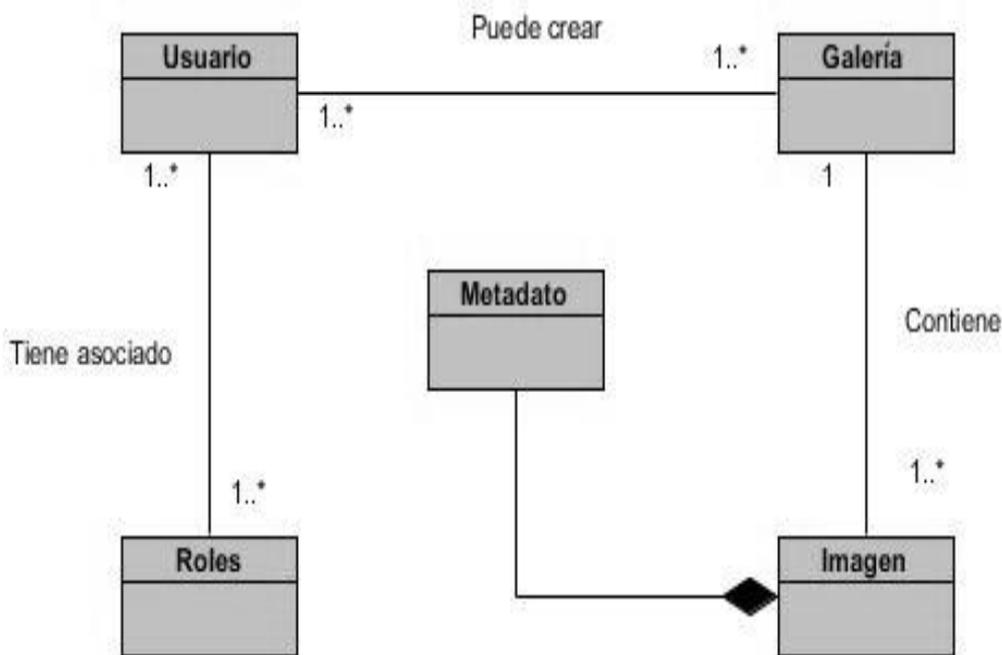


Ilustración 1: Modelo de dominio.

### Conceptos del modelo de dominio

- Usuario: es el que representa a los usuarios que visitan el sistema.
- Roles: es el que representa los roles que tienen los usuarios.
- Galería: está compuesta por las imágenes que hayan sido definidas con este propósito en el sitio.
- Imagen: es la que conforma una galería.
- Metadato: conjunto de datos que contiene una imagen.

## 2.5 Técnicas de obtención de requisitos

La obtención de requisitos es el proceso mediante el cual los interesados en un sistema de software descubren, revelan, articulan y entienden sus requisitos. En muchos casos, se requiere tiempo para llegar a especificar claramente lo que el interesado espera de la aplicación de software, por lo que se



necesario por parte de los analistas el empleo de técnicas que permitan establecer una buena comunicación con los interesados del producto y así lograr la satisfacción del cliente. A continuación se enuncian las principales técnicas utilizadas durante el proceso de desarrollo para recopilar los requisitos de software.

- **Entrevistas:** se realizó una entrevista con el cliente, permitiendo tomar conocimiento del problema y comprender los objetivos de la solución buscada.
- **Observación:** permitió comprender cómo se realiza en el departamento de Comunicación Audiovisual el proceso de almacenamiento y obtención de imágenes, así como identificar los requerimientos implícitos en los procesos reales.
- **Sistemas existentes:** se analizaron distintos sistemas ya desarrollados que estén relacionados con el objeto de estudio. Este estudio permitió la obtención de un cúmulo de ideas para la realización del sistema a desarrollar.
- **Lluvia de ideas:** se realizó a través de reuniones con un personal con experiencia en el tema, con el objetivo de generar la mayor cantidad de ideas posibles para el desarrollo del nuevo sistema.

## 2.6 Especificación de los requerimientos de software

### Requisitos funcionales

Los requisitos funcionales son capacidades o condiciones que el sistema debe cumplir. No alteran la funcionalidad del producto, por lo que se mantienen invariables sin importar las propiedades o cualidades con que se relacionen.

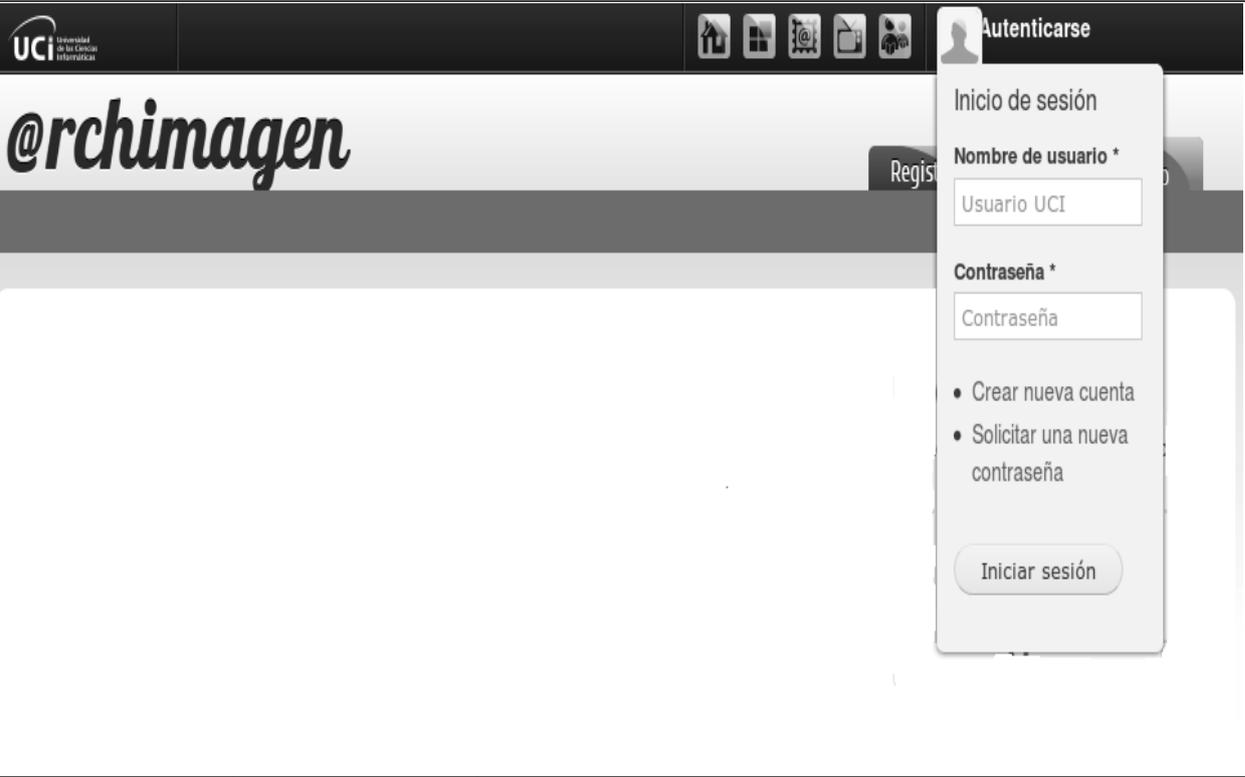
En la siguiente tabla se muestran los resultados obtenidos a partir del levantamiento de requerimientos realizado en el departamento de Comunicación Audiovisual de la UCI.

**Tabla 3:** Requisitos funcionales.

RF1_Añadir usuario	RF11_Editar galería de imágenes
RF2_Eliminar usuario	RF12_Mostrar galería de imágenes
RF3_Editar usuario	RF13_Buscar imagen
RF4_Autenticar usuario	RF14_Buscar galería
RF5_Añadir imagen	RF15_Comentar imagen
RF6_Eliminar imagen	RF16_Comentar galería
RF7_Editar imagen	RF17_Descargar imagen
RF8_Mostrar imagen	RF18_Añadir categoría a las imágenes
RF9_Crear galería de imágenes	RF19_Capturar metadatos de la imagen
RF10_Eliminar galería de imágenes	RF20_Embeber metadato fecha a la imagen

### Descripción de los requisitos funcionales críticos

**Tabla 4:** Descripción del requisito funcional crítico autenticar usuario.

Nº	Nombre	Descripción	Complejidad	Prioridad para el cliente
RF 4	Autenticar usuario.	El usuario va a la parte superior del sistema, donde se encuentra un formulario de autenticación. El usuario llena los campos (usuario y contraseña), el sistema verifica los datos introducidos con el rol que cumple dentro del sistema para asignar permisos.	Baja	Alta
<b>Prototipo</b>				
				
	<b>Campos</b>	<b>Tipos de Datos</b>	<b>Reglas o Restricciones</b>	
	<ul style="list-style-type: none"> <li>➤ usuario</li> <li>➤ contraseña</li> </ul>	<ul style="list-style-type: none"> <li>➤ varchar</li> <li>➤ varchar</li> </ul>		
	<b>Observaciones</b>			

**Tabla 5:** Descripción del requisito funcional crítico adicionar imagen.

Nº	Nombre	Descripción	Complejidad	Prioridad para el cliente
RF5	Adicionar imagen.	El usuario selecciona la opción "Adicionar Imagen".	Media	Alta
<b>Prototipo</b>				



## Adicionar Galería

Nombre \*

Mostrar pesos de la fila

Imagen

+ Adicionar Imagen

Añadir otro elemento

Autor \*

Nombre: Pedro Luis

Fecha

Fecha

05/20/2013

E.g., 06/20/2013

Hora

10:30

E.g., 10:30

Guardar

Vista previa

Campos	Tipos de Datos	Reglas o Restricciones
		➤ Para adicionar una galería en el sistema el usuario tiene que estar autenticado.
<b>Observaciones</b>		

Las especificaciones de los restantes requisitos funcionales pueden ser encontradas en los anexos (Ver [Anexo #1](#)).



## Requisitos no funcionales

Los requisitos no funcionales son propiedades o cualidades que el producto debe tener. Forman una parte significativa de la especificación. Son importantes para que los clientes y usuarios puedan valorar las características no funcionales del producto, pues si se conoce que el mismo cumple con todas las funcionalidades requeridas, las propiedades no funcionales como usable, seguro, conveniente y agradable, pueden marcar la diferencia entre un producto bien aceptado y uno con poca aceptación.

### Usabilidad

RnF 1.1 Una interfaz de manejo cómoda e intuitiva, que posibilite a los usuarios sin experiencia una rápida adaptación.

RnF 1.2 Debe contar con información actualizada, de fácil acceso, con un buen diseño visual.

### Fiabilidad

RnF 2.1 El gestor escogido debe presentar facilidades de administración de roles y usuarios restringiendo el acceso a los datos.

### Soporte

RnF 3.1 El grupo de soporte del centro CENIA se encarga de asesorar el sistema y brindarle mantenimiento.

### Restricciones de Diseño

RnF 4.1 Como proceso de desarrollo: El proceso de desarrollo con enfoque ágil orientado al segundo nivel de CMMI.

RnF 4.2 El sistema usa lenguajes de programación: PHP 5 o superior.

RnF 4.3 El servidor web que se utiliza es Apache 2.2.20.

RnF 4.4 El sistema gestor de base de datos que se utiliza es PostgreSQL 9.1.

RnF 4.5 El diseño de la base de datos está modelado con el Visual Paradigm 8.0.

RnF 4.6 El sistema operativo a utilizar en el entorno de desarrollo debe ser Ubuntu 12.04.

RnF 4.7 El repositorio principal, el entorno de prueba y el servidor de base de datos están montados sobre Ubuntu Server 9.04 o superior.

RnF 4.8 La plataforma de desarrollo utilizada es Netbeans 7.3.

### Interfaz

#### Interfaces de Usuario

RnF 5.1 El diseño gráfico es acorde con las pautas de diseño de la Universidad.



## Interfaces de Hardware

RnF 5.2 Debe existir una red de área local para que los usuarios de la Universidad puedan acceder a la aplicación.

RnF 5.3 El servidor: CPU Dual Core 2.0 GHZ o superior, memoria RAM de 4 GB (recomendado 6 GB), 250 GB HDD.

RnF 5.4 Para el desarrollo: PC Intel Pentium 4 o superior, CPU 3GHZ o superior, 512 MB RAM o superior, 160 GB HDD o superior.

RnF 5.5 Para explotación del cliente: PC Pentium 3 o superior, CPU 133 MHZ o superior, 256 RAM mínimo 512 RAM recomendada o superior.

## Requisitos Legales, de Derecho de Autor y otros

RnF 6.1 El CMS escogido como plataforma para el desarrollo de la aplicación está basado en la licencia GNU/GPL.

## Estándares Aplicables

RnF 7.1 Estándares de codificación acordes con el CMS escogido.

## 2.7 Modelado de datos

Un modelo es una representación de cualquier aspecto o tema extraído del mundo real, en una base de datos esta representación se realizara de forma gráfica.

Un modelo de datos es un conjunto de conceptos que permiten describir los datos, las relaciones que existen entre ellos, la semántica y las restricciones de consistencia [21].

De manera general el modelo de datos permite describir los elementos de la realidad que intervienen en un problema dado y la forma en que se relacionan esos elementos entre sí.

A continuación se muestran en la **Ilustración 2** las entidades que responden a las funcionalidades desarrolladas en el sistema representadas en el modelo físico.

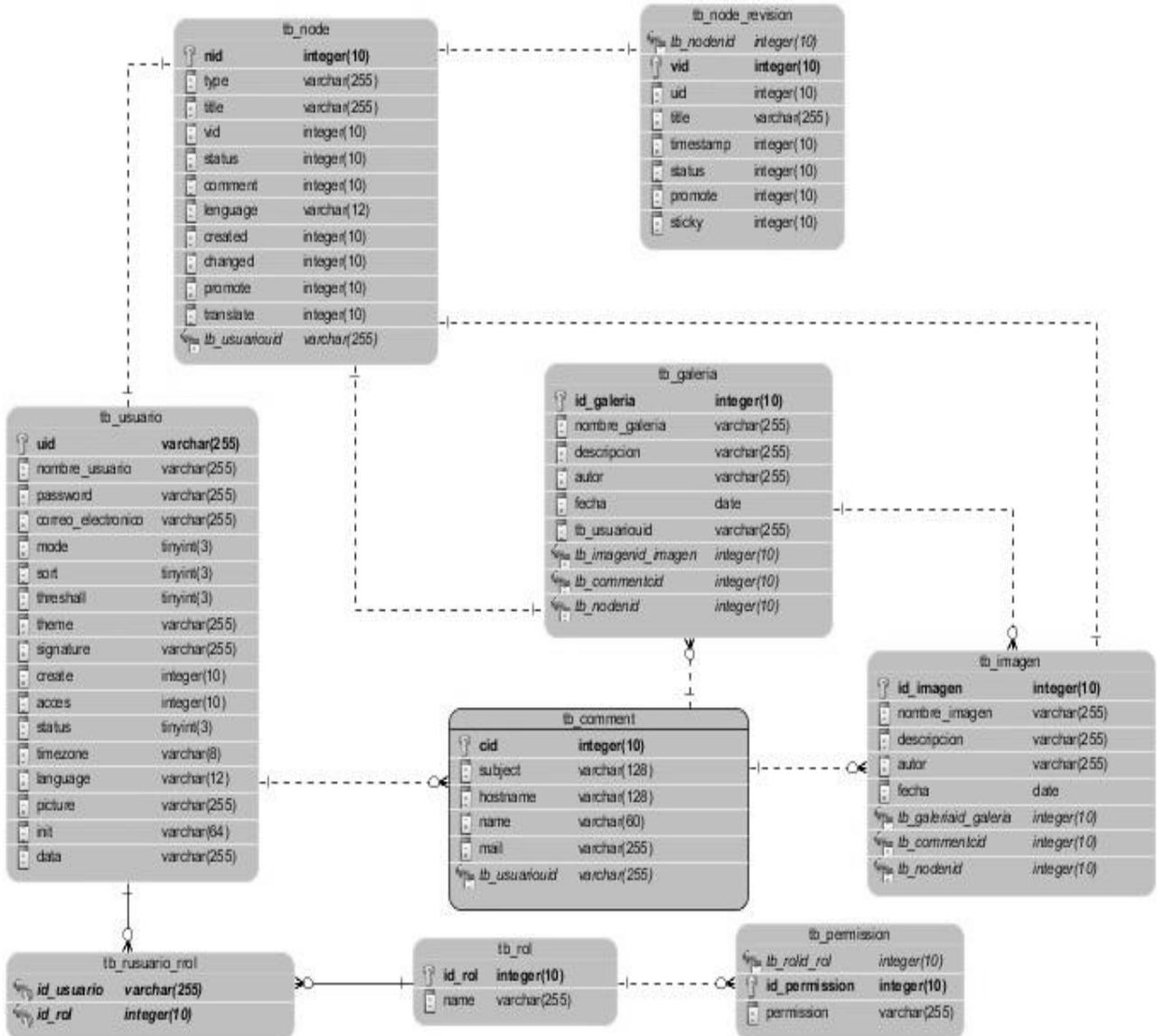


Ilustración 2: Modelo físico de datos.

**Descripción de las tablas de la base de datos.**

Tabla 7: Usuario.

tb_usuario		
Descripción	Tabla que contiene la información de los usuarios.	
Atributo	Tipo	Descripción
uid	integer	Identificador de la tabla tb_usuario.
nombre_usuario	varchar	Nombre de usuario.
password	varchar	Contraseña de usuario.
correo_electrónico	varchar	Correo electrónico del usuario.
mode	tinyint	Se utiliza para llevar el control de los usuarios.
sort	tinyint	Rol del usuario.



<i>theme</i>	<i>varchar</i>	Tema especificado para el usuario.
<i>signature</i>	<i>varchar</i>	Firma del usuario.
<i>created</i>	<i>integer</i>	Fecha de creación del usuario.
<i>access</i>	<i>integer</i>	Fecha de último acceso.
<i>login</i>	<i>integer</i>	Fecha y hora de la última vez que se autenticó el usuario.
<i>status</i>	<i>tinyint</i>	Estado del usuario (activo o bloqueado).
<i>timezone</i>	<i>varchar</i>	Zona horaria.
<i>language</i>	<i>varchar</i>	Idioma de la interfaz del sitio que se le presenta al usuario.
<i>picture</i>	<i>varchar</i>	Avatar del usuario.
<i>init</i>	<i>varchar</i>	Correo electrónico del usuario.
<i>data</i>	<i>varchar</i>	Información sobre la cuenta (registro).

**Tabla 8:** Rol.

<b>tb_rol</b>		
<b>Descripción</b>		
Tabla que guarda los roles definidos en el sistema.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
<i>id_rol</i>	<i>integer</i>	Identificador de la tabla <i>tb_rol</i> .
<i>name</i>	<i>varchar</i>	Rol del sistema.

**Tabla 9:** Usuario\_Rol.

<b>tb_rusuario_rrol</b>		
<b>Descripción</b>		
Tabla que surge a partir de la relación de mucho a mucho de la tabla <i>tb_usuario</i> y <i>tb_rol</i> .		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
<i>id_usuario</i>	<i>integer</i>	Identificador de la tabla <i>tb_usuario</i> .
<i>id_rol</i>	<i>integer</i>	Identificador de la tabla <i>tb_rol</i> .

**Tabla 10:** Permission.

<b>tb_permission</b>		
<b>Descripción</b>		
Tabla que guarda permisos que tienen los <i>roles</i> establecidos en el sistema.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
<i>id_permission</i>	<i>integer</i>	Identificador de la tabla <i>tb_permission</i> .
<i>id_rol</i>	<i>integer</i>	Identificador de la tabla <i>tb_rol</i> .
<i>permission</i>	<i>varchar</i>	Permisos que tiene el <i>tb_rol</i> .

**Tabla 11:** Imagen.

<b>tb_imagen</b>		
<b>Descripción</b>		
Tabla que guarda los datos asociados a las imágenes.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
<i>id_galería</i>	<i>integer</i>	Identificador de la tabla <i>galería</i> .
<i>id_imagen</i>	<i>integer</i>	Identificador de la tabla <i>imagen</i> .



nombre_imagen	<i>varchar</i>	Dirección de la imagen.
descripción	<i>varchar</i>	Especifica la descripción de la imagen.
autor	<i>varchar</i>	Nombre del autor de la imagen.
fecha	<i>date</i>	Fecha de creación de la imagen.

**Tabla 12:** Galería.

<b>tb_galería</b>		
<b>Descripción</b>	<i>Tabla que guarda los datos asociados a la galería.</i>	
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
<i>id_usuario</i>	<i>integer</i>	<i>Identificador de la tabla tb_usuario.</i>
<i>id_galería</i>	<i>integer</i>	<i>Identificador de la tabla tb_galería.</i>
<i>nombre</i>	<i>varchar</i>	<i>Nombre de la galería.</i>
<i>descripcion</i>	<i>varchar</i>	<i>Especifica la descripción de la galería.</i>
<i>fecha</i>	<i>date</i>	<i>Fecha de creación de la galería.</i>

**Tabla 13:** Comment.

<b>tb_comment</b>		
<b>Descripción</b>	<i>Tabla que guarda los datos asociados los comentarios.</i>	
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
<i>cid</i>	<i>integer</i>	<i>Identificador de la tabla tb_comment.</i>
<i>uid</i>	<i>integer</i>	<i>Identificador de la tabla tb_usuario.</i>
<i>subject</i>	<i>varchar</i>	<i>Asunto del comentario.</i>
<i>hostname</i>	<i>varchar</i>	<i>Dirección IP de la PC del usuario que comenta.</i>
<i>name</i>	<i>varchar</i>	<i>Nombre del usuario.</i>
<i>mail</i>	<i>Varchar</i>	<i>Dirección de correo electrónico.</i>

**Tabla 14:** Node.

<b>tb_node</b>		
<b>Descripción</b>	<i>Tabla que guarda los datos asociados los nodos.</i>	
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
<i>nid</i>	<i>integer</i>	<i>Identificador del nodo.</i>
<i>type</i>	<i>varchar</i>	<i>Tipo de contenido.</i>
<i>lenguaje</i>	<i>varchar</i>	<i>Define el lenguaje en que está el nodo.</i>
<i>title</i>	<i>varchar</i>	<i>Título del nodo.</i>
<i>uid</i>	<i>integer</i>	<i>Identificador del usuario que creó el nodo.</i>
<i>status</i>	<i>integer</i>	<i>1- si está publicado, 0- si no está publicado.</i>
<i>created</i>	<i>integer</i>	<i>Fecha de creación.</i>
<i>changed</i>	<i>integer</i>	<i>Fecha de la última modificación.</i>
<i>comment</i>	<i>integer</i>	<i>Cantidad de comentarios realizados sobre ese nodo.</i>



promote	<i>integer</i>	1- Si está promovido a la página principal, 0- si no.
sticky	<i>integer</i>	1- Si tiene la propiedad pegajosa en la cima de las listas.
translate	<i>integer</i>	Identificador del nodo que representa la traducción a otro lenguaje.

**Tabla 15:** Node\_revision.

<b>tb_node_revision</b>		
<b>Descripción</b>	<i>Tabla que guarda los datos asociados los nodos.</i>	
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
nid	<i>integer</i>	Identificador del nodo.
Vid	<i>integer</i>	Identificador de la versión.
Uid	<i>integer</i>	Identificador del usuario que creó el nodo.
Title	<i>varchar</i>	Título del nodo.
Timestamp	<i>int</i>	Fecha de creado.
status	<i>integer</i>	1- si está publicado, 0- si no está publicado.
promote	<i>integer</i>	1- Si está promovido a la página principal, 0- si no.
sticky	<i>integer</i>	1- Si tiene la propiedad pegajosa en la cima de las listas.

## 2.8 Patrones de diseño en Drupal

Debido a que Drupal define una ingeniería propia, el sistema utiliza algunos patrones de diseño que son propiamente de sistemas orientados a objetos, particularmente los patrones GOF (Gang of Four) que contribuyen a la construcción de un diseño más elegante y robusto. Algunos de estos patrones se explican a continuación:

### ➤ **Singleton (Instancia única)**

Si se piensan los módulos como objetos, entonces siguiendo este patrón, se puede mencionar que en general estos objetos no encapsulan datos, lo que separa un módulo de otro es el conjunto de funciones que contiene. Se garantiza de esta manera, la existencia de una única instancia de este objeto y la creación de un mecanismo de acceso global a dicha instancia, por lo que debe ser tratado como una clase con una instancia singleton.

En la aplicación se crea una instancia global del módulo “metadato” al cual se le hace peticiones en dependencia del contenido que se esté creando.

### ➤ **Bridge (Puente)**

Los módulos deben ser escritos independientemente del sistema de base de datos que se utiliza y la capa de abstracción que esta proporcionada para ello. Se pueden definir nuevas capas de base de datos de acuerdo a la Interfaz de Programación de Aplicaciones (API, por sus siglas en inglés) definida, donde se añade soporte para otros sistemas de base de datos sin necesidad de modificar el código del módulo, desacoplando de esta manera la abstracción de su implementación.

### ➤ **Command (Orden)**

Muchos de los ganchos de Drupal utilizan el patrón de mando para reducir el número de funciones que son necesarias para la aplicación, pasando la operación como un parámetro junto con los argumentos. De hecho, el propio sistema de gancho utiliza este patrón, a fin de que los módulos no tengan que definir cada gancho.

En el módulo “metadato” se evidencia este patrón al capturar toda la información referente a un nodo, donde se utiliza el gancho de Drupal `hook_node_presave()` y pasándole por parámetro la variable `nodo` que contiene toda la información referente al mismo.

### ➤ **Chain of Responsibility (Cadena de responsabilidad)**

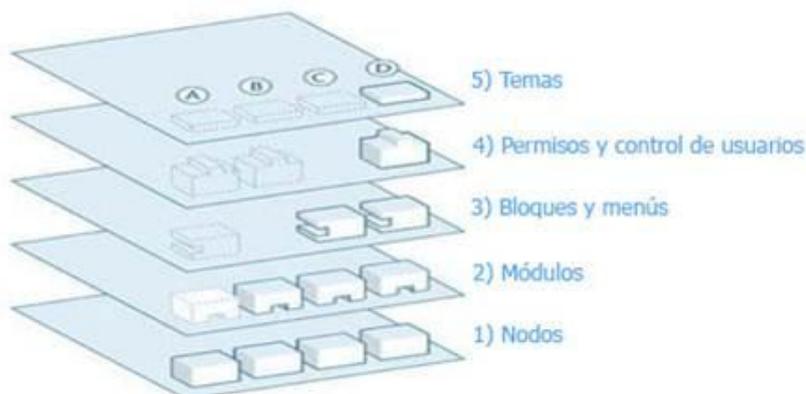
En cada solicitud de la página, el menú del sistema determina si hay un módulo para gestionar la solicitud, si el usuario tiene acceso a los recursos solicitados y la función que se llama para hacer el trabajo. Para ello, el mensaje se pasa a la opción del menú correspondiente a la vía de la solicitud. Si el elemento de menú no puede manejar la petición, se pasa a otro eslabón 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 [22].

En la aplicación se evidencia este patrón en las diferentes peticiones realizadas por los usuarios. Un ejemplo de ello es cuando el mismo se autentica en la aplicación y realiza la solicitud ver banco de imágenes. El menú del sistema determina si hay un módulo para gestionar esta solicitud. Si el usuario tiene acceso al recurso solicitado se le asocia el permiso “*access content*” para ver el banco de imágenes. En caso de que el usuario no posea los permisos para ejecutar la petición deseada el sistema niega el permiso o la cadena se agota.

## 2.9 Arquitectura de Drupal

### Estructuración del contenido en Drupal

Drupal estructura el contenido en una serie de elementos básicos, todos ellos mostrados en la **Ilustración 3**, estos son: nodos, módulos, bloques y menús, permisos de usuario y plantillas.



**Ilustración 3:** Arquitectura de Drupal.



Los nodos son los elementos básicos de Drupal, en los que la información es almacenada. Así cuanto más crece un sitio web, mayor es el número de nodos que lo componen. Es la primera capa de la arquitectura de Drupal.

Los módulos operan sobre los nodos y le dan funcionalidad a Drupal permitiendo incrementar su funcionalidad o adaptarlas a la necesidad de cada sitio.

Sobre la capa anterior está la que constituyen los bloques y menús. Dicha capa es la encargada de estructurar y organizar los contenidos en las páginas. Son los bloques y menús los elementos que albergan y permiten acceder al usuario a la salida generada y procesada por los módulos a partir de la información almacenada en los nodos.

La siguiente capa es la de control de usuarios y permisos. Drupal dispone de un registro de usuarios y roles que permiten especificar qué tareas pueden realizar de las que ofrecen los módulos instalados en el sitio así como el contenido que se puede crear, al que se puede acceder o el que puede visualizar cada usuario o tipo de usuario (rol). Esto es lo mismo que decir que esta capa controla las operaciones (visualización, edición, creación, etc.) que un usuario puede realizar sobre elementos de las capas inferiores.

La última capa, es la capa de las plantillas o *templates* es la que establece la apariencia o estilo de la información a mostrar al usuario. Esta separación entre información y aspecto permite cambiar la apariencia de un sitio web sin modificar el contenido [23].

### **Patrones de arquitectura presentes en la aplicación**

**Nodos:** en esta capa es donde se almacenan los nodos creados que responden a las funcionalidades de la aplicación. Los mismos son imagen, galería y comentario.

**Módulos:** en esta capa se encuentra el módulo “metadato” y demás módulos de la aplicación que brindan tanto interfaz como funcionalidad al sistema dándole respuesta a cada petición del usuario.

**Bloques y menús:** en esta capa se encuentran los bloques y menús del sistema como son:

- El bloque “¿Qué deseas hacer?” es el que permite al usuario seleccionar que acción principal realizará en el sistema, ya sea adicionar imagen, adicionar galería o banco de imágenes.
- El bloque “Usuarios Conectados” es el que muestra los últimos usuarios conectados al sistema.
- El bloque “Búsqueda General” es el que le permite al usuario realizar una búsqueda general en el sistema, ya sea por usuario o nodo.



- El bloque “Filtros de búsquedas” es el que le permite al usuario realizar los filtros de búsquedas en el sistema, ya sea por galería, imagen, categoría o autor.
- El bloque “Comentarios recientes” es el que le muestra al usuario los últimos 5 comentarios realizados en el sistema.
- El menú “Contáctenos” le permite al usuario enviar las sugerencias al administrador sobre el sistema.
- El menú “Galerías” le permite al usuario seleccionar la galería que desea conocer.
- El menú “Reportes” le permite al usuario observar las últimas acciones realizadas en el sistema a través de un calendario.

**Permisos y control de usuarios:** en esta capa se almacenan los permisos asociados a los roles que son asignados a cada usuario como son:

- access administration pages: es el permiso que se le otorga al rol para la administración del sistema.
- access all views: es el permiso que se le otorga a los roles para acceder a las imágenes y galerías.
- access comment: es el permiso que se le otorga al a los roles para realizar comentarios a las imágenes y galerías.
- access content: es el permiso que se le otorga a los roles para acceder al contenido.

**Temas:** es la capa que contiene el tema “CENIA” que muestra la interfaz de la aplicación al usuario.

## 2.10 Conclusiones parciales

En el capítulo concluido se definieron los requisitos funcionales y no funcionales, lo que permitió obtener el diseño del Sistema para la gestión de las imágenes que se utilizan en la publicación de noticias en el departamento de Comunicación Audiovisual de la Universidad de las Ciencias Informáticas. Con el análisis y diseño realizado se logró un mejor entendimiento de las principales funcionalidades del sistema. Se realizó además un análisis de la arquitectura, lo que permitió un mejor entendimiento del código de la aplicación. Al concluir este capítulo quedan resueltas la mayoría de las tareas propuestas para la elaboración de la solución al problema planteado.

## Capítulo #3. Implementación y prueba

### 3.1 Introducción

En el presente capítulo se describen los estándares de codificación, el diagrama de componentes y el diagrama de despliegue. Posteriormente se realizan las pruebas de caja negra y pruebas de rendimiento como carga y estrés para verificar el funcionamiento del sistema de acuerdo a los requisitos planteados.

### 3.2 Estándares de codificación

Un estándar de codificación es un conjunto de políticas por las que se tiene que regir un programador a la hora de desarrollar un programa e identificar la estructura del código.

El desarrollo del módulo “metadato” se rige en gran medida por los estándares de codificación establecidos para Drupal. A continuación se listan los más relevantes:

#### ➤ Indentación

La indentación consiste en insertar espacios en blanco o tabuladores en determinadas líneas de código para facilitar su comprensión. En programación se utiliza la indentación para anidar elementos.

En Drupal se debe indentar con 2 espacios, nunca con tabuladores. Además, no se debe dejar espacios en blanco al final de cada línea. En el siguiente ejemplo se muestra un fragmento de código con las indentaciones realizadas, de 2 espacios cada una (->->), y los saltos de línea al final de cada línea (sin dejar espacios) [24].

```
<?php
function forum_block_view($delta = '') {
->>$query = db_select('forum_index', 'f')
->>>>->fields('f')
->>>>->addTag('node_access');
->>switch ($delta) {
->>>>case 'active':
->>>>>>$title = t('Active forum topics');
->>>>>>$query
->>>>>>->orderBy('f.last_comment_timestamp', 'DESC')
->>>>>>->range(0, variable_get('forum_block_num_active', '5'));
->>>>>>break;
->>>>case 'new':
->>>>>>$title = t('New forum topics');
->>>>>>$query
->>>>>>->orderBy('f.created', 'DESC')
->>>>>>->range(0, variable_get('forum_block_num_new', '5'));
->>>>>>break;
->>}
->>$block['subject'] = $title;
->>// Cache based on the altered query.
->>$block['content'] = drupal_render_cache_by_query($query,
'forum_block_view');
->>$block['content']['#access'] = user_access('access content');
->>return $block;
}
?>
```



### ➤ Operadores

Los operadores binarios, que se utilizan entre dos valores, deben separarse de estos valores, a ambos lados del operador, por un espacio. Por ejemplo, **\$numero = 3**, en lugar de \$numero=3. Esto se aplica a operadores como +, -, \*, /, =, ==, !=, >, <, . (concatenación de cadenas), .=, +=, -=, etc.

Los operadores unarios como ++, -- no deben tener separación. Por ejemplo, **\$numero++**.

### ➤ Uso de comillas

Se pueden usar tanto las comillas simples ('cadena') como las comillas dobles ("cadena") para delimitar las cadenas de caracteres.

Las comillas dobles son necesarias si se desean incluir variables dentro de las cadenas de texto. Por ejemplo, "<h1>\$title</h1>". También se recomienda el uso de comillas dobles cuando el texto puede incluir alguna comilla simple.

### ➤ Uso de punto y coma (;) en código PHP

Aunque PHP permite escribir líneas de código individuales sin el terminador de línea (;), como por ejemplo <?php print \$title ?>. En Drupal es **siempre** obligatorio: <?php print \$title; ?>.

### ➤ Funciones

Los nombres de las funciones deben estar escritos en minúsculas y las palabras separadas por guión bajo. Además, se debe incluir siempre como prefijo el nombre del módulo, tema, etc., para evitar así duplicidad de funciones.

En su declaración, después del nombre de la función, el paréntesis de inicio de los argumentos debe ir **sin espacio**. Cada argumento debe ir separado por un espacio, después de la coma del argumento anterior.

---

```
function forum_help($path, $arg) {
```

---

En la llamada a la función se aplican las mismas reglas anteriores con respecto a los parámetros, como se muestra en el siguiente ejemplo:

---

```
$field = field_info_instance('node', 'taxonomy_forums', $node->type);
```

---

Como excepción, es posible usar más de un espacio antes de una asignación (=) para mejorar la presentación, cuando se estén realizando varias asignaciones en bloque [24]:

---

```
$numerol      = foo($a, $type);  
$primer_valor = foo2($b);  
$i            = foo3();
```

### ➤ Convención de nomenclatura

**Variabes:** se rigen por la nomenclatura camelCase. Siempre comienzan con minúscula y en caso de nombres compuestos la primera letra de cada palabra comienza con mayúscula.

- 1....\$variable
- 2....\$variableNombreCompuesto

**Constantes:** constantes siempre debe ser todo en mayúsculas, con caracteres de subrayado “\_” para separar palabras en caso de nombres compuestos. [23].

- 1....define (CONSTANTE, valor);
- 2....define (CONSTANTE\_COMPUESTO, valor);

### ➤ Estructuras de control

Con respecto a las estructuras de control, hay que tener en cuenta las siguientes normas:

Debe haber un espacio entre el comando que define la estructura (if, while, for, etc.) y el paréntesis de apertura. Esto es así para no confundir las estructuras de control con la nomenclatura de las funciones.

La llave de apertura {se situará en la misma línea que la definición de la estructura, separada por un espacio.

Se recomienda usar siempre las llaves {} aún en los casos en que no sea obligatorio su uso (una sola "línea" de código dentro de la estructura de control).

Las estructuras else y elseif se escribirán en la línea siguiente al cierre de la sentencia anterior [24].

```
//sentencia if  
if (condicion1 || condicion2) {  
    accion1;  
}  
elseif (condicion3 && condicion4) {  
    accion2;  
}  
else {  
    acciones_por_defecto;  
}
```

```
//sentencia switch
switch (condicion) {
    case 1:
        accion1;
        break;

    case 2:
        accion2;
        break;

    default:
        acciones_por_defecto;
}

//sentencia for
for ($i = 0; $i < 5; $i++) {
    acciones;
}
```

### ➤ Documentación

Todos los archivos deben de tener la documentación asociada al mismo. Para esto debe de cumplir con el siguiente bloque al principio de cada clase.

#### Clase:

```
1 /**
2 *Breve descripción de la clase
3 *
4 *PHP versión #
5 *
6 *@category Categoría de la clase implementada "Librería,
7 * Controladora, Modelo"
8 *@package Nombre del paquete o módulo al que pertenece
9 *@author Nombre y Apellidos del autor y correo electrónico
10*/
```

#### Funciones:

```
1 /**
2 *Breve descripción de la función
3 *
4 *@param tipo y nombre del parámetro (por cada parámetro que
5 * recibe la función)
6 *@return tipo que retorna
7 *@author Nombre y Apellidos del autor y correo electrónico
```



Los valores booleanos y nulos siempre se escriben con mayúscula, para facilitar la legibilidad del código usar un *enter* antes de las estructuras de control y definición de las funciones [24].

```
1....$variableBooleana = FALSE;  
2....$variableNula = NULL;  
3....  
4....if (condicion)  
5.... {  
6....//Bl  
7....}
```

### 3.3 Diagrama de componentes

Un diagrama de componentes muestra las organizaciones y dependencias lógicas entre componentes software, ya sean componentes de código fuente, binarios o ejecutables. Desde el punto de vista del diagrama de componentes se tienen en consideración los requisitos relacionados con la facilidad de desarrollo, la gestión del software, la reutilización, las restricciones impuestas por los lenguajes de programación y las herramientas utilizadas en el desarrollo [25].

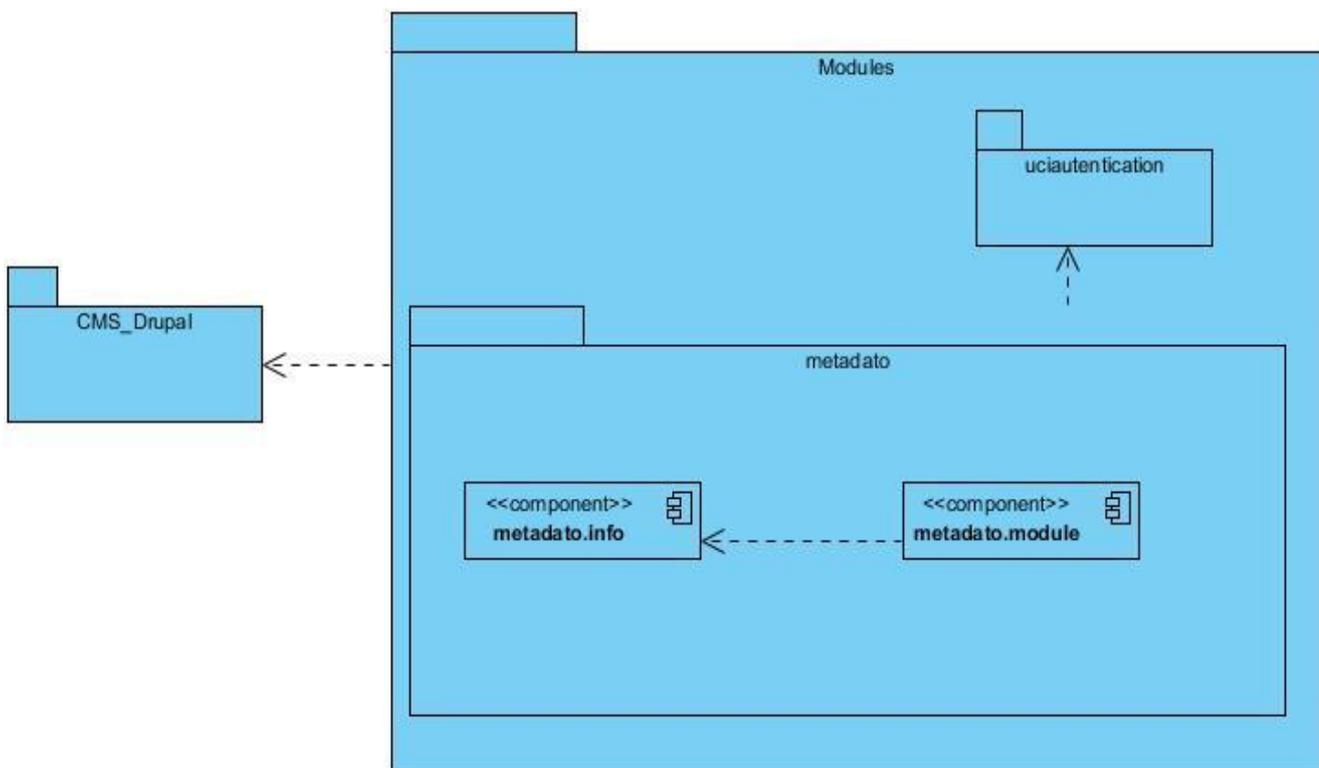


Ilustración 4: Diagrama de componentes.

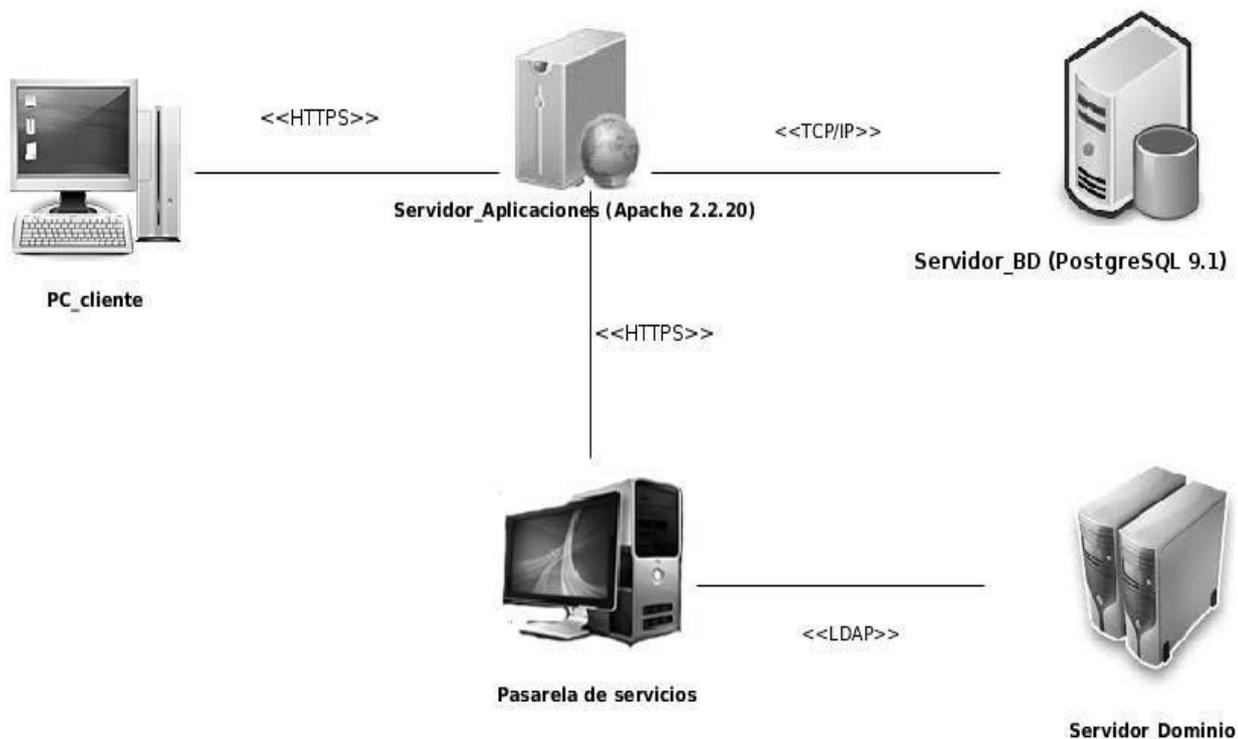
El módulo “metadato” cuenta con un conjunto de funcionalidades las cuales están implementadas en el componente metadato.module. Dicho componente presenta las funcionalidades necesarias para realizar

la captura de los metadatos que contienen las imágenes. Para poder ser instalado el mismo hace uso del componente `archivo.info` donde se recoge la información respecto al módulo.

Para que la aplicación lleve a cabo el almacenamiento y obtención de imágenes el usuario tiene que estar autenticado y para esto se utiliza el módulo `uciauthentication`.

### 3.4 Diagrama de despliegue

El diagrama de despliegue es un modelo de objetos que describe la distribución física del sistema, especificando la distribución de la funcionalidad del mismo entre los nodos de cómputo que utiliza. No es más que la representación física de todos los recursos que utiliza el sistema, por ejemplo, nodos con capacidad de procesamiento, módems, impresoras, entre otros [26].



**Ilustración 5:** Diagrama de despliegue.

La PC cliente representa el conjunto de computadoras a través de las cuales los usuarios pueden actualizar y consultar la información que se encuentra en el servidor web. Para acceder al sistema, las PC clientes utilizan una aplicación cliente, en este caso, un navegador web. La comunicación entre las PC clientes y el servidor web se establece utilizando el conjunto de protocolos de comunicación HTTPS.

En el servidor web, se ubican íntegramente las capas de presentación, lógica del negocio y de acceso a datos del sistema, así como los servicios que se brindan.

El nodo servidor de dominio representa el servidor en el cual se recogen funciones de almacenamiento de nombres de usuarios y contraseñas, además de los permisos asociados a cada grupo de usuarios.

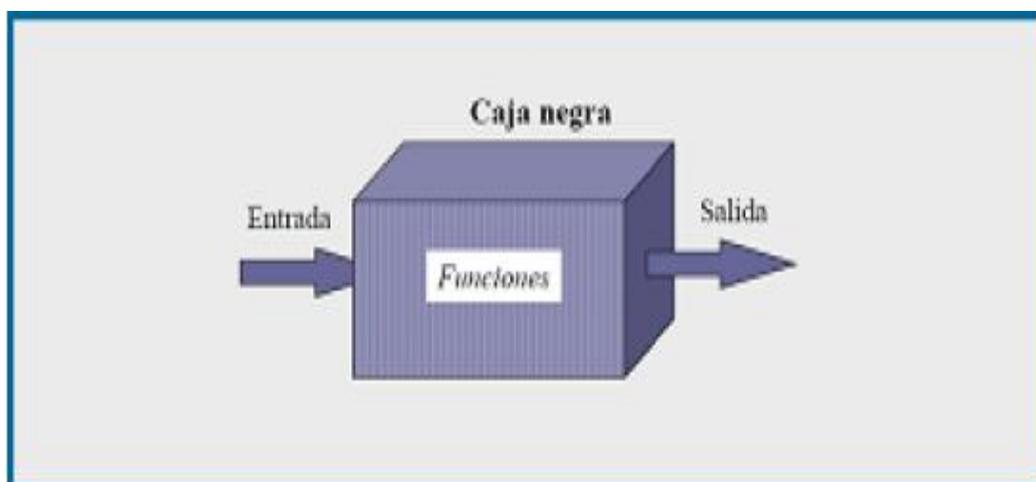
El nodo servidor de base de datos representa un servidor PostgreSQL, en el cual se ubica toda la información persistente del sistema, almacenándose los datos que son actualizados y consultados por los usuarios del sistema.

### 3.5 Prueba de software

Las pruebas son un conjunto de actividades que se pueden planificar por adelantado y llevar a cabo sistemáticamente. Por esta razón, se debe definir en el proceso de la ingeniería del software una plantilla para las pruebas del software: un conjunto de pasos en los que se puede situar los métodos específicos de diseño de casos de prueba [27].

#### 3.5.1 Método de caja negra

Las pruebas de caja negra se centran principalmente en los requisitos funcionales del software. Estas pruebas permiten obtener un conjunto de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un programa. En ellas se ignora la estructura de control, concentrándose en los requisitos funcionales del sistema y ejercitándolos [28].



**Ilustración 6:** Representación de prueba de Caja Negra.

**Fuente:** <http://www.monografias.com/trabajos75/proyectos-nformaticos/proyectos-informaticos3.shtml>.

A continuación se muestran los casos de prueba realizados al sistema mediante el método de Caja Negra utilizando la técnica de partición de equivalencia.

#### ➤ Autenticar usuario.

**Descripción:**



El usuario se sitúa en el formulario de autenticación que se encuentra en la parte superior derecha del sistema. Finalmente el sistema valida los datos introducidos por el usuario y el rol que desempeña dentro del sistema para asignar permisos.

**Tabla 16:** Caso de prueba "Autenticar usuario".

Escenario	Descripción	Usuario	Contraseña	Respuesta del sistema	Flujo Central
EC 1.1 Autenticar usuario correctamente.	Se autentica el usuario y accede a la aplicación.	V (aatellez).	V (La contraseña de la persona que se quiere autenticar).	El sistema valida los datos y accede a la aplicación con los permisos que posee.	1-El usuario da clic en el formulario de autenticación que se encuentra en la parte superior derecha del sistema. 2-El sistema muestra un formulario con los campos asociados. 3-El usuario introduce los datos y selecciona el botón "Iniciar sesión". 4-El sistema valida los datos. 5-El usuario accede a la aplicación.
EC 1.2 Autenticar usuario dejando campos vacíos.	Se autentica el usuario y accede a la aplicación.	V (aatellez).	I (El usuario deja el campo contraseña vacío).	El sistema valida los datos y muestra un mensaje "Usuario o Contraseña incorrecto".	1-El usuario da clic en el formulario de autenticación que se encuentra en la parte superior derecha del sistema. 2-El sistema



					<p>muestra un formulario con los campos asociados.</p> <p>3-El usuario introduce los datos y selecciona el botón "Iniciar sesión".</p> <p>4-El sistema valida los datos y muestra un mensaje "Usuario o Contraseña incorrecto".</p> <p>5-El usuario no accede a la aplicación.</p>
EC 1.3 Autenticar usuario con información en los campos incorrectos.	Se autentica el usuario y accede a la aplicación.	I (addiel99)	V (La contraseña de la persona que se quiere autenticar).	El sistema valida los datos y muestra un mensaje "El usuario "addiel99" o la contraseña son incorrectos".	<p>1-El usuario da clic en el formulario de autenticación que se encuentra en la parte superior derecha del sistema.</p> <p>2-El sistema muestra un formulario con los campos asociados.</p> <p>3-El usuario introduce los datos y selecciona el botón "Iniciar sesión".</p> <p>4-El sistema valida los datos y</p>



					<p>muestra un mensaje "Usuario o Contraseña incorrecto".</p> <p>5-El usuario no accede a la aplicación.</p>
<p>EC 1.4 Autenticar usuario con el dominio UCI no disponible.</p>	<p>Se autentica el usuario y accede a la aplicación.</p>	<p>V (aatellez).</p>	<p>V (La contraseña de la persona que se quiere autenticar).</p>	<p>El sistema valida los datos y muestra un mensaje "El Dominio UCI no está disponible".</p>	<p>1-El usuario da clic en el formulario de autenticación que se encuentra en la parte superior derecha del sistema.</p>
					<p>2-El sistema muestra un formulario con los campos asociados.</p> <p>3-El usuario introduce los datos y selecciona el botón "Iniciar sesión".</p> <p>4-El sistema valida los datos y muestra un mensaje "El Dominio UCI no está disponible".</p> <p>5-El usuario no accede a la aplicación.</p>

➤ **Adicionar imagen**

**Descripción:**



El usuario previamente autenticado accede a esta funcionalidad donde adiciona la imagen sin asociarla a una galería.

**Tabla 17:** Caso de prueba "Adicionar imagen".

Escenario	Descripción	Nombre	Descripción	Examinar	Autor	Fecha	Categoría	Respuesta del sistema	Flujo Central
EC 1.1 Adicionar imagen correctamente.	Se selecciona la opción Adicionar imagen.	V (El usuario llena el campo nombre de la imagen)	N/A	El usuario da clic en examinar y escoge la imagen a subir.	V (El campo autor es mostrado con el nombre del usuario).	El usuario selecciona la fecha en que sube la imagen.	V (El usuario selecciona una categoría relacionada con la imagen).	El sistema valida los datos y añade la imagen.	1-El usuario selecciona la opción "Adicionar imagen". 2-El sistema muestra el formulario con los campos asociados a este. 3-El usuario llena los campos y selecciona el botón "Guardar" 4-El sistema valida los datos. 5-Se adiciona la imagen y el sistema muestra la imagen con sus metadatos.



EC 1.2	Se selecciona la opción Adicionar imagen. dejando campos obligatorios vacíos.	I (El usuario deja el campo nombre vacío).	N/A	El usuario da clic en examinar y escoge la imagen a subir.	V (El campo autor es mostrado con el nombre del usuario).	El usuario selecciona la fecha en que sube la imagen.	V (El usuario selecciona una categoría relacionada con la imagen).	El sistema valida los datos y muestra un mensaje de error "El campo Nombre es obligatorio."	1-El usuario selecciona la opción "Adicionar imagen". 2-El sistema muestra el formulario con los campos asociados a este. 3-El usuario llena los campos y selecciona el botón "Guardar". 4-El sistema valida los datos y muestra un mensaje de error "El campo Nombre es obligatorio." 5-El sistema no adiciona la imagen.
EC 1.3	Se selecciona la opción Adicionar imagen dejando	V (El usuario llena el campo	N/A	El usuario da clic en examinar y escoge	I (El usuario desmarca el nombre	El usuario selecciona la	V (El usuario selecciona una	El sistema valida los datos y muestra un	1-El usuario selecciona la opción



campos obligatorios vacíos.	imagen.	nombre de la imagen)		la imagen a subir.	del autor).	fecha en que sube la imagen.	categoría relacionada con la imagen).	mensaje de error "El campo Autor es obligatorio."	"Adicionar imagen". 2-El sistema muestra el formulario con los campos asociados a este. 3-El usuario llena los campos y selecciona el botón Guardar. 4-El sistema valida los datos y muestra un mensaje de error "El campo Autor es obligatorio." 5-El sistema no adiciona la imagen
EC 1.4 Adicionar imagen dejando campos obligatorios vacíos.	Se selecciona la opción Adicionar imagen.	V (El usuario llena el campo nombre de la imagen)	N/A	El usuario da clic en examinar y escoge la imagen a subir.	V (El campo autor es mostrado con el nombre del usuario).	El usuario selecciona la fecha en que sube la imagen.	I (El usuario no selecciona una categoría para la imagen).	El sistema valida los datos y muestra un mensaje de error "El campo Categoría es	1-El usuario selecciona la opción "Adicionar Imagen". 2-El sistema muestra el



								obligatorio.”	formulario con los campos asociados a este. 3-El usuario llena los campos y selecciona el botón Guardar. 4-El sistema valida los datos y muestra un mensaje de error “El campo Categoría es obligatorio.” 5-El sistema no adiciona la imagen
EC 1.5	Se selecciona la opción Adicionar imagen.	V (El usuario llena el campo nombre de la imagen)	N/A	El usuario da clic en examinar y escoge la imagen a subir.	V (El campo autor es mostrado con el nombre del usuario).	El usuario selecciona la fecha en que sube la imagen.	V (El usuario selecciona una categoría para la imagen).	El sistema no añade la imagen.	1-El usuario selecciona la opción “Adicionar imagen”. 2-El sistema muestra el formulario con los campos asociados



a este.  
3-El usuario llena los campos y selecciona el botón "Cancelar".  
4-El sistema no adiciona la imagen.

➤ **Adicionar galería**

**Descripción:**

El usuario previamente autenticado accede a esta funcionalidad donde adiciona la galería y asocia las imágenes que contendrá.

**Tabla 18:** Caso de prueba "Adicionar galería".

Escenario	Descripción	Nombre	Imagen	Autor	Fecha	Respuesta del sistema	Flujo Central
EC 1.1 Adicionar galería correctamente.	Se selecciona la opción Adicionar galería.	V (El usuario llena el campo nombre de la galería).	El usuario escribe el nombre de una imagen previamente adicionada o selecciona la opción "Adicionar imagen" llena los campos asociados y dicha imagen se asocia a la galería.	V (El campo autor es mostrado con el nombre del usuario).	El usuario selecciona la fecha en que sube la galería.	El sistema valida los datos y añade la imagen.	1-El usuario selecciona la opción "Adicionar galería". 2-El sistema muestra el formulario con los campos asociados a este. 3-El usuario llena los campos y selecciona el botón "Guardar". 4-El sistema valida los datos.



							5-Se adiciona la galería y se muestra con sus imágenes asociadas.
EC 1.2	Se selecciona la opción Adicionar galería. dejando campos obligatorios vacíos.	I (El usuario deja el campo nombre vacío).	El usuario escribe el nombre de una imagen previamente adicionada o selecciona la opción "Adicionar Imagen" llena los campos asociados y dicha imagen se asocia a la galería.	V (El campo autor es mostrado con el nombre del usuario).	El usuario selecciona a la fecha en que sube la galería.	El sistema valida los datos y muestra un mensaje de error "El campo Nombre es obligatorio."	1-El usuario selecciona la opción "Adicionar galería". 2-El sistema muestra el formulario con los campos asociados a este. 3-El usuario llena los campos y selecciona el botón "Guardar". 4-El sistema valida los datos y muestra un mensaje de error "El campo Nombre es obligatorio." 5-El sistema no adiciona la galería.
EC 1.3	Se selecciona la opción Adicionar galería. dejando campos obligatorios	V (El usuario llena el campo nombre de la	El usuario escribe el nombre de una imagen previamente adicionada o	I (El usuario desmarca el nombre del autor).	El usuario selecciona a la fecha en que sube la galería.	El sistema valida los datos y muestra un mensaje de error "El campo Autor es	1-El usuario selecciona la opción "Adicionar galería". 2-El sistema



vacíos.		galería).	selecciona la opción "Adicionar Imagen" llena los campos asociados y dicha imagen se asocia a la galería.			obligatorio."	muestra el formulario con los campos asociados a este. 3-El usuario llena los campos y selecciona el botón "Guardar". 4-El sistema valida los datos y muestra un mensaje de error "El campo Autor es obligatorio." 5-El sistema no adiciona la galería.
EC 1.4 Cancelar "Adicionar galería".	Se selecciona la opción Adicionar galería.	V (El usuario llena el campo nombre de la galería).	El usuario escribe el nombre de una imagen previamente adicionada o selecciona la opción "Adicionar Imagen" llena los campos asociados y dicha imagen se asocia a la galería.	V (El campo autor es mostrado con el nombre del usuario.)	El usuario selecciona la fecha en que sube la galería.	El sistema no añade la imagen.	1-El usuario selecciona la opción "Adicionar galería". 2-El sistema muestra el formulario con los campos asociados a este. 3-El usuario llena los campos y selecciona el botón "Cancelar". 4-El sistema



							no adiciona la galería.
--	--	--	--	--	--	--	-------------------------

➤ **Eliminar galería de imágenes**

**Descripción:**

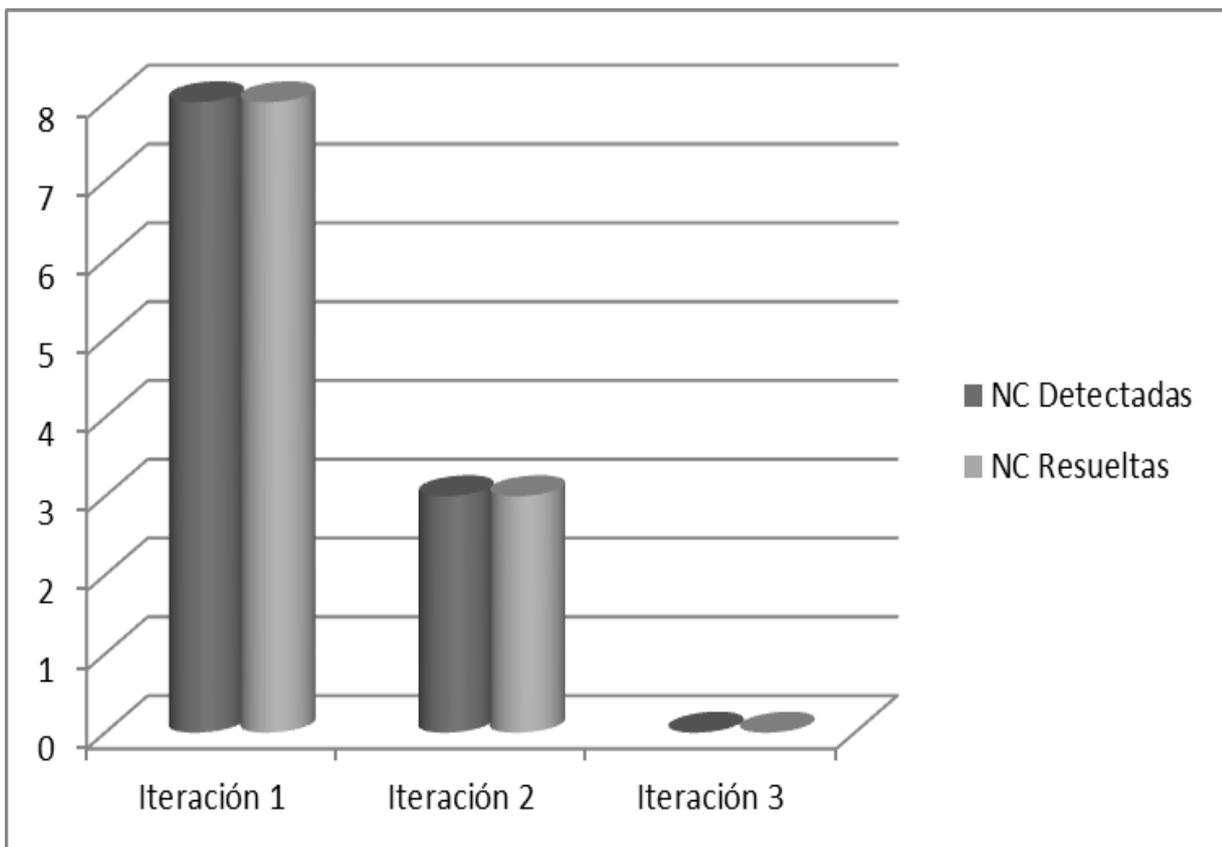
Para acceder a esta funcionalidad el usuario autenticado debe tener una galería creada. Se selecciona la opción “Eliminar”.

**Tabla 19:** Caso de Prueba "Eliminar Galería de imágenes".

Escenario	Descripción	Respuesta del sistema	Flujo Central
EC 1.1 Eliminar Galería de imágenes.	El usuario selecciona la opción “Eliminar”.	El sistema muestra un mensaje “Esta seguro que desea eliminar la galería”.	1-El usuario selecciona la opción “Aceptar” 2-El sistema elimina la galería.

**3.5.2 Resultado de las pruebas de caja negra**

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.



**Ilustración 7:** Cantidad de no conformidades por iteraciones.

En la **Ilustración 7** se puede observar cómo en la primera iteración se detectaron ocho no conformidades (seis a nivel de validación y dos errores ortográficos), número que se fue reduciendo



hasta la tercera iteración donde no se obtuvieron no conformidades. Cada una de las no conformidades detectas 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.

### **3.6 Pruebas de rendimiento**

Son pruebas dirigidas a evaluar la conformidad de un sistema o componente con requerimientos de desempeño específicos. Normalmente esto se lleva a cabo usando una herramienta de prueba automática para simular un gran número de usuarios, carga y volumen de información y para monitorear el desempeño del hardware. En este caso se decide utilizar la herramienta JMeter v 2.3.4. Entre los tipos de pruebas de rendimiento se encuentran:

#### **➤ Pruebas de carga**

Las pruebas de carga consisten en probar el funcionamiento del software bajo condiciones extremas. Estudia la especificación del software, las funciones que debe realizar, las entradas y las salidas analizando los valores límites.

#### **➤ Pruebas de estrés**

Las pruebas de estrés están diseñadas para enfrentar al programa a condiciones anormales. La prueba ejecuta un sistema de manera que demande recursos en cantidad, frecuencia o volúmenes anormales.

Es preciso realizar dichas pruebas a la aplicación ya que es necesario comprobar el rendimiento de la misma soportando la cantidad de usuarios que soliciten hacer peticiones y su comportamiento al aumentar esta carga con los mismos recursos disponibles.

Para realizar las pruebas de carga y de estrés a la aplicación se utiliza un servidor con las siguientes características: CPU: Intel Pentium 4, CPU M 350 @ 2.27GHz con memoria RAM de 1024 MB y HDD de 250 GB, además se escogen dos grupos de hilos (15 y 30) que representan la cantidad de usuarios que realizan las peticiones a la aplicación. Para 15 usuarios se realizan 1755 peticiones al servidor con un rendimiento de 11.8 segundos y para 30 usuarios se realizan 3510 peticiones al servidor con un rendimiento de 21.7 segundos. En la **Ilustración 8** se muestra la prueba realizada al sistema para un total de 15 hilos.



Actividades JMeter jue 10:57 es Addiel  
 archimagen.jmx (/home/addiel/Escritorio/archimagen.jmx) - Apache JMeter (2.3.4)

Archivo Editar Lanzar Opciones Ayuda 0 / 15

Plan de Pruebas  
 Grupo de Hilos  
 Aserción de Respuesta  
 /archimagen/  
 /archimagen/misc/jquery.ba-bbq  
 /archimagen/modules/field/ther...  
 /safebrowsing/downloads?client...  
 /archimagen/node/add/imagen  
 /archimagen/misc/throbber.gif  
 /spellcheck31/script/ssrv.cgi  
 Ver Árbol de Resultados  
 Informe Agregado  
 Banco de Trabajo

**Informe Agregado**  
 Nombre: Informe Agregado  
 Comentarios  
 Escribir todos los datos a Archivo  
 Nombre de archivo  Navegar... Log/Display Only:  Escribir en

Label	# Muestras	Media	Mediana	Linea de 90%	Mín	Máx
/archimagen/m...	15	62	64	67	53	
/archimagen/m...	15	24	25	31	13	
/archimagen/m...	15	57	55	63	52	
/archimagen/si...	15	41	41	50	28	
/archimagen/m...	15	15	15	26	8	
/archimagen/m...	15	15	16	25	4	
/archimagen/m...	15	23	25	29	11	
/archimagen/m...	15	38	36	53	26	
/archimagen/m...	15	30	30	35	23	
/archimagen/m...	15	15	14	26	1	
/archimagen/m...	15	49	47	56	41	
/archimagen/m...	15	23	16	59	6	
/archimagen/m...	15	22	22	31	13	
/archimagen/m...	15	27	26	36	22	
/archimagen/m...	15	21	21	27	15	
/archimagen/m...	15	13	10	19	1	
/archimagen/m...	15	75	74	82	57	1
/archimagen/m...	15	23	22	32	15	
/archimagen/si...	15	84	84	96	71	
/archimagen/si...	15	327	264	277	216	13
/archimagen/m...	15	287	358	385	153	3
/archimagen/si...	15	19	15	34	4	
/archimagen/si...	15	54	26	118	2	1
/archimagen/si...	15	32	9	86	2	
/archimagen/si...	15	7	6	15	1	
/archimagen/si...	15	20	17	43	2	
/archimagen/si...	15	5	4	13	1	
/spellcheck31/...	15	40050	40046	40070	40030	400
TOTAL	1755	1233	34	139	1	404

Include group name in label? Save Table Data

Ilustración 8: Prueba con 15 usuarios.

### 3.7 Conclusiones parciales

En el capítulo concluido se determinaron cuáles son los estándares de codificación a utilizar durante la implementación, lo cual permitió lograr una mayor legibilidad y limpieza del código, se define el diagrama de componentes y el diagrama de despliegue. Se describen las principales pruebas realizadas sobre la interfaz del software, indicando para cada una la respuesta de la aplicación. A través de la validación del sistema, aplicando pruebas funcionales y de rendimiento, se pudieron detectar errores y problemas para su posterior solución y permitieron monitorear el funcionamiento de la aplicación en diferentes entornos de ejecución y carga.



## Conclusiones generales

Una vez culminada la investigación se puede afirmar que los objetivos y tareas trazadas fueron cumplidos satisfactoriamente, alcanzando los siguientes resultados:

- El estudio de las tecnologías y herramientas permitió seleccionar las más idóneas para el desarrollo del nuevo sistema, estando acorde con las características del mismo.
- A partir del estudio de homólogos se identificaron las principales características y funcionalidades que contienen los sistemas que gestionan imágenes.
- La definición y la elaboración de los requisitos funcionales y no funcionales permitió obtener el diseño y 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 para la gestión de las imágenes que se utilizan en la publicación de noticias en el departamento de Comunicación Audiovisual de la Universidad de las Ciencias Informáticas.
- La utilización del sistema implementado contribuirá a una mejor gestión de imágenes para el departamento de Comunicación Audiovisual de la Universidad de las Ciencias Informáticas.
- La aplicación de pruebas permitió validar el correcto funcionamiento de la aplicación desarrollada simulando ambientes reales en diferentes entornos de ejecución.

Por todo lo expuesto anteriormente se concluye que los objetivos propuestos para el presente trabajo han sido cumplidos satisfactoriamente. La aplicación desarrollada contribuye de manera significativa al manejo de grandes volúmenes de imágenes que se utilizan en el departamento de Comunicación Audiovisual de la Universidad de las Ciencias Informáticas.

---

## Recomendaciones

A partir del estudio realizado en la presente investigación y teniendo en cuenta las ideas que surgieron durante el desarrollo del análisis y diseño de la aplicación, se realizan las siguientes recomendaciones:

- Agregarle nuevas funcionalidades de acuerdo con las expectativas que puedan surgir en el departamento de Comunicación Audiovisual de la Universidad de las Ciencias Informáticas.
- Extender el uso del sistema desarrollado hacia otras entidades.



## Referencias bibliográficas

1. INFORMÁTICAS, U. D. L. C. (2012). Misión | Portal de la Universidad de las Ciencias Informáticas Última actualización: 2012 2013/03/10/20:44:04. [Consultado el: 15 de febrero de 2013] Disponible en: <http://www.uci.cu/mision>
2. INVESTIGACIÓN, D. D. (2000). Llevando la Teoría a la Práctica. publicado el: 2000 2003 de 2000, última actualización: 2003. [Consultado el: 3 de abril de 2013] Disponible en: [www.library.cornell.edu/preservation/tutorial-spanish/tutorial\\_Spanish.pdf](http://www.library.cornell.edu/preservation/tutorial-spanish/tutorial_Spanish.pdf).
3. CODINA, L. (2007). Obtención, edición y gestión de imágenes mediante recursos de libre acceso. El Profesional de la Información, publicado el: 2007/10//Septiembre de 2007, última actualización: 2007/10//Septiembre. [Consultado el: 6 de abril de 2013] Disponible en: <http://www.lluiscodina.com/multimedia2007.pdf>.
4. PRESS, N. (2004). Understanding Metadata. National Information Standards Organization, publicado el: 2004 de 2004, última actualización: 2004. [Consultado el: 2013/03/10/19:50:33]. 20-pp. p. [Consultado el: 6 de abril de 2013] Disponible en: <http://www.niso.org/publications/press>. ISBN 1-880124-62-9.
5. PAULUS, C. V. (2007). METADATOS: Introducción e historia 20-pp. [Consultado el: 3 de abril de 2013] Disponible en: <http://users.dcc.uchile.cl/~cvasquez/introehistoria.pdf>.
6. ECURED Flickr [Consultado el: 6 de abril de 2013] Disponible en: [http://www.ecured.cu/index.php/Flickr#Beneficios\\_que\\_ofrece](http://www.ecured.cu/index.php/Flickr#Beneficios_que_ofrece).
7. Picasa Ares. (2009). Traducido por: Pinzon, D.;Navarrete, M.et al. SSlideShare Inc: publicado el: 2009 de 2009, última actualización: 2009. [Consultado el: 6 de abril de 2013]. Disponible en: <http://www.slideshare.net/mrosero1/picasa-ares>
8. UPLOAD, D. (2008). Dr. Upload, sistema de almacenamiento gratuito de imágenes de hasta 10 MB [Weblogs]. Última actualización: 2008/01/08/2013/03/10/20:15:51. [Consultado el: 6 de abril de 2013] Disponible en: <http://www.genbeta.com/web/dr-upload-sistema-de-almacenamiento-gratuito-de-imagenes-de-hasta-10-mb>
9. TELLADO, F. (2008). Arkiva – Comparte, almacena y más | Incubaweb [red de blogs]. Última actualización: 2008/07/14/2013/03/10/20:29:37. [Consultado el: 6 de abril de 2013] Disponible en: <http://www.incubaweb.com/arkiva-comparte-almacena-y-mas/>
10. LEYVA, Y. D. L. C. (2011). Sistema de Gestión Documental. Módulo Fototeca para las Fotografías de la Prensa. Universidad de las Ciencias Informáticas, 2011.



11. JAVIER TUYA, I. R. R., JOSÉ JAVIER DOLADO COSÍN (2009). Técnicas Cuantitativas para la Gestión en la Ingeniería del Software. 2009. 373 p. ISBN 978-84-9745-204-5.
12. Modelado de Sistemas con UML Última actualización: 2013/03/10/20:34:01. (Popkin Software and Systems). [Consultado el: 6 de abril de 2013] Disponible en: <http://es.tldp.org/Tutoriales/doc-modelado-sistemas-UML/multiple-html/>
13. Visual Paradigm for UML (ME) - (Paradigma Visual para UML (ME)) (Visual Paradigm for UML (ME)) por Visual Paradigm International Ltd. - reporte y descarga. [Sitio de descarga libre de software]. Última actualización: 2007/03/05/de 2013/05/07/14:55:25. Disponible en: [http://www.freedownloadmanager.org/es/downloads/Paradigma\\_Visual\\_para\\_UML\\_\(M%C3%8D\)\\_14720\\_p/files/223/Paradigma\\_Visual\\_para\\_UML\\_\(MÍ\)\\_14720\\_p.html](http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_(M%C3%8D)_14720_p/files/223/Paradigma_Visual_para_UML_(MÍ)_14720_p.html).
14. Gestores de Contenidos o CMSs | fcsites™ diseño web Sevilla. Última actualización: 2013/03/10/20:35:54. [Consultado el: 6 de abril de 2013] Disponible en: <http://www.fcsites.com/servicios/disenio-profesional-de-paginas-web/gestores-de-contenidos-o-cms/>
15. About Drupal | drupal.org. Última actualización: 2013/03/10/20:37:25. [Consultado el: 13 de abril de 2013] Disponible en: <http://drupal.org/about>
16. GAUCHAT, J. D. (2012). El gran libro de HTML5, CSS3 y JavaScript. 2012. 354 p. ISBN 978-84-267-1770-2.
17. (ASF), T. A. S. F. (2010). Apache HTTP Server 2. 2 2010. vol. IV, ISBN 978-1-59682-194-1.
18. YERA, A. C. Diseño y programación de bases de datos. 116 p. ISBN 978-84-9821-459-8.
19. FEATURES, N. I. NetBeans IDE - The Smarter and Faster Way to Code. [Consultado el: 3 de mayo de 2013] Disponible en: <https://netbeans.org/features/index.html>.
20. UCI, E. [Consultado el: 3 de mayo de 2013] Conferencia #2 Fase de Inicio, Modelo del Negocio. Disponible en: [http://www.ecured.cu/index.php/Modelo\\_de\\_dominio](http://www.ecured.cu/index.php/Modelo_de_dominio).
21. SLIDESHARE (2012). Modelo de datos. 2012, [Consultado el: 13 de abril de 2013] Disponible en: <http://www.slideshare.net/csalazarc/modelo-de-datos-14506949>.
22. MARTÍN, L. M. (2010). [Consultado el: 13 de abril de 2013] POO y patrones de diseño en DRUPAL Disponible en: <http://jitcode.blogspot.com/2010/06/poo-y-patrones-de-diseno-en-drupal.html>.
23. HURTADO, B. H. (2010). DESARROLLO WEB CON DRUPAL. Tutor: Gortázar, F. INGENIERÍA TÉCNICA EN INFORMÁTICA DE SISTEMAS. Universidad Rey Juan Carlos, 2010.



24. GÓMEZ, A. (2012). *Experto en Drupal 7*. 2012. 507 p. ISBN 978-84-939410-5-5.
25. ALVA, E. R. (2009). *Arquitectura de Software II*. 2009, 24 p. [Consultado el: 13 de abril de 2013]  
Disponible en: <http://es.scribd.com/doc/7884665/Arquitectura-de-Software-II-Diagrama-de-Componentes-y-Despliegue>.
26. FALGUERAS, B. C. (2003). *Ingeniería de software*. 2003. 314 p. ISBN 84-8318-997-6.
27. Pressman. 2002. *Ingeniería de software un enfoque práctico*. Quinta edición. 2002.
28. VERDECIA, H. G. (2012). *Proyectos informáticos* [Consultado el: 21 de mayo de 2013].  
Disponible en: <http://www.monografias.com/trabajos75/proyectos-nformaticos/proyectos-informaticos3.shtml>.

## Bibliografía consultada

1. APLICACIONES (2013). eCommerce con Drupal 7 – III Arquitectura de Drupal 7 [Consultado el: 6 de abril de 2013] Disponible en: <http://aplicacionez.com/e-commerce-con-drupal-7-iii-arquitectura-de-drupal-7/#top>
2. DAPENA, M. D. D. Definición del modelo del negocio y del dominio utilizando Razonamiento Basado en Casos. [Consultado el: 12 de abril de 2013] Disponible en: <http://www.inf.udec.cl/~revista/ediciones/edicion8/Rbc.pdf>
3. INTERNETHOY *Arquitectura de Drupal 7* [Consultado el: 3 de mayo de 2013] Disponible en: <http://internethoy.com/manuales/drupal-7/1584-arquitectura-de-drupal-7.html>.
4. JIMÉNEZ, S. A. L. *Definición de modelo de datos*. [Consultado el: 12 de abril de 2013] Disponible en: [http://itslr-alelopi.weebly.com/uploads/9/3/6/4/936494/fbd-tutorial\\_u2.pdf](http://itslr-alelopi.weebly.com/uploads/9/3/6/4/936494/fbd-tutorial_u2.pdf).
5. UNIVERSITY, C. M. (2006). CMMI® for Development, Version 1.2 2006, nº p. 573 pp. [Consultado el: 12 de mayo de 2013] Disponible en: [http://eva.uci.cu/file.php/174/CLASES/Clases\\_2009-2010/Semana\\_1/Act1\\_-\\_C1/Materiales\\_Complementarios/CMMI\\_v\\_1.2\\_Dev.pdf](http://eva.uci.cu/file.php/174/CLASES/Clases_2009-2010/Semana_1/Act1_-_C1/Materiales_Complementarios/CMMI_v_1.2_Dev.pdf).
6. CHORNET, V. G. (2012). DATOS DESCRIPTIVOS AÑADIDOS A LA IMAGEN DIGITAL. 2012, nº p. 7. [Consultado el: 12 de mayo de 2013] Disponible en: <http://dglab.cult.gva.es/ArxiuRegne/documents/Datosdescriptivos.pdf>.
7. IPTC (2009). IPTC Standard Photo Metadata. 2009, nº p. 52. [Consultado el: 12 de mayo de 2013] Disponible en: [www.iptc.org](http://www.iptc.org).
8. ARCHIVES, I. C. O. (2000). *STANDARDS ICA*. 2000. ISBN 0-9696035-5-X.
9. TESIC, J. (2005 ). *Metadata Practices for Consumer Photos*. 2005 nº p. 86-92. [Consultado el: 12 de mayo de 2013] Disponible en: <http://dl.acm.org/citation.cfm?id=1079952>.
10. *lluís codina: Análisis y Métodos en Información y Documentación Digital* [Consultado el: 6 de abril de 2013] Última actualización: 2013/03/10/19:34:10. (Análisis y Métodos en Información y Documentación Digital). [Consultado el: 12 de mayo de 2013] Disponible en: <http://www.lluiscodina.com/>
11. *PHOTOBOOK Factory; Que es PhotoBook* Última actualización: 2013/01/31/13:26:31. [Consultado el: 12 de mayo de 2013]. Disponible en: <http://photobookfactory.pe/photobook.php>
12. PHP (2013). PHP Manual (exif\_read\_data) [Consultado el: 13 de mayo de 2013]. Disponible en: <http://php.net/manual/es/function.exif-read-data.php>.



13. MÁRQUEZ, A. (2008). *Elementos de la imagen* Última actualización: 2008/03/30 (slideshare). [Consultado el: 12 de mayo de 2013]. Disponible en: <http://www.slideshare.net/alfmaba/elementos-de-la-imagen>
14. *Reglas de Extreme Programming* (2009). Sitio oficial de Extreme Programming. [Consultado el: 12 de mayo de 2013]. Disponible en: <http://www.extremeprogramming.org/rules.html>.
15. Sitio oficial de Extreme Programming (2009). [Consultado el: 17 de enero de 2013]. Disponible en: <http://www.extremeprogramming.org/>.
16. WelcometoNetbeans (2011). Sitio oficial del IDE Netbeans: [Consultado el: 13 de abril de 2013]. Disponible en: <http://netbeans.org/>.
17. ÁGILES, P. Qué es SCRUM [Consultado el: 13 de mayo de 2013]. Disponible en: <http://www.proyectosagiles.org/que-es-scrum>.
18. GÓMEZ, A. (2012). Experto en Drupal 7. 2012, nº p. 507. [Consultado el: 8 de febrero de 2013]. Disponible en: <http://drupaleros.uci.cu/>.



## Glosario de términos

**API:** Interfaz de Programación de Aplicaciones.

**CPU:** Central Processing Unit (unidad de proceso central), es el cerebro del ordenador y donde se producen la mayoría de los cálculos. En términos de potencia del ordenador, la CPU es el elemento más importante de un sistema informático.

**CASE:** Ingeniería de Software Asistida por Ordenador.

**CMMI:** Capability Maturity Model Integration (Integración de modelos de madurez de capacidades).

**GIF:** Graphics Interchange Format (Formato de Compresión de Imagen): es un formato gráfico utilizado ampliamente en la World Wide Web, tanto para imágenes como para animaciones.

**IDE:** Entorno de Desarrollo Integrado.

**Internautas:** persona que navega en Internet visitando páginas web y, por extensión, a cualquier persona que haciendo uso de una aplicación en una computadora obtiene información de Internet, o interactúa con otras personas: correo electrónico, compartir archivos, discusiones en foros, Facebook entre otros.

**Módulo:** Bloques de código que proveen funcionalidad extra o mejoras.

**JPG:** Joint Photographic Experts Group (Grupo Conjunto de Expertos en Fotografía) es el nombre de un comité de expertos que creó un estándar de compresión y codificación de archivos de imágenes fijas.

**PostgreSQL:** es un Sistema de Base de Datos Relacionales Orientadas a Objetos.

**PNG:** Portable Network Graphics (Gráficos de Red Portátiles): Formato desarrollado en buena parte para solventar las deficiencias del formato GIF y permite almacenar imágenes con una mayor profundidad de contraste.

**PSD:** Photoshop Document: es el formato original de Photoshop.

**SGBD:** Sistema Gestor de Base de Datos.

**Software:** es un programa o aplicación programada para realizar tareas específicas.

**Scrum:** Metodología ágil de desarrollo de software.

**Scripts:** es un guión o conjunto de instrucciones. Permiten la automatización de tareas creando pequeñas utilidades. Es muy utilizado para la administración de sistemas UNIX. Son ejecutados por un intérprete de línea de comandos y usualmente son archivos de texto.

**UML:** Lenguaje Unificado de Modelado.



---

**URL:** Uniform Resource Locator: protocolo para especificar direcciones en Internet.

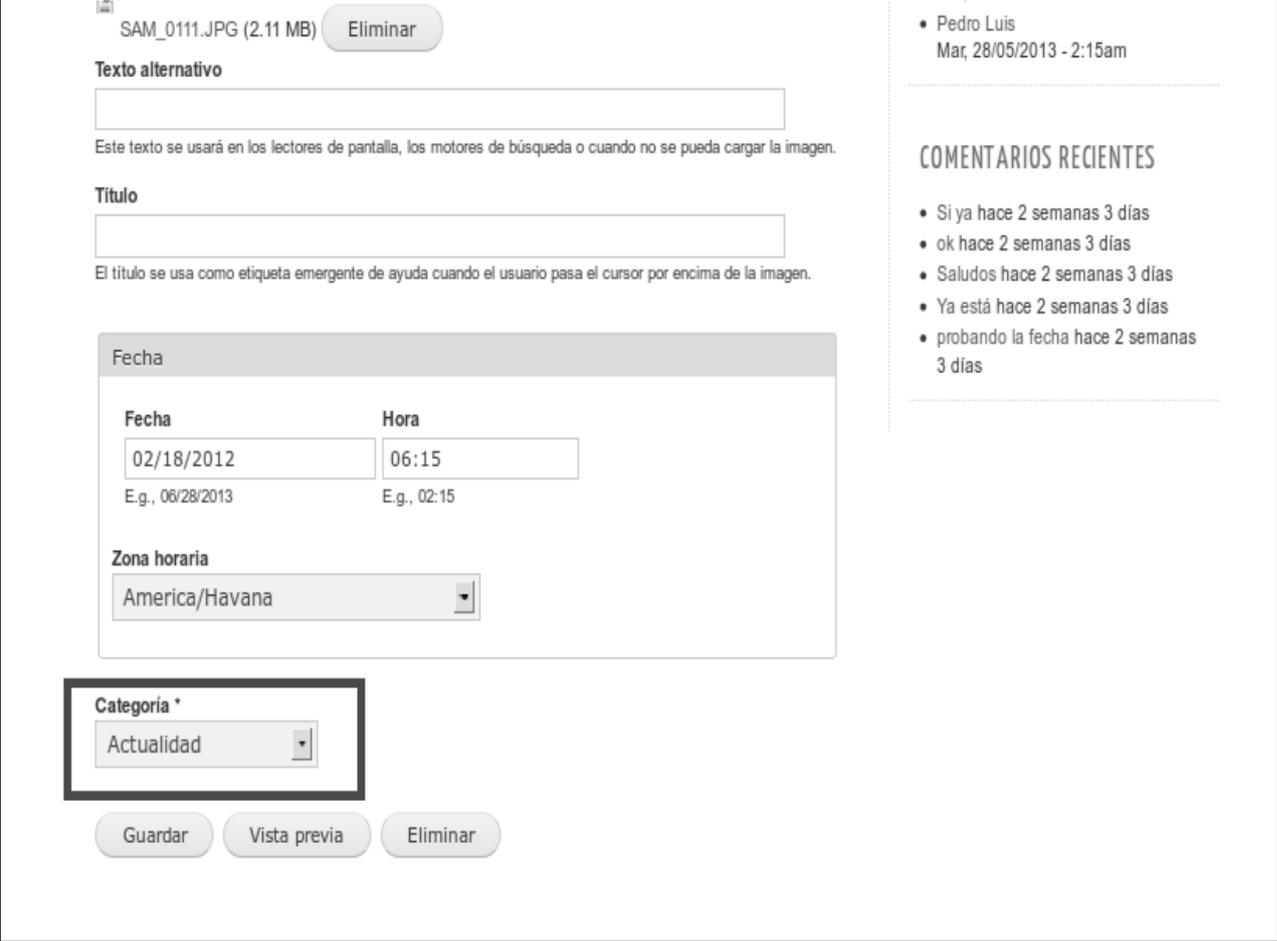
**XP:** Extreme Programming. Metodología ágil de desarrollo de software.



## Anexos

### Anexo #1: Descripción de los requisitos funcionales.

**Tabla 20:** Descripción del requisito funcional adicionar categorías a las imágenes.

Nº	Nombre	Descripción	Complejidad	Prioridad para el cliente
RF 18	Adicionar categorías a las imágenes.	El usuario selecciona la categoría asociada a la imagen.	Baja	Alta
<b>Prototipo</b>				
				
	<b>Campos</b>	<b>Tipos de Datos</b>	<b>Reglas o Restricciones</b>	
			➤ Para adicionar categorías a las imágenes el usuario tiene que estar autenticado en el sistema.	
	<b>Observaciones</b>			

**Tabla 21:** Descripción del requisito funcional editar imagen.

Nº	Nombre	Descripción	Complejidad	Prioridad para el cliente



RF 7	Editar imagen	El usuario selecciona la imagen que desea editar y selecciona la opción "Editar".	Baja	Alta
<b>Prototipo</b>				
				
<b>Campos</b>	<b>Tipos de Datos</b>	<b>Reglas o Restricciones</b>		
		<ul style="list-style-type: none"> <li>➤ Para editar la imagen el usuario tiene que estar autenticado en el sistema.</li> <li>➤ El usuario debe de tener creada al menos una imagen.</li> <li>➤ El usuario debe tener permisos de edición en el sistema.</li> </ul>		
<b>Observaciones</b>				

**Tabla 22:** Descripción del requisito funcional editar galería de imágenes.

Nº	Nombre	Descripción	Complejidad	Prioridad para el cliente
RF 11	Editar galería de imágenes.	El usuario selecciona la galería de imágenes que desea editar y selecciona la opción "Editar".	Baja	Alta
<b>Prototipo</b>				



UCI Universidad de las Ciencias Informáticas

Inicio Galerías Registro Contáctenos

Ver **Editar**

## Editar Galería Galería de Addiel

Nombre \*

Galería de Addiel

Mostrar pesos de la fila

Imagen	
+	addiel1 [nid:215] <input type="text"/>
+	+ Adicionar Imagen
+	addiel2 [nid:216] <input type="text"/>
+	+ Adicionar Imagen

¿QUÉ DESEAS HACER?

- Adicionar galería
- Adicionar imagen
- Ver banco de imágenes

USUARIOS CONECTADOS

Hay actualmente 1 usuario conectado.

- Addiel Alejandro

Campos	Tipos de Datos	Reglas o Restricciones
		<ul style="list-style-type: none"> <li>➤ Para editar la galería el usuario tiene que estar autenticado en el sistema.</li> <li>➤ El usuario debe de tener creada al menos una galería.</li> <li>➤ El usuario debe tener permisos de edición en el sistema.</li> </ul>
<b>Observaciones</b>		

Tabla 23: Descripción del requisito funcional mostrar imagen.

Nº	Nombre	Descripción	Complejidad	Prioridad para el cliente
RF 8	Mostrar imagen.	El usuario selecciona la imagen que desea ver.	Baja	Alta
<b>Prototipo</b>				



## Paisaje



Enviado por Javier el Mar, 21/05/2013 - 9:28pm

**Foto:**



**Fecha:**

Viernes, Diciembre 7, 2012 - 2:25am

**Autor:**

Javier

Categoría:

Variadas

Campos	Tipos de Datos	Reglas o Restricciones
		<ul style="list-style-type: none"> <li>➤ Para ver la imagen el usuario tiene que estar autenticado en el sistema.</li> <li>➤ El usuario debe tener permisos para ver el contenido.</li> </ul>
Observaciones		

**Tabla 24:** Descripción del requisito funcional mostrar galería de imágenes.

Nº	Nombre	Descripción	Complejidad	Prioridad para el cliente
RF 12	Mostrar galería de imágenes.	El usuario selecciona la galería que desea ver.	Baja	Alta
Prototipo				



## Galeria de Addiel

 Enviado por Addiel Alejandro el Mar, 07/05/2013 - 1:43pm

**Fecha:**

Martes, Mayo 7, 2013 - 1:45pm

**Imagen:**

addiel1

 Enviado por Addiel Alejandro el Mar, 07/05/2013 - 1:41pm

**Foto:**



**Fecha:**

Sábado, Febrero 18, 2012 - 6:53am

**Autor:**

Addiel Alejandro

Categoría:

Cuba

Campos	Tipos de Datos	Reglas o Restricciones
		<ul style="list-style-type: none"> <li>➤ Para ver la galería el usuario tiene que estar autenticado en el sistema.</li> <li>➤ El usuario debe tener permisos para ver el contenido.</li> </ul>
Observaciones		

**Tabla 25:** Descripción del requisito funcional eliminar galería de imágenes.

Nº	Nombre	Descripción	Complejidad	Prioridad para el cliente
RF 10	Eliminar galería de imágenes.	El usuario selecciona la galería que desea eliminar.	Baja	Alta
Prototipo				



### Editar Galería Galería JJ

Nombre \*

Galería JJ

Mostrar pesos de la fila

Imagen

pedrito [nid:241]

+ Adicionar Imagen

+ Adicionar Imagen

Añadir otro elemento

Autor \*

Nombre: Pedro Luis

Fecha

Fecha	Hora
<input type="text" value="05/28/2013"/>	<input type="text" value="02:15"/>
E.g., 06/28/2013	E.g., 02:15

Guardar    Vista previa    **Eliminar**

Campos	Tipos de Datos	Reglas o Restricciones
		<ul style="list-style-type: none"> <li>➤ Para eliminar la galería el usuario tiene que estar autenticado en el sistema.</li> <li>➤ El usuario debe de tener creada al menos una galería.</li> <li>➤ El usuario debe tener permisos de edición en el sistema.</li> </ul>
<b>Observaciones</b>		

Tabla 26: Descripción del requisito funcional eliminar imagen.

Nº	Nombre	Descripción	Complejidad	Prioridad para el cliente
RF 6	Eliminar imagen.	El usuario selecciona la imagen que desea eliminar.	Baja	Alta
<b>Prototipo</b>				



 SAM\_0111.JPG (2.11 MB) Eliminar

**Texto alternativo**

Este texto se usará en los lectores de pantalla, los motores de búsqueda o cuando no se pueda cargar la imagen.

**Título**

El título se usa como etiqueta emergente de ayuda cuando el usuario pasa el cursor por encima de la imagen.

**Fecha**

Fecha	Hora
<input type="text" value="02/18/2012"/>	<input type="text" value="06:15"/>
<small>E.g., 06/28/2013</small>	<small>E.g., 02:15</small>

**Zona horaria**

**Categoría \***

Guardar Vista previa Eliminar

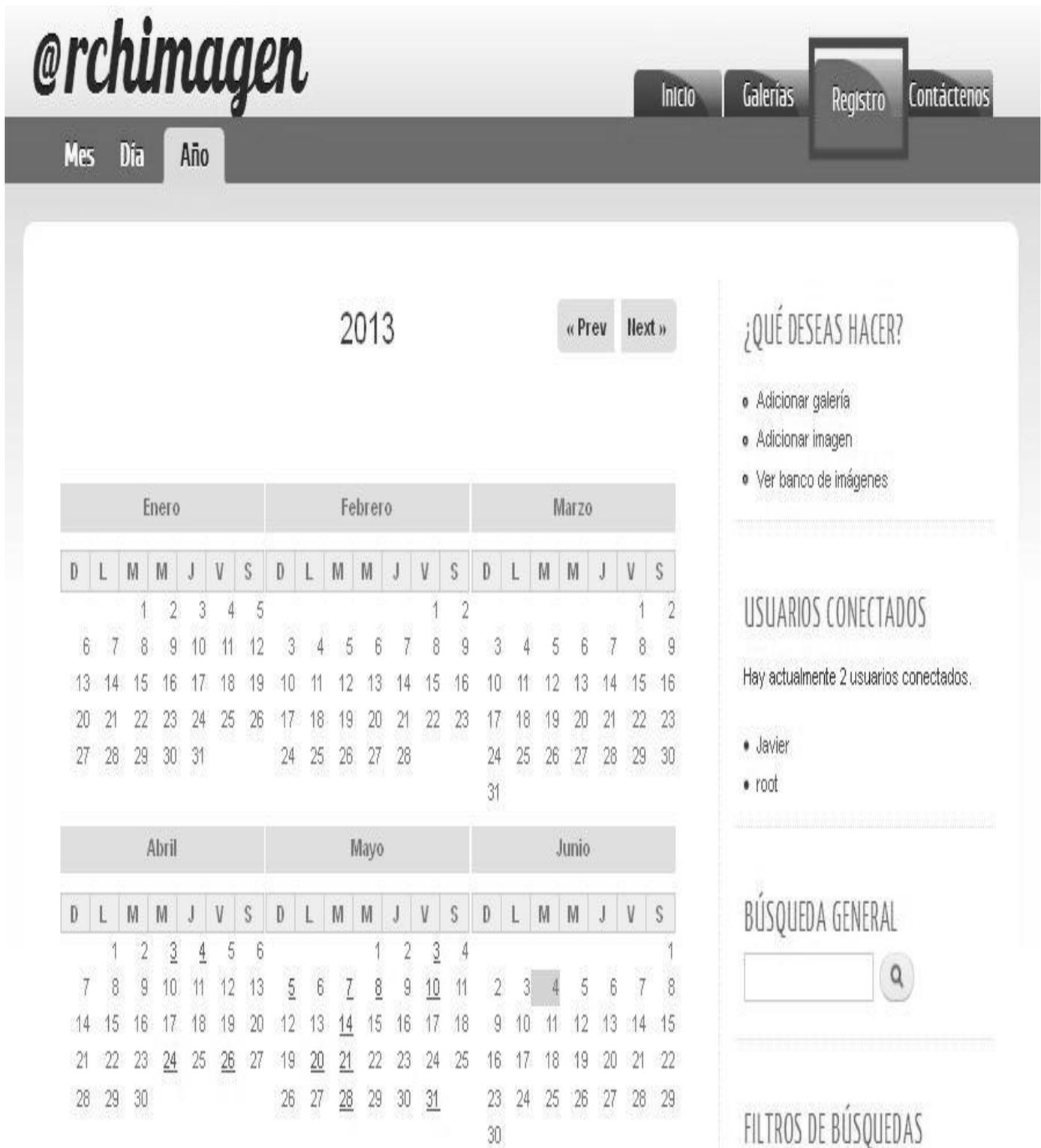
- Pedro Luis  
Mar, 28/05/2013 - 2:15am

**COMENTARIOS RECIENTES**

- Si ya hace 2 semanas 3 días
- ok hace 2 semanas 3 días
- Saludos hace 2 semanas 3 días
- Ya está hace 2 semanas 3 días
- probando la fecha hace 2 semanas 3 días

Campos	Tipos de Datos	Reglas o Restricciones
		<ul style="list-style-type: none"> <li>➤ Para eliminar la imagen el usuario tiene que estar autenticado en el sistema.</li> <li>➤ El usuario debe de tener almacenada al menos una imagen.</li> <li>➤ El usuario debe tener permisos de edición en el sistema.</li> </ul>
<b>Observaciones</b>		

Anexo #2: Representación del registro por días.



The screenshot shows the @rchimagen website interface. At the top, there is a navigation bar with buttons for 'Inicio', 'Galerías', 'Registro', and 'Contáctenos'. Below this is a header with 'Mes', 'Día', and 'Año' filters. The main content area displays a calendar for the year 2013, with months from January to June. The calendar grid shows days of the week (D, L, M, M, J, V, S) and dates. The 'Registro' button is highlighted in the top navigation. On the right side, there is a sidebar with the following sections:

- ¿QUÉ DESEAS HACER?**
  - Adicionar galería
  - Adicionar imagen
  - Ver banco de imágenes
- USUARIOS CONECTADOS**

Hay actualmente 2 usuarios conectados.

  - Javier
  - root
- BÚSQUEDA GENERAL**
 
- FILTROS DE BÚSQUEDAS**

Ilustración 9: Registro de acciones.

Anexo #3: Representación del menú de contacto.



## Contáctenos

Su nombre \*

Su dirección de correo electrónico \*

Asunto \*

Mensaje \*

Enviar una copia.

Enviar mensaje

### ¿QUÉ DESEAS HACER?

- Adicionar galería
- Adicionar imagen
- Ver banco de imágenes

### USUARIOS CONECTADOS

Hay actualmente 2 usuarios conectados.

- Javier
- root

### BÚSQUEDA GENERAL

### FILTROS DE BÚSQUEDAS

- Galería
- Categoría
- Autor

Ilustración 10: Menú "Contáctenos".

**Anexo #4:** Representación del banco de imágenes.

### Banco de imágenes

**Título**

**Buscar términos**

**Categoría**

- Cualquiera - ▼

Filtrar
Reiniciar

<p><b>Addiel</b></p> <p><small>Enviado por Addiel Alejandro el Dom, 05/05/2013 - 8:29pm</small></p> <p><b>Foto:</b></p>  <p><b>Fecha:</b> Domingo, Mayo 5, 2013 - 8:30pm</p> <p><b>Autor:</b> Addiel Alejandro</p> <p><b>Categoría:</b> Actualidad</p>	<p><b>addiel1</b></p> <p><small>Enviado por Addiel Alejandro el Mar, 07/05/2013 - 1:41pm</small></p> <p><b>Foto:</b></p>  <p><b>Fecha:</b> Sábado, Febrero 18, 2012 - 6:53am</p> <p><b>Autor:</b> Addiel Alejandro</p> <p><b>Categoría:</b> Cuba</p>	<p><b>addiel2</b></p> <p><small>Enviado por Addiel Alejandro el Mar, 07/05/2013 - 1:42pm</small></p> <p><b>Foto:</b></p>  <p><b>Fecha:</b> Sábado, Febrero 18, 2012 - 6:53am</p> <p><b>Autor:</b> Addiel Alejandro</p> <p><b>Categoría:</b> Cuba</p>
<p><b>ariel</b></p> <p><small>Enviado por Ariel el Lun, 20/05/2013 - 3:30pm</small></p> <p><b>Foto:</b></p>  <p><b>Fecha:</b> Domingo, Marzo 29, 2009 - 6:21am</p> <p><b>Autor:</b> Ariel</p> <p><b>Categoría:</b> Del Mundo</p> <p><b>Descripción:</b> Esta es una imagen de prueba</p>	<p><b>asdasda</b></p> <p><small>Enviado por admin el Mié, 08/05/2013 - 2:18pm</small></p> <p><b>Foto:</b></p>  <p><b>Fecha:</b> Miércoles, Diciembre 31, 1969 - 2:00am</p> <p><b>Autor:</b> admin</p> <p><b>Categoría:</b> Cuba</p>	<p><b>benq</b></p> <p><small>Enviado por admin el Mié, 08/05/2013 - 2:31pm</small></p> <p><b>Foto:</b></p>  <p><b>Fecha:</b> Viernes, Julio 6, 2012 - 9:47pm</p> <p><b>Autor:</b> admin</p> <p><b>Categoría:</b> Cultura</p>

**¿QUÉ DESEAS HACER?**

- Adicionar galería
- Adicionar imagen
- **Ver banco de imágenes**

---

**USUARIOS CONECTADOS**

Hay actualmente 1 usuario conectado.

- Pedro Luis

Área de filtros

**BUSQUEDA GENERAL**

 Q


---

**FILTROS DE BÚSQUEDAS**

- Galería
- Categoría
- Autor

1 2 3 4 5 siguiente › última »

**Ilustración 11:** Ver banco de imágenes.