



UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

UCI



PLATAFORMA PARA LA GESTIÓN DEL PROYECTO PORTALES_PHP

Trabajo de Diploma

Autores:

Zuyen Carralero Morales

Ariamna Trujillo Puerto

Tutor:

Ing. Maikel Manuel Fernández Fernández

Ciudad de La Habana

2007

Declaración de Autoría

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

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

Ariamna Trujillo Puerto

Firma del Autor

Maikel Manuel Fernández Fernández

Firma del Tutor

Zuyen Carralero Morales

Firma del Autor

Datos de Contacto

Tutor: Ing. Maikel Manuel Fernández.Fernández (maikelm@uci.cu)

Profesor graduado de Ingeniero en Informática en el Instituto Superior Politécnico José Antonio Echeverría. Ha impartido clases de las diferentes Matemáticas. Es profesor de la facultad 10 y en la actualidad se desempeña como jefe de proyecto dentro del proyecto "Portales PHP". Se especializa en las líneas de investigación de dicho proyecto, Desarrollo Web y Sistemas de Gestión de Contenidos.

“El secreto del éxito es la constancia en el propósito”

Benjamín Disraeli

Agradecimientos

...A la Revolución y a la Universidad de las Ciencias Informáticas por darnos la oportunidad de formarnos como profesionales.

...Al tutor de este trabajo, el Ing. Maikel Manuel Fernández por sus orientaciones, su ayuda constante y su preocupación.

...A nuestros padres, por el apoyo y la confianza que siempre han depositado en nosotras, de verdad muchísimas gracias.

...A nuestros hermanos y hermanas, por todo el cariño que nos dan siempre.

...A toda la familia, tíos, primos, abuelos, a los familiares presentes y no presentes, por su apoyo.

...A las amistades con las que hemos compartido estos intensos cinco años de nuestras vidas por su compañerismo, alegría y ayuda, por compartir tantos momentos.

...A nuestros profesores de la universidad por todo lo que nos enseñaron y que además de educarnos y formarnos como profesionales, han sido una gran familia.

...A Alexeis Companioni Guerra por ser tan especial, por su apoyo incondicional y por complacerme todos estos años.

...A mis tías Edita y Lidia, por estar a mi lado durante estos 5 años apoyándome en todo.

...A Héctor, Olguita y Delly, muchas gracias por dejarme encontrar a otros padres más acá en la uci, por su apoyo, sus consejos y atenciones.

Zuyen Carralero Morales

... A mis vecinos y amistades cercanas de mi pueblo, por su ánimo, apoyo y preocupación, por los momentos buenos que compartimos.

...En general a todos los que por razón de espacio no he mencionado pero han contribuido a la realización de esta tesis, o han dejado una huella en mi vida, mis agradecimientos y mi gratitud.

Ariamna Trujillo Puerto

Dedicatoria

Dedico este trabajo a mis padres, a mis abuelas y en especial a mi abuelito Manolo que la vida no le dio la oportunidad de verme graduada.

Zuyén Carralero

A quienes me enseñaron las cosas más importantes de la vida, mis padres, a mi hermano por las conversaciones y los momentos vividos, a mis abuelos maternos por el cariño que en vida me brindaron.

Ariamna Trujillo Puerto

Resumen

La Universidad de las Ciencias Informáticas como uno de los principales centros educacionales del país juega un papel fundamental en el desarrollo de software en Cuba. Entre los proyectos que se encuentran laborando actualmente está Portales_PHP, que es el encargado de realizar sitios Web, ya sea tanto a nivel nacional como internacional. Hoy día es muy complicado controlar todos los portales que se desarrollan en dicho proyecto. El trabajo relacionado con el control de información se realiza de forma manual, lo que implica que la obtención de reportes no esté en buen tiempo ni de forma organizada. No existe una vía para la divulgación de las actividades y la distribución de tareas para las personas que trabajan directamente con cada proyecto. Así como también se desconoce como va el desarrollo de cada sitio, los que trabajan en él, entre otras cuestiones que son necesarias a la vista del usuario. Así surge la Plataforma para la Gestión del Proyecto Portales_PHP, y tiene como objetivo principal automatizar el control de información relacionada con el trabajo de dicho proyecto. Además debe convertirse en un medio importante para la divulgación de todas las actividades que se llevan a cabo. Por lo que con la puesta en marcha de este sistema se brindará una respuesta más rápida y eficiente a las necesidades que actualmente existen.

<i>Agradecimientos</i>	5
<i>Dedicatoria</i>	6
<i>Resumen</i>	7
<i>Introducción</i>	1
Capítulo I. FUNDAMENTACIÓN TEÓRICA	5
1.1 Introducción	5
1.2 Sistema de Gestión de Contenidos (CMS).....	5
1.2.1 ¿Por que el nombre de Sistema de Gestión de Contenidos?	5
1.2.2 Historia y surgimiento de los CMS.....	6
1.2.3 Funciones de los sistemas de gestión de contenidos.....	7
1.2.4 Tipos de CMS	9
1.2.5 Presente de los CMS	10
1.2.6 Futuro de los CMS	10
1.2.7 Necesidad de un CMS.....	11
1.2.8 Creación de Contenidos	12
1.2.9 Gestión de contenido.....	13
1.2.10 Publicación.....	13
1.2.11 Presentación	14
1.2.12 CMS comerciales y de código abierto	14
1.3 Drupal.....	16
1.3.1 Distribuciones de Drupal	16
1.3.2 Licencia de Drupal	17
1.3.3 Características de Drupal	17
1.4 Soluciones de Código Abierto (Open Source).....	20
1.4.1 Servidor Web. Apache	21
1.4.2 Servidor de Bases de Datos. MySQL.....	23
1.4.3 Lenguaje de Programación. PHP	25
1.5 Metodología de Ingeniería de Software empleada.....	27
1.5.1 Proceso Unificado de Desarrollo (RUP).....	27
1.5.2 Lenguaje Unificado de Modelación (UML)	28
1.6 Herramienta para el Modelado (Visual Paradigm)	28
1.7 Conclusiones	29
Capítulo II. CARACTERÍSTICAS DEL SISTEMA	30
2.1 Introducción	30
2.2 Objeto de estudio	30
2.3 Objeto de automatización.....	31
2.4 Información que se maneja	32
2.5 Propuesta de sistema	33
2.6 Modelo de Negocio.....	33
2.6.1 Descripción de los procesos del negocio propuestos	34
2.6.2 Descripción de los actores y trabajadores del negocio	35

2.6.3 Diagrama de Casos de Uso del Negocio.....	35
2.6.4 Descripción de los casos de uso del negocio	36
2.6.5 Diagrama de clases del modelo de objetos.	42
2.7 Especificación de los requisitos de software.....	43
2.7.1 Requisitos funcionales	43
2.7.2 Requisitos no funcionales	45
2.8 Definición de los Casos de Uso del Sistema.....	47
2.8.1 Definición de los actores del sistema.....	47
2.8.2 Casos de uso del sistema.....	47
2.8.3 Diagrama de casos de uso del sistema	52
2.8.4- Descripción de los casos de uso del sistema.....	52
2.9 Conclusiones	69
Capítulo III. ANALISIS Y DISEÑO DEL SISTEMA.....	70
3.1 Introducción	70
3.2 Análisis	70
3.2.1 Diagramas de clases de análisis	70
3.2.2 Diagramas de interacción.....	76
3.3 Diseño	84
3.3.1 Diagrama de clases Web.....	84
3.3.2 Diseño de la Base de Datos.....	86
3.4 Definiciones de diseño que se apliquen	93
3.5 Tratamiento de errores	94
3.6 Seguridad	95
3.7 Interfaz.....	96
3.8 Concepción de la ayuda.	96
3.9 Conclusiones	97
Capítulo IV. IMPLEMENTACION Y PRUEBA.....	98
4.1 Introducción	98
4.2 Implementación.....	98
4.2.1 Diagrama de Despliegue	98
4.2.2 Diagrama de Componentes	99
4.3 Modelo de prueba.....	100
4.3.1 Descripción de los casos de prueba de integración.....	100
4.4 Conclusiones	105
Conclusiones	106
Recomendaciones	107
Bibliografía.....	108
Referencias Bibliográficas	109
Glosario de Términos	111

Introducción

La UCI es el centro de enseñanza superior más joven de la Isla. Concebida como universidad de nuevo tipo con el principal objetivo de formar profesionales con un indeclinable y firme compromiso con la Patria y la Revolución. Fue creada para dar soporte a la informatización del país, a la producción de software y a los servicios industriales.

La universidad, básicamente tiene tres líneas de producción: lograr la informatización del país, desarrollar los proyectos que indica la dirección del mismo y los grandes proyectos de exportación que el país tiene.

Está organizada en 10 facultades. Cada facultad está orientada a distintas líneas de trabajo, creando productos de suma utilidad para el desarrollo económico, político y social. Los proyectos son coordinados, gestionados y ejecutados desde cada una de las facultades de forma independiente y en cada uno existe una estructura capaz de cumplir todas las tareas que el proyecto genera. Los estudiantes se especializan en perfiles como las telecomunicaciones y la seguridad informática, los software de gestión, libres y educativos, realidad virtual, control automático, bioinformática, gestión empresarial, informática para la salud, modelación matemática, multimedia y portales.

En cuanto a la creación de portales existen varios grupos que los desarrollan, entre estos se encuentra el Grupo de Desarrollo del proyecto "Portales_PHP" perteneciente a la facultad 10 de dicha institución, facultad que trabaja sobre la base del Software Libre. El mismo está inmerso en la tarea de atender todo lo concerniente a la realización, gestión y orientación de Aplicaciones Web, ya sea a nivel universitario, nacional o internacional.

La línea de trabajo de este grupo se fundamenta en el empleo del Sistema de Gestión de Contenidos "Drupal", utilizando el lenguaje de desarrollo "PHP", interrelacionado con el gestor de base de datos "MySQL" y con servidor web "Apache".

El Proyecto Portales_PHP surge en el curso 2005-2006. Ha desarrollado proyectos en convenio con la Republica Bolivariana de Venezuela, en diferentes instituciones del país, y en la misma universidad. Entre los portales desarrollados se puede mencionar: el Portal del Sistema Autónomo de Registro y Notarías (SAREN), el Portal del Sistema Autónomo de Migración y Extranjería (SAIME), Brigadas de Instructores de Arte José Martí, el Portal del Centro de Estudios Martianos, la Intranet de la PNR, la Revista Patria Grande, el Portal RINDE, el Sitio de la Convención Informática 2007, el Sitio UCI. En la mayoría de estos proyectos se emplea para su desarrollo el Sistema de Gestión de Contenidos "Drupal". Las personas que se desempeñan como Arquitecto de la Información o Desarrolladores tienen en sus manos uno de los roles fundamentales de este grupo de trabajo.

En la actualidad este grupo de desarrollo independientemente que tiene la tarea de crear estas aplicaciones y en las cuales ha obtenido reconocimientos, no aprovecha la ventaja de emplearlas en su medio de trabajo como elemento principal para gestionar su propia información, lo que implica un detrimento con respecto a otros grupos de trabajo similares y le resta seriedad y autenticidad al trabajo que realiza. Este grupo estaría mejor organizado si contara con un soporte digital que brinde la posibilidad de gestionar toda la información necesaria a los usuarios.

Actualmente se hace imposible obtener y brindar información útil a los diferentes grupos de usuarios. El control de las actividades y la asignación de las tareas se realizan de forma manual y verbalmente por los encargados del proyecto lo que implica un proceso lento y hasta cierto punto desordenado. Se desperdicia la oportunidad del intercambio interactivo entre los usuarios y desarrolladores. El control global del proyecto también se hace complicado, puesto que son varios trabajos los que se desarrollan simultáneamente, todos con diferentes niveles de prioridad. Las personas relacionadas con el proyecto carecen de una vía directa e impersonal de conocer el estado de un proyecto en específico, a la vez que no cuentan con un espacio donde puedan obtener documentación sobre las herramientas fundamentales que se utilizan en la realización de los trabajos que se llevan a cabo.

Por lo anterior planteado el **problema científico** se enmarca en la siguiente interrogante: ¿Cómo gestionar la información del proyecto "Portales_PHP" que le es necesaria a su grupo de desarrollo?

Para solucionar la problemática existente se define el **objeto de estudio**: las aplicaciones Web como herramientas para la gestión de información.

Se ha delimitado como **campo de acción**: la puesta en práctica de una alternativa de desarrollo Web como medio para la gestión de la información en el proyecto "Portales_PHP".

El **objetivo principal** del presente trabajo consiste en: desarrollar una plataforma Web que permita el control de los proyectos que realiza el Grupo de Desarrollo del proyecto "Portales_PHP".

Los **objetivos específicos** están orientados para dotar a la plataforma propuesta de servicios que satisfagan los intereses y necesidades de los usuarios:

- Brindar la posibilidad de asignar y controlar las tareas destinadas a cada usuario.
- Mostrar al usuario noticias tanto nacionales como internacionales, así como información relacionada con los proyectos desarrollados o que se encuentran en desarrollo.
- Proveer un mecanismo de búsqueda.
- Ofrecer al usuario un mapa del sitio para facilitar la navegación por el mismo.
- Contar con un espacio para la gestión de documentación necesaria a los usuarios.
- Proveer de un servicio interactivo que permita el intercambio entre los diferentes usuarios.

Para darle cumplimiento a los objetivos planteados se proponen las **tareas** siguientes:

1. Realizar un estudio de la audiencia para saber las necesidades específicas para la realización de nuestro portal.
2. Revisión bibliográfica sobre el objeto de estudio y campo de acción.
3. Análisis y síntesis de la información obtenida en el punto anterior para tomar puntos de referencias con el objetivo de solucionar el problema.
4. Profundizar en la metodología de construcción de software a emplear (RUP).
5. Desarrollar nuevos tipos de contenidos.
6. Desarrollar vistas para la gestión de la información.

7. Crear la plantilla que empleará el sistema.
8. Extender los perfiles de usuarios.
9. Realizar el análisis del sistema y llevar a cabo el diseño y la implementación del mismo.
10. Realizar pruebas del software.

Como métodos científicos se manejaron los **métodos teóricos** y los **métodos empíricos**. Como método teórico se empleó el método analítico-sintético; logrando con este una mejor comprensión del trabajo, dividiéndolo en partes para su solución e integrando estas para lograr el resultado final. Como método empírico se empleó la observación, para conocer la evolución y desarrollo del sistema. Mediante este método se pudo establecer una comparación entre otros existentes con funcionalidades similares y la nueva propuesta, para de esta forma fundamentar el trabajo.

Se pretende finalmente conseguir un producto de software que resuelva las necesidades de este grupo de trabajo, logrando el control de los proyectos que se realizan, la posibilidad de gestionar documentación, asignar y controlar tareas así como mantener actualizado el estado de cada proyecto.

El presente documento se estructura en cuatro capítulos. El **Capítulo I** denominado "Fundamentación Teórica" donde se incluyen todos los aspectos teóricos que soportan este proyecto. El **Capítulo II** denominado "Características del sistema" donde se plantea el objeto de estudio, en cuanto al problema y situación problemática, objeto de automatización, información que se maneja, propuesta del sistema, modelo del negocio, se determinan además los requerimientos funcionales y no funcionales del sistema, agrupándolos en casos de uso del sistema. El **Capítulo III** denominado "Análisis y diseño del sistema" donde se determinan las clases que se utilizarán en la implementación del sistema y la relación entre ellas. La realización de los diagramas de interacción por cada caso de uso. Además se muestra todo el proceso de obtención de la base de datos y se definen los principios de diseño de interfaz y codificación y por último el **Capítulo IV** denominado "Implementación y prueba" donde se define la implementación a partir del diagrama de despliegue y el diagrama de componentes, además se determinan las pruebas a realizar en el sistema para verificar su integridad y si se ajusta a los requerimientos planteados. Se plantean posibles mejoras al sistema, bibliografía, referencias bibliográficas, anexos, y un glosario de términos.

Capítulo I. FUNDAMENTACIÓN TEÓRICA

1.1 Introducción

En este capítulo se describirán los principales aspectos asociados al desarrollo de herramientas para el control de proyectos, y se fundamentarán las consideraciones que han sido necesarias introducir. Además se describirán las tendencias y tecnologías actuales sobre las que se apoyará la propuesta.

1.2 Sistema de Gestión de Contenidos (CMS)

Realizar una Web puede ser un trabajo complicado y muy laborioso si no se dispone de las herramientas adecuadas. En el pasado las herramientas eran básicamente editores que permitían generar una página, que evolucionaron para incorporar el control de la estructura de la Web y otras funcionalidades, pero en general estaban enfocadas más a la creación que al mantenimiento. En los últimos años se ha desarrollado el concepto de Sistema de Gestión de Contenidos (*Content Management Systems* o CMS). (1)

Los Sistemas de Gestión de Contenidos son herramientas que permiten crear y mantener una Web con facilidad, encargándose de los trabajos más tediosos que hasta ahora ocupaban el tiempo de los administradores de las Webs.

Consiste en una interfaz que controla una o varias bases de datos donde se aloja el contenido del sitio. El sistema permite manejar de manera independiente el contenido por una parte y el diseño por otra. Así, es posible manejar el contenido y darle en cualquier momento un diseño distinto al sitio sin tener que volver a darle formato al contenido, además de permitir la fácil y controlada publicación en el sitio a varios editores. (2)

1.2.1 ¿Por que el nombre de Sistema de Gestión de Contenidos?

Sistema: Porque se trata generalmente de un conjunto de herramientas, apoyadas habitualmente por una base de datos, y que consisten en una serie de programas en un servidor web, y,

opcionalmente, una serie de programas cliente que permitan acceder fácilmente a esos programas en el servidor. (1)

Gestión de Contenidos: Porque desde el punto de vista del usuario del sistema, se trata de gestionar, de forma uniforme, accesible, y cómoda, un sitio web dinámico, con actualizaciones periódicas, y sobre el que pueden trabajar una o más personas, cada una de las cuales tiene una función determinada; desde el punto de vista del cliente, se trata de un sitio web dinámico, con apariencia e interfaz uniforme, con un diseño centrado en el usuario, y que permite llevar a cabo fácilmente las tareas para las que ha sido diseñado. (1)

1.2.2 Historia y surgimiento de los CMS

A principios de los años noventa, el concepto de sistemas de gestión de contenidos era desconocido. Algunas de sus funciones se realizaban con aplicaciones independientes: editores de texto y de imágenes, bases de datos y programación a medida.

En el año 1994 Illustra Information Technology utilizaba una base de datos de objetos como repositorio de los contenidos de una web, con el objetivo de poder reutilizar los objetos y ofrecía a los autores un entorno para la creación basado en patrones. La idea no cuajó entre el público y la parte de la empresa enfocada a la Web fue comprada por AOL, mientras que Informix adquirió la parte de bases de datos.

RedDot es una de las empresas pioneras que empezó el desarrollo de un gestor de contenidos el año 1994. No fue hasta a finales del año siguiente que presentaron su CMS basado en una base de datos.

Entre los CMS de código abierto uno de los primeros fue Typo 3, que empezó su desarrollo en el año 1997, en palabras de su autor, Kasper Skårhøj, "antes de que el término gestión de contenidos fuera conocido sobradamente".

PHPNuke, la herramienta que popularizó el uso de estos sistemas para las comunidades de usuarios en Internet, se empezó a desarrollar el año 2000. La primera versión supuso tres semanas de trabajo al creador, rescribiendo el código de otra herramienta, Thatware. (1)

1.2.3 Funciones de los sistemas de gestión de contenidos

Las características funcionales de los CMS las podemos agrupar en dos grandes grupos:(3)

FUNCIONES DE EDICIÓN DE CONTENIDOS	FUNCIONES DE EXPLOTACIÓN DE LA WEB
Creación de contenidos Gestión de contenidos Publicación de contenidos Presentación de contenidos	Entornos colaborativos Perfiles de usuarios Sindicación de contenidos Articulación de funcionalidades Integración de aplicaciones

Tabla 1: Características funcionales de un CMS.

Edición de contenidos

Bajo la categoría “edición de contenidos” vamos a agrupar todas las funcionalidades que ofrecen los sistemas de gestión de contenidos y que están relacionadas con el ciclo de vida básico de los contenidos. Estas funcionalidades son: (3)

- Creación de los contenidos: Los CMS ofrecen todo el utillaje informático necesario para desarrollar con comodidad la autoría y creación de los contenidos. En este sentido, acostumbran a suministrar un editor de texto WYSIWYG (what you see is what you get: ‘lo que ves es lo que obtendrás’, literalmente). Este editor, que funciona de una manera similar a los editores de texto tradicionales, permite que un contenido pueda ser creado por una persona que no posea conocimientos técnicos sobre la creación de páginas web, sobre lenguajes HTML o XML. Con esta función también se suministran soluciones de workflow (soluciones para ciclos de trabajo) que permiten que los contenidos puedan ser creados por distintas personas, descentralizando la autoría de los mismos y ofreciendo las ventajas que esto supone. Estos sistemas, además, ofrecen la posibilidad de enriquecer, en el mismo proceso de autoría, los contenidos creados con estándares de control (meta datos), asegurando de esta manera una ulterior explotación documental mucho más efectiva
- Gestión de los contenidos: La responsabilidad principal de esta tarea recae sobre un sistema de gestión de bases de datos que se encuentra incorporado en los sistemas de gestión de contenidos. La base de datos creada permite la posibilidad de dar de alta o baja un contenido en el depósito y la explotación (búsqueda y recuperación) de todos los contenidos que se han ido creando y añadiendo a ese depósito. Pero también, los contenidos no son los únicos elementos integrados en esa base

de datos. En ésta se almacenan principalmente también los datos relativos a los documentos (versiones hechas, autores, fecha de publicación, cambio y caducidad, etc.) y los datos y preferencias de los usuarios.

- Publicación de los contenidos: Los sistemas de gestión de contenidos suministran también soluciones tecnológicas que permiten realizar la publicación de las páginas web (de los contenidos) de una forma muy cómoda. Esta función ofrece la posibilidad de publicar de forma programada los contenidos (introduciendo una fecha de inicio y otra final de publicación), de diseñar interfaces de forma personalizada y de publicar en los diferentes formatos existentes (PDF, HTML, etc.).

- Presentación de los contenidos: Esta función fundamenta la separación entre el aspecto final de las páginas y el contenido de las mismas. Entre estas funciones habría que destacar la de permitir un acceso personalizado a la web a través del diseño de los menús de navegación, manteniendo los enlaces y favoreciendo, de esta manera, la usabilidad de la misma. Ofrece la posibilidad de transformar todos los contenidos en documentos con formato HTML, lo que garantiza la perfecta compatibilidad con los navegadores y plataformas y también permiten la posibilidad de restringir el tipo de usuarios de la web exigiendo un tipo de tecnología concreta (Java, Javascript, Flash, etc.) para la visualización de sus páginas.

Explotación de la web:

Las funciones relacionadas con la explotación de la web presentan una serie de soluciones importantes que puede ofrecer un CMS pero que no atañen exclusivamente al proceso de edición de contenidos. Las funciones pertenecientes a este grupo son: (3)

- Creación de Entornos Colaborativos: Esta función aprovecha las herramientas de workflow suministradas para la creación de los contenidos, se pueden solucionar, simplificar y mejorar aquellos procesos de la organización donde se puedan automatizar circuitos y flujos documentales. Estos sistemas permiten crear espacios virtuales de encuentro (foros) que habilitan la comunicación sincrónica y diacrónica entre los usuarios de los mismos

- Creación y control de perfiles de usuarios: Esta gestiona de forma segura los privilegios de los mismos. Con los CMS es posible introducir y controlar los perfiles de los usuarios de la web otorgándoles, de forma planificada, ciertos privilegios de acceso, autoría y transformación. Este

control también permite obtener informes sobre el comportamiento de explotación de los usuarios que habilitan la posibilidad de realizar estudios de usabilidad de la web e introducir mejoras a partir de los mismos. Estos sistemas permiten también que los usuarios autorizados puedan dar de alta (respecto a la autoría) a nuevos usuarios y éstos a unos terceros.

- Creación de un sistema de sindicación de contenidos: Esta función aprovecha la gestión de usuarios y perfiles, y con esto, el sistema permite la creación automática, en función de esos perfiles, de nuevos contenidos a partir de los ya existentes en el depósito. Esta función se basa, como en el caso de algunas de las anteriores, en la gestión independiente del contenido y la presentación de las páginas web.

- Introducción de nuevas funcionalidades: Permiten la integración a voluntad de nuevas funcionalidades en la web para cubrir el crecimiento y las nuevas necesidades que ésta deba cumplir.

- Integración de otras aplicaciones: Los CMS, utilizando un recurso del tipo API (Applications Programming Interface), son capaces de integrar dentro del mismo otras soluciones informáticas (aplicaciones de marketing o de comercio electrónico) externas al sistema.

1.2.4 Tipos de CMS

Existen diferentes CMS que se pueden agrupar en las siguientes categorías:

- Foros** : Crea un foro de discusión en línea donde la gente se puede reunir y discutir temas en los que están interesados. (4)

- Blogs** : Publica noticias o artículos en orden cronológico con espacio para comentarios y discusión. (4)

- Wikis** : Todo el mundo puede colaborar en los artículos y también permite espacio para discusiones. Indicado para material que irá evolucionando con el tiempo. (4)

- eCommerce** : Crea sitios para comercio electrónico. (4)

- Sitios web** : Sitios con contenido y diversa funcionalidad que sirve como fuente de información o como soporte a una comunidad.(4)

- **Moodle:** Es un sistema de gestión de la enseñanza.
- **CMS generales:** Permiten gestionar información de cualquier tipo, y son generalmente configurables y pluggables, es decir, ampliables con nuevas funcionalidades. (6)
- **Galería:** Software que permite administrar y mostrar contenido audiovisual, generalmente imágenes. (6)

En la aplicación propuesta se utilizará un CMS de propósito general.

1.2.5 Presente de los CMS

En la actualidad, aparte de la ampliación de las funcionalidades de los CMS, uno de los campos más interesantes es la incorporación de estándares que mejoran la compatibilidad de componentes, facilitan el aprendizaje al cambiar de sistema y aportan calidad y estabilidad.

Algunos de estos estándares son CSS, que permite la creación de hojas de estilo; XML, un lenguaje de marcas que permite estructurar un documento; XHTML, que es un subconjunto del anterior orientado a la presentación de documentos vía web; WAI, que asegura la accesibilidad del sistema; y RSS, para syndicar contenidos de tipo noticia.

También las aplicaciones que rodean los CMS acostumbran a ser estándar, como los servidores web Apache y ISS; los lenguajes PHP, Perl y Python; y las bases de datos MySQL y PostgreSQL. La disponibilidad para los principales sistemas operativos de estas aplicaciones y módulos, permite que los CMS puedan funcionar en diversas plataformas sin muchas modificaciones. (1)

1.2.6 Futuro de los CMS

Los CMS se convertirán en un artículo de consumo, cuando los productos se hayan establecido y más soluciones lleguen al mercado. Eso provocará una disminución de los precios en los productos comerciales y una mayor consistencia en las funcionalidades que ofrecen.

En este entorno, muchas empresas que implementan webs tendrán que cerrar.

Muchos proyectos fracasarán por no ajustarse a los estándares y no entender conceptos como usabilidad, arquitectura de la información, gestión del conocimiento y contenido.

El campo de los gestores de contenido madurará hasta conseguir un alto grado de consistencia y profesionalismo.

Se adoptarán estándares en el almacenaje, estructuración y gestión del contenido.

Se producirá una fusión entre gestión de contenidos, gestión de documentos y gestión de registros.

También se puede añadir la incorporación de sistemas de e-learning y gestión del conocimiento, y en los entornos de intranet corporativa, la posibilidad de acceder a otras fuentes de datos como por ejemplo sistemas de soporte de decisiones (*Decision Support Systems* o DSS). El campo de los CMS de código abierto tendría que seguir un desarrollo similar. (1)

1.2.7 Necesidad de un CMS

Se podría pensar que la utilidad de un sistema que gestione un entorno web no es necesaria para una web relativamente pequeña o que no se necesite tantas funcionalidades. Eso sólo podría ser cierto para un web con unas pocas páginas estáticas para el que no se prevea un crecimiento futuro ni muchas actualizaciones, lo que no es muy realista. En cualquier otro caso, la flexibilidad y escalabilidad que permiten estos sistemas, justifican su utilización en prácticamente cualquier tipo de web. (1)

Muchos usuarios particulares utilizan CMS gratuitos para elaborar y gestionar sus webs personales, obteniendo webs dinámicos llenos de funcionalidades. El resultado que obtienen es superior al de algunas empresas que se limitan a tener páginas estáticas que no aportan ningún valor añadido.

Entre los puntos más importantes que hacen útil y necesaria la utilización de un CMS están: (1)

- Inclusión de nuevas funcionalidades en la web: Esta operación puede implicar la revisión de multitud de páginas y la generación del código que aporta las funcionalidades. Con un CMS eso puede ser tan simple como incluir un módulo realizado por terceros, sin que eso suponga muchos cambios en la web. El sistema puede crecer y adaptarse a las necesidades futuras.
- Mantenimiento de gran cantidad de páginas: En una web con muchas páginas hace falta un sistema para distribuir los trabajos de creación, edición y mantenimiento con permisos de acceso a las diferentes áreas. También se tienen que gestionar los metadatos de cada documento, las versiones, la publicación y caducidad de páginas y los enlaces rotos, entre otros aspectos.

- Reutilización de objetos o componentes: Un CMS permite la recuperación y reutilización de páginas, documentos, y en general de cualquier objeto publicado o almacenado.
- Páginas interactivas: Las páginas estáticas llegan al usuario exactamente como están almacenadas en el servidor web. En cambio, las páginas dinámicas no existen en el servidor tal como se reciben en los navegadores, sino que se generan según las peticiones de los usuarios. De esta manera cuando por ejemplo se utiliza un buscador, el sistema genera una página con los resultados que no existían antes de la petición. Para conseguir esta interacción, los CMS conectan con una base de datos que hace de repositorio central de todos los datos de la web.
- Cambios del aspecto de la web: Si no hay una buena separación entre contenido y presentación, un cambio de diseño puede comportar la revisión de muchas páginas para su adaptación. Los CMS facilitan los cambios con la utilización, por ejemplo, del estándar CSS (*Cascading Style Sheets* u hojas de estilo en cascada) con lo que se consigue la independencia de presentación y contenido.
- Consistencia de la web: La consistencia en un web no quiere decir que todas las páginas sean iguales, sino que hay un orden (visual) en vez de caos. Un usuario nota enseguida cuándo una página no es igual que el resto de las de la misma web por su aspecto, la disposición de los objetos o por los cambios en la forma de navegar. Estas diferencias provocan sensación de desorden y dan a entender que el web no lo han diseñado profesionales. Los CMS pueden aplicar un mismo estilo en todas las páginas con el mencionado CSS, y aplicar una misma estructura mediante patrones de páginas.
- Control de acceso: Controlar el acceso a una web no consiste simplemente al permitir la entrada a la web, sino que comporta gestionar los diferentes permisos a cada área de la web aplicado a grupos o individuos.

1.2.8 Creación de Contenidos

Un CMS aporta herramientas para que los creadores sin conocimientos técnicos en páginas web puedan concentrarse en el contenido. Lo más habitual es proporcionar un editor de texto WYSIWYG, en el que el usuario ve el resultado final mientras escribe, al estilo de los editores comerciales, pero con un rango de formatos de texto limitado. Esta limitación tiene sentido, ya que

el objetivo es que el creador pueda poner énfasis en algunos puntos, pero sin modificar mucho el estilo general del sitio web.

Hay otras herramientas como la edición de los documentos en XML, utilización de aplicaciones informáticas con las que se integra el CMS, importación de documentos existentes y editores que permiten añadir marcas, habitualmente HTML, para indicar el formato y estructura de un documento.

Un CMS puede incorporar una o varias de estas herramientas, pero siempre tendría que proporcionar un editor WYSIWYG por su facilidad de uso y la comodidad de acceso desde cualquier ordenador con un navegador y acceso a Internet.

Para la creación del sitio propiamente dicho, los CMS aportan herramientas para definir la estructura, el formato de las páginas, el aspecto visual, uso de patrones, y un sistema modular que permite incluir funciones no previstas originalmente. (1)

1.2.9 Gestión de contenido

Los documentos creados se depositan en una base de datos central donde también se guardan el resto de datos de la web, cómo son los datos relativos a los documentos (versiones hechas, autor, fecha de publicación y caducidad, etc.), datos y preferencias de los usuarios, la estructura de la web, etc.

La estructura de la web se puede configurar con una herramienta que, habitualmente, presenta una visión jerárquica del sitio y permite modificaciones. Mediante esta estructura se puede asignar un grupo a cada área, con responsables, editores, autores y usuarios con diferentes permisos. Eso es imprescindible para facilitar el ciclo de trabajo (*workflow*) con un circuito de edición que va desde el autor hasta el responsable final de la publicación. El CMS permite la comunicación entre los miembros del grupo y hace un seguimiento del estado de cada paso del ciclo de trabajo. (1)

1.2.10 Publicación

Una página aprobada se publica automáticamente cuando llega la fecha de publicación, y cuando caduca se archiva para futuras referencias. En su publicación se aplica el patrón definido para toda la web o para la sección concreta donde está situada, de forma que el resultado final es un sitio web con un aspecto consistente en todas sus páginas. Esta separación entre contenido y forma permite

que se pueda modificar el aspecto visual de un sitio web sin afectar a los documentos ya creados y libera a los autores de preocuparse por el diseño final de sus páginas. (1)

1.2.11 Presentación

Un CMS puede gestionar automáticamente la accesibilidad de la web, con soporte de normas internacionales de accesibilidad como WAI, y adaptarse a las preferencias o necesidades de cada usuario. También puede proporcionar compatibilidad con los diferentes navegadores disponibles en todas las plataformas (Windows, Linux, Mac, Palm, etc.) y su capacidad de internacionalización lo permite adaptarse al idioma, sistema de medidas y cultura del visitante.

El sistema se encarga de gestionar otros aspectos como son los menús de navegación o la jerarquía de la página actual dentro del web, añadiendo enlaces de forma automática. También gestiona todos los módulos, internos o externos, que incorpore al sistema. Así por ejemplo, con un módulo de noticias se presentarían las novedades aparecidas en otra web, con un módulo de publicidad se mostraría un anuncio o mensaje animado, y con un módulo de foro se podría mostrar, en la página principal, el título de los últimos mensajes recibidos. Todo eso con los enlaces correspondientes y, evidentemente, siguiendo el patrón que los diseñadores hayan creado. (1)

1.2.12 CMS comerciales y de código abierto

Se puede hacer una primera división de los CMS según el tipo de licencia escogido. Por una parte están los CMS comercializados por empresas que consideran el código fuente un activo más que tienen que mantener en propiedad, y que no permiten que terceros tengan acceso. Por la otra tenemos los de código fuente abierto, desarrollados por individuos, grupos o empresas que permiten el acceso libre y la modificación del código fuente.

La disponibilidad del código fuente posibilita que se hagan personalizaciones del producto, correcciones de errores y desarrollo de nuevas funciones. Este hecho es una garantía de que el producto podrá evolucionar incluso después de la desaparición del grupo o empresa creadora.

Algunas empresas también dan acceso al código, pero sólo con la adquisición de una licencia especial o después de su desaparición. Generalmente las modificaciones sólo pueden hacerlas los mismos desarrolladores, y siempre según sus prioridades.

Los CMS de código abierto son mucho más flexibles en este sentido, pero se podría considerar que la herramienta comercial será más estable y coherente al estar desarrollada por un mismo grupo. En la práctica esta ventaja no es tan grande, ya que los CMS de código abierto también están coordinados por un único grupo o por empresas, de forma similar a los comerciales.

Utilizar una herramienta de gestión de contenidos de código abierto tiene otra ventaja que hace decidirse a la mayoría de usuarios: su coste. Habitualmente todo el software de código abierto es de acceso libre, es decir, sin ningún coste en licencias. Sólo en casos aislados se hacen distinciones entre empresas y entidades sin ánimo de lucro o particulares. En comparación, los productos comerciales pueden llegar a tener un coste que sólo una gran empresa puede asumir.

En cuanto al soporte, los CMS comerciales acostumbran a dar soporte profesional, con un coste elevado en muchos casos, mientras que los de código abierto se basan más en las comunidades de usuarios que comparten información y solución a los problemas. Las formas de soporte se pueden mezclar, y así encontramos CMS de código abierto con empresas que ofrecen servicios de valor añadido y con activas comunidades de usuarios. En el caso comercial también sucede, pero el coste de las licencias hace que el gran público se decante por otras opciones y por lo tanto las comunidades de soporte son más pequeñas.

Un problema que acostumbra a tener el software de código abierto es la documentación, generalmente escasa, dirigida a usuarios técnicos, o mal redactada. Este problema se agrava en el caso de los módulos desarrollados por terceros, que no siempre incorporan las instrucciones de su funcionamiento de forma completa y entendible.

En el mercado hay CMS de calidad tanto comerciales como de código abierto. Muchos CMS de código abierto están poco elaborados (aunque en plena evolución), pero también lo encontramos entre los comerciales. En definitiva, un buen CMS de código abierto es mucho más económico que su homólogo comercial, con la ventaja de disponer de todo el código fuente y de una extensa comunidad de usuarios.

Por todos estos motivos, y como apuesta por la filosofía del software libre, en este trabajo sólo se presentan algunos CMS de código abierto. (1)

1.3 Drupal

Drupal es un sistema de administración de contenido para sitios Web. Permite publicar artículos, imágenes, u otros archivos y servicios añadidos como foros, encuestas, votaciones, blogging y administración de usuarios y permisos. Drupal es un sistema dinámico: en lugar de almacenar sus contenidos en archivos estáticos en el sistema de ficheros del servidor de forma fija, el contenido textual de las páginas y otras configuraciones son almacenados en una base de datos y se editan utilizando un entorno Web incluido en el producto.

Está basado en PHP, que fue originalmente escrito por Dries Buytaert y es el software usado para impulsar los sitios web Debian Planet, Terminus1525, Spread Firefox y Kernel Trap.

"Drupal", es el equivalente fonético en inglés a la palabra neerlandesa ' druppel ' que significa ' gota '.

Está compuesto por una infraestructura base y un conjunto de módulos que ofrecen un amplio conjunto de funciones, incluyendo sistemas de comercio electrónico , galerías de fotos, administración de listas de correo electrónico , e integración de CVS. Es posible añadir módulos de terceros para modificar el comportamiento de Drupal u ofrecer nuevas funciones.

Drupal se usa, entre otros, en intranets de compañías, enseñanza en línea, comunidades de arte y administración de proyectos. Muchos piensan que la relevancia de Drupal en las comunidades de usuarios es lo que lo hace destacarse de la competencia. (7)

1.3.1 Distribuciones de Drupal

Hay varias distribuciones de Drupal personalizadas. Muchas de ellas están previamente preempaquetadas con módulos de software de terceros, pero algunas contienen modificaciones del núcleo y entre ellas están: (7)

- Distribución oficial
- CivicSpace: Edición especial para comunidades.
- DrupalEd: dirigida a usar Drupal en una clase en línea.
- Drupal For Bloggers: dirigida a bloggers, en particular exusuarios de MovableType.

1.3.2 Licencia de Drupal

Drupal se distribuye bajo la licencia GNU GPL, y por lo tanto es software libre. (7)

1.3.3 Características de Drupal

Características generales (8)

- **Ayuda on-line:** Un robusto sistema de ayuda online y páginas de ayuda para los módulos del 'núcleo', tanto para usuarios como para administradores.
- **Búsqueda:** Todo el contenido en Drupal es totalmente indexado en tiempo real y se puede consultar en cualquier momento.
- **Código abierto:** El código fuente de Drupal está libremente disponible bajo los términos de la licencia GNU/GPL. Al contrario que otros sistemas de 'blogs' o de gestión de contenido propietarios, es posible extender o adaptar Drupal según las necesidades.
- **Módulos:** La comunidad de Drupal ha contribuido muchos módulos que proporcionan funcionalidades como 'página de categorías', autenticación mediante jabber, mensajes privados, bookmarks, etc.
- **Personalización:** Un robusto entorno de personalización está implementado en el núcleo de Drupal. Tanto el contenido como la presentación pueden ser individualizados de acuerdo las preferencias definidas por el usuario.
- **URLs amigables:** Drupal usa el mod_rewrite de Apache para crear URLs que son manejables por los usuarios y los motores de búsqueda.

Gestión de usuarios (8)

- **Autenticación de usuarios:** Los usuarios se pueden registrar e iniciar sesión de forma local o utilizando un sistema de autenticación externo como Jabber, Blogger, LiveJournal u otro sitio Drupal. Para su uso en una intranet, Drupal se puede integrar con un servidor LDAP.

• **Permisos basados en roles:** Los administradores de Drupal no tienen que establecer permisos para cada usuario. En lugar de eso, pueden asignar permisos a un 'rol' y agrupar los usuarios por roles.

Gestión de contenido (8)

• **Control de versiones:** El sistema de control de versiones de Drupal permite seguir y auditar totalmente las sucesivas actualizaciones del contenido: qué se ha cambiado, la hora y la fecha, quién lo ha cambiado, y más. También permite mantener comentarios sobre los sucesivos cambios o deshacer los cambios recuperando una versión anterior.

• **Enlaces permanentes:** (*Permalinks*) Todo el contenido creado en Drupal tiene un enlace permanente asociado a él para que pueda ser enlazado externamente sin temor de que el enlace falle en el futuro.

• **Objetos de Contenido:** (*Nodos*) El contenido creado en Drupal es, funcionalmente, un objeto (Nodo). Esto permite un tratamiento uniforme de la información, como una misma cola de moderación para envíos de diferentes tipos, promocionar cualquiera de estos objetos a la página principal o permitir comentarios -o no- sobre cada objeto.

• **Plantillas:** (*Templates*) El sistema de temas de Drupal separa el contenido de la presentación permitiendo controlar o cambiar fácilmente el aspecto del sitio web. Se pueden crear plantillas con HTML y/o con PHP.

• **Sindicación del contenido:** Drupal exporta el contenido en formato RDF/RSS para ser utilizado por otros sitios web. Esto permite que cualquiera con un 'Agregador de Noticias', tal como *NetNewsWire* o *Radio UserLand* visualice el contenido publicado en la web desde el escritorio.

Blogging (8)

• **Agregador de noticias:** Drupal incluye un potente Agregador de Noticias para leer y publicar enlaces a noticias de otros sitios web. Incorpora un sistema de cache en la base de datos, con temporización configurable.

•**Soporte de Blogger API:** La API de Blogger permite que un sitio Drupal sea actualizado utilizando diversas herramientas, que pueden ser 'herramientas web' o 'herramientas de escritorio' que proporcionen un entorno de edición más manejable.

Plataforma (8)

•**Independencia de la base de datos:** Aunque la mayor parte de las instalaciones de Drupal utilizan MySQL, existen otras opciones. Drupal incorpora una 'capa de abstracción de base de datos' que actualmente está implementada y mantenida para MySQL y PostgreSQL, aunque permite incorporar fácilmente soporte para otras bases de datos.

•**Multiplataforma:** Drupal ha sido diseñado desde el principio para ser multi-plataforma. Puede funcionar con Apache o Microsoft IIS como servidor web y en sistemas como Linux, BSD, Solaris, Windows y Mac OS X. Por otro lado, al estar implementado en PHP, es totalmente portable.

•**Múltiples idiomas y Localización:** Drupal está pensado para una audiencia internacional y proporciona opciones para crear un portal multilingüe. Todo el texto puede ser fácilmente traducido utilizando una interfaz web, importando traducciones existentes o integrando otras herramientas de traducción como *GNU ettext*.

Administración y Análisis (8)

•**Administración vía Web:** La administración y configuración del sistema se puede realizar enteramente con un navegador y no precisa de ningún software adicional.

•**Análisis, Seguimiento y Estadísticas:** Drupal puede mostrar en las páginas web de administración informes sobre *referrals* (enlaces entrantes), popularidad del contenido, o de cómo los usuarios navegan por el sitio.

•**Registros e Informes:** Toda la actividad y los sucesos del sistema son capturados en un 'registro de eventos', que puede ser visualizado por un administrador.

Características de comunidad (8)

•**Comentarios enlazados:** Drupal proporciona un potente modelo de comentarios enlazados que posibilita seguir y participar fácilmente en la discusión sobre el comentario publicado. Los comentarios son jerárquicos, como en un grupo de noticias o un foro.

•**Encuestas:** Drupal incluye un módulo que permite a los administradores y/o usuarios crear encuestas on-line totalmente configurables.

•**Foros de discusión:** Drupal incorpora foros de discusión para crear sitios comunitarios vivos y dinámicos.

•**Libro Colaborativo:** Esta característica es única de Drupal y permite crear un proyecto o "libro" a ser escrito y que otros usuarios contribuyan contenido. El contenido se organiza en páginas cómodamente navegables.

Rendimiento y escalabilidad (8)

•**Control de congestión:** Drupal incorpora un mecanismo de control de congestión que permite habilitar y deshabilitar determinados módulos o bloques dependiendo de la carga del servidor. Este mecanismo es totalmente configurable y ajustable.

•**Sistema de Cache:** El mecanismo de cache elimina consultas a la base de datos incrementando el rendimiento y reduciendo la carga del servidor.

1.4 Soluciones de Código Abierto (Open Source)

La realidad en la que se encuentran envueltas las diferentes personas, empresas, entidades gubernamentales e instituciones de todo el mundo es sumamente compleja y restringida. Afortunadamente ha surgido una alternativa que permite la existencia de nuevos esquemas de desarrollo cultural y técnico basado en un nuevo modelo de código abierto y libre.

Código abierto (del inglés open source) es el término por el que se conoce el software distribuido y desarrollado libremente. Este término empezaron a utilizarlo en 1998 algunos usuarios de la comunidad del software libre, tratando de usarlo como reemplazo al ambiguo nombre original en inglés del software libre (free software).

Free en inglés puede significar diferentes cosas: gratuidad y libertad. Por ello, por un lado, permite pensar en "software por el que no hay que pagar" (software gratuito) y, por otro, se adapta al significado que se pretendió originalmente (software que posee ciertas libertades). El término para algunos no resultó apropiado como reemplazo para el ya tradicional free software, pues eliminaba la

idea de libertad (incluso hay algunos que usan —en inglés— el término libre software para evitar la ambigüedad de free).

Desde el punto de vista de una "traducción estrictamente literal", el significado obvio de "código abierto" es que "se puede mirar el código fuente", por lo que puede ser interpretado como un término más débil y flexible que el del software libre. Basado en ello se argumenta que un programa de código abierto puede ser software libre, pero también puede ser semilibre o incluso completamente no libre. Sin embargo, por lo general, un programa de código abierto puede ser y de hecho es software libre, como igualmente un programa Software Libre es Open Source. Esto ocurre dado que ambos movimientos reconocen el mismo conjunto de licencias y tiene principios equivalentes.

Un término que pretende resolver posibles ambigüedades o confusiones que ambos términos generan es FOSS (Free Open Source Software).

En la actualidad open source es utilizado para definir un movimiento nuevo de software (la Open Source Initiative), diferente al movimiento del Software Libre, incompatible con este último desde el punto de vista filosófico, y completamente equivalente desde el punto de vista práctico, de hecho, ambos movimientos trabajan juntos en el desarrollo práctico de proyectos. (9)

Seguidamente se describe brevemente las tres herramientas necesarias para el desarrollo de una aplicación Web siguiendo la línea del software libre.

1.4.1 Servidor Web. Apache

A partir del análisis de los datos obtenidos del sitio de Netcraft (10) donde constantemente se realizan encuestas sobre los servidores web que más se utilizan mundialmente, se puede apreciar con claridad en la Figura 1. la superioridad del servidor Web Apache sobre otros reconocidos como el Internet Information Server (IIS).

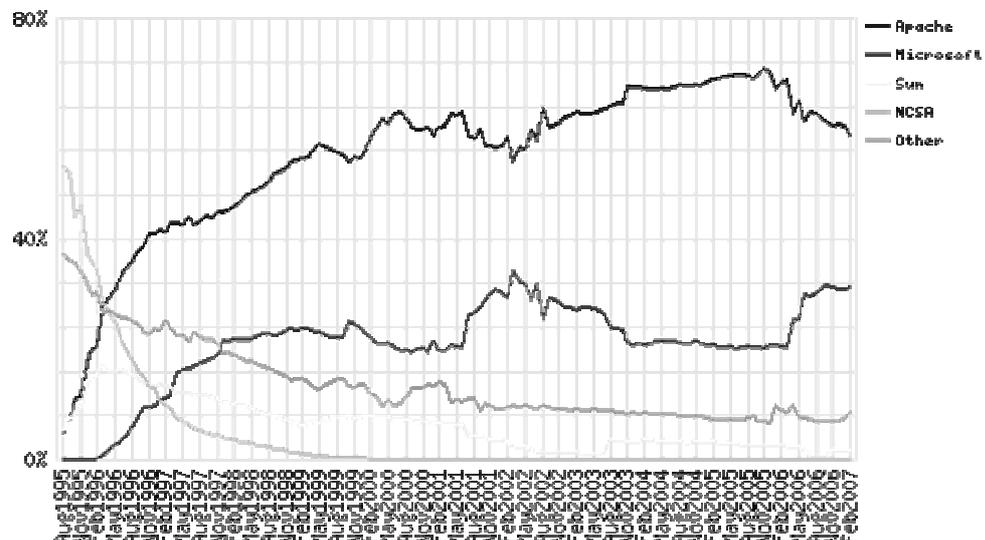


Ilustración 1: Comportamiento de encuestas de servidores más utilizados

Developer	January 2007	Percent	February 2007	Percent	Change
Apache	64312083	60.17	63869543	58.70	1.47
Microsoft	32898421	30.78	33833566	31.09	0.31
Sun	1749026	1.64	1845584	1.70	0.06
Zeus	551767	0.52	530596	0.49	0.03

Tabla2: Servidores Web más utilizados mundialmente. (Agosto 1995 –Febrero 2007)

De acuerdo a las estadísticas de Netcraft para el mes de febrero del 2007, Apache sigue siendo el servidor Web más utilizado en la red, aunque en los últimos tiempos ha bajado su por ciento a los 58.7%, es la primera vez que la porción del mercado de apache ha estado debajo de 60% desde septiembre del 2002. (10)

Apache además de constituir el servidor más utilizado, forma parte del proyecto de “código abierto”, significando que puede obtenerse libremente a través de Internet de forma gratuita la aplicación y su

código fuente, lo que ha permitido el desarrollo de funciones del Apache por muchos programadores con carácter voluntario, dando lugar a rápidas mejoras en su funcionalidad y posibilitando la rápida subsanación de errores o fallos de seguridad.

Aunque Apache es un servidor inicialmente pensado para ser ejecutado en servidores Unix, se ha realizado un verdadero esfuerzo de migración, de modo que es un software capaz de funcionar en una gran cantidad de plataformas: aix, aux, beos, bs2000-osd, bsdi, cygwin, darwin, dgux, digitalunix, freebsd, hpux, irix, linux, macosx, macosxserver, netbsd, netware, openbsd, os2, os390, osf1, qnx, reliantunix, rhapsody, sinix, solaris, sunos, unixware y win32. Eso sí, sin duda, Linux es su plataforma favorita y en la que más servidores de Internet están trabajando. (11)

Teniendo en cuenta las consideraciones anteriores, se puede concluir que una buena opción es el empleo de Apache como servidor web.

1.4.2 Servidor de Bases de Datos. MySQL.

MySQL es uno de los Gestores de Bases de Datos más populares desarrolladas bajo la filosofía de código abierto, puede utilizarse gratuitamente y su código fuente está disponible. Es desarrollado y mantenido por la empresa MySQL AB y considerado una de las aplicaciones de código abierto más exitosas, abriéndose camino gracias a su nivel competitivo y su accesibilidad. Entre los clientes mas destacados de la compañía se cuentan nada menos que la Amazon.com, NASA, Yahoo, Suzuki, Google y la Agencia Associated Press, Wikipedia. (12)

MySQL funciona sobre múltiples plataformas, incluyendo AIX, BSD, FreeBSD, HP-UX, GNU/Linux, Mac OS X, NetBSD, Novell Netware, OpenBSD, OS/2 Warp, QNX, SGI IRIX, Solaris, SunOS, SCO OpenServer, SCO UnixWare, Tru64, Windows 95, Windows 98, Windows NT, Windows 2000, Windows XP, Windows Vista y otras versiones de Windows. (12)

Inicialmente, MySQL carecía de elementos considerados esenciales en las bases de datos relacionales, tales como integridad referencial y transacciones. A pesar de ello, atrajo a los desarrolladores de páginas web con contenido dinámico, justamente por su simplicidad; aquellos elementos faltantes fueron llenados por la vía de las aplicaciones que la utilizan. Poco a poco los elementos faltantes en MySQL están siendo incorporados tanto por desarrollos internos, como por

desarrolladores de software libre. Entre las características disponibles en las últimas versiones se puede destacar:(12)

- Amplio subconjunto del lenguaje SQL. Algunas extensiones son incluidas igualmente.
- Disponibilidad en gran cantidad de plataformas y sistemas.
- Diferentes opciones de almacenamiento según si se desea velocidad en las operaciones o el mayor número de operaciones disponibles.
- Transacciones y claves foráneas.
- Conectividad segura.
- Replicación.
- Búsqueda e indexación de campos de texto.

Dentro de las Ventajas que MySQL ofrece se encuentran:

- Alto rendimiento.
- Elevada velocidad tanto al conectar con el servidor como al servir consultas y demás funciones.
- Excelentes utilidades de administración (backup, recuperación de errores, etc).
- Aunque se bloquee, no suele perder información ni corromper los datos.
- Excelente integración con PHP.
- No tiene límites en el tamaño de los registros.
- Significativo control de acceso, es decir, qué usuarios tienen acceso a que tablas y con qué permisos.
- MySQL se comporta mejor que otros motores de Base de Datos como SQL Server y Oracle a la hora de modificar ó añadir campos a una tabla.
- Es multi-hilo, o sea, maneja muchas conexiones al mismo tiempo, donde cada una tiene su propio hilo, de modo que ningún hilo tiene que esperar por otro, a menos que uno esté modificando una tabla que otro quiera acceder.

Según las cifras del fabricante, existirán más de seis millones de copias de MySQL funcionando en la actualidad, lo que supera la base instalada de cualquier otra herramienta de bases de datos. (12)

1.4.3 Lenguaje de Programación. PHP

PHP (siglas que originalmente significaban Personal Home Page) fue primero escrito por Rasmus Lerdorf como un simple conjunto de scripts de Perl para guiar a los usuarios en sus páginas. PHP actualmente significa Hypertext Preprocessor.

Es uno de los lenguajes de lado servidor más extendidos en la web. Nacido en 1994, se trata de un lenguaje de creación relativamente creciente que ha tenido una gran aceptación en la comunidad de webmasters debido sobre todo a la potencia y simplicidad que lo caracterizan.

PHP nos permite embeber sus pequeños fragmentos de código dentro de la página HTML y realizar determinadas acciones de una forma fácil y eficaz sin tener que generar programas íntegramente en un lenguaje distinto al HTML.

Se podría efectuar una comparación entre el lenguaje PHP y la tecnología ASP a la hora de generar páginas dinámicas estableciéndose algunas diferencias:

- PHP, aunque multiplataforma, ha sido concebido inicialmente para entornos UNIX y es en este sistema operativo donde se pueden aprovechar mejor sus prestaciones. ASP, siendo una tecnología Microsoft, está orientado hacia sistemas Windows, especialmente NT.
- Las tareas fundamentales que puede realizar directamente el lenguaje son definidas en PHP como funciones mientras que ASP invoca más frecuentemente los objetos. Por supuesto, esto no es más que una simple cuestión de forma ya que ambos lenguajes soportan igualmente ambos procedimientos.
- ASP realiza numerosas tareas sirviéndose de componentes (objetos) que deben ser comprados (o programados) por el servidor a determinadas empresas especializadas. PHP presenta una filosofía totalmente diferente y, con un espíritu más generoso, es progresivamente construido por colaboradores desinteresados que implementan nuevas funciones en nuevas versiones del lenguaje, dispone de un sofisticado método de manejo de variables, conformando un sistema robusto y estable.

Entre otras características del lenguaje PHP están:

- Seguridad:** El sistema debe poseer protecciones contra ataques. PHP provee diferentes niveles de seguridad, estos pueden ser configurados desde el archivo .ini.

- Simplicidad:** Se les debe permitir a los programadores generar código productivamente en el menor tiempo posible. Usuarios con experiencia en C y C++ podrán utilizar PHP rápidamente.

Otra característica a tener en cuenta sería la conectividad. PHP dispone de una amplia gama de librerías, y agregarle extensiones es muy fácil. Esto le permite al PHP ser utilizado en muchas áreas diferentes, tales como encriptado, gráficos, XML y otras. (13)

Ventajas adicionales de PHP

- PHP corre en (casi) cualquier plataforma utilizando el mismo código fuente, pudiendo ser compilado y ejecutado en algo así como 25 plataformas, incluyendo diferentes versiones de Unix, Windows (95,98,NT,ME,2000,XP,2003 Server) y Macs. Como en todos los sistemas se utiliza el mismo código base, los scripts pueden ser ejecutados de manera independiente al OS.

- La sintaxis de PHP es similar a la del C, por esto cualquiera con experiencia en lenguajes del estilo C podrá entender rápidamente PHP. Entre los lenguajes del tipo C incluimos al Java y Javascript, de hecho mucha de la funcionalidad del PHP se la debe al C en funciones como fread() o strlen(), así que muchos programadores se sentirán como en casa.

- PHP es completamente expandible. Está compuesto de un sistema principal, un conjunto de módulos y una variedad de extensiones de código.

- Muchas interfaces distintas para cada tipo de servidor. PHP actualmente se puede ejecutar bajo Apache, IIS, AOLServer, Roxen y THHTTPD. Otra alternativa es configurarlo como modulo CGI.

- Puede interactuar con muchos motores de bases de datos tales como MySQL, MS SQL, Oracle, Informix, PostgreSQL, y otros muchos. Siempre podrás disponer de ODBC para situaciones que lo requieran.

- Una gran variedad de módulos cuando un programador PHP necesite una interfase para una librería en particular, fácilmente podrá crear una API para esta. Algunas de las que ya vienen

implementadas permiten manejo de gráficos, archivos PDF, Flash, Cybercash, calendarios, XML, IMAP, POP, etc.

- Rapidez. PHP generalmente es utilizado como modulo de Apache, lo que lo hace extremadamente veloz. Esta completamente escrito en C, así que se ejecuta rápidamente utilizando poca memoria.
- PHP es Open Source, lo cual significa que el usuario no depende de una compañía específica para arreglar cosas que no funcionan, además no estás forzado a pagar actualizaciones anuales para tener una versión que funcione. (14)

1.5 Metodología de Ingeniería de Software empleada

Todo sistema a implementar se rige por una metodología. Esta no es más que un conjunto de pasos a seguir en pos del resultado final. En la esfera del desarrollo de software, se busca obtener un producto de alta calidad, que satisfaga las necesidades del usuario. La metodología en la ingeniería de software sería entonces el conjunto ordenado de pasos a seguir para lograr un software de alta calidad que satisfaga las necesidades del usuario, define quien debe hacer, qué, cómo y cuando debe hacerlo. No se puede decir que exista una metodología para el desarrollo de software que sea universal, sino que esta se adapta teniendo en cuenta las características del proyecto.

1.5.1 Proceso Unificado de Desarrollo (RUP)

Para el desarrollo de este sistema se empleó el proceso unificado de desarrollo de software (RUP¹). Este es un proceso para el desarrollo de software orientados a objetos, cuyo objetivo y tendencia actual es su utilización para la construcción de sistemas más grandes, más completos y de más alta calidad. Este es un proceso dirigido por casos de uso, es iterativo e incremental y centrado en la arquitectura.(15) Sigue un conjunto de etapas, cada una de ellas con un flujo de trabajo determinado. RUP establece refinamientos sucesivos de una arquitectura ejecutable, construida como un prototipo evolutivo. Los objetivos de cada iteración se establecen en función de las evaluaciones de las iteraciones anteriores.

¹ Rational Unified Process.

1.5.2 Lenguaje Unificado de Modelación (UML)

RUP utiliza UML² como lenguaje que le permite la modelación de sistemas con tecnología orientada a objetos, así como de modelaje visual que permite una abstracción del sistema y sus componentes. Anteriormente existían diversos métodos y técnicas Orientadas a Objetos, con muchos aspectos en común pero utilizando distintas notaciones, se presentaban inconvenientes para el aprendizaje, aplicación, construcción y uso de herramientas, lo que generó la creación del UML como estándar para la modelación de sistemas de software, pero con posibilidades de ser aplicado a todo tipo de proyectos. Esta notación cubre tanto lo conceptual (procesos de negocios, funciones del sistema), como lo concreto (clases, componentes de software reutilizable, bases de datos) mediante el empleo de modelos gráficos propios (diagramas).

1.6 Herramienta para el Modelado (Visual Paradigm)

Todo el proceso de análisis y diseño se desarrolló en Visual Paradigm. El Visual Paradigm es una poderosa herramienta CASE de modelación visual. Utiliza UML para el modelado, permitiendo crear tipos diferentes de diagramas en un ambiente totalmente visual. Es muy sencillo de usar, fácil de instalar y actualizar. Genera código para varios lenguajes. Tiene integrado el MS Visio³ y es compatible con otras ediciones. Es una herramienta para ser utilizada en la creación de software libre, siendo esta una de las características que dio lugar a que se seleccionara para el desarrollo del sistema, sin dejara de mencionar que el “Visual Paradigm, además ofrece:

- Un entorno de creación de diagramas para UML 2.0.
- Diseño centrado en casos de uso y enfocado al negocio que generan un software de mayor calidad.
- Uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación.
- Capacidades de ingeniería directa e inversa.
- Modelo y código que permanece sincronizado en todo el ciclo de desarrollo.
- Disponibilidad de múltiples versiones, para cada necesidad.

² Unified Model Language.

³ Microsoft Visio.

- Disponibilidad de integrarse en los principales IDEs⁴.
- Disponibilidad en múltiples plataformas. (16)

Por otra parte, posibilita la representación gráfica de los diagramas permitiendo ver el sistema desde diferentes perspectivas, como el de componentes, despliegue, secuencia, casos de uso, clase, actividad, estado, entre otros. Además, identifica requisitos y comunica información, se centra en cómo los componentes del sistema interactúan entre ellos, sin entrar en detalles excesivos, además, permite ver las relaciones entre los componentes del diseño y mejora la comunicación entre los miembros del equipo usando un lenguaje gráfico.

Tiene disponible distintas versiones: Enterprise, Professional, Standard, Modeler, Personal y Community. Facilita licencias especiales para fines académicos. El utilizado en el desarrollo del trabajo que se expone se utilizó la versión Enterprise.

1.7 Conclusiones

Luego de haber realizado un análisis teórico general se puede concluir el capítulo indicando los lenguajes que serán utilizados para la implementación de la propuesta. Se utilizarán PHP y MySQL, lenguajes que se han convertido en estándares para el desarrollo de aplicaciones web. PHP se ha convertido en un poderoso lenguaje de programación que se ejecuta del lado del servidor y que ha sido creado específicamente para el desarrollo de aplicaciones web, y MySQL como un poderoso motor de bases de datos relacionales que proporciona el repositorio perfecto para aplicaciones que requieran acceso rápido y confiable a la información. Se utilizará Apache como servidor web, que es el servidor más utilizado en la red, acaparando el 69.70%.

Además se analizaron los Sistemas de Gestión de Contenidos y en especial Drupal que por las ventajas que brinda será el utilizado en la aplicación. A estas ventajas se le añade que son software libre, lo que permite iniciar el desarrollo de aplicaciones sin necesidad de pagar licencias por ello.

⁴ Integrated Drive Electronics (En español Entorno Integrado de Desarrollo).

Capítulo II. CARACTERÍSTICAS DEL SISTEMA

2.1 Introducción

En el presente capítulo se describe la propuesta de solución de este trabajo, se detallan las características que el sistema debe tener, los objetivos estratégicos de la organización y procesos de negocio que los soportan, el flujo actual de los procesos involucrados en el campo de acción y un análisis crítico de cómo se ejecutan actualmente esos procesos.

Además se modela el negocio propuesto, identificándose los actores, trabajadores y los casos de uso correspondientes. Se plantean los requisitos funcionales y no funcionales de la aplicación a desarrollar y se modela la misma en términos de casos de uso de sistema.

2.2 Objeto de estudio

El principal objetivo de la creación del Proyecto Portales_PHP es desarrollar proyectos web de diferentes tipos y asesorar en este campo a otros proyectos que estén realizando trabajos similares.

La obra básica del proyecto es el desarrollo de aplicaciones web. Actualmente está diseñada de tal forma que las actividades a realizar pasan por 4 etapas fundamentales:

- La definición de la información y la forma en que esta se representará y se gestionará.
- El diseño gráfico.
- La construcción.
- La transferencia.

Todos estos argumentos conforman el proceso de trabajo del ente en cuestión.

La información generada en cada una de las etapas anteriormente mencionadas, se mueve de persona a persona y muchas veces se duplica e incluso ha sucedido que diversas personas manejan diferentes versiones de la misma.

Actualmente los procesos no fluyen de forma óptima, y en esto incide la no existencia de una herramienta que controle los mismos. En muchas ocasiones no todas las personas que requieren cierta información la encuentran a su disposición en un momento determinado, es esta una de las principales dificultades. Además se pierde mucha información y tiempo por la cierta desorganización que existe, lo que provoca que incluso se dupliquen esfuerzos en muchas actividades.

2.3 Objeto de automatización

Los principales procesos a automatizar en el proyecto son:

- La gestión de proyectos.
- La capacitación del personal involucrado.
- La gestión de la documentación.
- El control de personas.
- La asignación de tareas.
- El control de actividades

En cuanto al desarrollo de proyectos el sistema debe incluir de cada proyecto: nombre, descripción, cliente, contactos, estados, cronología, esta última debe tener un recorrido de la vida del proyecto, que incluya todas las actividades y versiones que se han obtenido del mismo. Además debe reflejar las personas que trabajan en él, también incluye las fechas de inicio y fin del proyecto, la url, el alias, el costo, datos técnicos.

La capacitación del personal involucrado nos indica que el sistema debe presentar una sección donde se incluyan materiales necesarios para la preparación de las personas en las distintas tecnologías que se emplean en el trabajo del proyecto, así como indicaciones sobre temas que deben ser estudiados o que resultarían interesantes.

La gestión de la documentación representa la documentación propia de cada proyecto, esto debe relacionarse de alguna manera con cada proyecto en cuestión. Documentación no solo se refiere a la existencia de documentos, pueden ser también presentaciones, componentes, fragmentos de código. Deben existir formas de recuperar esta información por distintas vías.

El control de personas nos muestra que debe existir un espacio donde se cuente con la información del personal, datos básicos como nombre, rol, año, dirección de correo, imagen.

Por otra parte el control de actividades sería una primera aproximación al problema en cuestión, pues este incluye el control exacto del tiempo de trabajo de las personas, la generación de estadísticas sobre este aspecto, y otros puntos donde cada persona pueda encontrar las tareas que debe realizar, así como su fecha de cumplimiento.

En cuanto a la asignación de tareas se hace necesario contar con un espacio donde cada usuario pueda conocer de forma actualizada la tarea que se le ha asignado con su descripción y la fecha de inicio y fin para evitar atrasos que pueden perjudicar en la entrega a tiempo del producto.

Cada una de estas fases genera una serie de recursos (documentos, paquetes de código fuente, imágenes, tutoriales, componentes, etc.) que de alguna forma deben ser almacenados, controlados y garantizar una fácil recuperación de los mismos.

Hoy en la universidad existe una herramienta de desarrollo colaborativo como lo es el Gforge. Esta es una herramienta confeccionada con el objetivo de ayudar a los proyectos a desarrollar de forma comunitaria y no presencial, basándose en técnicas de gestión de la calidad del software, provee hospedaje para proyectos, foros, listas de discusión, seguimiento de errores, herramientas para crear y controlar el acceso al código fuente en el repositorio Subversion entre muchas más herramientas y facilidades. (17)

El Gforge sería la herramienta fundamental para este tipo de actividades, pero existe poca cultura de uso y poco conocimiento también. Este trabajo no pretende sustituir esta herramienta, sino mejorar de alguna forma la vida del proyecto, aquí se incluirá información que no es necesaria que la gestione el Gforge, como la propia documentación para la capacitación, la bitácora del proyecto, etc.

2.4 Información que se maneja

La información que se maneja en muchas ocasiones es similar para muchos proyectos y en otros casos tiene sus particularidades en dependencia del proyecto en cuestión. Entre la información con la que siempre se debe contar están los datos del personal involucrado en el proyecto y las tareas

que se deben realizar para el cumplimiento final del proyecto. También existen otras informaciones que son comunes a todos los proyectos en desarrollo, se puede mencionar en este apartado los documentos de estudio.

2.5 Propuesta de sistema

Teniendo en cuenta las necesidades existentes en el proyecto se propone la construcción de una plataforma web que permita el control de los procesos que se realizan sistemáticamente en el proyecto, permitiendo gestionar documentación, asignar y controlar el cumplimiento de tareas y mantener actualizado el estado de cada proyecto.

El objetivo principal del presente trabajo de diploma es desarrollar una plataforma de trabajo para el control de proyectos para lo que se propone la utilización de un Sistema de Gestión de Contenidos (Drupal) para montar la aplicación final. Dicha aplicación debe brindar información actualizada sobre el estado de cada proyecto en cualquier momento que el sitio sea visitado. Además de la información debe brindar una serie de servicios como: foro, gestión de usuarios y tareas, sección de descarga de materiales, también otras funciones mas orientadas a la información.

2.6 Modelo de Negocio

La utilización de un Modelo de Negocio pretende definir el contexto en que se desarrolla el sistema a automatizar, así como comprender de manera fácil la estructura y dinámica que tiene lugar en cada proyecto que se desarrolla en el Proyecto Portales_PHP, con el objetivo de vislumbrar los problemas actuales, obtener los requisitos por los que se debe regir el software a desarrollar e identificar las mejoras que se proponen con el mismo. El presente capítulo expone el modelo del negocio y a partir de este, con los requisitos funcionales y no funcionales identificados, se procede a la modelación del sistema que se propone.

Para realizar la modelación de los procesos se ha empleado el Lenguaje Unificado de Modelado (UML – Unified Model Language) que está respaldado por notaciones gráficas y un meta-modelo que ayuda a describir y diseñar sistemas de software, con el uso del Visual Paradigm, herramienta que facilita al usuario de un ambiente de trabajo amigable e intuitivo, y tiene la característica de ser un software no propietario.

2.6.1 Descripción de los procesos del negocio propuestos

Para tener una visión de cómo se desarrollan los procesos en el Proyecto Portales_PHP, se realizó un estudio que tuvo como objetivo lograr la modelación de este negocio lo cual se muestra a continuación.

Cuando un usuario (considerando usuario a cualquier persona que requiera o brinde información del o al proyecto) se presenta a la entidad para solicitar información de un proyecto en específico que está en desarrollo, es atendido por una persona encargada de ello (líder del proyecto), este busca el proyecto solicitado en el registro de proyectos(entidad que contiene la información sobre todos los proyectos existentes ya sean en funcionamiento o en proceso de realización), en caso que exista el proyecto solicitado, procede a brindar la información requerida. Si el proyecto que se ha pedido no se encuentra en desarrollo, se le informa al usuario que el proyecto solicitado no está en realización.

Por otra parte el usuario, como cliente, solicita al líder del proyecto la realización de un nuevo proyecto. Este comprueba, en el registro de proyectos, que exista personal disponible para ocuparse de la realización del proyecto y si existe la posibilidad, da alta al nuevo proyecto, con la información requerida y asigna al personal que trabajará en él. El líder del proyecto le informa al usuario que se puede realizar el nuevo proyecto y actualiza el registro de proyectos. En el caso que no haya personal disponible para el trabajo solicitado, se le informa al usuario que no se puede efectuar su solicitud.

La otra parte fundamental del negocio es la muestra de tareas. Cuando un usuario se presenta a la entidad para solicitar conocer si tiene tarea a realizar, es atendido por el líder del proyecto el cual busca en el registro de tareas (entidad que contiene la información sobre las tareas que tiene asignada cada usuario), en caso que exista tarea para el usuario en cuestión, procede a informar la misma. Si no posee, se le informa al usuario que no tiene tarea a realizar.

2.6.2 Descripción de los actores y trabajadores del negocio

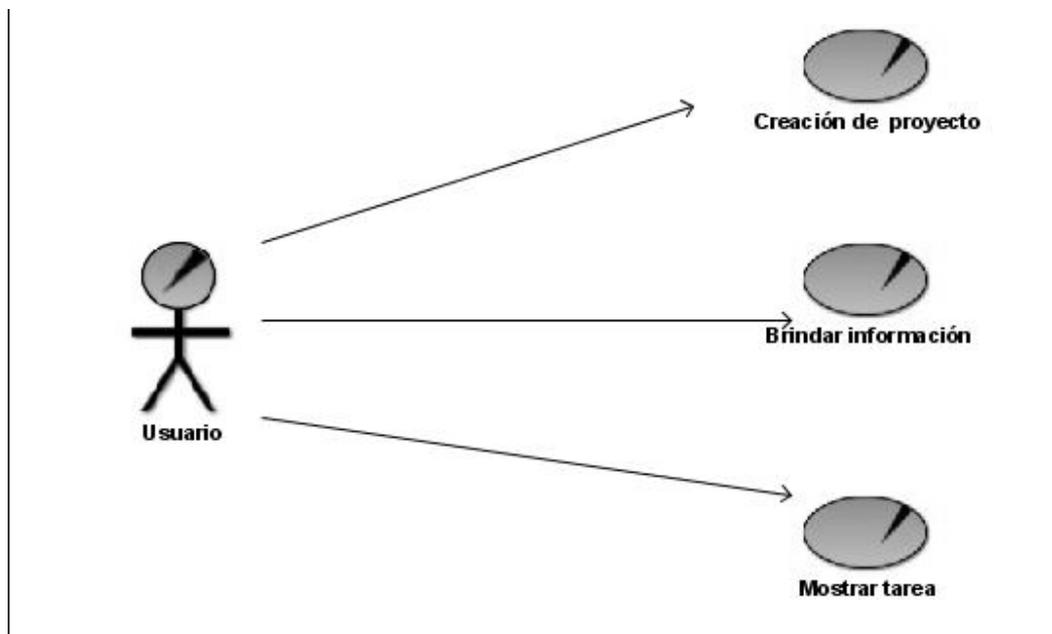
A continuación se detallan las personas que se relacionan con los procesos (actores y trabajadores del negocio). Los actores del negocio, o sea, aquellas personas, grupos, entidades, máquinas, etc. que interactúan con el ambiente del negocio, se han representado con nombres que expresan su rol dentro de él. Por otra parte, los trabajadores del negocio identificados, representan a aquellas personas que laboran en el proyecto y que se relacionan de algún modo con la gestión de las actividades.

Actores del negocio	Justificación
Usuario	El usuario puede ser cualquier persona que requiera o brinde información del o al proyecto.

Trabajadores del negocio	Justificación
Líder del proyecto	Atiende a los usuarios que desean conocer información sobre un proyecto o necesitan un servicio.

2.6.3 Diagrama de Casos de Uso del Negocio

Diagrama de casos de uso del negocio



2.6.4 Descripción de los casos de uso del negocio

A continuación se representan los casos de uso detalladamente con su respectivo diagrama de actividad.

• **Caso de uso:** Brindar información

Caso de Uso	Brindar información
Actores	Usuario
Trabajadores	Líder del proyecto
Resumen	El caso de uso se inicia cuando el usuario se presenta a solicitar información sobre un proyecto en específico que se está desarrollando. El líder del proyecto busca el proyecto en cuestión en el registro de proyectos, en caso de que exista procede a brindar la información deseada, en caso contrario se le informa al usuario que dicho proyecto no se está desarrollando.
Flujo Normal de Eventos	

Acción del Actor	Respuesta del Negocio
<p>1- El usuario solicita información sobre un portal.</p> <p>2- El usuario recibe la información y se retira.</p>	<p>1.1- El líder del proyecto busca en el registro de proyectos dicho portal.</p> <p>1.2- El líder del proyecto busca la información solicitada.</p> <p>1.3- El líder del proyecto muestra dicha información.</p>
Flujo Alternativo de Eventos	
Acción del Actor	Respuesta del Negocio
	<p>1.1- Si no se encuentra el portal el líder de proyecto informa que dicho portal no se realiza en el proyecto</p>

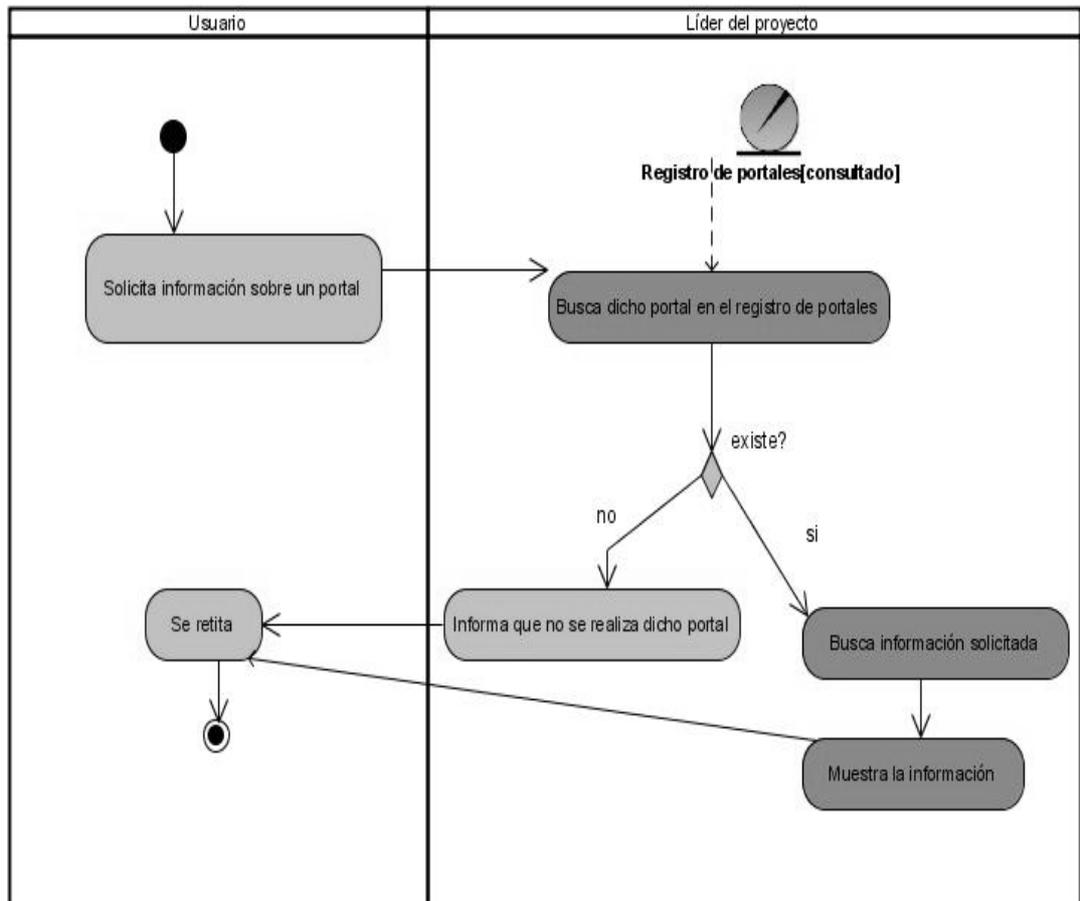


Ilustración 2: Diagrama de actividades del caso de uso Brindar información.

• Caso de uso: Creación de proyecto

Caso de Uso	Creación de proyecto
Actores	Usuario (Cliente)
Trabajadores	Líder del proyecto
Resumen	El caso de uso se inicia cuando el usuario (cliente) se presenta a solicitar que se desarrolle un proyecto nuevo. El líder del proyecto procede a dar alta al

	nuevo proyecto con su información inicial y asigna el personal que trabajará en el mismo.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Negocio
<p>1- El usuario solicita la realización de un nuevo proyecto.</p> <p>2- El usuario recibe confirmación y se retira</p>	<p>1.1- El líder del proyecto comprueba que exista capacidad en cuanto a personas para afrontar el proyecto en el registro de portales.</p> <p>1.2- El líder del proyecto da alta al nuevo proyecto.</p> <p>1.3- Asigna el nuevo proyecto al personal designado.</p> <p>1.4- El líder del proyecto informa que puede realizar el proyecto y actualiza el registro de proyectos.</p>
Flujo Alternativo de Eventos	
Acción del Actor	Respuesta del Negocio
	<p>1.1- Si no existe capacidad de personal informa que no puede afrontar la realización del proyecto.</p>

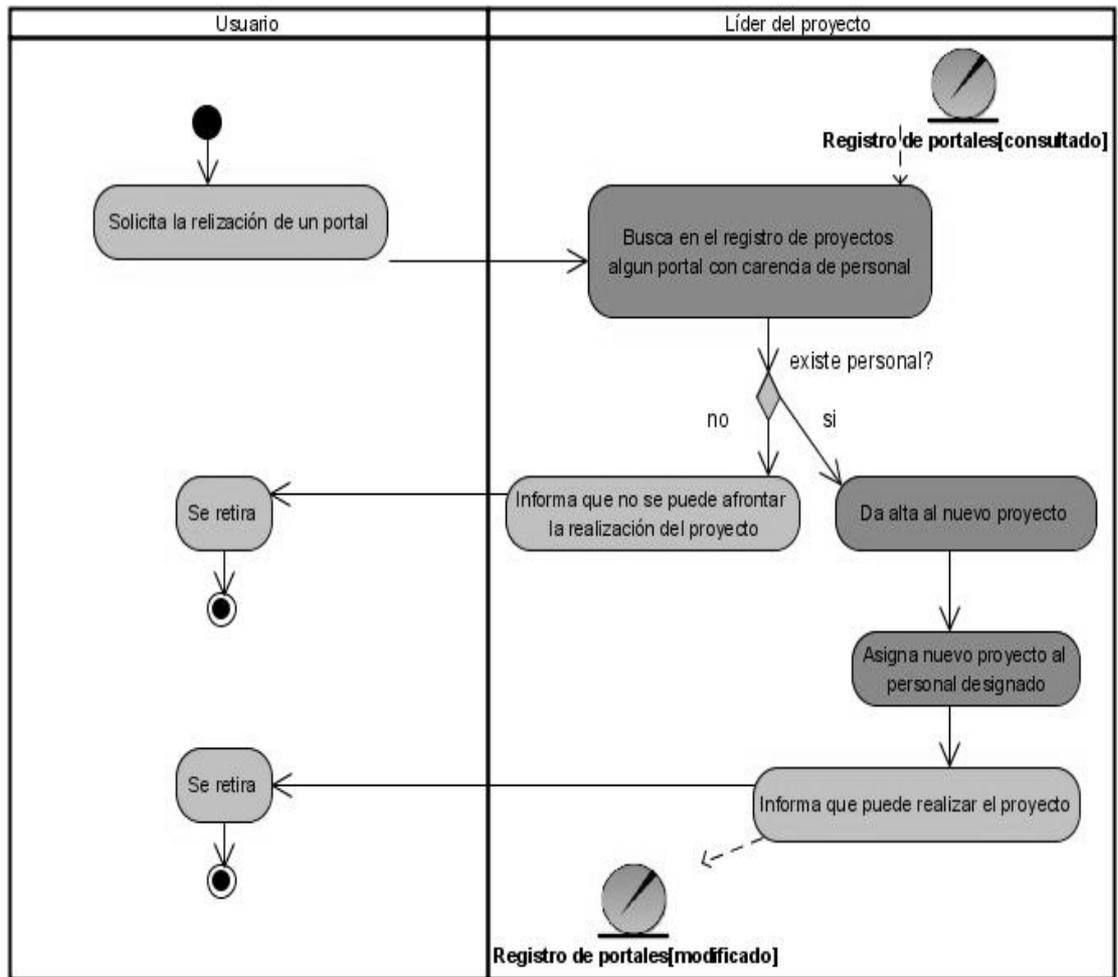


Ilustración 3: Diagrama de actividades del caso de uso Creación de proyecto.

•Caso de uso: Mostrar tarea

Caso de Uso	Mostrar tarea
Actores	Usuario
Trabajadores	Líder del proyecto

Resumen	El caso de uso se inicia cuando el usuario se presenta para preguntar si tiene alguna tarea asignada. El líder del proyecto busca en el registro de tareas el usuario en cuestión, en caso de que tenga asignada alguna tarea procede a informar la tarea que le corresponde, en caso contrario se le informa al usuario que no tiene tarea asignada.	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Negocio	
<p>1- El usuario pregunta si tiene alguna tarea.</p> <p>2- El usuario recibe la tarea y se retira.</p>	<p>1.1-El líder del proyecto busca en el registro de tareas dicho usuario.</p> <p>1.2-El líder del proyecto busca si dicho usuario tiene asignada alguna tarea.</p> <p>1.3-El líder del proyecto informa la tarea correspondiente.</p>	
Flujo Alterno de Eventos		
Acción del Actor	Respuesta del Negocio	
	<p>1.2- Si no tiene asignada ninguna tarea el líder de proyecto informa que por el momento no tiene tarea a realizar.</p>	

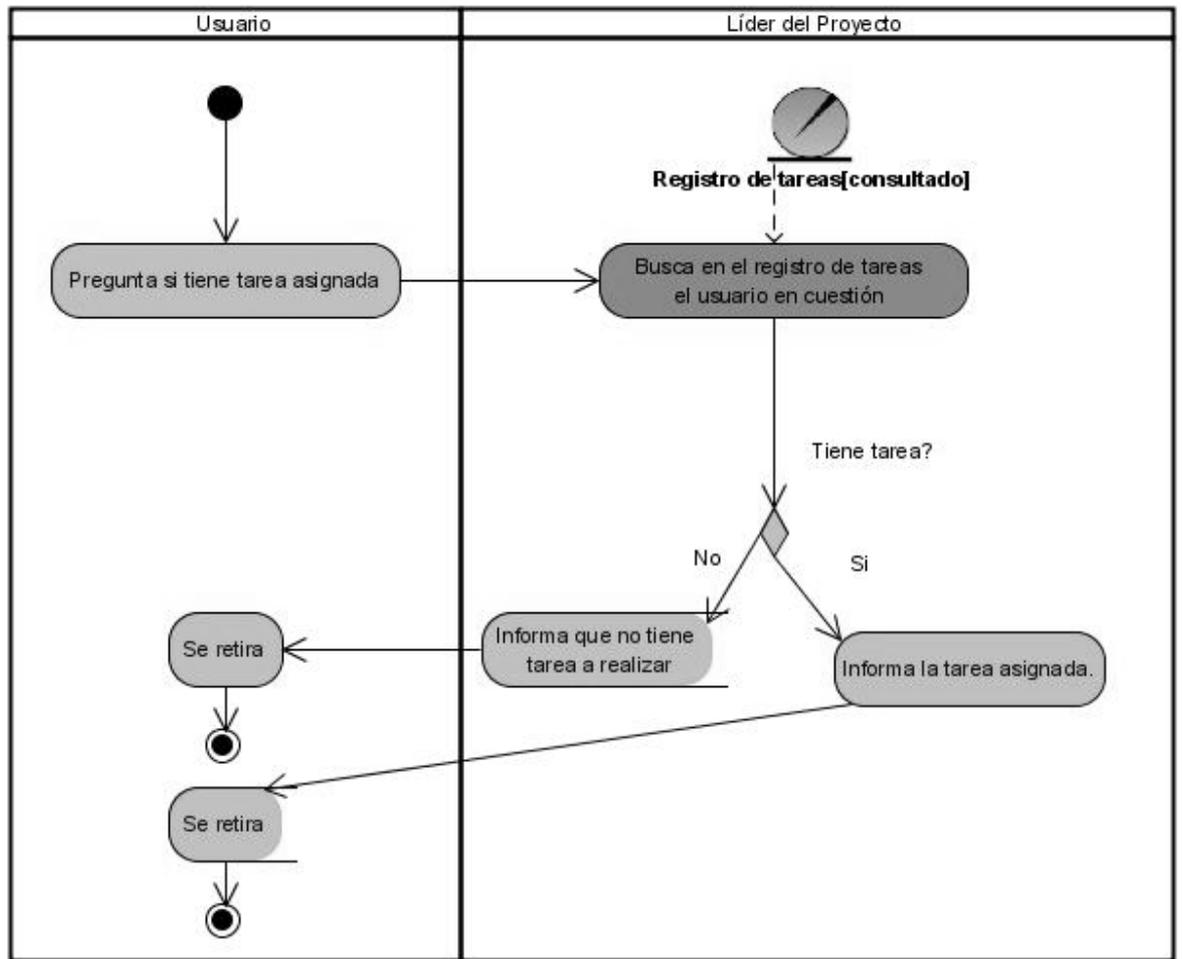
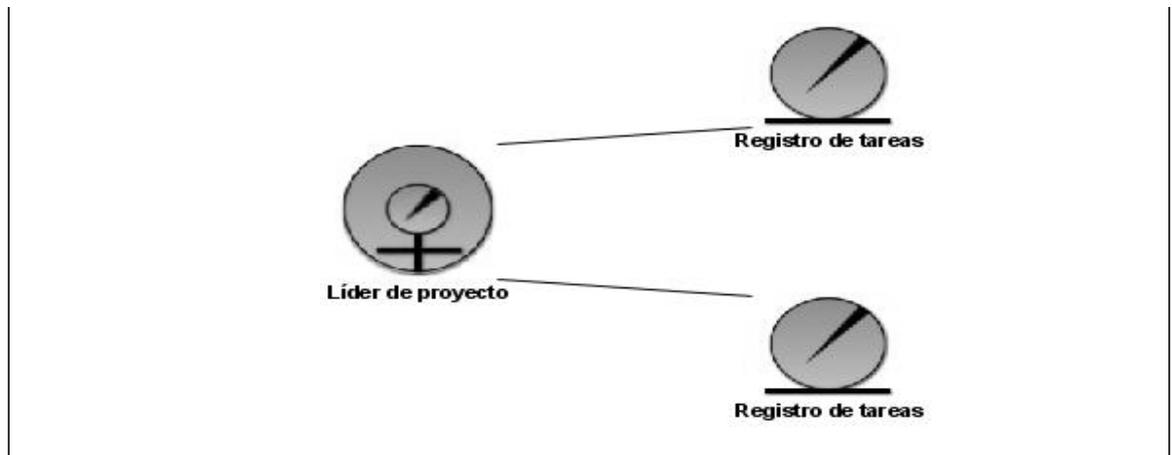


Ilustración 4: Diagrama de actividades del caso de uso Mostrar Tareas.

2.6.5 Diagrama de clases del modelo de objetos.

Diagrama de clases del modelo de objetos



2.7 Especificación de los requisitos de software.

2.7.1 Requisitos funcionales

Luego de haber analizado el modo en que se llevan a cabo las actividades vinculadas a la gestión del proyecto es posible identificar aquellas funcionalidades que son imprescindibles en el sistema a desarrollar. Estas se listan a continuación:

- 1- Intercambiar información
- 2- Registrar un nuevo proyecto
- 3- Gestionar proyecto
 - 3.1- Modificar proyecto
 - 3.2- Eliminar proyecto
- 4- Consultar información
- 5- Publicar noticia
- 6- Gestionar noticia
 - 6.1- Modificar noticia
 - 6.2- Eliminar noticia

7- Realizar búsqueda

7.1- Buscar estudiantes vinculados al proyecto

7.2 -Buscar proyectos que se desarrollan

7.3-Buscar documentos

8- Gestionar personal

8.1- Registrar personal

8.2 -Dar baja a personal vinculado al proyecto

8.3 -Modificar información del personal

9- Registrar documentación

10- Gestionar documentación

10.1-Modificar documentación necesaria al proyecto

10.2-Eliminar documentación necesaria al proyecto

11- Autenticar usuario

12- Registrar cliente

13- Gestionar cliente

13.1-Modificar información del cliente

13.2-Eliminar cliente

14- Registrar actividades

15- Gestionar actividades

15.1- Modificar actividades

15.2- Eliminar actividades

16- Asignar tarea

17- Gestionar tarea

17.1- Modificar tarea

17.2- Eliminar tarea

18- Consultar tarea

2.7.2 Requisitos no funcionales

Los requisitos no funcionales del sistema son propiedades o cualidades que el producto debe presentar. Son las características que hacen al producto atractivo, usable, rápido o confiable. La razón fundamental de que estos requisitos formen parte del producto es brindarle a este las características deseadas.

•Apariencia o interfaz externa:

- Fácil de utilizar tanto por los usuarios que accedan a él por primera vez como por los habituales.
- Interfaz sencilla.

•Usabilidad:

- La misma información debe ser accesible desde distintas partes del sistema, además su diseño debe ser tal que no complique la forma de trabajo.

•Rendimiento:

- El tiempo de respuesta debe ser corto porque el sistema es interactivo. Son pantallas dinámicas por lo que el acceso a la información debe ser rápido y de la manera más fácil posible.
- La base de datos debe estar normalizada, para garantizar el rendimiento óptimo de la misma.

•Soporte:

- El sistema deberá ser probado, instalado y configurado por los especialistas que también se ocuparán de su mantenimiento.

•Portabilidad:

- Este producto deberá poder funcionar en cualquier sistema operativo.

•Seguridad:

-El sistema debe ser capaz de trabajar por secciones, donde cada usuario tenga definido permisos y privilegios que definen las operaciones que puede realizar el mismo, las secciones deben limitar su tiempo de vida.

•**Software:**

- Necesita la aplicación un conjunto de software para funcionar: gestor de base de datos MySQL4, tener un servidor web Apache y contar con PHP4 instalado.

•**Hardware:**

- El servidor requiere capacidad para soportar una base de datos que debe ir creciendo y una velocidad para responder a las consultas que se realicen vía Web, se propone al menos 512 Mb de RAM, y al menos 5Gb de espacio de disco duro.

•**Políticos-culturales:**

-El sistema debe estar completamente identificado con los principales conceptos del campo de trabajo, para que el usuario lo asocie con la vida real. Debe estar identificado y responder a los principios e intereses de la sociedad.

•**Legales:**

-El empleo de este producto no deberá violar ninguna ley o licencia asociada a este o a terceros. Para su implementación se utiliza Drupal, producto bajo licencia GPL.

•**Confiabilidad:**

-La información que almacene, procese y genere el sistema debe ser confiable, además deben existir mecanismos de mantenimiento para garantizar la respuesta ante posibles fallos.

•**Ayuda y documentación en línea:**

- Es necesario que el sistema cuente con mensajes de ayuda que indiquen que operación realiza cada componente. El sistema contará con un mapa de navegación, el cual permitirá al usuario navegar por el sitio.

2.8 Definición de los Casos de Uso del Sistema.

2.8.1 Definición de los actores del sistema

Actores	Justificación
Líder del proyecto	Trabajador del proyecto que es el encargado de velar que se mantenga actualizado el portal de manera correcta.
Usuario registrado	Usuario que se encuentra accediendo al portal para buscar información.
Usuario anónimo	Usuario que accede al portal para buscar información pero no se registra.

2.8.2 Casos de uso del sistema

A continuación se presenta los casos de uso del sistema obtenidos teniendo en cuenta los requisitos funcionales del sistema.

CU-1	Intercambiar información
Actor	Usuario registrado
Descripción	El usuario accede a la sección del foro que presenta el portal y a través del mismo puede intercambiar con otros usuarios que participen en el mismo.
Referencia	RF 1, CU-11

CU-2	Registrar proyecto
Actor	Líder del proyecto
Descripción	El líder del proyecto accede a la sección de registrar un nuevo proyecto a realizar.
Referencia	RF 2, CU- 11, CU-12

CU-3	Gestionar proyecto
Actor	Líder del proyecto
Descripción	El líder del proyecto accede a la sección de proyectos y a través de la misma puede modificar o eliminar el proyecto seleccionado.
Referencia	CU- 2, RF 3, RF 3.1, RF 3.2, CU- 11

CU-4	Consultar información
Actor	Usuario anónimo
Descripción	El usuario anónimo solicita información que se encuentra en el portal.
Referencia	RF 4

CU-5	Publicar noticias
Actor	Líder del proyecto
Descripción	El líder del proyecto accede a la sección de crear noticia para introducir una nueva.
Referencia	RF 5, CU-11

CU-6	Gestionar noticias
Actor	Líder del proyecto
Descripción	El líder del proyecto accede a la sección noticias y en la opción editar puede modificar o eliminar la noticia seleccionada.
Referencia	CU-5, RF 6, RF 6.1, RF 6.2, CU-11

CU-7	Realizar búsquedas
Actor	Usuario anónimo
Descripción	El usuario solicita realizar una búsqueda en el portal.
Referencia	RF 7, RF 7.1,RF 7.2, RF 7.3

CU-8	Gestionar personal
Actor	Líder del proyecto
Descripción	El líder del proyecto accede a la sección de gestionar el personal vinculado al proyecto y puede registrar, eliminar o modificar sus datos.
Referencia	RF8, RF8.1, RF8.2, RF8.3, CU-11

CU-9	Gestionar documentación
Actor	Líder del proyecto
Descripción	El líder del proyecto accede a la sección documentación y en la opción editar puede modificar o eliminar la documentación seleccionada.
Referencia	CU- 10, RF 10, RF 10.1, RF 10.2, CU-11

CU-10	Registrar documentación
Actor	Líder del proyecto
Descripción	El líder del proyecto accede a la sección de crear documentación para introducir una.
Referencia	RF 9, CU-11

CU-11	Autenticar
Actor	Usuario registrado
Descripción	El usuario registrado entra al sistema el cual solicita autenticación para poder distinguir los privilegios de los usuarios. El usuario se autentica y el sistema comprueba que los datos sean válidos.
Referencia	RF 11

CU-12	Registrar cliente
Actor	Líder del proyecto
Descripción	El líder del proyecto accede a la sección de crear cliente.
Referencia	RF 12, CU-11

CU-13	Gestionar cliente
Actor	Líder del proyecto
Descripción	El líder del proyecto accede a la sección cliente y en la opción editar puede modificar o eliminar el cliente seleccionado.
Referencia	CU- 12, RF 13, RF 13.1, RF 13.2, CU-11

CU-14	Asignar tareas
Actor	Líder del proyecto
Descripción	El líder del proyecto accede a la sección crear tareas y se la asigna a un usuario determinado.
Referencia	CU-8, RF 16, CU-11

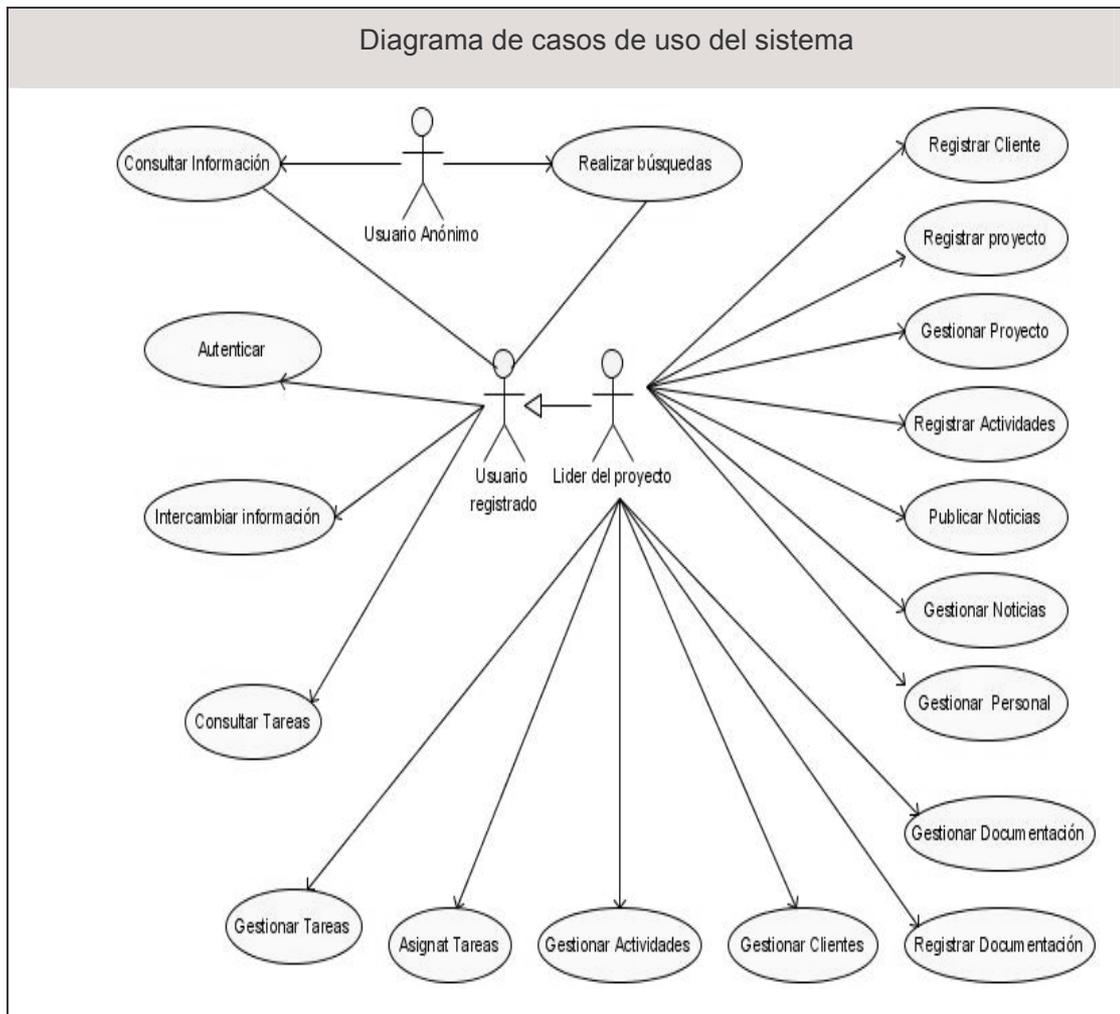
CU-15	Gestionar tareas
Actor	Líder del Proyecto
Descripción	El líder del proyecto accede a la sección tareas y en la opción editar puede modificar o eliminar la tarea seleccionada.
Referencia	CU- 14, RF 17, RF 17.1, RF 17.2, CU-11

CU-16	Registrar actividades
Actor	Líder del proyecto
Descripción	El líder del proyecto accede a la sección de crear actividad (bitácora).
Referencia	RF 14, CU- 2, CU-11

CU-17	Gestionar actividades
Actor	Líder del Proyecto
Descripción	El Líder del proyecto accede a la sección Proyecto, selecciona la actividad programada y en la opción editar puede modificar o eliminar la actividad seleccionada.
Referencia	CU- 16,RF 15, RF 15.1, RF 15.2, CU-11

CU-18	Consultar tareas
Actor	Usuario registrado
Descripción	El usuario registrado accede a la sección de consultar tareas para ver sus tareas.
Referencia	RF 4, CU-11, CU-2

2.8.3 Diagrama de casos de uso del sistema



2.8.4- Descripción de los casos de uso del sistema

El presente acápite será dedicado a la descripción de los casos de uso del sistema que fueron modelados en el diagrama. La descripción se realizará especificando el propósito general, el resumen de su funcionamiento, las condiciones que deben existir para que el mismo ocurra y el estado en que quedará el sistema luego de su instanciación.

Caso de uso	
1	Intercambiar información
Propósito	Permitirle al usuario registrado intercambiar información a través de un foro.
Actores: Usuario registrado	
Resumen: El caso de uso se inicia cuando el usuario registrado accede al sistema. Se muestra la página principal la cual contiene la sección para acceder al foro.	
Referencias	RF 1, CU-11
Precondiciones	Debe estar publicado un tema de debate.
Postcondiciones	Se envíe la opinión introducida.
Acción del actor	Respuesta del sistema
1-El usuario accede al sistema.	1.1- El sistema muestra la pagina principal.
2-El usuario registrado accede a la sección "Foro".	2.1- El sistema muestra la opción de autenticación y muestra los temas del foro.
3-El usuario registrado se autentica en el foro.	3.1- El sistema permite interactuar con el foro.
Flujo Alterno	
Acción del actor	Respuesta del sistema
	3.1- En caso de no introducir los datos correctamente el sistema muestra un mensaje de error de autenticación.

Caso de uso	
2	Registrar proyecto
Propósito	Permitirle al líder del proyecto crear un nuevo proyecto.
Actores: Líder del proyecto	
Resumen: El caso de uso se inicia cuando el líder del proyecto accede al sistema para registrar un nuevo proyecto.	
Referencias	RF 2, CU- 11, CU-12
Precondiciones	Estar autenticado en el sistema como líder proyecto.
Postcondiciones	Se cree un proyecto nuevo.
Acción del actor	Respuesta del sistema
1- El líder del proyecto accede a la página principal.	1.1- El sistema muestra la pagina principal.
2- El líder del proyecto accede a la sección de "Proyectos" del módulo crear	2.1- El sistema muestra los datos a

contenido nuevo.	introducir.
3- El líder del proyecto introduce los datos del proyecto que va a ser registrado y presiona el botón "Enviar".	3.1- El sistema introduce al nuevo proyecto.
Flujo Alternativo	
	2.1- En caso de producirse algún error al introducir los datos el sistema muestra un mensaje informándolo.

Caso de uso	
3	Gestionar proyecto
Propósito	Permitirle al líder del proyecto gestionar los datos de un nuevo proyecto.
Actores: Líder del proyecto	
Resumen: El caso de uso se inicia cuando el líder del proyecto accede al sistema para modificar los datos de un proyecto o eliminar un proyecto.	
Referencias	CU- 2, RF 3, RF 3.1, RF 3.2, CU- 11
Precondiciones	Estar autenticado en el sistema como líder proyecto.
Postcondiciones	Se modifique o se elimine algún proyecto.
Acción del actor	Respuesta del sistema
1- El líder del proyecto accede a la página principal.	1.1- El sistema muestra la pagina principal.
2- El líder del proyecto selecciona el proyecto que desea gestionar.	2.1- El sistema muestra el proyecto con sus datos.
3- El líder del proyecto accede a la sección "Editar".	3.1- El sistema muestra los campos a gestionar del proyecto. -Si el líder del proyecto decide modificar los datos de un proyecto ir a la sección "Modificar Información del Proyecto". -Si el líder del proyecto decide eliminar un proyecto, ir a la sección "Eliminar

	Proyecto”.
Flujo Alternativo	
Sección “Modificar Información del Proyecto”	
4- El líder del proyecto realiza las modificaciones deseadas y las envía a través del botón “Enviar”.	4.1- El sistema actualiza los datos.
Flujo Alternativo	
	4.1- En caso de producirse algún error al introducir los datos el sistema muestra un mensaje informándolo.
Sección “Eliminar Proyecto”	
4- El líder del proyecto presiona el botón “Eliminar”.	4.1- El sistema pide confirmación. 4.2- El sistema elimina el proyecto.

Caso de uso	
4	Consultar información
Propósito	Permitirle al usuario acceder a la información.
Actores: Usuario anónimo	
Resumen: El caso de uso se inicia cuando el usuario anónimo accede al sistema. Se muestra la página principal la cual permite acceder a las informaciones publicadas en el sitio.	
Referencias	RF 4
Precondiciones	Debe existir información publicada.
Postcondiciones	Se debe mostrar la información.
Acción del actor	Respuesta del sistema
1- El usuario anónimo accede a la página principal.	1.1- El sistema muestra la pagina principal.
2- El usuario anónimo accede a la sección en la cual se encuentre la información deseada.	2.1- El sistema muestra la información en cuestión.
Flujo Alterno	
Acción del actor	Respuesta del sistema

--	--

Caso de uso	
5	Publicar noticia
Propósito	Permitirle al líder del proyecto publicar noticias en el sistema.
Actores: Líder del proyecto	
Resumen: El caso de uso se inicia cuando el líder del proyecto accede al sistema. Se muestra la página que permite acceder a la sección para publicar noticias.	
Referencias	RF 5, CU-11
Precondiciones	Estar autenticado en el sistema como líder proyecto.
Postcondiciones	Se cree una noticia.
Acción del actor	Respuesta del sistema
1- El líder del proyecto accede a la página principal.	1.1- El sistema muestra la página principal.
2- El líder del proyecto accede a la sección "noticias" del módulo crear contenido nuevo.	2.1- El sistema muestra los campos a llenar.
3- El líder del proyecto introduce los datos a publicar y presiona el botón "Enviar".	3.1- El sistema introduce la nueva noticia.
Flujo Alterno	
Acción del actor	Respuesta del sistema
	3.1- En caso de producirse algún error al introducir los datos el sistema muestra un mensaje informándolo.

Caso de uso	
6	Gestionar noticias
Propósito	Permitirle al líder del proyecto gestionar los datos de una noticia.
Actores: Líder del proyecto	
Resumen: El caso de uso se inicia cuando el líder del proyecto accede al sistema para modificar los datos de una noticia o eliminarla.	
Referencias	RF 5, RF 6, RF 6.1, RF 6.2, CU-11, CU-5
Precondiciones	Estar autenticado en el sistema como líder proyecto. Exista noticia publicada para gestionar.
Postcondiciones	Se modifique una noticia o se elimine.
Acción del actor	Respuesta del sistema

1- El líder del proyecto accede a la página principal.	1.1- El sistema muestra la pagina principal.
2- El líder del proyecto accede a la sección "Noticia" del menú principal.	2.1- El sistema muestra las noticias publicadas.
3- El líder del proyecto selecciona la noticia que desea gestionar.	3.1- El sistema muestra la noticia con sus datos.
4- El líder del proyecto accede a la sección "Editar".	4.1- El sistema muestra los campos a gestionar de la noticia. -Si el líder del proyecto decide modificar los datos de una noticia, ir a la sección "Modificar Noticia". -Si el líder del proyecto decide eliminar una noticia, ir a la sección "Eliminar Noticia".
Flujo Alternativo	
Sección "Modificar Noticia"	
5- El líder del proyecto realiza las modificaciones deseadas y las envía a través del botón "Enviar".	5.1- El sistema actualiza los datos.
Flujo Alternativo	
	5.1- En caso de producirse algún error al introducir los datos el sistema muestra un mensaje informándolo.
Sección "Eliminar Noticia"	
5- El líder del proyecto presiona el botón "Eliminar".	5.1- El sistema pide confirmación. 5.2- El sistema elimina la noticia.

Caso de uso	
7	Realizar búsquedas
Propósito	Permitirle al usuario anónimo realizar búsquedas en el sistema.

Actores: Usuario anónimo	
Resumen: El caso de uso se inicia cuando el usuario anónimo accede al sistema. Se muestra la página principal la cual contiene la sección de búsquedas.	
Referencias	RF 7, RF 7.1, RF 7.2, RF 7.3
Precondiciones	Debe existir información publicada del contenido que se desee buscar.
Postcondiciones	Debe visualizarse el resultado de la búsqueda.
Acción del actor	Respuesta del sistema
1- El usuario registrado accede a la página principal.	1.1- El sistema muestra la pagina principal.
2- El usuario registrado accede a la sección de búsqueda e introduce la búsqueda a realizar y presiona el botón "Buscar".	2.1- El sistema muestra el resultado.
Flujo Alternativo	
Acción del actor	Respuesta del sistema
	2.1 En caso que no exista resultado para la búsqueda el sistema muestra un mensaje informándolo.

Caso de uso	
8	Gestionar personal
Propósito	Permitir registrar, eliminar y modificar personal vinculado al proyecto.
Actores: Líder del proyecto	
Resumen: El caso de uso se inicia cuando el líder del proyecto accede al sistema para registrar, eliminar o modificar los datos de las personas vinculadas al proyecto.	
Referencias	RF8, RF8.1, RF8.2, RF8.3, CU-11
Precondiciones	Estar autenticado en el sistema como líder proyecto.
Postcondiciones	Se debe registrar, eliminar o modificar personal.
Acción del actor	Respuesta del sistema
1- El líder del proyecto accede al sistema.	1.1- El sistema muestra la pagina principal.

<p>2. El líder del proyecto accede a la sección de “usuario” del módulo administrar.</p>	<p>2.1- El sistema muestra la sección usuario</p> <p>-Si el líder del proyecto decide registrar personal, ir a la sección “Registrar personal”.</p> <p>-Si el líder del proyecto decide modificar los datos de una persona vinculada al proyecto, ir a la sección "Modificar Información del Personal".</p> <p>-Si el líder del proyecto decide eliminar a una persona vinculada al proyecto, ir a la sección "Dar de baja a personal vinculado al proyecto”.</p>
<p>Sección “Registrar Personal”</p>	
<p>1- El líder del proyecto accede a la opción de “añadir usuario”.</p>	<p>1.1- El sistema muestra los campos a llenar.</p>
<p>2- Introduce los datos de la persona que va a ser registrada y presiona el botón “Enviar”.</p>	<p>2.1- El sistema introduce a la nueva persona.</p>
<p>Flujo Alternativo</p>	
	<p>2.1- En caso de producirse algún error al introducir los datos el sistema muestra un mensaje informándolo.</p>
<p>Sección “Modificar Información del personal”</p>	
<p>1- El líder del proyecto accede a la opción de “lista”.</p>	<p>1.1-El sistema muestra las personas vinculadas al proyecto.</p>
<p>2-El líder del proyecto selecciona la persona a la cual le va a modificar los datos y selecciona la opción “editar”.</p>	<p>2.1- El sistema muestra los campos de datos a modificar.</p>
<p>3-El líder del proyecto realiza las modificaciones deseadas y presiona el botón “Enviar”.</p>	<p>3.1- El sistema actualiza los datos.</p>
<p>Flujo Alternativo</p>	

	3.1- En caso de producirse algún error al introducir los datos el sistema muestra un mensaje informándolo.
Sección “Dar de baja a personal vinculado al proyecto”	
1- El líder del proyecto accede a la opción de “lista”.	1.1- El sistema muestra las personas vinculadas al proyecto.
2- El líder del proyecto selecciona la persona que va a eliminar y selecciona la opción “editar”.	2.1- El sistema muestra los campos de datos a modificar.
3- El líder del proyecto selecciona el botón “Eliminar”.	3.1- El sistema pide confirmación. 3.2- El sistema elimina a la persona.

Caso de uso	
9	Gestionar documentación
Propósito	Permitir eliminar y modificar documentación necesaria al proyecto.
Actores	Líder del proyecto
Resumen:	El caso de uso se inicia cuando el líder del proyecto accede al sistema para eliminar o modificar los datos de los documentos del proyecto.
Referencias	CU- 10, RF 10, RF 10.1, RF 10.2, CU-11
Precondiciones	Estar autenticado en el sistema como líder proyecto. Exista documentación para modificar.
Postcondiciones	Se debe eliminar o modificar documentación.
Acción del actor	Respuesta del sistema
1- El líder del proyecto accede al sistema.	1.1-El sistema muestra la pagina principal.
2. El líder del proyecto accede a la sección “Documentación” del menú principal.	2.1- El sistema muestra los documentos publicados.
3- El líder del proyecto selecciona el documento que desea gestionar.	3.1- El sistema muestra el documento con sus datos.
4- El líder del proyecto accede a la sección “Editar”.	4.1- El sistema muestra los campos a gestionar del documento. -Si el líder del proyecto decide modificar alguna documentación, ir a la sección "Modificar documentación". -Si el líder del proyecto decide eliminar alguna documentación ir a la sección

	"Eliminar documentación".
Sección "Modificar documentación"	
5- El líder del proyecto realiza las modificaciones deseadas y las envía a través del botón "Enviar".	5.1- El sistema actualiza los datos.
Flujo Alternativo	
	2.1- En caso de producirse algún error al introducir los datos el sistema muestra un mensaje informándolo.
Sección "Eliminar documentación"	
5- El líder del proyecto presiona el botón "Eliminar".	5.1- El sistema pide confirmación. 5.2- El sistema elimina el documento.

Caso de uso	
10	Registrar documentación.
Propósito	Permitirle al líder del proyecto registrar una nueva documentación.
Actores: Líder del proyecto	
Resumen: El caso de uso se inicia cuando el líder del proyecto accede al sistema para introducir un nuevo documento.	
Referencias	RF 9, CU-11
Precondiciones	Estar autenticado en el sistema como líder proyecto.
Postcondiciones	Se debe registrar una nueva documentación.
Acción del actor	Respuesta del sistema
1- El líder del proyecto accede al sistema.	1.1- El sistema muestra la pagina principal.
2- El líder del proyecto accede a la sección "documentos del proyecto" del módulo crear contenido nuevo.	2.1- El sistema muestra los campos a llenar.
3- El líder del proyecto introduce los datos a publicar y presiona el botón "Enviar".	3.1- El sistema introduce el nuevo documento.
Flujo Alternativo	

	3.1- En caso de producirse algún error al introducir los datos el sistema muestra un mensaje informándolo.
--	--

Caso de uso	
11	Autenticar
Propósito	Verificar que el usuario registrado pertenezca al sistema para poder otorgarle los permisos correspondientes.
Actores: Usuario registrado	
Resumen: El caso de uso se inicia cuando el usuario registrado accede al sistema para entrar sus datos y autenticarse en el mismo. El sistema comprueba que los datos entrados sean correctos para permitir al líder del proyecto hacer uso de sus privilegios.	
Referencias	RF 11
Precondiciones	Se deben introducir los datos correctamente.
Postcondiciones	Se habilitan las funcionalidades según los privilegios que tenga.
Acción del actor	Respuesta del sistema
1- El usuario registrado introduce sus datos (usuario y contraseña) y los envía.	1.1- El sistema comprueba los datos entrados. 1.2- Muestra las áreas a las que tiene permiso acceder.
Flujo alternativo	
Acción del actor	Respuesta del sistema
	1.2- En caso de no ser correctos los datos entrados el sistema muestra un mensaje de error y vuelve al paso 1 del flujo básico.

Caso de uso	
12	Registrar cliente
Propósito	Permite registrar un nuevo cliente al proyecto creado.
Actores: Líder del proyecto	
Resumen: El caso de uso se inicia cuando el líder del proyecto accede a la sección de registrar cliente al proyecto.	

Referencias	RF 12, CU-11	
Precondiciones	Estar autenticado en el sistema como líder proyecto.	
Postcondiciones	Se debe registrar un cliente.	
Acción del actor	Respuesta del sistema	
1- El líder del proyecto accede al sistema.	1.1- El sistema muestra la página principal.	
2- El líder del proyecto accede a la sección de "clientes" del módulo crear contenido nuevo.	2.1- El sistema muestra los campos a llenar.	
3- Introduce los datos del cliente que van a ser registrados y presiona el botón "Enviar".	3.1- El sistema introduce el nuevo cliente.	
Flujo alternativo		
	3.1- En caso de producirse algún error al introducir los datos el sistema muestra un mensaje informándolo.	

Caso de uso		
13	Gestionar clientes	
Propósito	Permitir eliminar y modificar clientes del proyecto.	
Actores	Líder del proyecto	
Resumen:	El caso de uso se inicia cuando el líder del proyecto accede al sistema para eliminar o modificar los datos de algún cliente del proyecto.	
Referencias	CU- 12, RF 13, RF 13.1, RF 13.2, CU-11	
Precondiciones	Estar autenticado en el sistema como líder proyecto. Debe existir un cliente al cual gestionar.	
Postcondiciones	Se debe modificar o eliminar un cliente.	
Acción del actor	Respuesta del sistema	
1- El líder del proyecto accede al sistema.	1.1- El sistema muestra la pagina principal.	
2. El líder del proyecto accede a la sección "contenido" del módulo administrar.	2.1- El sistema muestra todos los contenidos publicados.	

3- El líder del proyecto selecciona clientes en el campo "tipo" y lo envía a través del botón "filtrar".	3.1- El sistema muestra todos los clientes publicados.
4- El líder del proyecto presiona el botón "Editar" correspondiente al cliente que desee.	4.1- El sistema muestra los campos a gestionar del cliente. -Si el líder del proyecto decide modificar algún dato de un cliente, ir a la sección "Modificar Cliente". -Si el líder del proyecto decide eliminar algún cliente ir a la sección "Eliminar Cliente".
Sección "Modificar Cliente"	
5- El líder del proyecto realiza las modificaciones deseadas y las envía a través del botón "Enviar".	5.1- El sistema actualiza los datos.
Flujo Alternativo	
	2.1- En caso de producirse algún error al introducir los datos el sistema muestra un mensaje informándolo.
Sección "Eliminar Cliente"	
5- El líder del proyecto presiona el botón "Eliminar".	5.1- El sistema pide confirmación. 5.2- El sistema elimina el cliente.

Caso de uso	
14	Asignar tarea
Propósito	Permite asignar tareas
Actores:	Líder del proyecto
Resumen:	El caso de uso se inicia cuando el líder del proyecto accede al sistema para registrar una tarea. El sistema comprueba que todos los datos estén completados.
Referencias	CU-8, RF 16, CU-11
Precondiciones	Estar autenticado en el sistema como líder proyecto.
Postcondiciones	Se debe registrar una tarea.
Acción del actor	Respuesta del sistema
1- El líder del proyecto accede al sistema.	1.1- El sistema muestra la página principal.

2- El líder del proyecto accede a la sección de "tarea" del módulo crear contenido nuevo.	2.1- El sistema muestra los campos a llenar.
3- Introduce la tarea que va a ser registrada y presiona el botón "Enviar".	3.1- El sistema introduce la nueva tarea.
Flujo alternativo	
Acción del Actor	Respuesta del sistema
	3.1- En caso de producirse algún error al introducir los datos el sistema muestra un mensaje informándolo.

Caso de uso	
15	Gestionar tareas.
Propósito	Permitir eliminar y modificar tareas.
Actores	Líder del proyecto.
Resumen:	El caso de uso se inicia cuando el líder del proyecto accede al sistema para eliminar o modificar los datos de alguna tarea.
Referencias	CU- 14, RF 17, RF 17.1, RF 17.2, CU-11
Precondiciones	Estar autenticado en el sistema como líder proyecto. Debe existir una tarea a la cual gestionar.
Postcondiciones	Se debe modificar o eliminar una tarea.
Acción del actor	Respuesta del sistema
1- El líder del proyecto accede al sistema.	1.1- El sistema muestra la pagina principal.
2- El líder del proyecto accede a la sección "contenido" del módulo administrar.	2.1- El sistema muestra todos los contenidos publicados.
3- El líder del proyecto selecciona tarea en el campo "tipo" y lo envía a través del botón "filtrar".	3.1- El sistema muestra todas las tareas publicadas.

4- El líder del presiona el botón "Editar" correspondiente a la tarea que desee.	2.1- El sistema muestra los campos a gestionar de la tarea. -Si el líder del proyecto decide modificar alguna tarea, ir a la sección "Modificar Tarea". -Si el líder del proyecto decide eliminar alguna tarea ir a la sección "Eliminar Tarea".
Sección "Modificar Tarea"	
5- El líder del proyecto realiza las modificaciones deseadas y las envía a través del botón "Enviar".	5.1- El sistema actualiza los datos.
Flujo Alternativo	
	5.1- En caso de producirse algún error al introducir los datos el sistema muestra un mensaje informándolo.
Sección "Eliminar Tarea"	
5- El líder del proyecto presiona el botón "Eliminar".	5.1- El sistema pide confirmación. 5.2- El sistema elimina la tarea.

Caso de uso	
16	Registrar actividades
Propósito	Permite registrar actividades para un proyecto que haya sido creado.
Actores: Líder del proyecto	
Resumen: El caso de uso se inicia cuando el líder del proyecto accede al sistema para registrar una nueva actividad (bitácora).	
Referencias	RF 14, CU- 2, CU-11
Precondiciones	Estar autenticado en el sistema como líder proyecto.
Postcondiciones	Se registre una nueva actividad.
Acción del actor	Respuesta del sistema
1- El líder del proyecto accede al sistema.	1.1- El sistema muestra la página principal.

2- El líder del proyecto accede a la sección de "bitácora" del módulo crear contenido nuevo.	2.1- El sistema muestra los campos a llenar.
3- Introduce las actividades que van a ser registradas y presiona el botón "Enviar".	3.1- El sistema introduce la nueva actividad.
Flujo alternativo	
Acción del Actor	Respuesta del sistema
	3.1- En caso de producirse algún error al introducir los datos el sistema muestra un mensaje informándolo.

Caso de uso	
17	Gestionar actividades.
Propósito	Permitir eliminar y modificar actividades.
Actores	Líder del proyecto
Resumen: El caso de uso se inicia cuando el líder del proyecto accede al sistema para eliminar o modificar los datos de alguna actividad (bitácora).	
Referencias	CU- 16, RF 15, RF 15.1, RF 15.2, CU-11
Precondiciones	Estar autenticado en el sistema como líder proyecto. Debe existir una actividad a la cual gestionar.
Postcondiciones	Se debe modificar o eliminar una actividad.
Acción del actor	Respuesta del sistema
1- El líder del proyecto accede al sistema.	1.1- El sistema muestra la pagina principal.
2- El líder del proyecto accede a la sección "contenido" del módulo administrar.	2.1- El sistema muestra todos los contenidos publicados.
3- El líder del proyecto selecciona bitácora en el campo "tipo" y lo envía a través del botón "filtrar".	3.1- El sistema muestra todas las actividades publicadas.
4- El líder proyecto presiona el botón "Editar" correspondiente a la actividad que desee.	2.1- El sistema muestra los campos a gestionar de la actividad. -Si el líder del proyecto decide modificar alguna actividad, ir a la sección "Modificar Actividad". -Si el líder del proyecto decide eliminar

	alguna actividad ir a la sección "Eliminar Actividad".
Sección "Modificar Actividad"	
5- El líder del proyecto realiza las modificaciones deseadas y las envía a través del botón "Enviar".	5.1- El sistema actualiza los datos.
Flujo Alternativo	
	5.1- En caso de producirse algún error al introducir los datos el sistema muestra un mensaje informándolo.
Sección "Eliminar Actividad"	
5- El líder del proyecto presiona el botón "Eliminar".	5.1- El sistema pide confirmación. 5.2- El sistema elimina la actividad.

Caso de uso	
18	Consultar tareas
Propósito	Permitirle al usuario registrado acceder a la información de las tareas.
Actores: Usuario registrado	
Resumen: El caso de uso se inicia cuando el usuario registrado accede al sistema. Se muestra la página principal la cual permite acceder a las tareas publicadas en el sitio.	
Referencias	RF 4, CU-11, CU-2
Precondiciones	Debe existir tareas publicadas El usuario debe estar registrado.
Postcondiciones	Se debe mostrar la tarea.
Acción del actor	Respuesta del sistema
1- El usuario registrado accede a la página principal.	1.1- El sistema muestra la pagina principal.

2- El usuario registrado accede a la sección "Tareas" en el menú principal.	2.1- El sistema muestra las tareas a realizar.
---	--

2.9 Conclusiones

A través del desarrollo del presente capítulo se han descrito los procesos que ocurren en el Proyecto Portales_PHP. Con este estudio se ha logrado la comprensión de las actividades que se realizan en el negocio y se han analizado las reglas que son parte indispensable de este y que deberá cumplir el sistema. Del mismo modo se ha realizado un estudio de las funciones que debe desarrollar la aplicación a fin de lograr los objetivos para los cuales se implementará. Ha constituido también un resultado muy importante de este capítulo la identificación de los casos de uso del sistema y su descripción, puntos que guiarán la posterior construcción del software propuesto.

Capítulo III. ANALISIS Y DISEÑO DEL SISTEMA

3.1 Introducción

Con las actividades examinadas en los capítulos precedentes se proporciona una vista interna del sistema. Se refinan los requisitos y se estructuran en base a clases. Luego en este capítulo se continúa refinando el sistema, hasta obtener los objetos que interactúan para cumplir los requisitos funcionales y no funcionales antes definidos.

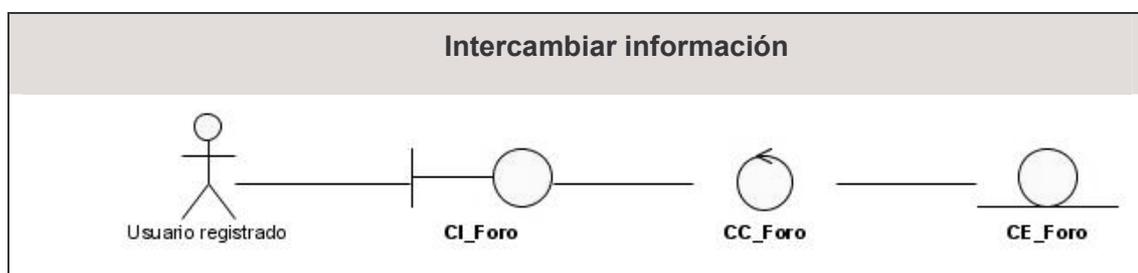
Con todo el conjunto de diagramas que a continuación se muestran, se pretende crear una base para el modelo de implementación.

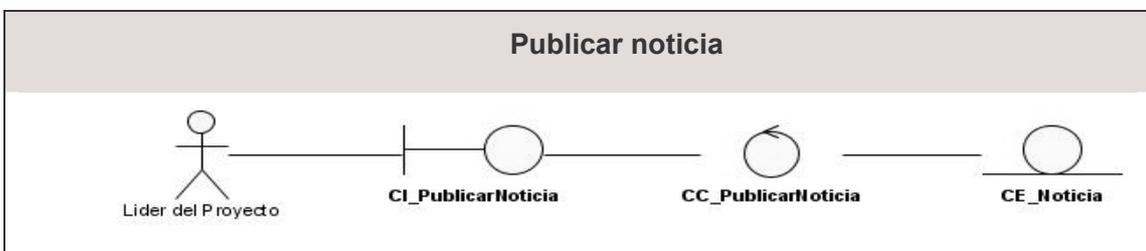
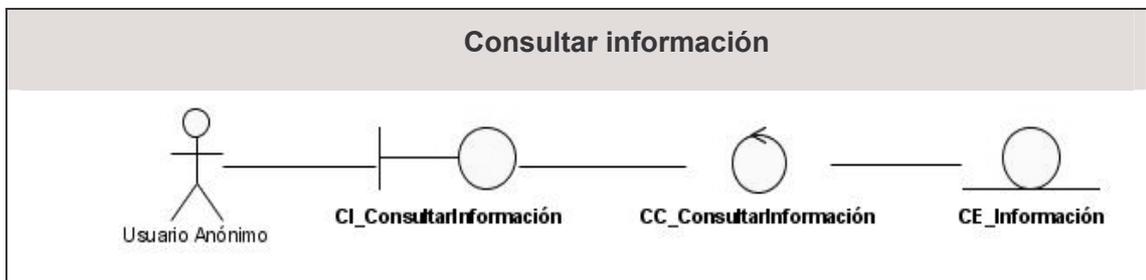
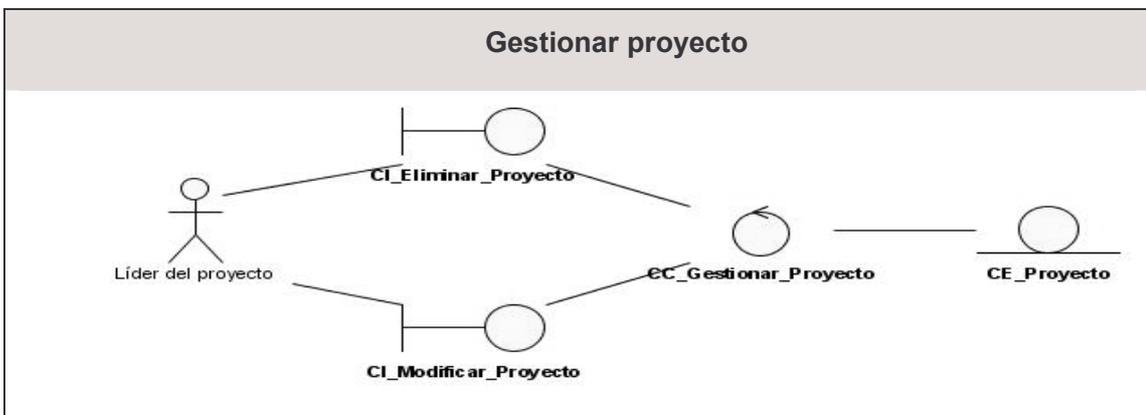
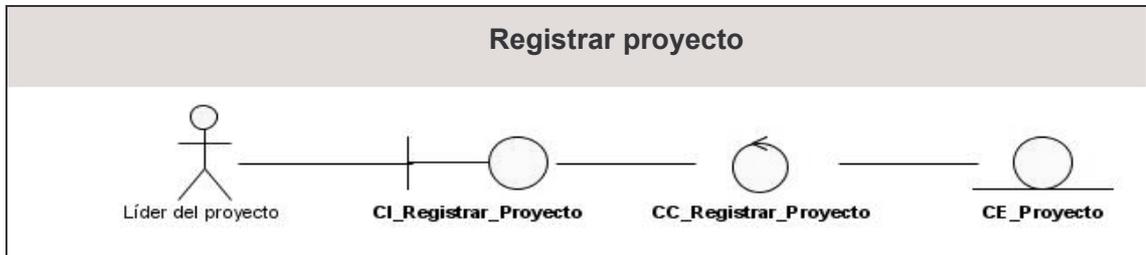
En el presente capítulo se desarrolla el flujo de trabajo de Análisis y Diseño del sistema. Se presenta el modelo de clases de análisis. Se muestran los diagramas de colaboración por cada realización de casos de uso, así como el diagrama de clases del diseño. Además se realiza en este capítulo la descripción de las clases, se diseña la base de datos y se describen las tablas.

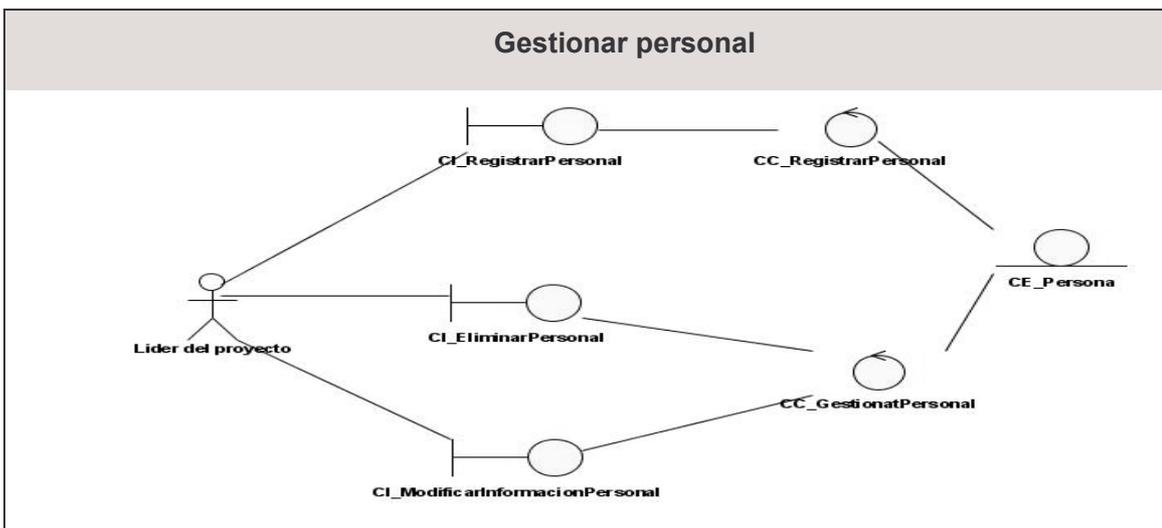
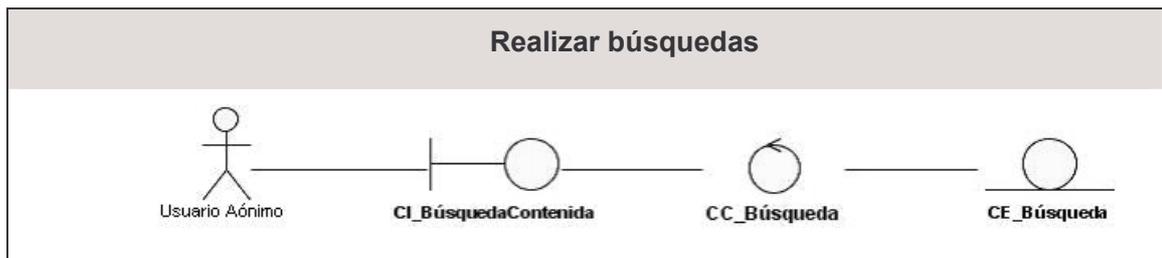
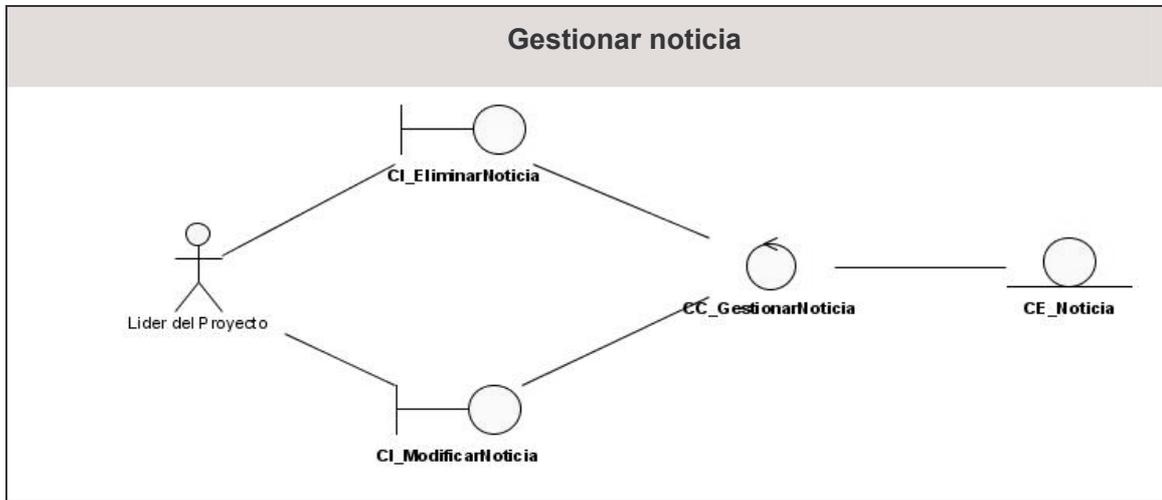
3.2 Análisis

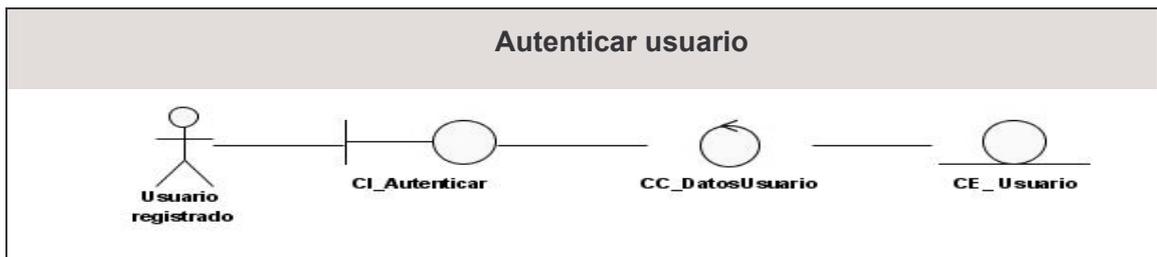
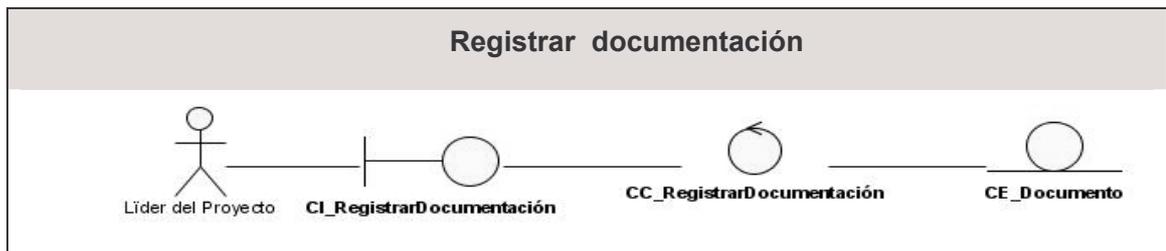
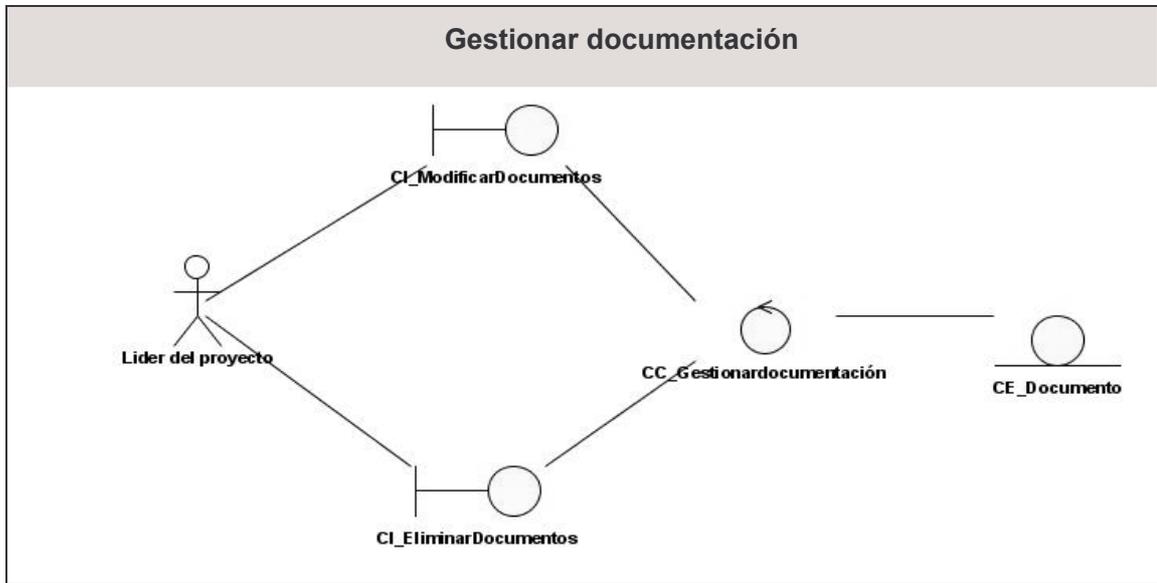
El modelo de análisis es una aproximación al modelo del diseño, es el resultado de la actividad de analizar los casos de uso. A continuación se presenta el modelo de análisis correspondiente a los casos de usos descritos.

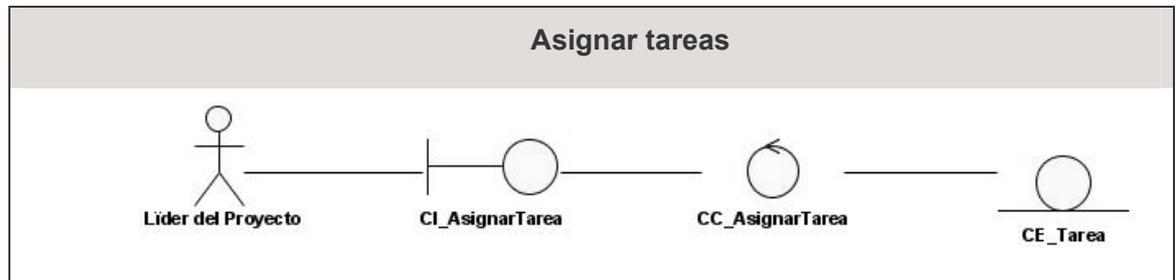
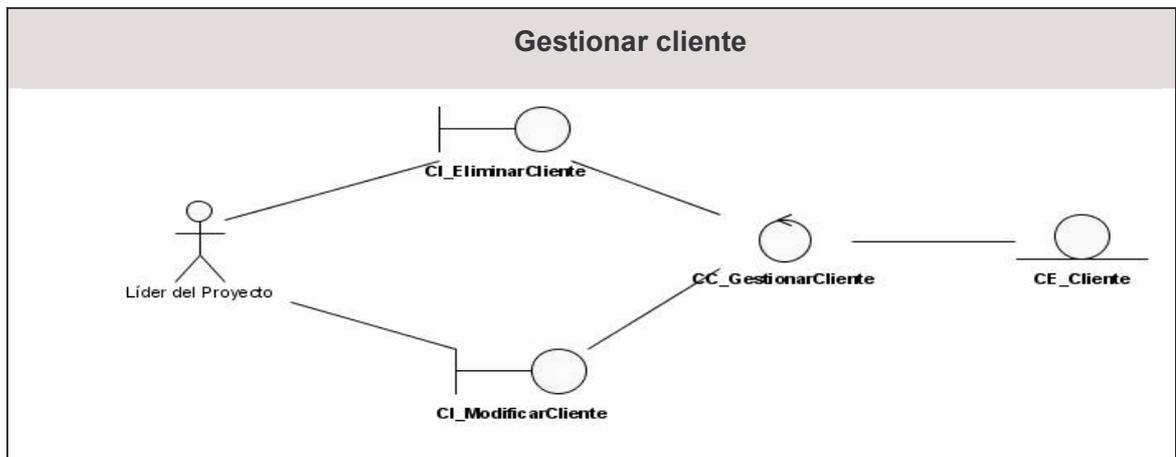
3.2.1 Diagramas de clases de análisis

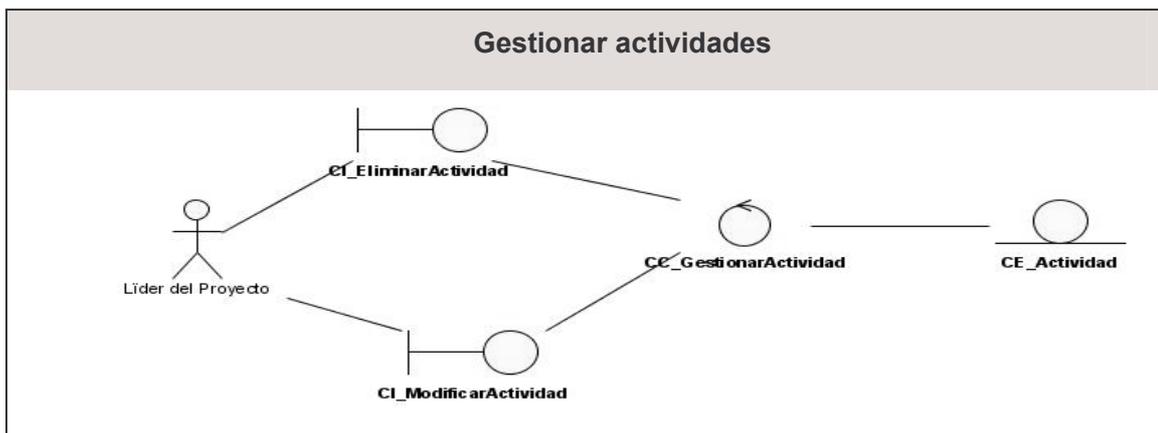
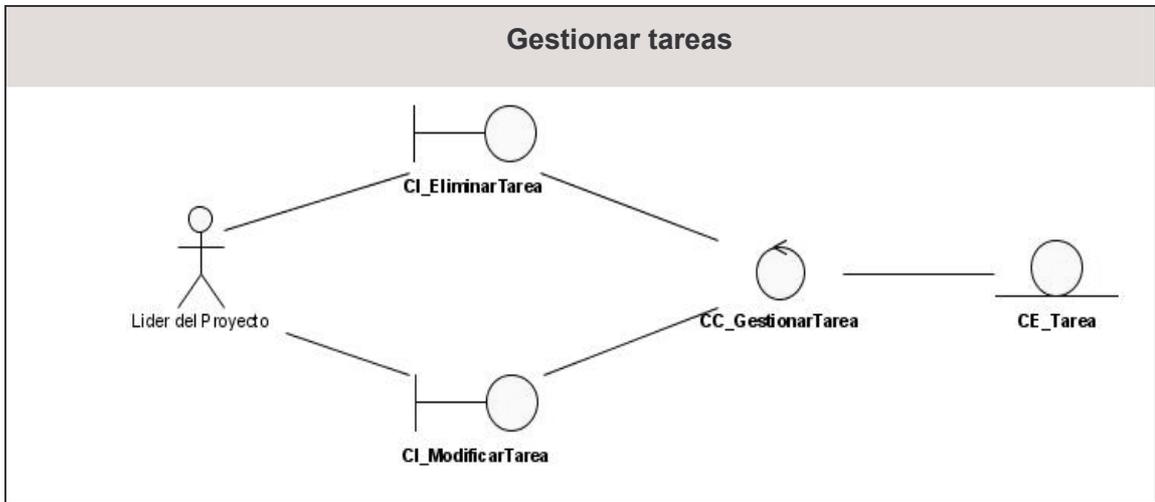


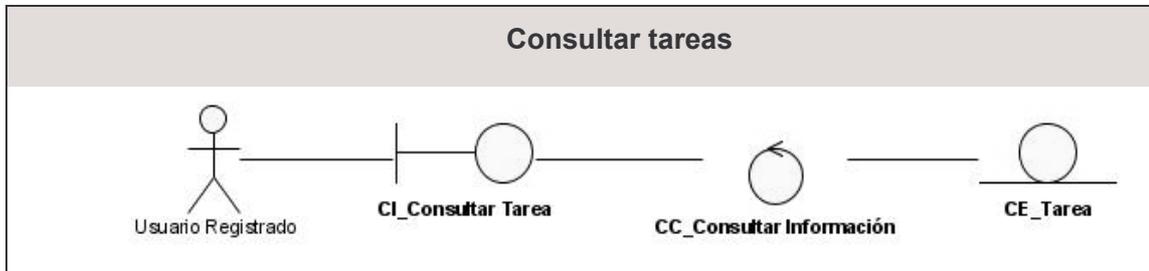




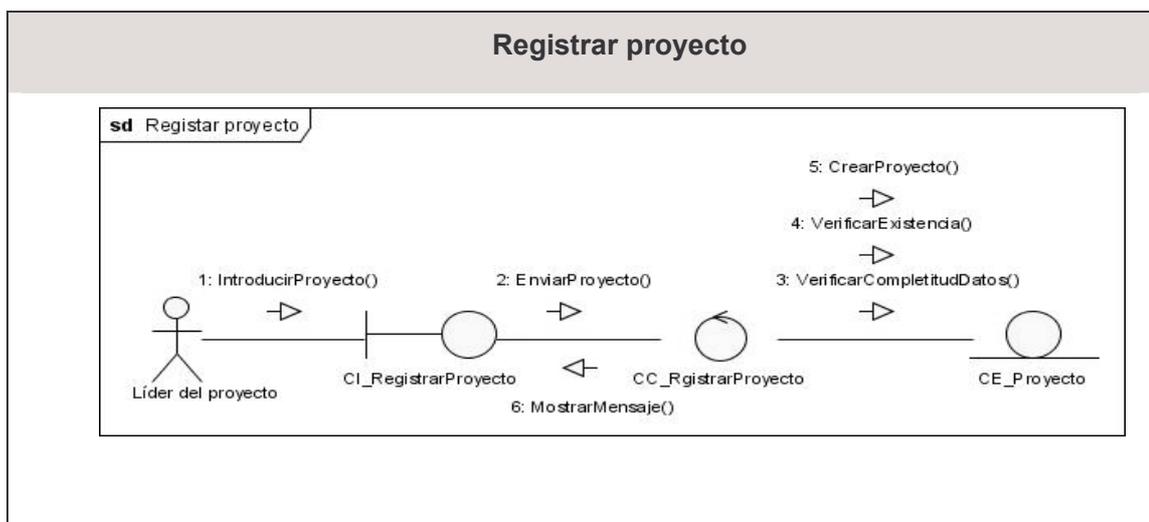
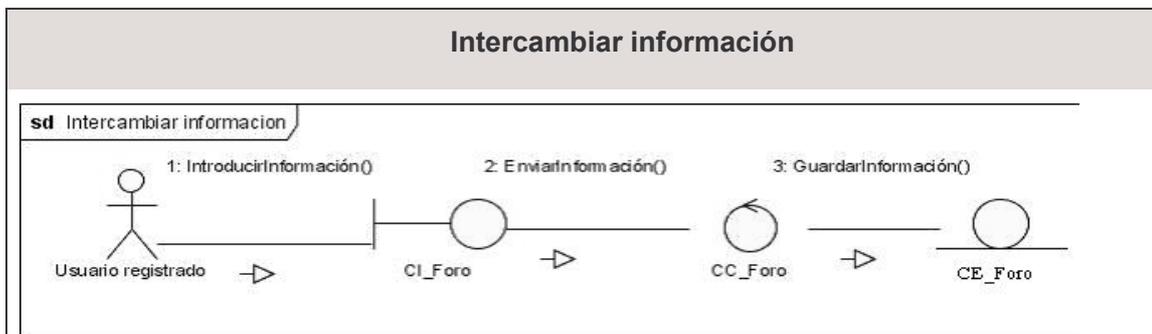


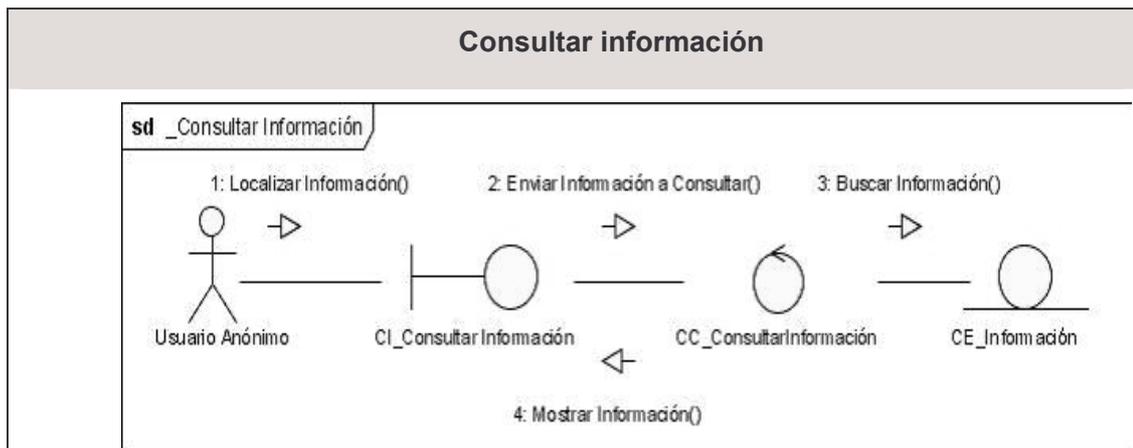
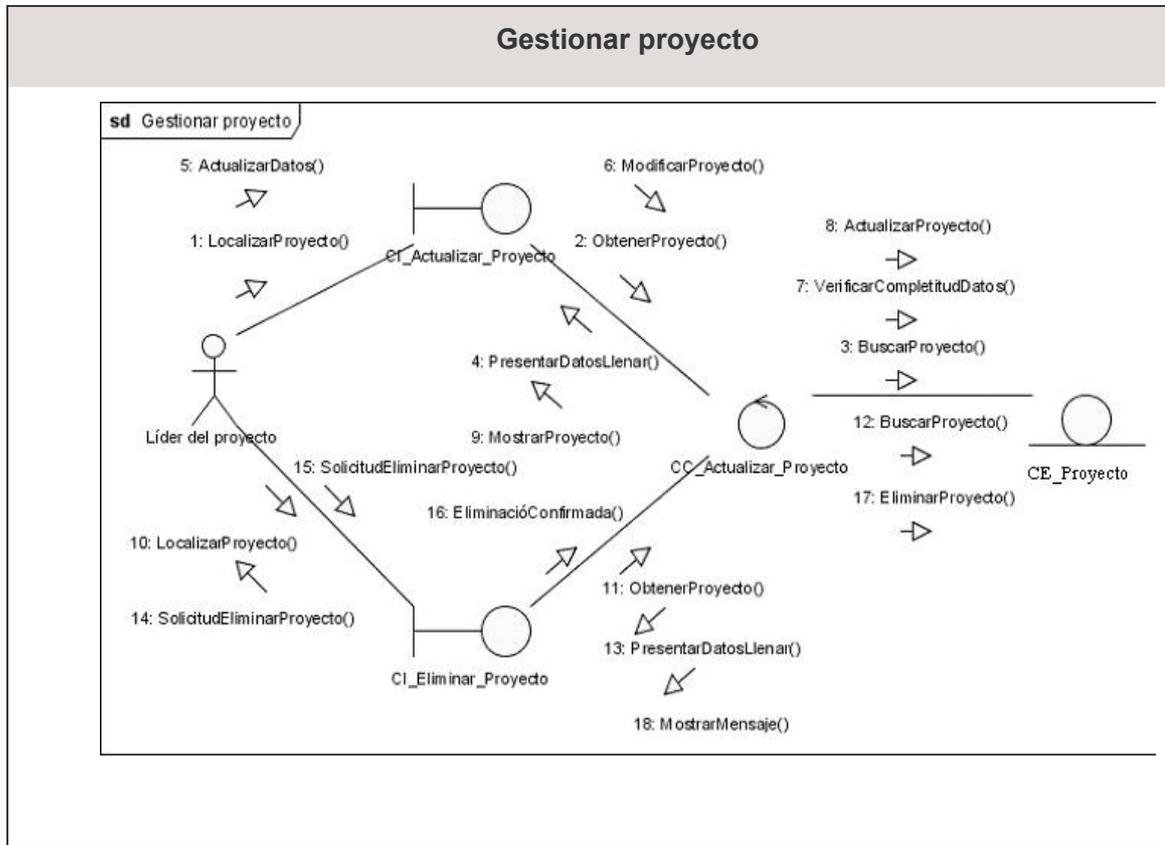


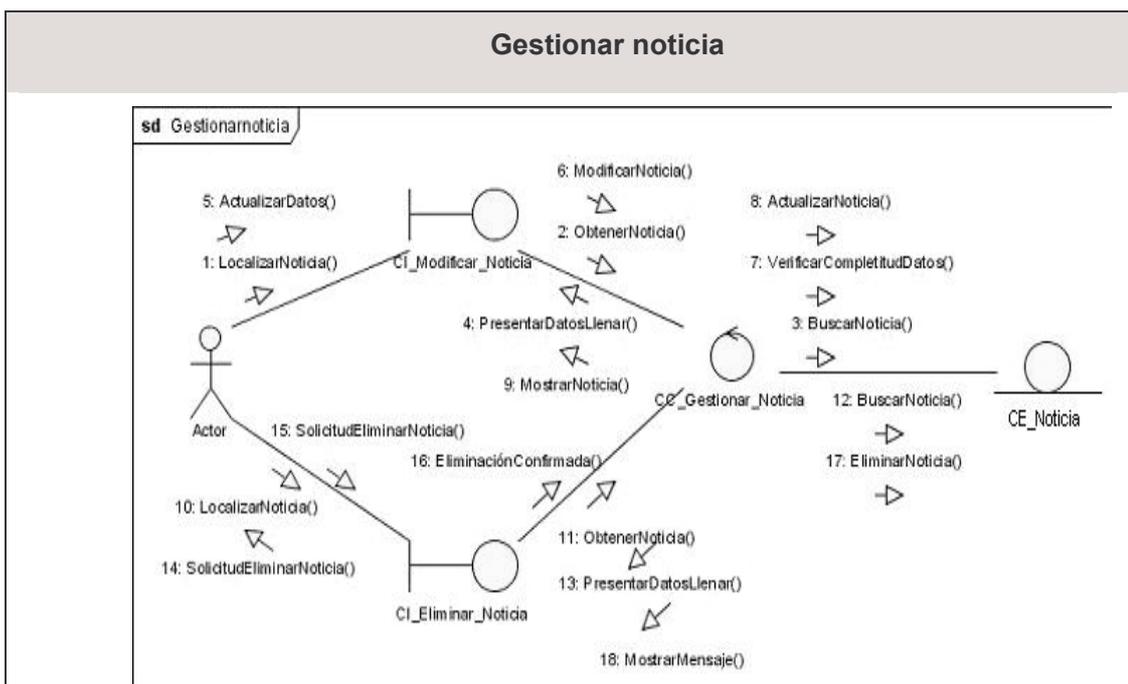
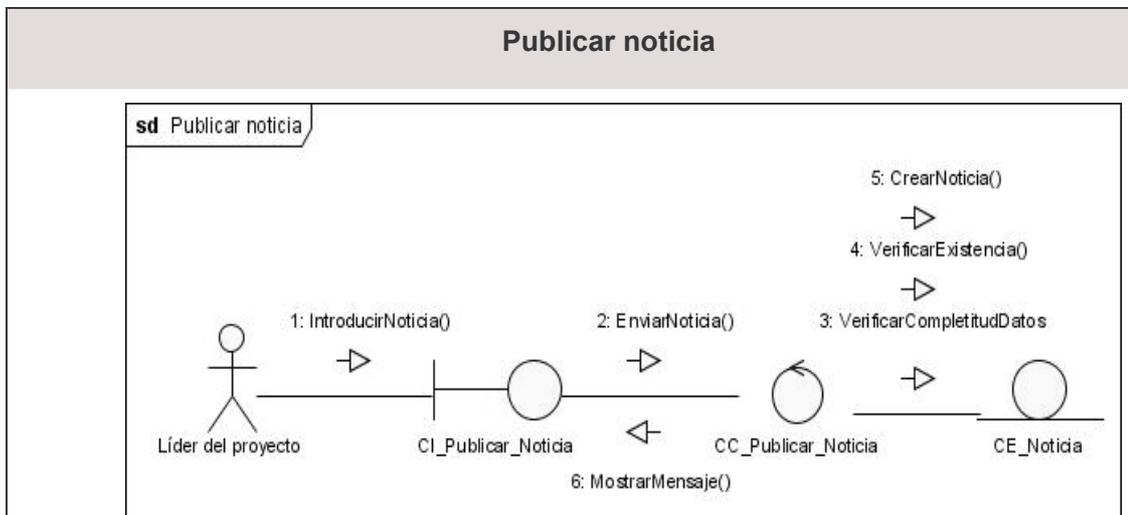


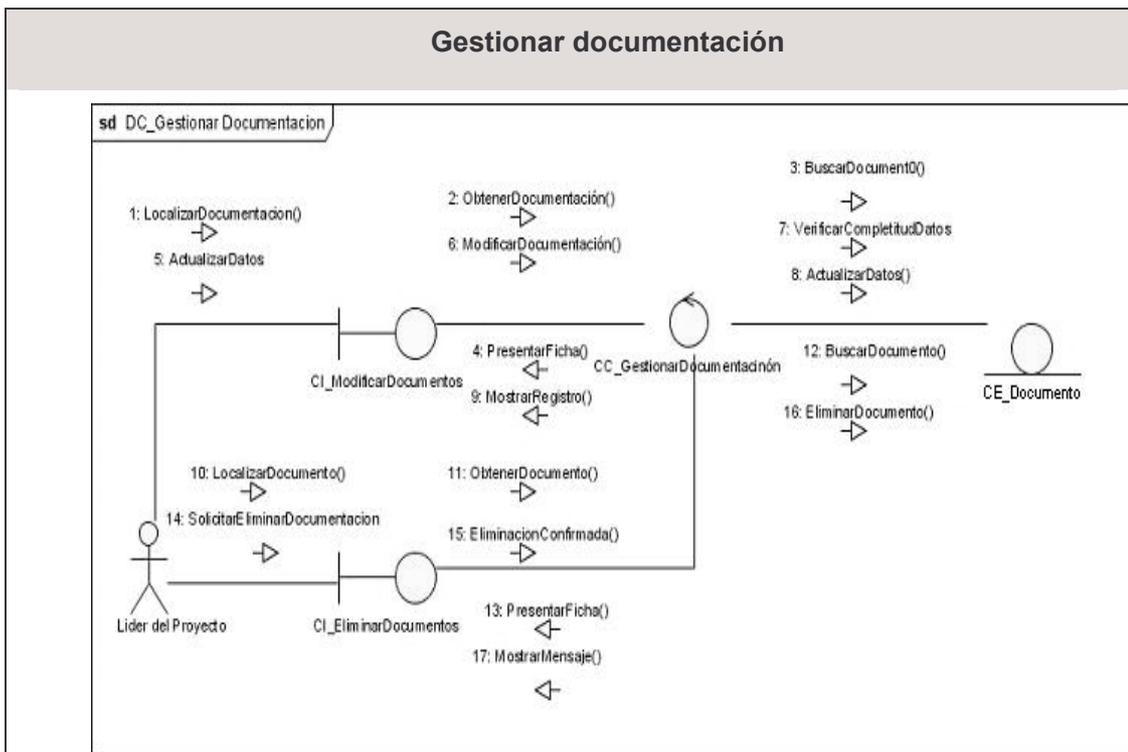
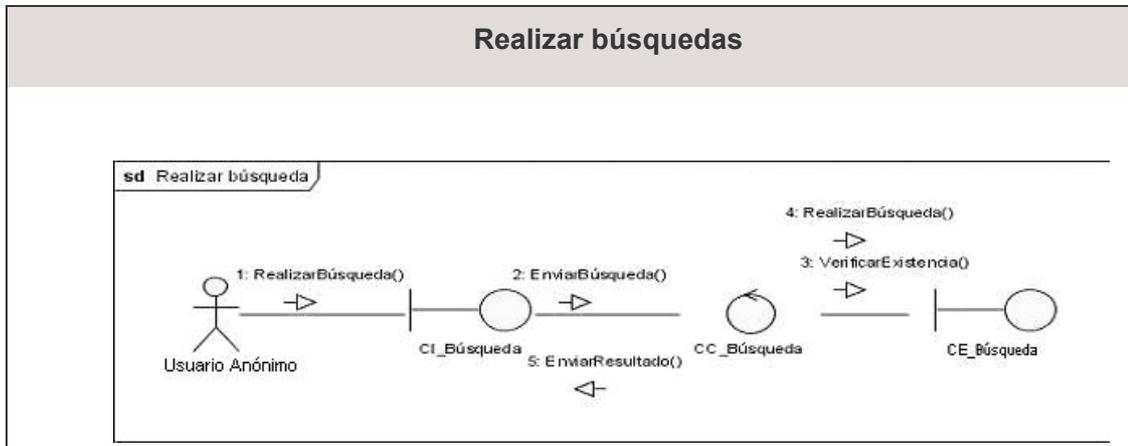


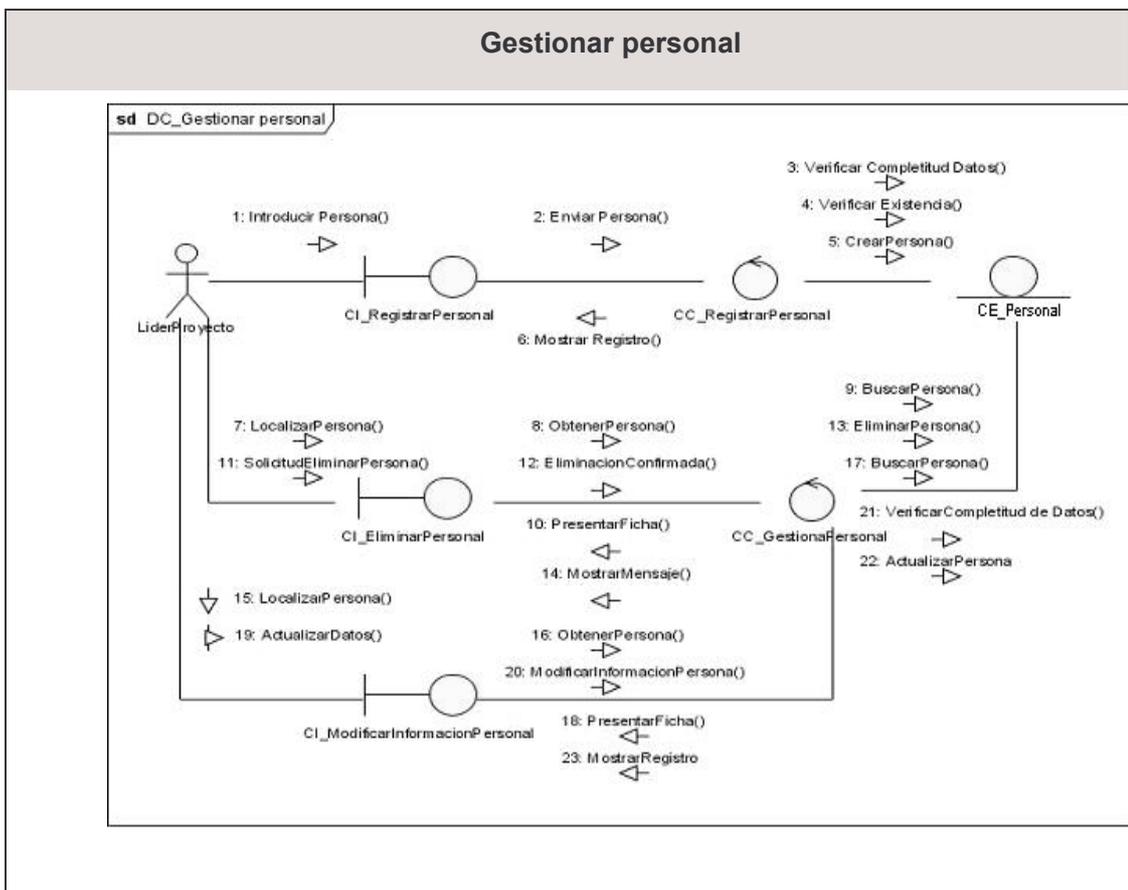
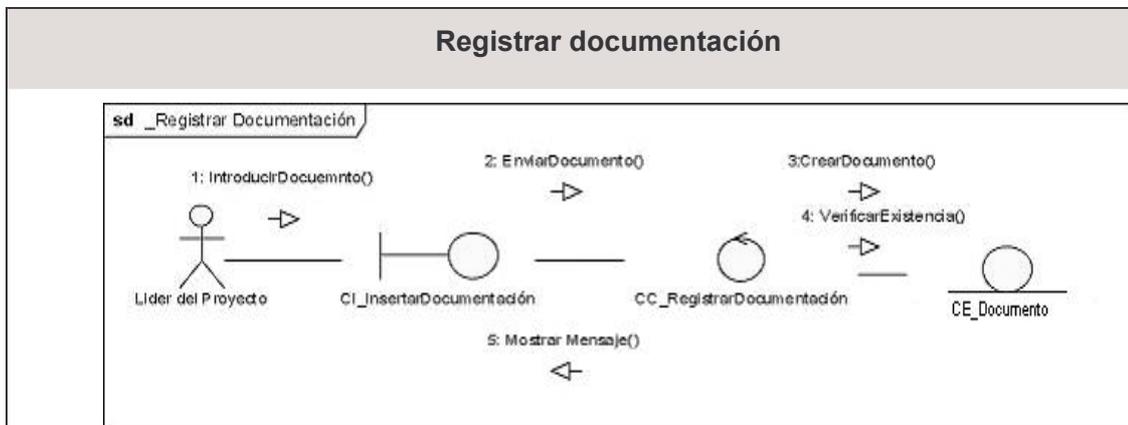
3.2.2 Diagramas de interacción

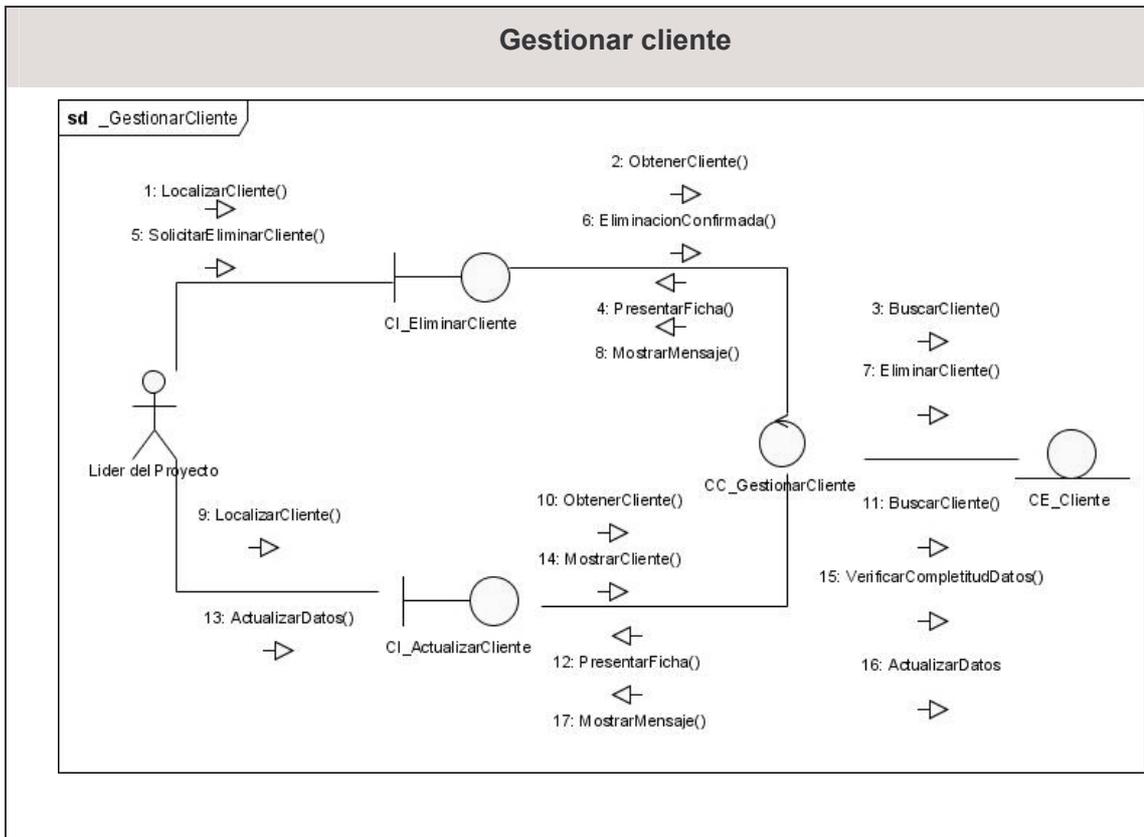
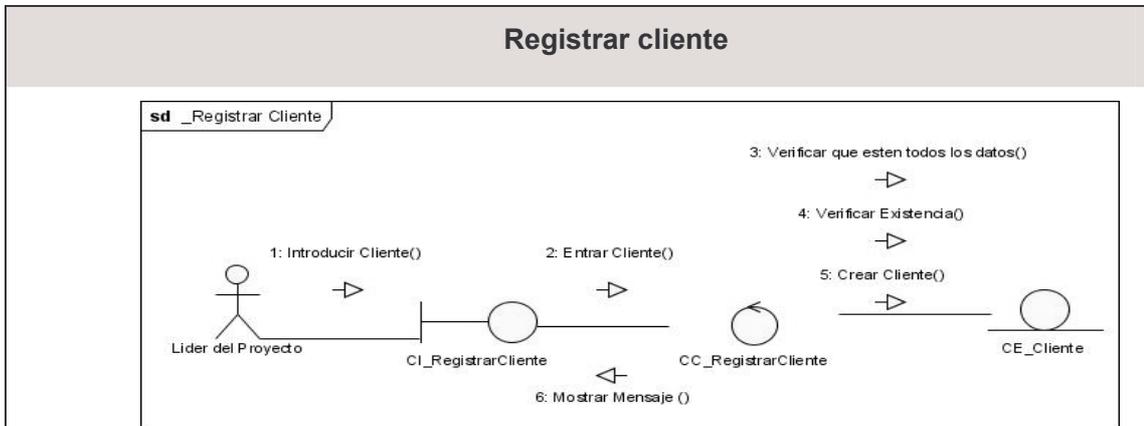


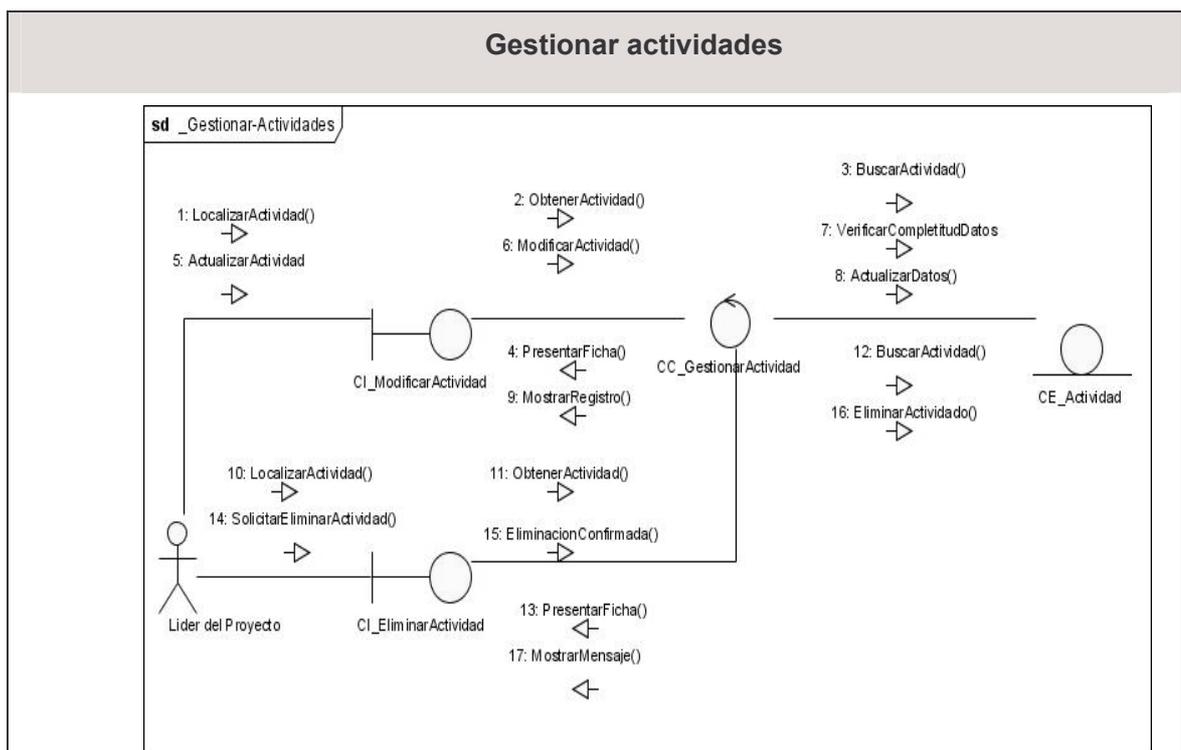
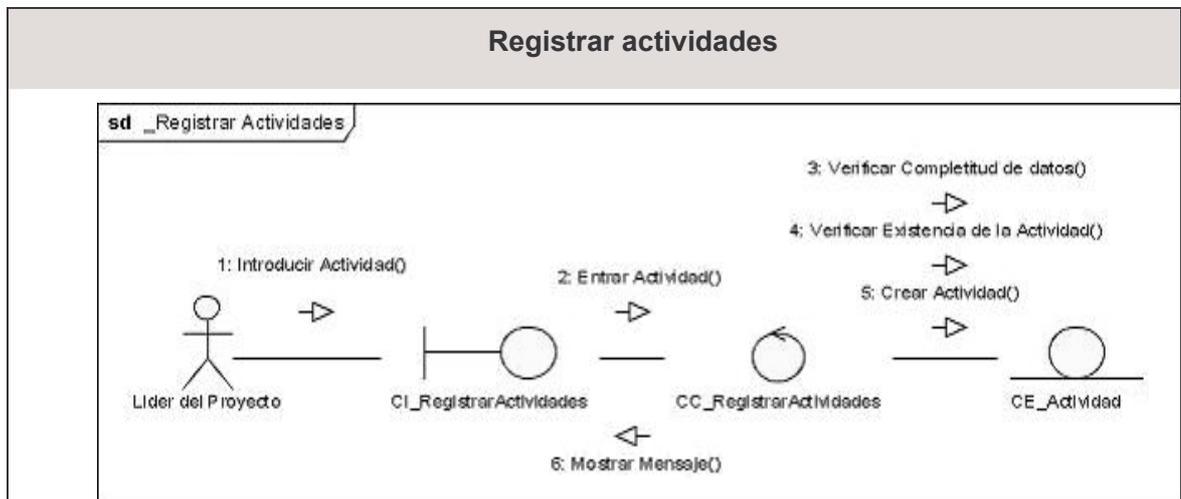


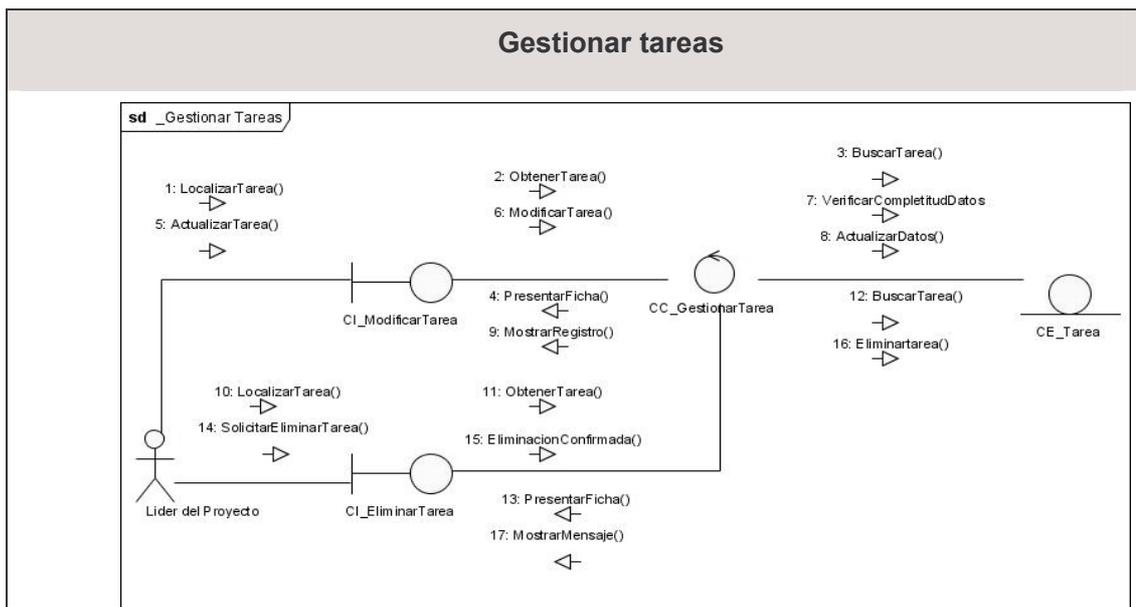
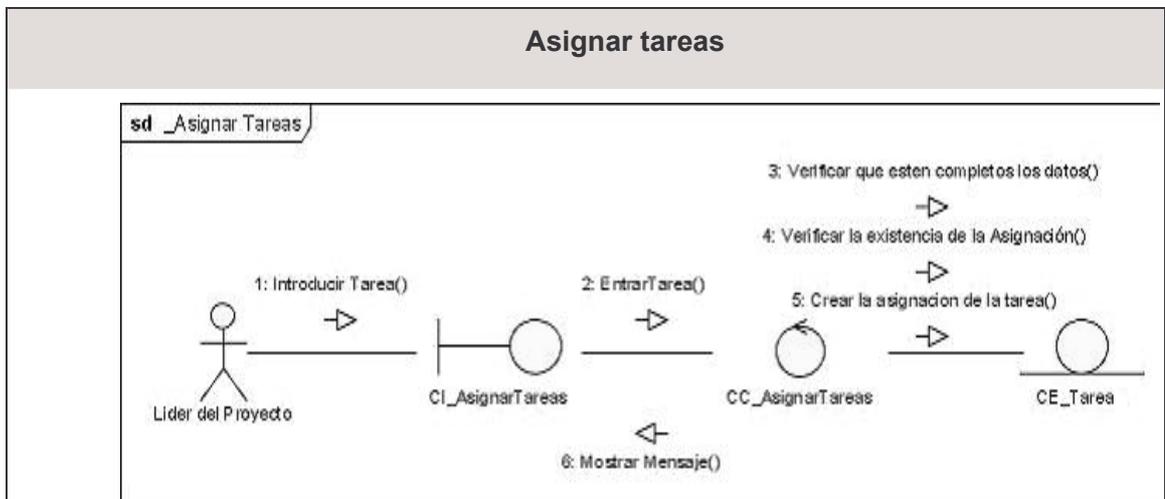


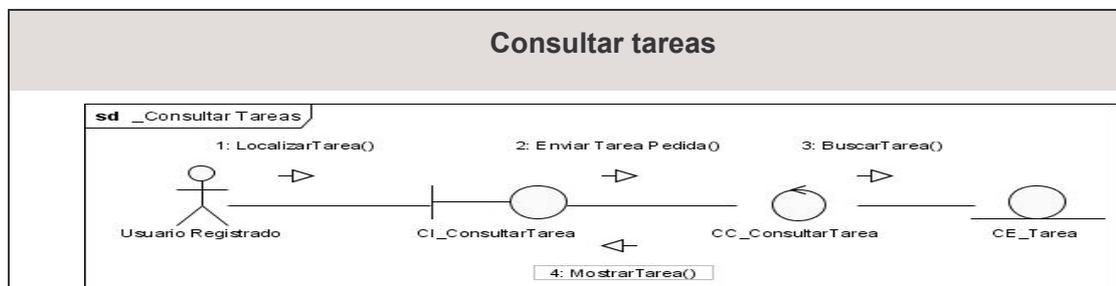
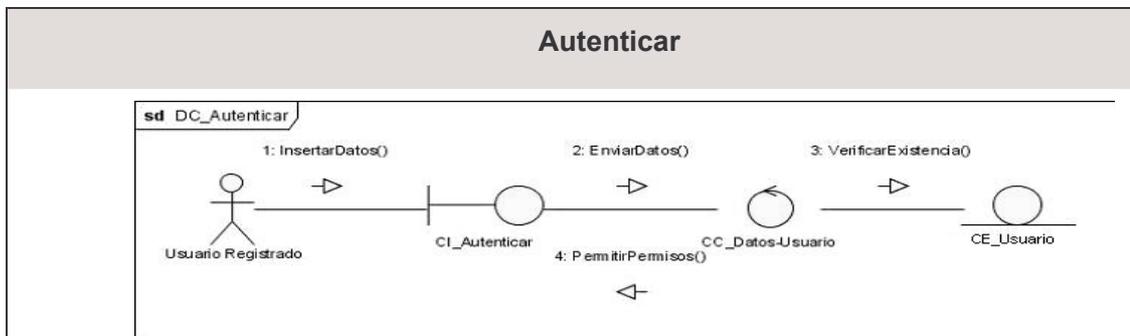












3.3 Diseño

3.3.1 Diagrama de clases Web

El diagrama de clases web debido a que la aplicación fue realizada utilizando el CMS Drupal incluye los paquetes de los módulos del mismo.

Drupal provee un tipo de contenido genérico llamado "node", y permite la creación, edición y publicación de este tipo de contenido. Los documentos creados se depositan en una base de datos central, donde también se guardan el resto de datos de la web, como son los datos relativos a los documentos (autor, fecha de publicación, título, etc.), datos y preferencias de los usuarios, la estructura de la web, etc.

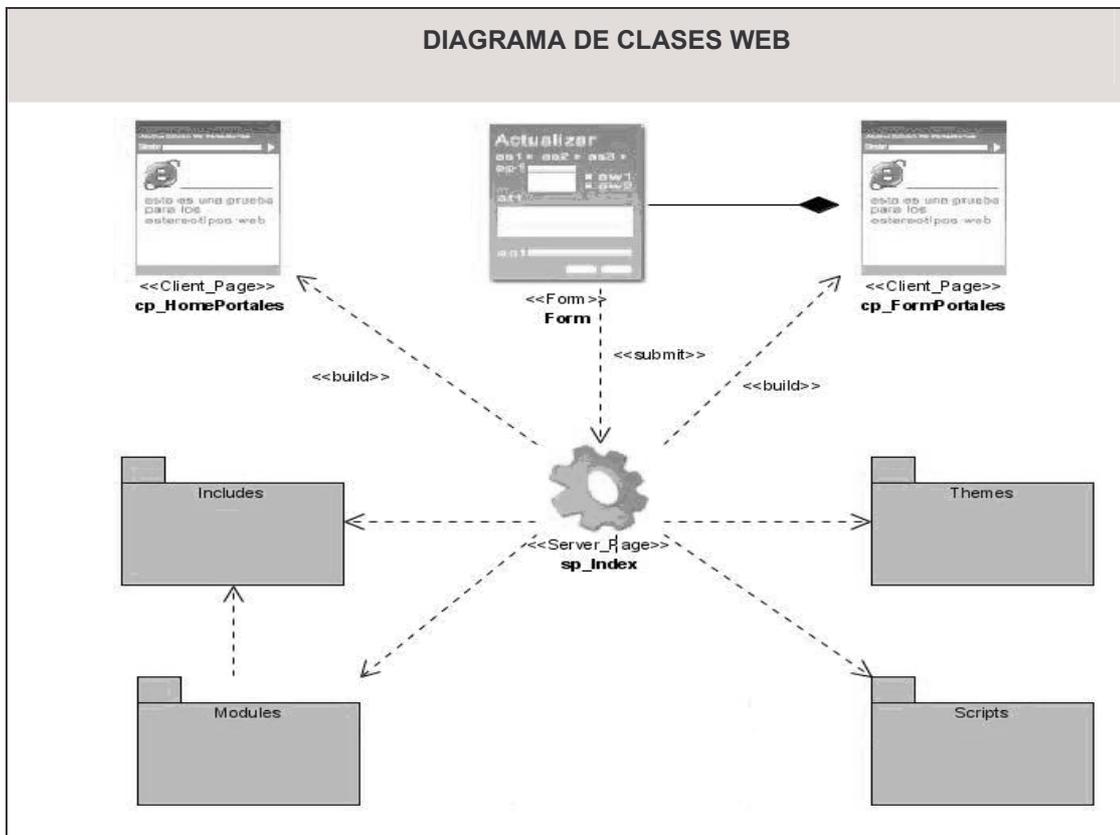
Posee un potente sistema de seguridad, basado en la creación de usuarios y roles, en dependencia de los permisos que se necesiten para administrar el sitio y acceder a los módulos.

Las páginas que genera Drupal puede tener o no formularios, en dependencia de lo que se desee. Es importante señalar que este CMS contiene una única pagina servidora.

Para una mejor comprensión del diagrama de clases web, en la siguiente tabla se describe por paquetes los módulos de Drupal representados en el mismo.

Paquetes	Descripción
Themes	Contiene las distintas plantillas, cuando se desee incluir un nuevo diseño, se copia la plantilla dentro de esta carpeta.
Includes	Contiene un conjunto de ficheros indispensables para el funcionamiento de Drupal. Incluye todo un conjunto de implementaciones de clases, que son empleados por los módulos. (incluye las clase de acceso a datos)
Modules	Incluye todos los módulos, que permiten las distintas funcionalidades del CMS, cuando desee incorporar un nuevo módulo sólo se tiene que copiar en dicha carpeta.
Scripts	Contiene un conjunto de ficheros indispensables para el funcionamiento. Fundamentalmente orientados al aspecto visual, css y java script

Tabla 3: Descripción de las carpetas que contiene Drupal.

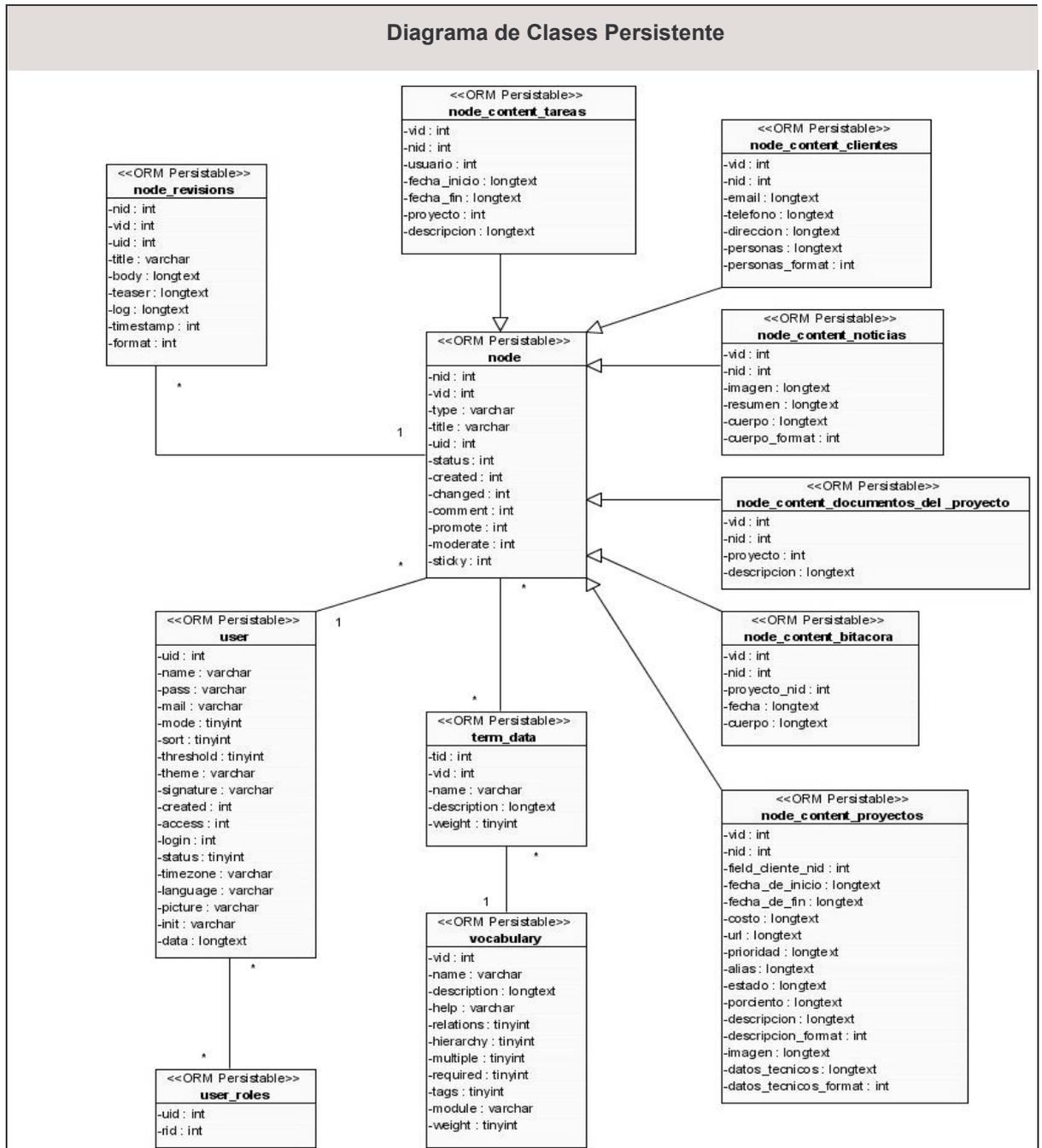


3.3.2 Diseño de la Base de Datos

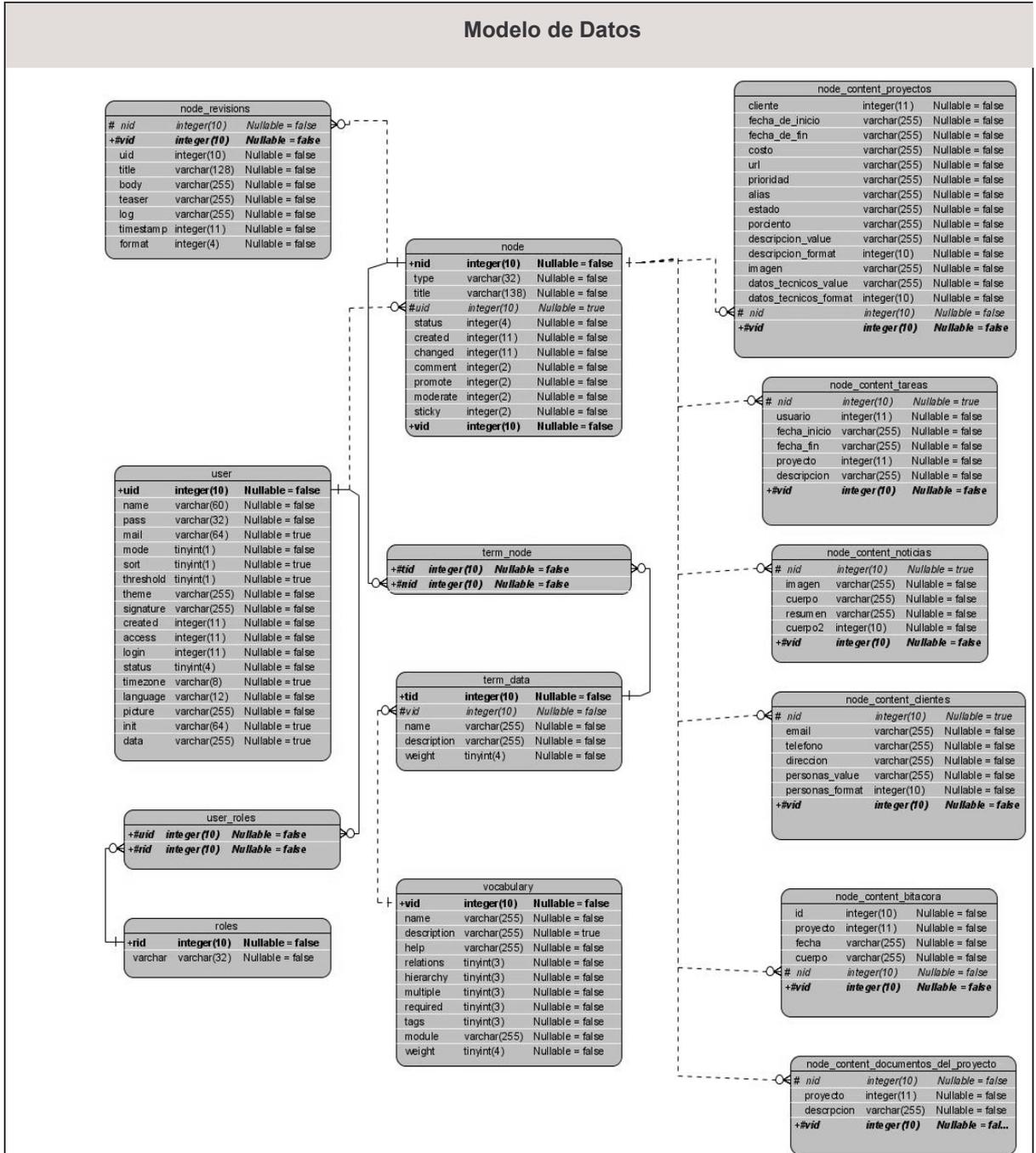
El diseño de la Base de Datos tiene como objetivo asegurar que los datos persistentes se almacenen de forma consistente y eficiente, define el comportamiento que debe ser implementado en la BD.

3.3.2.1 Diagrama de clases persistentes

Las clases persistentes son muy importantes, pues representan la información que el sistema necesita almacenar a largo plazo.



3.3.2.2 Modelo de Datos



El modelo de datos se utilizó para definir los cambios de las clases persistentes en las estructuras de datos persistentes utilizadas en el portal. Se creó a partir de las clases persistentes obtenidas previamente, esto permitió mantener las estructuras que garantizaran la integridad, consistencia y confiabilidad de los datos.

3.3.2.3 Descripción de las tablas.

Nombre: node		
Descripción: Tabla donde se almacenan todos los contenidos del portal.		
Atributo	Tipo	Descripción
<u>nid</u>	int	Id del contenido.
<u>vid</u>	int	Id del vocabulario.
type	varchar	Tipo de contenido.
title	varchar	Título del contenido.
uid	int	Usuario que posteó el contenido.
status	int	Dice si el contenido está publicado o no.
created	int	Fecha de creación.
changed	int	Fecha de última actualización.
comment	int	Si permite o no el envío de comentarios relacionados a este contenido.
promote	int	Si está o no promovido a la página principal.
moderate	int	Si se encuentra en la cola de moderación.
sticky	int	Si está con prioridad sobre otros contenidos.

Nombre: node_revisions		
Descripción: Tabla donde se almacenan los datos completos de los contenidos.		
Atributo	Tipo	Descripción
<u>nid</u>	int	Id del contenido.
<u>vid</u>	int	Id del vocabulario.
uid	int	Usuario que posteó el contenido.
title	varchar	Título del contenido.
body	longtext	Cuerpo del contenido.
teaser	longtext	Resumen del contenido.
log	longtext	Acciones que se van realizando en el contenido.
timestamp	int	Fecha de creación.
format	int	Formato del contenido, puede ser html o php.

Nombre: node_content_proyectos		
Descripción: Datos del contenido proyectos.		
Atributo	Tipo	Descripción
vid	int	Id de vocabulario.
nid	int	Id del contenido.
cliente_nid	int	Id del cliente al que responde este proyecto.
fecha_de_inicio	longtext	Fecha en que comienza la realización del proyecto.
fecha_de_fin	longtext	Fecha tope para concluir el proyecto.
costo	longtext	Costo del proyecto a desarrollar.
url	longtext	Dirección web del proyecto, si tiene
prioridad	longtext	Prioridad del proyecto, puede ser alta, baja, etc.
alias	longtext	Forma de llamar comúnmente al proyecto.
estado	longtext	Fase en que se encuentra el proyecto.
porciento	longtext	Grado de desarrollo del proyecto
descripcion	longtext	Descripción del proyecto.
descripcion_format	longtext	Formato de la descripción, puede ser html o php.
imagen	longtext	Imagen o logo del proyecto.
datos_tecnicos_value	longtext	Datos técnicos del proyectos, incluye necesidades de hard y sofá.
datos_tecnicos_format	int	Formato de datos técnicos, puede ser html o php.

Nombre: node_content_tareas		
Descripción: Datos de las tareas.		
Atributo	Tipo	Descripción
vid	int	Id de vocabulario.
nid	int	Id del contenido.
field_usuario_uid	int	Id del usuario al que se le asigna la tarea.
fecha_inicio_value	longtext	Fecha en que comienza la realización de la tarea.
fecha_fin_value	longtext	Fecha tope para concluir la tarea.
proyecto	int	Id del proyecto al que pertenece la tarea.
descripcion	longtext	En que consiste la tarea.

Nombre: node_content_clientes

Descripción: Datos de los clientes.		
Atributo	Tipo	Descripción
vid	int	Id de vocabulario.
nid	int	Id del cliente.
email	longtext	Correo electrónico del cliente.
telefono	longtext	Teléfono del cliente.
direccion	longtext	Dirección del cliente.
personas_value	longtext	Personas que pertenecen a la institución que representa el cliente.
personas_format	int	Formato del campo persona, puede ser php o html.

Nombre: node_content_noticias		
Descripción: Datos de las noticias.		
Atributo	Tipo	Descripción
vid	int	Id de vocabulario.
nid	int	Id de la noticia.
imagen	longtext	Logo de la noticia.
resumen	longtext	Breve descripción de la noticia.
cuerpo	longtext	Información completa de la noticia.
field_cuerpo_0_format	int	Formato del cuerpo de la noticia, puede ser html o php.

Nombre: node_content_documentos_del_proyecto		
Descripción: Datos de los documentos pertenecientes a un proyecto en cuestión.		
Atributo	Tipo	Descripción
vid	int	Id de vocabulario.
nid	int	Id del documento del proyecto.
proyecto	int	Id del proyecto al que pertenece el documento.
descripción	longtext	Descripción del documento.

Nombre: node_content_bitacora		
Descripción: Actividades del proyecto.		
Atributo	Tipo	Descripción
vid	int	Id de vocabulario.
nid	int	Id de la actividad.
proyecto	int	Id del proyecto al que pertenece la actividad.

fecha	longtext	Fecha a realizar la actividad.
cuerpo	longtext	Descripción de la actividad.

Nombre: term_data		
Descripción: Nombre de las categorías.		
Atributo	Tipo	Descripción
<u>tid</u>	int	Id de la categoría.
vid	int	Id de vocabulario.
name	varchar	Nombre de la categoría.
description	longtext	Descripción de la categoría.
weight	tinyint	Peso de la categoría.

Nombre: term_node		
Descripción: Relación entre nodos y categorías. Surge a partir de la relación de mucho a mucho de la tabla node y term_data.		
Atributo	Tipo	Descripción
<u>nid</u>	int	Identificador de la tabla node.
tid	int	Identificador de la tabla term_data.

Nombre: vocabulary		
Descripción: Contiene los datos de los términos de los vocabularios.		
Atributo	Tipo	Descripción
<u>vid</u>	int	Id del vocabulario.
name	varchar	Nombre del vocabulario.
description	longtext	Significado del vocabulario.
help	varchar	Ayuda del vocabulario.
relations	tinyint	Relación con otros vocabularios.
hierarchy	tinyint	Jerarquía de vocabularios, si existe.
multiple	tinyint	Si permite selección múltiple o no.
required	tinyint	Si es requerido o no.
tags	tinyint	Etiqueta.
module	varchar	Módulo de una categoría.
weight	tinyint	Peso del vocabulario

Nombre: user		
---------------------	--	--

Descripción: Datos de los usuarios		
Atributo	Tipo	Descripción
<u>uid</u>	int	Id del usuario.
name	varchar	Nombre.
pass	varchar	Contraseña.
mail	varchar	Correo.
mode	tinyint	Para llevar el control de los usuarios
sort	tinyint	Rol de usuario.
theme	varchar	Plantilla que usa el usuario.
signature	varchar	Firma del usuario.
created	int	Fecha en que se creo el usuario.
access	int	Fecha de último acceso.
login	int	Fecha y hora de la ultima vez que se logueó el usuario en formato unix
status	tinyint	Si está logueado o no.
timezone	varchar	Zona horaria que emplea el usuario
language	varchar	Lenguaje en que se le presenta la interfaz al usuario.
picture	varchar	Foto.

Nombre: user_roles		
Descripción: Relación entre usuario y roles.		
Atributo	Tipo	Descripción
<u>uid</u>	int	Id del usuario.
<u>rid</u>	int	Id del rol.

Nombre: roles		
Descripción: Relación entre usuario y roles.		
Atributo	Tipo	Descripción
<u>rid</u>	int	Id del rol.
name	int	Nombre del rol.

3.4 Definiciones de diseño que se apliquen

La aplicación que propone este trabajo está destinada a la gestión y control de las actividades del Proyecto Portales_PHP, lo que implica que el logo de la misma identifique la herramienta que se utiliza en el proyecto. El diseño debe ser sencillo, atractivo y de fácil navegabilidad de forma que le

sea cómodo al usuario. En este caso el diseño de la aplicación se logra con la construcción de una plantilla para el Sistema de Gestión de Contenido Drupal.

Por otra parte el módulo de administración no debe ser muy complejo de manipular, proporcionando así al líder del proyecto una plataforma de edición y publicación sencilla, flexible y potente. Aprovechando las bondades de Drupal como Sistema de Gestión de Contenido se logran prácticamente todos estos requisitos.

3.5 Tratamiento de errores

El tratamiento de los errores es un paso indispensable para el buen funcionamiento del sistema. Los errores más comunes que se pueden presentar están relacionados con el trabajo con la base de datos, fundamentalmente en los procesos de inserción, eliminación y/o modificación de datos. Se deben establecer mecanismos que visualicen la información para evitar en la medida de lo posible la introducción manual de los datos y minimizar los errores que pueda tener la aplicación. Se deben mostrar mensajes indicando los errores cometidos.

Para el tratamiento de errores se utilizó mensajes de texto en la misma página donde se ejecutaba la acción, permitiéndole al usuario corregir los errores con facilidad y continuar.

Los errores se hacen con el sistema de captura de errores de Drupal, una vez que ocurre una excepción el usuario es redireccionado a una página de error con el mensaje correspondiente.

A continuación se visualizan a través de imágenes ejemplos de tratamientos de errores de la aplicación.

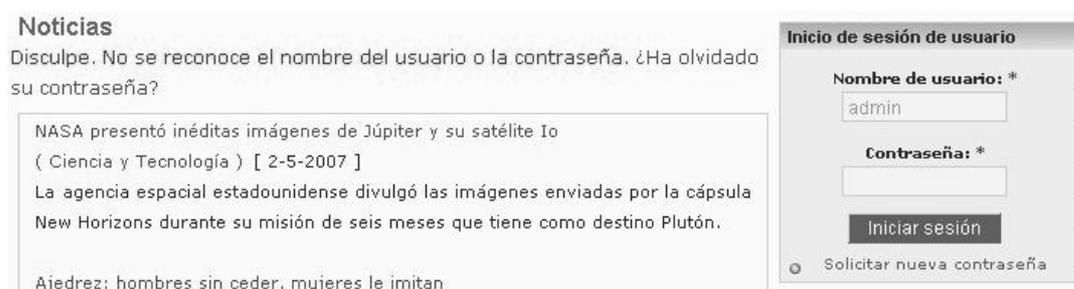


Ilustración 5: Verificación de los datos requeridos y en formato correcto de la autenticación del sitio.

Los campos especificados deben ser completados de lo contrario el sistema devolverá una página con señalamientos en los campos que faltan por completar.

The screenshot shows a web form titled "Enviar artículo". At the top, there are two bullet points indicating validation errors: "El campo Título es necesario." and "El campo Cuerpo es necesario.". Below these, the "Título:" field is marked with an asterisk and is empty. Underneath, there is a "Categorías" section with a dropdown menu. Two sub-sections are visible: "documentos:" with a dropdown menu showing "artículo" and "clasificación:" with a dropdown menu showing "accesibilidad". At the bottom, the "Cuerpo:" field is marked with an asterisk and is empty.

Ilustración 6: Verificación de la completitud de los campos requeridos

Por otra parte cada formulario se encarga de la validación de sus datos para evitar errores de concepto. Y se utilizan mensajes de confirmación, para acciones que son irreversibles como es el caso de las eliminaciones. Un ejemplo se evidencia a continuación.

The screenshot shows a confirmation dialog box with the title "¿Está seguro de que quiere eliminar Martí?". Below the title, it says "Esta acción no se puede deshacer.". At the bottom, there are two buttons: "Eliminar" and "Cancelar".

Ilustración 7: Verificación de eliminación de un contenido.

3.6 Seguridad

En el mundo de hoy donde la información constituye un elemento de vital importancia, es necesario que existan mecanismos de seguridad que garanticen la protección de los datos que se manipulan

en la aplicación y que los usuarios según su rol solo puedan realizar las acciones que le atañen y por tanto obtener información únicamente de las tareas para las que tienen autorización.

A partir de esta idea, para mantener la confiabilidad y legitimidad de la información que se maneja, se tuvo en cuenta para el acceso al sitio, una autenticación previa, donde el sistema comprueba que las credenciales coincidan con las almacenadas y le da los privilegios correspondientes. En caso contrario se redirecciona nuevamente el usuario a la página principal y se le brinda la posibilidad de autenticarse nuevamente.

El líder del proyecto es el encargado de mantener el sistema, crear usuarios y darle los niveles de acceso correspondientes a cada uno, además tiene la posibilidad de eliminar y modificar las propiedades de los mismos.

Además se mantiene un control de acceso a cada página, logrando que los usuarios solo tengan vía a la información que realmente les concierne. En este sentido, fue definido un módulo de administración de manera tal que se pudiera asignar permisos a los usuarios del sistema.

Una buena práctica a seguir sería realizar salvadas sistemáticas de la base de datos para evitar pérdidas de información por cualquier motivo que pueda presentarse. Estas salvadas deberán almacenarse en otros sistemas de cómputo fuera de donde radica el servidor de base de datos o en algún dispositivo de almacenamiento.

3.7 Interfaz.

Para la realización del diseño de las interfaces se tuvo en cuenta la organización de los elementos a mostrar; de forma tal que en todas las interfaces que tienen elementos comunes, estos ocupen la misma posición en cada una de ellas. Se conservaron los mismos patrones de colores, tamaño y forma. En todas las pantallas prevaleció el azul en varias tonalidades, claro para el texto y algo más oscuro para los encabezados, el gris fue usado en los bordes y el amarillo para los vínculos.

3.8 Concepción de la ayuda.

Todas las páginas del portal contienen un texto explicando cómo realizar las operaciones, por ejemplo las entradas que vaya a realizar el líder del proyecto, contienen un texto claro y conciso del dato que debe ser introducido. Además la búsqueda nos brinda una ayuda de cómo mejorar la

misma en caso de no obtener ningún resultado. También el portal cuenta con un mapa que ayuda a la navegabilidad del usuario en el sitio.

3.9 Conclusiones

Este capítulo representa el grueso del trabajo, con su terminación se ha tratado lo referente a la etapa de diseño del sistema. Se han representado los diagramas de clase de diseños (agrupados por casos de uso, para una mayor comprensión), así como los diagramas de clases persistentes y el modelo de datos, estos últimos forman parte del diseño de la base de datos propuesta. Por otra parte se explicaron los estándares de diseños por los que se rigió el trabajo, el tratamiento de errores, la seguridad del sitio y la concepción de la ayuda.

Capítulo IV. IMPLEMENTACION Y PRUEBA

4.1 Introducción

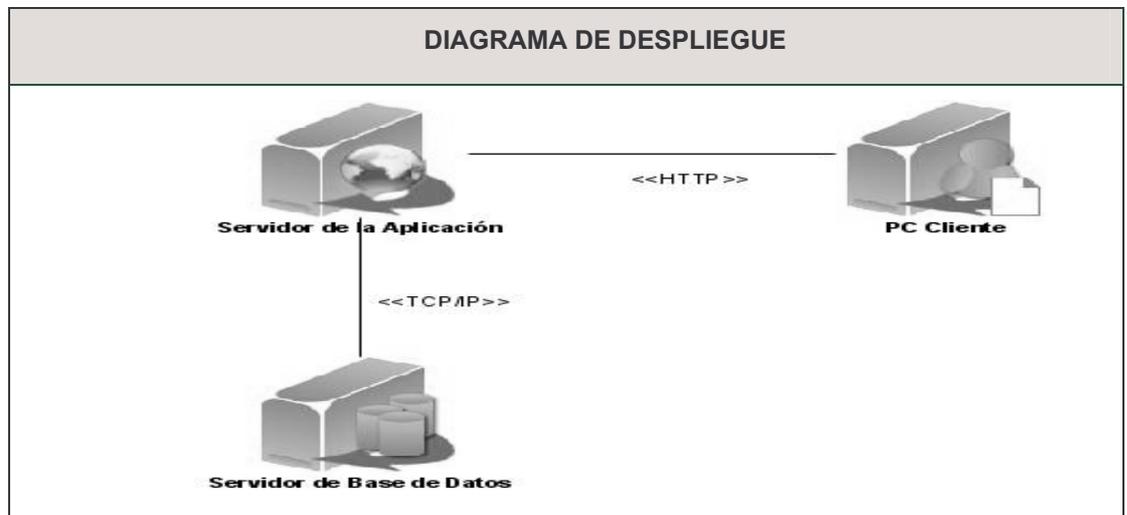
En este capítulo se desarrollan los flujos de trabajo implementación y prueba. Se presentará el Modelo de Implementación mediante el Diagrama de Componente y el Diagrama de Despliegue obtenidos a partir del diseño realizado a cada uno de los casos de uso del sistema. También se realizará un análisis de los casos de prueba para mostrar los posibles errores que puede tener el sistema y así se logra un mejoramiento del funcionamiento de la aplicación.

4.2 Implementación

La implementación se comienza con el resultado del diseño y se implementa el sistema en términos de componentes, es decir, ficheros de código fuente, scripts, ficheros de código binario, ejecutables y similares.

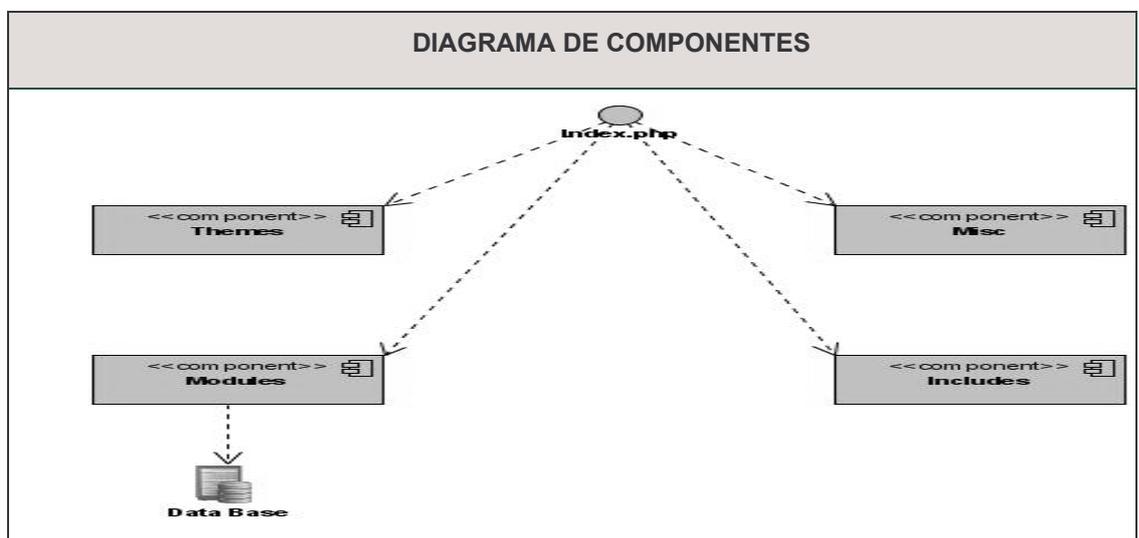
4.2.1 Diagrama de Despliegue

El diagrama de despliegue permite apreciar de forma visual como se encuentran relacionados físicamente los componentes de la aplicación. En este caso la aplicación se encuentra hospedada en un servidor Web, la misma se comunica con un sistema de gestión de base de datos (MySQL).



4.2.2 Diagrama de Componentes

El diagrama de componentes ayuda a un mejor entendimiento del modelo de implementación. Con el se representan los componentes lógicos de la aplicación así como las relaciones de dependencia que existen entre ellos.



A continuación se muestra la explicación de los archivos expuestos en el diagrama de componentes:

Carpetas	Descripción
Includes	Contiene un conjunto de ficheros indispensables para el funcionamiento.(18)
Misc	Incluye elementos que tienen que ver con el diseño, y funcionamiento (imágenes, ficheros).(18)
Modules	Incluye todos los módulos, que permiten las distintas funcionalidades del CMS, cuando desee incorporar un nuevo módulo solo tengo que copiarlo en dicha carpeta.(18)
Themes	Incluye las distintas plantillas, cuando desee incluir un nuevo diseño, copio la plantilla en esta carpeta.(18)
Data Base	Bases de datos de la aplicación
Index.php	Es el punto de inicio de la aplicación, a partir de esta entrada se solicitan los diferentes módulos del CMS.

Tabla 4: Contenido de los archivos del Diagrama de componentes.

4.3 Modelo de prueba.

Seguidamente se describen los casos de prueba desarrollados por cada caso de uso definido, especificándose los resultados y las condiciones que debe dar el sistema.

4.3.1 Descripción de los casos de prueba de integración.

Nombre del caso de uso: Autenticar

Entrada	Resultados	Condiciones
El usuario desea acceder a su módulo de administración pero no entra el usuario y/o la contraseña correcta.	El sistema emite un mensaje de error "Disculpe. No se reconoce el nombre del usuario o la contraseña. ¿Ha olvidado su contraseña?".	Se mantiene la pantalla de autenticación y no se le permite al usuario acceder a la aplicación. Se pregunta si ha olvidado la contraseña para que contacte con el web master y se la mande por correo.

El usuario desea acceder a su módulo de administración pero no entra el campo contraseña.	El sistema emite un mensaje de error "El campo Contraseña es necesario".	Se mantiene la pantalla de autenticación y no se le permite al usuario acceder a la aplicación.
---	--	---

Nombre del caso de uso: Registrar proyecto

Entrada	Resultados	Condiciones
El líder del proyecto desea crear un nuevo proyecto pero no introduce todos los campos obligatorios (nombre, cliente, prioridad, fecha de inicio, fecha de fin, url, estado, datos técnicos y descripción).	El sistema emite un mensaje de error informando los datos que este necesita (los campos obligatorios que dejó vacíos).	La operación se repite hasta que el líder del proyecto corrige el error.

Nombre del caso de uso: Gestionar proyecto

Entrada	Resultados	Condiciones
El líder del proyecto desea modificar un proyecto pero no introduce todos los campos obligatorios (nombre, cliente, prioridad, fecha de inicio, fecha de fin, url, estado, datos técnicos y descripción).	El sistema emite un mensaje de error informando los datos que este necesita (los campos obligatorios que dejó vacíos).	La operación se repite hasta que el líder del proyecto corrige el error.

Nombre del caso de uso: Publicar noticia

Entrada	Resultados	Condiciones
---------	------------	-------------

El líder del proyecto desea publicar una nueva noticia pero no introduce todos los campos obligatorios (título, resumen, cuerpo).	El sistema emite un mensaje de error informando los datos que este necesita (los campos obligatorios que dejó vacíos).	La operación se repite hasta que el líder del proyecto corrige el error.
---	--	--

Nombre del caso de uso: Gestionar noticia

Entrada	Resultados	Condiciones
El líder del proyecto desea modificar una nueva noticia pero no introduce todos los campos obligatorios (título, resumen, cuerpo).	El sistema emite un mensaje de error informando los datos que este necesita (los campos obligatorios que dejó vacíos).	La operación se repite hasta que el líder del proyecto corrige el error.

Nombre del caso de uso: Realizar búsquedas

Entrada	Resultados	Condiciones
El usuario desea realizar una búsqueda en el sitio pero no introduce la palabra correcta.	El sistema emite un mensaje de error "Su búsqueda no produjo ningún resultado" y muestra consejos de cómo realizar una mejor búsqueda.	La operación se repite hasta que el usuario corrige el error.

Nombre del caso de uso: Gestionar personal

Entrada	Resultados	Condiciones
El líder del proyecto desea gestionar datos del personal vinculado al proyecto pero no introduce todos los campos obligatorios (nombre, apellidos, correo electrónico y año).	El sistema emite un mensaje de error informando los datos que este necesita (los campos obligatorios que dejó vacíos).	La operación se repite hasta que el líder del proyecto corrige el error.

Nombre del caso de uso: Gestionar documentación

Entrada	Resultados	Condiciones
El líder del proyecto desea gestionar datos los documentos pero no introduce todos los campos obligatorios (título, descripción y proyecto).	El sistema emite un mensaje de error informando los datos que este necesita (los campos obligatorios que dejó vacíos).	La operación se repite hasta que el líder del proyecto corrige el error.

Nombre del caso de uso: Registrar documentación

Entrada	Resultados	Condiciones
El líder del proyecto desea registrar documentos pero no introduce todos los campos obligatorios (título, descripción y proyecto).	El sistema emite un mensaje de error informando los datos que este necesita (los campos obligatorios que dejó vacíos).	La operación se repite hasta que el líder del proyecto corrige el error.

Nombre del caso de uso: Registrar cliente

Entrada	Resultados	Condiciones
El líder del proyecto desea registrar datos de los clientes pero no introduce todos los campos obligatorios (nombre, dirección y personas).	El sistema emite un mensaje de error informando los datos que este necesita (los campos obligatorios que dejó vacíos).	La operación se repite hasta que el líder del proyecto corrige el error.

Nombre del caso de uso: Gestionar cliente

Entrada	Resultados	Condiciones
----------------	-------------------	--------------------

El líder del proyecto desea gestionar datos de los clientes pero no introduce todos los campos obligatorios (nombre, dirección y personas).	El sistema emite un mensaje de error informando los datos que este necesita (los campos obligatorios que dejó vacíos).	La operación se repite hasta que el líder del proyecto corrige el error.
---	--	--

Nombre del caso de uso: Asignar tareas

Entrada	Resultados	Condiciones
El líder del proyecto desea asignar tareas pero no introduce todos los campos obligatorios (título, usuario, fecha de inicio, fecha de fin, proyecto, descripción).	El sistema emite un mensaje de error informando los datos que este necesita (los campos obligatorios que dejó vacíos).	La operación se repite hasta que el líder del proyecto corrige el error.

Nombre del caso de uso: Gestionar tareas

Entrada	Resultados	Condiciones
El líder del proyecto desea modificar datos de tareas pero no introduce todos los campos obligatorios (título, usuario, fecha de inicio, fecha de fin, proyecto, descripción).	El sistema emite un mensaje de error informando los datos que este necesita (los campos obligatorios que dejó vacíos).	La operación se repite hasta que el líder del proyecto corrige el error.

Nombre del caso de uso: Registrar actividades

Entrada	Resultados	Condiciones
---------	------------	-------------

El líder del proyecto desea registrar datos de las actividades pero no introduce todos los campos obligatorios (título, proyecto, fecha y cuerpo).	El sistema emite un mensaje de error informando los datos que este necesita (los campos obligatorios que dejó vacíos).	La operación se repite hasta que el líder del proyecto corrige el error.
--	--	--

Nombre del caso de uso: Gestionar actividades

Entrada	Resultados	Condiciones
El líder del proyecto desea modificar datos de las actividades pero no introduce todos los campos obligatorios (título, proyecto, fecha y cuerpo).	El sistema emite un mensaje de error informando los datos que este necesita (los campos obligatorios que dejó vacíos).	La operación se repite hasta que el líder del proyecto corrige el error.

4.4 Conclusiones

En el capítulo que se concluye se han abordado los flujos de implementación y prueba propuestos por RUP, llevándose a cabo en el caso del primero la modelación del diagrama de despliegue donde se especificó la distribución de los sistemas de cómputo que conforman la aplicación con sus características y las conexiones y protocolos que los acoplan y también el diagrama de componentes de la aplicación web. Como parte del flujo de prueba se realizaron pruebas a casos de usos donde se evidenciaban.

Conclusiones

Como resultado del trabajo realizado, se ha podido obtener un modelo de solución, utilizando RUP como metodología y representaciones UML para la modelación de todas las fases del proyecto.

Se elaboró una aplicación Web basada en estándares de código abierto que automatiza la labor del proyecto Portales_PHP, garantizando la integridad, autenticidad y confidencialidad de los datos. La misma brinda una interfaz sencilla, muy fácil de navegar por el usuario.

Con la puesta en marcha de este sistema se logrará una celeridad de los procesos de brindar información, controlar todas las actividades que se desarrollan, divulgación y documentación de las herramientas, control y asignación de tareas y controlar datos del personal, los clientes y de los proyectos en general. Además se reducirá el desgaste físico y mental de los trabajadores del proyecto, lo que permitirá ganar en eficiencia y productividad.

Recomendaciones

1. Implantar sistemas similares con la misma funcionalidad en todos los proyectos que se desarrollan en la universidad
2. Continuar investigando para lograr mayores funcionalidades y nuevas mejoras en futuras versiones.
3. Se recomienda que el trabajo sea tomado como material de estudio para aquellas personas que vayan a realizar una aplicación similar o un trabajo referente al tema expuesto.
4. Que se continúe desarrollando y analizando todas las funcionalidades que se pueden brindar a los usuarios del software.
5. Emplear para versión futura el modelo de base de datos de Dotproject y el Drupal para la parte de presentación.

Bibliografía

Adiestramiento en la instalación y actualización del Portal Nacional de Desarrollo e Integración de software libre. Consultado: 20/05/07, Disponible en: <http://drupal.org.es>.

ALEJANDRO MARTÍNEZ, R. M. *Guía a Rational Unified Process.*

ALFONSO, X. C. G. J. M. *Introducción a los Sistemas de Gestión de Contenidos (CMS) de código abierto.* 29.11.04, Publicado: 29/11/04, Consultado: 2/11/07, Disponible en: <http://mosaic.uoc.edu/articulos/cms1204.html>

ALVAREZ, M. A. *DesarrolloWeb.* Consultado: 15/12/06, Disponible en: <http://www.desarrolloweb.com/php/>.

CABALLERO, I. *"Visual Paradigm".* Consultado: 20/05/07, Disponible en: http://alarcos.inf-cr.uclm.es/per/fgarcia/isoftware/doc/LabTr1_VP.pdf.

DONDO, A. ¿Por qué elegir PHP? Consultado: 3/03/07, nº Disponible en: <http://www.webtaller.com/maletin/articulos/por-que-elegir-php.php>.

Gforge. Consultado: 10/03/07, Disponible en: <https://gforge.uci.cu>.

GUERVOS, J. J. M. *Introducción a los sistemas de gestión de contenidos (Tipos de CMS).* Publicado: 14/03/05, Consultado: 16/11/06, Disponible en: <http://geneura.ugr.es/~jmerelo/tutoriales/cms>.

GUTIÉRREZ, M. P. M. *Funciones de los sistemas de gestión de contenidos.* Publicado el: 10/04/05, Consultado 16/11/06, Disponible en: <http://www.ub.es/bid/14monto2.htm>.

Linux. Consultado: 02/02/07, Disponible en: <http://linux.bankhacker.com/software/Apache/>.

PRESSMAN.,R.S. *Ingeniería del Software. Un enfoque práctico,* Disponible en: <http://bibliodoc.uci.cu/pdf/reg02689.pdf>

REYERO, J. A. *Características de Drupal.* Publicado el: 17/07/05, Disponible en: <http://www.drupal.org.es/caracteristicas>.

RUMBAURG, I. J. G. B. Y. J. *"El Proceso Unificado del Software".* 2000.

Servidor Web. Apache. Consultado: 20/03/07, Disponible en: <http://news.netcraft.com/>

Tipos de CMS"Moodle". Consultado: 2/02/07, Disponible en: <http://www.cibersociedad.net/congres2006/gts/comunicacio.php?lengua=ca&id=880>

Referencias Bibliográficas

1. ALFONSO, X. C. G. J. M. Introducción a los Sistemas de Gestión de Contenidos (CMS) de código abierto. 29.11.04, Publicado: 29/11/04, Consultado: 2/11/07, Disponible en: <http://mosaic.uoc.edu/articulos/cms1204.html>.
2. CMS. Publicado el: 16/05/07, Consultado: 20/05/07, Disponible en: http://es.wikipedia.org/wiki/Sistema_de_gesti%C3%B3n_de_contenido.
3. GUTIÉRREZ, M. P. M. Funciones de los sistemas de gestión de contenidos. Publicado el: 10/04/05, Consultado 16/11/06, Disponible en: <http://www.ub.es/bid/14monto2.htm>.
4. GUERVOS, J. J. M. Introducción a los sistemas de gestión de contenidos(Tipos de CMS). Publicado: 14/03/05, Consultado: 16/11/06, Disponible en: <http://geneura.ugr.es/~jmerelo/tutoriales/cms>.
5. Tipos de CMS"Moodle". Consultado: 2/02/07, Disponible en: <http://www.cibersociedad.net/congres2006/gts/comunicacio.php?llengua=ca&id=880>
6. Tipos de CMS" Generales y de Galería". Ultima Actualización: 16/05/07, Consultado: 20/05/07, Disponible en: http://es.wikipedia.org/wiki/Sistema_de_gesti%C3%B3n_de_contenido.
7. Drupal. Ultima Actualización: 16/05/07, Consultado: 20/05/07, Disponible en: <http://es.wikipedia.org/wiki/Drupal>.
8. REYERO, J. A. Características de Drupal. Publicado el: 17/07/05, Disponible en: <http://www.drupal.org.es/caracteristicas>.
9. Soluciones de Código Abierto. Ultima Actualización: 16/04/07, Consultado: 20/04/07, Disponible en: http://es.wikipedia.org/wiki/C%C3%B3digo_abierto.
10. Servidor Web. Apache. Consultado: 20/03/07, Disponible en: <http://news.netcraft.com/>
11. Linux. Consultado: 02/02/07, Disponible en: <http://linux.bankhacker.com/software/Apache/>.

12. Servidor de Bases de Datos. MySQL. Última Publicación: 7/05/07, Consultado: 20/05/07, Disponible en: <http://es.wikipedia.org/wiki/MySQL#Plataformas>.
13. ALVAREZ, M. A. DesarrolloWeb. Consultado: 15/12/06, Disponible en: <http://www.desarrolloweb.com/php/>.
14. DONDO, A. ¿Por qué elegir PHP? Consultado: 3/03/07, nº Disponible en: <http://www.webtaller.com/maletin/articulos/por-que-elegir-php.php>.
15. RUMBAURG, I. J. G. B. Y. J. "El Proceso Unificado del Software". 2000.
16. CABALLERO, I. "Visual Paradigm". Consultado: 20/05/07, Disponible en: http://alarcos.inf-cr.uclm.es/per/fgarcia/isoftware/doc/LabTr1_VP.pdf.
17. Gforge. Consultado: 10/03/07, Disponible en: <https://gforge.uci.cu>.
18. Adiestramiento en la instalación y actualización del Portal Nacional de Desarrollo e Integración de software libre. Consultado: 20/05/07, Disponible en: <http://drupal.org.es>.

Glosario de Términos

Apache: Servidor de páginas web de código abierto para diferentes plataformas (UNIX, Windows, etc.)

Blog (Web log): Diario en formato web. Puede ser un diario personal o un conjunto de noticias, ordenado por fecha.

CMS (Content Management System): Sistema que facilita la gestión de contenidos en todos sus aspectos: creación, mantenimiento, publicación y presentación. También se conoce como Web Content Management (WCM) sistema de gestión de contenido de webs.

GPL (General Public License): Licencia que permite el uso y modificación del código para desarrollar software libre, pero no propietario.

Software libre: Código abierto o código libre. Software que distribuye de forma libre su código fuente, de forma que los desarrolladores pueden hacer variaciones, mejoras o reutilizarlo en otras aplicaciones. También conocido como free software.

PHP (PHP Hypertext Preprocessor): Lenguaje de programación para el desarrollo de webs dinámicas, con sintaxis parecida a la C. Originalmente se conocía como Personal Home Page tools, herramientas para páginas personales (en Internet).

URL (Uniform Resource Locator): Dirección de un recurso en la web. Tiene el formato `protocol://màquina.domini:port/ruta/recurs`. Por ejemplo `http://www.uoc.edu/dt/20396/index.html` donde no se indica el puerto porque el protocolo HTTP tiene uno por defecto (80).

Web: Sistema para presentar información en Internet basado en hipertexto. Cuando se utiliza en masculino (el web, un web) se refiere a un sitio web entero, en cambio si se utiliza en femenino (la web, una web) se refiere a una página web concreta dentro del sitio web.

WYSIWYG (What You See Is What You Get): Traducido: lo que ves es lo que obtienes, que aplicado a la edición significa trabajar con un documento con el aspecto real que tendrá. Editar una página de HTML en un editor que no sea WYSIWYG, implica trabajar con los códigos que indican el formato que tendrá el texto, sin ver el resultado final.

TICs: Tecnologías de la Informática y las Comunicaciones.