

Universidad de las Ciencias Informáticas

FACULTAD 2



“Sistema de Entorno Integrado de Desarrollo de Informatización”

Trabajo de diploma para optar por el título de Ingeniero Informático

Autores: Surelys Brossard Sánchez

Halena Yayset Leyva Carrasquero

Tutor: Ing. Janett Tase Hernández

Ciudad de la Habana, Junio 2008

“Año 50 de la Revolución”

DECLARACIÓN DE AUTORÍA

Por este medio declaramos que somos las únicas autoras de este trabajo y autorizamos a la Universidad de las Ciencias Informáticas (UCI) para que hagan el uso que estimen pertinente con este trabajo.

Para que así conste firmamos la presente a los __ días del mes de junio del 2008.

Halena Leyva Carrasquero

Surelys Brossard Sánchez

Ing. Janett Tase Hernández

Firma de la Autor(a)

Firma de la Autor(a)

Firma del tutor(a)

DATOS DE CONTACTO

Síntesis de la Tutora: Graduada en el año 2005, desde ese año forma parte de la Dirección de Informatización de la UCI atendiendo al seguimiento y control de los proyectos. Ha impartido clases de Práctica Profesional, Álgebra Lineal y Matemática Discreta.

Profesión: Ing. Informática

Categoría docente: Instructor

Años de graduada: 3

Agradecimientos.

A la obra revolucionaria cúmulo de justicia y honor, por convertir este sueño en realidad de muchos.

A nuestra tutora Janett Tase por su apoyo y sus consejos para realizar el trabajo.

A Dionísio Estrada por tener siempre horario abierto para ayudarnos.

A Dariena Ramírez, porque su atención y ayuda se convirtieron en un eslabón fundamental para el desarrollo de este trabajo.

A Yudiél A. Tamayo por enseñarnos de la mejor manera en que consistía y como debíamos proceder para hacer mejor el trabajo.

A todos los que contribuyeron para el logro de este trabajo.

Agradecimientos.

De Halena:

A mi madre, el ángel que cuida mis pasos, que rodea el tiempo de fuerzas para que no desfallezca, que enfrenta al mundo, a la muerte y sale siempre airosa por entregarme un beso, a ti porque nunca has intentado desaparecer.

A mi reina, dueña absoluta de los parajes más bellos de mi alma, cómplice de todos mis sueños, por robar mis tristezas, por sacudir mi locura, por entregarme tus apretones de alma, por convertirme en quien soy gracias abuela porque vivir a tu lado es el mejor premio que me dio la vida.

A mi tía madre porque siempre inundas de valor y amor todos los espacios de mi vida. A mi hermano Arian porque te adoro y porque el sueño de ser como tú me trajo hasta aquí. A Carlitos mi hermano bello por siempre robarme una sonrisa.

A mi novio Abelardo porque me diste un espacio en tu ser para llenarlo de mí, porque soportas mis desvaríos, porque desde que existes, los agujeros de mi corazón desaparecieron.

A la Nene, porque nunca noté ausencia de cariño en tus manos.

A una gran amiga, Doris porque siempre tienes un sí en los labios cuando te necesito.

A alguien que no está pero que hubiera sido muy feliz de ver este momento a Matías.

A Maurice mi bóveda de secretos, a la tropa que reunió en torno a él.

A Made mi hermana porque no hace falta decir más nada, a Iralys, La ÑK, Mary porque lágrimas y sonrisas conforman nuestra historia.

A Yamí, Rita, Sure por soportar mis malos amaneceres. Al grupo 2503 por abrirme sus brazos, al 2502 porque empezamos juntos y seguiremos juntos para toda la vida.

Al PkT de las tertulias porque conocerlos hizo más linda mi existencia.

A Edístio Verdecía por ser un excelente profesor y enseñarme parte de lo que hoy se.

A Bárbara Laborit porque siempre encontré apoyo firme en sus palabras.

A todos gracias porque muchos tienen un pedacito en esta historia.

Agradecimientos.

De Surelys

A mi madre Arellys por luchar tanto por hacerme entender que debía llegar hasta aquí.

A mi padre Rafael por que siempre creyó desde el principio que yo podía lograrlo.

A todos mis profesores que tuvieron gran influencia en mi preparación profesional, en especial a Fernando R. Rodríguez Marzo y Lester Rodríguez Vallejo.

A mi familia que siempre fue apoyo seguro e incondicional. A mis abuelos que no vacilaron en contribuir en mi crianza.

A mis tíos que más que eso han sido padres. A mis hermanos y primos que no por ser más pequeños dejaron de preocuparse.

A la persona que desde que empezó mi carrera lejos de mi hogar se ha convertido en mi principal apoyo ejerciendo el papel de padre, de madre y amigo, José Manuel Cordero Cruzata.

A la más fiel e incondicional amiga de mi madre para quien soy una hija más, a mi madrina y a Eduardo Piñeiro, todos han estado junto a mí, a pesar de la distancia apoyándome en cada momento necesario en esta carrera.

A mis amigos los de verdad, cada uno de ellos saben de que les estoy hablando, los que nunca me fallaron, a los que quiero por lo que han sembrado, no los voy a mencionar, no porque no sean especiales, sino porque son muchos para listar aquí, pero cada quien sabe el papel que jugó en mi carrera y algunos en la confección de este trabajo, no solo con apoyo moral sino ayudándome a trabajar directamente en él.

Dedicatoria

Surelys:

A ella porque desde que nací nunca se ha apartado de mí, aún estando lejos su corazón y pensamiento siempre han estado conmigo. A ella por no desfallecer un día para educarme, por seguir cada uno de mis pasos, guiarme, apoyarme y convertirme en la persona que soy.

A ella, porque cuando creyó que las fuerzas se le agotaban para hablarme y aconsejarme las encontraba nuevamente y me demostró que la vida va más allá de un conjunto de reglas a cumplir, que había que trazarse una meta y luego, aunque el camino fuese duro, sacrificarse para lograr ser alguien en la vida. A ella, que nunca perdió la fe, ni la esperanza en mí, que se convirtió en guía, ejemplo, compañera, amiga, confidente, en fin en mi razón de ser, de estar y de vivir. A mi madre Arellys Sánchez Robert.

A él aunque ya no esté conmigo y si realmente como dicen las almas buenas van al cielo, allá ha de estar muy orgulloso de mí, de verme realizarme y de saber que influyó mucho en mi preparación. A mi abuelo Rafael E. Brossard Serrano.

Halena:

A mi madre por luchar contra el universo por robarme un beso, por derrumbar los muros que cercaban mis sueños, a mi madre por enseñarme a amar a creer a vivir.

A mi abuela porque vivió cada segundo a mi lado este sueño, porque hoy vibra su alma al compás continuo de mi aliento.

Resumen

La Dirección de Informatización de la Universidad de las Ciencias Informáticas dirige, organiza, y coordina todos los procesos de informatización internos en cada una de las áreas que rigen la vida de la universidad, desde la perspectiva de una Ciudad Digital. Además propone la arquitectura base de los proyectos que responden a ella, garantiza el soporte a la infraestructura de aplicaciones y servicios, contando con un personal altamente capacitado. El objetivo de este trabajo consiste en desarrollar un sistema integrado a la Plataforma de Soporte de la UCI que gestione los servicios de Hospedaje de Base de Datos y aplicaciones, repositorios de código, salvas, control de versiones, gestión de publicaciones en el nodo central y el acceso a los mismos a través de solicitudes.

En este trabajo se exponen elementos importantes del proceso de creación del Sistema de Entorno Integrado de Informatización destacando la fundamentación del uso del Sistema Gestor de Contenidos Drupal. Se describe el proceso de construcción del sistema a través de la metodología RUP enfatizando en la documentación de los procesos actuales y los propuestos. Se determinó la factibilidad de la realización del software teniendo en cuenta análisis de costos y beneficios.

El resultado final de esta investigación es la obtención de SEIDI Sistema de Entorno Integrado de Desarrollo de Informatización, que cumple con los requisitos de la Dirección de Informatización y que promueve la transformación de la UCI en un prototipo para la informatización de la sociedad cubana, basado en el funcionamiento armónico de las tecnologías y los servicios informáticos.

INTRODUCCIÓN	1
CAPÍTULO I FUNDAMENTACIÓN TEÓRICA	4
1.1 APLICACIONES CLIENTE/SERVIDOR.	4
1.1.1 <i>Sistemas para la Gestión de Solicitudes. Tendencias actuales.</i>	5
1.2 ¿QUÉ ES UN CMS?	8
1.2.1 <i>Distintos tipos de CMS</i>	8
1.2.2 <i>Puntos importantes que hacen útil y necesaria la utilización de un CMS.</i>	9
1.2.3 <i>Principales Sistemas de Gestión de Contenidos del Mercado</i>	11
1.3 FUNDAMENTACIÓN DE LA METODOLOGÍA, LENGUAJES Y HERRAMIENTAS A UTILIZAR.	17
1.3.1 <i>Procesos de desarrollo de software.</i>	17
1.3.2 <i>Software libre.</i>	20
1.3.3 <i>Lenguaje de programación PHP.</i>	20
1.3.4 <i>Sistemas Gestores de Base de Datos (SGBD).</i>	21
1.3.5 <i>Servidor HTTP Apache.</i>	24
1.3.6 <i>Dreamweaver.</i>	25
1.4 CONCLUSIONES.....	25
CAPÍTULO II CARACTERÍSTICAS DEL SISTEMA.....	27
2.1 OBJETO DE ESTUDIO.	27
2.1.1 <i>Problema y Situación Problemática.</i>	27
2.1.2 <i>Objeto de Automatización.</i>	27
2.1.3 <i>Información que se maneja</i>	28
2.2 PROPUESTA DEL SISTEMA.	29
2.3 MODELO DE NEGOCIO.	29
2.3.1 <i>Actores y Trabajadores del negocio.</i>	29
2.3.2 <i>Diagrama de Casos de Uso del negocio.</i>	30
2.3.2 <i>Descripción textual Casos de Uso del negocio.</i>	30
2.3.3 <i>Diagrama de Actividades.</i>	30
2.3.4 <i>Diagrama de clases del modelo de objetos.</i>	30
2.4 REQUERIMIENTOS DEL SISTEMA.....	30
2.4.1 <i>Especificación de los requisitos funcionales del software.</i>	30
2.4.2 <i>Requerimientos no funcionales del Sistema.</i>	32
2.5 MODELO DE CASOS DE USO DEL SISTEMA.....	34
2.5.1 <i>Definición de actores del Sistema.</i>	34
2.5.2 <i>Distribución de los casos de uso por paquetes.</i>	35
2.5.3 <i>Diagrama de Casos de Uso del sistema.</i>	39
2.5.4 <i>Descripción de los Casos de Uso del Sistema</i>	39
2.6. CONCLUSIONES	39
CAPÍTULO III ANÁLISIS Y DISEÑO DEL SISTEMA.....	40
3.1 DIAGRAMAS DE CLASES DEL ANÁLISIS	40
3.2. <i>DISEÑO</i>	43
3.2.1 <i>Diagrama de clases del diseño Web del sistema.</i>	48
3.3. DISEÑO DE LA BASE DE DATOS.....	49
3.3.1 <i>Diagrama de Clases Persistentes.</i>	50
3.3.2 <i>Modelo físico de datos</i>	51
3.4. PRINCIPIOS DE PROTECCIÓN Y SEGURIDAD.	52
3.5. TRATAMIENTO DE ERRORES	52
3.6. CONCLUSIONES	53
CAPÍTULO IV IMPLEMENTACIÓN.	54

4.1.	DIAGRAMA DE DESPLIEGUE.	54
4.2.	DIAGRAMA DE COMPONENTES.	54
4.3.	CONCLUSIONES.	56
CAPÍTULO V ESTUDIO DE FACTIBILIDAD.....		57
5.1.	PLANIFICACIÓN BASADA EN CASOS DE USO. ANÁLISIS DE PUNTOS DE CASOS DE USO.	57
5.1.1.	<i>Cálculo de los Puntos de casos de uso Desajustados.</i>	57
5.1.2.	<i>Factor de peso de los actores sin ajustar (UAW).....</i>	58
5.1.3.	<i>Factor de peso de los casos de uso sin ajustar (UUCW).....</i>	59
5.1.4.	<i>Cálculo de los Puntos de casos de uso ajustados.....</i>	59
5.2.	ESTIMACIÓN DE ESFUERZO A TRAVÉS DE LOS PUNTOS DE CASOS DE USO.	62
5.2.1.	<i>Calcular el esfuerzo de todo el proyecto.</i>	63
5.3.	BENEFICIOS TANGIBLES E INTANGIBLES	64
5.4.	ANÁLISIS DE COSTOS Y BENEFICIOS	64
5.5.	CONCLUSIONES	65
CONCLUSIONES.....		66
RECOMENDACIONES.....		67
REFERENCIAS BIBLIOGRÁFICAS.		68
BIBLIOGRAFÍA.....		70
GLOSARIO DE TÉRMINOS.....		71
ANEXOS.....		75

Introducción

En el año 2002 surge la Universidad de las Ciencias Informáticas, proyecto que tiene como objetivo fundamental la formación de profesionales teniendo en cuenta nuevos planes de estudio que insertan la producción de software en el proceso docente. Con vista a la organización de estos planes surge la Infraestructura Productiva (IP) compuesta por un conjunto de direcciones que atienden los diferentes proyectos que lleva a cabo la universidad. Entre las direcciones que pertenecen a esta infraestructura se encuentra la Dirección de Informatización, encargada de convertir la UCI en un prototipo para la informatización de la sociedad cubana, basado en el funcionamiento armónico de las tecnologías y los servicios informáticos, propiciando el uso ordenado y masivo de las mismas. Esta dirección organiza, coordina, chequea y diseña la informatización de todos los procesos internos en cada una de las áreas que rigen la vida de la universidad, desde la perspectiva de una Ciudad Digital. Además propone la arquitectura base para la informatización de la UCI, garantiza el soporte a los productos terminados y brinda servicios a los proyectos en ejecución.

Existe un grupo de especialistas dentro de esta dirección que con alto grado de preparación se encargan de brindar servicios de soporte a las aplicaciones que surgen del proceso de informatización de la UCI. El proceso de solicitud a los servicios de Hospedaje de Base de Datos y aplicaciones, salvos, control de versiones, gestión de publicaciones en el nodo central, entre otros definidos por este grupo de especialistas, en la actualidad no resulta efectivo. Uno de los mayores problemas existentes es el hecho de que, tanto el proceso de solicitud de servicios por los interesados, como el registro y revisión de todo el volumen de información acumulada al respecto, se efectúa manualmente, lo que provoca que el proceso sea lento, y muy complicado a la hora de realizar búsquedas ante alguna reclamación por parte de los usuarios. Además dificulta tanto el control estadístico de las solicitudes realizadas como la integridad del documento en cuestión. Unido a esto se encuentra que el proceso de gestión de la publicación de los servicios que son brindados, no está automatizado.

La necesidad de dar solución a las situaciones antes expuestas, ha servido de motivación para la realización de este trabajo lo que determina que el **problema científico** radique en ¿Cómo solucionar los problemas existentes en el proceso de gestión de servicios y solicitudes que brinda el grupo de soporte del área de informatización?

El **objeto de estudio** para el desarrollo de este trabajo se centra en el proceso de gestión de las solicitudes de servicios de soporte, delimitando como **campo de acción** la gestión de los servicios

que brinda el grupo de soporte de la dirección de Informatización de la UCI y las solicitudes a los mismos.

La automatización de las solicitudes de los mencionados servicios beneficiaría a la comunidad universitaria, integrando la tecnología al servicio de los usuarios y restándole dificultad al proceso, por lo que **se defiende la idea** de construir una herramienta capaz de llevar a cabo la gestión de los servicios y las solicitudes de manera eficiente.

El **objetivo general** de la investigación es desarrollar una solución que permita gestionar los servicios brindados por el grupo de soporte y el proceso de solicitud a estos servicios.

Para el cumplimiento del objetivo general se desglosan los siguientes objetivos específicos:

- Definir los principales elementos teóricos-metodológicos para el análisis del problema a resolver.
- Desarrollar el análisis y diseño de una aplicación que gestione los servicios y las solicitudes a los mismos en la UCI.
- Obtener el sistema con las características definidas en los procesos de análisis y diseño.

Las tareas de investigación llevadas a cabo para el logro de los objetivos son:

- Investigación de temas relacionados con los diferentes sistemas de solicitudes que existen en Cuba y el mundo.
- Revisión de aplicaciones de Gestión de Solicitudes.
- Selección de las herramientas, lenguajes y metodologías a utilizar.
- Documentación de la información referente al análisis, diseño e implementación del sistema.
- Diseño de la Base de Datos.
- Implementación de la aplicación.

El documento que se presenta se encuentra estructurado en cinco capítulos. A continuación se desglosa el contenido de los mismos.

Capítulo 1 Fundamentación teórica: En este capítulo se trata la situación de las tecnologías y tendencias actuales que tienen vinculación con la solución que se propone. Se ofrecen argumentos que explican el porqué de la selección de determinada herramienta o tecnología para el desarrollo de la aplicación.

Capítulo 2 Características del sistema: Está dedicado al estudio de los procesos del negocio y las características del sistema, se definen modelos, actividades, actores y trabajadores del sistema, así como los requisitos y casos de uso del mismo.

Capítulo 3 Análisis y Diseño del Sistema: Se procede a representar la realización de los casos de uso. También se muestran los diagramas de clases del análisis y el diseño, el modelo de datos y las clases persistentes.

Capítulo 4: En este capítulo se documenta el flujo de trabajo de implementación. Contiene el modelo de despliegue y diagrama de componentes.

Capítulo 5: Este capítulo comprende el estudio de factibilidad del software, en el se expone el análisis de costos y beneficios de la realización del sistema.

Capítulo I Fundamentación Teórica

Este capítulo está destinado a brindar una panorámica de la situación actual de los sistemas con características similares al que se propone como solución. Se realiza un análisis de las técnicas, tecnologías y herramientas que pudieran ser utilizadas en la construcción del sistema que se pretende desarrollar. Se describen los principales conceptos asociados al dominio del problema que son necesarios para entender el modelo de negocio y la propuesta de solución.

1.1 Aplicaciones cliente/servidor.

La arquitectura cliente servidor es la base para todas las aplicaciones que hoy se pueden encontrar en Internet y en casi la totalidad de las Intranet corporativas que hoy existen en la mayoría de las empresas.

International Business Machines (IBM) define al modelo Cliente/Servidor. "Es la tecnología que proporciona al usuario final el acceso transparente a las aplicaciones, datos, servicios de cómputo o cualquier otro recurso del grupo de trabajo y/o, a través de la organización, en múltiples plataformas. El modelo soporta un medio ambiente distribuido en el cual los requerimientos de servicio hechos por estaciones de trabajo inteligentes o "clientes", resultan en un trabajo realizado por otros computadores llamados servidores". (1)

Decir además que las aplicaciones cliente/servidor son una relación entre procesos corriendo en máquinas separadas donde el servidor es un proveedor de servicios y el cliente es un consumidor de servicios. Tanto el servidor como el cliente interactúan por un mecanismo de envío de mensajes: pedido de servicio y respuesta

En esta arquitectura toda aplicación de un sistema de información está caracterizada por tres componentes básicos:

- Presentación / Captación de Información
- Procesos
- Almacenamiento de la Información

Existen casos en los que esta arquitectura es un poco más compleja, o sea, incluye un nuevo elemento: una aplicación que se ejecuta en el servidor. Este tipo de arquitectura permite manejar lógica de negocio a través de una "aplicación Web", que es como se denomina a este tipo de sistemas.

La aplicación que se ejecuta en el servidor se encarga de controlar el estado del negocio y de gestionar los datos almacenados con ayuda de algún Sistema Gestor de Bases de Datos (SGBD).

En la actualidad se ha generalizado el uso de aplicaciones Web dada las grandes posibilidades que brindan, y dado que los clientes sólo necesitan un navegador, capaz de interpretar código con formato HTML (acrónimo en inglés de HyperText Markup Language (lenguaje de marcas hipertextuales)), para hacer uso de ellas: no tienen que instalar ningún componente de software adicional. Por tanto se propone que el sistema a desarrollar consista en una aplicación Web teniendo en cuenta la diversidad de terminales en los que podrá ser utilizado el sistema.

1.1.1 Sistemas para la Gestión de Solicitudes. Tendencias actuales.

Los Sistemas de Gestión de Solicitudes no son más que aplicaciones informáticas que tienen como función principal permitir el acceso de los usuarios a determinados servicios, son una vía rápida y fácil para comunicarse con los proveedores de servicios desde cualquier lugar.

Por lo general estos sistemas están desarrollados en los lenguajes de programación PHP, Perl, ASP o Java. Los datos y la configuración son almacenados en una base de datos o una serie de archivos de texto. Cada versión provee funciones o capacidades diferentes sobre todo en la forma de procesar las solicitudes. En su mayoría se encuentran incluidos en sitios web de pequeñas y grandes empresas tanto informáticas como de otras ramas.

Se realizó un estudio para analizar las características y tendencias actuales de los sistemas de gestión de solicitudes y servicios, existen numerosos sistemas de este tipo, la mayor parte de ellos son muy caros o es muy difícil la adaptación de los mismos a las características de la universidad. En este acápite se mencionan algunos de estos sistemas y se brinda una breve descripción de los mismos, tomando en cuenta características generales que permiten identificar las tendencias actuales en el diseño de estos sistemas.

Sistemas de Gestión de Solicitudes en el Mundo.

SICOP Soluciones.

SICOP Soluciones constituye una organización especializada en la aplicación de tecnologías de Internet y Cómputo a los negocios como fuente de ventajas competitivas. Su sistema ofrece a los usuarios distintos servicios encaminados a brindar soporte entre otras facilidades.

Entre los servicios que ofertan se encuentran:

- Internet Web Hosting.
- Servicios para PyMEs.
- Servicios de Redes.
- Servicios en Línea.
- Mantenimiento de Sitios Web. (2)

Este sistema se encuentra montado sobre el CMS Mambo, utilizando PHP como lenguaje de programación, presenta una estructura similar a la que se desea construir aunque no es posible su utilización en la construcción de la solución debido a que los fabricantes del producto no respondieron a las solicitudes de envío de código de los autores de este trabajo.

Sagelogiccontrol.

Este sistema constituye una Web de Soporte que brinda un servicio de Asistencia técnica On-Line pensado para y por los usuarios que, sin necesidad de tener grandes conocimientos de informática y a través de un interface intuitivo, puedan acceder y beneficiarse de un portal de servicios exclusivos para sus clientes. Entre los servicios que brinda este sistema se encuentran los siguientes:

- Actualizaciones On-Line: Permite la descarga de las actualizaciones de los Programas, de forma inmediata, con un importante ahorro de tiempo y costos.
- Novedades legales: Acceso y recepción diaria de las resoluciones aparecidas en el resumen de prensa sobre temas legales y artículos de interés.
- Información técnica: Facilita de forma anticipada, toda la información relativa a las últimas noticias respecto a sus aplicaciones (cambios legales, mejoras, acceso a la Base de Datos.)
- Telemantenimiento: Servicio inmediato de intervención remota mediante un software específico y una conexión telefónica. (3)

Aún cuando la actividad fundamental de este sistema no es brindar soporte, comprende un área destinada a la gestión de servicios a sus clientes. Este sistema de gestión se programó en el lenguaje ASP, no es un sistema de código abierto por lo que se hace imposible la obtención del código fuente para la reutilización y modificación del mismo de acuerdo a las características especiales del negocio que se estudia.

Sistema de Gestión de Solicitudes de Business Objects.

Sistema de Gestión de Solicitudes que permite crear y enviar solicitudes de soporte técnico en línea. Al conectarse el usuario a la URL de la compañía tiene la posibilidad de realizar las solicitudes de soporte en línea. Cuando se crea una solicitud es inmediatamente enviada a equipos de soporte. Aunque la interfaz inicial del sistema se encuentra en idioma inglés se da la posibilidad de teclear la solicitud en cualquier lengua, siendo atendida posteriormente.

Entre las cualidades que posee este sistema está la posibilidad de gestionar los registros de productos de los usuarios, verificar el estado de sus solicitudes en cualquier momento e intercambiar notas y comentarios con el especialista de soporte encargado de las solicitudes. También se almacenan informaciones sobre los entornos técnicos de los productos, ahorrando las repeticiones de las mismas informaciones técnicas cada vez que se haga una solicitud. (4)

La utilización de las ventajas de este sistema no fue posible debido a que el mismo es un software propietario y por tanto se encuentra fuera de los lineamientos de la universidad.

Sistemas de Gestión de Solicitudes en Cuba.

Babel es un sistema de gestión de solicitudes cubano que integra las tecnologías de la información a la gestión de solicitudes de los servicios de traducción e interpretación. Este sistema ofrece a sus usuarios la información precisa sobre el estado de su solicitud y además, las competencias del traductor, al aumentar el valor añadido de cada recurso que interviene en el proceso. Esta herramienta de trabajo permite la organización, clasificación de la información y la recuperación de documentos con oportunas normas de seguridad.

Características tecnológicas del sistema de gestión de solicitudes Babel:

1. Es una aplicación programada en PHP.
2. Para la gestión de los datos se utiliza el Sistema Gestor de bases de datos relacionales MySQL.

Babel v2.0 es una solución propuesta por la Facultad de Ingeniería Eléctrica de la Universidad Central de Las Villas para la Unidad de Traducción del Centro de Información. Los objetivos perseguidos con este sistema:

Realizar una aplicación capaz de administrar los diferentes flujos de trabajo que se generan en cada servicio solicitado. Lograr un sistema integrado de gestión vía Web en el que se conjugarán tres interfaces fundamentales: la de usuario, traductor y administrador.

- Permitir que la aplicación realice búsquedas dentro de los datos almacenados, de acuerdo a diferentes criterios de selección.
- Generar reportes para saber el estado actual de las solicitudes que se realizan.

Este sistema aparece como una poderosa herramienta que integra y facilita la utilización de los servicios de traducción a través de la red. En la Intranet de ETECSA su uso provee un considerable ahorro de llamadas y transportación entre entidades geográficamente distantes que requieran de los servicios de traducción e interpretación, permite informatizar y optimizar el proceso de traducción. (5)

Los autores de este sistema establecieron contacto con los autores del sistema Babel quienes informaron que no estaban autorizados a proveer el código fuente de la aplicación por lo que no se utiliza ningún elemento de este sistema en la construcción de la solución.

Sistemas de Gestión de Solicitudes en la UCI.

En la Universidad de las Ciencias Informáticas no existe ningún sistema encargado de gestionar las solicitudes realizadas por los usuarios que desean beneficiarse con un determinado servicio de soporte.

1.2 ¿Qué es un CMS?

Un Sistema de gestión de contenidos (Content Management System en inglés, abreviado CMS) es un programa que permite crear una estructura de soporte para la creación y administración de contenidos por parte de los participantes. 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 y el diseño.

Los CMS primariamente fueron desarrollados para la creación de sitios interactivos y más adelante para manejar grandes volúmenes de contenidos.

1.2.1 Distintos tipos de CMS

Podemos agrupar los CMS existentes actualmente según el tipo de sitio que permiten gestionar. A continuación se muestran los más representativos:

- Genéricos: Ofrecen la plataforma necesaria para desarrollar e implementar aplicaciones que den solución a necesidades específicas. Pueden servir para construir soluciones de gestión de contenidos, para soluciones de comercio electrónico, blogs, portales,

- Específicos para ONG: Nacen para cubrir las necesidades de las ONG (Organización no Gubernamental), ofreciendo una plataforma de servicios de Internet que en ocasiones incluye además del CMS herramientas para el fundraising, entiéndase como el proceso de recaudación de fondos, mediante la solicitud de donaciones de particulares, empresas, fundaciones benéficas, o agencias gubernamentales.
- Foros: Sitios que permiten la discusión en línea donde los usuarios pueden reunirse y discutir temas en los que están interesados.
- Blogs: Publicación de noticias o artículos en orden cronológico con espacio para comentarios y discusión.
- eCommerce: Permite la creación de Sitios web para comercio electrónico.
- Portal: Sitio web con contenido y funcionalidad diversa que sirve como fuente de información o como soporte a una comunidad. Permite administrar y generar automáticamente un portal o sitio web que muestra contenido audiovisual, normalmente imágenes.
- e-Learning: Sirve para la enseñanza de conocimientos. Los usuarios son los profesores y estudiantes.
- Publicaciones digitales: Son plataformas especialmente diseñadas teniendo en cuenta las necesidades de las publicaciones digitales, tales como periódicos y revistas.

La finalidad de estas aplicaciones es proveer a los usuarios de un sistema que permita el desarrollo de aplicaciones Web que gestionen o manipulen información y en sentido general contenidos de forma dinámica sin la necesidad de tener conocimientos técnicos de programación y diseño de aplicaciones en general; además brindan la posibilidad de que se le añadan funcionalidades a través de addons o plugins que enriquecen los servicios que brindan.

De acuerdo a las características de la situación problemática planteada que conlleva a pensar en una solución diferente de un portal, foro o publicación digital se hace efectiva la utilización de un CMS que brinde la posibilidad de construir un sistema específico y adecuado a las necesidades que tiene actualmente la Dirección de Informatización, por lo que se decide usar un CMS genérico, permitiendo disminuir el tiempo de puesta en marcha del sistema.

1.2.2 Puntos importantes que hacen útil y necesaria la utilización de un CMS.

- 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.

- 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.
- 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 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.
- Cambios del aspecto de la web. Los CMS facilitan los cambios con la utilización, por ejemplo, del estándar Cascading Style Sheets u hojas de estilo en cascada (CSS) con lo que se consigue la independencia de la presentación y el contenido.
- Consistencia de la web. La consistencia en una aplicación web no quiere decir que todas las páginas sean iguales, sino que existe un orden (visual) en vez de caos. Los CMS pueden aplicar un mismo estilo en todas las páginas con el mencionado CSS, y aplicar una misma estructura mediante patrones de páginas, además al basarse en plantillas toman el control total de la imagen y la percepción del contenido de su sitio.
- Control de acceso. Controlar el acceso a un sistema web no consiste simplemente en permitir la entrada a la web, sino que gestiona los diferentes permisos a cada área aplicados a grupos o individuos.”

(7)

1.2.3 Principales Sistemas de Gestión de Contenidos del Mercado.

Drupal.

El sistema de administración de contenidos Drupal de código abierto fue creado por Dries Buytaert. Se encuentra disponible para Apache o IIS (Internet Information Service), multiplataforma disponible para Unix, Linux, BSD, Solaris, Windows, Mac OS.

Este administrador permite publicar contenidos, incluye foros y blog. Esta información es almacenada en una base de datos. Los contenidos del sitio son editados a través de un panel de administración web e incluye múltiples idiomas. En la configuración del sitio se pueden administrar los temas, configurar reportes de errores, sistemas de archivos, localización, funcionamiento, información y mantenimiento del sitio.

Características generales de Drupal

- Friendly URLs - Drupal usa el mod_rewrite de Apache para permitir personalizar las URL para optimización en los motores de búsqueda.
- Módulos - La comunidad de Drupal ha contribuido con muchos módulos que proporcionan la funcionalidad extra a Drupal.
- Ayuda en línea - Al igual que muchos proyectos de código abierto, cuenta un robusto sistema de ayuda en línea.
- De código abierto - El código fuente de Drupal está libremente disponible bajo los términos de la Licencia Pública General de GNU (GPL) 2. A diferencia de los sistemas de gestión de contenido de pago, Drupal tiene disponible el código fuente en su totalidad para ampliar o adaptar, según sea necesario.
- Personalización - Un robusto entorno de personalización está en el núcleo de Drupal. Tanto el contenido y la presentación pueden ser individualizados basados en las preferencias definidas por el usuario.
- Permisos basados en roles - Los administradores de Drupal no tienen que configurar los permisos para cada usuario, simplemente asignar usuarios a roles y darles permisos.
- Búsquedas - Todo el contenido en Drupal está en el índice de búsqueda, lo que facilita la información al usuario.

Gestión de usuarios

- Autenticación de los usuarios - Los usuarios se pueden registrar y autenticar a nivel local ó utilizando una fuente de autenticación externa como Jabber, Blogger ú otro sitio de Drupal. Para uso en una intranet, Drupal se puede integrar con un servidor LDAP (Protocolo Ligeró de Acceso al Directorio).

Gestión de contenidos

- Encuestas - Drupal viene con un módulo de encuesta que permite a los administradores ó usuarios crear encuestas y mostrarlas en distintas páginas.
- Plantillas - El sistema de plantillas de Drupal separa el contenido de la presentación permitiendo controlar el aspecto del sitio. Las plantillas se crean con el estándar de codificación HTML y PHP lo que significa que no se tiene que aprender un lenguaje propio de plantillas.
- Hilos de comentarios - Drupal ofrece un potente modelo de hilos de comentarios para permitir discusiones sobre el contenido publicado. Los comentarios son jerárquicos, como en un grupo de noticias o foro.
- Control de versiones - El control de versiones de Drupal rastrea las actualizaciones del contenido, registra la fecha y hora de los cambios realizados.

Bloggíng

- Soporte para Blogger - Esta característica permite a los sitios de Drupal que se actualicen mediante diferentes herramientas.

Plataforma

- Soporte para Apache ó IIS, Unix / Linux / BSD / Solaris / Windows / Mac OS X - Drupal ha sido diseñado desde el principio para ser multi-plataforma. No sólo se puede utilizar con Apache o IIS de Microsoft, también se puede usar Drupal en Linux, BSD, Solaris, Windows y Mac OS X.
- Independencia de la base de datos -Drupal es construido con una capa de abstracción de bases de datos que le permite utilizar Drupal con MySQL y PostgreSQL, y otras bases de datos propietarias.
- Multi-idioma - Drupal está diseñado para satisfacer las necesidades de una audiencia internacional y proporciona un marco completo para crear un sitio web en varios idiomas. Todo el texto puede ser traducido utilizando una interfaz gráfica de usuario, mediante la importación

de las traducciones existentes, o mediante la integración con otras herramientas de traducción como el GNU gettext.

Administración y análisis

- Análisis, Seguimiento y Estadísticas - Drupal permite la impresión de informes basados en navegador con información sobre referencias, el contenido y la forma de navegación de los usuarios de un sitio.
- Presentación de informes - Todas las actividades importantes y eventos del sistema son capturados en un registro de eventos para ser revisados por un administrador en un momento posterior.
- Administración vía web - Drupal se puede administrar completamente mediante un navegador web, lo que permite acceder a ella desde todo el mundo y no requiere software adicional para ser instalado en el equipo.

Comunidad

- Foros de discusión - Drupal posee un completo foro de discusión para crear sitios comunales dinámicos.

Rendimiento y escalabilidad (8)

- Almacenamiento en caché.

Ventajas:

- Creación de sub-apartados en los foros.
- Método avanzado de clasificación de artículos.
- Acceso a la web muy rápido.
- Permite a los usuarios configurar su propio perfil en dependencia de los permisos otorgados por el administrador.

Desventajas:

- Es obligatorio cerrar sesión antes de salir del sitio.
- Los foros son un poco desorganizados. (9)

Joomla

El sistema Joomla! es un sistema de administración de contenidos de código abierto, desarrollado en PHP. La primera versión fue publicada en el 2005, una versión de Mambo a la que se le realizaron modificaciones utilizando Ajax.

Permite la publicación de contenidos en Internet e Intranet. La información y configuración es almacenada utilizando bases de datos. Joomla! incluye características para hacer caché de las páginas, incluye feed RSS, búsquedas del sitio, encuestas, calendarios y versiones imprimibles de las páginas.

Mediante el panel de control se pueden administrar los contenidos, definiéndolos como archivados o no publicados en los cuales se pueden definir fechas de inicio y final de su publicación, es posible además la administración de categorías, secciones, menú, lenguajes del sitio y cuentas de usuarios. También encontramos la configuración global donde se definen las descripciones del sitio, permisos, idiomas, estadísticas, servidor de correo, entre otros.

Características:

- Organización de las secciones y categorías del sitio web.
- Publicación de contenidos sin limitaciones utilizando un sencillo editor para dar formatos y estilos a los textos e imágenes.
- Escalabilidad e implementación de nuevas funcionalidades, permitiendo instalar, desinstalar y administrar componentes y módulos.
- Administración de usuarios, permitiendo enviar correos electrónicos masivos.
- Administración del diseño y aspecto estético del sitio mediante la utilización de plantillas.
- Administración de la navegación y del menú del sitio.
- Posee estadísticas de visitas, mostrando información del navegador, sistema operativo y visitas por páginas.

Ventajas:

- Extensiones existentes creadas por la comunidad de usuarios.
- Mejora sustancialmente la usabilidad, manejabilidad y escalabilidad.
- Permite la compatibilidad hacia atrás con versiones previas de componentes, plantillas, módulos y otras extensiones.

- Aumenta la integración de aplicaciones externas a través de servicios web y la autenticación remota, como el Protocolo Ligero de Acceso al Directorio (LDAP).

Desventajas:

- El código HTML generado no es semántico.
- Deficiente planificación de la interfaz administrativa.
- Limitación en las opciones para personalizar rangos de usuarios.
- Dependencia excesiva del Javascript en su Panel de Administración.
- Es un poco lento. (10)

Wordpress

El sistema de gestión de contenidos WordPress está enfocado a la creación de blogs. Desarrollado en PHP y MySQL bajo licencia GPL.

Posee un enorme crecimiento en su comunidad de usuarios, las causas principales son su licencia, facilidad de uso y sus características como gestor de contenidos. Además de su enorme comunidad de desarrolladores y diseñadores, que se encargan de crear plugins y themes para la comunidad.

Wordpress permite crear una bitácora o sitio y la instalación de múltiples blogs. Otra de sus funciones es la del sistema de creación de plantillas, que permiten al usuario activar o desactivar las plantillas.

Características:

- Fácil instalación, actualización y personalización.
- Múltiples autores.
- Separa el contenido y el diseño en XHTML y CSS.
- Posee herramientas de comunicación entre blogs
- Permite ordenar artículos en categorías y subcategorías.
- Instalación de plugins.

Ventajas:

- No necesita que el desarrollador posea conocimientos técnicos de PHP.
- Tiene un agregador de RSS estándar.

- Automáticamente se conecta y configura bases de datos MySQL sin requerir ningún esfuerzo o conocimiento.
- Automáticamente importa imágenes.
- Cuenta con categorías.
- Permite personalizar los menús. (11)

XOOPS

Xoops es un sistema de administración de contenidos, por sus siglas en inglés Extensible Object Oriented Portal System (Sistema de portal extensible orientado a objetos). Permite a los usuarios crear sitios web dinámicos. Está escrito en PHP y utiliza MySQL como servidor de Base de Datos, cuenta con licencia GPL.

Mediante Xoops se pueden crear módulos de noticias. Permite la incorporación de foros, enlaces, descargas y documentos para sitios con grandes cantidades de contenidos.

Características:

- Utiliza MySQL como servidor de base de datos.
- Sitio completamente modularizado. Permite la administración de todos los módulos del sitio.
- Soporte para idiomas multi-byte como el chino, japonés, chino simplificado y chino tradicional, coreano, etc.
- Permite la creación de grupos de usuarios.
- Los administradores pueden seleccionar los temas del sitio que deseen.

Ventajas:

- Es relativamente sencillo de manejar.
- Tiene múltiples módulos.
- Genera código HTML y CSS bastante válido. (9)

Luego de un análisis de las características comunes y específicas de cada CMS se considera que Drupal es el más indicado construir el sistema que proveerá de un mecanismo eficiente de gestión de solicitudes y servicios. Este CMS cuenta con un módulo nombrado flujo de trabajo que permite la asignación de estados a contenidos específicos, permitiendo restringir la publicación de los mismos,

esta funcionalidad se convierte en punto clave para la selección del mismo, pues es de vital importancia para la ejecución del proceso de gestión de solicitudes. Drupal posee gran compatibilidad con el sistema gestor de base de datos MySQL.

Drupal se presenta como uno de los entornos de desarrollo Open source más populares por el momento y elegido mejor gestor de contenido de 2007 por la Web Packt Publishing. Su amplia comunidad de usuarios y largos años de experiencia han permitido construir un software estable y potente sobre el que plantear proyectos web a pesar de su, todavía, prolongada curva de aprendizaje. Además en cuanto a la clasificación, organización, y ordenación de la información (categorización taxonómica), Drupal es notablemente superior a Joomla y Wordpress. (10)

1.3 Fundamentación de la metodología, lenguajes y herramientas a utilizar.

1.3.1 Procesos de desarrollo de software.

En la actualidad para llevar adelante exitosos proyectos informáticos, hay que hablar de procesos de desarrollo, tanto de software cómo de todas las metodologías que se usen. “El objetivo de un proceso de desarrollo es elevar la calidad del software a través de una mayor transparencia y control sobre los procesos, es labor del proceso de desarrollo hacer que esas medidas para aumentar la calidad, el costo y el tiempo, sean reproducibles en cada desarrollo. “ (11)

Proceso Unificado de Rational. RUP

El **Proceso Unificado de Rational (RUP)**, el original inglés *Rational Unified Process*) es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. RUP fue creado por Rational software, filial IBM está basado en el seguimiento de una serie de normas aplicadas a cuatro etapas del desarrollo software: iniciación, elaboración, construcción y transición.

Características esenciales

Los autores destacan que el proceso de software propuesto por RUP tiene tres características esenciales: está dirigido por los Casos de Uso, está centrado en la arquitectura, y es iterativo e incremental.

- **Gestión de requisitos:** RUP brinda una guía para encontrar, organizar, documentar, y seguir los cambios de los requisitos funcionales y restricciones. Utiliza una notación de Caso de Uso y escenarios para representar los requisitos.

- Desarrollo de software iterativo: Desarrollo del producto mediante iteraciones con hitos bien definidos, en las cuales se repiten las actividades pero con distinto énfasis, según la fase del proyecto.
- Desarrollo basado en componentes: La creación de sistemas intensivos en software requiere dividir el sistema en componentes con interfaces bien definidas, que posteriormente serán ensamblados para generar el sistema. Esta característica en un proceso de desarrollo permite que el sistema se vaya creando a medida que se obtienen o se desarrollan sus componentes.
- Verificación continua de la calidad: Es importante que la calidad de todos los artefactos se evalúe en varios puntos durante el proceso de desarrollo, especialmente al final de cada iteración. En esta verificación las pruebas juegan un papel fundamental y se integran a lo largo de todo el proceso. Para todos los artefactos no ejecutables las revisiones e inspecciones también deben ser continuas.
- Gestión de los cambios: El cambio es un factor de riesgo crítico en los proyectos de software. Los artefactos software cambian no sólo debido a acciones de mantenimiento posteriores a la entrega del producto, sino que durante el proceso de desarrollo, especialmente importantes por su posible impacto, son los cambios en los requisitos. Por otra parte, otro gran desafío que debe abordarse es la construcción de software con la participación de múltiples desarrolladores, posiblemente distribuidos geográficamente, trabajando a la vez en distintas plataformas. La ausencia de disciplina rápidamente conduciría al caos. La Gestión de Cambios y de Configuración es la disciplina de RUP encargada de este aspecto (11)

Modelado visual (usando UML).

“UML (Unified Modeling Language) o Lenguaje de Modelación Unificado es un lenguaje gráfico para especificar, construir, visualizar y documentar las partes o artefactos. Pueden ser artefactos: un modelo, una descripción que comprende el desarrollo de software que se basen en el enfoque Orientado a Objetos, utilizándose también en el diseño Web. UML usa procesos de otras metodologías, aprovechando la experiencia de sus creadores, eliminó los componentes que resultaban de poca utilidad práctica y añadió nuevos elementos.” (12)

“De forma general las principales características son:

- Lenguaje unificado para la modelación de sistemas.
- Tecnología orientada a objetos.

- Corrección de errores viables en todas las etapas.
- Aplicable para tratar asuntos de escala inherentes a sistemas complejos de misión crítica, tiempo real y cliente/servidor.

UML proporciona un lenguaje estándar para escribir planos de software. Sin embargo ningún lenguaje es suficiente para expresar todos los matices de todos los modelos en todos los dominios y en todo momento. Por lo tanto, UML ha sido diseñado para ser abierto-cerrado, haciendo posible extender el lenguaje de forma controlada. Los mecanismos de extensibilidad de UML incluyen:

- Estereotipos.
- Valores etiquetados.
- Restricciones.” (13)

Rational Rose

Rational Rose es la herramienta CASE que comercializan los desarrolladores de UML y que soporta de forma completa la especificación del UML 1.1.

Esta herramienta propone la utilización de cuatro tipos de modelos para realizar un diseño del sistema, utilizando una vista estática y otra dinámica de los modelos del sistema, uno lógico y otro físico. Permite crear y refinar estas vistas creando de esta forma un modelo completo que representa el dominio del problema y el sistema de software.

Rational Rose utiliza un proceso de desarrollo iterativo controlado, donde el desarrollo se lleva a cabo en una secuencia de iteraciones. Cada iteración comienza con una primera aproximación del análisis, diseño e implementación para identificar los riesgos del diseño, los cuales se utilizan para conducir la iteración, primero se identifican los riesgos y después se prueba la aplicación para que éstos se hagan mínimos.

Rose permite que haya varias personas trabajando a la vez en el proceso iterativo controlado, para ello posibilita que cada desarrollador opere en un espacio de trabajo privado que contiene el modelo completo y tenga un control exclusivo sobre la propagación de los cambios en ese espacio de trabajo.

También es posible descomponer el modelo en unidades controladas e integrarlas con un sistema para realizar el control de proyectos. Esta herramienta permite la generación de código en distintos lenguajes de programación a partir de un diseño en UML.

Rational Rose proporciona mecanismos para realizar la denominada Ingeniería Inversa, es decir, a partir del código de un programa, se puede obtener información sobre su diseño. (14)

1.3.2. Software libre.

Actualmente existe en la universidad y en el país en general una tendencia hacia el uso en grado ascendente del software libre, para una mejor comprensión del tema es necesario hacer énfasis en su concepto.

¿Qué es Software Libre?

Es el software que, una vez obtenido, puede ser usado, copiado, estudiado, modificado, mejorado y redistribuido libremente. El software libre suele estar disponible gratuitamente en Internet, o a precio del coste de la distribución a través de otros medios; sin embargo no es obligatorio que sea así y, aunque conserve su carácter de libre, puede ser vendido comercialmente. “El calificativo «libre» del software libre se refiere a libertad, no a gratuidad. (15)

1.3.3. Lenguaje de programación PHP.

PHP es un lenguaje de programación pensado en la Web de forma que es ideal para la creación de páginas dinámicas. PHP es la versión libre del sistema equivalente de Microsoft ASP.

PHP es un lenguaje encapsulado dentro de los documentos HTML por lo que permite que el diseñador gráfico del sistema pueda trabajar de forma independiente al programador. Por lo tanto una web dinámica programada utilizando este lenguaje contiene una serie de documentos PHP que el servidor Apache interpreta proporcionando al cliente documentos HTML con el resultado de las órdenes. (16)

El código escrito en PHP tiene la peculiaridad de ser independiente de la plataforma, esto quiere decir que una aplicación escrita en PHP puede ser ejecutada en cualquier sistema operativo, gracias a que existen intérpretes de PHP para casi o todos los principales sistemas operativos del mercado. (17)

PHP 5 es la última versión del lenguaje para la creación de páginas de servidor y aplicaciones web; una renovación total que se ha extendido a numerosos aspectos, como la interacción con otras tecnologías, la nueva sintaxis de orientación a objetos, y un sinnúmero de novedades más.

Las características de este lenguaje son:

- Lenguaje sencillo y de sintaxis cómoda, similar a C, Perl y Java.
- Permite las Técnicas de Programación Orientada a Objetos.

- Es interpretado, aunque resulta muy rápido por estar integrado con Apache.
- Utiliza un sistema propio de administración de recursos y dispone de un sofisticado método de manejo de variables, conformando un sistema robusto y estable.
- Multiplataforma a partir de la versión 4.0.
- Puede interactuar con muchos motores de bases de datos: MySQL, ODBC, Oracle, DB2, Microsoft SQL Server, Access, etc.
- Dispone de soporte IMAP, SMTP, FTP, SNMP, XML, NNTP, interfaces Flash Macromedia y generación dinámica de gráficos y documentos PDF.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Permite crear los formularios para la Web.
- Biblioteca nativa de funciones sumamente amplia e incluida.
- Completamente expandible. Está compuesto de un sistema principal, un conjunto de módulos y una variedad de extensiones de código. (18)

Hoy Cuba está apostando al software libre y de código abierto, por lo que en la UCI existen líneas definidas para el desarrollo de aplicaciones siguiendo esta alternativa para la producción, como dato curioso se tiene: que más del 70 % de las aplicaciones que circulan en la intranet de la UCI están realizadas sobre software libre y de ellas más del 90 % en PHP que es una de las tecnologías aprobadas por la Dirección de Informatización de la universidad por ser más liviano y fácil de utilizar en la mayoría de sitios web y portales. De aquí la importancia de que el Sistema de Entorno Integrado se construya dentro de las líneas definidas para la informatización de la UCI.

1.3.4. Sistemas Gestores de Base de Datos (SGBD).

“Las aplicaciones en red son cada día más numerosas y versátiles. Debido a la diversidad de lenguajes y de bases de datos existentes, la manera de comunicar entre unos y otras sería realmente complicada a gestionar de no ser por la existencia de estándares que nos permiten la realización de las operaciones básicas de una forma universal.” (19)

“Un SGBD tiene los siguientes objetivos específicos:

- Independencia de los datos y los programas de aplicación.
- Minimización de la redundancia.

- Integración y sincronización de las bases de datos.
- Integridad de los datos.
- Seguridad y protección de los datos.
- Facilidad de manipulación de la información.
- Control centralizado.

El Structured Query Language (SQL) es una herramienta para organizar, gestionar y recuperar datos almacenados en una base de datos informática. Como su propio nombre indica, es un lenguaje que se puede utilizar para interactuar con una base de datos. Además Constituye un potente SGBD que está totalmente habilitado para Web.” (20) Pero a pesar de todas las ventajas que presenta este SGBD, tiene el inconveniente de que, al igual que Oracle, no es un sistema libre.

En el mundo informático existen muchos SGBD con soporte SQL, los más utilizados son: DB2, Oracle, SQL Server, Sybase ASE, MySQL, PostgreSQL, y entre ellos los más usados son MySQL y PostgreSQL para las aplicaciones de software libre y SQL Server para las de software privativo.

MySQL.

Posee un diseño multihilo que permite soportar una gran carga de forma muy eficiente. MySQL fue creada por la empresa sueca MySQL AB.

Este gestor de bases de datos es, uno de los más usados en la actualidad. Esta gran aceptación es debida, en parte, a que existen infinidad de librerías y otras herramientas que permiten su uso a través de gran cantidad de lenguajes de programación, además de su fácil instalación y configuración.

Las principales características de este gestor de bases de datos son las siguientes:

- Aprovecha la potencia de sistemas multiprocesador, gracias a su implementación multihilo.
- Soporta gran cantidad de tipos de datos para las columnas.
- Dispone de API's en gran cantidad de lenguajes (C, C++, Java, PHP).
- Gran portabilidad entre sistemas.
- Soporta hasta 32 índices por tabla.
- Escalabilidad y límites

- Soporte a grandes bases de datos. MySQL Server soporta bases de datos que contienen 50 millones de registros.
- Gestión de usuarios y contraseñas, manteniendo un muy buen nivel de seguridad en los datos.

Sin lugar a duda, lo mejor de MySQL es su velocidad a la hora de realizar las operaciones, lo que le hace uno de los gestores que ofrecen mayor rendimiento. Su bajo consumo lo hace apto para ser ejecutado en una máquina con escasos recursos sin ningún problema. Las utilidades de administración de este gestor son envidiables para muchos de los gestores comerciales existentes, debido a su gran facilidad de configuración e instalación. Tiene una probabilidad muy reducida de corromper los datos, incluso en los casos en los que los errores no se produzcan en el propio gestor, sino en el sistema en el que está. El conjunto de aplicaciones Apache-PHP-MySQL es uno de los más utilizados en Internet en servicios de foro y de buscadores de aplicaciones. (21)

PostgreSQL.

“Este SGBD intenta ser un sistema de bases de datos de mayor nivel que MySQL, a la altura de Oracle, Sybase o Interbase. Posee Licencia BSD.

Ventajas:

- Por su arquitectura de diseño, escala muy bien al aumentar el número de CPUs y la cantidad de RAM.
- Soporta transacciones y desde la versión 7.0, claves ajenas (con comprobaciones de integridad referencial).
- Tiene mejor soporte para triggers y procedimientos en el servidor.
- Tiene características orientadas a objetos.

Inconvenientes:

- Consume más recursos y carga más el sistema.
- Limite del tamaño de cada fila de las tablas a 8k
- Es de 2 a 3 veces más lenta que MySQL.
- Menos funciones en PHP. (21)

¿Por qué MySQL?

Luego de analizadas las características y facilidades de los SGBD presentados que cumplen las líneas de informatización de la UCI, se decide usar MySQL como SGBD, por las siguientes razones:

- No se necesitará de un manejo complejo de la información.
- PHP maneja más fácil a MySQL que a PostgreSQL, debido a la gran cantidad de funciones que tiene explícitas.
- MySQL es multiplataforma.
- Es la elección perfecta para sistemas en el que la velocidad y el número de accesos concurrentes es algo primordial.
- Es una base de datos muy rápida en la lectura cuando utiliza el motor no transaccional MyISAM, pero puede provocar problemas de integridad en entornos de alta concurrencia en la modificación.
- Alta integración con el CMS Drupal.

1.3.5 Servidor HTTP Apache.

El servidor Web Apache es, según numerosos estudios, el principal servidor de la Web desde hace diez años. Respaldo por una comunidad de desarrollo brillante, se sustenta en un amplio número de personas y organizaciones, desde gigantes de la talla de IBM hasta consultores particulares.

Transparencia y diversidad son las principales características de Apache. Su arquitectura modular, construida sobre un pequeño núcleo, se adapta a las necesidades específicas de cada usuario.

Es el servidor web por excelencia, su robustez y estabilidad hacen que cada vez millones de servidores reiteren su confianza en este programa.

Apache es una muestra, al igual que el sistema operativo Linux, de que el trabajo voluntario y cooperativo dentro de Internet es capaz de producir aplicaciones de calidad profesional difíciles de igualar.

Características esenciales:

- Corre en una multitud de Sistemas Operativos, lo que lo hace prácticamente universal.
- Es una tecnología gratuita de código fuente abierto.
- Es un servidor altamente configurable de diseño modular.
- Trabaja lenguajes como Perl, PHP y otros lenguajes de script.

- Provee todo el soporte que se necesita para tener páginas dinámicas.
- Apache permite personalizar la respuesta ante los posibles errores que se puedan dar en el servidor. (22)

1.3.6 Dreamweaver.

Dreamweaver es una herramienta de diseño de páginas web muy avanzada, tal como se ha afirmado en muchos medios. Aunque sea un experto programador de HTML, el usuario que lo maneje, siempre encontrará en este programa razones para utilizarlo, sobre todo en lo que a productividad se refiere.

Cumple perfectamente el objetivo de diseñar páginas con aspecto profesional, y soporta gran cantidad de tecnologías:

- Hojas de estilo y capas.
- Javascript para crear efectos e interactividades.
- Inserción de archivos multimedia.

Además es un programa que se puede actualizar con componentes, que fabrica tanto Macromedia como otras compañías, para realizar otras acciones más avanzadas. El código generado por este programa es de buena calidad. (23)

1.4 Conclusiones.

Teniendo en cuenta la necesidad de la creación de una aplicación para la informatización del proceso de gestión de solicitudes de servicios y el análisis de las potencialidades del uso de las tecnologías que han sido propuestas a lo largo del presente capítulo se llegó a la conclusión de que el sistema utilizará como gestor de bases de datos MySQL, la programación se desarrollará con el lenguaje PHP y se utilizará el CMS Drupal. Como metodología para el desarrollo del sistema se ha decidido utilizar el Proceso Unificado de Desarrollo (RUP), metodología que hará uso del Lenguaje Unificado de Modelado (UML). La decisión de seleccionar esta metodología, además de ser objetivo de la institución y considerando la poca experiencia en el uso de otras metodologías de los autores se debe a la consideración de que las metodologías ágiles estudiadas, presentaron desventajas en su uso, fundamentalmente en la documentación del software lo que incidiría en el mantenimiento y mejora futura del sistema, así como la integración del módulo que se propone al Sistema Central de Soporte de la Universidad que en estos momentos se encuentra en análisis y diseño. También se tuvieron en

cuenta los lineamientos planteados por la Dirección de Seguridad Informática en conjunto con la Dirección de Informatización.

Capítulo II Características del sistema.

En este capítulo se realiza el modelado del negocio y el levantamiento de requerimientos, identificando los procesos que intervienen en el negocio actual así como los posibles a automatizar. Se determinan los Casos de Uso del negocio con los actores y trabajadores presentes en la realización de los mismos. Este capítulo contiene además los diagramas que visualizan la ejecución de los procesos identificados.

2.1. Objeto de estudio.

2.1.1 Problema y Situación Problemática.

La Universidad de las Ciencias Informáticas (UCI), basada en su estrategia de desarrollo, está enfrascada en el proyecto UCI-Ciudad Digital cuyo objetivo es crear una micro-sociedad informatizada, que pudiera constituir el modelo de sociedad cubana del futuro. Como parte de esta misión el área de Informatización realiza diferentes proyectos para la automatización de los servicios que se brindan en el centro.

Justamente, el objeto de estudio de este trabajo lo constituye la gestión de solicitudes de servicios de soporte que son brindados en dicha universidad, proceso que actualmente se realiza de forma manual utilizando para ello una planilla de solicitud, lo que conlleva a una pérdida sustancial de tiempo desde la solicitud hasta la autorización y correspondiente ejecución de los servicios. Se determina que el **problema científico** radica en ¿Cómo solucionar los problemas en el proceso de gestión de servicios y solicitudes que brinda el grupo de soporte del área de informatización?

2.1.2 Objeto de Automatización.

Serán objeto de automatización los procesos del negocio relacionados con la gestión de los servicios de soporte tanto de aplicaciones como de proyectos en desarrollo, así como la gestión de solicitudes. Existen otros procesos derivados de los mencionados que en consecuencia también serán objeto de automatización como es el caso de las apelaciones que existan por denegaciones de servicios.

Procesos a automatizar.

Gestionar Servicios.

Este proceso contiene las actividades de inserción, modificación y eliminación de servicios referentes al soporte y mantenimiento de aplicaciones y proyectos en desarrollo, es llevado a cabo por el grupo

de soporte que determina la disponibilidad de determinado servicio. Entre los servicios a brindar se encuentran Hospedaje de Base de Datos y aplicaciones, salvadas, control de versiones y gestión de publicaciones en el nodo central.

Gestionar Estado de la solicitud.

El objetivo principal de este proceso es determinar el estado de una solicitud de servicios. Estos estados pueden ser:

- Pendiente (no se ha procedido a la revisión todavía)
- Aprobado (solicitud aprobada servicio en vías de concesión)
- Denegado (se rechaza y tendrá anexado los motivos por los cuales se determinó este estado)
- Apelado (el usuario solicita revisión de la denegación de su solicitud.)

Llenado y trámite de las solicitudes.

Los representantes de los proyectos interesados en obtener algún servicio llenan una planilla de solicitud que transita por una jerarquía de aprobación hasta que el servicio solicitado pueda ser habilitado. El sistema que se pretende implementar agilizará este proceso.

Retroalimentación al interesado con las gestiones realizadas.

Actualmente la persona que llena una solicitud no es notificada en ningún momento del estado en que se encuentra su solicitud. A través del software que se propone los solicitantes pueden acceder a su solicitud y verificar el estado en el que se encuentra la misma en cualquier momento del proceso.

Trámite de la Solicitud de Apelación.

Este proceso brinda un mecanismo de apelación para cuando se deniega una solicitud de servicio. Si el solicitante no se encuentra conforme con la decisión o las condiciones iniciales que provocaron la denegación desaparecen. El grupo de soporte valora nuevamente la solicitud atendiendo a los motivos expuestos en la apelación.

2.1.3 Información que se maneja.

Se debe manejar toda la información de las solicitudes a los servicios en las que se incluye, el servicio solicitado, origen de la solicitud específicamente nombre del proyecto al que pertenece el solicitante, así como el rol que desempeña dentro del mismo. Es de suma importancia el manejo de la información

referente a los servicios de soporte brindados que incluye el nombre y una breve descripción de los mismos.

2.2 Propuesta del sistema.

Con el fin de anular la situación problemática planteada en este capítulo se propone un módulo dentro del sistema central de Soporte de la UCI que permita la realización de los procesos de Gestión de Solicitudes y Gestión de Servicios. El sistema contempla la esencia de los procesos del negocio y propone variaciones en la ejecución de los mismos.

2.3 Modelo de negocio.

Con el fin de obtener una visión clara de los procesos que intervienen en el negocio así como lograr un entendimiento entre los clientes y desarrolladores se realiza la modelación del negocio. Donde se describen los Casos de Uso del negocio, identificando quiénes participan y las actividades que requieren automatización.

2.3.1 Actores y Trabajadores del negocio.

Actores del negocio	Justificación
Solicitante	Se desempeña como representante de un proyecto convirtiéndose en usuario de los servicios del grupo de soporte. Es el que inicia los procesos referentes a las solicitudes.

Tabla 1 Actores del negocio

Trabajadores del negocio	Justificación
Proveedor de servicios	Persona que pertenece al grupo de soporte, estudia las solicitudes de servicios recibidas y determina el estado de las mismas (pendiente, aprobado, denegado, apelado) brindando o no el servicio solicitado.

Tabla 2 Trabajadores del negocio

2.3.2. Diagrama de Casos de Uso del negocio.

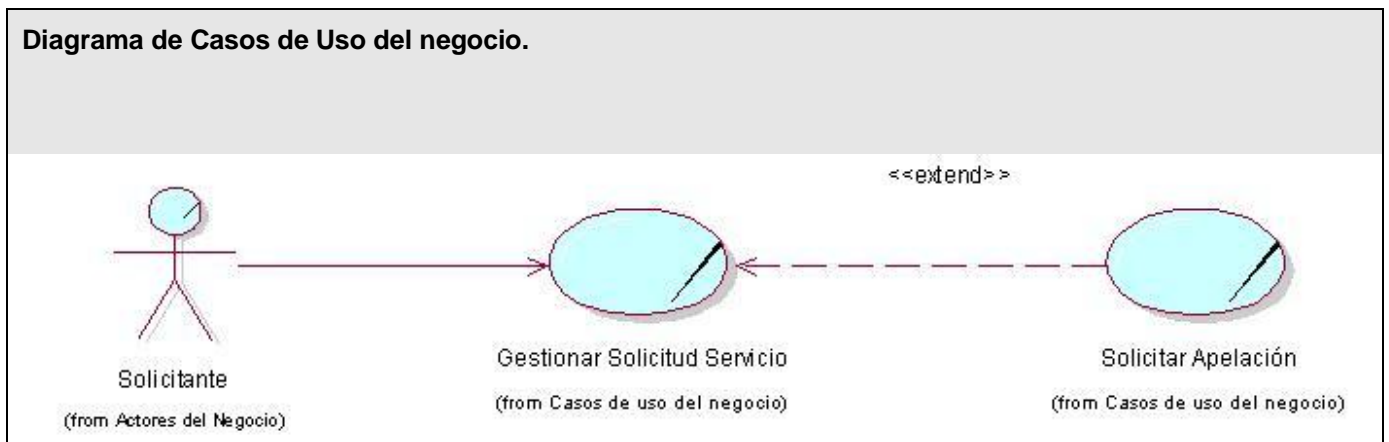


Figura 1 Diagrama de Casos de Uso del negocio

2.3.2 Descripción textual Casos de Uso del negocio.

Ver [Anexo 1](#).

2.3.3 Diagrama de Actividades.

Ver [Anexo 2](#).

2.3.4 Diagrama de clases del modelo de objetos.

Ver [Anexo 3](#).

2.4 Requerimientos del sistema

2.4.1. Especificación de los requisitos funcionales del software.

Los requisitos son las condiciones o capacidades que tienen que ser alcanzadas o poseídas por el sistema para satisfacer un contrato, estándar, u otro documento impuesto formalmente. O sea, los requisitos son los que definen qué es lo que el sistema debe hacer, para lo cual se identifican las funcionalidades requeridas y las restricciones que se imponen.

Módulo de administración

R1 Gestionar los Módulos del sistema.

R1.1 Mostrar el listado de módulos existentes.

R1.2 Desactivar módulo.

R1.3 Activar módulo.

R2 Gestionar los Bloques del sistema.

R2.1 Añadir un bloque al sistema.

R2.1.1 Entrar datos del bloque a crear.

R2.2 Activar bloque existente.

R2.2.1 Especificar los datos peso y ubicación del bloque dentro del sistema.

R2.3 Configurar un bloque.

R2.3.1 Modificar los datos de un bloque específico, estos son Descripción del bloque (obligatorio), Título del bloque y Cuerpo del bloque.

R2.4 Desactivar bloque existente.

R2.5 Eliminar un bloque.

R2.6 Mostrar listado de bloques existentes.

R3 Gestionar Tipos de contenido del sistema

R3.1 Mostrar el listado de tipos de contenido existentes.

R3.2 Crear un tipo de contenido.

R3.3 Eliminar un tipo de contenido.

Módulo de solicitante.

R4 Gestión de Solicitudes.

R4.1 Crear Solicitud de Servicio.

R4.2 Crear Solicitud de Apelación.

R4.3 Crear Solicitud de Publicación de Sitios Web.

R4.4 Mostrar Historial de Solicitudes

R5 Módulo de Proveedor

R5.1. Gestionar Reportes.

R 5.1.1 Obtener reporte de solicitudes por estado.

R5.1.2 Obtener reportes de solicitudes por servicios.

R5.2. Determinar estado de las solicitudes de servicio y publicación de sitios web.

R5.2.1 Aprobar Solicitud.

R5.2.2 Denegar Solicitud.

2.4.2 Requerimientos no funcionales del Sistema.

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

Apariencia o interfaz externa:

Los colores y formatos que identificarán al sistema tendrán una tonalidad azul refrescante y se utilizará un diseño para resoluciones de 800x600 y 1024X768.

- Interfaz con pocas imágenes para no demorar las respuestas al usuario.
- Diseño sencillo y claro, con reconocimiento visual a través de elementos visibles que identifiquen cada una de las funcionalidades.
- Diseño formal, serio y con una navegación sugerente, teniendo en cuenta el fin con el que se desarrolla la aplicación.

Rendimiento:

La disponibilidad de trabajo en red contra el servidor es constante. Teniendo como prioridad garantizar que el tiempo de respuesta a las solicitudes de los usuarios al sistema sea mínimo. El sistema deberá ser lo más estable y confiable posible.

Soporte:

Se usarán dos servidores, uno para la base de datos y otro para el servidor de aplicaciones.

Extensibilidad: Se garantizará la inserción de módulos nuevos en el futuro, ello sin afectar lo realizado hasta el momento. Las nuevas funcionalidades deberán estar debidamente documentadas.

Software:

- Navegador compatible o superior con Internet Explorer 4, o Netscape Navigator.
- Macromedia Dreamweaver MX 2004.
- Adobe Photoshop CS2.

- MySQL.
- Apache.
- PHP 5.

Portabilidad:

El sistema debe poder ejecutarse sobre los sistemas Windows y Linux.

Seguridad:

- Garantizar que cada usuario solo tenga los privilegios necesarios de acuerdo a su rol.
- Garantizar que las funcionalidades del sistema se muestren de acuerdo a los privilegios que tenga el usuario que este activo.
- Protección contra acciones no autorizadas o que puedan afectar la integridad de los datos.
- Verificación sobre acciones irreversibles (eliminaciones o modificaciones).

Confiabilidad:

- La herramienta de implementación CMS Drupal tiene soporte para recuperación ante fallos y errores.

Integridad:

- La información será protegida contra corrupción y estados inconsistentes.

Disponibilidad:

- El sistema deberá estar disponible durante las 24 horas del día, por lo que la Dirección de Informatización provee los recursos necesarios para garantizar la disponibilidad.

Usabilidad

- El sistema puede ser usado por cualquier persona, que posea conocimientos básicos en el manejo de la computadora y de un ambiente Web en sentido general.

Hardware:

- Para usar el sistema se necesita una computadora con procesador Pentium con al menos 500 MHz de velocidad, 5 GB de disco duro y 64 MB de Memoria RAM.
- El servidor donde se montará el sistema debe tener como mínimo 40 GB de disco duro, 256 MB de memoria RAM.

2.5 Modelo de Casos de Uso del Sistema.

Utilizando las facilidades que brinda el UML, se capturan los requisitos funcionales del sistema y se representan mediante un diagrama de Casos de Uso. Para ello se definen de acuerdo a lo planteado en los epígrafes anteriores, cuales serían los actores que van a interactuar con el sistema, y los Casos de Uso que van a representar las funcionalidades.

Teniendo en cuenta los requerimientos planteados en este trabajo y con la tarea de cumplir los objetivos trazados, en el sistema que se propone la existencia de tres módulos:

- Módulo de administración: se encargará de la configuración del sistema así como de la asignación de permisos para los distintos editores de contenido.
- Módulo de proveedor de servicios: se encargará de la creación del contenido servicios así como de la gestión de las solicitudes.
- Módulo de usuario: se encargará de la creación del contenido solicitud.

2.5.1 Definición de actores del Sistema.

Los actores representan a cualquier persona o sistema que interactúa con el sistema. Un actor puede que:

- Sólo brinde información de entrada al sistema.
- Sólo reciba la información del sistema.
- Brinde y reciba información.

ACTORES DEL SISTEMA	JUSTIFICACIÓN
Solicitante	Es el actor que representa a los representantes de los proyectos que acceden a los servicios de soporte y mantenimiento mediante el sistema.
Administrador	Representa a una persona que configura y controla el comportamiento del sistema, así como la asignación de permisos para los usuarios del mismo.

<p>Proveedor de Servicios</p>	<p>Representa a los usuarios que pueden aprobar o denegar solicitudes así como insertar o modificar información referente a los servicios dentro del sistema.</p>
-------------------------------	---

Tabla 3 Actores del Sistema.

2.5.2 Distribución de los Casos de Uso por paquetes.

Se agruparon los diferentes Casos de Uso del sistema en paquetes que engloban las funcionalidades específicas de cada actor y las comunes a cada grupo de actores. Los paquetes creados con sus Casos de uso correspondientes son:

Subsistema: Gestión de Solicitante.

- Solicitar Apelación.
- Solicitar Servicio.
- Solicitar Publicación de Sitios Web.

Subsistema: Gestión de Proveedor.

- Gestionar estados de las solicitudes.
- Gestionar Servicios.
- Solicitar Reportes.

Subsistema: Gestión de Administración.

- Gestionar Tipo de Contenido.
- Gestionar Bloques.
- Gestionar Módulos.

A continuación se ofrece una breve descripción de cada uno de los subsistemas:

Subsistema Gestión de Solicitante: Engloba todas las funcionalidades referentes al módulo de solicitante. Se destacan los Casos de Uso Solicitar Servicio y Solicitar Apelación en estos Casos de Uso el usuario tendrá la posibilidad de crear y enviar una nueva solicitud que será revisada en momentos posteriores.

Subsistema Gestión de Administración: Comprende todo el comportamiento de la gestión de configuración del sistema. Se encuentran entre sus Casos de Uso: Gestionar Tipo de Contenido, Gestionar Bloques y Gestionar Módulos.

Subsistema Gestión de Proveedores: Contiene las principales funcionalidades de gestión de solicitudes y servicios. Entre sus Casos de Uso se encuentran Gestionar Servicio y Gestionar Estado de las Solicitudes

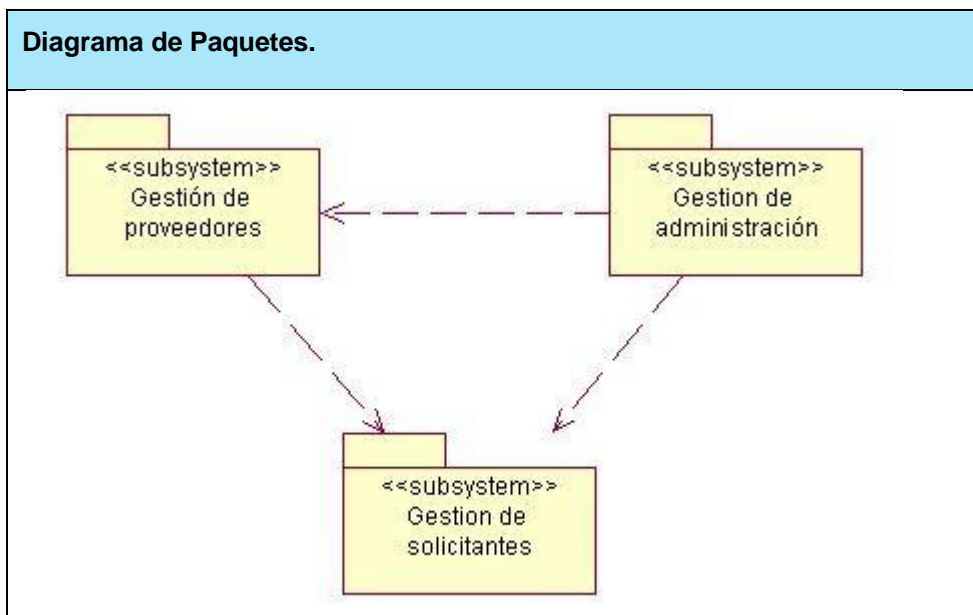


Figura 2 Diagrama de Paquetes

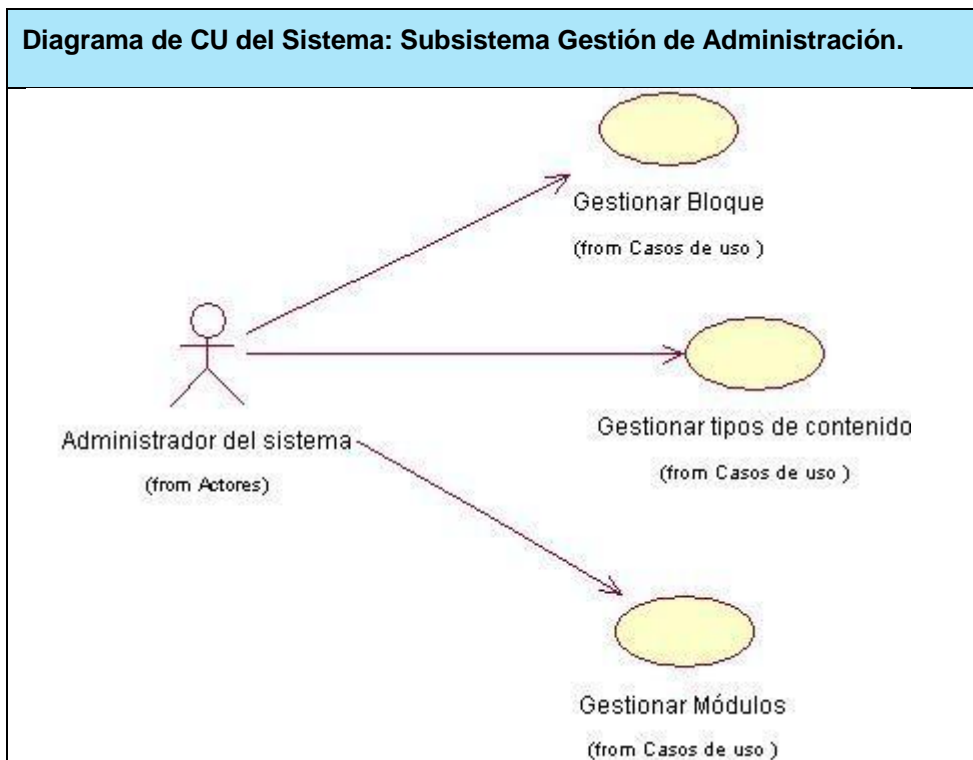


Figura 3 Subsistema Gestión de Administración

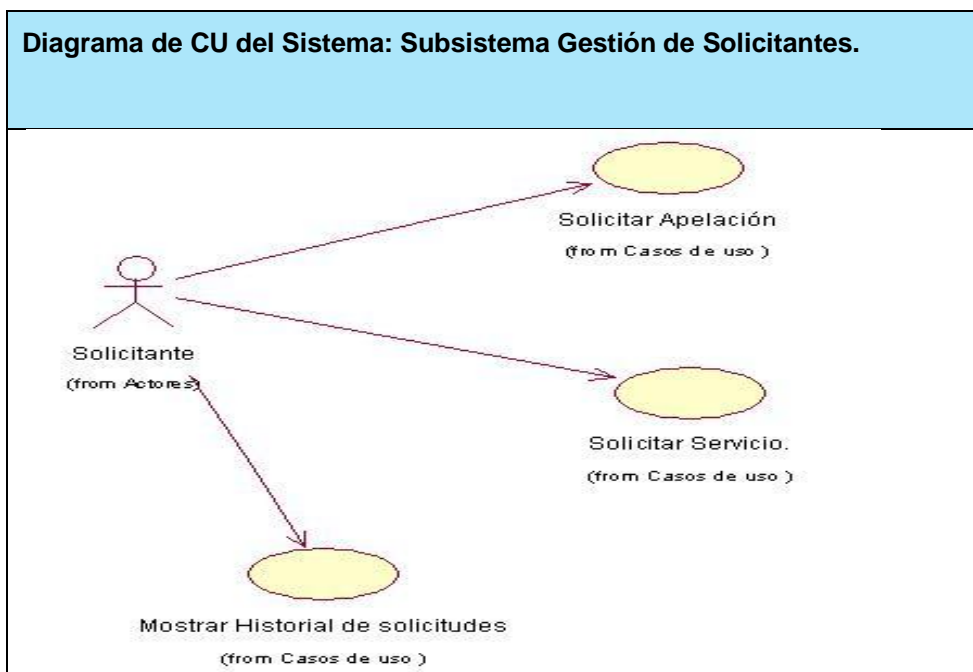


Figura 4 Diagrama de CU Subsistema Gestión de Solicitantes

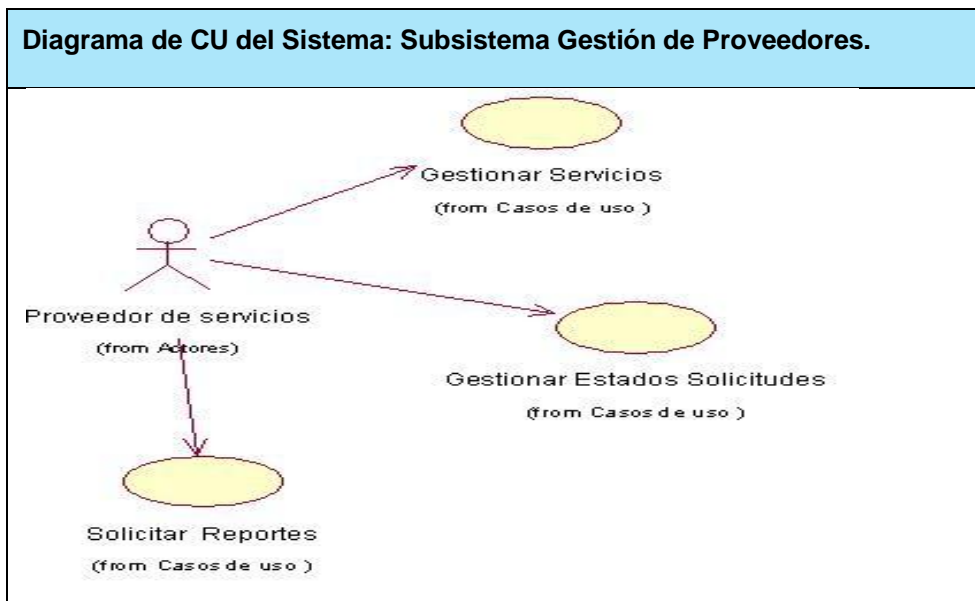


Figura 5 Diagrama de CU Subsistema Gestión de Proveedores.

2.5.3. Diagrama de Casos de Uso del sistema.

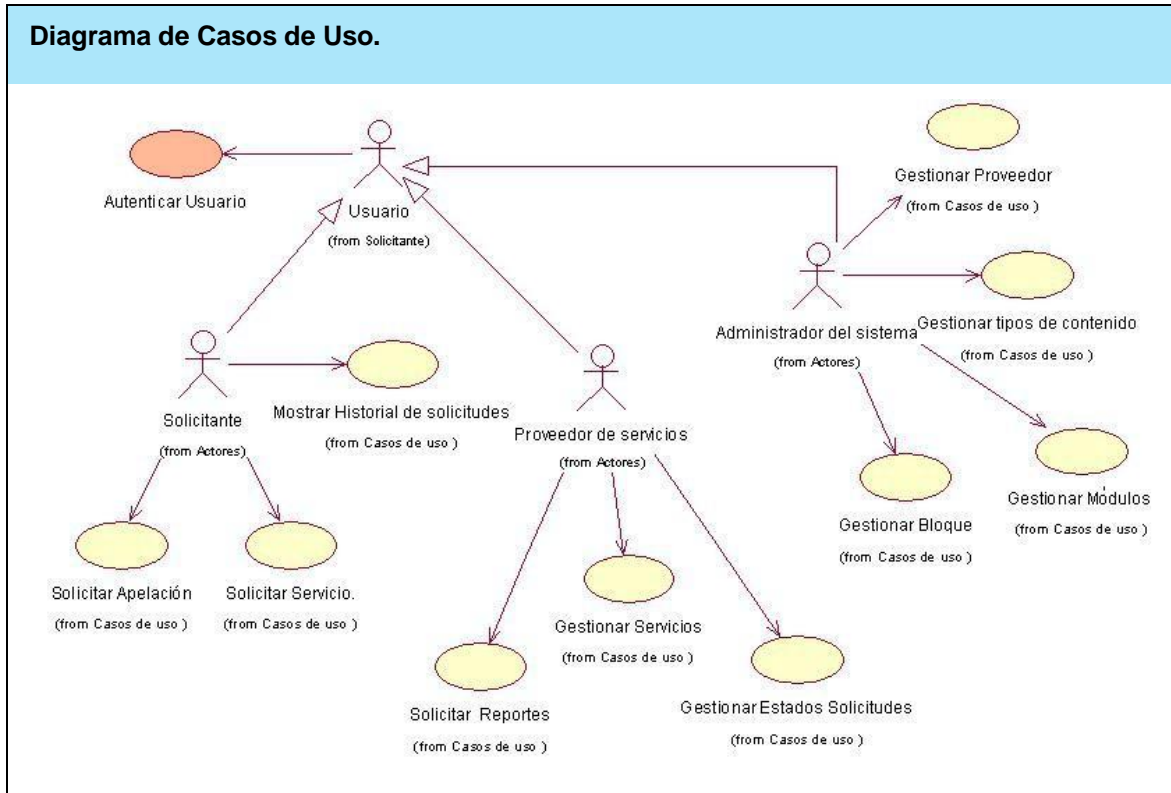


Figura 6 Diagrama de CU del Sistema

2.5.4 Descripción de los Casos de Uso del Sistema.

En esta sección se presenta la descripción de los Casos de Uso, a éstos han sido asociadas las pantallas correspondientes que muestra el sistema que se obtuvo como resultado de esta investigación.

Mediante los Casos de Uso expandidos se describe paso a paso la secuencia de eventos que los actores utilizan para completar un proceso a través del sistema. Ver [Anexo4](#)

2.6. Conclusiones.

En este capítulo se desarrolló la propuesta de solución del sistema a partir del análisis de los procesos del negocio. Se representaron los procesos presentes en el sistema a través del Diagrama de Casos de Uso del Sistema. Comenzando la construcción del sistema, con el planteamiento de los requisitos y la definición de los Casos de Uso del Sistema.

Capítulo III Análisis y diseño del sistema

Este capítulo incluye todos los elementos del análisis y diseño del sistema y describe cada uno de los modelos que componen este flujo de trabajo. Se desarrolla una propuesta de solución en la cual se hace uso del Sistema de Gestión de Contenidos Drupal. Además se presenta el diseño de la base de datos a través del diagrama de clases persistentes y modelo físico de datos.

3.1 Diagramas de clases del análisis

El análisis tiene como objetivo fundamental la comprensión de los requisitos del software, es un modelo general que no depende del lenguaje de programación que será utilizado en la construcción de la solución propuesta.

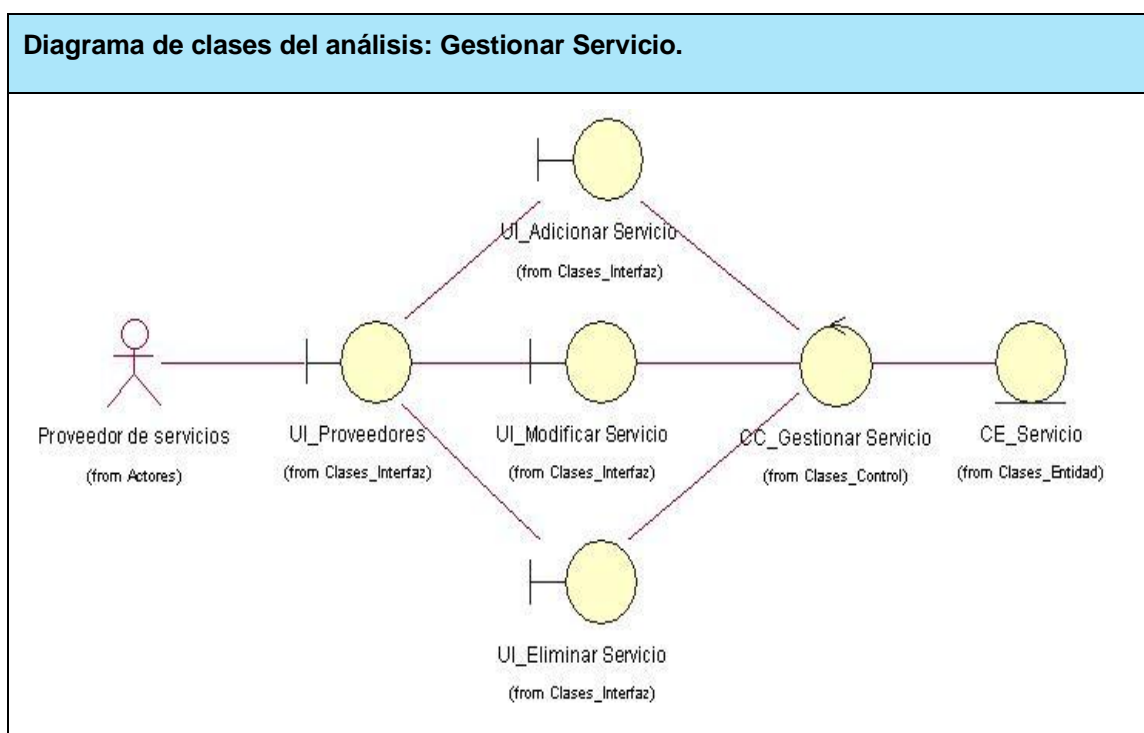


Figura 7 Diagrama de clases del análisis: Gestionar Servicio

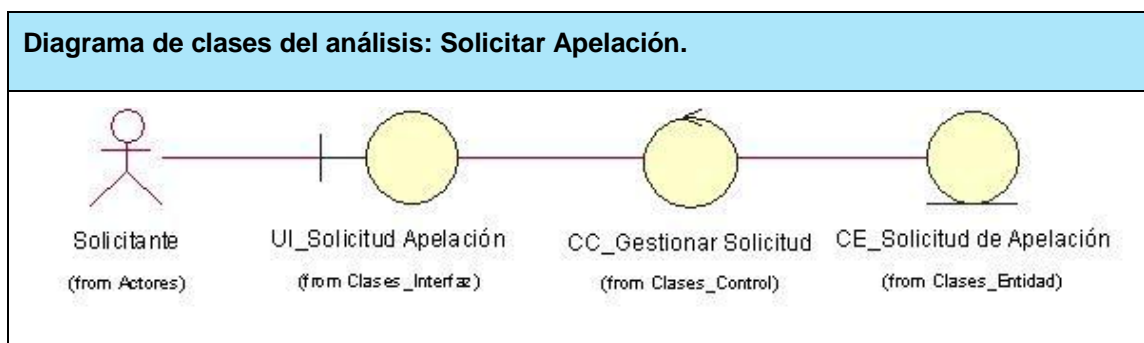


Figura 8 Diagrama de clases del análisis: Solicitar Apelación.

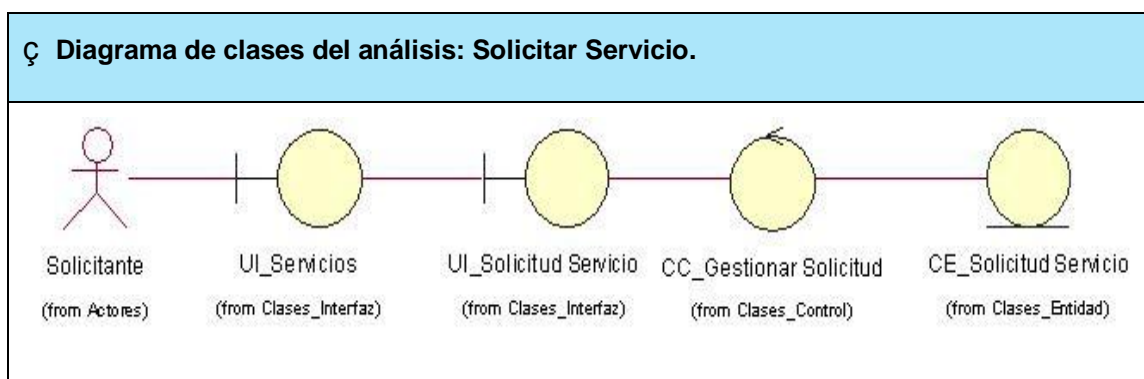


Figura 9 Diagrama de clases del análisis: Solicitar Servicio.

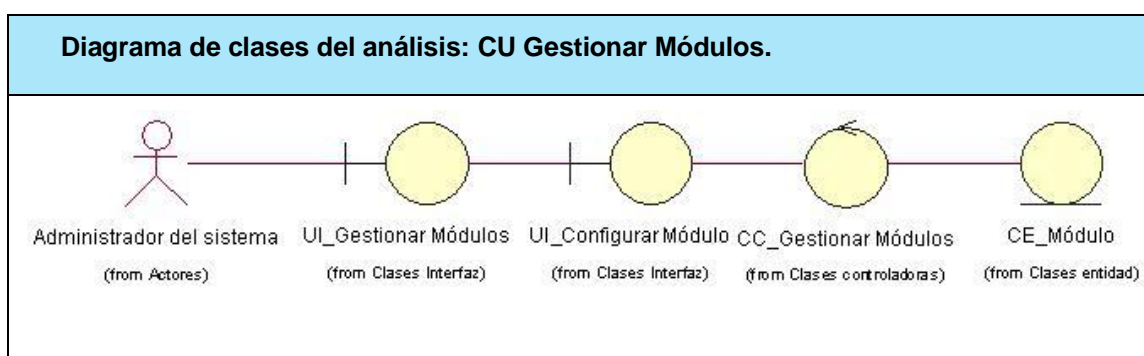


Figura 10 Diagrama de clases del análisis: CU Gestionar Módulos.

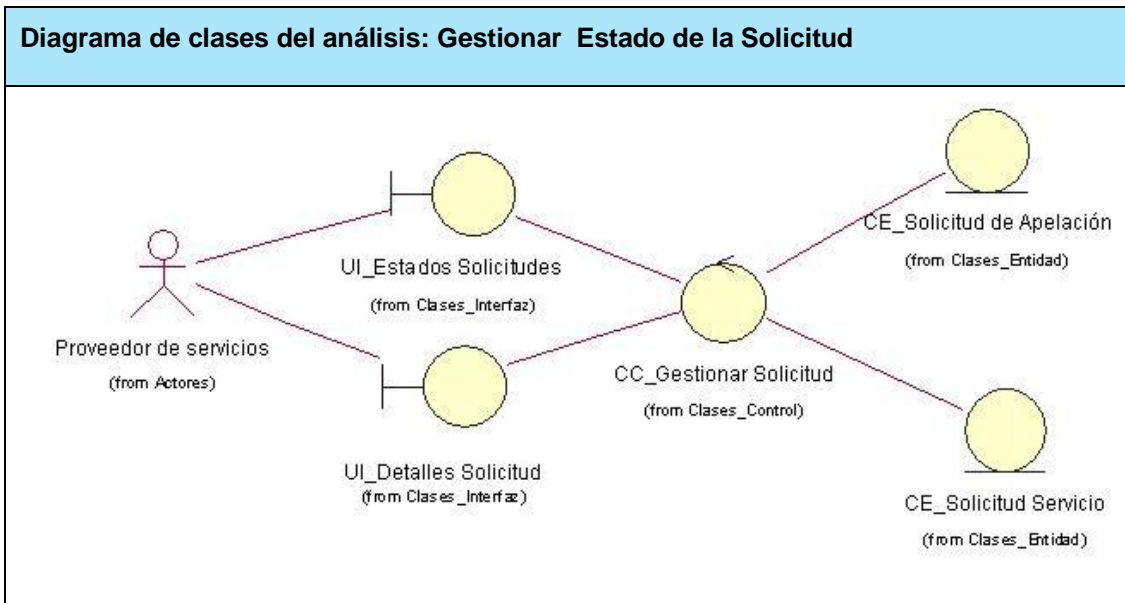


Figura 11 Diagrama de clases del análisis: Gestionar Estado de la Solicitud

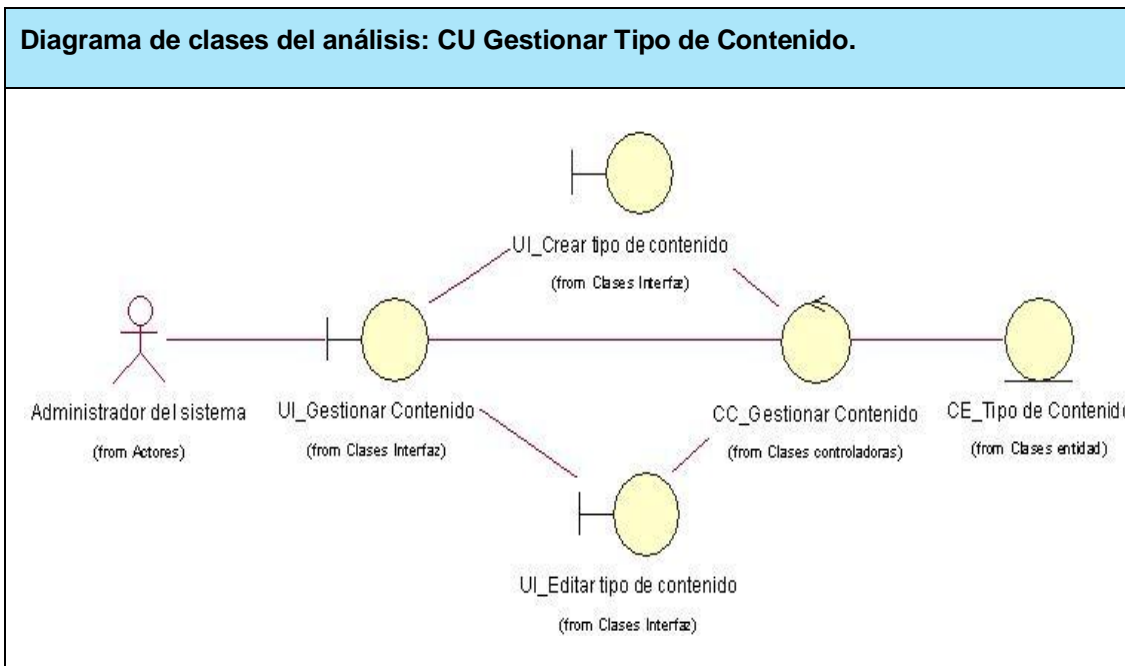


Figura 12 Diagrama de clases del análisis: CU Gestionar Tipo de Contenido.

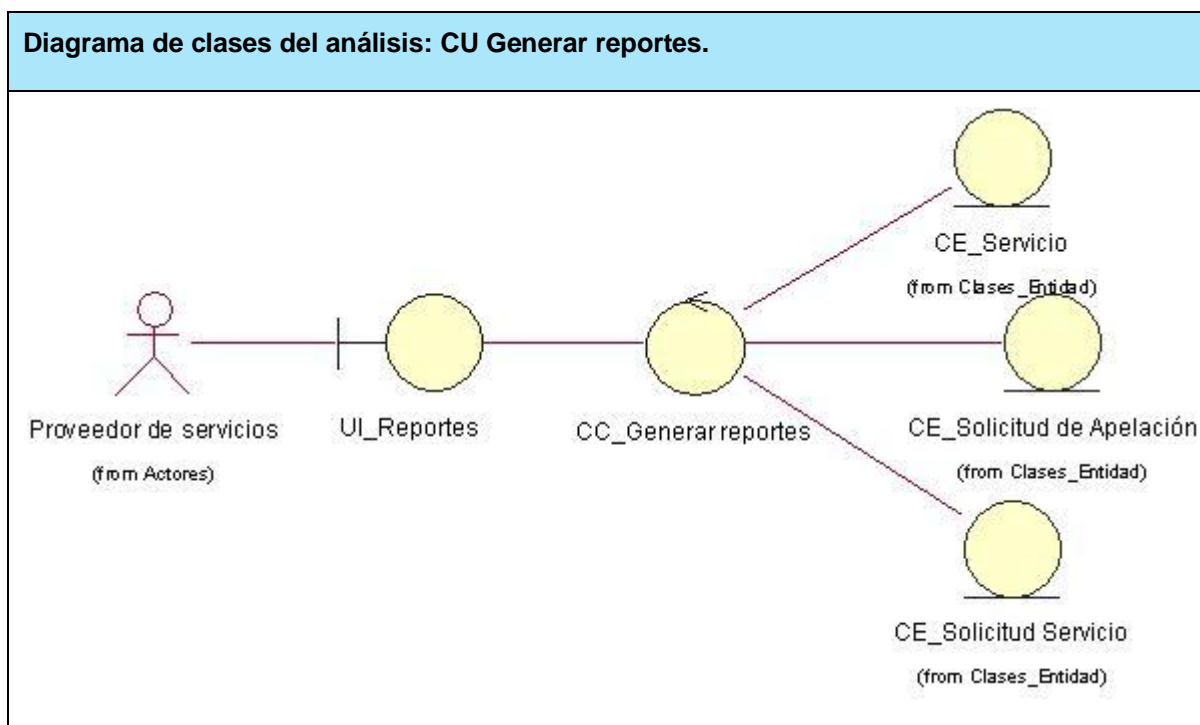


Figura 13 Diagrama de clases del análisis: CU Generar reportes.

Diagramas de Interacción

Ver [Anexo 5](#)

3.2. Diseño

Drupal es un entorno de desarrollo para la gestión de contenidos, extensible, escrito en el lenguaje PHP. El núcleo de Drupal se complementa con un importante número de **extensiones** que son las que ofrecen verdadera funcionalidad a este entorno. Podría verse como un enorme juego de Lego en el que los desarrolladores deciden cómo combinar y configurar los diferentes módulos para crear su nuevo sitio web.

Este potente CMS es altamente escalable y permite una fácil adaptación dentro de las metodologías de la ingeniería web. Desde el punto de vista del desarrollador, plantear proyectos con esta herramienta favorece la agilización del proceso de puesta en marcha e implementación de un sistema, permitiendo dedicar mayor tiempo al análisis y evaluación del negocio.

En esta sección del capítulo se explica cómo se integra el diseño web de la aplicación a las estructuras definidas en el CMS Drupal.

Drupal introduce el concepto de **nodo** como sinónimo de tipo de contenido, cualquier recurso que se ingrese al sistema pasa a ser un nodo, entiéndase por nodo artículos, historias, posts, encuestas, imágenes, libros colaborativos, reseñas y otros. Este nuevo concepto permite estandarizar la información asignando las mismas características a distintos objetos y la posibilidad de tener toda la información catalogada.

Las características básicas de Drupal son suficientes como para lanzar un sitio completo, el núcleo incluye: administración de usuarios, páginas, vocabulario, comentarios y sindicación. Su potencialidad reside en su estructura modular que permite ser extendida y personalizada a gusto y placer del administrador para distintas funciones por medio de módulos de fácil instalación. Los módulos contienen funcionalidades que pueden ser utilizadas o no dependiendo de la activación de los mismos en el núcleo del sistema. Los más significativos son aquellos que permiten la creación y visualización de todo tipo de contenido y la organización flexible de usuarios.

A continuación se explican brevemente los módulos activados para el desarrollo de la aplicación:

- **Workflow:** El módulo permite la creación y asignación arbitraria de los flujos de trabajo a los tipos de contenido. Entiéndase por flujo de trabajo a una serie de estados que puede tomar un tipo de contenido determinado y las transiciones que pueden sucederse entre ellos. En el sistema propuesto se hace necesaria la utilización de este módulo, se activa para que sea aplicable al tipo de contenido solicitud de servicio y solicitud de publicación de sitios web. Se crea un flujo de trabajo denominado Solicitudes que permite asignarle a los contenidos mencionados uno de los estados que se listan a continuación:
 - Aprobado.
 - Denegado.
 - Pendiente.
 - Apelado.
- **Workflow Access:** Este módulo permite definir individualmente el control de acceso de los usuarios a los flujos de trabajos definidos en el módulo descrito anteriormente. Se definieron los siguientes permisos:

Roles	Permisos
-------	----------

Proveedor de Servicios	Puede ejecutar transiciones de estados, determinando el estado final de las Solicitudes de Servicio y Solicitudes de Publicación de Sitios Web.
Solicitante	Puede visualizar tanto el estado que toma su solicitud como los comentarios realizados por el proveedor de servicios.

Tabla 4 Control de acceso al flujo de trabajo.

- **Workflow_ng:** Permite personalizar Drupal incorporándolo con el módulo Workflow de forma muy flexible, contiene reglas basadas en acciones impulsadas por el sistema, o sea, hace posible configurar acciones que se disparan sobre determinados acontecimientos.

Este módulo se activa con el fin de mantener al solicitante informado por medio de un correo electrónico del estado que toma su solicitud luego de que el proveedor realiza el proceso de revisión pertinente.

- **Views:** El módulo proporciona un método flexible para los diseñadores de sitios en Drupal, permite controlar cómo se presentan las listas de contenido (nodos). Esta herramienta es esencialmente un constructor de consultas inteligentes. Puede construir la consulta, ejecutarla, y mostrar los resultados. Se utilizó para la creación de vistas que muestran los servicios disponibles que son insertados a través de los proveedores de servicios, sobre él también se construyeron todas las vistas referentes al tratamiento de las solicitudes.
- **CCK:** Este módulo permite la creación de atributos o campos específicos para los tipos de contenido definiendo además el tipo de dato que pudiera recibir el campo. Se utiliza para la creación de los campos que se listan a continuación:

Tipo de Contenido Solicitud de Servicio.

Campo	Tipo de dato
Asunto de la solicitud	Campo de texto
Servicio a Solicitar	Referencia a nodo

Rol dentro del proyecto	Campo de texto
Proyecto productivo al que pertenece	Campo de texto
Perteneciente al área	Campo de texto

Tabla 5 Campos definidos para la Solicitud de Servicio.

Tipo de Contenido Solicitud de Publicación de Sitios Web.

Campo	Tipo de dato
Asunto de la solicitud	Campo de texto
Título de la Aplicación	Campo de texto
URL propuesta	Campo de texto
Breve descripción	Campo de texto
Perteneciente al área	Campo de texto
Cuota Inicial de Espacio	Double
Tecnología	Campo de texto
Nombre de la Base de datos	Campo de texto
Tecnología de la Base de Datos	Campo de texto
Actualizador Nombre y Apellidos	Campo de texto

Tabla 6 Campos definidos la Solicitud de Publicación de Sitios Web.

Tipo de Contenido Solicitud de Apelación

Campo	Tipo de dato
Asunto de la solicitud	Campo de texto
Solicitud denegada	Referencia a nodo
Motivos de la Apelación	Campo de texto

Tabla 7 Campos definidos para la Solicitud de Apelación.

Tipo de Contenido Servicio

Campo	Tipo de dato
Nombre del Servicio	Campo de texto
Breve descripción	Campo de texto

Tabla 8 Campos definidos para los Servicios.

- **CCK Validation:** Contiene funcionalidades que permiten incluir en el sistema, páginas de validación de los campos de los tipos de contenido, insertándolas como campos adicionales de cada tipo de contenido.
- **Content:** Permite la creación de nuevos tipos de contenido. Este módulo pertenece al núcleo del CMS sobre él se construyeron los tipos de contenido, servicios, solicitudes de servicios y solicitudes de publicación de aplicaciones web.
- **Content templates:** Posibilita la modificación de la plantilla de un tipo de contenido. Permite la modificación de la interfaz donde se muestra el contenido además de asociarle al mismo, funciones específicas que personalizan el tipo de contenido. Utilizando las funcionalidades de este módulo se hizo posible la creación del formato de las solicitudes, lo que originó que se crearan documentos similares a los existentes en formato duro.

- **Node reference:** Define un tipo de campo para la referencia de un nodo a otro, se hizo necesaria la activación de este módulo en la creación de los campos Servicio a Solicitar y Solicitudes Denegadas que requerían vinculación directa con los tipos de contenido Servicios y Solicitudes de Servicio y publicación de Sitios Web.

Drupal permite la construcción y configuración de bloques en el sistema, para la solución se definió el bloque Solicitudes en el sistema que muestra la cantidad de solicitudes realizadas, agrupadas por estado, esta información es accesible a los proveedores de servicios y a los usuarios determinando para éstos últimos el acceso a la visibilidad solo de las solicitudes creadas por ellos mismos.

Es importante señalar que este CMS contiene una única página servidora. Las páginas que se van creando en Drupal pueden tener o no formularios, en dependencia de lo que se desee realizar.

3.2.1. Diagrama de clases del diseño Web del sistema.

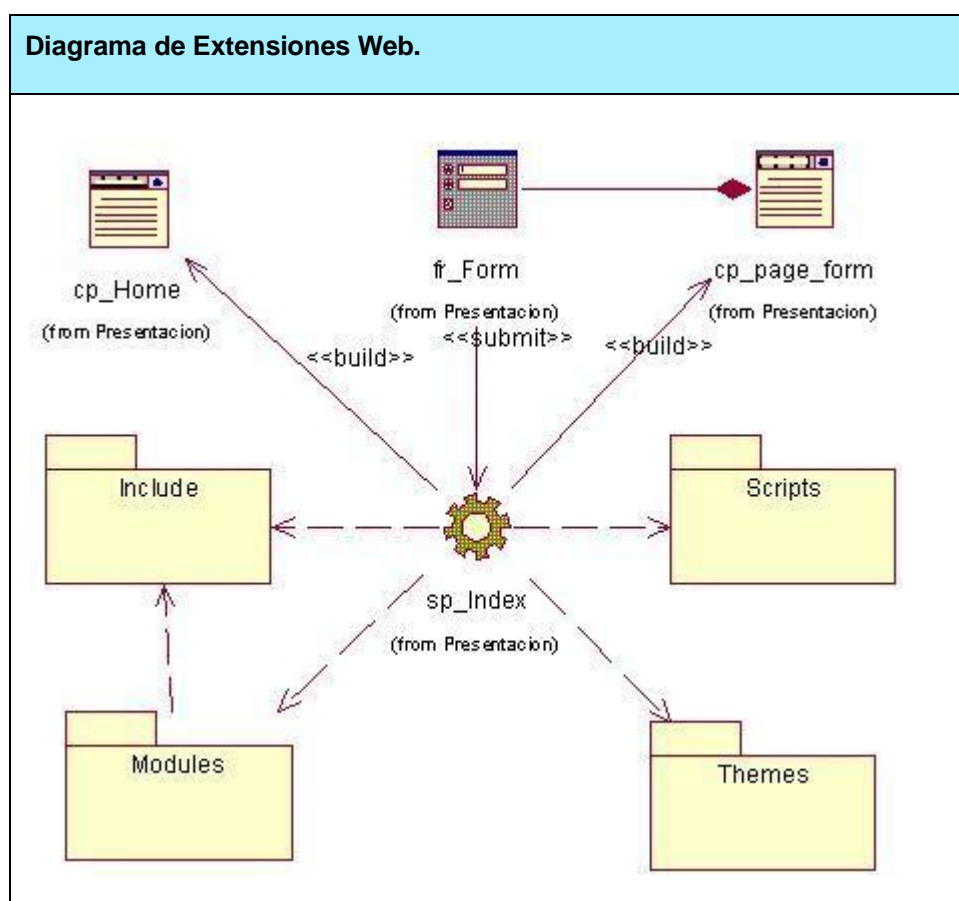


Figura 14 Diagrama de Extensiones Web

Para un mejor entendimiento del diagrama de clases de diseño, se decidió explicar por paquetes los elementos que componen al CMS Drupal.

Themes: Incluye las distintas plantillas que sirven de elemento base para crear el diseño de la aplicación.

Includes: Contiene un conjunto de ficheros indispensables para el funcionamiento, como es el caso de Database: que provee las funcionalidades de acceso a la base de datos de Drupal.

Modules: Incluye todos los módulos que permiten crear las funcionalidades del sistema a construir sobre el CMS. Estos módulos pueden ser modificados para adecuarlos a las características específicas de un sistema, además el CMS permite la inserción de nuevos módulos. Estos módulos pueden permanecer activos o inactivos posibilidad que brinda la sección de Administración de Drupal.

Scripts: Contiene un conjunto de ficheros indispensables para el funcionamiento de la aplicación, fundamentalmente orientados al aspecto visual, CSS y java script.

3.3. Diseño de la Base de datos

El diseño de la base de datos es de vital importancia para la aplicación, teniendo en cuenta que los sistemas gestores de bases de datos brindan la posibilidad de almacenar la información, recuperarla, actualizarla y eliminarla según la acción que necesite ejecutar el usuario.

Si se parte del análisis que la aplicación está desarrollada en el Sistema de Administración de Contenidos Drupal cabe destacar que la mayoría de las entidades son creadas y manejadas por el propio CMS, por lo que entonces no se tomaron en cuenta dichas entidades para la confección del modelo de datos.

Se diseña la Base de Datos teniendo en cuenta el diagrama de clases persistentes y el modelo físico de datos.

3.3.1. Diagrama de Clases Persistentes.

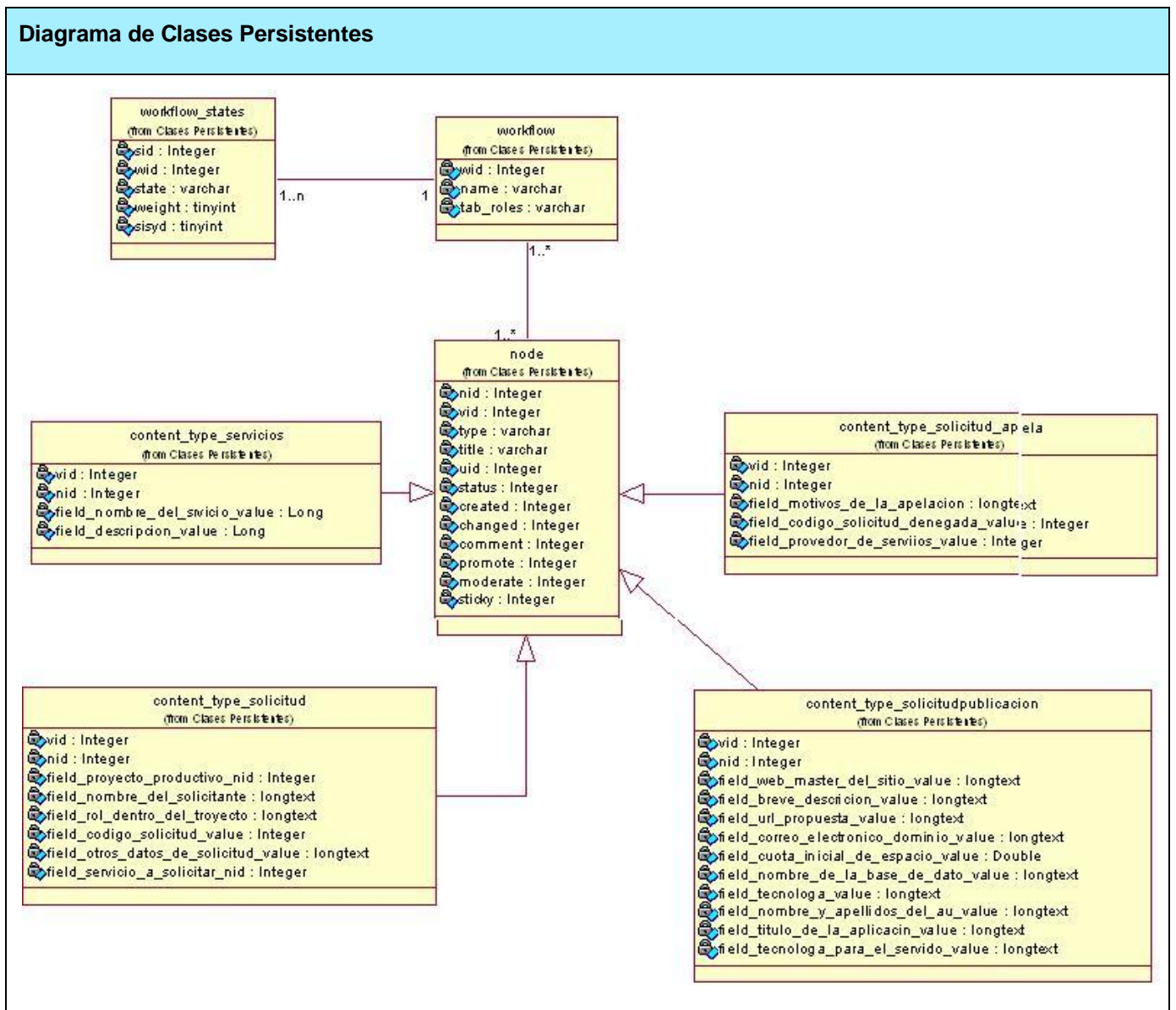


Figura 15 Diagrama de Clases Persistentes.

3.3.2. Modelo físico de datos.

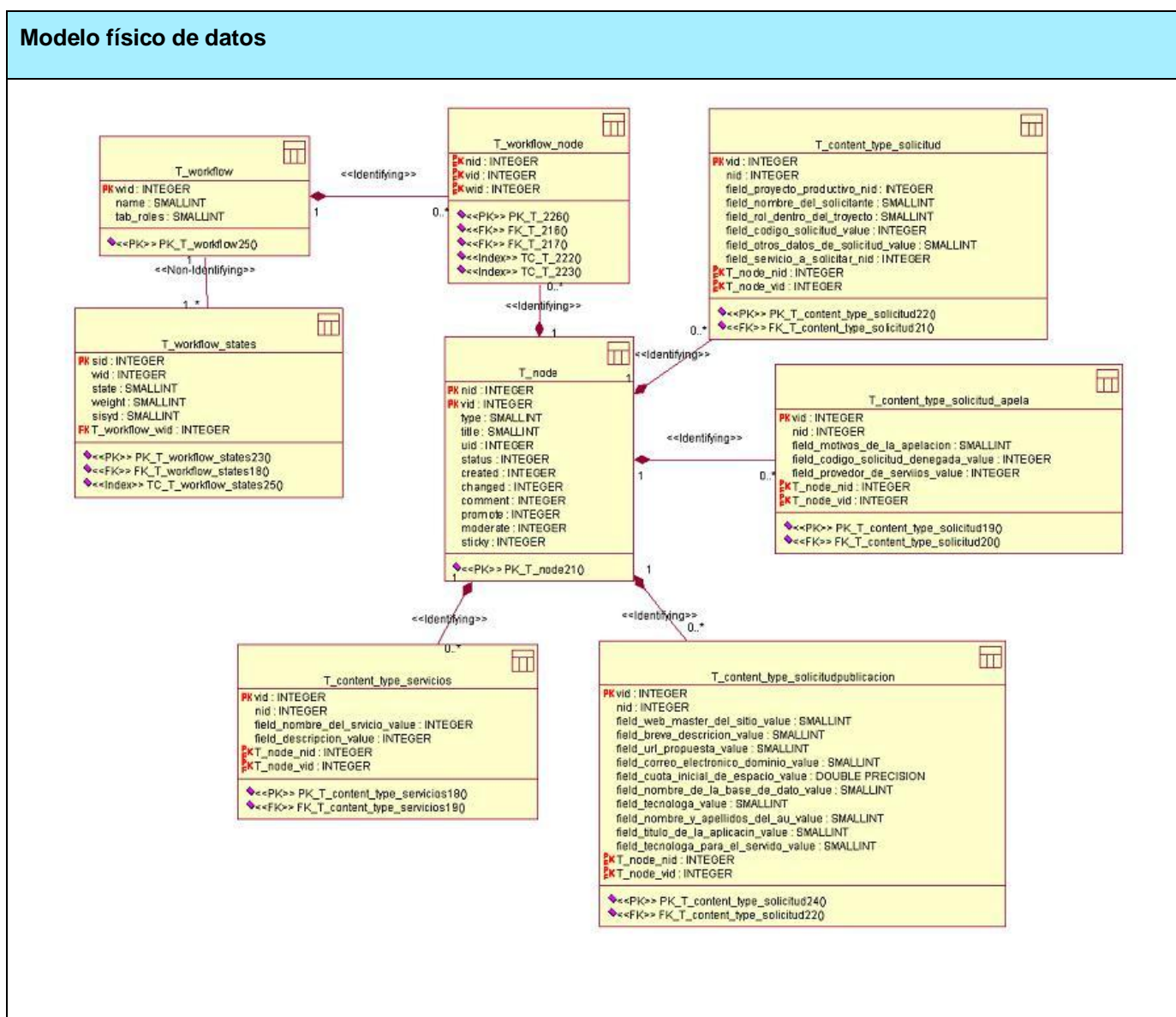


Figura 16 Modelo físico de datos.

Estándares en la interfaz de la aplicación.

Se utilizan entre los colores tonalidades de azul, gris y blanco, para resaltar algunos bloques. Se utilizó en el diseño el tema Crystal 4x de Drupal, se configuraron algunos elementos de este tema como el logo y el banner para lograr el diseño personalizado de la aplicación.

Con vista a lograr un diseño consistente de la interfaz de la aplicación, en todas las páginas se respeta el esquema Cabecera-Navegador-Contenido.

3.4. Principios de protección y seguridad.

Se definen tres roles Solicitante, Proveedor de Servicio y Administrador del Sistema para los cuales se delimitan niveles de acceso a la información. Se mencionan las funcionalidades permitidas para cada rol.

Rol	Funcionalidades permitidas
Solicitante	Creación de Solicitud de Servicio. Creación de Solicitud de Apelación. Creación de solicitud de Publicación de Sitios Web. Acceso al Historial de Solicitudes. Acceso a la lista de servicios disponibles. Acceso al bloque Cantidad de Solicitudes en el Sistema.
Proveedor de Servicio	Acceso al bloque Cantidad de Solicitudes en el Sistema. Administración del flujo de trabajo Solicitudes. Inserción de Servicios. Modificación de Servicios. Eliminación de Servicios.
Administrador del sistema	Administración del sistema.

Tabla 9 Control de acceso del sistema.

3.5. Tratamiento de errores

Para la prevención de errores provocados por entradas incorrectas de datos, una de las técnicas empleadas es conducir a los usuarios solamente por los caminos que concuerden con las funcionalidades del sistema que le son permitidas, inhabilitando el resto de los controles de la interfaz que no necesiten ser utilizados en cada una de las operaciones.

Se capturan los errores y se lanzan excepciones en forma de mensajes de texto en la misma página donde se ejecutó la acción, de forma que el usuario permanezca informado y pueda continuar su trabajo, además, se usan mensajes de confirmación, para acciones que son irreversibles como es el

caso de las eliminaciones. El sistema cuenta con un registro detallado de los errores permitiendo realizar posteriores estudios estadísticos sobre la estabilidad del sistema.

3.6. Conclusiones

En este capítulo se describió el flujo de trabajo de análisis y diseño. Se mostraron todos los diagramas de clases tanto para el análisis como para el diseño, el diagrama de clases persistentes y el modelo de datos de la aplicación. Además se explican los mecanismos a utilizar para el tratamiento de errores y la seguridad del sistema.

Capítulo IV Implementación.

En el presente capítulo se desarrolla el flujo de trabajo de implementación. Se presenta el Modelo de Implementación mostrando las dependencias entre las partes del código del sistema a través del Diagrama de Componentes. Se expone el Diagrama de Despliegue el cuál se utiliza para modelar la vista estática, en la que se muestra la situación física de los componentes lógicos desarrollados.

4.1. Diagrama de Despliegue.

El diagrama de despliegue permite apreciar de forma visual como se encuentran relacionados físicamente los componentes Hardware y Software de la aplicación, es decir, la configuración de los elementos de procesamiento en tiempo de ejecución y los componentes software.

En el diagrama de despliegue se sitúa el software en el hardware que lo contiene. El sistema se encuentra hospedado en un servidor Web con sistema operativo Linux, se emplea un servidor de base de datos independiente al servidor de aplicación estableciéndose HTTP como protocolo de comunicación entre el Cliente y el Servidor Web y el TCP/IP entre el servidor de aplicación y la base de datos. La comunicación con el servidor de correos se hace efectiva mediante el protocolo SMTP.

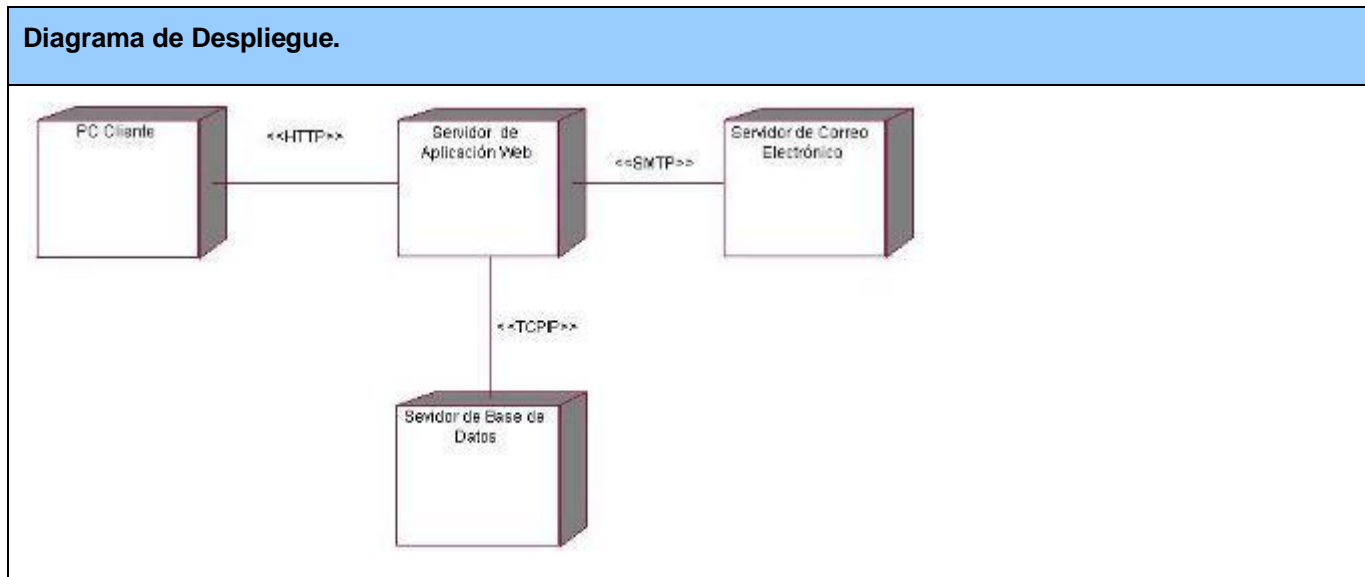


Figura 17 Diagrama de Despliegue.

4.2. Diagrama de componentes.

El diagrama de componentes muestra las organizaciones lógicas entre los componentes software, sean estos componentes de código fuente, binarios o ejecutables. En su confección se tiene en cuenta

los requisitos relacionados con la facilidad de desarrollo, la gestión de software, la reutilización, y las restricciones impuestas por los lenguajes de programación y las herramientas utilizadas en el desarrollo. Los elementos modelados dentro del diagrama se dividen en paquetes y componentes. (24)

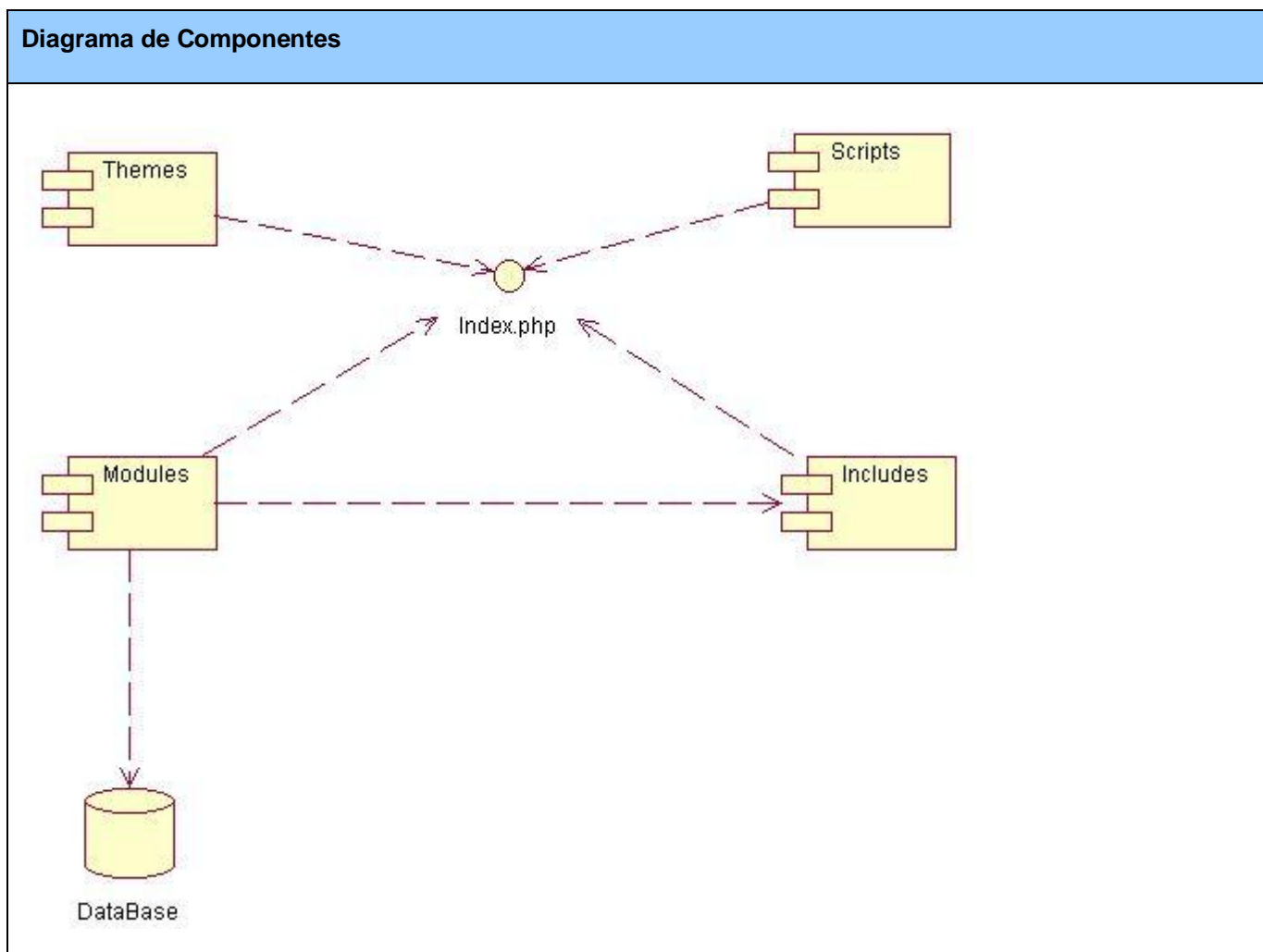


Figura 18 Diagrama de Componentes.

A continuación se explican los componentes expuestos anteriormente en el diagrama de componentes:

Carpetas	Descripción
Includes	Contiene un conjunto de ficheros indispensables para el funcionamiento esencialmente ficheros de instalación y configuración inicial del CMS.

Modules	Incluye todos los módulos, que contienen las distintas funcionalidades del CMS. Permite la inserción de nuevos módulos provistos por la comunidad de usuarios.
Themes	Incluye las distintas plantillas provistas por el CMS, en esta carpeta se incluyen las plantillas creadas por los usuarios.
DataBase	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, constituye un conjunto de funcionalidades que inician todos los elementos de Drupal.

Tabla 10 Diagrama de Componentes.

4.3. Conclusiones.

En este capítulo se abordó el flujo de trabajo de implementación mostrándose la realización de los diagramas que lo componen, el diagrama de despliegue donde se especificó la distribución de los sistemas de cómputo que conforman la aplicación con sus características, las conexiones y protocolos que los acoplan y el diagrama de componentes de la aplicación web con sus distintos elementos.

Capítulo V Estudio de factibilidad.

El estudio de factibilidad es un paso importante en la realización de un proyecto permite obtener las estimaciones de esfuerzo, tiempo de desarrollo en meses, costo del producto y la cantidad de personas que se necesitan para desarrollar el proyecto. En este capítulo se describen la estimación de costos del sistema propuesto basado en la técnica de Análisis de Puntos de Casos de Uso.

Modelos De Estimación.

Existen diferentes modelos de estimación como son:

Los Empíricos donde los datos que soportan la mayoría de los modelos de estimación obtienen una muestra limitada de proyectos.

El Modelo COCOMO modelo constructivo de costos que está constituido por los siguientes modelos de manera jerárquica:

Modelo I. El Modelo COCOMO básico calcula el esfuerzo y el costo del desarrollo de Software en función del tamaño del programa, expresado en las líneas estimadas.

Modelo II. El Modelo COCOMO intermedio calcula el esfuerzo del desarrollo de software en función del tamaño del programa y de un conjunto de conductores de costos que incluyen la evaluación subjetiva del producto, del hardware, del personal y de los atributos del proyecto.

Modelo III. El modelo COCOMO avanzado incorpora todas las características de la versión intermedia y lleva a cabo una evaluación del impacto de los conductores de costos en cada caso (análisis, diseño, etc.) del proceso de ingeniería de Software.

Herramientas Automáticas De Estimación que permiten al planificador estimar costos y esfuerzos. Aunque existen muchas herramientas automáticas de estimación, todas exhiben las mismas características generales y todas requieren de una o más clases de datos. (25)

5.1. Planificación basada en Casos de Uso. Análisis de Puntos de Casos de Uso.

5.1.1. Cálculo de los Puntos de Casos de Uso Desajustados.

$$UUCP=UAW+UUCW$$

Donde:

UUCP: Puntos de casos de Uso sin ajustar.

UAW: Factor de peso de los actores sin ajustar.

UUCW: Factor de peso de los Casos de Uso sin ajustar.

5.1.2. Factor de peso de los actores sin ajustar (UAW).

Tipo de Actor	Descripción	Factor de Peso	Actores	Total
Simple	Sistema con sistema a través de interfaz de programación.	1	0	0
Medio	Sistema con sistema mediante protocolo de interfaz basado texto.	2	0	0
Complejo	Persona que interactúa el sistema mediante interfaz gráfica.	3	3	9

Tabla 11 Factor de peso de los actores sin ajustar.

$$UAW = \sum \text{cant actores} * \text{peso}$$

$$UAW = 3 * 3 = 9$$

5.1.3. Factor de peso de los Casos de Uso sin ajustar (UUCW).

Tipo de CU	Descripción	Factor de Peso	Cantidad de CU	Total
Simple	El Caso de Uso tiene de 1 a 3 transacciones.	5	7	35
Medio	El Caso de Uso tiene de 4 a 7 transacciones.	10	2	20
Complejo	El Caso de Uso tiene más de 8 transacciones.	15	0	0

Tabla 12 Factor de peso de los Casos de Uso sin ajustar.

$$UUCW = \sum cantCU * Peso$$

$$UUCW = 55$$

$$UUCP = UAW + UUCW$$

$$UUCP = 9 + 55 = 64$$

5.1.4. Cálculo de los Puntos de Casos de Uso ajustados.

$$UCP = UUCP * TCF * EF$$

Donde:

UCP: Puntos de Casos de Uso ajustados.

UUCP: Puntos de Casos de Uso sin ajustar.

TCF: Factor de complejidad técnica.

EF: Factor de ambiente.

El factor de complejidad técnica (TCF) se calcula mediante la cuantificación de un conjunto de factores que determinan la complejidad técnica del sistema. Cada factor se cuantifica en un valor desde 0 (aporte irrelevante) hasta 5 (aporte muy relevante).

Significado de los valores:

0: No presente o sin influencia.

1: Influencia incidental o presencia incidental.

2: Influencia moderada o presencia moderada.

3: Influencia media o presencia media.

4: Influencia significativa o presencia significativa.

5: Fuerte influencia o fuerte presencia.

Factor	Descripción	Peso	Valor asignado	Comentario
T1	Sistema distribuido.	2	0	El sistema es centralizado.
T2	Tiempo de respuesta.	1	1	La velocidad de respuesta es de apenas segundos de tiempo.
T3	Eficiencia del usuario final.	1	1	Escasas restricciones de eficiencia.
T4	Procesamiento interno complejo.	1	1	No hay cálculos complejos.
T5	El código debe ser reutilizable.	1	5	Se requiere que el código sea reutilizable.
T6	Facilidad de instalación.	0.5	4	Existen requerimientos de facilidad de instalación.
T7	Facilidad de uso.	0.5	3	Normal
T8	Portabilidad.	2	4	Se requiere que el sistema sea portable.
T9	Facilidad de cambio.	1	3	Se requiere un costo moderado de mantenimiento

T10	Concurrencia.	1	0	No hay concurrencia
T11	Incluye objetivos especiales de seguridad.	1	5	Se requiere seguridad.
T12	Provee acceso directo a terceras partes.	1	1	Los usuarios no tienen accesos directos.
T13	Se requieren facilidades especiales de entrenamiento a usuarios.	1	1	Sistema fácil de usar por cualquier usuario.

Tabla 13 Complejidad Técnica.

$$TCF = 0.6 + 0.01 * \Sigma (\text{peso} * \text{valor asignado})$$

$$TCF = 0.6 + 0.01 * 29.5$$

$$TCF = 0.895$$

El factor de ambiente (EF) está relacionado con las habilidades y entrenamiento del grupo de desarrollo que realiza el sistema. Cada factor se cuantifica con un valor desde 0 (aporte irrelevante) hasta 5 (aporte muy relevante).

Factor	Descripción	Peso	Valor asignado	Comentario
E1	Familiaridad con el modelo de proyecto utilizado.	1.5	4	El grupo está bien familiarizado con el modelo.
E2	Experiencia en la aplicación.	0.5	2	Los desarrolladores no habían trabajado con en este tipo de aplicación.
E3	Experiencia en orientación a objetos.	1	4	El grupo programa orientado a objetos.
E4	Capacidad del analista líder.	0.5	2	No existe un especialista, aunque se tienen

				conocimientos bastos.
E5	Motivación.	1	5	El grupo está altamente motivado
E6	Estabilidad de los requerimientos.	2	3	Pueden surgir cambios después de la integración con el sistema central de soporte.
E7	Personal part-time.	-1	3	Parte del grupo es part-time.
E8	Dificultad del lenguaje de programación.	-1	2	Se usará el lenguaje PHP.

Tabla 14 factor Ambiente.

$$EF = 1.4 - 0.03 * \Sigma (\text{peso} * \text{valor asignado})$$

$$EF = 1.4 - 0.03 * 18$$

$$EF = 0.86$$

$$UCP = UUCP * TCF * EF$$

$$UCP = 64 * 0.895 * 0.86$$

$$UCP = 49.26$$

5.2. Estimación de esfuerzo a través de los puntos de Casos de Uso.

$$E = UCP * CF$$

Donde:

E: Esfuerzo estimado en horas hombres.

UCP: Punto de Casos de Usos ajustados.

CF: Factor de conversión.

Para obtener el factor de conversión (CF) se cuentan cuantos valores de los que afectan el factor ambiente (E1...E6) están por debajo de la media (3), y los que están por arriba de la media para los restantes (E7, E8). Si el total es 2 o menos se utiliza el factor de conversión 20 Horas-Hombres / Punto de Casos de Uso. Si el total es 3 ó 4 se utiliza el factor de conversión 28 Horas-Hombres/ Punto de

Casos de Uso. Si el total es mayor o igual que 5 se recomienda efectuar cambios en el proyecto ya que se considera que el riesgo de fracaso del mismo es demasiado alto.

En este caso se puede decir que:

$$CF = 20 \text{ Horas-Hombres} / \text{Punto de Casos de Uso}$$

$$E = UCP * CF$$

$$E = 49.26 * 20$$

$$E = 985.216 \text{ Horas-Hombre}$$

5.2.1. Calcular el esfuerzo de todo el proyecto.

Actividad	Porcentaje	Horas-Hombres
Análisis	20.00%	492,608
Diseño	10.00%	246,304
Implementación	40.00%	985,216
Pruebas	10.00%	246,304
Sobrecarga (otras actividades)	20.00%	492,608
Total	100.00%	2463,04

Tabla 15 Porcentaje de esfuerzo por flujo de trabajo.

$$ET = 10.26 \text{ mes-hombre.}$$

$$ET = 2463.04 \text{ horas-hombre.}$$

Tiempo total del Proyecto.

$$\text{Tiempo} = ET / CH$$

$$\text{Tiempo} = 10.26 \text{ meses} / 2 \text{ hombres}$$

$$\text{Tiempo} = 5.1 \text{ meses}$$

Costo del Proyecto.

Se asume como salario promedio mensual \$ 100.00.

CHM= 2 * Salario promedio

CHM= 200.00 \$/mes

Costo= Salario Promedio * ET

Costo= 200.00 * 10.26

Costo= \$ 2052.

5.3. Beneficios tangibles e intangibles

El Sistema para la Gestión de las Solicitudes de Servicios de Soporte no es un producto pensado para fines comerciales, su principal objetivo es resolver los problemas existentes en el proceso de gestión de solicitudes de dichos servicios en la UCI. El beneficio fundamental del sistema es contar con una aplicación Web flexible, dinámica y de interfaz agradable que permita registrar, actualizar y conocer de una forma más precisa, segura y en el menor tiempo posible la información referente a las solicitudes. Por tanto, los beneficios inmediatos son fundamentalmente intangibles:

- Disminución del tiempo y esfuerzo que se invierte en esta tarea que se realiza, hasta el momento, de forma manual.
- Fácil detección de errores en el proceso de solicitud de servicios.
- Rápido acceso y publicación de la información actualizada de las solicitudes.
- Fácil procesamiento de la información y obtención, dinámica, de reportes referentes al comportamiento de las solicitudes en cualquier momento.

Como beneficio tangible asociado al desarrollo de la aplicación se señala la Reducción de los gastos en materiales de oficina destinados a la confección de solicitudes.

5.4. Análisis de costos y beneficios

Desarrollar un producto informático resulta costoso, justificar entonces su realización depende de los beneficios que reportarían su implantación y utilización.

Como ya se ha mencionado en el epígrafe anterior, el software que se propone aportaría beneficios inmediatos intangibles de suma importancia, pues una vez implantado, contribuirá a aumentar la eficiencia en la gestión de las solicitudes de servicios referentes al soporte tanto a las aplicaciones en

ejecución como a los proyectos en desarrollo dentro del centro, al proporcionar comodidad, rapidez y confiabilidad a la hora del llenado, búsqueda y verificación de las solicitudes. Se obtiene un mayor control por parte de los proveedores, así como un análisis certero de los comportamientos de dichas solicitudes.

La tecnología utilizada para el desarrollo del sistema es libre, por tanto no hay que incurrir en gastos en el pago de licencias de uso. El sistema es portable por lo que un cambio de plataforma para la implantación del mismo es viable y factible. Analizando el costo del proyecto, y los beneficios que reporta, se puede concluir que su implementación es factible.

5.5. Conclusiones

En este capítulo se describió el estudio de factibilidad correspondiente a la solución propuesta, donde se puntualizan los costos estimados, los recursos humanos implicados, el tiempo de desarrollo y los beneficios que alcanzará al ser implantado. La aplicación reportará beneficios significativos e importantes para la gestión de las solicitudes en la UCI, lo que indica que es factible la implementación de la herramienta propuesta.

Conclusiones.

Con este trabajo se presenta un sistema que permite gestionar toda la información referente a los servicios y solicitudes del grupo de soporte de la Dirección de Informatización, gracias a esto se logrará una mayor eficiencia en la prestación de los mismos.

Se obtuvo el Sistema de Entorno Integrado de Desarrollo de Informatización que permite a sus usuarios:

- Solicitar Servicios de Soporte a través de la Web.
- Determinar el estado de las solicitudes.
- Gestionar Servicios de Soporte
- Generar reportes atendiendo a criterios definidos por el proveedor de servicios.

Se llega a la conclusión que el sistema propuesto soluciona los problemas que originaron la investigación permitiendo el cumplimiento del objetivo planteado.

La aplicación aporta los siguientes beneficios:

- Automatización del proceso de gestión de solicitudes.
- Se provee de una plataforma central para la publicación de servicios de soporte.

Recomendaciones.

Se recomienda:

- Continuar el desarrollo del módulo con la integración del mismo al Sistema de Soporte Central.
- Proveer al sistema de funcionalidades específicas que permitan su integración al Sistema de Autenticación central de la UCI.
- Integrar al Sistema de Gestión de proyectos de la Infraestructura Productiva.
- Establecer un diseño según las pautas que propone la Dirección de Comunicación Visual.

Referencias Bibliográficas.

1. Escalona, D. C. Arquitectura Cliente/Servidor. *DIPLOMADO/DIPLO-C-SERVER*. [Online] 2004. [Cited: noviembre 15, 2007.] www.fet2005.cs.buap.mx.
2. SICOP Soluciones. [Online] SICOP Soluciones, 2007. [Cited: diciembre 12, 2007.] www.sicops.com.
3. Raventós, Josep M^a. Sage logic control. [Online] Sage Logic Control, S.A. 2008, 2008. [Cited: enero 17, 2008.] www.sagelogiccontrol.com.
4. Business Objects an SAP Company. [Online] Business Objects SA, 2008. [Cited: enero 23, 2008.] support.businessobjects.com.
5. Álvarez, Báez. Babel. [Online] [Cited: febrero 4, 2008.] www.congreso-info.cu/UserFiles/File/Info/Info2006/Ponencias/37.pdf.
6. Alemán Betancourt, Iris. Biblioteca. *Sistema Automatizado para la Gestión de las Solicitudes de Servicios Telemáticos*. [Online] 2004. www.biblodoc.uci.cu.
7. Corai, Ricardo. Sistema de Administración de Contenidos . [Online] 2006. www.cmsenespanol.com/sistemas-de-administracion-de-contenidos.
8. Garcia, X. C. and Alfonso, J. M. Sistemas de Gestión de Contenidos mosaic.uoc.edu/articulos/cms1204.html. *Introducción a los Sistemas de Gestión de Contenidos*. [Online] 2004. [Cited: febrero 6, 2008.] mosaic.uoc.edu/articulos/cms1204.html.
9. Pérez Valdés, Damián. Maestros del Web. *Una serie de administradores de contenidos*. [Online] 2007. [Cited: enero 22, 2008.] www.maestrosdelweb.com/editorial/una-serie-de-administradores-de-contenidos.
10. Dani. 5 Líneas. *Curva de aprendizaje de Drupal*. [Online] marzo 28, 2007. [Cited: enero 25, 2008.]
11. Molpeceres, A. Procesos de Desarrollo RUP, XP y FDD. [Online] 2002. [Cited: enero 25, 2008.] www.javahispano.org/download/articulos/metodos_desarrollo.pdf.
12. Salinas, P. *Tutorial de UML Volume, DOI*. 2003.
13. Boch, G. and Rumbaugh, J. *El lenguaje unificado de modelado. Manual de referencia Volume, DOI*: . 2000.
14. La Fuente, Guillermo. GIDIS. *UML*. [Online] 2000. gidis.ing.unlpam.edu.ar/personas/glafuente/uml/uml.html.

15. Categorías de Software Libre y no Libre . [Online] GNU, 2006. [Cited: enero 15, 2008.] from <http://www.gnu.org/home.es.html>.
16. Dondo, Agustín. PHP en castellano. *¿Por qué elegir PHP?* [Online] Enero 17 Enero, 2007. www.programacion.com/php/articulo/porquephp/.
17. Bakken, S. *Manual de PHP. PHP Documentation Group*. 2005.
18. Charre Ojeda, Francisco. PHP5. [Online] junio 2004. www.agapea.com/PHP-5-n154892i.htm.
19. Desarrollo Web. *Tutorial de SQL*. [Online] 2004. [Cited: enero 30, 2008.] [//www.desarrolloweb.com/manuales/9/](http://www.desarrolloweb.com/manuales/9/).
20. Ramirez, Y. S. *Propuesta para un sistema de Catalogación y Recuperación de Recursos de Información*. Ciudad de la Habana, . : Instituto Superior Politécnico “José Antonio Echeverría, 2004.
21. Fedora. *MySQL vs. PostgreSQL. Volume, DOI:.* (2006).
22. Nick, Key [//www.agapea.com/Desarrollo-de-modulos-y-aplicaciones-con-Apache-n1244406i.htm/](http://www.agapea.com/Desarrollo-de-modulos-y-aplicaciones-con-Apache-n1244406i.htm/)>. Desarrollo de módulos y aplicaciones con Apache . [Online] [Cited: febrero 1, 2008.] www.agapea.com/Desarrollo-de-modulos-y-aplicaciones-con-Apache.
23. Angel Alvarez, Miguel. Desarrollo Web.com. *Dreamweaver*. [Online] Desarrollo Web.com. [Cited: diciembre 13, 2007.]
24. Fernández Vila, Ana. Diagramas de Implementación. [Online] 2001. www-gris.det.uvigo.es/~avila/UML/node48.html.
25. Mario, Peralta. ITBA. [Online] 2000. <http://www.itba.edu.ar/capis/webcapis/planma-esp.html>.
26. Pecos, Daniel. Net Pecos. [Online] [Cited: enero 19, 2008.] www.netpecos.org/docs/mysql_postgres/index.html.
27. Moreno Martínez, Gerardo. Monografias.com. [Online] Monografias.com S.A. , 1997. [Cited: febrero 24, 2008.] <http://www.monografias.com/trabajos5/insof/insof.shtml>.
28. JD. BananaTools. *Características generales de Drupal*. [Online] enero 2, 2008. [Cited: enero 10, 2008.] www.banangatools.com/drupal/conceptos-basicos/18/caracteristicas-generales-de-drupal.

Bibliografía.

- DBS. (1992). "DBS Servicios informáticos."
http://www.dbs.es/opencms/servicios/contenidos/gestion_de_contenidos.html.
- DesarrolloWeb. (2004). "Desarrollo Web: Manual "Tutorial de SQL"."
<http://www.desarrolloweb.com/manuales/9/>.
- Foundation, W. (2007) Adobe Dreamweaver. Volume, DOI:
- Guervos, J. J. M. (2005).
<http://geneura.ugr.es/~jmerelo/tutoriales/cms/>.
- INEI, I. N. d. E. e. I. (2004). Arquitectura Cliente/Servidor.
- Infórmate, C. I. "Glosario de términos bibliotecológicos y de Ciencias de la Información."
http://www.uh.cu/facultades/fcom/portal/interes_glosa_terminos.htm
- Molpeceres, A. (2005). "Procesos de desarrollo: RUP, XP y FDD."
<http://www.javahispano.org/articles.article.action?id=76>.
- Perdure, T. (2005) MySQL and PostgreSQL Compared. Volume, DOI:
- Phpnuke. "phpnuke.org."
<http://www.phpnuke.org>.
- Sitepoint (2004) The J2EE guy still doesn't get PHP Volumen, DOI:
- SuperAlumnos.net (2006) Drupal me enamora! Volumen, DOI:
- Tramullas, J. (2005). "Herramientas de Software Libre para la gestión de contenidos."
<http://www.hipertext.net/web/pag258.htm>
- ANÓNIMO. CMS MATRIX.
<http://cmsmatrix.org/>
- Página web oficial de Drupal.
<http://drupal.org.es/>
- PRESSMAN., R. S. Tomado de la biblioteca UCI. Ingeniería del software. Un enfoque práctico.
<http://bibliodoc.uci.cu/pdf/reg02689.pdf>

Glosario de Términos.

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.

Unix: Es un sistema operativo portable dada la portabilidad de un software se define como su dependencia de la plataforma en la que corre, es multitarea ya que permite que varios procesos sean ejecutados al mismo tiempo compartiendo uno o más procesadores y multiusuario.

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.

Actor: Alguien o algo, fuera del sistema o negocio que interactúa con el sistema o negocio.

Base de datos (BD): Conjunto de datos interrelacionados, almacenados más o menos con carácter permanente en la computadora, puede ser considerada una colección de datos variables en el tiempo.

SGBD: Es el software que permite la utilización y/o la actualización de datos almacenados en una (o varias) base(s) de datos por uno o varios usuarios desde diferentes puntos de vistas a la vez.

SQL (Structured Query Language): Conjunto estándar de comandos para gestionar base datos relacionales por su misma característica relacional.

TCP/IP (Transmission Control Protocol/Internet Protocol): Es el conjunto de protocolos que definen a Internet. Originalmente diseñado para el sistema operativo UNIX, hoy en día existe software TCP/IP disponible para la mayoría de los sistemas operativos. Para poder utilizar la Internet, su computador debe tener software TCP/IP.

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.

Artefactos: Una parte de la información que (1) es producida, modificada, o usada por un proceso, (2) define un área de responsabilidad, y (3) está sujeta al control de versión. Un artefacto puede ser un

modelo, un elemento del modelo, o un documento. Un documento puede adjuntar otros documentos. Una parte de la información que es usada o producida por un proceso de desarrollo del software. Un artefacto puede ser un modelo, una descripción, o un software.

Apache: El servidor HTTP Apache es un software (libre) servidor HTTP de código abierto para plataformas Unix, Windows y otras, que implementa el protocolo HTTP/1.1.

Blogger: un servicio de creación y alojamiento de blogs propiedad de Google.

Case: (*Computer-Aided Software Engineering*) es el uso de un método asistido por computadora para organizar y controlar el desarrollo de un software.

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

Gestión: Conjunto de trámites que se llevan a cabo para resolver un asunto.

Gestión de información: Comprende las actividades relacionadas con la obtención de la información adecuada, a un precio adecuado, en el tiempo y lugar adecuado, para tomar la decisión adecuada.

Gestión de Solicitudes: Aplicaciones informáticas que tienen como función principal permitir el acceso de los usuarios a determinados servicios.

GNU/GPL: Es una licencia creada por la Free Software Foundation a mediados de los 80, y está orientada principalmente a proteger la libre distribución, modificación y uso de software. Su propósito es declarar que el software cubierto por esta licencia es software libre y protegerlo de intentos de apropiación que restrinjan esas libertades a los usuarios.

HTML: el Lenguaje de Marcado de HiperTextos (*HTML - HyperText Markup Language*) es un lenguaje que sirve para especificar la estructura de documentos que se visualizan a través de Internet usando programas de despliegue.

HTTP: HTTP o *HiperText Transfer Protocol* (protocolo de transferencia de hipertexto) Es el grupo de reglas, o protocolos, que gobiernan la transferencia de hipertexto entre dos o más computadoras. Es muy cómodo y fácil de usar para transferir texto, imágenes, sonido, etc.

IMAP: (acrónimo inglés de **I**nternet **M**essage **A**ccess **P**rotocol) es un protocolo de red de acceso a mensajes electrónicos almacenados en un servidor. Mediante IMAP se puede tener acceso al correo electrónico desde cualquier equipo que tenga una conexión a Internet.

Internet: Conjunto de redes de computadoras y equipos unidos físicamente mediante cables que conectan puntos de todo el mundo. Estos cables pueden ser: de redes locales, telefónicas, convencionales, digitales o canales de fibra óptica. En algunos tramos, los datos se transmiten vía satélite o a través de servicios como la telefonía celular. Forman parte de Internet computadoras personales, bases de datos, servicios, fax y números telefónicos.

LDAP: Protocolo a nivel de aplicación que permite el acceso a un servicio de directorio ordenado y distribuido para buscar diversa información en un entorno de red. LDAP puede considerarse una base de datos al que pueden realizarse consultas

Linux: Denominación de un sistema operativo tipo-Unix y el nombre de un núcleo. Es uno de los paradigmas más prominentes del software libre y del desarrollo del código abierto, cuyo código fuente está disponible públicamente.

Microsoft: Compañía de software más grande del mundo. Fue fundada en 1975 por Paul Allen y Bill Gates. Aunque también se conoce por sus lenguajes de programación y aplicaciones para computadores personales, el éxito sobresaliente de Microsoft se debe a sus sistemas operativos DOS y Windows.

Microsoft Windows: (conocido simplemente como Windows) es un sistema operativo con interfaz gráfica para computadoras personales cuyo propietario es la empresa Microsoft.

MySQL AB: Compañía de software fundada en 1995, creadora del sistema administrador de bases de datos relacionales MySQL, y una de las más grandes empresas de software libre del mundo

MyISAM: Tecnología de almacenamiento de datos usada por defecto por el sistema administrador de bases de datos relacionales MySQL. Este tipo de tablas están basadas en el formato ISAM pero con nuevas extensiones.

On Line: En línea o sea que puede ser accedido por la Web

Proceso: Secuencia de actividades invocadas para producir un producto de software.

Software: Palabra en inglés utilizada para indicar a los programas de computadoras, a las aplicaciones.

Software Libre: Es el software que, una vez obtenido, puede ser usado, copiado, estudiado, modificado y redistribuido libremente.

URL: URL significa un localizador de Recursos Uniformes. De hecho los URLs son para los bits de información, el equivalente a las calles en una dirección de Internet. Los URL son uno de varios métodos para localizar, manejar e identificar bits de información en Web. La mayoría de los URL tienen tres partes principales: el protocolo, el nombre del anfitrión y el directorio. El protocolo representa la manera en que el documento será accedido; el nombre del anfitrión es el sistema informático en Internet donde se almacena la información, y el directorio representa la localización del archivo o cualquier otro tipo de información en el anfitrión.

World Wide Web: WWW (*World Wide Web*) es un sistema de información global, interactivo, dinámico, distribuido, gráfico, basado en Hipertexto, con plataforma de enlaces cruzados, que se ejecuta en Internet. Red alrededor del mundo. Es otra de las maneras de referirse a Internet.

Zope: Es un servidor de aplicaciones web escrito en el lenguaje de programación Python. Puede ser manejado casi totalmente usando una interfaz de usuario basada en páginas Web.

Módulo: Colección de funciones que se enlazan, ofreciendo funcionalidades adicionales a las predefinidas en el núcleo de Drupal. Son complementos para ampliar, construir o mejorar la funcionalidad básica del sistema gestor de contenidos.

API's: (del inglés Application Programming Interface - Interfaz de Programación de Aplicaciones) es el conjunto de funciones y procedimientos (o métodos si se refiere a programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

Anexos.

Anexo 1

Descripción Textual de los Casos de Usos del Negocio.

Casos de Uso Gestionar Servicio: Escenario Solicitar Servicio.

Caso de Uso:	Gestionar Servicio	
ACTORES:	Solicitante	
TRABAJADORES:	Proveedor de servicio	
PROPÓSITO:	Permitir que el actor pueda solicitar un servicio necesario para el desarrollo y puesta en marcha de su proyecto.	
RESUMEN:	El caso de uso se inicia cuando el actor contacta a la dirección de informatización con el objetivo de solicitar de uno o varios servicios que son necesarios para dar mantenimiento y soporte a determinado proyecto.	
CU ASOCIADOS:		
Flujo normal de los eventos:		
Acción del actor	Respuesta del proceso de negocio	
1- El actor entrega la solicitud al responsable del grupo de soporte.	1.1 El responsable del grupo de soporte de informatización recoge la solicitud. 1.2 El responsable del grupo de soporte asigna a la solicitud el estado pendiente. 1.3 El responsable del grupo de soporte archiva la solicitud para posterior estudio.	
Flujo alterno de los eventos:		
Acción del actor	Respuesta del proceso de negocio	
1-	1.4 El responsable del grupo de soporte no acepta la solicitud por estar incompleta.	
Poscondiciones	Es solicitado un servicio.	
Prioridad	Crítico	

Tabla 16 Descripción del CU del Negocio Gestionar Servicio: Escenario Solicitar Servicio

Casos de Uso Gestionar Servicio: Escenario Solicitar Estado Solicitud

Caso de Uso:	Solicitar Estado Solicitud	
ACTORES:	Solicitante	
TRABAJADORES:	Proveedor de servicio.	
PROPÓSITO:	Permitir que un solicitante pueda tener acceso a ver en que estado quedó la solicitud que realizó.	
RESUMEN:	El caso de uso se inicia cuando el actor solicita el estado final de solicitud de servicio.	
CU ASOCIADOS:	CU Solicitar Servicio.	
PRECONDICIONES:	El actor debe haber realizado alguna solicitud	
Flujo normal de los eventos:		
Acción del actor	Respuesta del proceso de negocio	
1- El actor solicita estado de su solicitud.	1.1- El responsable del grupo de soporte busca la solicitud. 1.2- Informa al solicitante el estado que se determinó para su solicitud.	
Flujo alterno de los eventos:		
Acción del actor	Respuesta del proceso de negocio	
1-	2 El responsable del grupo de soporte le informa que no tiene registrada dicha solicitud.	
Poscondiciones	El actor conoce el estado de su solicitud.	
Prioridad	Secundario.	

Tabla 17 Descripción del CU del Negocio Gestionar Servicio: Solicitar Estado Solicitud.

Casos de Uso Solicitar Apelación.

Caso de Uso:	Solicitar apelación	
ACTORES:	Solicitante	
TRABAJADORES:	Proveedor de servicio.	
PROPÓSITO:	Permitir que un solicitante solicite apelación si no está de acuerdo con el estado final de la solicitud.	
RESUMEN:	El caso de uso se inicia cuando el Solicitante contacta a la dirección de informatización para realizar una apelación debido a la denegación de su solicitud de servicio.	
CU ASOCIADOS:	Solicitar Servicio.	
PRECONDICIONES:	La solicitud no puede haber sido objeto de apelación en procesos anteriores.	
Flujo normal de los eventos:		
Acción del actor	Respuesta del proceso de negocio	
1 El solicitante recibe estado final de su solicitud.	2.1- El proveedor de servicios recoge la solicitud de apelación.	
2 El solicitante entrega la solicitud de apelación.	2.2- Busca solicitud de servicios denegada referida en apelación.	
	2.3- Cambia estado de la solicitud de servicio a apelado	
	2.4- Anexa la solicitud de apelación a la solicitud de servicio.	
Flujo alterno de los eventos:		
Acción del actor	Respuesta del proceso de negocio	
	2.3 El responsable del grupo de soporte no recoge la solicitud por tener errores en su confección.	
	2.4-La solicitud de apelación es rechazada por haber estado involucrada en otro proceso de apelación anterior.	
Poscondiciones	La solicitud de apelación al servicio solicitado es realizada.	
Prioridad	Crítico	

Tabla 18 Descripción del CU del Negocio Solicitar Apelación

Anexo 2

Diagrama de Actividades.

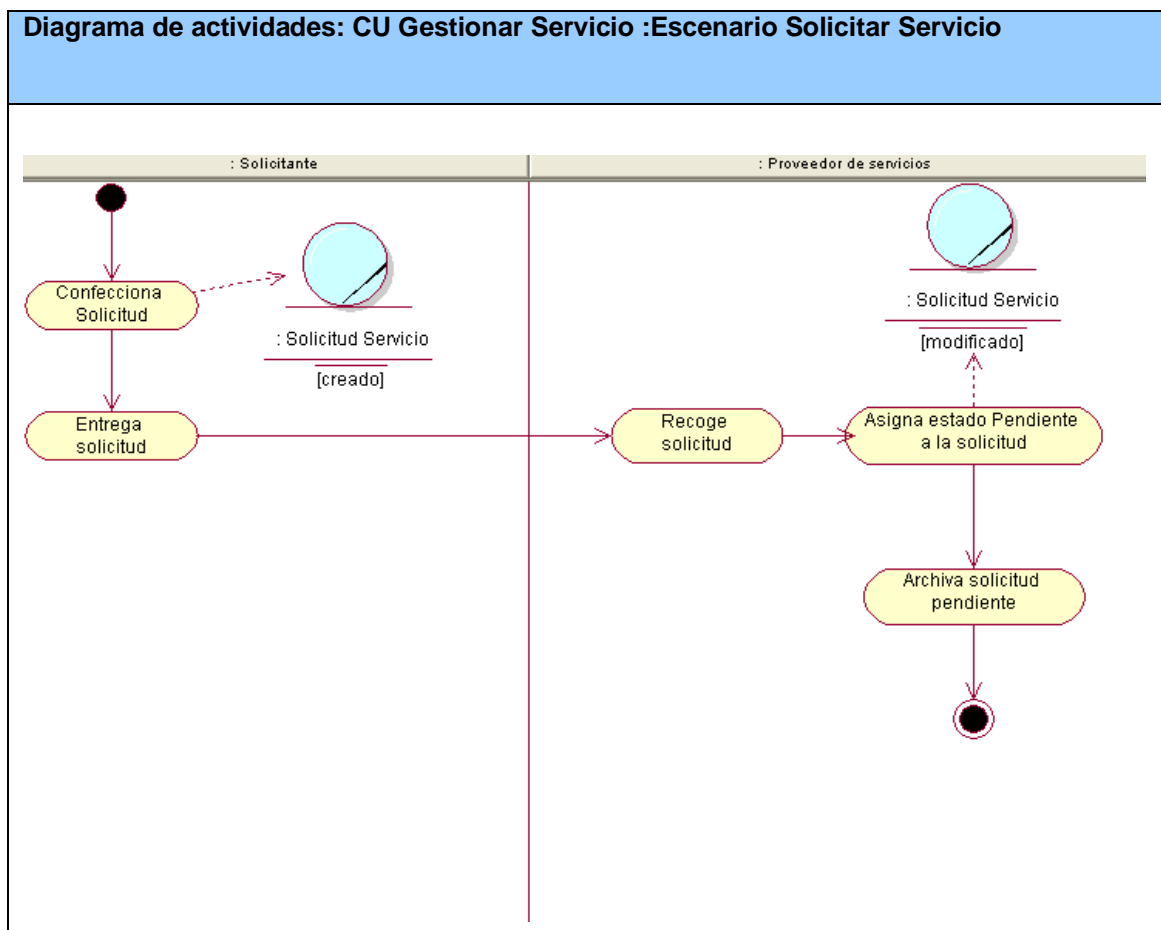


Figura 19 Diagrama de Actividades CU Gestionar Servicio: Escenario Solicitar Servicio

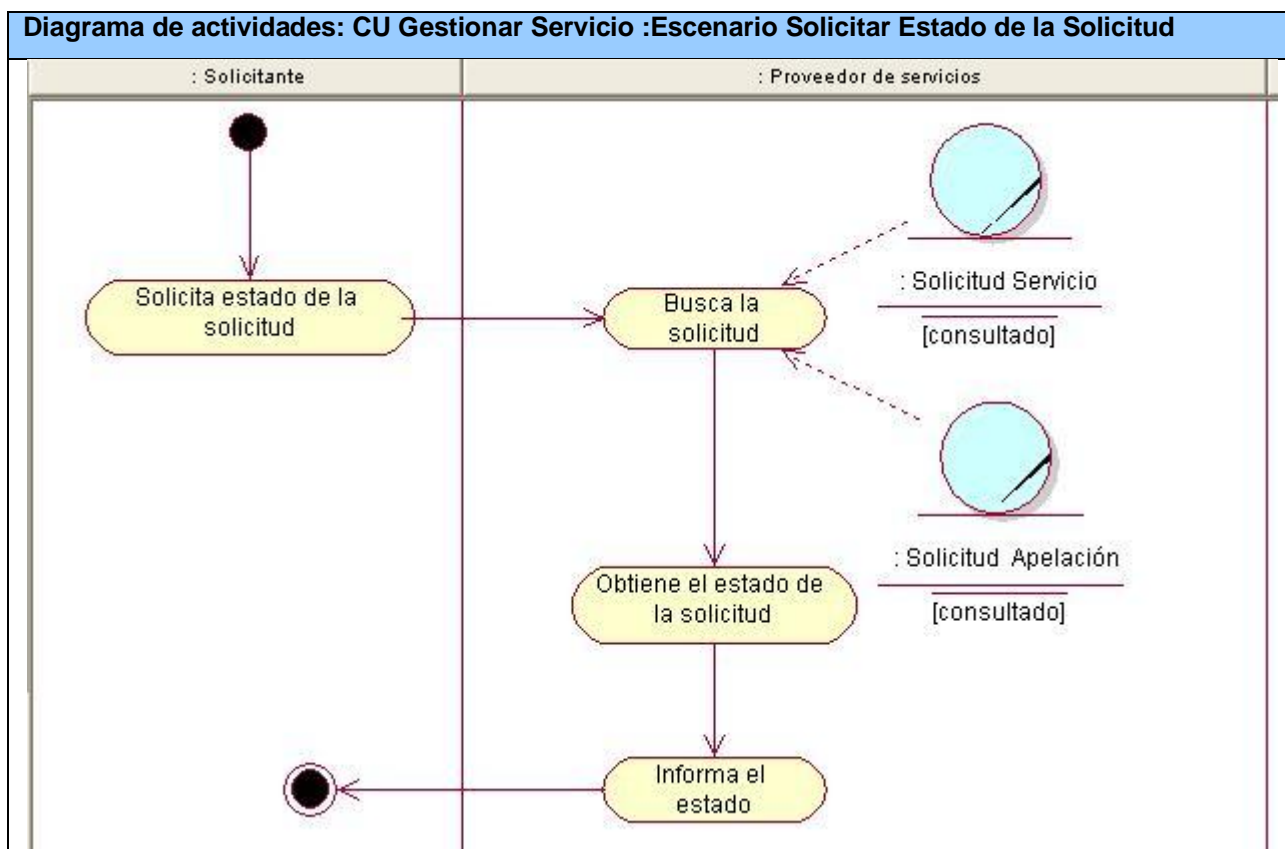


Figura 20 Diagrama de actividades: CU Gestionar Servicio: Escenario Solicitar Estado de la Solicitud

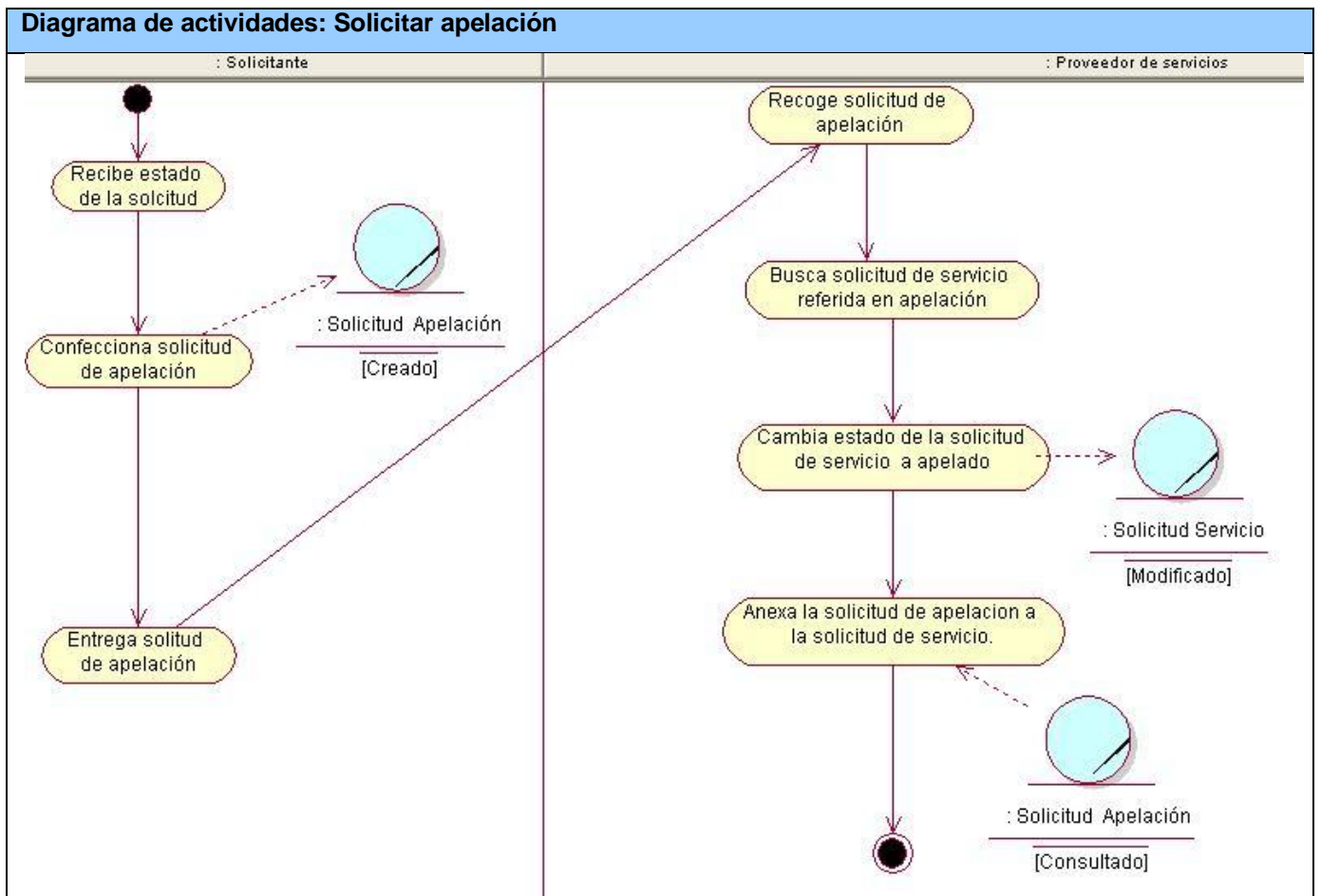


Figura 21 Diagrama de Actividades CU Solicitar Apelación

Anexo 3.

Diagrama de clases del modelo de objetos.

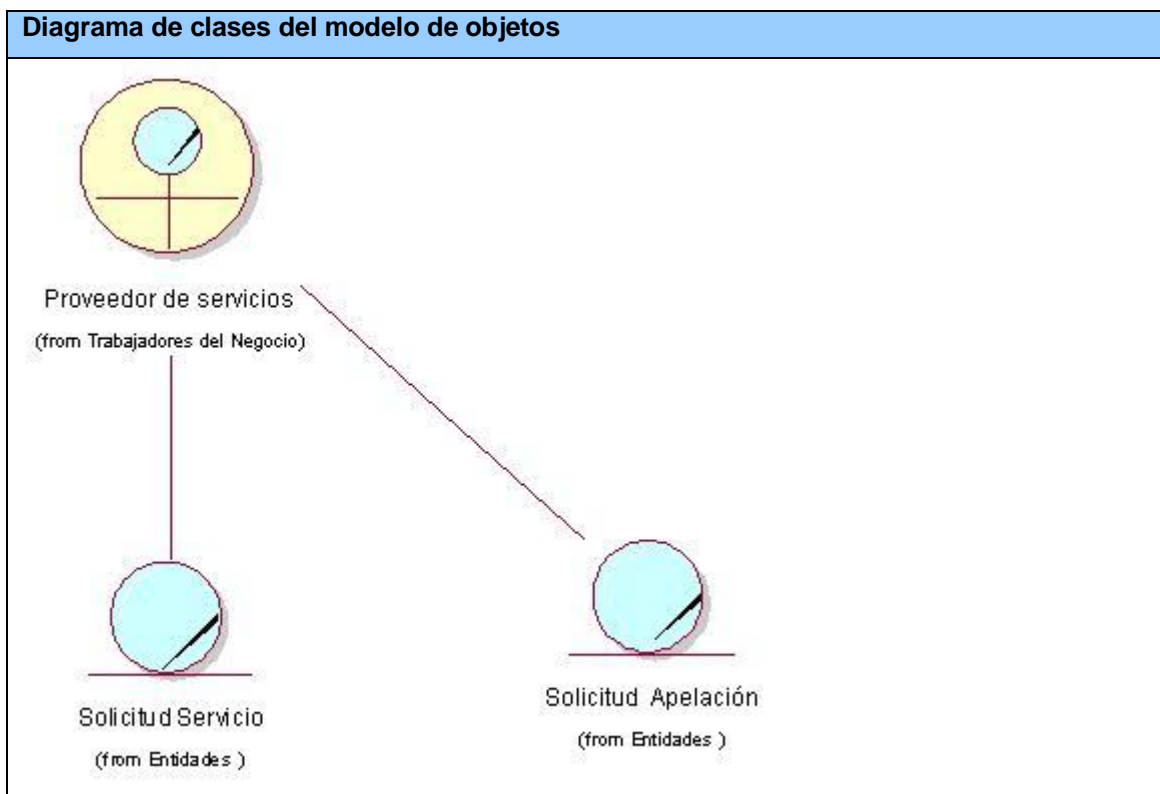


Figura 22 Diagrama de Actividades CU Solicitar Apelación

Anexo 4

Descripción textual de Casos de Uso del Sistema

Caso de Uso Gestionar Servicios

Caso de Uso:	Gestionar Servicios.
Actores:	Proveedor de servicios.
Propósito:	Insertar, eliminar o modificar en el sistema la información de los servicios de soporte que brinda la dirección de informatización.
Resumen:	El proveedor de servicios adiciona, elimina o actualiza determinado servicio.
Referencia:	RF1
Precondiciones:	El actor debe estar autenticado en el sistema con una cuenta dentro del rol Proveedor de Servicios.
Poscondiciones:	Se actualiza la información de los servicios.
Flujo normal de los eventos:	
Sección “Adicionar Servicio”	
Acciones del Actor	Respuesta del Sistema
1. El actor solicita acceso a la sección Proveedores.	1.1 El sistema muestra vínculos a las opciones <ul style="list-style-type: none"> ➤ Gestionar Proveedor. ➤ Gestionar Servicio. ➤ Solicitudes Pendientes.
2. El actor selecciona la opción Gestionar Servicio.	2.1 El sistema brinda las opciones <ul style="list-style-type: none"> ➤ Adicionar Servicio. ➤ Editar Servicio.
3. El actor selecciona la opción Adicionar Servicio.	3.1 El sistema muestra la interfaz de registro de un nuevo servicio mediante un formulario que solicita los datos del servicio (nombre del servicio, breve descripción del servicio) que se desea adicionar.
4. El actor inserta los datos del nuevo servicio y envía la información.	4.1 El sistema comprueba que todos los campos del formulario han sido completados. 4.2 El sistema comprueba que todos los datos

	<p>de formulario son válidos.</p> <p>4.3 El sistema comprueba que el servicio no esté en la base de datos.</p> <p>4.4 El sistema almacena la información del nuevo servicio.</p> <p>4.5 El sistema muestra una notificación informando al usuario que se Adicionó con éxito el servicio</p>
--	---

Curso Alterno de los eventos:

Acciones del Actor	Acciones del Actor
---------------------------	---------------------------

	4.1 El sistema muestra una notificación “El servicio ya se encuentra en el sistema”
--	---

Prioridad:	Crítico
-------------------	----------------

Prototipo



Flujo normal de los eventos:

Sección “Eliminar Servicio”

Acciones del Actor	Respuesta del Sistema
---------------------------	------------------------------

1. El actor solicita acceso a la sección Proveedores.	<p>1.1 El sistema muestra vínculos a las siguientes opciones:</p> <ul style="list-style-type: none"> ➤ Gestionar Proveedor. ➤ Gestionar Servicio. ➤ Solicitudes Pendientes.
---	--

2. El actor selecciona la opción Gestionar Servicio.	2.1 El sistema brinda las opciones: <ul style="list-style-type: none"> ➤ Adicionar Servicio. ➤ Editar Servicio.
3. El actor selecciona la opción Editar Servicio.	3.1 El sistema muestra la lista de servicios existentes en la aplicación.
4. El actor selecciona el servicio que desea eliminar y activa el vínculo editar.	4.1 EL sistema muestra el servicio en detalle.
5. El actor activa el botón eliminar.	5.1 El sistema muestra un mensaje de confirmación “¿Está seguro que desea eliminar el o los servicios seleccionados?”
6. El actor acepta el mensaje.	6.1 El sistema elimina el servicio seleccionado, borrándolo de la lista y eliminándolo de la base de datos del sistema. 6.2 El sistema muestra un mensaje “El servicio seleccionado ha sido eliminado.”
Curso Alternativo de los eventos:	
Acciones del Actor	Respuesta del Sistema
Prioridad:	Crítico
Prototipo	



Sección “Modificar Servicio”

Flujo normal de los eventos:

Acciones del Actor	Respuesta del Sistema
1. El actor solicita acceso a la sección Proveedores.	1.1 El sistema muestra vínculos a las siguientes opciones: <ul style="list-style-type: none"> ➤ Gestionar Proveedor. ➤ Gestionar Servicio. ➤ Solicitudes Pendientes.
2. El actor selecciona la opción Gestionar Servicio	2.1 El sistema brinda las opciones: <ul style="list-style-type: none"> ➤ Adicionar Servicio ➤ Editar Servicio
3. El actor selecciona la opción Editar Servicio.	3.1 El sistema muestra la lista de servicios existentes en la aplicación.
4. El actor selecciona el servicio que desea actualizar	4.1 El sistema muestra los detalles del servicio seleccionado


<p>5. El actor modifica los campos que desee actualizar en el servicio y envía la información.</p>	<p>5.1 El sistema comprueba que todos los campos del formulario están completos. 5.2 El sistema comprueba que todos los datos del formulario son válidos 5.3 El sistema guarda los nuevos datos del servicio modificado.</p>
<p>Curso Alterno de los eventos:</p>	
<p>Acciones del Actor</p>	<p>Respuesta del Sistema</p>
<p>Prioridad:</p>	<p>Crítico</p>
<p>Prototipo</p>	
	

Tabla 19 Descripción del Caso de Uso Gestionar Servicios.

Casos de Uso Gestionar Estado Solicitud

Caso de Uso:	Gestionar Estados Solicitudes	
Actores:	Proveedor de Servicios.	
Propósito:	Permite que el proveedor de servicio determine el estado de una solicitud.	
Resumen:	El caso de uso se inicia cuando el actor accede a la sección del sistema Proveedores donde puede revisar las solicitudes pendientes y apeladas para determinar el estado final de las mismas que puede ser aprobada o denegada.	
Referencia:	R1.4	
Precondiciones:	El actor debe estar autenticado en el sistema con una cuenta que pertenezca al rol Proveedor de Servicios.	
Poscondiciones:	La solicitud pasa de estado Pendiente a aprobada o denegada.	
Flujo normal de los eventos:		
Acciones del Actor	Respuesta del Sistema	
1. El actor accede a la opción Proveedores.	1.1 El sistema muestra una lista con la cantidad de solicitudes recibidas por estado.	
2. El actor selecciona un estado.	1.2 El sistema muestra una lista de las solicitudes que poseen el estado solicitado.	
3. El actor selecciona una solicitud.	3.1 El sistema muestra en detalle la información de la solicitud (proyecto que la realiza, líder de proyecto, área a la que pertenece, especificaciones de la solicitud etc.)	
4. El actor activa la opción Determine Estado.	4.1 El sistema muestra la interfaz perteneciente a la opción seleccionada..	
5. El usuario especifica el estado final que desea que tome la		


solicitud	<p>5.1 El sistema modifica el estado de la solicitud.</p> <p>5.2 El sistema envía una notificación al solicitante por correo informando el estado en que quedó su solicitud.</p>
Curso Alterno de los eventos:	
Acciones del Actor	Respuesta del Sistema
Prioridad:	Crítico
Prototipo	
	

Tabla 20 Descripción del Caso de Uso Gestionar Estados Solicitud.

Casos de Uso Solicitar Servicio

Caso de Uso:	Solicitar Servicio	
Actores:	Solicitante	
Propósito:	Permite al solicitante realizar una solicitud de un servicio determinado.	
Resumen:	El caso de uso se inicia cuando el actor accede a la aplicación con el fin de solicitar determinado servicio brindado por los proveedores.	
Referencia:	R1.4	
Precondiciones:	El solicitante debe poseer una cuenta del dominio UCI	
Poscondiciones:	La solicitud de servicio queda almacenada en el sistema.	
Flujo normal de los eventos:		
Acciones del Actor	Respuesta del Sistema	
1. El actor solicita acceso a la sección Servicios Disponibles.	1.1 El sistema muestra una lista de los servicios disponibles y sus descripciones.	
2. El usuario selecciona el servicio que desea solicitar y activa la opción solicitar.	2.1 El sistema muestra una interfaz donde el usuario introduce los datos de la solicitud del servicio que desea.	
3. El usuario entra los datos de la solicitud.	3.1 El sistema valida la información de la solicitud. 3.2 El sistema almacena la solicitud. 3.3 El sistema envía una notificación a los proveedores informando de la existencia de una nueva solicitud. 3.4 El sistema muestra un mensaje "Solicitud Enviada con éxito."	
Curso Alternativo de los eventos:		
Acciones del Actor	Respuesta del Sistema	
Prioridad:	Crítico	

Prototipo	
<p>Asunto: *</p> <input type="text"/> <p>Datos del servicio</p> <p>Servicio a solicitar: *</p> <p>Publicación de aplicaciones web</p> <p>Observaciones adicionales:</p> <div style="border: 1px solid #ccc; height: 40px;"></div> <p>Datos del solicitante</p> <p>Rol dentro del proyecto: *</p> <p>Lider de Proyecto</p> <p>Proyecto productivo al que pertenece: *</p> <p>Gestión de Graduados</p> <p>Perteneciente al área: *</p> <input type="text"/> <p><input type="button" value="Vista previa"/> <input type="button" value="Enviar"/></p>	<p>sistema</p> <p>Pendientes 0</p> <p>Aprobada 0</p> <p>Denegada 0</p> <p>Apelada 0</p> <p>Es usuario</p> <p>solicitante</p> <ul style="list-style-type: none"> <input type="checkbox"/> Proveedores <input type="checkbox"/> Servicios <input type="checkbox"/> Solicitudes <ul style="list-style-type: none"> <input type="checkbox"/> Solicitud de apelación <input type="checkbox"/> Solicitud de Publicación de sitios Web <input checked="" type="checkbox"/> Solicitud de servicio <input type="checkbox"/> Mi cuenta <input type="checkbox"/> Cerrar sesión

Tabla 21 Descripción del Caso de Uso Solicitar Servicios.

Casos de Uso Solicitar Apelación.

Caso de Uso:	Solicitar Apelación
Actores:	Solicitante
Propósito:	Permite al solicitante realizar una solicitud de apelación en caso de no estar de acuerdo si el estado de una solicitud de servicio queda Denegado.
Resumen:	El caso de uso se inicia cuando el actor solicita una apelación al proveedor de servicio mediante el sistema porque su solicitud no fue aceptada por los proveedores.
Referencia:	R1.4
Precondiciones:	El solicitante debe poseer una cuenta del dominio UCI EL solicitante debe pertenecer a un proyecto registrado en la aplicación. La solicitud a la que se le solicita una la apelación no puede haber sido apelada anteriormente.
Poscondiciones:	La solicitud de apelación a un servicio queda almacenada en el sistema.


Flujo normal de los eventos:	
Acciones del Actor	Respuesta del Sistema
1. El actor solicita acceso a la sección solicitudes apelación.	1.1 El sistema muestra las solicitudes realizadas y los estados de las mismas.
2. El actor selecciona la solicitud que desea apelar.	2.1 El sistema muestra una interfaz donde el usuario introduce una breve descripción de los motivos de la apelación.
3. El actor entra los datos de la apelación.	3.1 El sistema guarda la información de la apelación. 3.2 El sistema envía un correo al proveedor para notificar la nueva apelación realizada.
Curso Alternativo de los eventos:	
Acciones del Actor	Respuesta del Sistema
2. El actor selecciona una solicitud que ya ha sido apelada anteriormente.	2.1 El sistema muestra una notificación informando al usuario que no es válida la apelación.
Prioridad:	Secundario
Prototipo	
	

Tabla 22 Descripción del Caso de Uso Solicitar Apelación.

Casos de Uso Mostrar Historial Solicitudes

Caso de Uso:	Mostrar Historial Solicitudes	
Actores:	Solicitante	
Propósito:	Permite al solicitante ver un historial de todas las solicitudes que ha realizado con su respectivo estado.	
Resumen:	El caso de uso se inicia cuando el actor solicita ver el historial que tiene de todas las solicitudes realizadas.	
Referencia:		
Precondiciones:	El solicitante debe poseer una cuenta del dominio UCI	
Poscondiciones:	El sistema mostrará un historial de todas las solicitudes que realizó el solicitante.	
Flujo normal de los eventos:		
Acciones del Actor	Respuesta del Sistema	
1. El actor solicita acceso a la sección Mostrar Historial de Solicitudes.	1.1 El sistema busca si tiene solicitudes de servicio realizadas. 1.2 El sistema busca si tiene solicitudes de apelación realizada para algún servicio. 1.3 El sistema muestra todas las solicitudes que tiene realizada el actor incluyendo fecha de creación de las mismas.	
Curso Alternativo de los eventos:		
Acciones del Actor	Respuesta del Sistema	
1. El actor solicita acceso a la sección Mostrar Historial de Solicitudes.	El sistema muestra un mensaje "No ha realizado ninguna solicitud "	
Prioridad:	Secundario	
Prototipo		

Historial de solicitudes

[Principal](#)

Historial de solicitudes

Estado actual de la solicitud	Fecha de creación	Servicio solicitado	Autor de la Solicitud	Solicitud
apelado	04/06/2008 - 15:12	Publicación de aplicaciones web	halena	Solicitud para salva de código
apelado	04/06/2008 - 16:50	Publicación de aplicaciones web	halena	Solicitud para que se brinde el servicio 2
	04/06/2008 - 18:33		halena	Solicitud de apelacion de la solicitud de servicio paar salva de código
	05/06/2008 - 22:32		halena	m.kljhsdf;ilksdh
pendiente	06/06/2008 - 14:41	Repositorio de Base de Datos	halena	Prueba

Tabla 22 Descripción del Caso de Uso Mostrar Historial Solicitudes.

Anexo 5

Diagrama de colaboración del análisis.

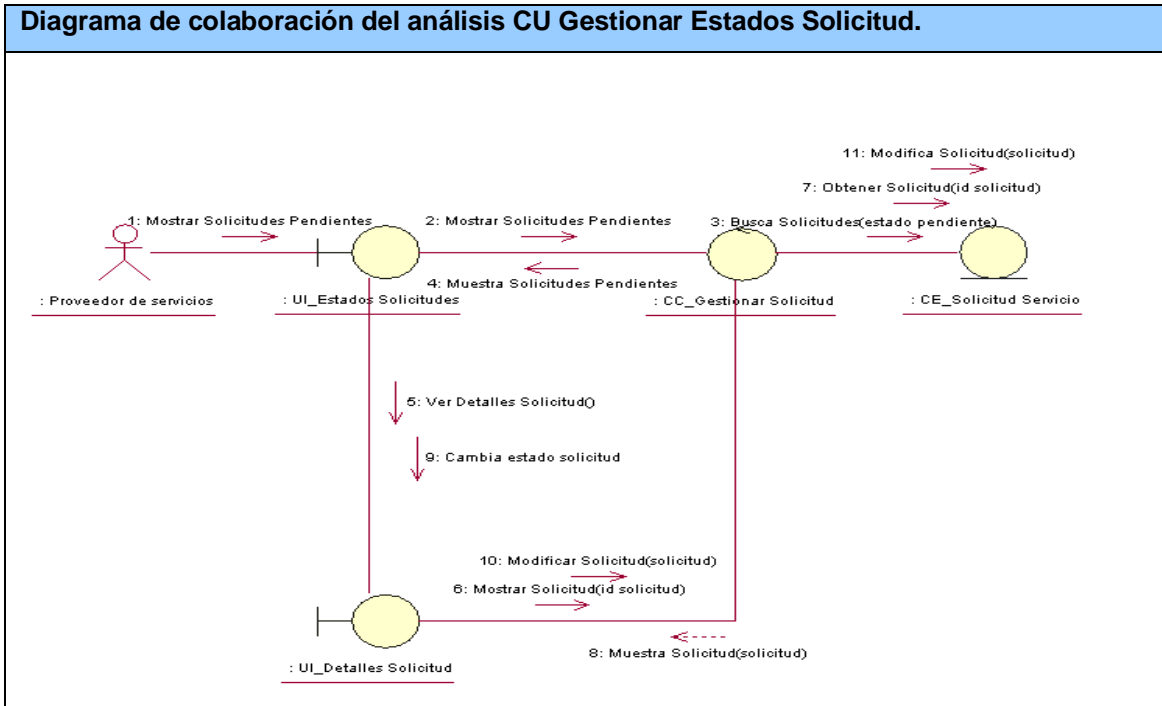


Figura 23 Diagrama de colaboración del análisis CU Gestionar Estados Solicitud.

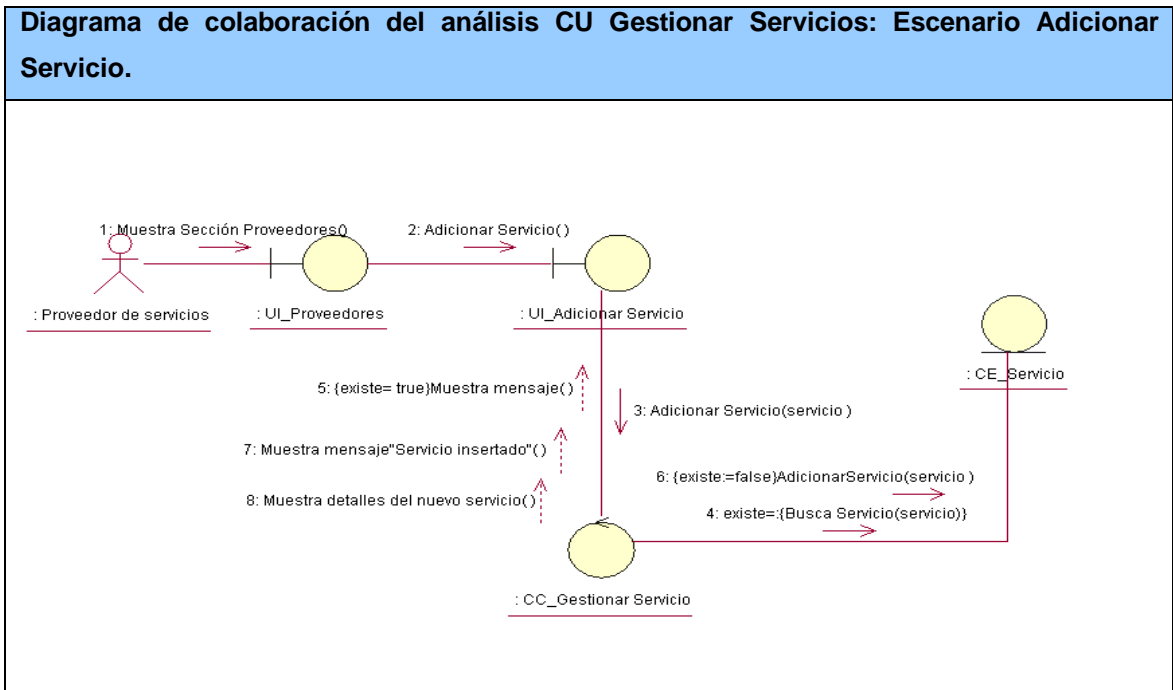


Figura 24 Diagrama de colaboración del análisis CU Gestionar Servicios: Escenario Adicionar Servicio.

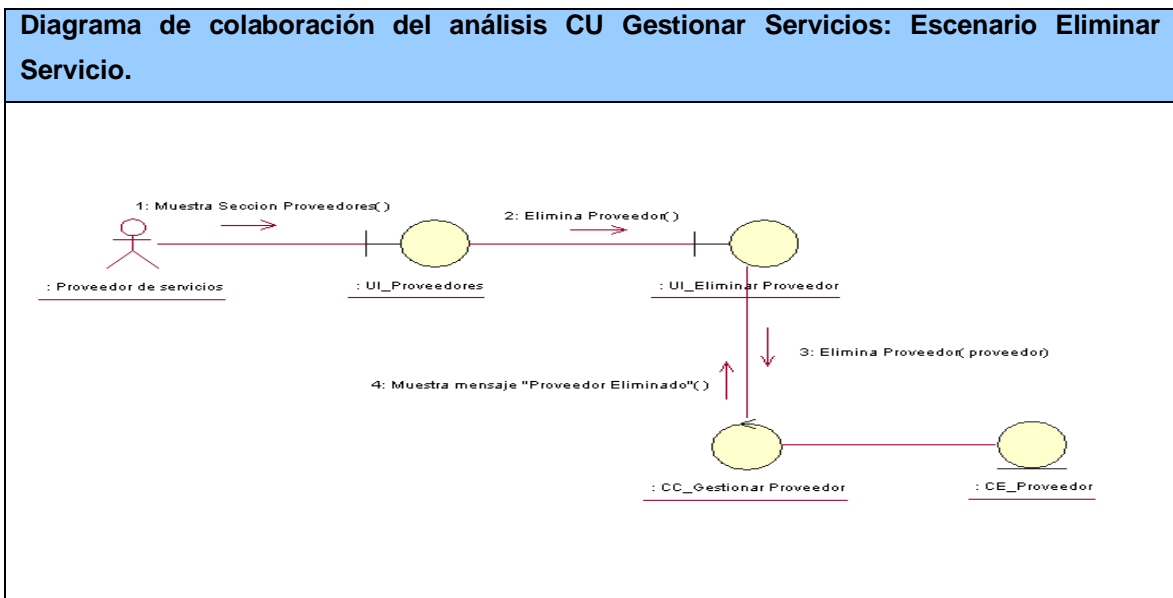


Figura 25 Diagrama de colaboración del análisis CU Gestionar Servicios: Escenario Eliminar Servicio.

Diagrama de colaboración del análisis CU Gestionar Servicios: Escenario Modificar Servicio.

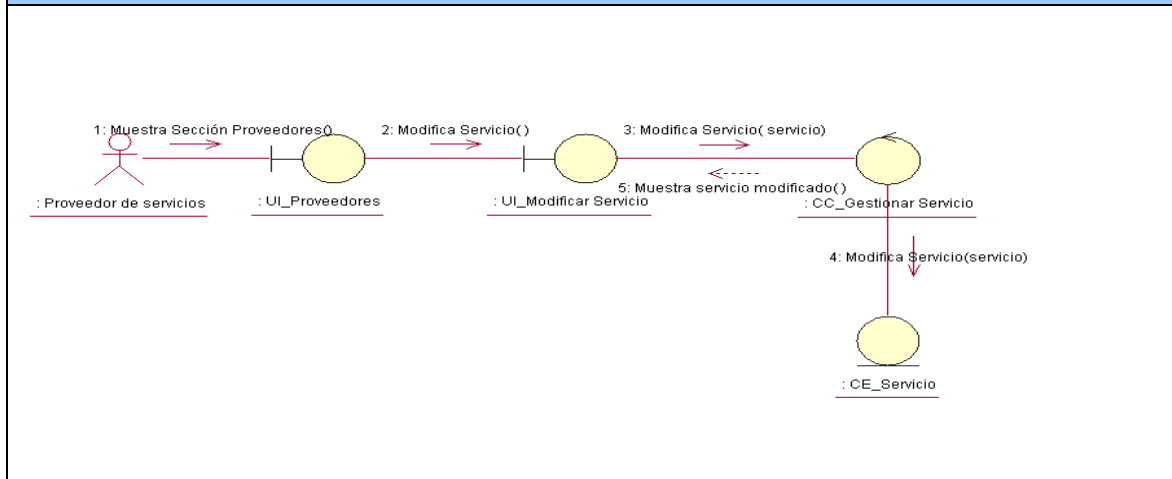


Figura 26 Diagrama de colaboración del análisis CU Gestionar Servicios: Escenario Modificar Servicio.

Diagrama de colaboración del análisis CU Solicitar Apelación.

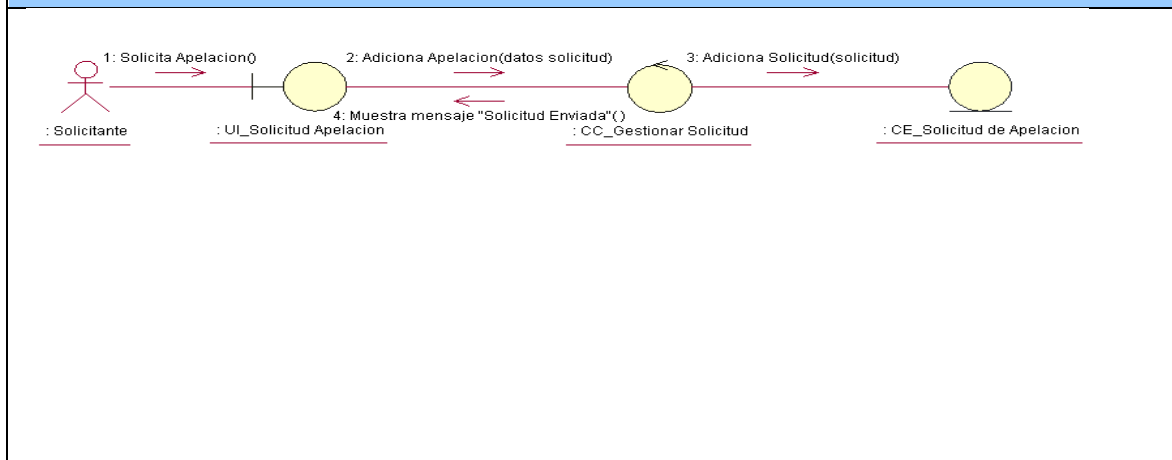


Figura 27 Diagrama de colaboración del análisis CU Solicitar Apelación.

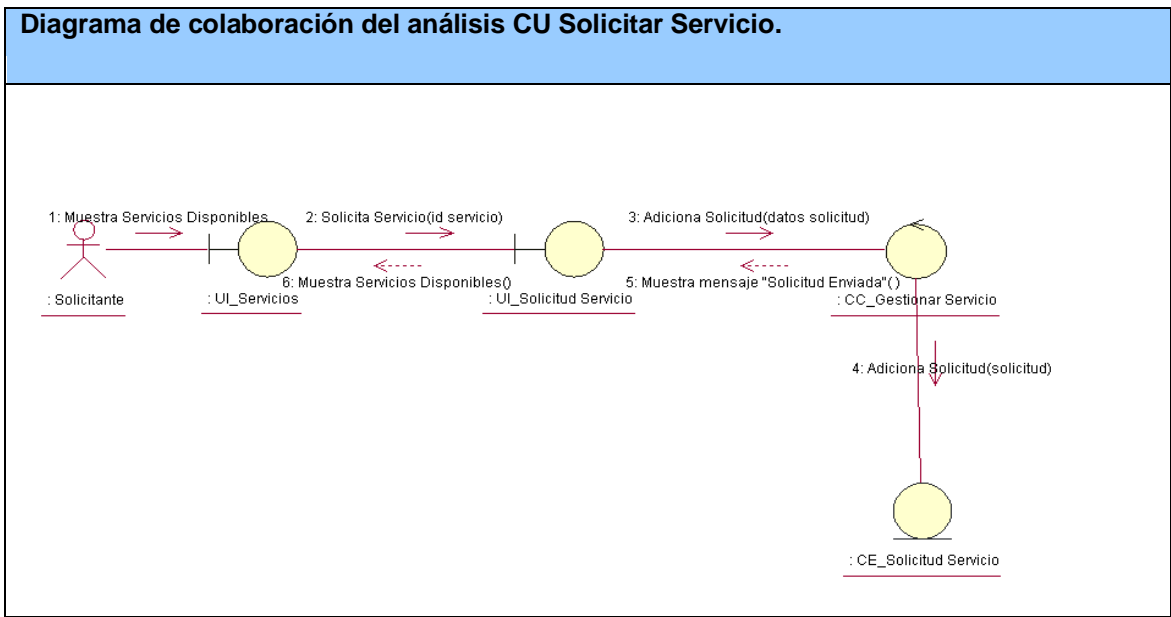


Figura 28 Diagrama de colaboración del análisis CU Solicitar Apelación.

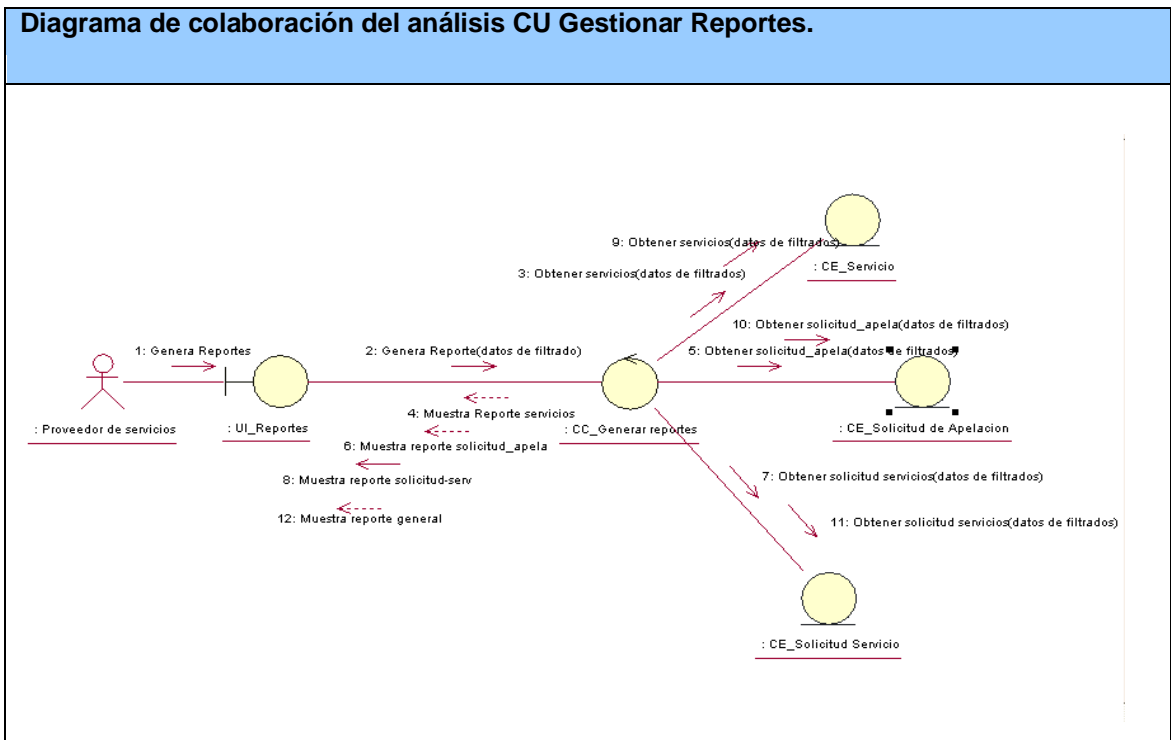


Figura 29 Diagrama de colaboración del análisis CU Gestionar Reportes.

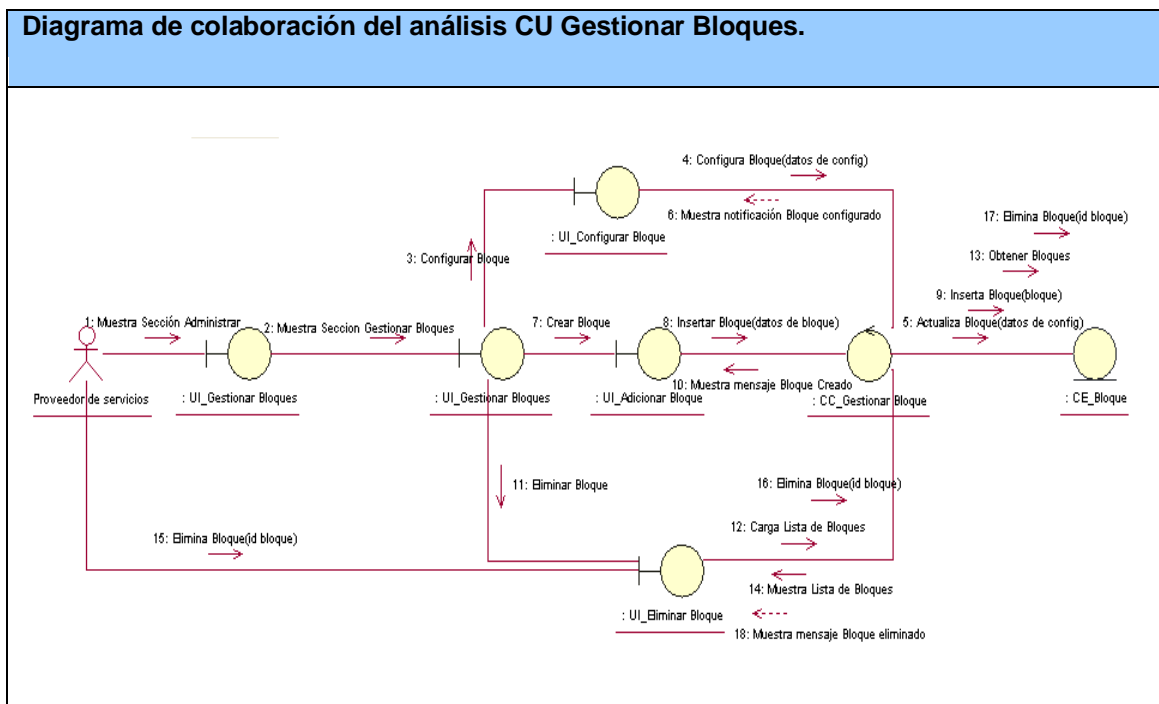


Figura 30 Diagrama de colaboración del análisis CU Gestionar Bloques.

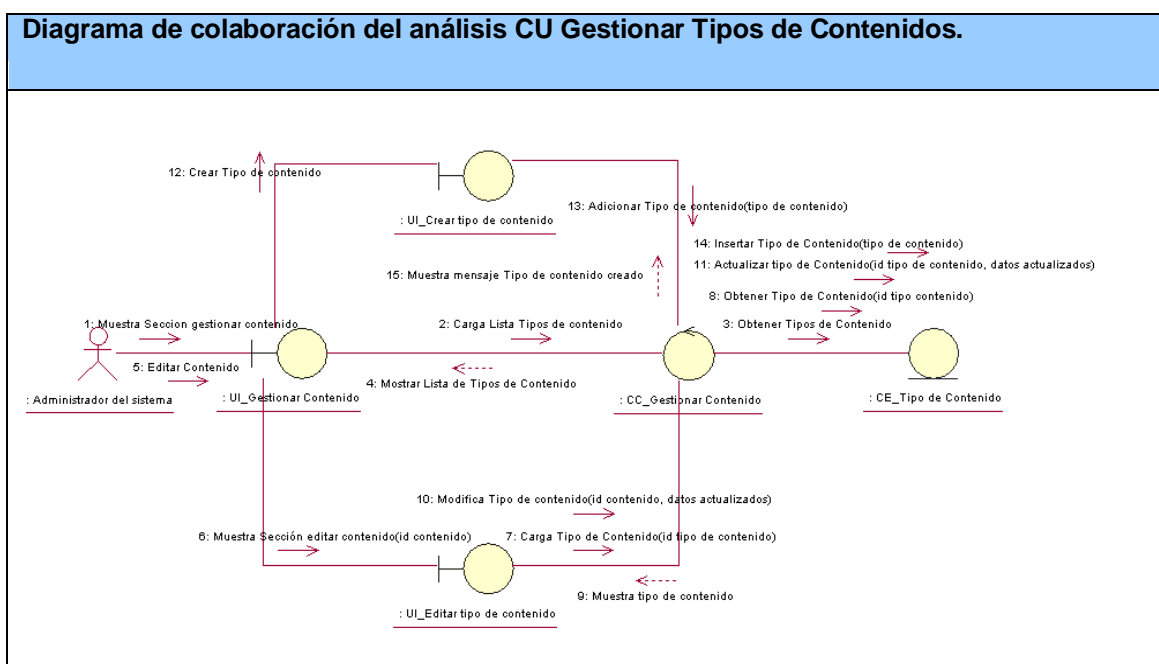


Figura 31 Diagrama de colaboración del análisis CU Gestionar Tipos de Contenido.

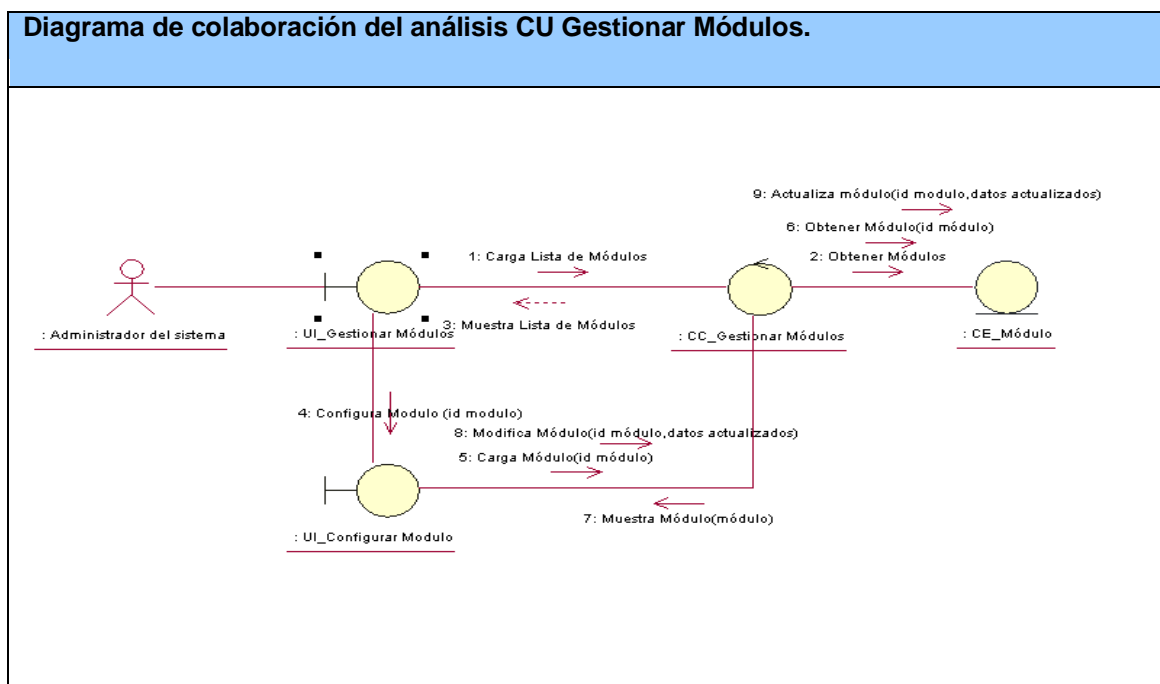


Figura 32 Diagrama de colaboración del análisis CU Gestionar Módulo