

UNIVERSIDAD DE LAS CIENCIAS INFORMATICAS  
FACULTAD 10



## Análisis y Diseño de la Intranet de Residencias de Protocolo del Consejo de Estado de la República de Cuba.

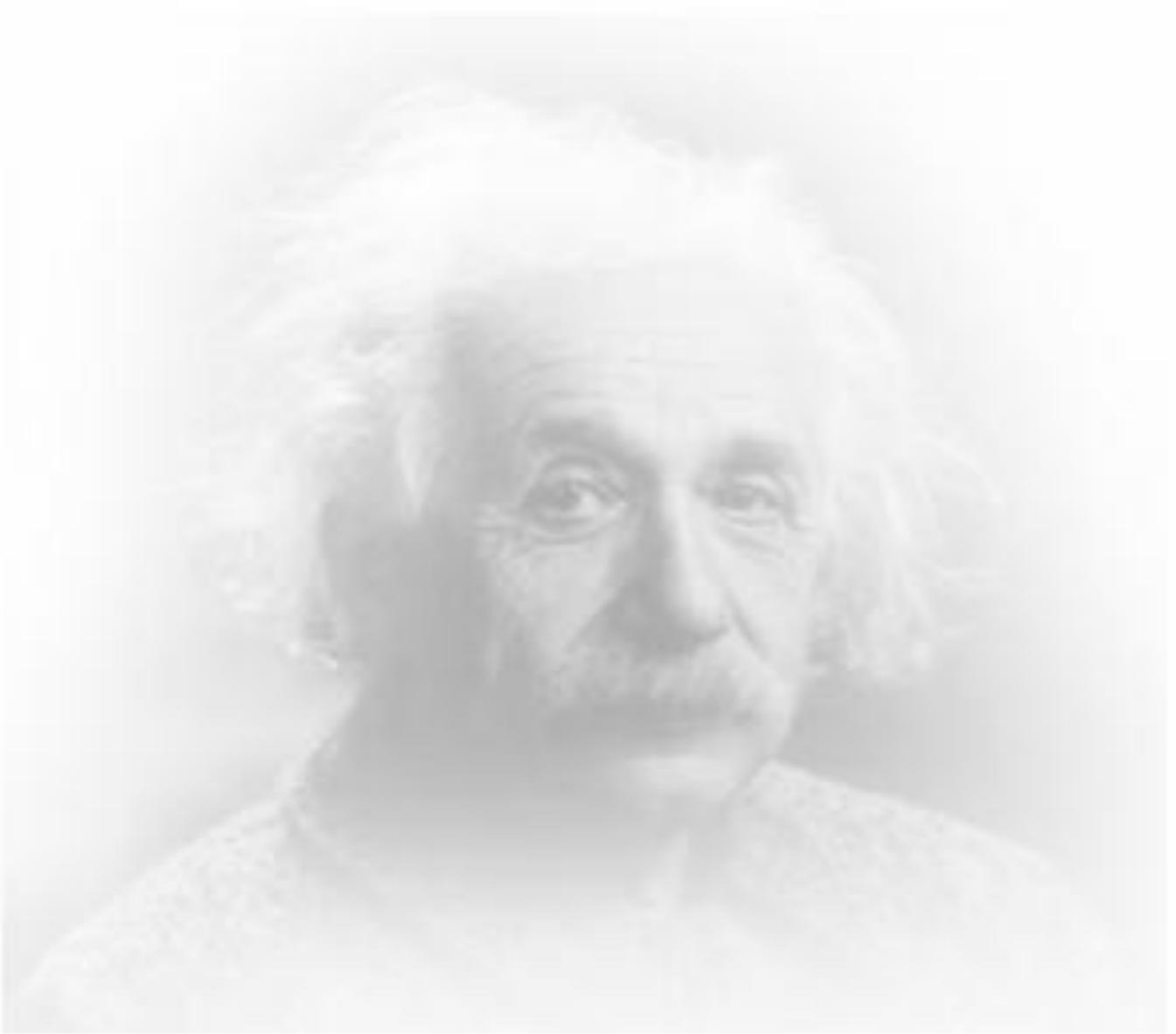
---

TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE INGENIERO EN CIENCIAS INFORMÁTICAS.

Autores: Yolaine Yaima Matienzo Montero.  
Yuniet Collazo García

Tutor: Ing. Indira Peraza Sierra.

Ciudad de la Habana  
2009



*"Nunca consideres el estudio como una obligación sino como una oportunidad para penetrar en el bello y maravilloso mundo del saber."*

*Albert Einstein*

## DECLARACIÓN DE AUTORÍA

Declaramos que somos los únicos autores de este trabajo y autorizamos a la Facultad 10 de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

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

---

Yolaine Yaima Matienzo Montero

---

Yuniet Collazo García

---

Ing. Indira Peraza.

## ***Agradecimientos:***

**“A todos los que de una forma u otra han hecho posible el sueño de muchos años.”**

### ***Yolaine:***

- ❖ *A la **Revolución** por permitirme crecer profesionalmente en una universidad de excelencia y formar parte de las tropas del futuro.*
- ❖ *A mi **mamichuli** del alma y mi **papito querido** por llenarme de tanto amor, por darme las fuerzas para seguir adelante, por todas las noches de desvelo esperando este momento, por ser los mejores papas del mundo, sin ustedes nada de esto hubiera sido posible, los amo.*
- ❖ *A **mamiyé** por ser tan especial, tan única, por su preocupación constante y por todo el amor que siempre me ha transmitido.*
- ❖ *A **papi Julio** y **abuela Sonia** que hoy no están pero los llevo en el corazón a donde quiera que voy, gracias por confiar en mí, por todo el cariño que siempre me dieron y ojalá que donde quiera que estén se sientan orgullosos.*
- ❖ *A mi novio **Kiki** por compartir conmigo momentos inolvidables, por convertir mis tristezas en alegrías, y cada alegría en un sueño, Te quiero.*
- ❖ *A **Mercy** y **familia** por su apoyo incondicional, su ejemplo, sus constantes consejos, por permitirme ser parte de ustedes y quererme como tal, por estar a mi lado siempre, por todas las alegrías, tristezas y el tiempo compartido y sobre todo porque gran parte de lo que soy hoy se lo debo a cada uno de ustedes. Mil gracias de todo corazón.*
- ❖ *A mis **tíos, primos, abuelos, Rafe** y **familia en general**, por ser como son, por quererme y apoyarme cada una de mis decisiones, por estar a mi lado en cualquier circunstancia de la vida.*
- ❖ *A **Blanqui** que a pesar de la distancia me abrió sus brazos ganándose mi cariño y admiración.*
- ❖ *A mis **amigos Maikel** y **Alexei**, por permitirme contar con ustedes siempre que los necesité.*
- ❖ *A **Héctor, Diosmi, William, Indira** y **Tomás** por ser excelentes guías, compañeros, amigos, por su apoyo incondicional.*
- ❖ *Al **grupo 10502** que demostraron que juntos es posible salir adelante y por compartir estos 5 años.*

- ❖ A la **gente de Venezuela** que me apoyaron durante el tiempo que estuve allá, en especial a **mimi** por todo su cariño y protección.

**Yuniet:**

- ❖ A **Fidel Castro** y la **Revolución** por crear esta universidad y ofrecerme la oportunidad de estudiar en ella.
- ❖ A **Mami** y **Papi** por el apoyo incondicional que recibo de ustedes todos los días, por educarme y guiarme por el camino correcto, por todo el tiempo de esfuerzos y sacrificios que han dedicado a mí, siempre estaré orgulloso de ustedes, los quiero mucho.
- ❖ A mi **Hermano** por ser la persona con la que siempre puedo contar y por hacerme sentir orgulloso con sus resultados. **Chin:** puedes tener la seguridad de que siempre estaré ahí cuando me necesites.
- ❖ A mi novia **Mary** por compartir a mi lado los últimos 3 años, por recordarme cada momento de felicidad que hemos pasado juntos, por entenderme y apoyarme cuando más lo necesitaba, por aceptarme con mis defectos y virtudes, por hacerme sentir parte de ti, espero que sepas que te amo mi niña.
- ❖ A mis **Abuelos, Tíos** y **Primos** por brindarme sus sabios consejos y experiencias y por dedicar una parte de su tiempo a preocuparse por mí.
- ❖ A mi otro hermano: **Hardam** por todas esas cosas que sabes hacer por mí y por llegar a ser la persona que más me conoce.
- ❖ A mi **Grupo 10502** por compartir juntos estos 5 años, por ser mis amigos y poder contar con ustedes. Disculpen sino menciono alguno de sus nombres, creo que sería injusto porque siento que con cada uno he compartido momentos que son experiencias para recordar toda la vida.
- ❖ A **William, Adrian** e **Indira** por jugar un papel importante en el desarrollo de este trabajo.
- ❖ A mis **Profes** que de una forma u otra contribuyeron a mi superación profesional y me hicieron ver la vida desde diferentes puntos de vistas.
- ❖ Al profe **Tomás** por permitirme intercambiar criterios y opiniones, por haberlo conocido antes de culminar mis estudios universitarios, porque personas como usted no logramos conocer muy a menudo.
- ❖ Para finalizar quiero agradecer a todas aquellas personas que en algún momento de estos 5 años me han estrechado su mano y han contribuido a mi superación como profesional.

*Dedicatoria:*

*“A nuestros padres, porque todo lo que somos es el fruto de su sacrificio y amor infinito.”*

## RESUMEN

Las Casas de Protocolo del Consejo de Estado de la República de Cuba tienen como principal objetivo brindar hospedaje a personalidades de Cuba y el resto del mundo, ofreciendo una atención de excelencia. Actualmente cuentan con una intranet que brinda información acerca de la institución y sus servicios. Dicha intranet no representa una solución acorde a los intereses y objetivos del lugar, ni logra estar a la altura de sus huéspedes, es por ello que se concibió una nueva propuesta de diseño a partir de las no conformidades requeridas.

El presente trabajo de diploma tiene el propósito de diseñar una nueva intranet para las Residencias de Protocolo con el objetivo de satisfacer los requerimientos actuales en correspondencia con el nivel que se desea alcanzar en el lugar. Lo anterior deriva primeramente en la obtención de un mejor diseño gráfico que se identifique con sus instalaciones y resulte atractivo y práctico para sus huéspedes, y en segundo lugar ampliar y lograr una mejor ubicación de la información. Para cumplir con las expectativas propuestas de forma más eficiente se decidió hacer uso de un Sistema de Administración de Contenidos (CMS) que junto a las herramientas y tecnologías adecuadas permitieron modelar una aplicación web idónea a las Casas de Protocolo del Consejo de Estado de la República de Cuba sirviendo como guía y punto de partida para la implementación de un producto confiable y eficiente.

## PALABRAS CLAVES

Protocolo, web, Drupal, información, ingeniería de software.

# ÍNDICE

<b>INTRODUCCIÓN.....</b>	<b>1</b>
<b>CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.....</b>	<b>4</b>
1.1  Introducción. ....	4
1.2  Portal Web .....	4
1.3  Sistemas de Administración de Contenidos (CMS).....	5
1.3.1  ¿Por qué utilizar los CMS en el desarrollo de portales webs?.....	5
1.3.2  CMS Drupal. ....	6
1.4  Lenguaje de programación. ....	7
1.4.1  PHP (PHP Hypertext Pre-processor).....	7
1.4.2  HTML (HyperText Markup Language) .....	8
1.4.3  CSS (Cascading Style Sheets) .....	9
1.4.4  JavaScript. ....	9
1.5  Entorno de Desarrollo Integrado (IDE). ....	10
1.5.1  Quanta Plus. ....	10
1.5.2  Zend Studio.....	10
1.6  Gestor de Base de Datos.....	11
1.6.1  PostgreSQL. ....	11
1.6.2  MySQL.....	12
1.6.3  Base de datos centralizadas y distribuidas.....	12
1.7  Servidores dedicados y compartidos.....	13
1.8  Servidor Web. ....	13
1.8.1  Apache.....	14
1.9  Metodologías de Desarrollo de Software.....	15
1.9.1  Extreme Programming (XP).....	15
1.9.2  Rational Unified Process (RUP).....	17
1.10  Herramienta para la Modelación Visual del Sistema. ....	18
1.10.1  Visual Paradigm.....	19

1.11	Conclusiones.....	20
<b>CAPÍTULO 2: CARACTERIZACIÓN DEL SISTEMA.....</b>		<b>21</b>
2.1	Introducción.....	21
2.2	Problema.....	21
2.3	Objeto de Automatización.....	22
2.4	Información que se maneja.....	22
2.5	Propuesta de Sistema.....	22
2.6	Diagrama de clases del Modelo de Dominio .....	22
2.7	Especificación de los Requisitos del Software.....	25
2.7.1	Requerimientos Funcionales .....	25
2.7.2	Requerimientos No Funcionales.....	28
2.8	Definición de los Casos de Uso .....	29
2.8.1	Actores del Sistema .....	29
2.8.2	Casos de uso del sistema.....	30
2.8.3	Diagrama de casos de uso del sistema.....	33
2.8.4	Descripción de los casos de uso del sistema.....	34
2.9	Conclusiones .....	61
<b>CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA.....</b>		<b>63</b>
3.1	Introducción.....	63
3.2	Análisis.....	63
3.2.1	Diagrama de clases del análisis.....	63
3.2.2	Diagrama de interacción (Colaboración).....	68
3.3	Diseño.....	76
3.3.1	Diagrama de clases del diseño web.....	77
3.3.2	Diseño de la Base de Datos.....	91
3.3.2.1	Diagrama de clases persistentes.....	92
3.3.2.2	Modelo de datos.....	93
3.3.2.3	Descripción de las tablas.....	94
3.4	Diagrama de Despliegue.....	94

3.5 Conclusiones .....	95
<b>CONCLUSIONES.....</b>	<b>96</b>
<b>RECOMENDACIONES .....</b>	<b>97</b>
<b>REFERENCIAS BIBLIOGRÁFICAS.....</b>	<b>98</b>
<b>BIBLIOGRAFÍA.....</b>	<b>99</b>
<b>ANEXOS .....</b>	<b>102</b>
<b>GLOSARIO DE TÉRMINOS .....</b>	<b>112</b>

## Introducción

Las Residencias de Protocolo del Consejo de Estado de la República de Cuba desde su creación han constituido un lugar acogedor para sus huéspedes siendo estos: Jefes de Estados, Primeras Damas, embajadores, delegaciones internacionales, grandes personalidades de la ciencia y la cultura, invitados de la máxima dirección del país, entre otros. Por tal motivo, estas instalaciones están condicionadas para ofrecer servicios de excelencia que incluyen: alojamiento, gastronomía, transporte, aeropuerto y tintorería, cuentan además con salones de protocolos, centros de elaboración de alimentos, instalaciones de almacenamientos y varias oficinas administrativas. Las residencias están decoradas con obras de arte, cuadros y cerámicas de destacados pintores y ceramistas nacionales y extranjeros, jardinería de primera donde el visitante puede deleitarse del paisaje cubano, disfrutar de su entorno y conocer acerca de nuestra cultura. Además todo el personal que labora para la atención a los huéspedes está altamente calificado. Toda la información referente a la institución y sus servicios están centralizados en una intranet que opera sobre su red local la cual sirve de bienvenida digital para los huéspedes luego que estos accedan al navegador web desde las residencias. La realidad es que dicha intranet no representa una solución acorde con los intereses y objetivos del lugar, teniendo en cuenta parámetros tan importantes como el diseño, la organización y distribución de sus contenidos y la información de los servicios que en ella se brindan.

En efecto, este tipo de servicio requiere que su contenido esté actualizado, sea de utilidad y además debe constituir un mecanismo mediante el cual los usuarios interactúan no solo con el lugar sino con el exterior, despertando el interés y la motivación en los mismos. Lógicamente su fracaso o éxito depende de factores tales como: la confidencialidad para poder garantizar que la información sólo sea accesible por las personas autorizadas, la integridad que posibilita que los datos estén protegidos ante cambios no autorizados, la disponibilidad para que los usuarios siempre puedan acceder a los recursos cada vez que los necesiten y muy importante, la utilización de un diseño adecuado que incentive y sea el reflejo de la institución.

Por todo lo antes expuesto se define como **problema científico**:

¿Cómo mejorar el proceso de gestión de la información en las Residencias de Protocolo del Consejo de Estado para lograr mayor satisfacción de huéspedes y directivos?

El **objeto de estudio** lo constituye el proceso de gestión de la información.

El **campo de acción** está enmarcado en el proceso de gestión de la información en las Residencias de Protocolo del Consejo de Estado de la República de Cuba.

Para la realización de este trabajo se trazó como **objetivo general**:

Modelar una aplicación web correspondiente a la Intranet de las Residencias de Protocolo del Consejo de Estado de la República de Cuba para mejorar la gestión de información y la satisfacción de huéspedes y directivos.

Para dar cumplimiento al objetivo general se han definido los siguientes **objetivos específicos**:

1. Realizar un estudio referente a los portales web y las herramientas básicas para su desarrollo.
2. Definir la metodología de desarrollo de software más adecuada para guiar el proceso de desarrollo de la nueva propuesta.
3. Proporcionar una documentación fiable que sirva de base para el desarrollo de una aplicación eficiente.

La **idea que se defiende** es que realizando un diseño adecuado de la intranet teniendo en cuenta las necesidades requeridas, se obtiene una aplicación adaptable a las exigencias y nivel de sus usuarios, logrando consigo efectos directos en la satisfacción de los mismos.

Para el cumplimiento de los objetivos descritos anteriormente se proponen las siguientes tareas de investigación:

- Estudiar la utilización y aplicación de las intranets.
- Estudiar el funcionamiento de Drupal como Sistema de Administración de Contenido.
- Estudiar los Sistemas Gestores de Bases de Datos.
- Investigar los distintos tipos de servidores web.
- Estudiar diferentes lenguajes de programación utilizados en las aplicaciones web.
- Estudiar las herramientas para la modelación visual del sistema.
- Elaborar la documentación siguiendo la metodología seleccionada.
- Analizar las no conformidades detectadas en la revisión de la intranet que se encuentra actualmente en las Residencias de Protocolo para capturar los requisitos funcionales de la nueva propuesta.
- Elaborar los artefactos derivados del flujo de trabajo análisis y diseño.

Para apoyar el desarrollo de la investigación se emplean los siguientes **métodos científicos**:

- ✓ Método teórico:

El Analítico-Sintético permite centrarse en la esencia de los fenómenos y en los rasgos fundamentales que los distinguen, para ello se basa en el análisis de documentos, teorías y otros materiales, en este

caso hacen posible la toma de dediciones para la selección de la metodología adecuada que se ajuste a las necesidades requeridas del software.

✓ Método Empírico:

Las observaciones que se realizan facilitan la obtención de la información necesaria para el levantamiento de los requisitos.

El siguiente trabajo de diploma está estructurado de la siguiente manera:

**Capítulo 1: Fundamentación teórica.**

Se procede a realizar un estudio del estado del arte referente al tema que se aborda. De esto se deriva el estudio e investigación de las metodologías de desarrollo del software, los Sistemas de Administración de Contenidos (CMS) haciendo énfasis en el CMS Drupal, algunos de los lenguajes de programación existentes, gestores de base de datos, servidores web y las herramientas a utilizar para la modelación visual del sistema, llegando a obtener la fundamentación teórica de las tecnologías seleccionadas.

**Capítulo 2: Caracterización del sistema.**

Se define el objeto de automatización, información que se maneja, propuesta del sistema, modelo de dominio y los requisitos funcionales con los que contará el sistema a automatizar a partir de las no conformidades recogidas, se muestran las descripciones de cada caso de uso así como el diagrama de casos de uso del sistema.

**Capítulo 3: Análisis y diseño.**

Se realiza el análisis y diseño del sistema. Se muestran los diagramas de clases del análisis y diseño así como los diagramas de interacción correspondientes. También se realiza una descripción de las clases que permitirán las funcionalidades del sistema.

# Capítulo 1: Fundamentación Teórica.

## 1.1 Introducción.

En el presente capítulo se refleja primeramente el concepto de Portal Web así como los diferentes tipos de portales que existen, haciendo énfasis en las intranets. Se aborda el concepto de CMS, una breve explicación sobre la necesidad de utilizar estos sistemas de administración de contenidos en el desarrollo de portales webs. Se describen las características y ventajas del CMS Drupal, en tal sentido son mencionados y explicados los lenguajes de programación y herramientas que éste utiliza. Además se evalúan las metodologías de desarrollo de software más actuales.

## 1.2 Portal Web

Los portales web surgen como alternativa ante el problema de brindar diversa cantidad de información, recursos y servicios a los usuarios.

Existen diferentes tipos de portales: los horizontales y los verticales. La diferencia radical entre ambos, es que los Portales Horizontales están buscando ofrecer sus servicios para los usuarios en general, sin importar cuál será el contenido del portal, mientras que los Portales Verticales enfocan sus servicios para alguna institución determinada mediante la implementación de Intranets, Extranets o Portales Verticales. (VILLANUEVA, 2007)

En el caso de las Casas de Protocolo se reajusta el tipo de portal vertical (intranet) que es accesible y adaptable a las necesidades de dicha institución.

Las intranets son parecidas a los sitios Web y, como éstos, utilizan protocolos de Internet, con la diferencia de que son redes internas exclusivas de la empresa. El desarrollo de intranets ofrece innumerables beneficios para las empresas e instituciones así como también constituyen la cara digital de las mismas, seguidamente se mencionan las principales ventajas que hacen que su uso sea imprescindible para mantener la competencia.

- Ahorro de costes en las comunicaciones.
- Fácil acceso desde cualquier ordenador.
- Centralización y actualización de la información.

- Define perfiles de usuario y niveles de seguridad que garantizan acceso sólo a las personas indicadas.
- Asegura que la información no se pierda y que llegue a la persona apropiada.
- Comparte la información en forma general, por grupos o individualmente.
- Perfeccionamiento de los recursos por encontrarse disponible la información para todos los que pudieran necesitarla.
- Optimiza la información unificándola y facilitando su tratamiento.

### 1.3 Sistemas de Administración de Contenidos (CMS).

¿Qué es un CMS?

Sistemas de gestión de contenidos (en inglés Content Management Systems o CMS) no son más que software que se utilizan principalmente para facilitar la gestión de webs, ya sea en Internet o en una intranet. (CUERDA, 2004)

Si se fuese a traducir al lenguaje de un ingeniero informático, los Sistemas de Gestión de Contenidos serían:

Sistema: Se refiere a un conjunto de herramientas, que utilizan una base de datos y que alojan una serie de programas en un servidor web, facilitando el acceso a estos programas desde otros programas clientes.

Gestión de Contenidos: Desde el punto de vista del usuario del sistema, se trata de gestionar, de forma uniforme, accesible, y cómoda, un sitio web dinámico, con actualizaciones periódicas, y sobre el que pueden trabajar una o más personas, cada una de las cuales tiene una función determinada. Desde el punto de vista del cliente, se trata de un sitio web dinámico, con apariencia e interfaz uniforme, con un diseño centrado en el usuario, y que permite llevar a cabo fácilmente las tareas para las que ha sido diseñado. (MERELO, 2005)

Los CMS poseen dos funciones fundamentales: facilitar la creación de contenidos y la presentación de esos contenidos, también tienen otras dos funciones importantes: gestión de contenidos y mantenimiento de los mismos; aunque estas dos se pueden incluir en la anterior. Un CMS provee las herramientas necesarias para gestionar el ciclo de vida de los contenidos: creación, gestión, presentación, mantenimiento y actualización.

#### 1.3.1 ¿Por qué utilizar los CMS en el desarrollo de portales webs?

Cuando se tiene una web relativamente pequeña o que no posee muchas funcionalidades se podría pensar que no es necesario la utilización de un CMS, para confirmar esto, se tiene que tener bien definido como será la web, en el caso de que la web que se quiera montar, fuese unas pocas páginas estáticas de las cuales no

se prevé un crecimiento ni actualización futura, entonces sería un error malgastar el tiempo intentando usar un CMS para dicha web, aunque la tendencia hoy en día no está dirigida a realizar sitios con este estilo. Los usuarios independientes utilizan CMS gratuitos para crear y gestionar sus webs personales, obteniendo webs dinámicas con innumerables ventajas y funcionalidades, lo que hace que las mismas sean superiores a las empresas e instituciones que se limitan a tener páginas estáticas que no aportan ningún valor añadido.

Existe una gran variedad de tipos de CMS los cuales pueden agruparse según el tipo de sitio que permiten gestionar. Dentro de los CMS que se ajustan a los portales se encuentran: PHP-Nuke, Joomla, Drupal, Plone, e-107. Se hace énfasis en Drupal por las características y ventajas que presenta, las cuales se describen posteriormente, otra de las razones que justifican su selección es por ser el utilizado por el grupo de desarrollo de portales, el cual se encargará de desarrollar la Intranet de las Residencias de Protocolo.

### **1.3.2 CMS Drupal.**

Drupal originalmente fue escrito por Dries Buytaert, el cual en el año 2001 decide liberar su código bajo la licencia GNU/GPL, garantizando así el crecimiento del proyecto y que siempre fuese libre.

Es un Sistema de Gestión de Contenidos que permite publicar artículos, imágenes u otros archivos. Dentro de los principales servicios que presta se encuentran: los foros, encuestas, votaciones, blogs y administración de usuarios y permisos.

Es un sistema dinámico, que en lugar de almacenar sus contenidos en archivos estáticos, los almacena en una base de datos, utilizando un entorno web que viene incluido en el producto permitiendo la edición de los mismos.

Dentro de las principales características que posee Drupal se encuentran:

- Está construido de forma modular, haciendo posible la incorporación de nuevas funcionalidades o módulos que permiten lograr una aplicación adaptable a las necesidades que se requieran.
- Se caracteriza por su rendimiento y escalabilidad, añadiendo un mecanismo de control de congestión que permite habilitar y deshabilitar determinados módulos o bloques dependiendo de la carga del servidor. Los cambios de configuración de los módulos, así como la eliminación e incorporación de ellos no influye en un mal funcionamiento de Drupal, debido al que sistema logra adaptarse a estas circunstancias cambiantes y lograr un rendimiento favorable. Además posee un mecanismo de eliminación del método de búsqueda en bases de datos utilizando la cache para reducir la carga en el servidor.

- Su flexibilidad radica en las facilidades que ofrece relacionadas con el diseño, para lograr esto separa los contenidos de la presentación, lo que permite controlar o cambiar con mayor facilidad el diseño del sitio.
- Es multiplataforma, siendo compatible con GNU/Linux, Windows, Mac OSX, BSD, y Solaris. Tiene soporte tanto para Apache como Internet Information Server (IIS).
- Posee independencia de la base de datos, se puede utilizar tanto MySQL como PostgreSQL, así como otras bases de datos que son propietarias.

Las ventajas antes mencionadas unido a la relevancia que ha alcanzado Drupal en las comunidades de usuarios hacen que sea uno de los CMS más destacados en la actualidad, lo cual es una realidad ya que la cantidad de usuarios de Drupal cada día va en aumento, a esto se le suma los innumerables aportes que realizan terceros con el fin de agregarle nuevos módulos.

#### **1.4 Lenguaje de programación.**

Después de elegir a Drupal como Sistema de Administración de Contenidos para el desarrollo de la Intranet de las Residencias de Protocolo, se procede a estudiar los principales lenguajes que utiliza Drupal en su código.

##### **1.4.1 PHP (PHP Hypertext Pre-processor)**

PHP fue creado originalmente por Rasmus Lerdorf en 1994, pero actualmente la implementación principal de PHP es producida por “The PHP Group”. Está publicado bajo la “Licencia PHP”, considerada esta licencia como software libre.

PHP es un lenguaje interpretado de amplio uso a nivel mundial en aplicaciones web, cuyo código puede ser embebido dentro de código HTML. Se ejecuta en servidores web, tomando el código en PHP como su entrada y creando páginas web como salida.

Son diversas las ventajas que proporciona PHP, dentro de ellas se puede mencionar, que es un lenguaje multiplataforma siendo compatible con GNU/Linux, Windows y Mac OSX, posee amplia conectividad con los gestores de base de datos existentes como MySQL, Postgres, Oracle, ODBC, DB2 y Microsoft SQL Server además de ser un lenguaje orientado a objetos desde su última versión PHP 5.

PHP ha servido en gran medida para el desarrollo de CMS y Frameworks así como otras tantas aplicaciones. Actualmente son varias las herramientas que utilizan este lenguaje para producir software, destacándose el

Zend Studio, Eclipse, Quanta, NetBeans entre otros, algunos comerciales y otros gratuitos. En relación con los servidores web y en especial con Apache posee un módulo para interactuar con este así como con otros.

### **1.4.2 HTML (HyperText Markup Language)**

Fue desarrollado originalmente por Tim Berners-Lee hacia la década de los 80, pero no fue hasta noviembre de 1995 que se llega a estandarizar con el HTML 2.0. A partir de 1996, los estándares de HTML fueron publicados por otra compañía de estandarización llamada W3C (World Wide Web Consortium), la cual lanza en 1998 la versión 4.0 y posteriormente la versión 4.01 que corrige algunos errores de la versión anterior.

Este organismo se dedicó a partir de ese entonces a estandarizar el XHTML despreocupándose de la continuidad de HTML 4.01, por lo que surge la compañía WHATWG (Web Hypertext Application Technology Working Group) de la unión de grandes empresas como Apple, Mozilla y Opera, con vistas a trabajar en el futuro estándar de HTML 5.0. A consecuencia de esto la W3C volvió a dedicar parte de su tiempo a la evolución del último estándar HTML 4.0.

La funcionalidad de este lenguaje está dada por instrucciones en un código especial compuesto por las llamadas “tags”, estas etiquetas se le pasan al navegador, éste las interpreta y le muestra al lector la página web.

Es un lenguaje muy sencillo que está hecho para describir hipertexto, en otras palabras, presentar texto de forma estructurada y agradable, facilitando la inserción de enlaces y multimedia. En el proceso de estructuración del contenido dentro del HTML se trabaja con: títulos, párrafos de texto normal, enumeraciones, definiciones y citas.

HTML 4.0 trae incorporado los mecanismos para hojas de estilo, ejecución de scripts, marcos, tablas con mayor funcionalidad y numerosas facilidades con los formularios, además de ofrecer mejoras de accesibilidad para personas con discapacidades.

Una de las principales ventajas que proporciona este lenguaje es poder embeber código PHP dentro de su código, además de otros lenguajes que poco a poco se han ido sumando a esta facilidad.

HTML es la base estructural que utiliza la World Wide Web para crear páginas web ya sean desde las más sencillas hasta las más complejas.

### 1.4.3 CSS (Cascading Style Sheets)

Hacia finales de 1996 la W3C publicó la primera versión de CSS conocida por CSS1, en 1998 se publica la versión CSS2, y a partir de ese entonces dicha compañía sigue trabajando en la versión 3. Actualmente los navegadores usan una versión CSS 2.1, que no es más que una versión corregida del CSS2.

CSS es un lenguaje formal que define la presentación estética de un documento estructurado y escrito en HTML. Este lenguaje separa el contenido de la presentación, creando estilos que definen la forma en que se muestran los elementos HTML y XML.

Algunas de las principales ventajas que ofrecen las hojas de estilos son:

- Poder controlar el estilo y formato de varias páginas Web al mismo tiempo, lo que convierte al HTML en un documento versátil y liviano.
- Hace que los códigos HTML resulten más fáciles de entender, lo cual está dado a la definición de los estilos en ficheros separados.
- Las páginas se cargan con mayor rapidez debido a la existencia de poco código HTML en la página.

### 1.4.4 JavaScript.

Este lenguaje nace a raíz de la unión de las compañías de Netscape y Sun Microsystems, donde posteriormente se presentó a ECMA (European Computer Manufacturers Association) con el objetivo de normalizarlo, resultando finalmente el EcmaScript.

JavaScript es un lenguaje de programación interpretado, no es necesario compilar los programas para poderlos ejecutar. Los programas escritos en este lenguaje se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios.

Las principales ventajas de JavaScript son:

- Su código se ejecuta del lado del cliente evitando sobrecarga en el servidor.
- Se asemeja a otros lenguajes de programación potentes como C++ o Java, resultando fácil de aprender para todo aquel que tenga experiencias con los mismos.
- Se complementa muy bien con HTML y PHP en la programación de páginas web.
- Es seguro y fiable.

## **1.5 Entorno de Desarrollo Integrado (IDE).**

Tras la necesidad que se presenta de utilizar PHP como lenguaje de programación, se selecciona como IDE de desarrollo el Zend Studio y el editor web Quanta Plus, el primero es una herramienta comercial y el segundo gratuito. En la actualidad son las más utilizadas por los desarrolladores en lenguaje PHP, junto a Eclipse que últimamente va tomando su espacio dentro de los programadores, pero que en este caso no será utilizada.

### **1.5.1 Quanta Plus.**

Quanta Plus es un editor web desarrollado especialmente para el escritorio KDE. Soporta varios lenguajes como: HTML, Javascript, CSS, PHP, SQL, XML, ColdFusion, Perl, Zope o C++, siendo en la actualidad uno de los editores más utilizados en el desarrollo de aplicaciones web en Linux.

Algunas de las funcionalidades que presenta este editor web son: coloreado de sintaxis para todos los lenguajes que soporta, presencia de cajas de diálogo contextuales para facilitar la elección de etiquetas HTML, un excelente navegador de directorios integrado para acceder fácilmente a los ficheros que se puedan necesitar, un panel completo para la pre visualización del resultado final y dentro de la propia aplicación se encuentra un validador de HTML.

Quanta Plus se está convirtiendo rápidamente en un editor maduro con gran cantidad de funcionalidades que lo impulsan a convertirse en un IDE.

### **1.5.2 Zend Studio.**

Zend Studio es un editor web orientado a la programación de páginas PHP con facilidades en la gestión de proyectos y depuración de código. Es un IDE multiplataforma que tiene compatibilidad con GNU/Linux, Windows y Mac OSX y está escrito en lenguaje Java.

Consta de dos partes importantes que agrupan funcionalidades: cliente y servidor. La primera contiene la interfaz de edición y ayuda, permitiendo además la depuración de scripts y la segunda ofrece una mayor potencia en la depuración, instalando o configurando el apache y el módulo PHP para trabajar juntos en la depuración del código.

Posee una ayuda contextual con todas las librerías de funciones del lenguaje PHP, asistiendo en todas las necesidades de código al programador, como nombres de funciones y parámetros que reciben.

Como todo editor web avanzado facilita diversas funcionalidades tales como: permitir editar varios archivos al mismo tiempo y moverse fácilmente entre ellos, marcar a qué elementos corresponden los inicios y cierres de las etiquetas, paréntesis o llaves, desplazarse al principio o final de una función e identificación automática del código.

Si se desea aumentar la productividad de los proyectos realizados en lenguaje PHP, Zend Studio puede contribuir en gran medida. Todas las opciones y funcionalidades que dispone están pensadas con acierto, razones que sustentan la selección del mismo para el desarrollo del trabajo.

## **1.6 Gestor de Base de Datos.**

Los principales gestores de base de datos que utiliza Drupal son: PostgreSQL y MySQL. Ambos se encuentran dentro de los más usados en la actualidad en el desarrollo de aplicaciones web.

### **1.6.1 PostgreSQL.**

Postgres surgió a partir del proyecto Ingres de la Universidad de Berkeley, después es desarrollado comercialmente por la Relational Technologies/Ingres Corporation, posteriormente este proyecto es retomado por Michael Stonebraker en Berkeley y finalmente en 1996 pasa a ser software libre bajo la licencia BSD (Berkeley Software Distribution) tomando el nombre de PostgreSQL.

Es un sistema de gestión de bases de datos objeto-relacional que aunque no es puramente orientado a objetos si posee características similares, tales como: la herencia, tipos de datos, funciones, restricciones, disparadores, reglas e integridad transaccional.

Otras características relevantes de PostgreSQL son: (QUIÑONES, 2007)

- Es altamente extensible siendo capaz de soportar operadores, funciones, métodos de acceso y tipos de datos definidos por los usuarios.
- Corre en casi todos los principales sistemas operativos: Linux, Unix, BSDs, Mac OS, Beos y Windows.
- Documentación muy bien organizada, pública y libre, con comentarios de los propios usuarios.
- Soporte nativo para los lenguajes más populares del medio: PHP, C, C++, Perl, Python.
- Soporte de todas las características de una base de datos profesional (triggers, store procedures, funciones, secuencias, relaciones, reglas, tipos de datos definidos por usuarios, vistas y vistas materializadas).

Actualmente PostgreSQL está ampliamente considerado como el sistema de bases de datos de código abierto más avanzado, esto está enmarcado por poseer muchas características que tradicionalmente sólo se podían ver en productos comerciales de alto calibre. (GONZALEZ, 2009)

A pesar de ser PostgreSQL un avanzado gestor de base de datos, en relación con la web no es considerable utilizarlo en aplicaciones que no contengan un alto grado de complejidad, ya que consume muchos recursos y en comparación con MySQL posee menos funciones en PHP. En este sentido se decide no utilizar a PostgreSQL como gestor de base de datos para Drupal en el desarrollo de la Intranet de las Residencias de Protocolo.

### **1.6.2 MySQL.**

MySQL es una idea original nacida de la empresa de código abierto MySQL AB, donde su objetivo principal era crear a MySQL cumpliendo con el estándar SQL y a la vez tratar de lograr una mayor velocidad, fiabilidad y usabilidad. Actualmente Sun Microsystems desarrolla MySQL bajo la licencia GNU/GPL y mantiene el desarrollo de una versión privativa.

MySQL es un sistema de gestión de bases de datos relacional con las siguientes características:

- Es multiplataforma.
- Sus funciones SQL están implementadas usando una librería altamente optimizada.
- Soporta gran cantidad de tipos de datos para las columnas.
- Alta velocidad al realizar operaciones, logrando que sea uno de los gestores con mejor rendimiento.
- Dispone de API's en gran cantidad de lenguajes (C, C++, Java, PHP).

En estos momentos MySQL es uno de los gestores de base de datos más usado en el mundo, preferentemente en las aplicaciones web, justificando esto su poco consumo de recursos, tanto de memoria como de CPU y por presentar una mejor integración con PHP.

Finalmente se decide utilizar MySQL como gestor de base de datos para Drupal, con el fin de alcanzar principalmente una alta velocidad en la red de las Residencias de Protocolo, aprovechando las ventajas que proporciona su uso en aplicaciones web.

### **1.6.3 Base de datos centralizadas y distribuidas.**

Ante la necesidad de usar un Sistema Gestor de Base de Datos surge la decisión de utilizar una base de datos centralizada o distribuida para la Intranet de las Residencias de Protocolo. Una base de datos

centralizada es una base de datos que está ubicada físicamente en un único lugar, donde casi todas las funciones relacionadas con la misma se llevan a cabo en una sola computadora. Una base de datos distribuida está compuesta por varios sistemas de bases de datos llamados nodos o localidades que se encuentran operando en sitios diferentes y están conectados entre sí, en cada uno de estos nodos se realizan transacciones locales y a la vez pueden realizarse transacciones globales hacia otros nodos conectados. Estas bases de datos distribuidas hacen posible que los datos estén disponibles en los lugares donde se necesiten con mayor frecuencia y posibilitan además que otros usuarios puedan acceder a las mismas.

Finalmente se decide usar una base de datos centralizada para las Residencias de Protocolo debido a que no existe un gran volumen de información, es posible y más fácil tenerla centralizada donde una misma persona sea la encargada de trabajar con ella y además la utilización de una base de datos distribuida incurriría en un gasto de recursos informáticos, materiales, monetarios y humanos tras la necesidad de agregar nuevos nodos.

### **1.7 Servidores dedicados y compartidos.**

Un servidor compartido contiene varias web alojadas en un mismo servidor, compartiendo capacidad de procesamiento, memoria RAM y espacio en el disco duro de la máquina. Un servidor dedicado aloja una única web, lo que ofrece como ventajas: privacidad y seguridad para la protección de los datos, rendimiento, potencia y velocidad con la utilización exclusiva de los recursos del servidor y un mejor control total sobre las aplicaciones instaladas en el mismo, siendo esta la solución idónea para las empresas e instituciones que dependen de su web y de la base de datos para el funcionamiento de la misma.

En el caso de las Residencias de Protocolo se decide utilizar un servidor compartido debido a que la información que se maneja en dicha institución no es muy compleja, donde existe poco tráfico en la red, validando con ello ser la solución más óptima.

### **1.8 Servidor Web.**

A grandes rasgos un servidor web es un programa que se encuentra ejecutándose continuamente en un ordenador, recibe las peticiones de los usuarios desde otros ordenadores y se encarga de dar respuesta enviando como resultado una página web o diferentes tipos de información de acuerdo a la petición solicitada.

Después de seleccionar a Drupal como Sistema de Administración de Contenidos para implementar la Intranet de las Residencias de Protocolo, el cual se encuentra implementado en código PHP y decidir además utilizar MySQL como su gestor de Base de Datos, se hace necesario seleccionar el servidor web que más se ajuste a estos requerimientos. Es por ello que se procede a usar el servidor web Apache.

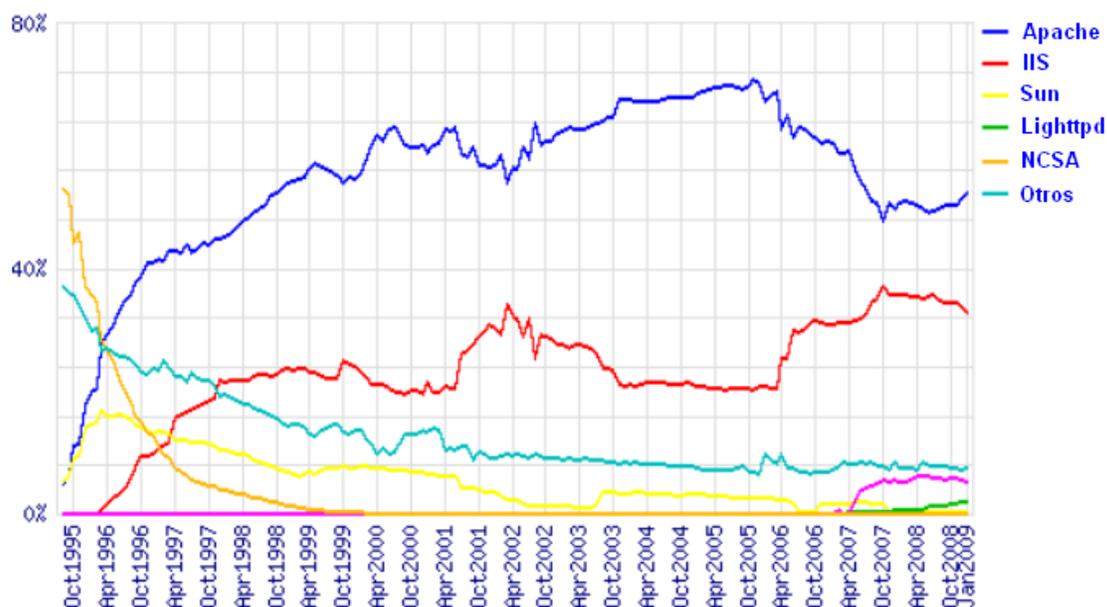
### 1.8.1 Apache.

Apache nace a raíz del código de National Center for Supercomputing Applications (NCSA), proyecto que se suspende en 1998 y que durante un tiempo formó parte del código inicial de Apache. Se encuentra bajo la licencia "Apache", la cual es descendiente de las BSD (Berkeley Software Distribution) y a su vez forma parte de las licencias de software libre.

Este servidor web está compuesto por un núcleo central y un conjunto de módulos, posee además una interfaz que permite administrar y decidir cuáles funcionalidades incluir, logrando así poder construir un servidor web personalizado.

Algunas de las principales ventajas que brinda Apache son:

- Es un servidor web potente, flexible y multiplataforma.
- Es altamente configurable y de diseño modular lo que ha contribuido a que muchos programadores ayuden a desarrollar diversidad de módulos, adaptables a él. Actualmente existen muchos módulos para Apache tanto para gran cantidad de lenguajes (incluido PHP), así como para una variedad de servicios web.
- Permite personalizar las respuestas en caso de posibles errores que puedan ocurrir en el servidor.
- Facilita la configuración de creación y gestión de logs, permitiendo tener un mayor control sobre lo que está sucediendo en el servidor.



**Figura 1: Estadística de utilización de los principales Servidores Web en todos los dominios de Internet (Agosto-1995 a Enero-2009).**

### **1.9 Metodologías de Desarrollo de Software.**

El impacto del software en nuestra sociedad y en la cultura continúa siendo profundo. Al mismo tiempo que crece su importancia, la comunidad del software trata continuamente de desarrollar tecnologías que hagan más sencillo, rápido y menos costosa la construcción de programas de computadora de alta calidad. La tecnología que comprende un proceso, un juego de métodos y un conjunto de herramientas se llama *ingeniería de software*. (PRESSMAN, 1998)

Piattini definió como metodología al "... conjunto de procedimientos, técnicas, herramientas, y un soporte documental que ayuda a los desarrolladores a realizar nuevo software". (CATALDI, Z, 2000)

Y Maddison define metodología como un conjunto de filosofías, fases, procedimientos, reglas, técnicas, herramientas, documentación y aspectos de formación para los desarrolladores de sistemas de información. (CATALDI, 2000)

Con el paso de los años estas metodologías se han venido desarrollando para el proceso de software, todas con sus ventajas, desventajas y utilidad en algunos tipos de proyectos y problemas, no existe una metodología de software universal, esta debe adecuarse a las características específicas de cada proyecto (recursos y equipo de desarrollo) que exigen que el proceso sea configurable.

A la hora de afrontar el desarrollo de este trabajo, se han evaluado las metodologías más usadas actualmente para elegir la que más se ajusta al mismo. Entre ellas se puede mencionar a *Extreme Programming (XP)* y *Rational Unified Process (RUP)*.

#### **1.9.1 Extreme Programming (XP)**

Extreme Programming se encuentra entre las llamadas Metodologías Ágiles, que surgieron como una alternativa a los procesos de desarrollo de software tradicionales.

Sus principios fundamentales son:

- Realimentación entre el cliente y el equipo de desarrollo.
- Comunicación fluida entre todos los participantes.
- Simplicidad en las soluciones implementadas.

Entre las principales características de XP que la diferencian de otras metodologías de desarrollo se encuentran:

- Es una metodología adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico.
- Es una de las metodologías de desarrollo de software con más éxito en la actualidad.
- Se utiliza en proyectos con equipo de desarrollos pequeños y con plazo de entrega corto.
- La metodología consiste en una programación rápida o extrema. Una particularidad es que tiene como miembro del equipo al usuario final. (CASTILLO, 2009)

**Esta metodología de desarrollo del software está dividida en 4 fases:**

1ª Fase: Planificación del proyecto.

En esta fase se describen las historias de usuarios las cuales son realizadas con la misma finalidad que los casos de usos pero con algunas diferencias. También se hace necesario la creación de un plan de publicaciones o "Release plan" para indicar las historias de usuarios correspondientes que serán publicadas en las versiones del programa. El proyecto se divide en iteraciones de corta duración y al comenzar cada iteración es donde se seleccionan las historias de usuario definidas anteriormente en el plan. El objetivo principal es mostrar una versión nueva a cada instante para mostrarle al cliente, saber lo que opina y luego seguir programando de forma que se mantenga la comunicación entre el cliente y los desarrolladores. Para lograr una planificación eficiente es necesario desarrollar reuniones diarias para que los desarrolladores expongan sus problemas, soluciones e ideas de forma conjunta.

2ª Fase: Diseño

Sugiere diseños simples, sencillos y libres de complejidad. Usar glosarios de términos y una correcta especificación de los nombres de métodos y clases que ayude a comprender el diseño y facilite sus posteriores ampliaciones y la reusabilidad del código. Propone además usar la técnica de refactorizar lo que significa mejorar y modificar la estructura y codificación de códigos ya creados sin alterar su funcionalidad.

3ª Fase: Codificación

Para la codificación se utilizan patrones de codificación ya creados y estándares.

El modelo de trabajo se realiza usando repositorios de códigos y modelos de desarrollo colectivo.

4ª Fase: Pruebas

Hay que someter a test a las distintas clases del sistema omitiendo los métodos más triviales.

### 1.9.2 Rational Unified Process (RUP)

Rational Unified Process se encuentra dentro de las metodologías robustas. Entre las características principales que la diferencian de otras metodologías especialmente de XP, se encuentra que la misma es adecuada para proyectos grandes y de alta calidad. Todo su ciclo de vida es iterativo e incremental, centrado en la arquitectura y dirigido por casos de usos. Utiliza como lenguaje de modelado el UML.

En su modelación RUP define como sus principales elementos (JACOBSON, 1999):

- Trabajadores (“quién”)

Precisa responsabilidades (rol) y habilidades de un individuo o grupo de individuos, sistema automatizado o máquina, que trabajan en conjunto como un equipo para realizar determinadas actividades o desarrollar determinados artefactos.

- Actividades (“cómo”)

Es una tarea que tiene un propósito claro, es realizada por un trabajador y manipula elementos.

- Artefactos (“qué”)

Pieza de información tangible que es creada, modificada y usada por los trabajadores al realizar actividades. Este puede ser además un modelo, un elemento de un modelo, o un documento.

- Flujo de actividades (“Cuándo”)

Secuencia de actividades realizadas por trabajadores y que produce un resultado de valor observable.

#### **Fases de la metodología RUP:**

Cada ciclo en RUP está compuesto por cuatro fases, y dentro de cada una de ellas los líderes o los desarrolladores pueden descomponer en iteraciones el trabajo. Cada fase termina con un hito y cada uno de estos se determina por la disponibilidad de un conjunto de artefactos.

*Fases que propone RUP (JACOBSON, 1999):*

#### 1ª Fase: Inicio

En esta fase se establece un acuerdo entre todos los interesados acerca de los objetivos del proyecto. Esta fase es significativamente primaria para el desarrollo del software, pues se asegura de identificar los riesgos relacionados con el negocio y requerimientos. Para proyectos de mejora de software existente esta fase es más breve y se centra en asegurar que vale la pena y es posible desarrollar el proyecto.

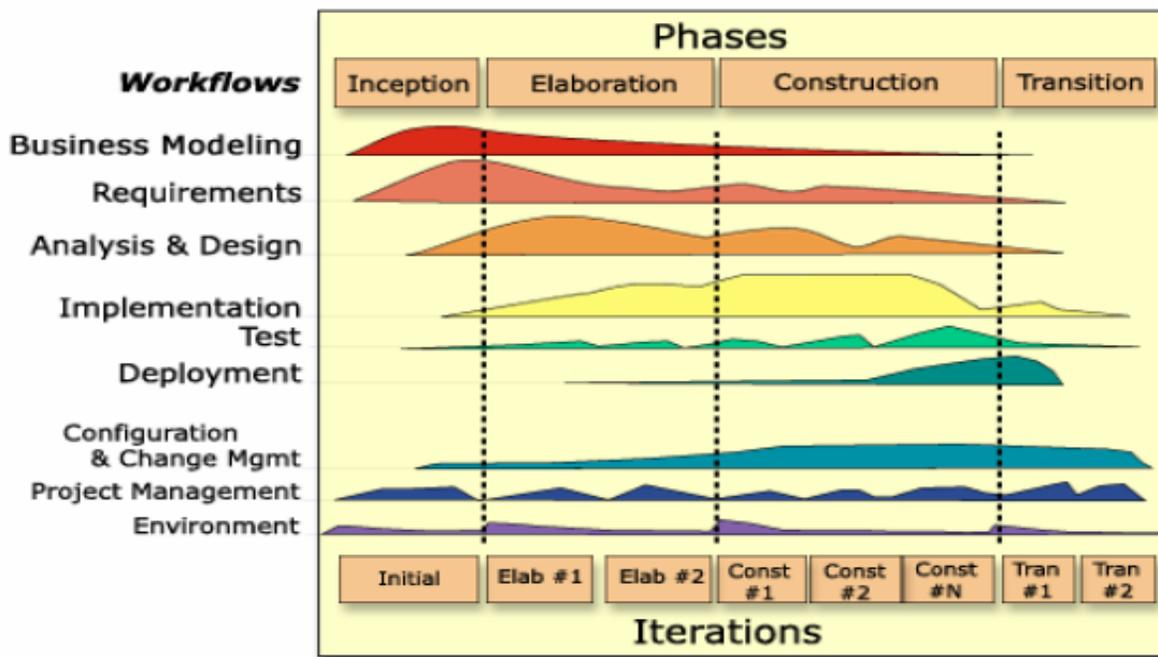
#### 2ª Fase: Elaboración

En la fase de Elaboración se establece la arquitectura base del sistema para proveer bases estables para el esfuerzo de diseño e implementación en la siguiente fase. La arquitectura debe abarcar todas las consideraciones de mayor importancia de los requerimientos y una evaluación del riesgo.

**3ª Fase: Construcción**

El objetivo general de la fase de construcción es obtener un producto listo para ser distribuido como versión beta y ser sometido a pruebas. Esta fase es la más compleja de todas pues se necesita de más personal y de más tiempo, por esto además requiere de más iteraciones que las fases anteriores.

**4ª Fase: Transición:** Esta fase se centra en implantar el producto en su entorno de operación. La forma en que el proyecto lleva a cabo este objetivo varía con la naturaleza de la relación del producto con su mercado. Además define nueve flujos de trabajo, los seis primeros son ingenieriles y los restantes son de soporte.



**Figura 2: Fases y flujos de trabajo de RUP.**

**1.10 Herramienta para la Modelación Visual del Sistema.**

Las herramientas CASE (Computer-Aided Software Engineering) son generalmente aplicadas a cualquier sistema o colección de herramientas que ayudan a automatizar el proceso de diseño y desarrollo del

software. Impiden a los programadores tratar tan directamente con el hardware, permitiéndoles trabajar con un alto nivel de abstracción en la definición del sistema.

Existen diferentes tipos de sistemas CASE, pero en este caso la investigación se enfoca en las herramientas de diseño, las cuales auxilian a grandes equipos de ingenieros en la especificación de sistemas de software y ayudan a automatizar la escritura de arquitecturas y documentación e integrar automáticamente esas piezas generadas en el IDE del desarrollador.

Muchas herramientas CASE utilizan el Lenguaje de Modelado Unificado (UML) posibilitando a los ingenieros de software la habilidad para crear especificaciones de sistemas que pueden ser relativamente fácil de traducir en código sostenible y que funcione. Dentro de ellas se destaca Visual Paradigm, herramienta CASE que se utiliza actualmente en algunas facultades de la Universidad de las Ciencias Informáticas por ser una herramienta que puede ser utilizada en la producción de software libre, principal motivo por lo que se selecciona para el desarrollo de la intranet.

### **1.10.1 Visual Paradigm.**

Para fomentar la selección de la herramienta CASE: Visual Paradigm, se mencionan a continuación un grupo de características y ventajas que hacen de la misma la más adecuada para el desarrollo del trabajo:

- Es una herramienta muy potente, fácil de usar e instalar.
- Posee un entorno de creación de diagramas para UML.
- Permite revertir el código fuente a modelos UML.
- Facilita generar código fuente desde los diagramas.
- Disponibilidad en múltiples plataformas.
- Uso de un lenguaje estándar común para todo el equipo de desarrollo que facilita la comunicación.
- Incluye los objetos más recientes de UML, además de diagramas de casos de uso, de clases, actividades, reversa instantánea para Java, C++, DotNet Exe/dll, XML, XML Schema, y Corba IDL.
- Ofrece soporte para Rational Rose.
- Integración con Microsoft Visio.
- Permite generar reportes y documentación en HTML/PDF.

### **1.11 Conclusiones.**

Después del estudio realizado de los diferentes aspectos tratados en este capítulo quedan sentadas las bases para el desarrollo del trabajo. Se decide como solución al problema planteado el diseño de una intranet que sea adaptable a las necesidades y exigencias de las Casas de Protocolo, utilizando para ello:

- Sistema de Administración de Contenidos (CMS): Drupal.
- Lenguaje de Programación: PHP.
- Entorno de Desarrollo Integrado (IDE): Zend Studio y Quanta Plus.
- Gestor de Base de Datos: MySql.
- Servidor Web: Apache.

Se seleccionó además, RUP como metodología de desarrollo para el modelado del sistema con UML como lenguaje de representación visual y Visual Paradigm como herramienta para la modelación visual del sistema.

## Capítulo 2: Caracterización del Sistema.

### 2.1 Introducción.

En el presente capítulo se abordan las características que el sistema debe tener, se describe el problema, se hace un análisis del proceso que será objeto de automatización, la información que se maneja, se presenta la propuesta del sistema, se detallan los requisitos funcionales y no funcionales que debe tener el mismo, así como la realización del modelo conceptual.

Se definirán además los actores y casos de uso del sistema, las relaciones que existen entre ambos representados en el diagrama de casos de uso del sistema y finalmente se describirán los casos de uso teniendo en cuenta la acción del actor y la respuesta del sistema.

### 2.2 Problema

Las Residencias de Protocolo del Consejo de Estado de la República de Cuba ofrecen sus servicios a grandes personalidades tanto nacionales como internacionales, es por ello que requieren que los mismos sean de excelencia. La intranet que se encuentra actualmente en las Casas de Protocolo no representa una solución acorde a las necesidades de la misma, atendiendo principalmente a parámetros tan importantes como la utilización de un diseño adecuado que sea el reflejo de la institución, lo cual está enmarcado en:

- Mal uso de los colores.
- La pantalla inicial le resta elegancia al sitio.
- Muchas de las imágenes utilizadas no representan la realidad del lugar.
- La estructuración y distribución de los contenidos no es la más óptima.
- Falta de enriquecimiento en las galerías de imágenes.
- Ineficiencia del sistema de encuesta.

Todas estas deficiencias descritas anteriormente traen consigo la insatisfacción de los huéspedes de las Residencias de Protocolo y de la dirección de la misma.

### **2.3 Objeto de Automatización**

El proceso que serán objetos de automatización es la gestión de la información referente a las Residencias de Protocolo del Consejo de Estado de la República de Cuba.

### **2.4 Información que se maneja**

La información manejada está enmarcada en: historia de las Casas de Protocolo, datos relacionados con las residencias, en qué año fueron creadas, a quién perteneció, la ubicación de las mismas, las obras de arte que decoran dichas residencias con algunos datos de sus autores, la información de los servicios que se ofertan a los huéspedes y el sitio histórico resaltando la figura de Celia Sánchez Manduley.

### **2.5 Propuesta de Sistema**

Con el fin de cumplir con los objetivos y requerimientos planteados en este trabajo, el sistema propuesto debe poseer dos módulos, un módulo administrativo donde se realizarán las actualizaciones de los contenidos y otro que será el punto de acceso a toda la información referente a la Intranet.

Todos los huéspedes tendrán acceso a dicha Intranet desde las residencias, los mismos podrán navegar e interactuar con las funcionalidades que la misma brinda, mientras que la administración de los contenidos se realizará solamente por el administrador o alguna persona autorizada por este que será el encargado de mantener el sistema disponible y actualizado.

### **2.6 Diagrama de clases del Modelo de Dominio**

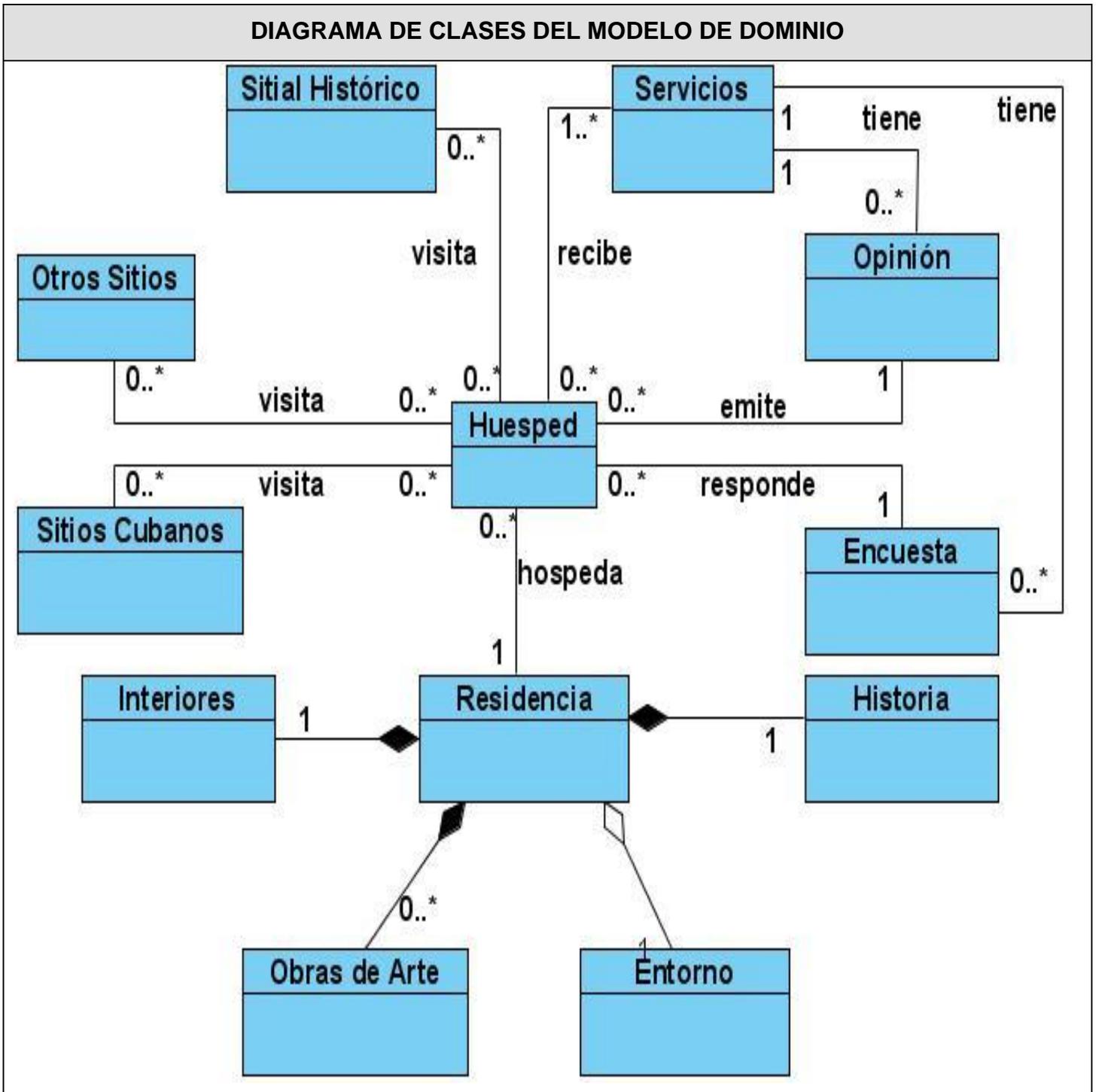
Después de realizado un estudio de los procesos que se van a efectuar, se llegó a la conclusión, que el negocio estudiado tiene muy bajo nivel de estructuración, donde los flujos de información se encuentran difusos, y cuando se desea realizar una actividad, múltiples personas intervienen en la misma, lo que implica un solapamiento de responsabilidades, además es difícil establecer las reglas de funcionamiento, por lo que se propone realizar un modelo de dominio.

Se realizará el modelo de dominio, porque permite de manera visual mostrar al usuario los principales conceptos que se manejan en el dominio del sistema en desarrollo. Esto ayuda a los usuarios, clientes, desarrolladores e interesados a utilizar un vocabulario común para poder entender el contexto en que se enmarca el sistema.

Es necesario tener un vasto conocimiento de cómo debe funcionar el proceso en cuestión, para poder capturar correctamente los requisitos y así poder construir un sistema con las características que el cliente desee. Éste modelo va a contribuir posteriormente a identificar algunas clases que se utilizarán en el sistema. Primeramente se identificarán todos los conceptos que se utilizarán en el diagrama, mediante un **glosario de términos**:

- **Residencia:** Lugar donde estará alojado el huésped.
- **Entorno:** Ambiente que rodea las residencias.
- **Obras de Arte:** Obras que decoran la residencia.
- **Historial:** Historia de las Residencias de Protocolo.
- **Interiores:** Interiores de las residencias.
- **Huésped:** Visitante que estará hospedado en la residencia un tiempo determinado.
- **Servicios:** Servicios que serán prestados a los huéspedes.
- **Encuesta:** Encuesta que llena el huésped en su estancia en el recinto.
- **Opinión:** Opinión emitida por el huésped sobre algún servicio.
- **Sitio Histórico:** Información sobre el sitio histórico.
- **Sitios Cubanos:** Sitios cubanos que puede visitar el huésped.
- **Otros Sitios:** Otros sitios que pueden ser de interés para el huésped.

A continuación se muestra el Modelo de Dominio correspondiente:



## **2.7 Especificación de los Requisitos del Software.**

El flujo de trabajo de requerimientos es uno de los más importantes, porque en él se establece qué es lo que tiene que hacer exactamente el sistema que se construya. En esta línea los requisitos son el contrato que se debe cumplir, de modo que los usuarios finales tienen que comprender y aceptar los requisitos que se especifiquen. Se dividen en dos grupos: los requisitos funcionales y los requisitos no funcionales.

### **2.7.1 Requerimientos Funcionales**

Una vez descrito el modelo de dominio, para poder identificar que debe hacer el sistema y entender su funcionamiento, es fundamental conocer los requisitos funcionales que el sistema debe cumplir.

Los requerimientos funcionales deben comprenderlo tanto los desarrolladores como los usuarios, a continuación se relacionan los que debe cumplir la aplicación a desarrollar:

#### **R1. Seleccionar Idioma.**

El sistema debe permitir al visitante seleccionar en qué idioma desea que se muestre la información. El idioma puede ser inglés ó español.

#### **R2. Contactar Administrador.**

El sistema debe ofrecer al visitante un vínculo a la dirección de correo electrónico del administrador del sitio.

#### **R3 Visualizar información**

El sistema debe mostrarle al visitante toda la información disponible.

##### **R3.1. Visualizar información general de la residencia.**

El sistema debe mostrar al visitante la información referente a la residencia en la cual se encuentra hospedado:

- Obras de arte.
- Interiores de la residencia.
- Datos de la residencia

##### **R3.2 Visualizar información general de los servicios que brinda la residencia.**

El sistema debe mostrar al visitante la información referente a los servicios de:

- Transporte
- Aeropuerto
- Tintorería
- Gastronomía

### **R3.3 Visualizar la información de enlaces a sitios cubanos y otros sitios.**

El sistema debe mostrar al visitante un listado de sitios cubanos organizados por categorías, así como de otros sitios.

#### **Categorías de los Sitios Cubanos:**

- Salud.
- Prensa nacional.
- Turismo.
- Cultura.
- Gobierno.
- Educación.
- Ciencia.
- Eventos.
- Deporte.

### **R3.4 Visualizar la información del Sital Histórico.**

El sistema debe mostrar al visitante la información referente al Sital Histórico.

### **R3.5. Visualizar entorno.**

El sistema debe mostrar al visitante imágenes del entorno que rodea la residencia.

### **R3.6 Visualizar Portada**

El sistema debe mostrar al visitante breve información de las casas de protocolo, cuando surgieron, características generales de las mismas y otros datos de interés.

## **R4 Emitir Opinión.**

El sistema debe ofrecerle al visitante la posibilidad de emitir su opinión sobre los servicios que recibe en la residencia.

### **R4.1 Ofrecer opinión sobre los servicios que brinda la residencia.**

El sistema debe permitir al visitante ofrecer su opinión por cada servicio que brinda la residencia.

### **R4.2 Contestar encuesta sobre los servicios que brinda la residencia.**

El sistema debe permitir al visitante contestar una encuesta por cada servicio que brinda la residencia.

## **R5 Autenticar usuario.**

El sistema debe permitir al administrador autenticarse.

## **R6 Crear contenido.**

El sistema de brindar al administrador poder agregar un contenido de un tipo de contenido existente.

**R6.1 Añadir residencia.**

El sistema debe brindar al administrador la posibilidad de agregar nuevas residencias.

**R6.2 Añadir obras de arte.**

El sistema debe brindar al administrador la posibilidad de agregar obras de arte a las residencias.

**R6.3 Añadir bebidas.**

El sistema debe brindar al administrador la opción de agregar nuevas bebidas.

**R6.4 Añadir sitios.**

El sistema debe permitir al administrador agregar nuevos sitios a las diferentes categorías.

**R7 Gestionar contenidos.**

El sistema debe permitir al administrador gestionar los contenidos (residencia, obras de arte, bebidas y sitios).

**R7.1 Editar contenidos.**

El sistema debe permitir al administrador editar los contenidos (residencia, obras de arte, bebidas y sitios).

**R7.2 Eliminar contenidos.**

El sistema debe permitir al administrador eliminar los contenidos (residencia, obras de arte, bebidas y sitios).

**R8 Actualizar información de los servicios.**

El sistema debe brindar al administrador la posibilidad de actualizar la información referente a los servicios.

**R8.1 Actualizar información del servicio de transporte.**

El sistema debe brindar al administrador la posibilidad de actualizar la información referente al servicio de transporte.

**R8.2 Actualizar información del servicio de aeropuerto.**

El sistema debe brindar al administrador la posibilidad de actualizar la información referente al servicio de aeropuerto.

**R8.3 Actualizar información del servicio de tintorería.**

El sistema debe brindar al administrador la posibilidad de actualizar la información referente al servicio de tintorería.

**R8.4 Actualizar información del servicio de gastronomía.**

El sistema debe brindar al administrador la posibilidad de actualizar la información referente al servicio de gastronomía.

**R9 Actualizar la carta menú.**

El sistema debe brindar al administrador la posibilidad de actualizar la carta menú.

**R10 Gestionar encuestas sobre los servicios que brinda la residencia.**

El sistema debe permitir al administrador gestionar las encuestas para cada servicio que brinda la residencia.

**R10.1 Crear encuestas.**

El sistema debe permitir al administrador crear encuestas sobre los servicios que brinda la residencia.

**R10.2 Editar encuestas.**

El sistema debe permitir al administrador editar las encuestas sobre los servicios que brinda la residencia.

**R10.3 Eliminar encuestas.**

El sistema debe permitir al administrador eliminar encuestas sobre los servicios que brinda la residencia.

**R11 Administrar opiniones.**

El sistema debe brindar al administrador la posibilidad gestionar las opiniones emitidas por los visitantes.

**R11.1 Revisar opiniones.**

El sistema debe permitir al administrador la posibilidad de revisar las opiniones emitidas por los visitantes.

**R11.2 Eliminar opiniones.**

El sistema debe permitir al administrador la posibilidad de eliminar opiniones.

**R12 Actualizar Entorno.**

El sistema debe permitir al administrador actualizar las imágenes e información del entorno.

## **2.7.2 Requerimientos No Funcionales**

Una vez analizado las funcionalidades que el sistema debe cumplir se hace necesario analizar las propiedades que el producto software debe tener, de lo cual se deriva: las restricciones del entorno o de la implementación, rendimiento, dependencias de la plataforma, facilidad de mantenimiento, entre otras. A esto se le conoce como requisitos no funcionales.

Entre los requerimientos no funcionales del sistema propuesto se encuentran:

### **Usabilidad**

- El sistema podrá ser usado de forma fácil por cualquier persona, con pocos conocimientos de computación.

### **Confiabilidad**

- La información contenida en el sistema debe ser totalmente confiable.
- Deben implementarse mecanismos para garantizar la respuesta ante posibles fallos lo más rápido posible.

### **Rendimiento**

El sistema demora en una transición aproximadamente 3 segundos como máximo, por detrás de la aplicación Web se realizarán consultas a bases de datos y validaciones de formularios que contengan información, programación en el cliente y en el servidor por lo que se requiere de cierto software y hardware para cumplir con los tiempos de transición de la interfaz de usuario.

### **Recursos tecnológicos requeridos:**

#### Software:

- Servidor Web: Apache 2.0.
- Intérprete de aplicaciones Web (Mozilla Firefox ó Internet Explorer).
- Sistema Gestor de Bases de Datos: MYSQL 5,0.
- Lenguaje de programación: PHP 5.

#### Hardware:

- 512 Mb de memoria RAM
- 20 Gb de Disco duro.

### **Seguridad**

Garantizar que la información sea editada únicamente por las personas que tienen permisos para realizar esta actividad.

- Protección contra acciones no autorizadas o que puedan afectar la integridad de los datos.

### **Interfaz**

- El producto debe ser legible y con colores de la entidad
- Diseño orientado a llamar la atención del usuario y con una navegación sencilla.
- Construcción de enlaces rápidos o anclas para los documentos muy largos.

## **2.8 Definición de los Casos de Uso**

### **2.8.1 Actores del Sistema**

**Tabla 1: Definición de los actores del sistema.**

Actor	Descripción
Administrador	Es el encargado de gestionar toda la información del sistema.
Visitante	Es el encargado de seleccionar el idioma en que se mostrará el contenido de la Intranet, establecer contacto con la entidad, realizar encuestas y ofrecer su opinión y podrá visualizar todo el contenido de la misma.

### 2.8.2 Casos de uso del sistema.

Los casos de uso del sistema que aparecen a continuación tienen como objetivo satisfacer los requisitos funcionales descritos con anterioridad.

**Tabla 2: CU Seleccionar Idioma.**

CU- 1	Seleccionar idioma.
Actor	Visitante
Descripción	El visitante selecciona el idioma con el cual desea que se muestren los contenidos del sitio.
Referencia	R1

**Tabla 3: CU Contactar Administrador.**

CU- 2	Contactar administrador
Actor	Visitante
Descripción	El visitante decide contactar al administrador del sitio.
Referencia	R2

**Tabla 4: CU Visualizar Información.**

CU- 3	Visualizar Información
Actor	Visitante
Descripción	El visitante solicita visualizar una información en el sitio.
Referencia	R3, R3.1, R3.2, R3.3, R3.4, R3.5, R3.6

**Tabla 5: CU Emitir Opinión.**

CU- 4	Emitir Opinión
Actor	Visitante
Descripción	El visitante decide emitir una opinión o llenar una encuesta.
Referencia	R4, R4.1, R4.2

**Tabla 6: CU Autenticar Administrador.**

CU- 5	Autenticar administrador.
Actor	Administrador
Descripción	El administrador decide autenticarse en el sitio.
Referencia	R5

**Tabla 7: CU Crear Contenido.**

CU- 6	Crear contenido
Actor	Administrador
Descripción	El administrador decide crear un contenido.
Referencia	R6, R6.1, R6.2, R6.3, R6.4

**Tabla 8: CU Gestionar Contenido.**

CU- 7	Gestionar contenido
Actor	Administrador
Descripción	El administrador decide editar o eliminar contenido.
Referencia	R7, R7.1, R7.2

**Tabla 9: CU Actualizar información de los servicios.**

CU- 8	Actualizar información de los servicios
Actor	Administrador
Descripción	El administrador decide actualizar la información de los servicios de gastronomía, transporte, aeropuerto y tintorería.
Referencia	R8, R8.1, R8.2, R8.3, R8.4

**Tabla 10: CU Actualizar carta menú.**

CU- 9	Actualizar carta menú
Actor	Administrador
Descripción	El administrador decide actualizar la carta menú.
Referencia	R9

**Tabla 11: CU Gestionar encuestas.**

CU- 10	Gestionar encuestas
Actor	Administrador
Descripción	El administrador decide crear, actualizar o eliminar una encuesta.
Referencia	R10, R10.1, R10.2, R10.3

**Tabla 12: CU Administrar opiniones.**

CU- 11	Administrar opiniones
Actor	Administrador.
Descripción	El administrador decide revisar o eliminar las opiniones ofrecidas por los visitantes.
Referencia	R11, R11.1, R11.2

**Tabla 13: CU Actualizar entorno.**

CU- 12	Actualizar entorno
Actor	Administrador
Descripción	El administrador decide actualizar el entorno.
Referencia	R12

### 2.8.3 Diagrama de casos de uso del sistema.

En el siguiente diagrama se representa la relación entre los actores y los casos de uso del sistema. Especificar para el mayor entendimiento del mismo que se realizó un CRUD Parcial (patrón de caso de uso) en el CU: Gestionar contenido dividiendo el mismo en dos CU, el CU: Gestionar Contenido donde se editan y eliminan los contenidos y el CU: Crear contenido, donde se crean los contenidos.



**2.8.4 Descripción de los casos de uso del sistema.**

A continuación se describen los casos de uso del sistema.

**Tabla 14: Seleccionar idioma**

<b>Caso de Uso:</b>	Seleccionar idioma.
---------------------	---------------------

<b>Actores:</b>	Visitante
<b>Resumen:</b>	El caso de uso se inicia cuando el visitante decide cambiar el idioma, selecciona el idioma deseado (inglés o español) y el sistema muestra los contenidos en el idioma seleccionado, finalizando así el caso de uso.
<b>Relación con otros CU</b>	
<b>Precondiciones:</b>	
<b>Referencias:</b>	R1
<b>Prioridad</b>	Secundario
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. El visitante selecciona el vínculo para cambiar el idioma.	1.1. El sistema le muestra el contenido de la intranet en el idioma que el usuario seleccionó. Se termina el caso de uso.
<b>Poscondiciones</b>	Se cambia el idioma del sitio.

**Tabla 15: Contactar administrador.**

<b>Caso de Uso:</b>	Contactar administrador.
<b>Actores:</b>	Visitante.
<b>Resumen:</b>	El caso de uso se inicia cuando el visitante decide contactar al administrador del sitio y selecciona el vínculo de contacto. El sistema lo envía a un cliente de correo con la dirección de correo predeterminada, el visitante llena los campos, finalizando así el caso de uso.

<b>Relación con otros CU</b>	
<b>Precondiciones:</b>	
<b>Referencias:</b>	R2
<b>Prioridad</b>	Secundario.
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. El visitante accede al vínculo de contactar.	1.1 El sistema muestra un cliente de correo con la dirección predeterminada para que el visitante pueda enviar el mensaje al administrador del sitio. terminando así el caso de uso
2. El visitante llena los campos y selecciona la opción de enviar.	El sistema procesa el mensaje y lo envía, terminando así el caso de uso.
<b>Poscondiciones</b>	Se envía un mensaje al administrador del sitio.

**Tabla 16: Visualizar información**

<b>Caso de Uso:</b>	Visualizar la información.
<b>Actores:</b>	Visitante.
<b>Resumen:</b>	El caso de uso se inicia cuando el visitante desea visualizar la información sobre las residencias, los servicios, enlaces a sitios cubanos y otros sitios, entorno, sitio histórico ó información relacionada con las Residencias de Protocolo, el sistema muestra la información requerida por el visitante y finaliza así el caso de uso.

<b>Relación con otros CU</b>	
<b>Precondiciones:</b>	
<b>Referencias:</b>	R3, R3.1, R3.2, R3.3, R3.4, R3.5, R3.6.
<b>Prioridad</b>	Crítico
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. El visitante accede a la opción que desea visualizar.	<p>1.1 El sistema le muestra la información solicitada.</p> <p>a) Si desea visualizar información sobre la residencia, ir a la Sección “Residencia”.</p> <p>b) Si desea visualizar información sobre los servicios, ir a la Sección “Servicios”.</p> <p>c) Si desea visualizar información sobre los sitios web cubanos y otros sitios, ir a la Sección “Enlaces”</p> <p>d) Si desea visualizar información sobre el sitio histórico, ir a la Sección “Sitio Histórico”.</p> <p>e) Si desea visualizar información sobre el entorno, ir a la Sección “Entorno”.</p>

	f) Si desea visualizar información sobre la historia de las Casas de Protocolo, ir a la Sección “Portada”.
<b>Flujo Normal de Eventos</b>	
<b>Sección “Residencia”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
2. El visitante accede al menú residencia.	<p>2.1 El sistema le muestra un submenú con los vínculos: Datos de la Residencia, Interior y Obras de Artes.</p> <p>a) Si desea visualizar información sobre datos de la residencia, ir a la Sub-Sección “Datos de la Residencia” de la Sección “Residencia”</p> <p>b) Si desea visualizar información sobre interiores, ir a la Sub-Sección “Interior” de la Sección “Residencia”</p> <p>c) Si desea visualizar información sobre las obras de arte, ir a la Sub-Sección “Obras de Arte” de la Sección “Residencia”</p>
<b>Sub-Sección “Datos de la Residencia” de la Sección “Residencia”</b>	

Acción del Actor		Respuesta del Sistema
3. El visitante selecciona el submenú Datos de la Residencia.		3.1. El sistema le muestra la información relacionada con la residencia. Se termina el caso de uso.
<b>Poscondiciones</b>		
<b>Sub-Sección “Interior” de la Sección “Residencia”</b>		
Acción del Actor		Respuesta del Sistema
3. El visitante selecciona el submenú Interior.		3.1. El sistema le muestra una galería de imágenes con los interiores de la residencia. Se termina el caso de uso.
<b>Poscondiciones</b>		
<b>Sub-Sección “Obras de Arte” de la Sección “Residencia”</b>		
Acción del Actor		Respuesta del Sistema
3. El visitante selecciona el submenú Obras de Arte.		3.1. El sistema le muestra una galería de imágenes con las obras de arte que se encuentran en la residencia. Se termina el caso de uso.
<b>Poscondiciones</b>		
<b>Flujo Normal de Eventos</b>		
<b>Sección “Servicios”</b>		
Acción del Actor		Respuesta del Sistema
2. El visitante accede al menú Servicios		2.1. El sistema le muestra un submenú con los vínculos: Gastronomía,

	<p>Transporte, Tintorería y Aeropuerto.</p> <p>a) Si desea visualizar información sobre gastronomía, ir a la Sub-Sección “Gastronomía” de la Sección “Servicios”</p> <p>b) Si desea visualizar información sobre el transporte, ir a la Sub-Sección “Transporte” de la Sección “Servicios”</p> <p>c) Si desea visualizar información sobre tintorería, ir a la Sub-Sección “Tintorería” de la Sección “Servicios”</p> <p>d) Si desea visualizar información sobre el aeropuerto, ir a la Sub-Sección “Aeropuerto” de la Sección “Servicios”</p>
<b>Sub-Sección “Gastronomía” de la Sección “Servicios”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
3. El visitante selecciona el submenú Gastronomía.	3.1. El sistema le muestra la información relacionada con la misma y dos vínculos: Carta Menú y Bebidas.
4. El visitante selecciona uno de los vínculos de Gastronomía.	4.1. Si selecciona “Carta Menú” el sistema le muestra la carta menú del día con vínculos a cada plato.

5. El visitante selecciona un plato dentro de la carta menú.	5.1. El sistema le muestra una imagen del plato seleccionado. Se termina el caso de uso.
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	4.1. Si selecciona la opción “Bebidas”, el sistema le muestra las bebidas por categorías.
5. El visitante selecciona el tipo de bebida que desea ver.	5.1. El sistema le muestra una galería de imágenes relacionadas con el tipo de bebida seleccionada. Se termina el caso de uso.
<b>Sub-Sección “Transporte” de la Sección “Servicios”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
3. El visitante selecciona el submenú Transporte.	3.1 El sistema le muestra información referente al servicio de transporte. Se termina el caso de uso.
<b>Poscondiciones</b>	
<b>Sub-Sección “Tintorería” de la Sección “Servicios”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
3. El visitante selecciona el submenú Tintorería.	3.1 El sistema le muestra información referente al servicio de tintorería. Se termina el caso de uso.
<b>Poscondiciones</b>	

<b>Sub-Sección “Aeropuerto” de la Sección “Servicios”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
3. El visitante selecciona el submenú Aeropuerto.	3.1 El sistema le muestra información referente al servicio de aeropuerto. Se termina el caso de uso.
<b>Poscondiciones</b>	
<b>Flujo Normal de Eventos</b>	
<b>Sección “Enlaces”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
2. El visitante accede al menú Enlaces	<p>2.1. El sistema le muestra un submenú con los vínculos: “Sitios Cubanos” y “Otros Sitios”</p> <p>a) Si desea visualizar información sobre los sitios cubanos, ir a la Sub-Sección “Sitios Cubanos” de la Sección “Enlaces”.</p> <p>b) Si desea visualizar información sobre otros sitios, ir a la Sub-Sección “Otros Sitios” de la Sección “Enlaces”.</p>
<b>Sub-Sección “Sitios de Cuba” de la Sección “Enlaces”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
3. El visitante selecciona el submenú Sitios Cubanos.	3.1. El sistema le muestra un listado con los sitios cubanos y los vínculos a estos.

4. El visitante selecciona uno de los sitios cubanos disponibles.	4.1. El sistema lo envía al sitio web seleccionado. Se termina el caso de uso.
<b>Sub-Sección “Otros Sitios” de la Sección “Enlaces”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
3. El visitante selecciona el vínculo Otros Sitios.	3.1. El sistema le muestra un listado con otros sitios y los vínculos a estos.
4. El visitante selecciona uno de los otros sitios disponibles.	4.1. El sistema lo envía al sitio web seleccionado. Se termina el caso de uso.
<b>Poscondiciones</b>	
<b>Flujo Normal de Eventos</b>	
<b>Sección “Sitial Histórico”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
2. El visitante accede al menú Sitial Histórico.	2.1. El sistema le muestra información referente al sitial de histórico. Se termina el caso de uso.
<b>Poscondiciones</b>	
<b>Flujo Normal de Eventos</b>	
<b>Sección “Entorno”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
2. El visitante accede al menú Entorno.	2.1. El sistema le muestra al visitante las imágenes del entorno que rodea la residencia. Se termina el caso de uso.
<b>Poscondiciones</b>	

Flujo Normal de Eventos	
Sección "Portada"	
Acción del Actor	Respuesta del Sistema
2. El visitante accede al menú Portada.	2.1. El sistema le muestra una breve información de las residencias de protocolo, características generales de las mismas y otros datos de interés. Se termina el caso de uso.
<b>Poscondiciones</b>	

**Tabla 17: Emitir opinión**

<b>Caso de Uso:</b>	Emitir Opinión.
<b>Actores:</b>	Visitante.
<b>Resumen:</b>	El caso de uso se inicia cuando el visitante decide ofrecer su opinión sobre un servicio en particular o contestar una encuesta. En caso de que desee enviar su opinión el sistema muestra un formulario, el visitante llena los campos y los envía, el sistema guarda dicha información. En caso de que desee llenar una encuesta el sistema le muestra la misma, el visitante la contesta y la envía, el sistema guarda la encuesta y finaliza así el caso de uso.
<b>Relación con otros CU</b>	
<b>Precondiciones:</b>	El visitante debe encontrarse dentro de algún servicio.
<b>Referencias:</b>	R4, R4.1, R4.2

<b>Prioridad</b>	Secundario.
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. El visitante desea ofrecer su opinión sobre un servicio en particular o contestar una encuesta.	<p>1.1. El sistema ofrece las siguientes opciones:</p> <p>a) Si decide ofrecer su opinión sobre el servicio ir a la Sección “Ofrecer Opinión”.</p> <p>b) Si decide contestar una encuesta ir a la sección “Contestar Encuesta”.</p>
<b>Flujo Normal de Eventos</b>	
<b>Sección “Ofrecer Opinión”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
2. El visitante accede a la opción “opinión”	2.1 El sistema le muestra un formulario con los datos que deben ser llenados.
3. El visitante llena los campos solicitados.	3.1 El sistema verifica que los datos sean válidos.
	3.2 El sistema guarda la opinión y envía al usuario un mensaje “la opinión ha sido enviada correctamente”
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>

	3.1 Si los datos son incorrectos el sistema devuelve al usuario un mensaje de alerta “la opinión no ha sido enviada, los datos son incorrectos”. Ir a la acción 2.1.
<b>Poscondiciones</b>	Queda registrada la opinión del visitante.
<b>Flujo Normal de Eventos</b>	
<b>Sección “Contestar Encuesta”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
2. El visitante accede a la acción “encuesta”	2.1 El sistema muestra la encuesta.
3. El visitante emite su voto y selecciona la opción enviar.	3.1 El sistema verifica que los datos sean correctos.
	3.2 El sistema guarda la encuesta y envía al usuario un mensaje “la encuesta ha sido enviada correctamente”
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	3.1 Si los datos son incorrectos el sistema devuelve al usuario un mensaje de alerta “la encuesta no ha sido enviada, los datos son incorrectos”. Ir a la acción 2.1.
<b>Poscondiciones</b>	Queda registrada la encuesta emitida por el visitante.

**Tabla 18: Autenticar administrador.**

<b>Caso de Uso:</b>	Autenticar administrador.	
<b>Actores:</b>	Administrador	
<b>Resumen:</b>	El caso de uso se inicia cuando el administrador intenta autenticarse en el sistema. El sistema muestra un formulario con los campos que necesitan ser llenados. El administrador introduce los datos y los envía, el sistema valida dichos datos y le da los permisos correspondientes finalizando así el caso de uso.	
<b>Relación con otros CU:</b>		
<b>Precondiciones:</b>		
<b>Referencias:</b>	R5	
<b>Prioridad:</b>	Crítico	
<b>Flujo Normal de Eventos</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	
1. El administrador escribe en la barra de dirección del navegador la dirección de la página de autenticación.	1.1 El sistema le muestra un formulario con los campos de usuario y contraseña.	
2. El administrador se autentifica insertando sus datos.	2.1. El sistema verifica que los datos sean válidos.	
	2.2. El sistema asigna los permisos. Se termina el caso de uso.	
<b>Flujos Alternos</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	
	2.1. Si los datos son incorrectos el	

	sistema devuelve al usuario un mensaje de alerta “los datos entrados son incorrectos”. Ir a la acción 1.1.
<b>Poscondiciones:</b>	El administrador queda autenticado en el sitio.

**Tabla 19: Crear contenido.**

<b>Caso de Uso:</b>	Crear contenido.
<b>Actores:</b>	Administrador
<b>Resumen:</b>	El caso de uso se inicia cuando el administrador decide crear contenidos del tipo residencia, obras de arte, bebidas o sitios. El sistema muestra un formulario con los campos que deben ser llenados y el administrador crea el contenido finalizando así el caso de uso.
<b>Relación con otros CU</b>	CU- Autenticar Administrador
<b>Precondiciones:</b>	El administrador debe estar autenticado.
<b>Referencias:</b>	R6, R6.1, R6.2., R6.3, R6.4.
<b>Prioridad</b>	Crítico
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. El administrador accede al vínculo crear contenido.	1.1 El sistema muestra un listado con todos los tipos de contenidos existentes.  a) Si desea crear un contenido del tipo residencia, ir a la Sección “Añadir

	<p>Residencia”.</p> <p>b) Si desea crear un contenido del tipo obras de arte, ir a la Sección “Añadir Obras de Arte”.</p> <p>c) Si desea crear un contenido del tipo bebidas, ir a la Sección “Añadir Bebidas”.</p> <p>d) Si desea crear un contenido del tipo sitios, ir a la Sección “Añadir Sitios”.</p>
<b>Flujo Normal de Eventos</b>	
<b>Sección “Añadir Residencia”</b>	
Acción del Actor	Respuesta del Sistema
2. El administrador accede al tipo de contenido Residencia.	2.1 El sistema muestra un formulario con los campos que deben ser llenados.
3. El administrador introduce los datos	3.1 El sistema valida los datos.
	3.2 El sistema crea el contenido. Se termina el caso de uso.
<b>Flujos Alternos</b>	
Acción del Actor	Respuesta del Sistema
	3.1. Si los datos son incorrectos el sistema devuelve al usuario un mensaje de alerta “los datos entrados son incorrectos”. Ir la acción 2.1.

<b>Poscondiciones</b>	Queda creado un contenido de residencia.
<b>Flujo Normal de Eventos</b>	
<b>Sección “Añadir Obras de Arte”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
2. El administrador accede al tipo de contenido Obras de Arte.	2.1 El sistema muestra un formulario con los campos que deben ser llenados.
3. El administrador introduce los datos	3.1 El sistema valida los datos.
	3.2 El sistema crea el contenido. Se termina el caso de uso.
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	3.1. Si los datos son incorrectos el sistema devuelve al usuario un mensaje de alerta “los datos entrados son incorrectos”. Ir a la acción 2.1.
<b>Poscondiciones</b>	Queda creado un contenido de Obras de Arte.
<b>Flujo Normal de Eventos</b>	
<b>Sección “Añadir Bebidas”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
2. El administrador accede al tipo de contenido Bebidas.	2.1 El sistema muestra un formulario con los campos que deben ser llenados.
3. El administrador introduce los datos	3.1 El sistema valida los datos.

	3.2 El sistema crea el contenido. Se termina el caso de uso.
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	3.1. Si los datos son incorrectos el sistema devuelve al usuario un mensaje de alerta “los datos entrados son incorrectos”. Ir a la acción 2.1.
<b>Poscondiciones</b>	Queda creado un contenido de Bebidas.
<b>Flujo Normal de Eventos</b>	
<b>Sección “Añadir Sitios”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
2. El administrador accede al tipo de contenido Sitios.	2.1 El sistema muestra un formulario con los campos que deben ser llenados.
3. El administrador introduce los datos	3.1 El sistema valida los datos.
	3.2 El sistema crea el contenido. Se termina el caso de uso.
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	3.1. Si los datos son incorrectos el sistema devuelve al usuario un mensaje de alerta “los datos entrados son incorrectos”. Ir a la acción 2.1.
<b>Poscondiciones</b>	Queda creado un contenido de Sitios.

**Tabla 20: Gestionar contenidos.**

<b>Caso de Uso:</b>	Gestionar contenidos.	
<b>Actores:</b>	Administrador	
<b>Resumen:</b>	El caso de uso se inicia cuando el administrador decide actualizar o eliminar los contenidos (residencia, obras de arte, bebidas y sitios). El sistema muestra las opciones para actualizar o eliminar, el administrador realiza las operaciones necesarias y finaliza así el caso de uso.	
<b>Relación con otros CU</b>	CU- Autenticar Administrador	
<b>Precondiciones:</b>	El administrador debe estar autenticado.	
<b>Referencias:</b>	R7, R7.1, R7.2.	
<b>Prioridad</b>	Crítico	
<b>Flujo Normal de Eventos</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	
1. El administrador accede al vínculo contenido.	1.1 El sistema muestra un listado con todos los contenidos.  a) Si decide actualizar un contenido ir a la Sección "Editar Contenido".  b) Si decide eliminar un contenido ir a la Sección "Eliminar Contenido".	
<b>Flujo Normal de Eventos</b>		

<b>Sección “Editar Contenido”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
2. El administrador selecciona el contenido que desea editar.	2.1 El sistema muestra un formulario con los datos del contenido, brindando la posibilidad de modificar los mismos.
3. El administrador modifica los datos y los envía.	3.1 El sistema valida los datos.
	3.2 El sistema los actualiza. Se termina el caso de uso.
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	3.1 Si los datos son incorrectos el sistema devuelve al usuario un mensaje de alerta “los datos entrados son incorrectos”. Ir a la acción 2.1.
<b>Poscondiciones</b>	Queda actualizado el contenido.
<b>Flujo Normal de Eventos</b>	
<b>Sección “Eliminar Contenido”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
2. El administrador selecciona el contenido que desea eliminar.	2.1 El sistema pide confirmación para eliminar el contenido.
3. El administrador confirma la eliminación.	3.1 El sistema elimina el contenido. Se termina el caso de uso.
<b>Poscondiciones</b>	Queda eliminado el contenido.

**Tabla 21: Actualizar información de los servicios.**

<b>Caso de Uso:</b>	Actualizar información de los servicios.	
<b>Actores:</b>	Administrador.	
<b>Resumen:</b>	El caso de uso se inicia cuando el administrador decide actualizar la información de los servicios de gastronomía, transporte, aeropuerto y tintorería. El sistema muestra un formulario con los campos que pueden ser editados. El administrador cambia los datos y los envía. El sistema actualiza los datos, finalizando así el caso de uso.	
<b>Relación con otros CU</b>	CU- Autenticar Administrador	
<b>Precondiciones:</b>	El administrador debe estar autenticado.	
<b>Referencias:</b>	R8, R8.1, R8.2, R8.3, R8.4	
<b>Prioridad</b>	Secundario.	
<b>Flujo Normal de Eventos</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	
1. El administrador selecciona el servicio que desea actualizar.	1.1. El sistema muestra un formulario con los datos que pueden ser editados.	
2. El administrador modifica los datos y los envía.	2.1 El sistema valida los datos.	
	2.2. El sistema actualiza los datos. Se termina el caso de uso.	
<b>Flujos Alternos</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	
	2.1 Si los datos son incorrectos el sistema	

	devuelve al usuario un mensaje de alerta "los datos son incorrectos". Ir a la acción 1.1.
<b>Poscondiciones</b>	Queda actualizado el servicio.

**Tabla 22: Actualizar carta menú.**

<b>Caso de Uso:</b>	Actualizar carta menú.
<b>Actores:</b>	Administrador.
<b>Resumen:</b>	El caso de uso se inicia cuando el administrador decide actualizar la carta menú. El sistema muestra un formulario con los campos que pueden ser editados. El administrador cambia los datos y los envía. El sistema actualiza los datos, finalizando así el caso de uso.
<b>Relación con otros CU</b>	CU- Autenticar Administrador
<b>Precondiciones:</b>	El administrador debe estar autenticado.
<b>Referencias:</b>	R9
<b>Prioridad</b>	Crítico.
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. El administrador selecciona el vínculo de actualizar la carta menú.	1.1. El sistema muestra un formulario con los datos que pueden ser editados.
2. El administrador modifica los datos y envía la carta menú.	2.1 El sistema valida los datos.
	2.2 El sistema actualiza la carta menú y la

	muestra. Se termina el caso de uso.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	2.1 Si los datos son incorrectos el sistema devuelve al usuario un mensaje de alerta "los datos son incorrectos". Ir a la acción 1.1.
Poscondiciones	Queda actualizada la carta menú.

**Tabla 23: Gestionar Encuestas**

<b>Caso de Uso:</b>	Gestionar encuestas.
<b>Actores:</b>	Administrador.
<b>Resumen:</b>	El caso de uso se inicia cuando el administrador decide crear, actualizar o eliminar una encuesta. El sistema muestra las opciones para crear, actualizar o eliminar, el administrador realiza las operaciones necesarias y finaliza así el caso de uso.
<b>Relación con otros CU</b>	CU- Autenticar Administrador
<b>Precondiciones:</b>	El administrador debe estar autenticado.
<b>Referencias:</b>	R10, R10.1, R10.2, R10.3
<b>Prioridad</b>	Secundario.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema

<p>1. El administrador accede al vínculo encuesta.</p>	<p>1.1 El sistema muestra un listado con todas las encuestas.</p> <p>a) Si decide crear una encuesta ir a la Sección “Crear Encuesta”.</p> <p>b) Si decide actualizar una encuesta ir a la Sección “Actualizar Encuesta”.</p> <p>c) Si decide eliminar una Encuesta ir a la Sección “Eliminar Encuesta”.</p>
<p><b>Flujo Normal de Eventos</b></p>	
<p><b>Sección “Crear Encuesta”</b></p>	
<p><b>Acción del Actor</b></p>	<p><b>Respuesta del Sistema</b></p>
<p>2.1 administrador selecciona crear encuesta.</p>	<p>2.1. El sistema muestra un formulario con los datos que deben ser llenados.</p>
<p>3. El administrador llena los campos y envía la encuesta.</p>	<p>3.1 El sistema valida los datos.</p>
	<p>3.2 El sistema actualiza la encuesta. Se termina el caso de uso.</p>
<p><b>Flujos Alternos</b></p>	
<p><b>Acción del Actor</b></p>	<p><b>Respuesta del Sistema</b></p>
	<p>3.1 Si los datos son incorrectos el sistema devuelve al usuario un mensaje de alerta “los datos son incorrectos”. Ir a la acción</p>

	2.1.
<b>Poscondiciones</b>	Queda creada la encuesta.
<b>Flujo Normal de Eventos</b>	
<b>Sección “Actualizar Encuesta”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
2. El administrador selecciona la encuesta a modificar.	2.1. El sistema muestra un formulario con los datos que pueden ser editados.
3. El administrador modifica los datos y envía la encuesta.	3.1 El sistema valida los datos.
	3.2 El sistema actualiza la encuesta. Se termina el caso de uso.
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	3.1 Si los datos son incorrectos el sistema devuelve al usuario un mensaje de alerta “los datos son incorrectos”. Ir a la acción 2.1.
<b>Poscondiciones</b>	Queda actualizada la encuesta.
<b>Flujo Normal de Eventos</b>	
<b>Sección “Eliminar Encuesta”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
2. El administrador selecciona la encuesta a eliminar.	2.1 El sistema solicita confirmación para eliminar la encuesta.

3. El administrador confirma la eliminación.	3.1 El sistema elimina la encuesta. Se termina el caso de uso.
<b>Poscondiciones</b>	Queda eliminada la encuesta.

**Tabla 24: Administrar opiniones.**

<b>Caso de Uso:</b>	Administrar opiniones.
<b>Actores:</b>	Administrador.
<b>Resumen:</b>	El caso de uso se inicia cuando el administrador decide revisar o eliminar opiniones. El sistema muestra un listado con las opiniones de los visitantes. El administrador revisa o elimina la deseada y el sistema muestra o elimina la opinión respectivamente, finalizando así el caso de uso.
<b>Relación con otros CU</b>	CU- Autenticar Administrador
<b>Precondiciones:</b>	El administrador debe estar autenticado.
<b>Referencias:</b>	R11, R11.1, R11.2
<b>Prioridad</b>	Secundario.
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. El administrador selecciona el vínculo opiniones.	1.1. El sistema muestra un listado de las opiniones que han ofrecido los visitantes.  Si decide revisar una opinión ir a la Sección "Revisar Opinión".

	Si decide eliminar una opinión ir a la Sección "Eliminar Opinión".
<b>Flujo Normal de Eventos</b>	
<b>Sección "Revisar Opinión"</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
2. El administrador selecciona la opinión que desea revisar.	2.1 El sistema muestra la opinión seleccionada. Se termina el caso de uso.
<b>Poscondiciones</b>	
<b>Flujo Normal de Eventos</b>	
<b>Sección "Revisar Opinión"</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
2. El administrador selecciona la opinión que desea eliminar.	2.1 El sistema elimina la opinión seleccionada. Se termina el caso de uso.
<b>Poscondiciones</b>	Queda eliminada una opinión.

**Tabla 25: Actualizar entorno.**

<b>Caso de Uso:</b>	Actualizar entorno.
<b>Actores:</b>	Administrador.
<b>Resumen:</b>	El caso de uso se inicia cuando el administrador decide actualizar el entorno. El sistema muestra un formulario con los campos que pueden ser editados. El administrador modifica los datos y los envía. El sistema actualiza los datos, finalizando así el caso de uso.

<b>Relación con otros CU</b>	CU- Autenticar Administrador	
<b>Precondiciones:</b>	El administrador debe estar autenticado.	
<b>Referencias:</b>	R12	
<b>Prioridad</b>	Secundario.	
<b>Flujo Normal de Eventos</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	
1. El administrador selecciona el vínculo actualizar Entorno.	1.1. El sistema muestra un formulario con los datos que pueden ser editados.	
2. El administrador modifica los datos y los envía.	2.1 El sistema valida los datos.	
	2.2 El sistema actualiza el entorno. Se termina el caso de uso.	
<b>Flujos Alternos</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	
	2.1 Si los datos son incorrectos el sistema devuelve al usuario un mensaje de alerta "los datos son incorrectos". Ir a la acción 1.1.	
<b>Poscondiciones</b>	Queda actualizado el entorno.	

## 2.9 Conclusiones

En este capítulo se describió el sistema, los requisitos funcionales y no funcionales que debía cumplir el mismo, los autores y casos de uso del sistema así como la relación que existe entre ambos representados gráficamente en un diagrama. Se detalló paso a paso cada caso de uso, la acción de los autores y la respuesta del sistema a cada acción.

De esta forma quedan sentadas las bases para comenzar a construir el sistema teniendo en cuenta los requerimientos especificados en el capítulo.

## Capítulo 3: Análisis y Diseño del Sistema.

### 3.1 Introducción.

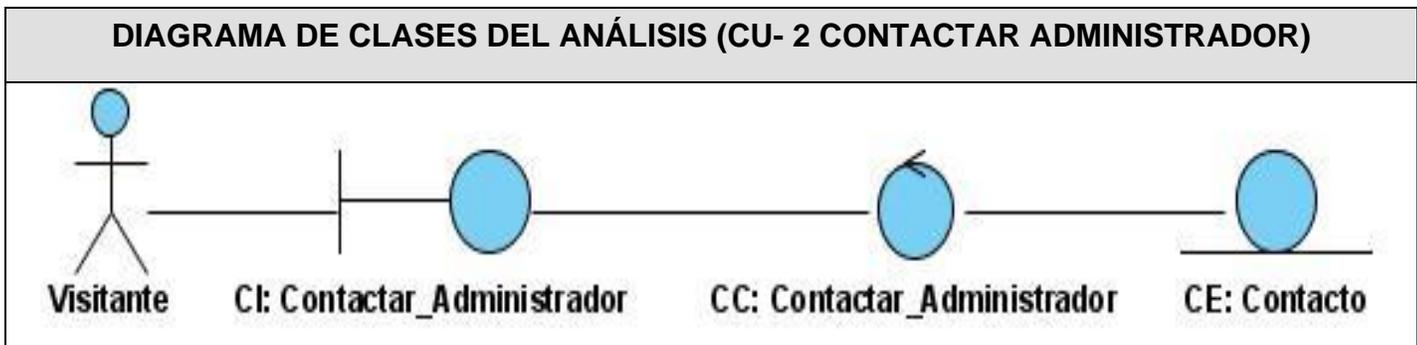
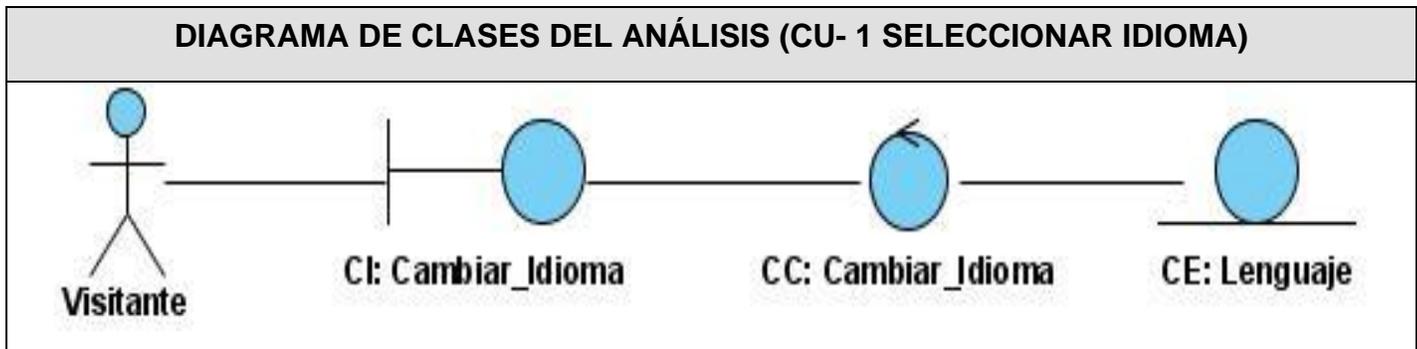
En el presente capítulo se describe el flujo de trabajo Análisis y Diseño, se muestran los diagramas de clases tanto del análisis como del diseño, se presenta el diagrama de colaboración correspondiente a cada caso de uso del sistema, se diseña la base de datos, se conforma el diagrama de clases persistentes y el modelo de datos, finalmente se describen las tablas y se muestra el diagrama de despliegue.

### 3.2 Análisis

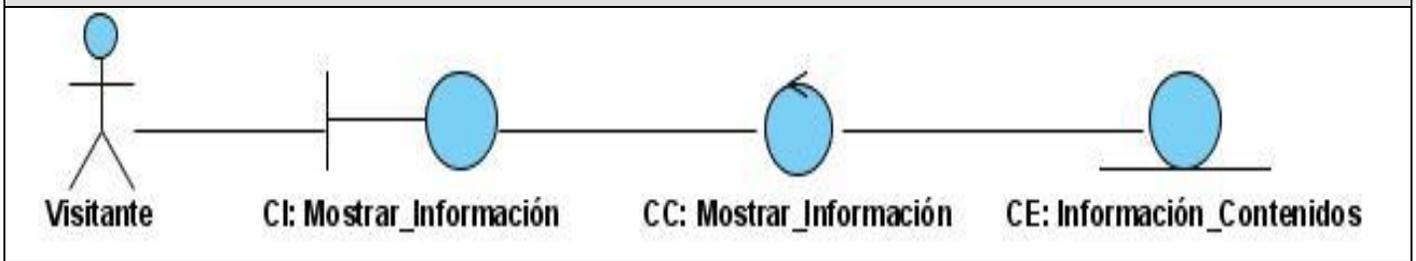
En el análisis se analizan los requisitos funcionales capturados y descritos anteriormente, de forma que se refinen y estructuran los mismos, con el propósito de lograr una comprensión más precisa de dichos requerimientos, para obtener una visión del sistema completo.

A continuación se refleja el diagrama de clases de análisis correspondiente a los casos de usos descritos.

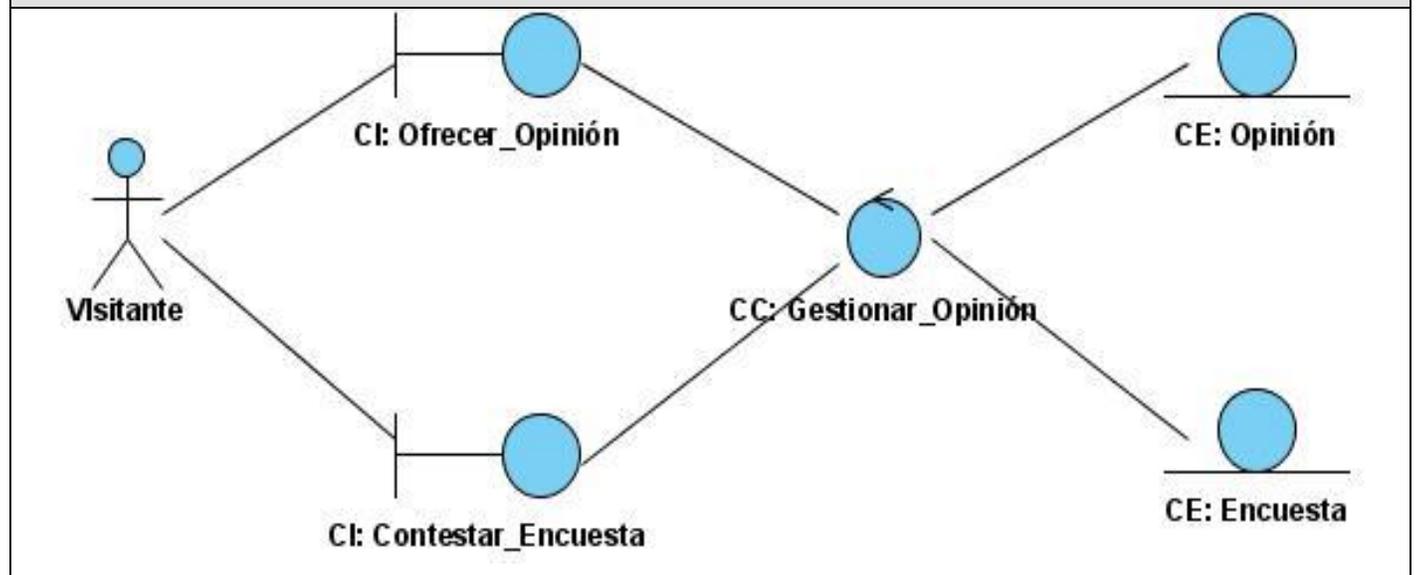
#### 3.2.1 Diagrama de clases del análisis.



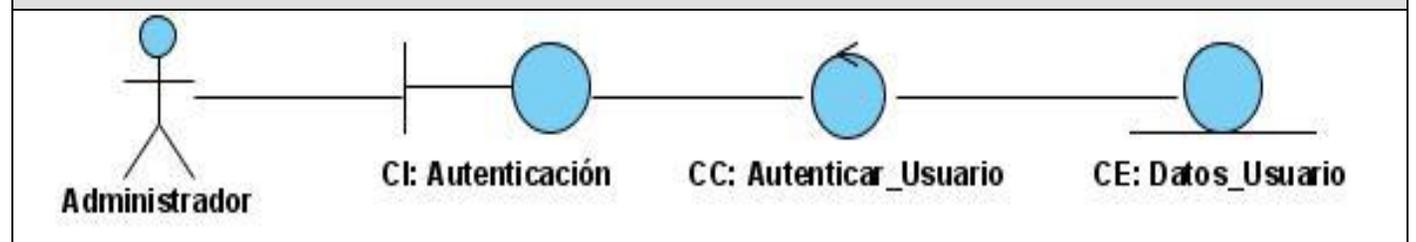
**DIAGRAMA DE CLASES DEL ANÁLISIS (CU- 3 VISUALIZAR INFORMACIÓN)**

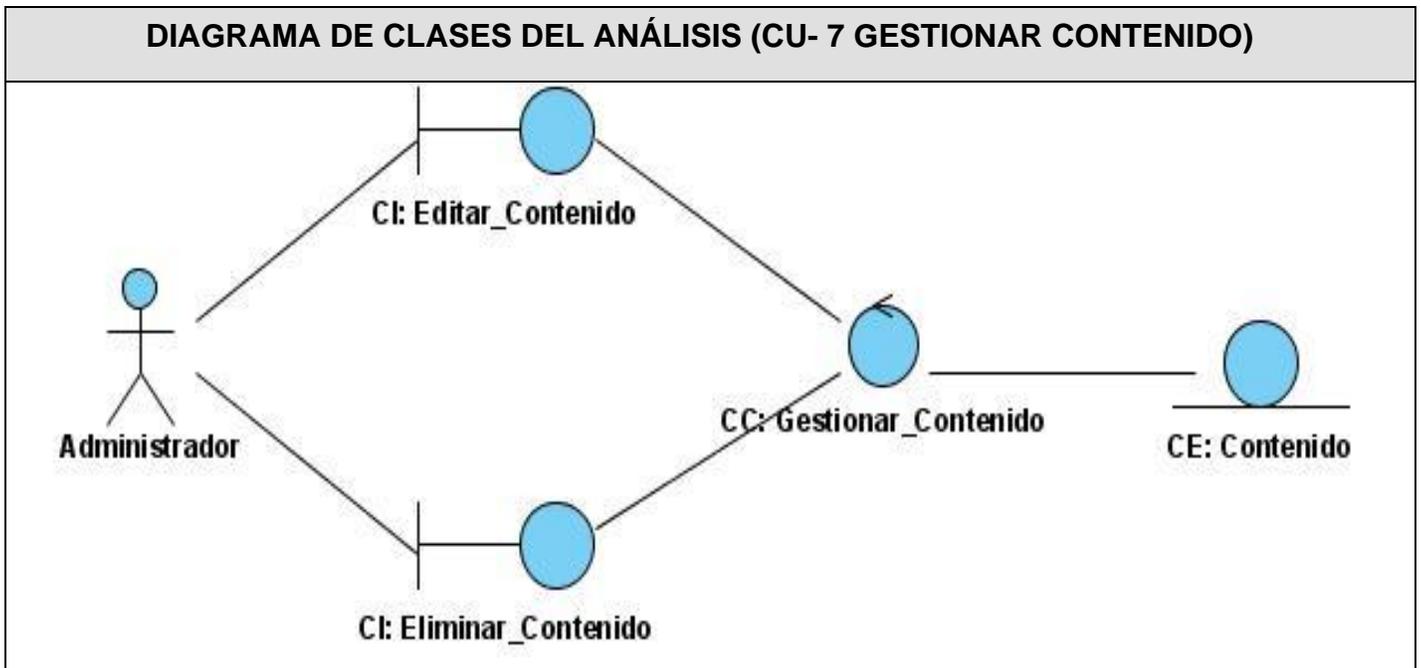
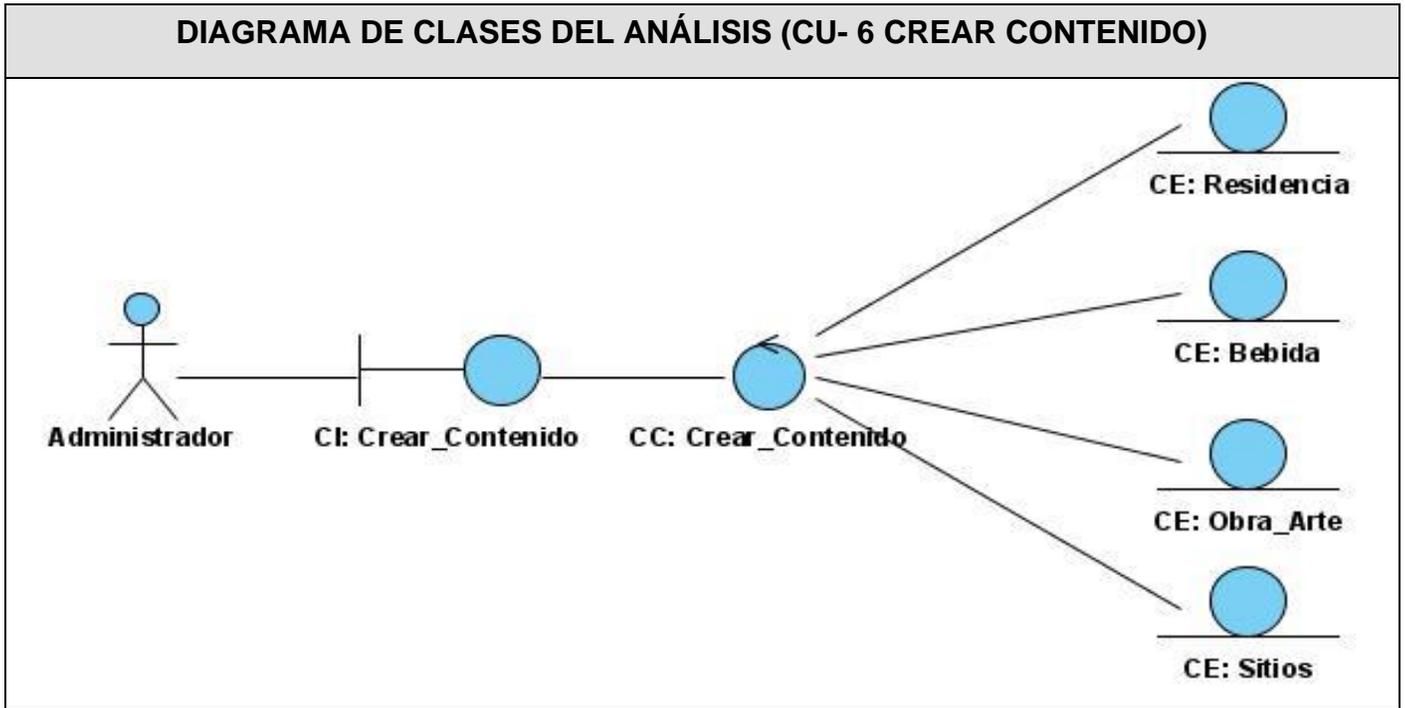


**DIAGRAMA DE CLASES DEL ANÁLISIS (CU- 4 EMITIR OPINIÓN)**



**DIAGRAMA DE CLASES DEL ANÁLISIS (CU- 5 AUTENTICAR ADMINISTRADOR)**

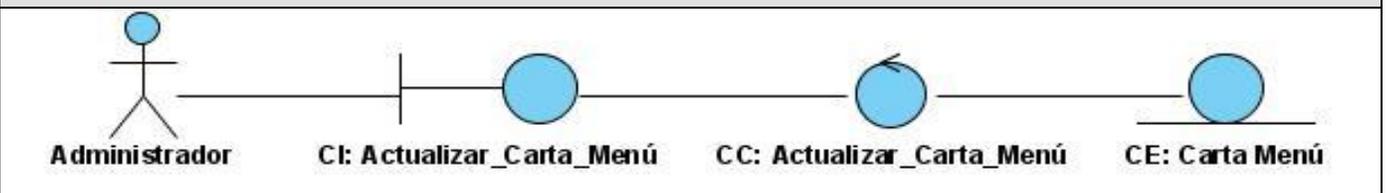




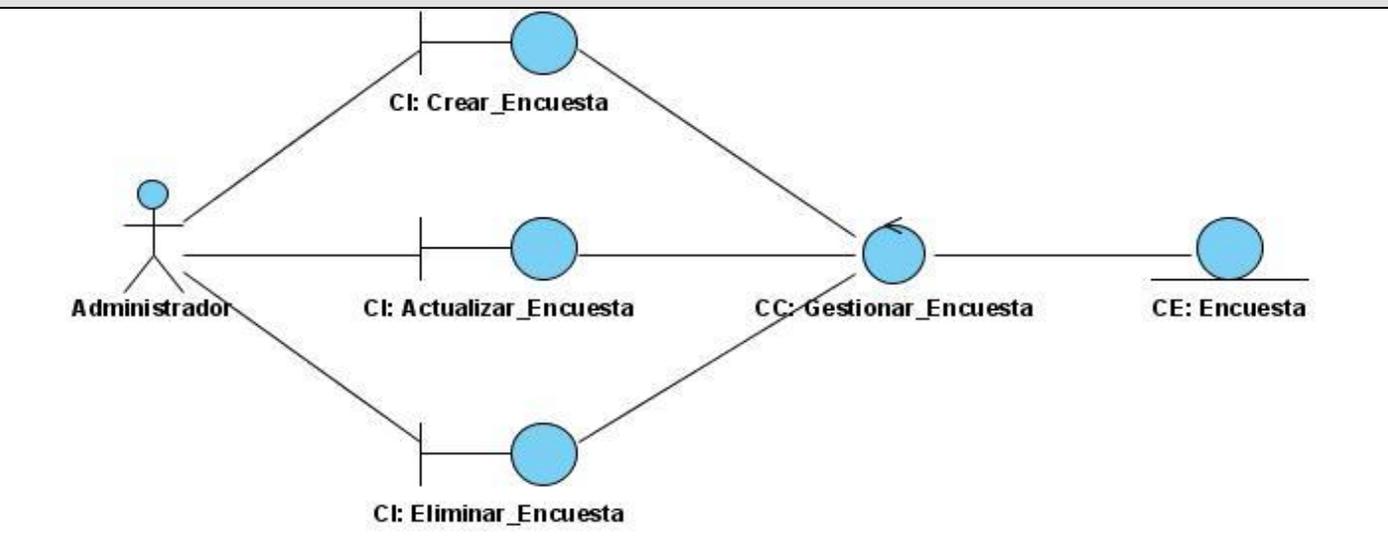
**DIAGRAMA DE CLASES DEL ANÁLISIS (CU- 8 ACTUALIZAR INFORMACIÓN SERVICIOS)**



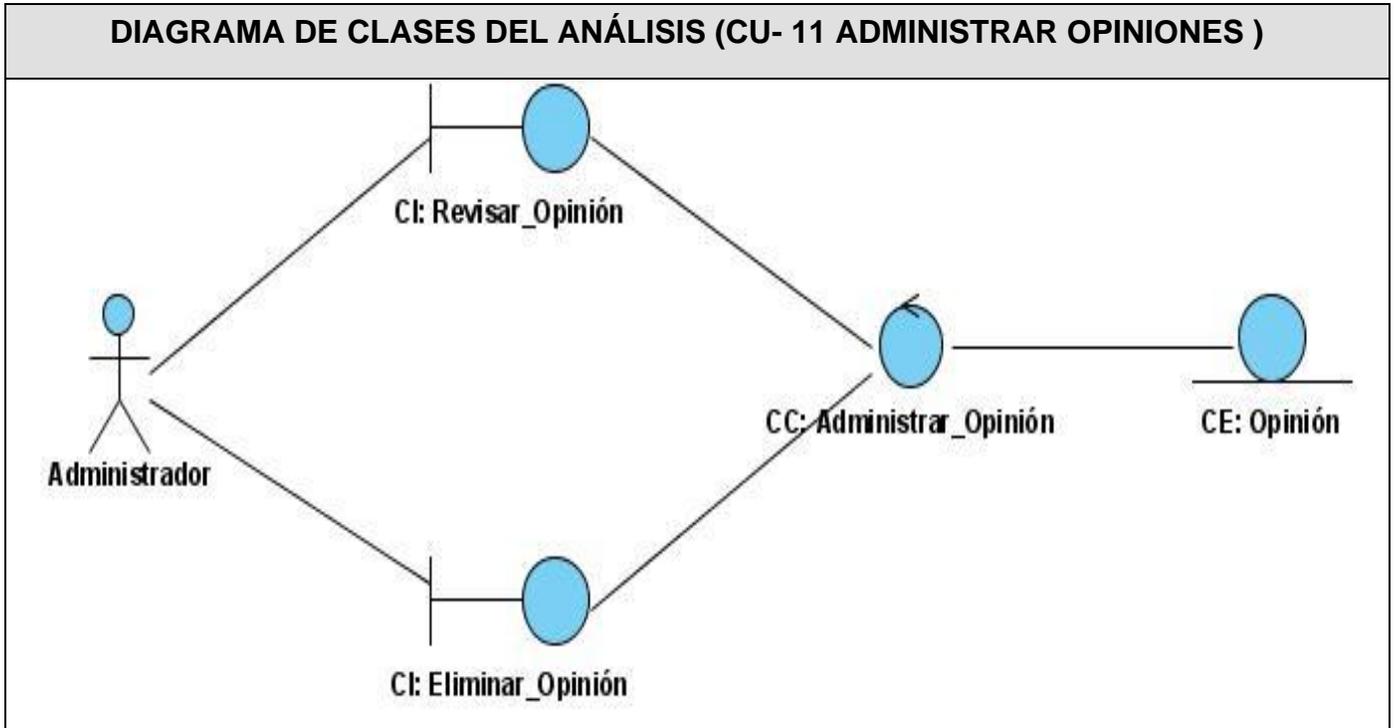
**DIAGRAMA DE CLASES DEL ANÁLISIS (CU- 9 ACTUALIZAR CARTA MENÚ)**



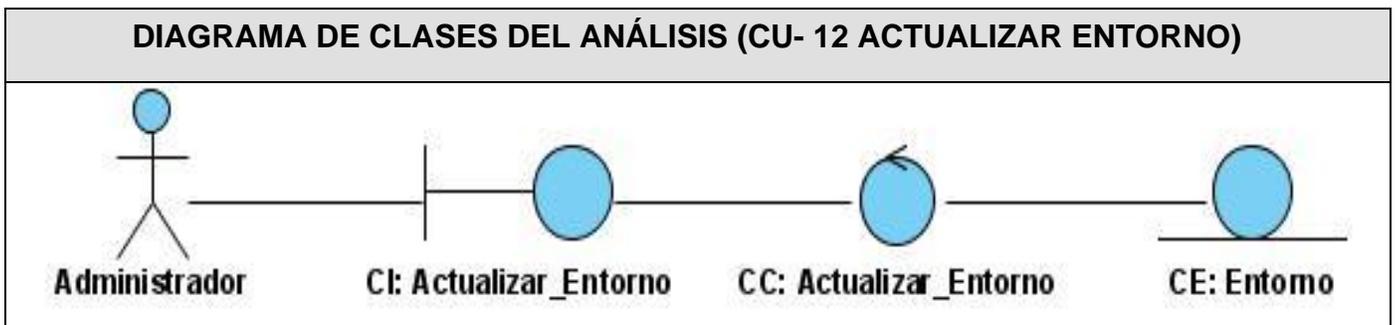
**DIAGRAMA DE CLASES DEL ANÁLISIS (CU- 10 GESTIONAR ENCUESTAS)**



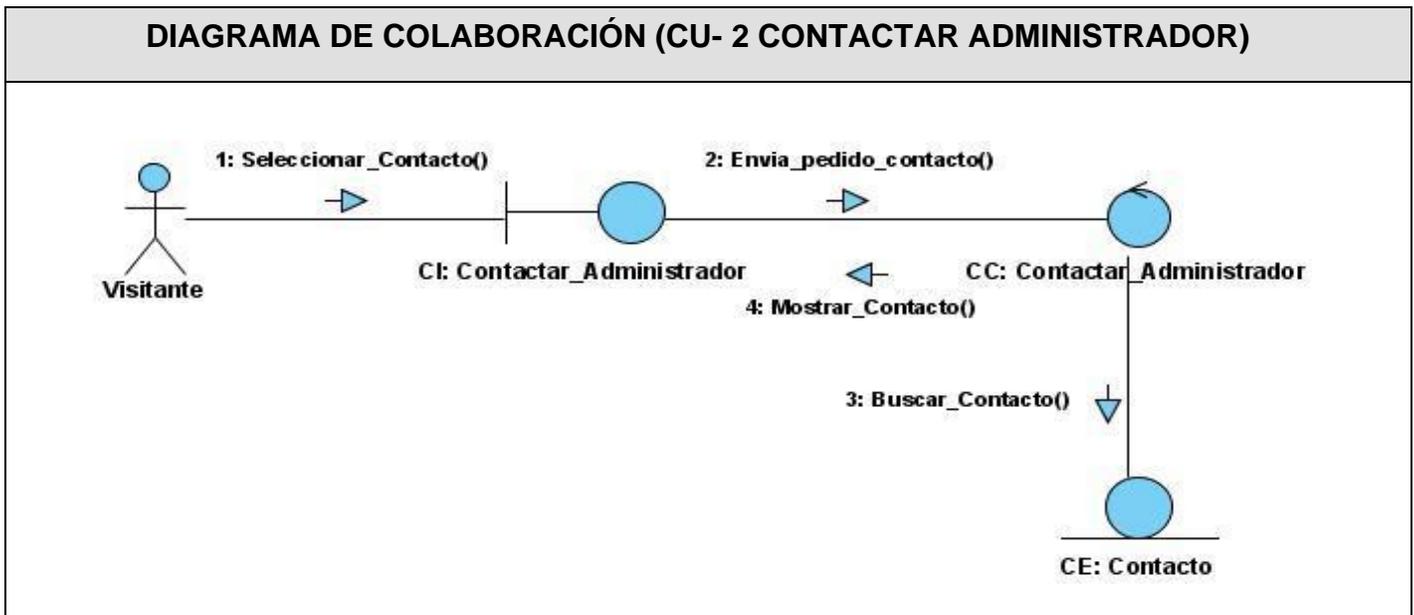
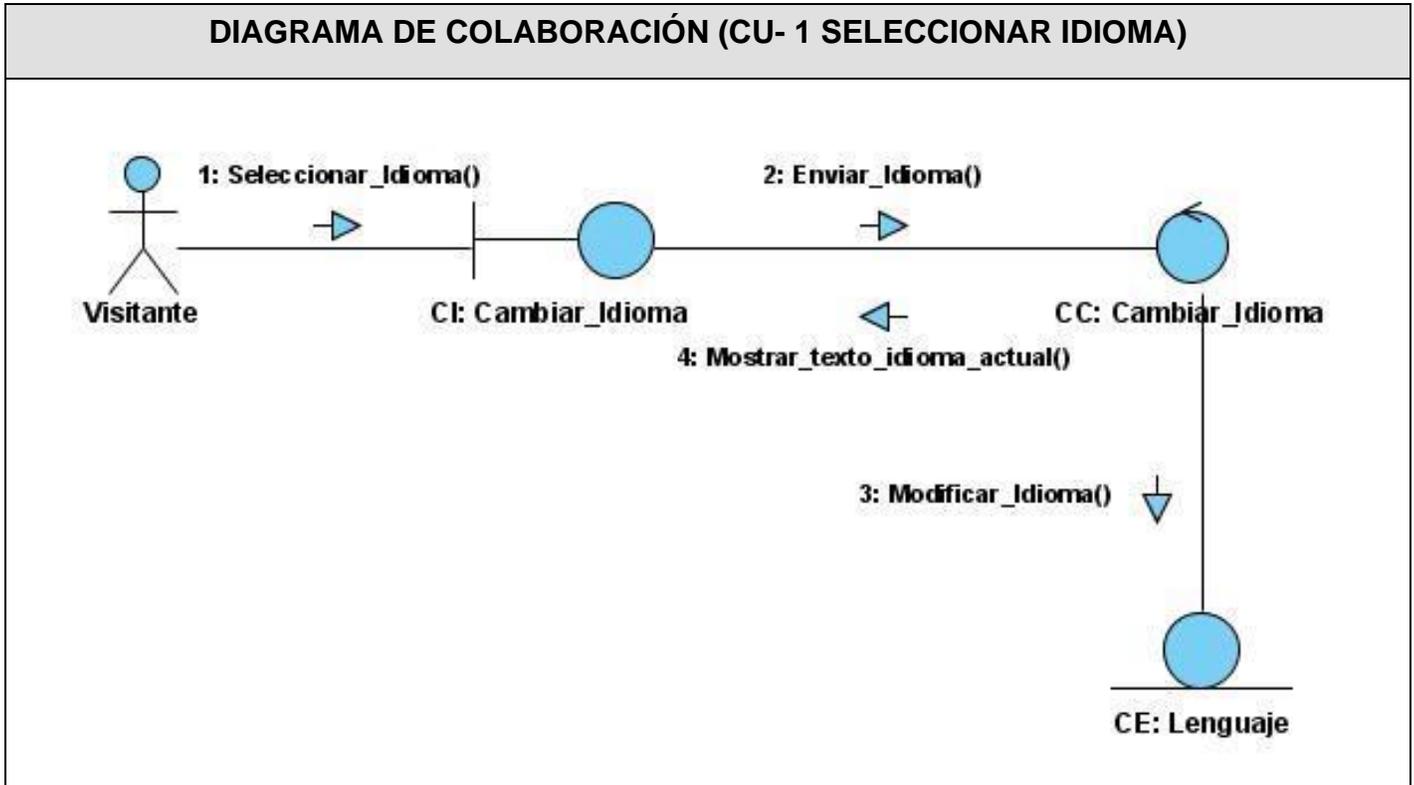
**DIAGRAMA DE CLASES DEL ANÁLISIS (CU- 11 ADMINISTRAR OPINIONES )**



**DIAGRAMA DE CLASES DEL ANÁLISIS (CU- 12 ACTUALIZAR ENTORNO)**



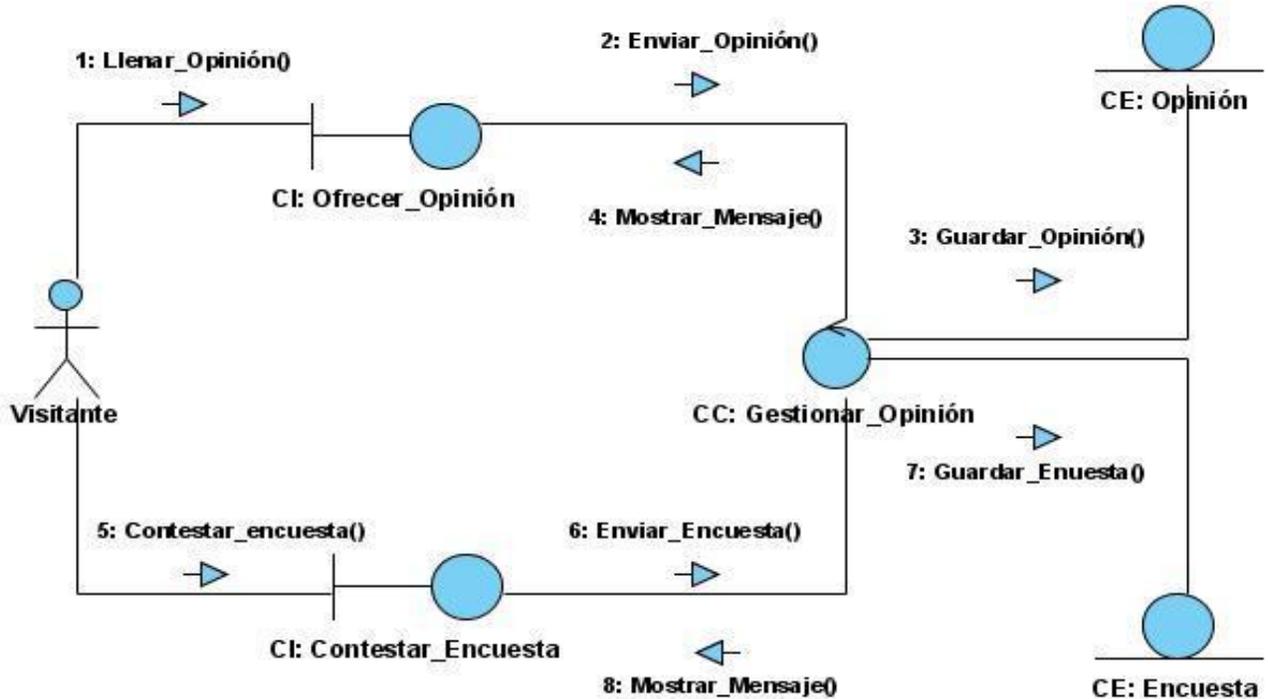
3.2.2 Diagrama de interacción (Colaboración).



**DIAGRAMA DE COLABORACIÓN (CU- 3 VISUALIZAR INFORMACIÓN)**



**DIAGRAMA DE COLABORACIÓN (CU- 4 EMITIR OPINIÓN)**



**DIAGRAMA DE COLABORACIÓN (CU- 5 AUTENTICAR ADMINISTRADOR)**

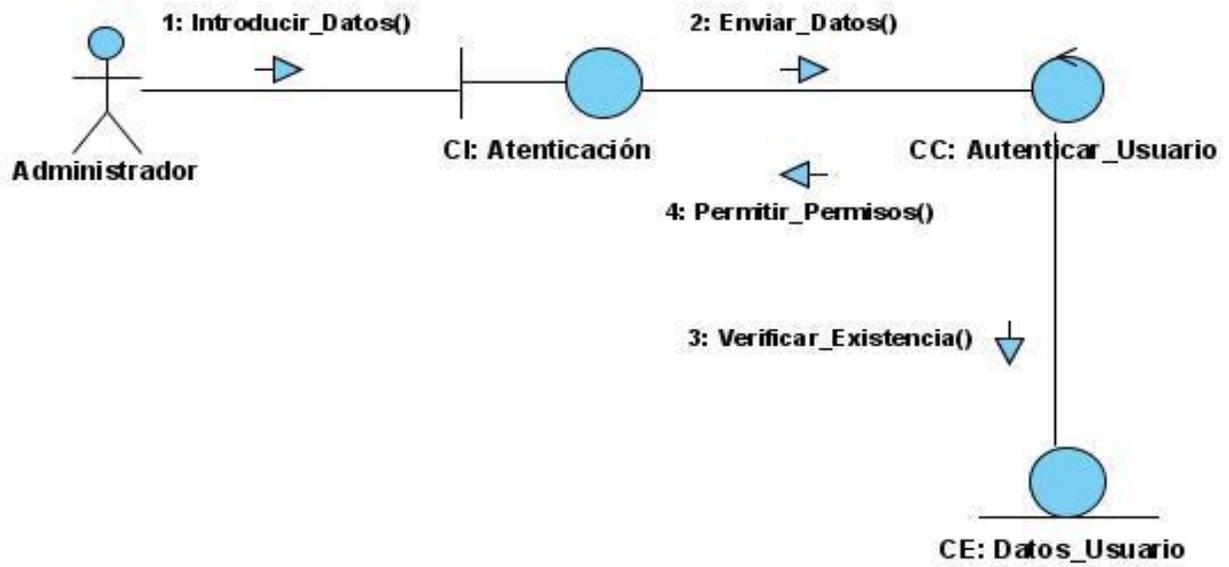
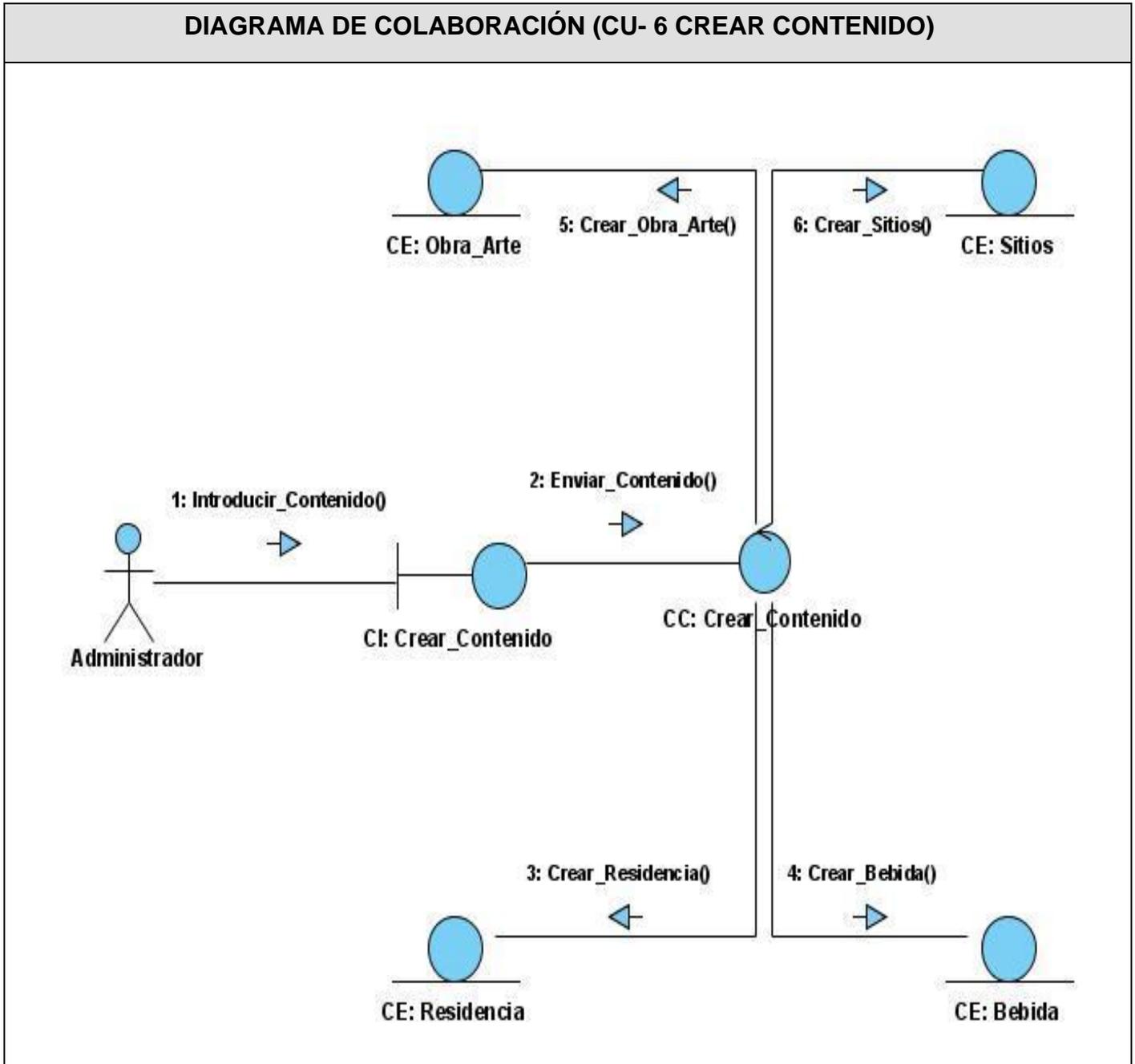
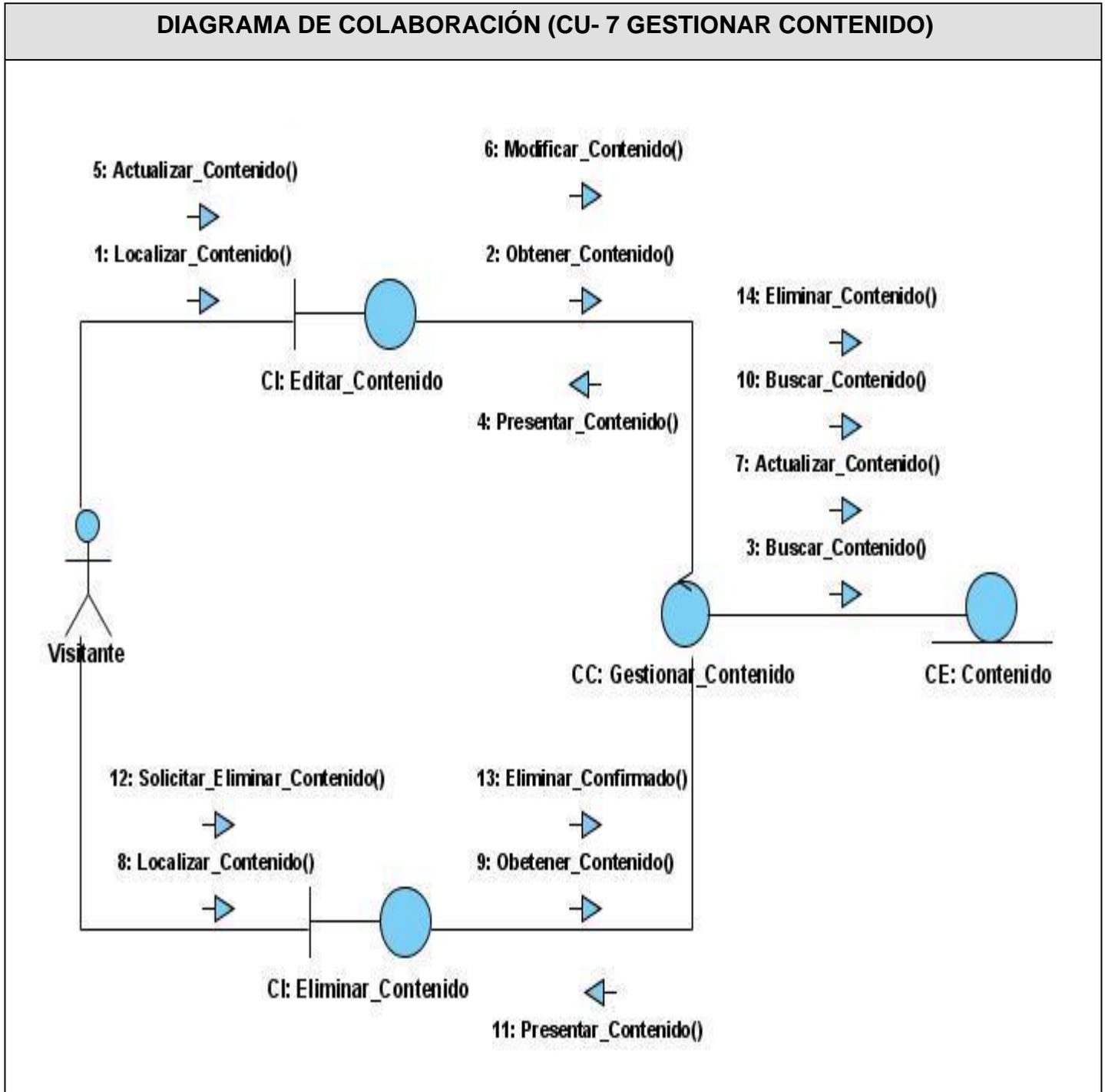
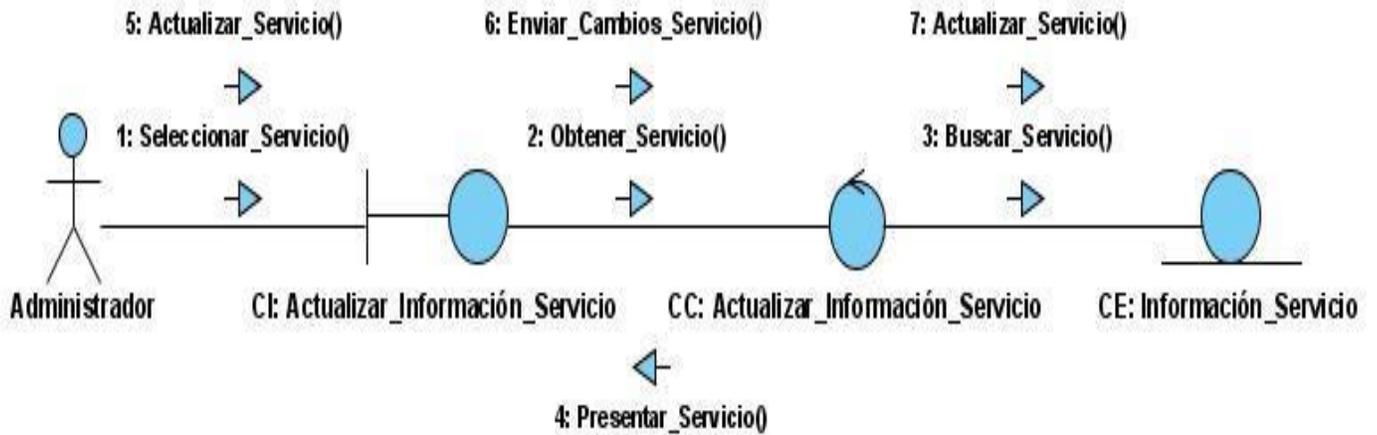


DIAGRAMA DE COLABORACIÓN (CU- 6 CREAR CONTENIDO)





**DIAGRAMA DE COLABORACIÓN (CU- 8 ACTUALIZAR INFORMACIÓN SERVICIOS)**



**DIAGRAMA DE COLABORACIÓN (CU- 9 ACTUALIZAR CARTA MENÚ)**

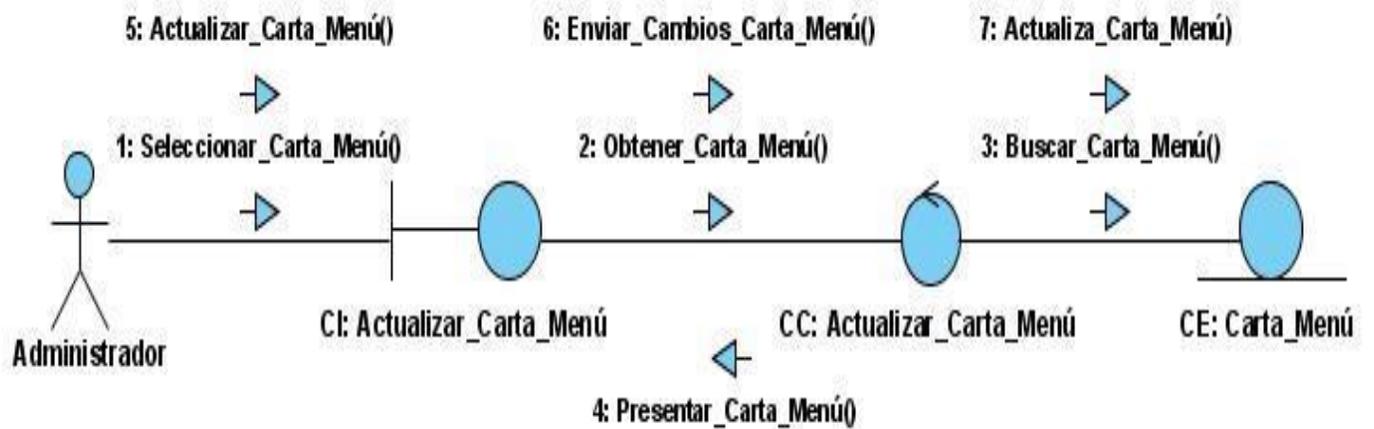


DIAGRAMA DE COLABORACIÓN (CU- 10 GESTIONAR ENCUESTAS)

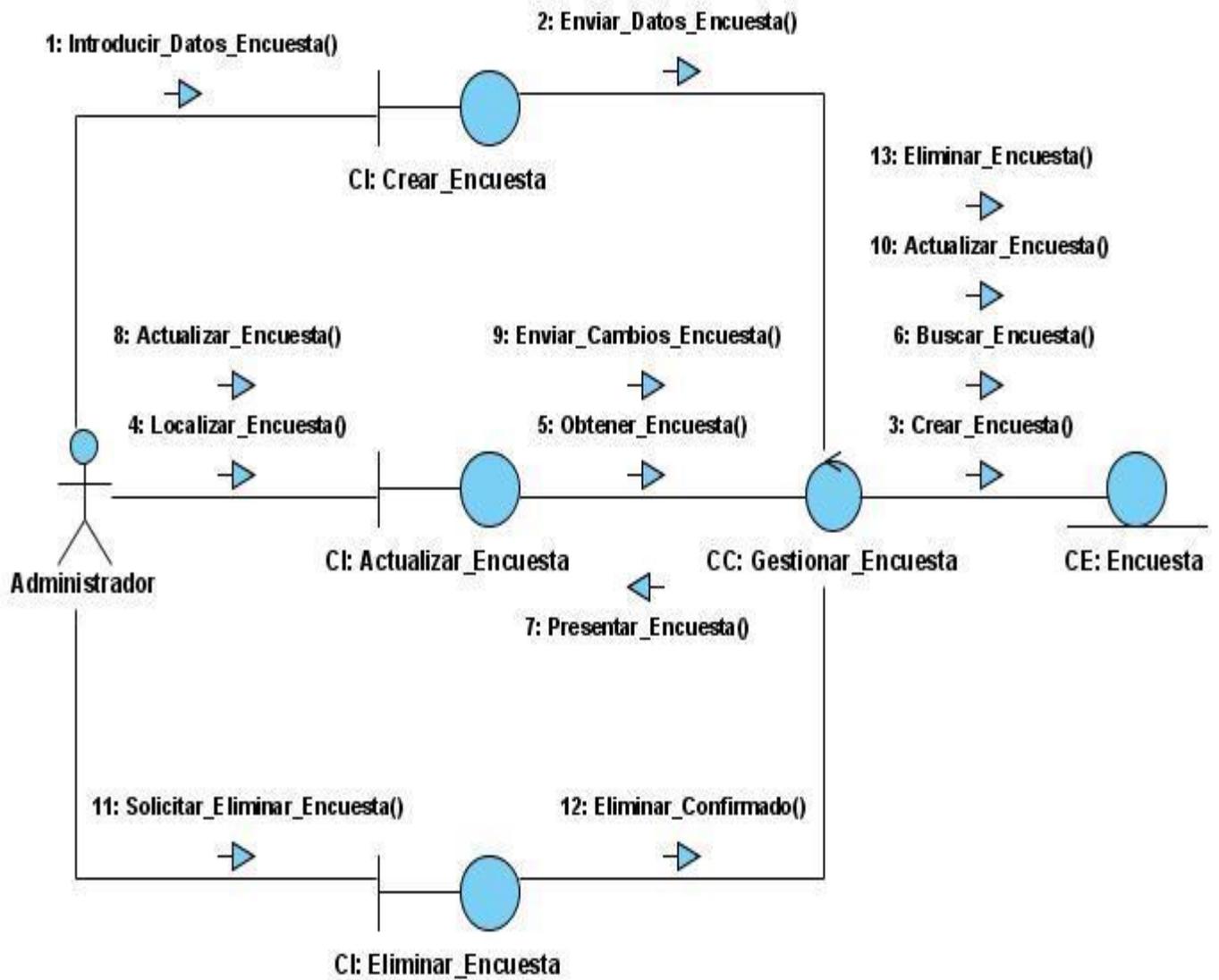
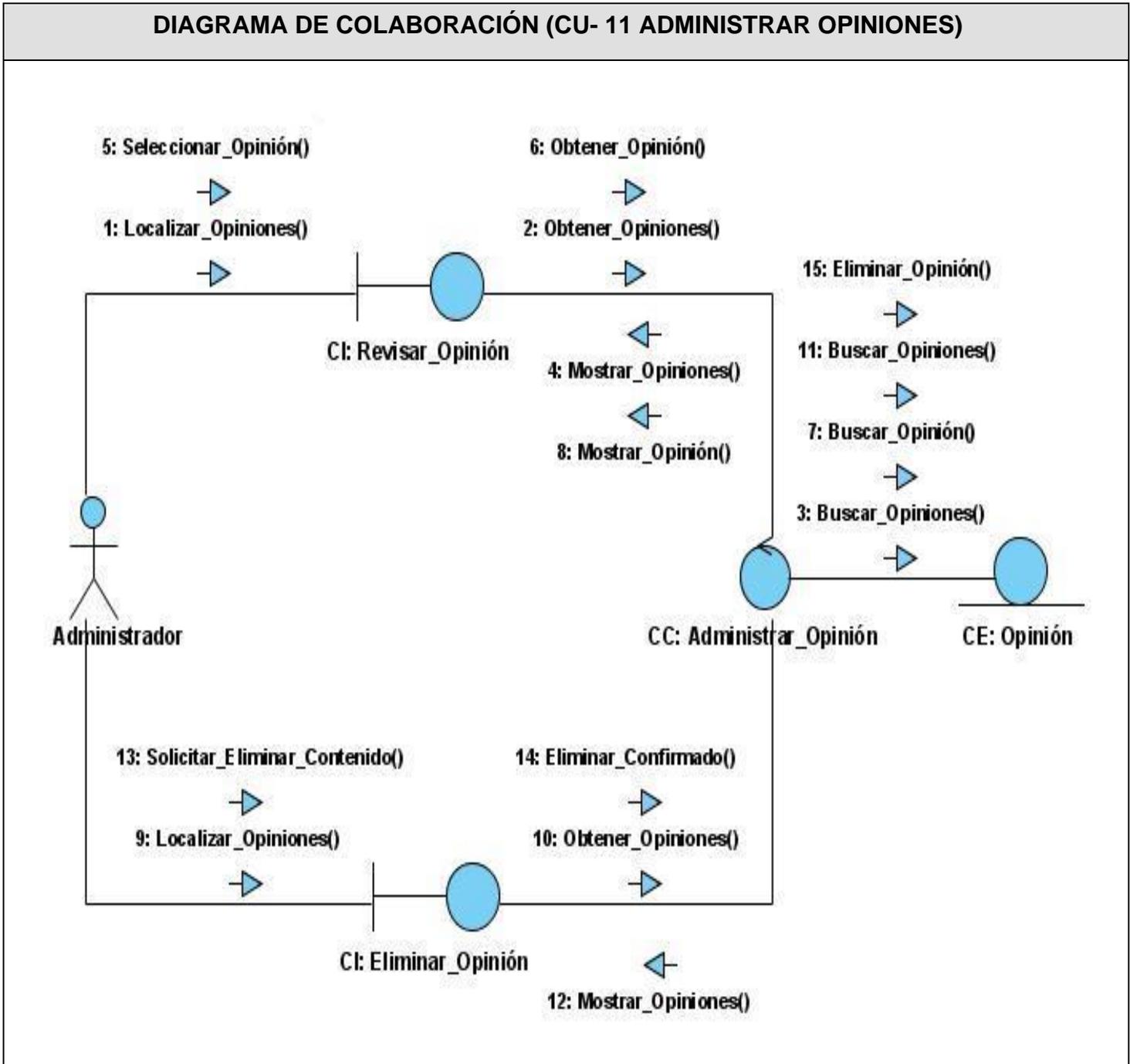
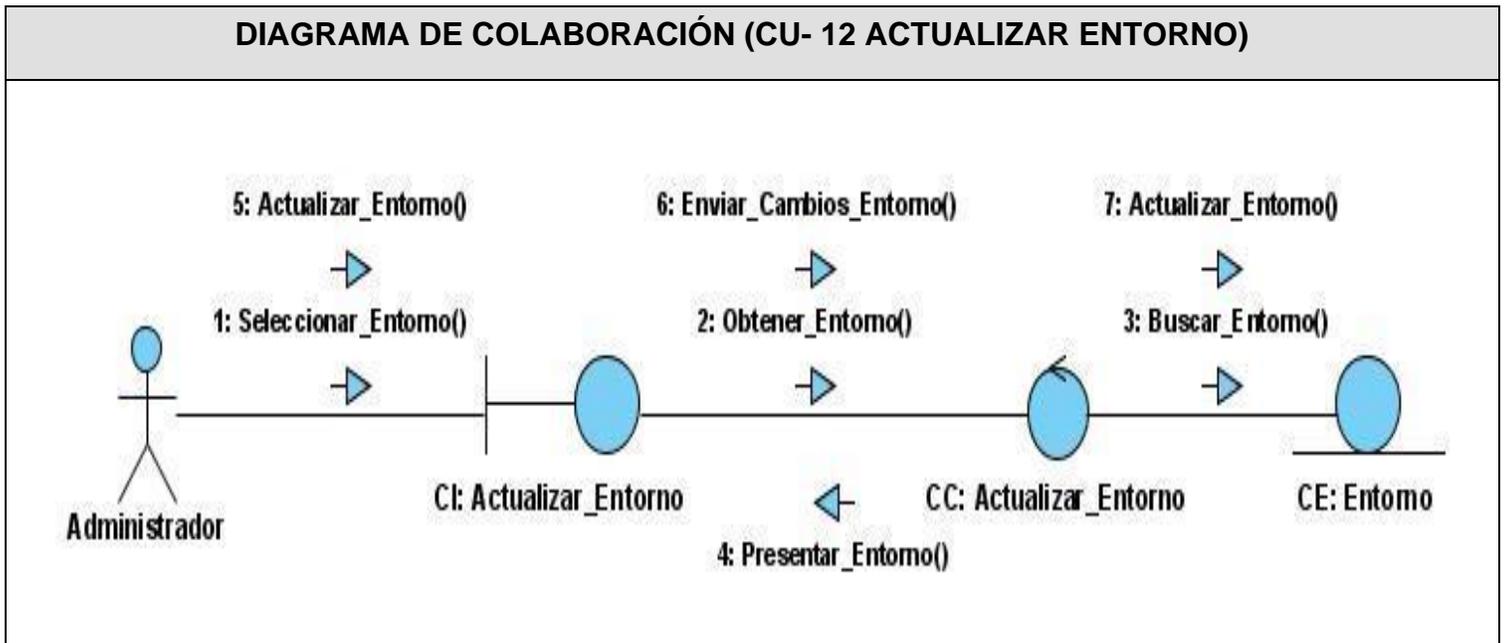


DIAGRAMA DE COLABORACIÓN (CU- 11 ADMINISTRAR OPINIONES)



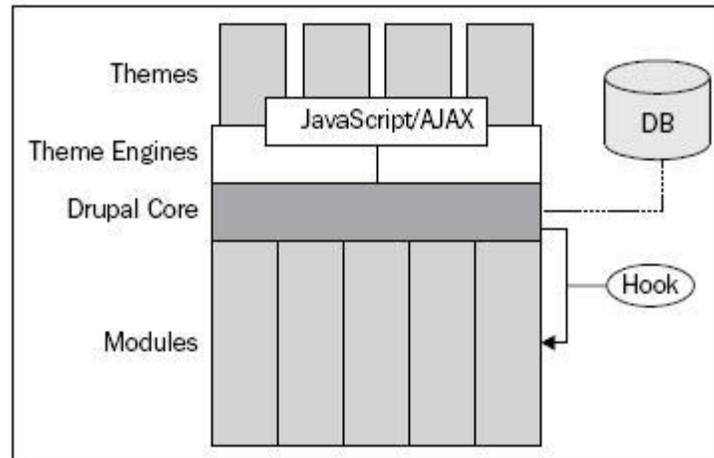


### 3.3 Diseño

El diseño es un refinamiento del análisis, que tiene en cuenta los requisitos no funcionales, en definitiva cómo cumple el sistema sus objetivos (Kruchten, 2001).

Para el desarrollo de la propuesta se hace uso del CMS Drupal, que utiliza el patrón Modelo Vista Controlador (MVC). Este es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. El patrón MVC se ve frecuentemente en aplicaciones web, donde la vista es la página HTML y el código que provee de datos dinámicos a la página; el modelo es el sistema de gestión de base de datos y la lógica de negocio; y el controlador es el responsable de recibir los eventos de entrada desde la vista. (ANONIMO, 2008)

En la siguiente figura se muestra como está estructurado en capas el CMS Drupal, donde la vista son "Themes", "Themes Engines", "JavaScript/Ajax"; el modelo es la base de datos (DB), y el controlador es "Drupal Core" y "Modules" que representa los módulos; donde el "hook" son funciones para facilitar la comunicación entre el Core de Drupal y los módulos.



**Figura 3: Arquitectura del CMS Drupal.**

### 3.3.1 Diagrama de clases del diseño web.

A continuación se muestra el diagrama de paquetes que genera el CMS Drupal, tomando el mismo como base para la representación de los diagramas de clases del diseño referente al sistema descrito.

El CMS Drupal provee un tipo de contenido genérico llamado “node”, y permite la creación, edición y publicación de este tipo de contenido. Los documentos creados se depositan en una base de datos central, donde también se guardan el resto de datos de la web. Posee un potente sistema de seguridad, basado en la creación de usuarios y roles, en dependencia de los permisos que se necesiten para administrar el sitio y acceder a los módulos.

Para que se comprenda de forma sencilla el diagrama de clases de diseño, se decidió describir por paquetes los módulos de drupal.

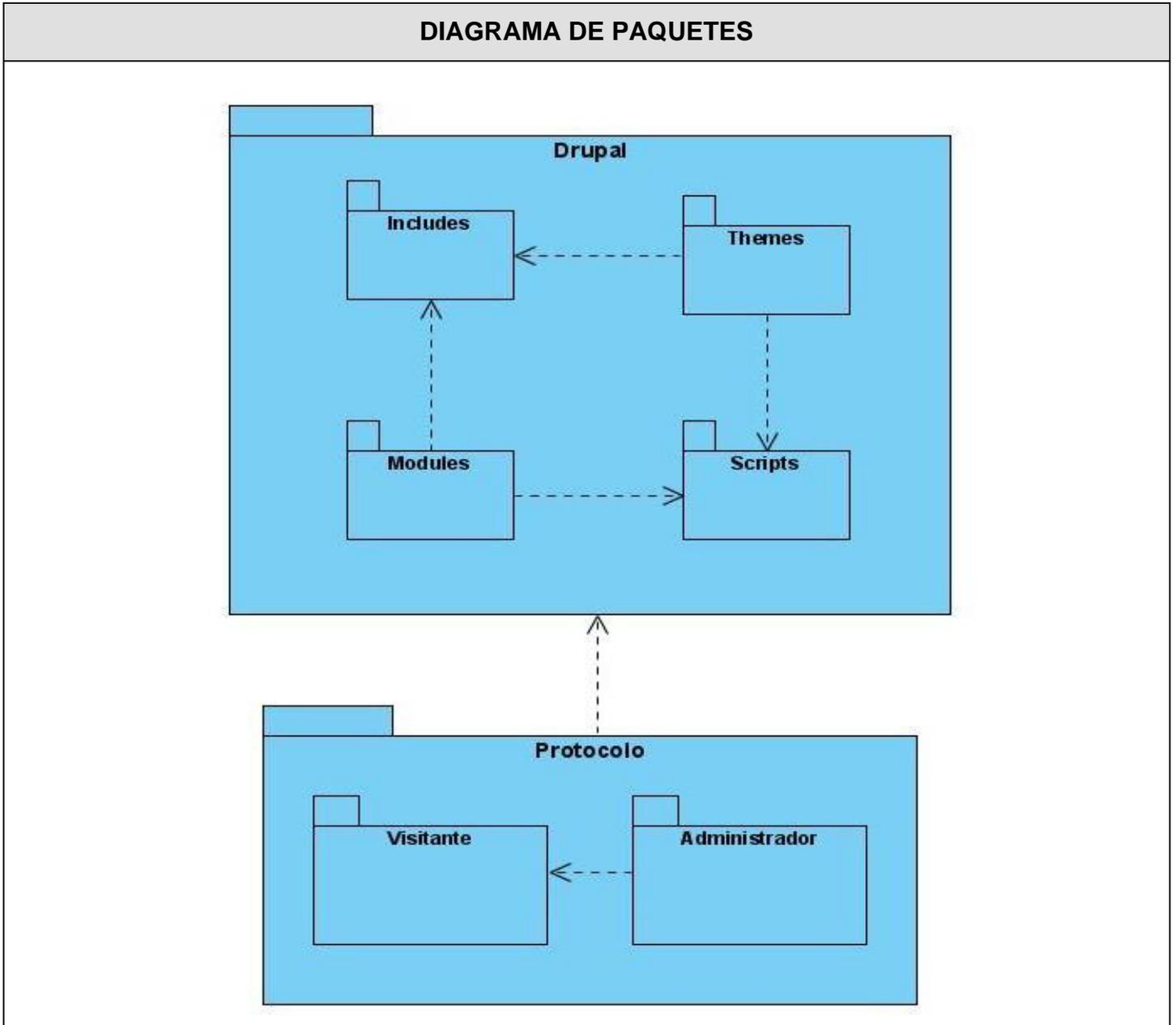
**Themes:** Incluye las distintas plantillas, cuando se desee incluir un nuevo diseño, se copia la plantilla dentro de esta carpeta.

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

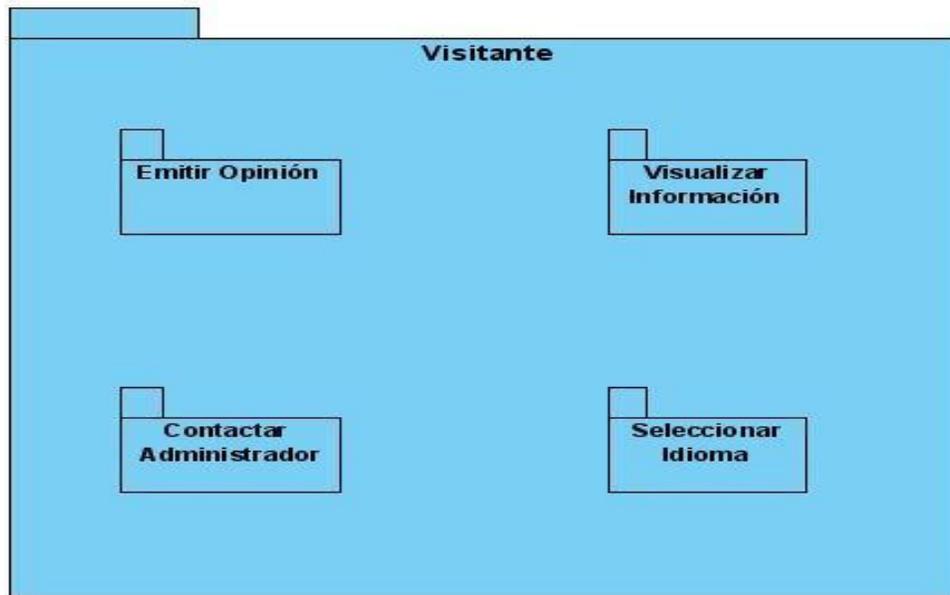
**Modules:** Incluye todos los módulos, que permiten las distintas funcionalidades del CMS, cuando desee incorporar un nuevo módulo sólo se tiene que copiar en dicha carpeta.

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

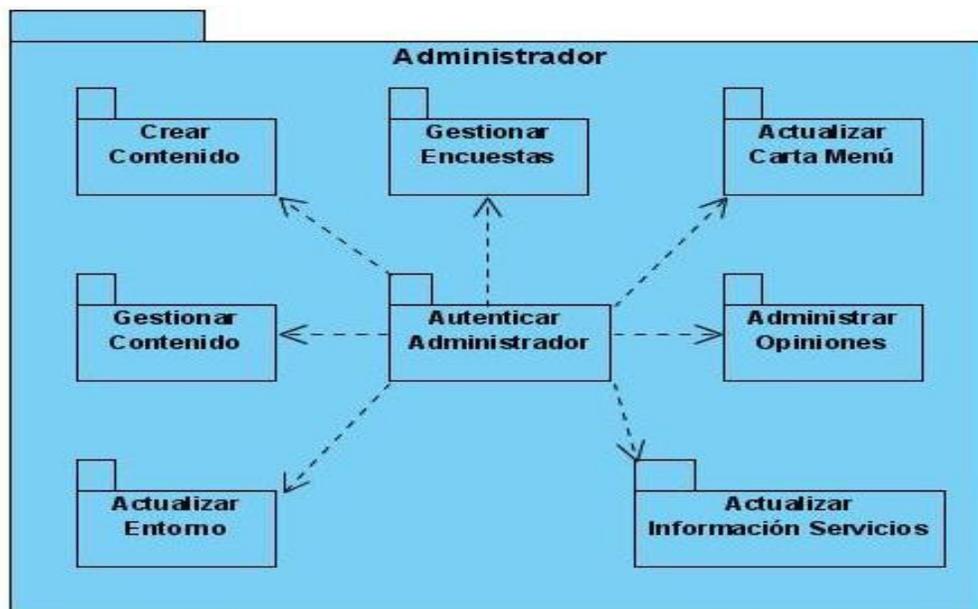
Luego de explicado el paquete Drupal posteriormente se muestra el diagrama de paquetes donde el paquete Protocolo está relacionado con el núcleo de dicho paquete.



El paquete Protocolo contiene dos paquetes: el paquete visitante y el paquete administrador. A continuación se muestran individualmente cada uno de estos paquetes y los paquetes que contienen.

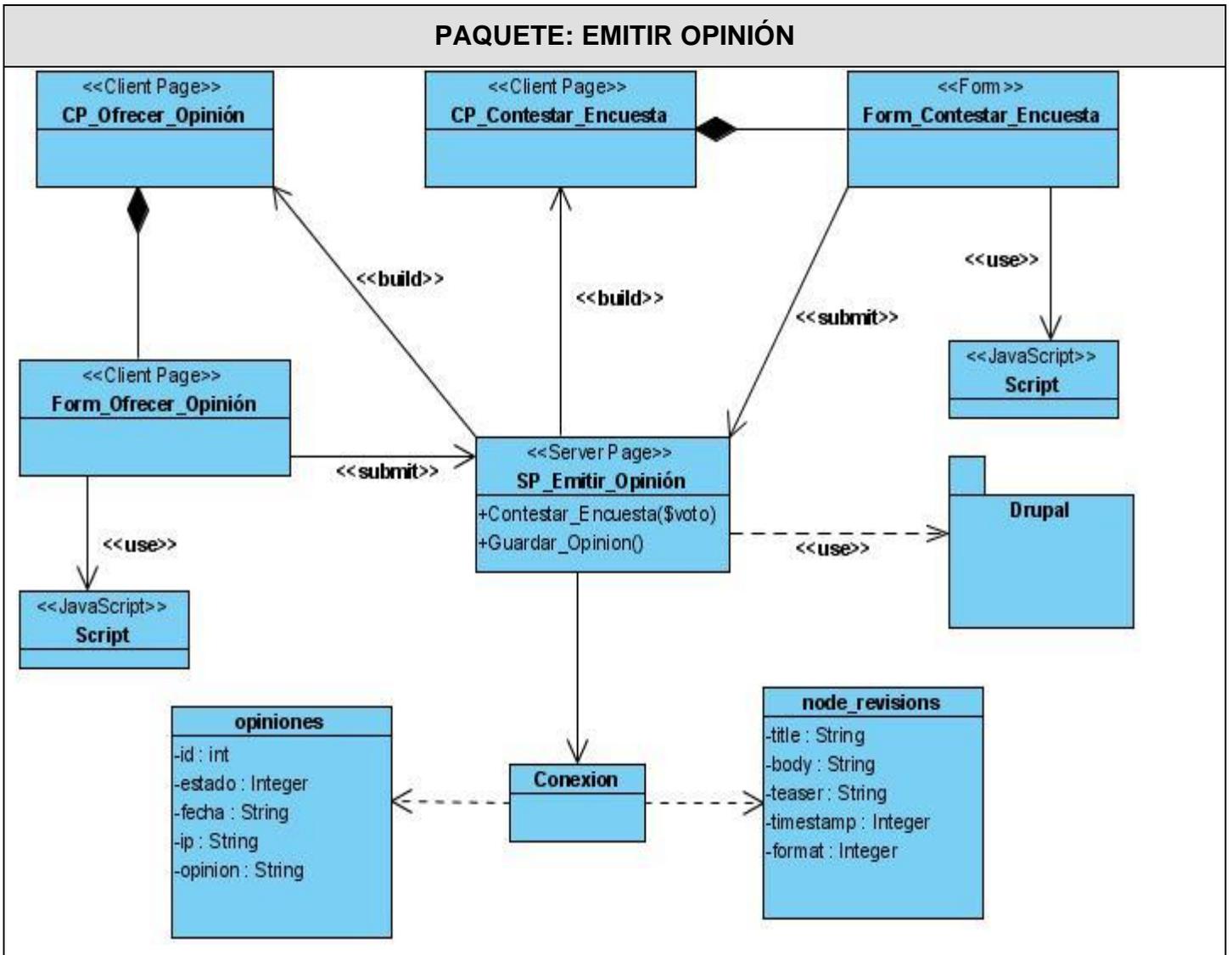


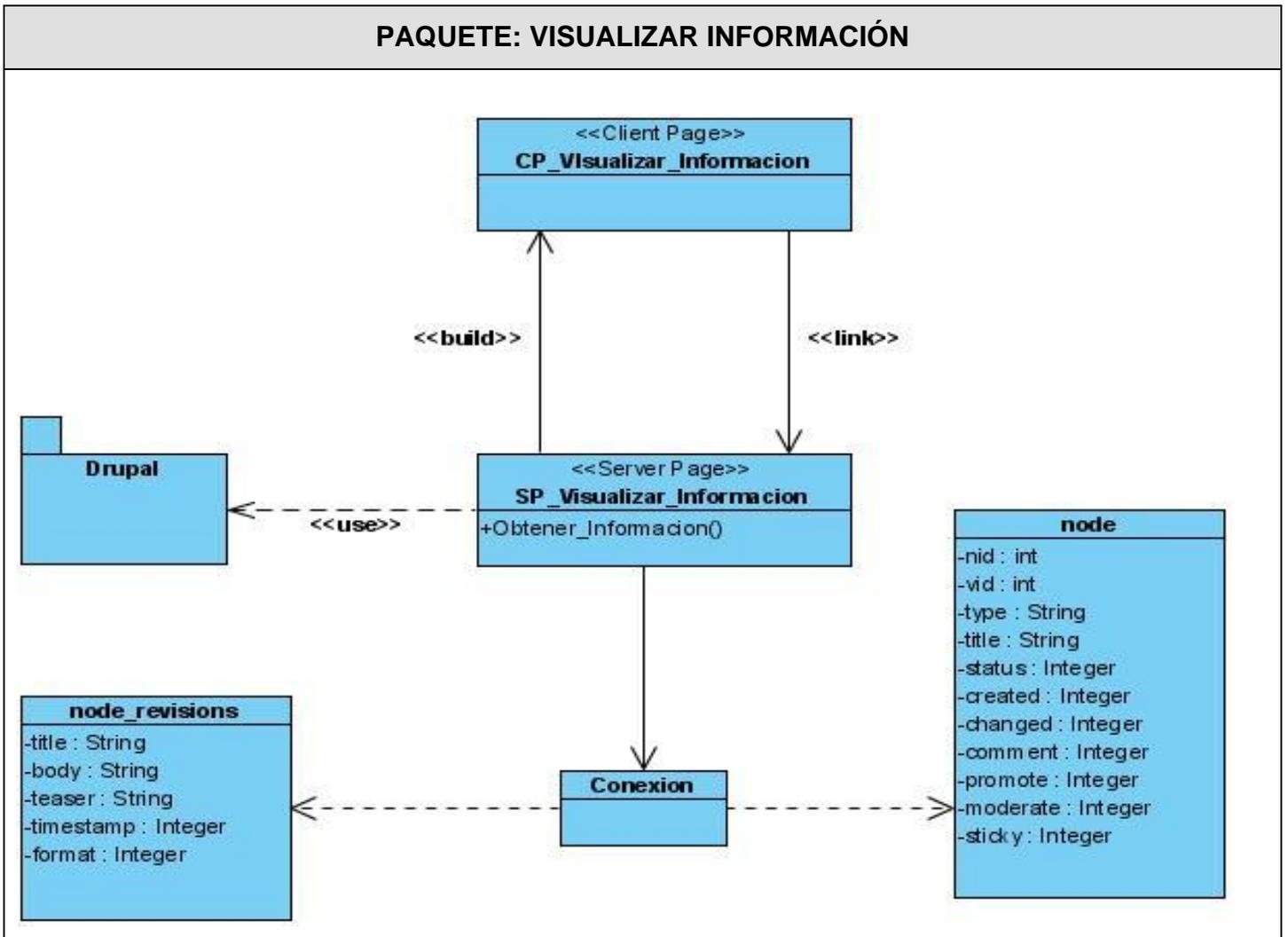
El paquete Visitante contiene los paquetes: Emitir Opinión, Visualizar Información, Contactar Administrador y Seleccionar Idioma.

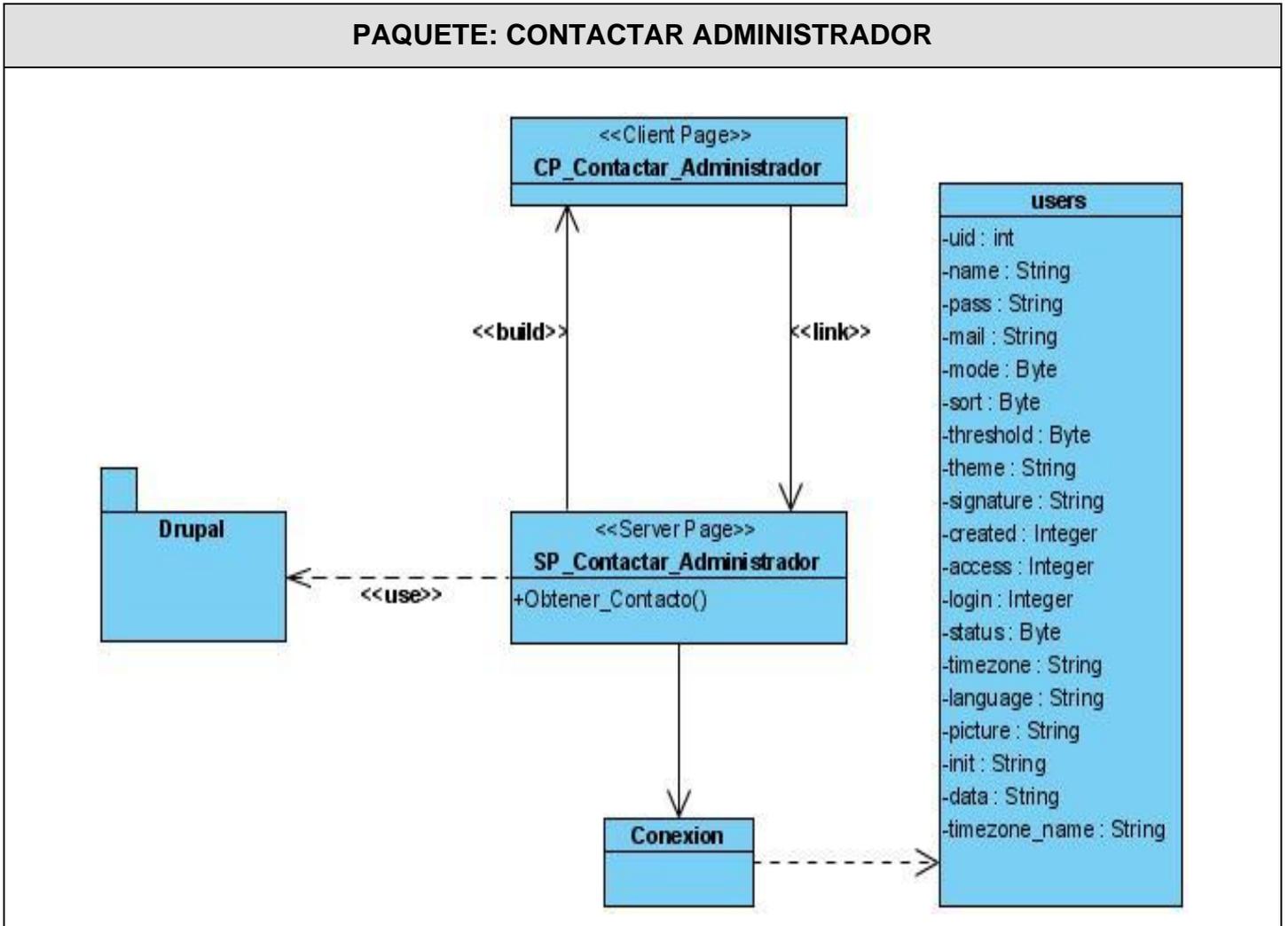


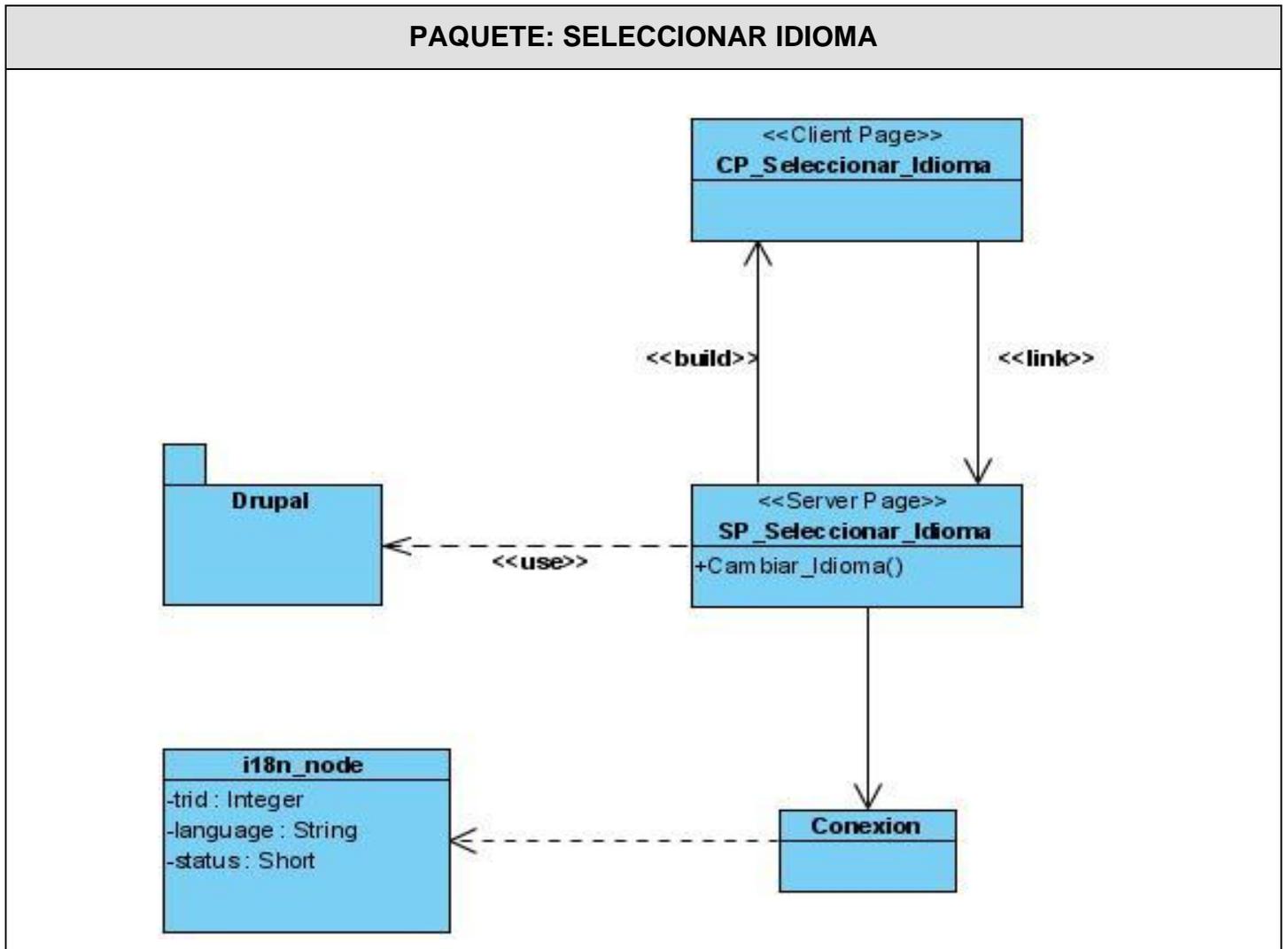
El paquete Administrador contiene los paquetes: Crear Contenidos, Gestionar Encuestas, Actualizar Carta Menú, Gestionar Contenido, Administrar Opiniones, Actualizar Entorno, Actualizar Información de los Servicios y todos se relacionan con el paquete Autenticar Administrador.

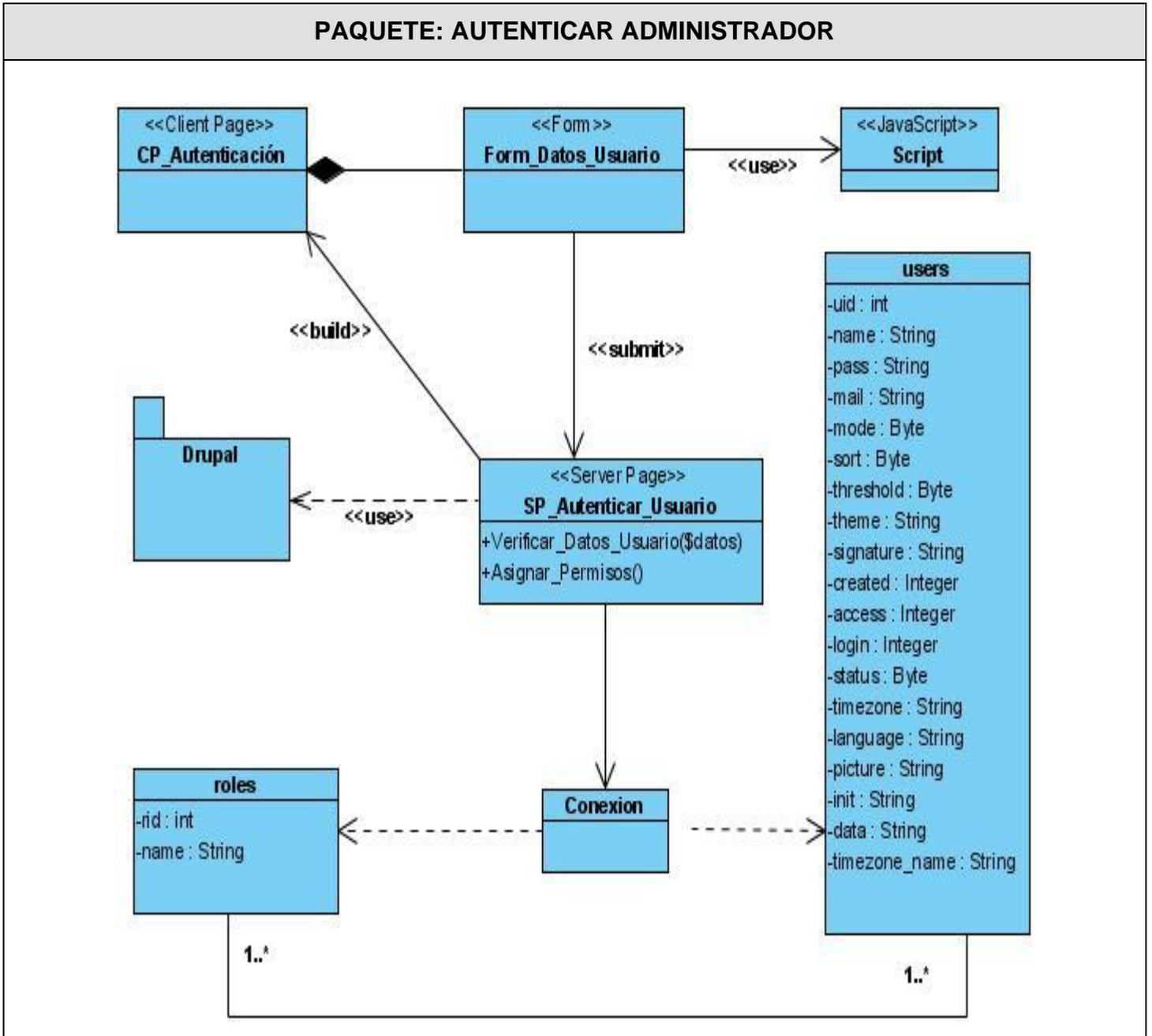
Seguidamente se muestra el diagrama de clases del diseño correspondiente a cada paquete.

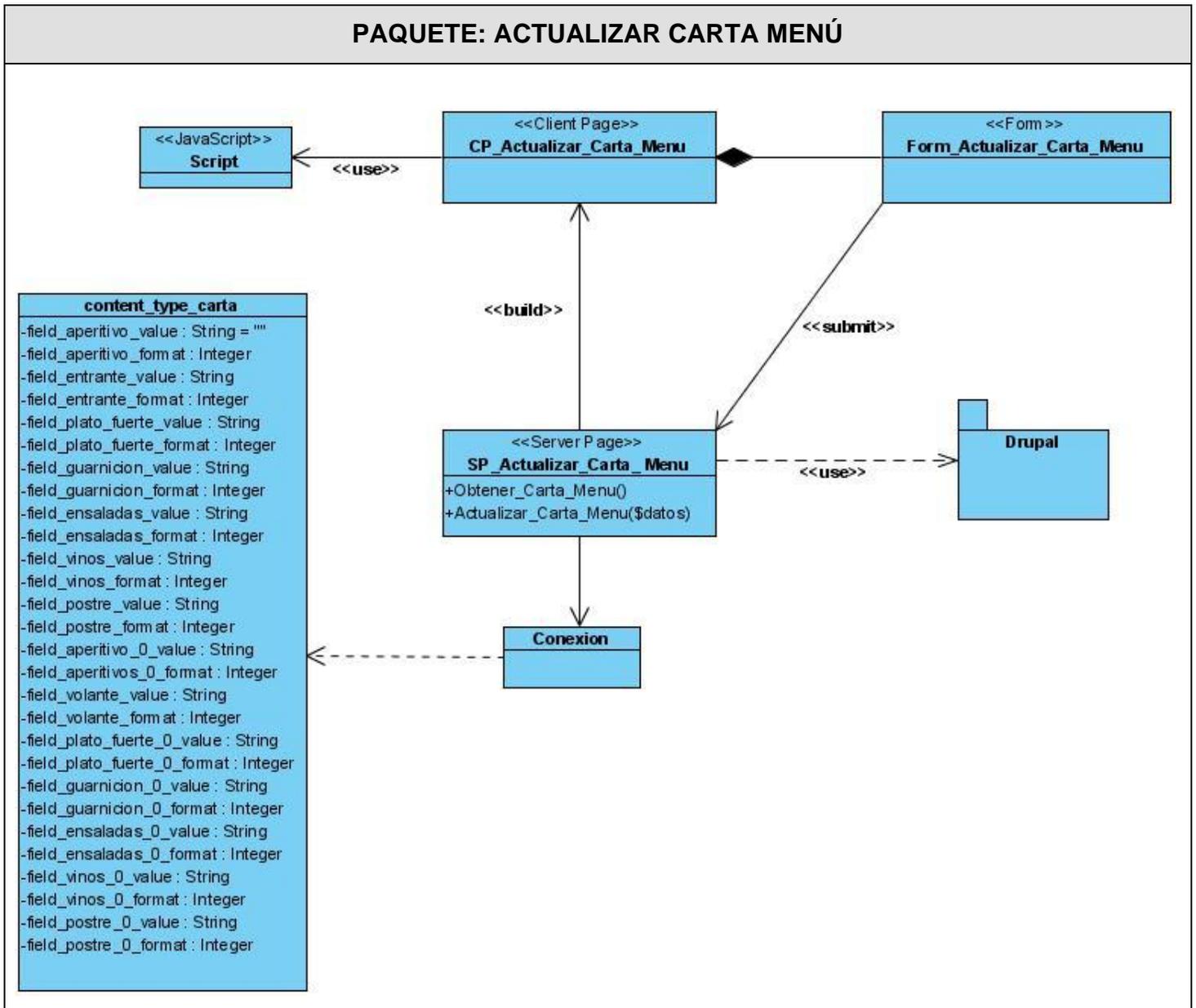


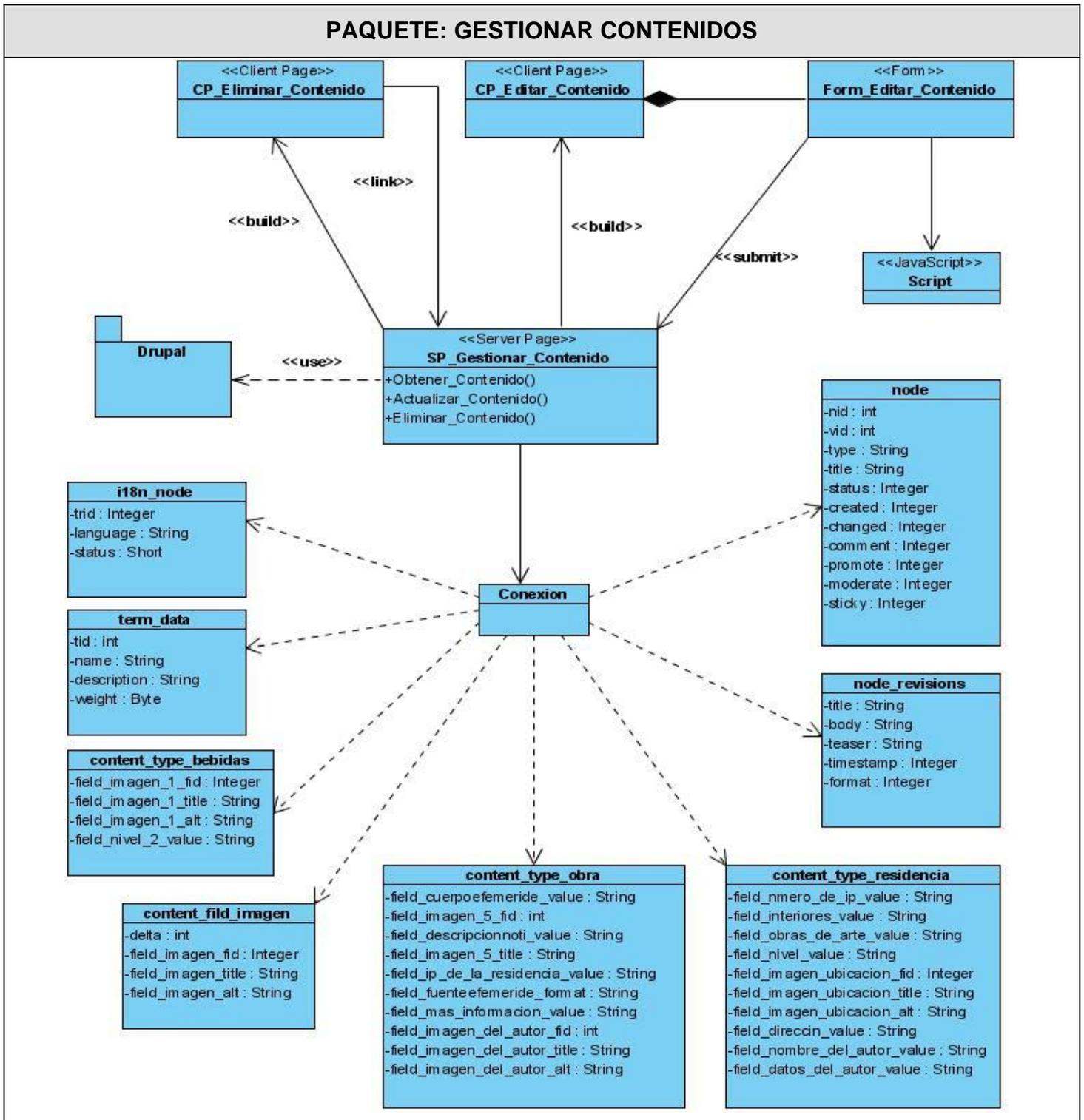


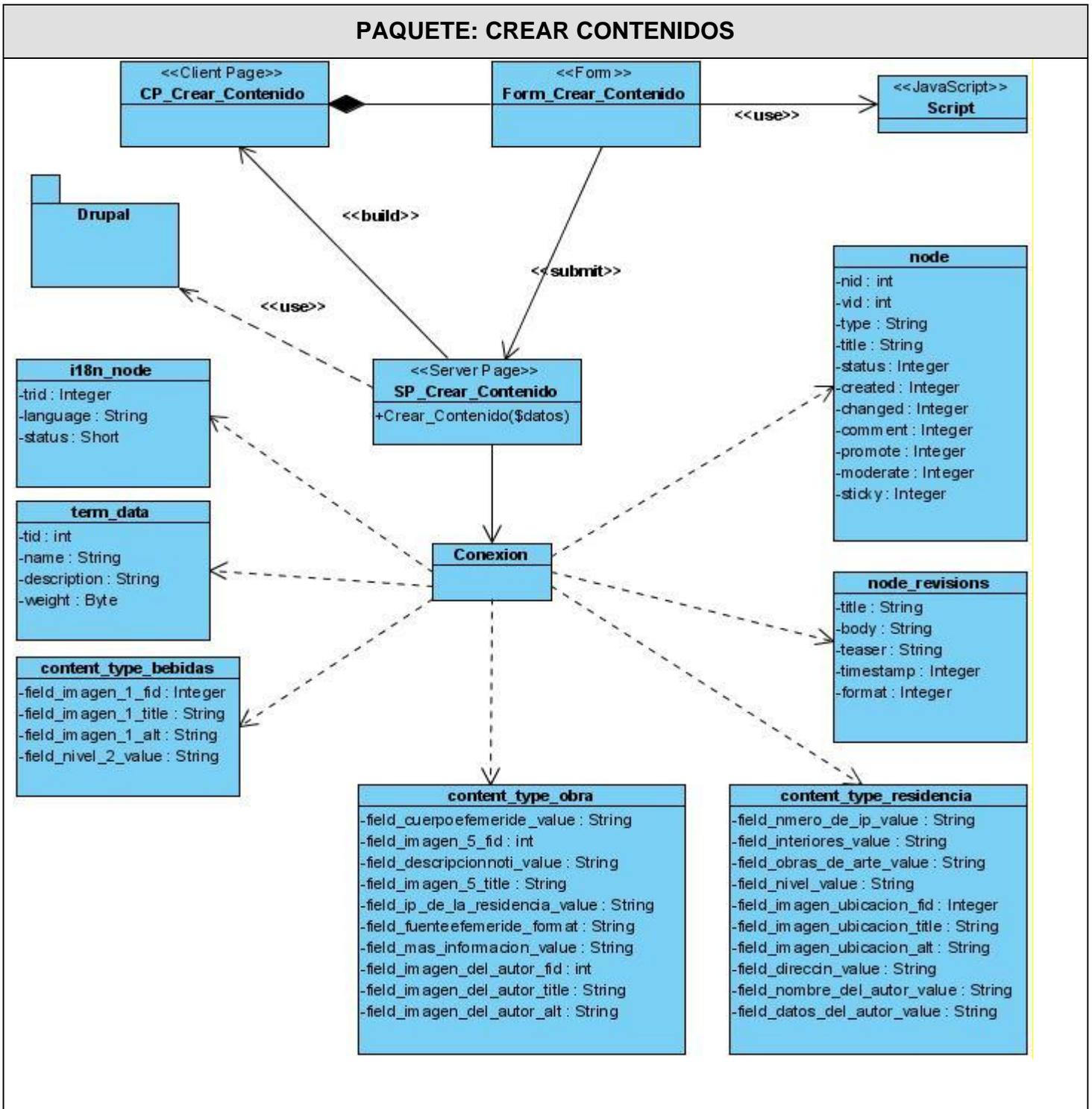


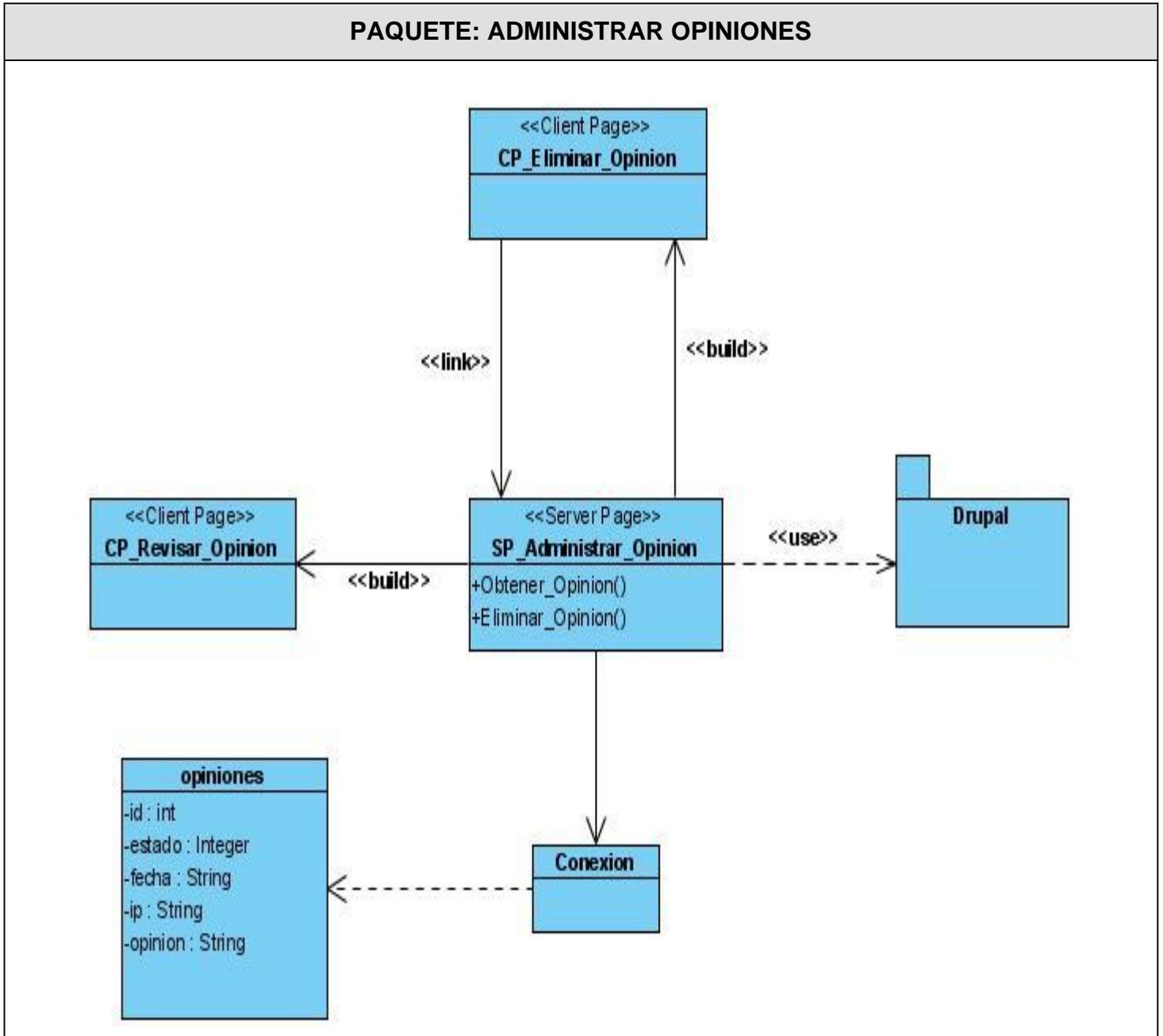


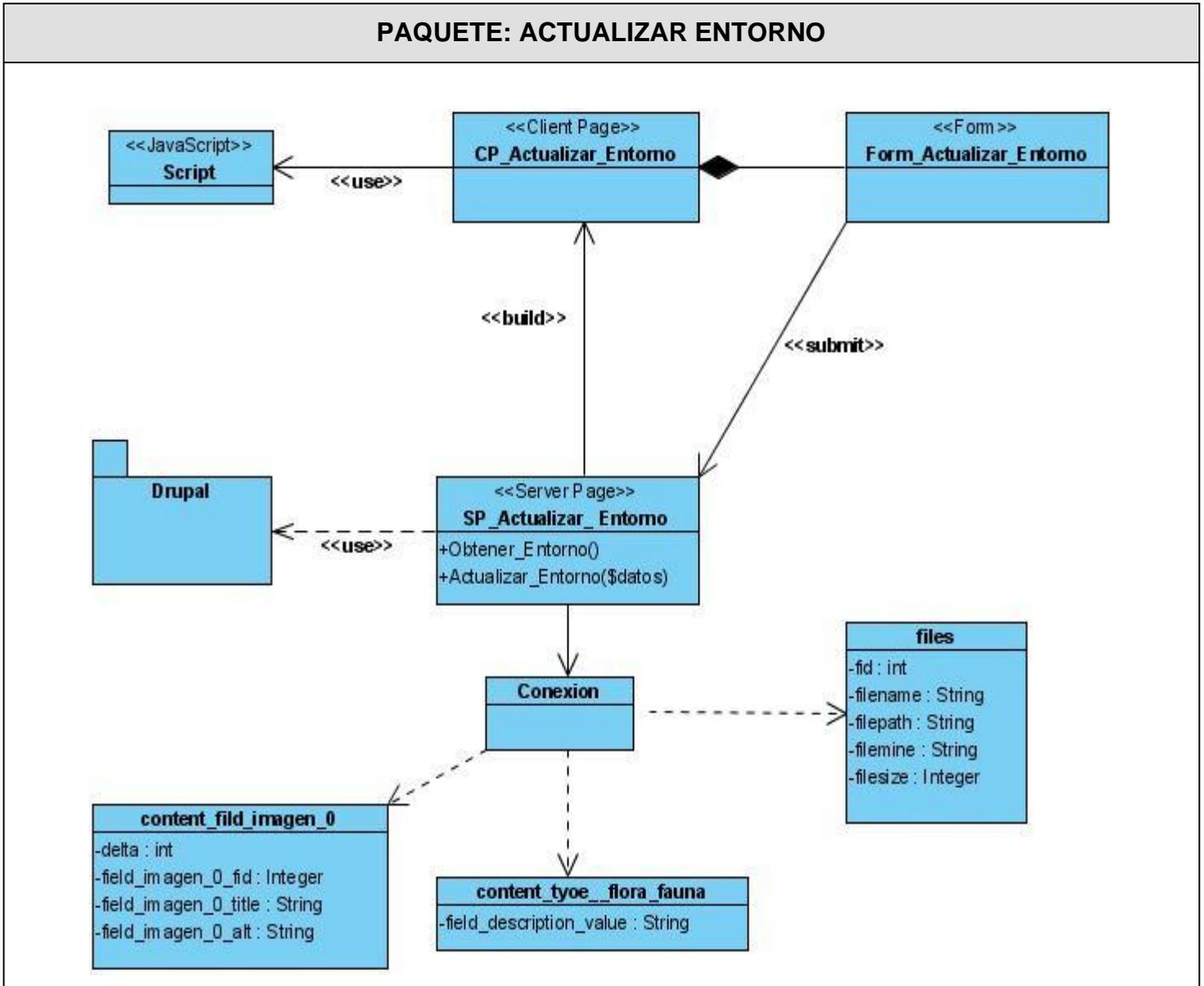


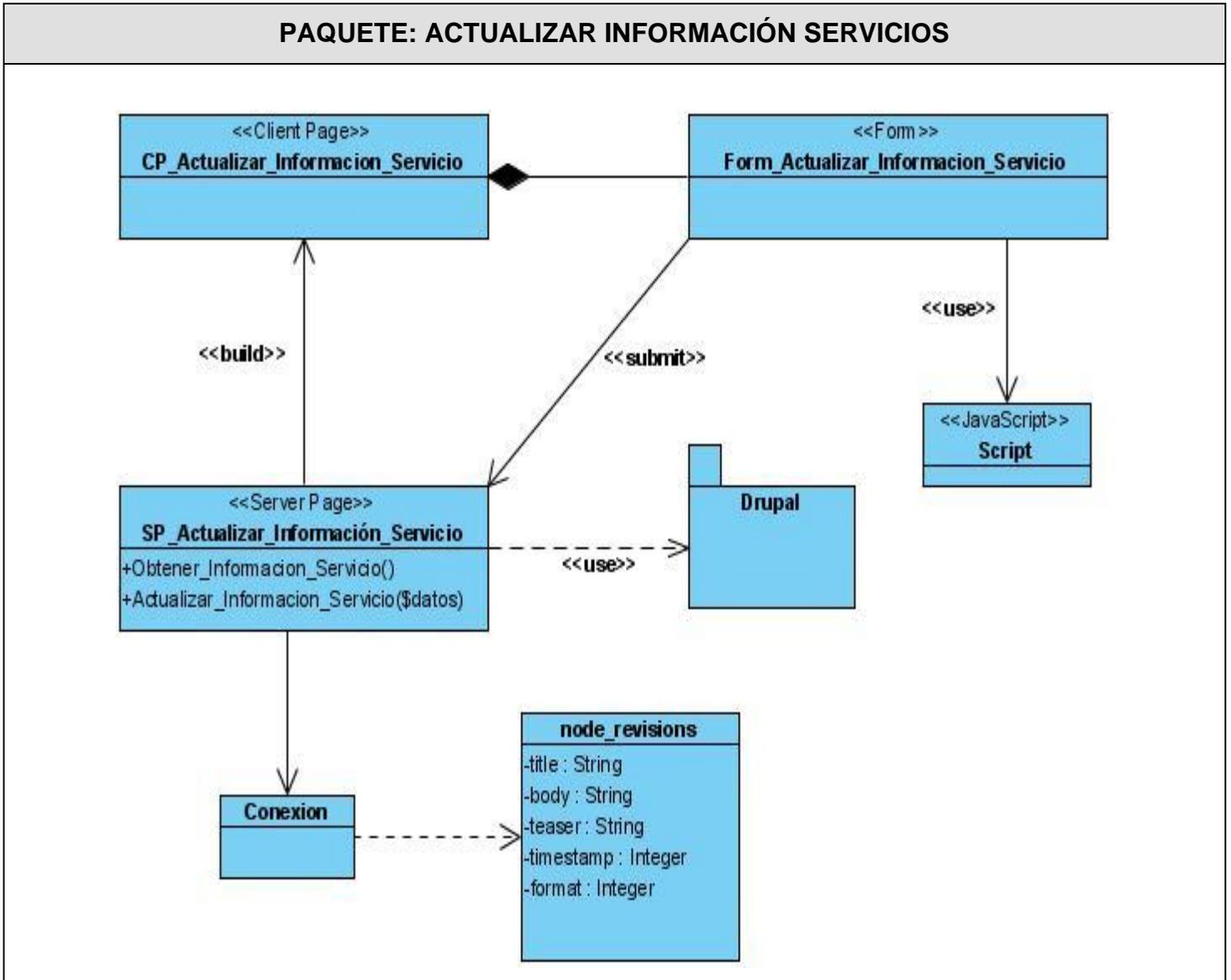


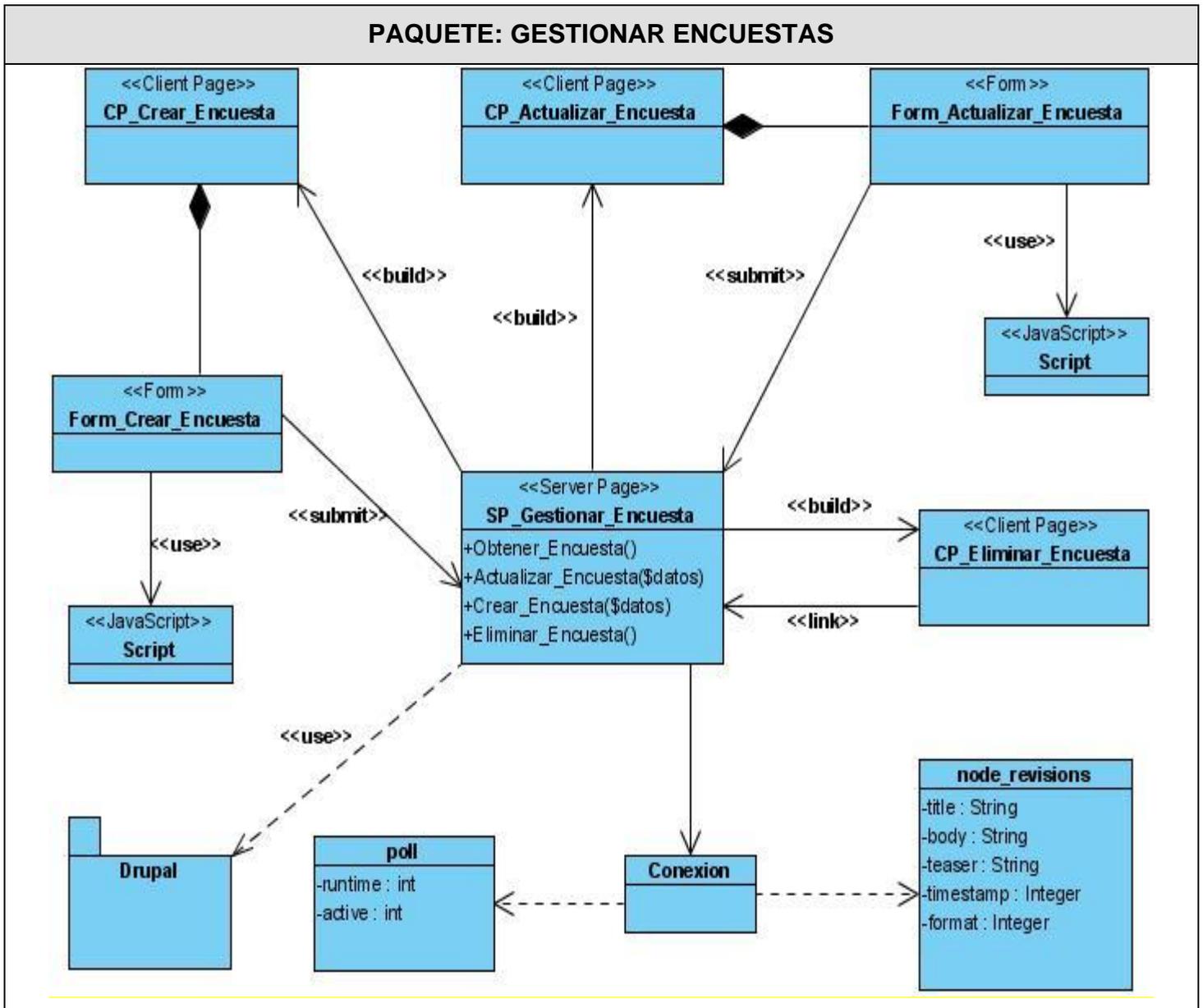










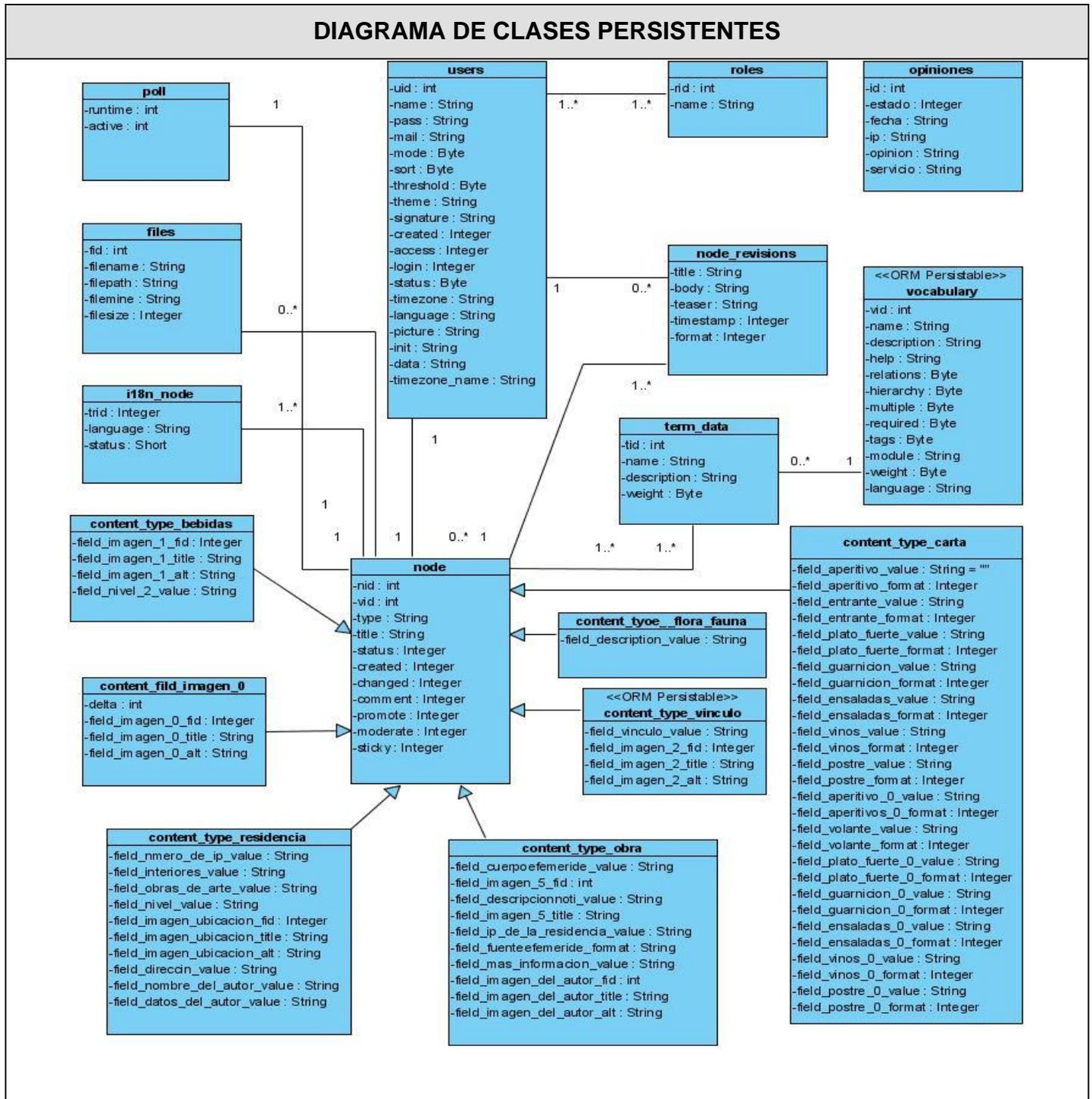


### 3.3.2 Diseño de la Base de Datos.

El objetivo principal del diseño de bases de datos es generar tablas que modelan los registros en los que se guarda la información del sistema.

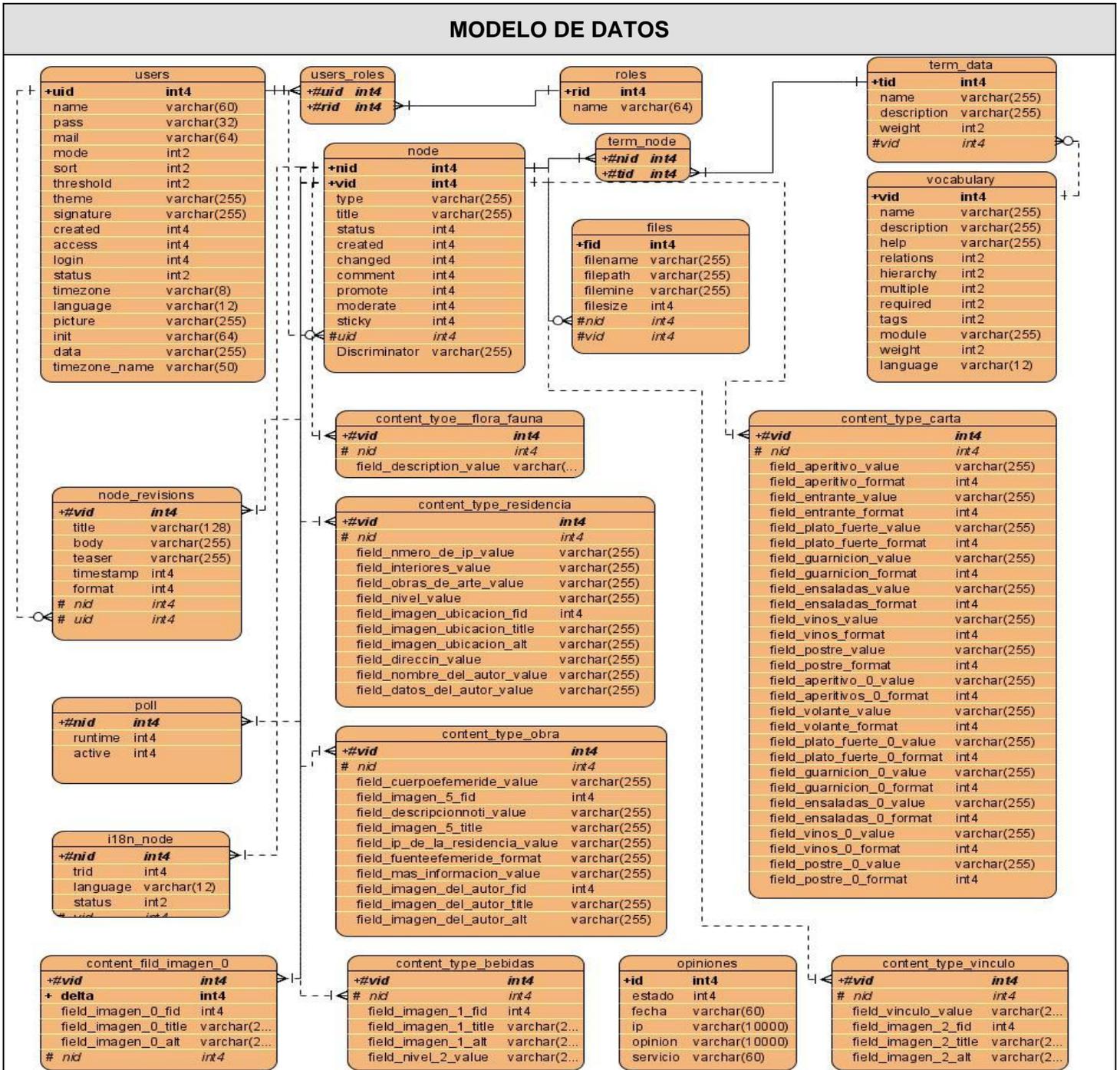
Posteriormente se ilustra el diagrama de clases persistentes correspondiente.

3.3.2.1 Diagrama de clases persistentes.



### 3.3.2.2 Modelo de datos.

A partir del diagrama de clases persistentes se obtuvo el modelo de datos que se muestra a continuación:



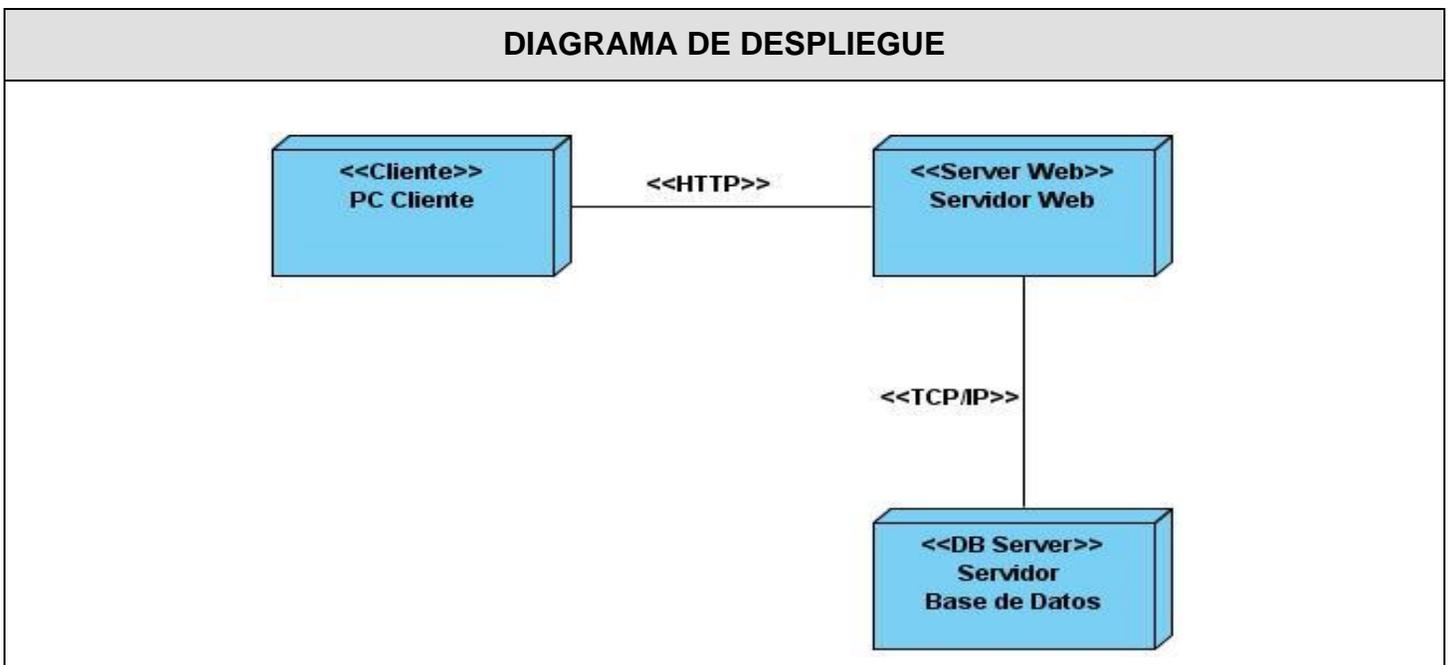
### 3.3.2.3 Descripción de las tablas.

Luego de ilustrado el diagrama de clases persistentes y el modelo de datos, se procede a describir cada una de las tablas de la base de datos, donde se detallan los atributos correspondientes a cada tabla, el tipo y una breve descripción de los mismos. ([Ver Anexos](#))

### 3.4 Diagrama de Despliegue.

En el diagrama de despliegue se muestra cómo y dónde se desplegará el sistema. Las máquinas físicas y los procesadores se representan como nodos, y la construcción interna puede ser representada por nodos o artefactos embebidos. Los estereotipos permiten precisar la naturaleza del equipo: dispositivos, procesadores y memoria. (ANÓNIMO, 2007)

A continuación se muestra el diagrama de despliegue correspondiente a la aplicación, donde se puede observar que la misma se encontrará desplegada en una unidad de procesamiento, contenida dentro del Servidor de Aplicaciones Web (Apache). La base de datos estará desplegada en una unidad de procesamiento corriendo el servicio de MySQL y la asociación entre los nodos representa la ruta de comunicación entre ellos.



### **3.5 Conclusiones**

En este capítulo se describió el flujo de trabajo Análisis y Diseño, se mostraron los diagramas de clases tanto del análisis como del diseño, así como el diagrama de colaboración correspondiente a cada caso de uso del sistema, se realizó el diseño de la base de datos, se concedió el diagrama de clases persistentes y el modelo de datos, se describieron las tablas y finalmente se ilustró el diagrama de despliegue.

## Conclusiones

Con el desarrollo de este trabajo se lograron los objetivos trazados:

- Se realizó un estudio profundo sobre los portales web y las herramientas básicas para su desarrollo, lográndose en tal sentido conocer las tendencias actuales y la selección de las herramientas y tecnologías más apropiadas para su implementación.
- La documentación se elaboró de forma eficiente siguiendo la metodología RUP para el proceso de desarrollo del software, por ser la más adecuada ante el problema planteado.
- Se obtuvo un diseño que cumple con los requerimientos del modelado y sirve de base para la futura implementación de la intranet, donde la calidad se corresponde con el nivel de la institución cliente y los visitantes que la prestigian.

Lo antes planteado le da cumplimiento al objetivo principal: Modelar una aplicación web correspondiente a la Intranet de las Residencias de Protocolo del Consejo de Estado de la República de Cuba para mejorar la gestión de información y la satisfacción de huéspedes y directivos.

## Recomendaciones

Con el objetivo de disponer de una aplicación que automatice el proceso de gestión de la información en las Casas de Protocolo se recomienda:

- Realizar los flujos de trabajo siguientes al Análisis y Diseño para lograr la aplicación que se requiere y una documentación más completa.
- Continuar con la investigación para aumentar las funcionalidades en vista de obtener mejoras futuras.
- Mantenerse actualizado en el uso de las nuevas tecnologías y herramientas con el objetivo de permanecer siempre en la competencia.
- Incluir al diseño un buscador de contenido para facilitar el manejo de la información.
- Se recomienda que el trabajo sea tomado como referencia para aquellas instituciones que requieran una aplicación similar.

## Referencias Bibliográficas

**CASTILLO, O, FIGUEROA, D y SEVILLA, H. 2009.** Tripod. [En línea] 2009. [Citado el: 15 de Febrero de 2009.] <http://programacionextrema.tripod.com/fases.htm>.

**CATALDI, Z. 2000.** *Metodología de diseño, desarrollo y evaluación de software educativo*. Facultad de Informatica.UNLP. 2000. Tesis de Magíster en Informática. ISBN 960-34-0204-2.

**CUERDA, X, MINGUILLAN, J. 2004.** mosaic. [En línea] 2004. [Citado el: 30 de Enero de 2009.] articulo. <http://mosaic.uoc.edu/articulos/cms1204.html>.

**GONZALEZ, C. 2009.** usabilidadWeb. [En línea] 2009. [Citado el: 4 de Febrero de 2009.] Base de Datos PostgreSQL,SQL avanzado y PHP. <http://www.usabilidadweb.com.ar/postgre.php>.

**JACOBSON, I. 1999.** *El proceso Unificado de Desarrollo de Software*. 1999. ISBN: 84-7829-036-2.

**MERELO, J. 2005.** Geneaura. [En línea] 2005. [Citado el: 30 de Enero de 2009.] <http://geneura.ugr.es/~jmerelo/tutoriales/cms/>.

**ANÓNIMO, 2008.** Openpopuli. [En línea] 2008. [Citado el: 7 de Abril de 2009.] Estructura MVC. [http://www.openpopuli.com/manual/estructura\\_mvc.php](http://www.openpopuli.com/manual/estructura_mvc.php).

**ANÓNIMO, 2007.** Sparxsystems. [En línea] 2007. [Citado el: 19 de Mayo de 2009.] <http://www.sparxsystems.com.ar/download/ayuda/index.html?deploymentdiagram.htm>.

**PRESSMAN, R S. 1998.** *Ingeneria de Software,un enfoque practico,Addison Wesley*. 1998. ISBN: 0-201-17888-5.

**QUIÑONES y E. 2007.** [En línea] 2007. [Citado el: 6 de Febrero de 2009.] Introduccion a PostgreSQL. [http://www.postgresql.org.pe/articles/introduccion\\_a\\_postgresql.pdf](http://www.postgresql.org.pe/articles/introduccion_a_postgresql.pdf).

**VILLANUEVA, A. 2007.** Desarrollando web. [En línea] 2007. [Citado el: 30 de Enero de 2009.] <http://www.desarrollandoweb.com/tipos-de-portales.php>.

## Bibliografía

**AGUILAR, V. 2007.** fedora-es. [En línea] 2007. [Citado el: 6 de Febrero de 2009.] Manual de MySQL vs. PostgreSQL. <http://www.fedora-es.com/node/189>.

**ALVAREZ, M. 2003.** DesarrolloWeb. [En línea] 2003. [Citado el: 3 de Febrero de 2009.] <http://www.desarrolloweb.com/articulos/1178.php>.

**ANONIMO. 2008.** Openpopuli. [En línea] 2008. [Citado el: 7 de Abril de 2009.] Estructura MVC. [http://www.openpopuli.com/manual/estructura\\_mvc.php](http://www.openpopuli.com/manual/estructura_mvc.php).

**ANÓNIMO, 2007.** Sparxsystems. [En línea] 2007. [Citado el: 19 de Mayo de 2009.] <http://www.sparxsystems.com.ar/download/ayuda/index.html?deploymentdiagram.htm>.

**B, BOS. 2005.** W3C. [En línea] 2005. [Citado el: 3 de Febrero de 2009.] Web Style Sheets. <http://www.w3.org/Style>.

**BLANCO, D, CASTRO, M. 2007.** *Diseño de la intranet de Protocolo del Consejo de Estado de la República de Cuba*. 2007. Tesis de Diploma.

**CASTILLO, O, FIGUEROA, D y SEVILLA, H. 2009.** Tripod. [En línea] 2009. [Citado el: 15 de Febrero de 2009.] <http://programacionextrema.tripod.com/fases.htm>.

**CATALDI, Z. 2000.** *Metodología de diseño, desarrollo y evaluación de software educativo*. Facultad de Informática.UNLP. 2000. Tesis de Magister en Informática. ISBN 960-34-0204-2.

**2006.** Ciberaula. [En línea] 2006. [Citado el: 7 de Febrero de 2009.] Una introduccion a APACHE. [http://linux.ciberaula.com/articulo/linux\\_apache\\_intro/](http://linux.ciberaula.com/articulo/linux_apache_intro/).

**CUERDA, X, MINGUILLAN, J. 2004.** mosaic. [En línea] 2004. [Citado el: 30 de Enero de 2009.] articulo. <http://mosaic.uoc.edu/articulos/cms1204.html>.

**CUZCANO, S. 2008.** Scribd. [En línea] 2008. [Citado el: 26 de Febrero de 2009.] Modelo de Análisis. <http://www.scribd.com/doc/7930106/Modelo-de-Analisis-Saul-Cuzcano-Quintin>.

**FERNANDEZ, F. 2006.** Desarrollo de páginas web en Linux. [En línea] 2006. [Citado el: 2 de Febrero de 2009.] <http://mundopc.net/articulos/desarrollo-de-paginas-web-en-linux/>.

**FERNANDEZ, G. 2002.** *Introduccion a Extreme Programming. Ingenieria de software II.* 2002.

**GONZALEZ, C. 2009.** usabilidadWeb. [En línea] 2009. [Citado el: 4 de Febrero de 2009.] Base de Datos PostgreSQL,SQL avanzado y PHP. <http://www.usabilidadweb.com.ar/postgre.php>.

**JACOBSON, I. 1999.** *El proceso Unificado de Desarrollo de Software.* 1999. ISBN: 84-7829-036-2.

**KRUCHTEN, P. 1996.** *A Rational Development Process.* 1996.

**LAFFONE, E. 2005.** KDE Documentation. [En línea] 2005. [Citado el: 3 de Febrero de 2009.] Manual de usuario de Quanta PLus. <http://docs.kde.org/kde3/es/kdewebdev/quanta/introduction-3-2.html>.

**MARTINEZ, I. 2007.** Editores Web-Quanta Plus. [En línea] 2007. [Citado el: 3 de Febrero de 2009.] <http://indira-informatica.blogspot.com/2007/04/editores-web-quanta-plus.html>.

**MERELO, J. 2005.** Geneaura. [En línea] 2005. [Citado el: 30 de Enero de 2009.] <http://geneura.ugr.es/~imerelo/tutoriales/cms/>.

**PRESSMAN, R S. 1998.** *Ingenieria de Software,un enfoque practico,Addison Wesley.* 1998. ISBN: 0-201-17888-5.

**QUIÑONES y E. 2007.** [En línea] 2007. [Citado el: 6 de Febrero de 2009.] Introduccion a PostgreSQL. [http://www.postgresql.org.pe/articles/introduccion\\_a\\_postgresql.pdf](http://www.postgresql.org.pe/articles/introduccion_a_postgresql.pdf).

**2007.** *Resultados de la Revisión del Sitio Web de Residencias de Protocolo.* 2007. p. 6.

**ROBLES, E. 2009.** Slideshare. [En línea] 2009. [Citado el: 25 de Febrero de 2009.] El Flujo Análisis. <http://www.slideshare.net/juliopari/13-clase-flujo-de-analisis>.

**2009.** Sitio Oficial de PHP. [En línea] 2009. [Citado el: 2 de Febrero de 2009.] <http://www.php.net>.

**2009.** Sitio Oficial MySQL. [En línea] 2009. [Citado el: 6 de Febrero de 2009.] Principales características de MySQL. <http://dev.mysql.com/doc/refman/5.0/es/features.html>.

**2008.** Sitio Oficial Netcraft. [En línea] 2008. [Citado el: 6 de Febrero de 2009.] Web Server Survey. [http://news.netcraft.com/archives/2008/12/24/december\\_2008\\_web\\_server\\_survey.html](http://news.netcraft.com/archives/2008/12/24/december_2008_web_server_survey.html).

**2009.** Sitio Oficial PostgreSQL. [En línea] 2009. [Citado el: 6 de Febrero de 2009.] <http://www.postgresql.org/>.

**2009.** Sitio Oficial Servidores Dedicados. [En línea] 2009. [Citado el: 13 de Mayo de 2009.] <http://www.servidoresdedicados.com>

**VILLANUEVA, A. 2007.** Desarrollando web. [En línea] 2007. [Citado el: 30 de Enero de 2009.] <http://www.desarrollandoweb.com/tipos-de-portales.php>.

**2009.** W3C. [En línea] 2009. [Citado el: 3 de Febrero de 2009.] Guia Breve de CSS. <http://www.w3c.es/divulgacion/guiasbreves/HojasEstilo>.

## Anexos

### Descripciones de las tablas de la base de datos.

<b>Nombre:</b>	Node	
<b>Descripción:</b>	Contiene todos los contenidos de la intranet. Es una generalización de las tablas: content_type_bebidas, content_type_carta, content_type_flora_fauna, content_type_vinculo, content_type_obra, content_type_residencia, content_field_imagen, content_field_imagen_obras y content_field_imagen_0.	
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
nid	Int	Identificador de la tabla node
vid	Int	Identificador de vocabulario
type	Varchar	Tipo de contenido
tittle	Varchar	Título del contenido
status	Int	Identifica el estado de publicación del contenido
created	Int	Fecha en la que se creó el contenido
changed	Int	Fecha en la que se modificó el contenido
comment	Int	Se permite el envío o no de comentarios referente a un contenido
promote	Int	Indica el grado de promoción de un contenido, en dependencia del valor que tenga el contenido, aparece o no en la página principal.
moderate	Int	Indica si el contenido se encuentra en la cola de moderación
sticky	Int	Indica si el contenido tiene prioridad sobre otros contenidos.
uid	Int	Identificador de la tabla user

<b>Nombre:</b>	content_type_bebidas	
<b>Descripción:</b>	Contiene información de las bebidas. Es una especialización de la tabla node	
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
vid	int	Identificador de vocabulario
nid	int	Identificador de la tabla node
field_imagen_1_fid	int	Identificador de imagen
field_imagen_1_title	varchar	Nombre de la imagen
field_imagen_1_alt	varchar	Texto alternativo de la imagen
field_nivel_2_value	varchar	Nivel de la bebida

<b>Nombre:</b>	content_type_carta	
<b>Descripción:</b>	Contiene toda la información de las del menú del día. Es una especialización de la tabla node	
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
vid	int	Identificador de vocabulario
nid	int	Identificador de la tabla node
field_aperitivo_value	varchar	Contiene al valor del aperitivo del almuerzo
field_aperitivo_format	int	Valor para filtrar el contenido del campo de texto aperitivo del almuerzo
field_entrante_value	varchar	Contiene al valor entrante del almuerzo
field_entrante_format	int	Valor para filtrar el contenido del campo de texto entrante del almuerzo del almuerzo
field_plato_fuerte_value	varchar	Contiene al valor plato fuerte
field_plato_fuerte_format	int	Valor para filtrar el contenido del campo de texto plato fuerte del almuerzo
field_guarnicion_value	varchar	Contiene al valor guarnición del almuerzo
field_guarnicion_format	int	Valor para filtrar el contenido del campo de texto guarnición del almuerzo

field_ensaladas_value	varchar	Contiene al valor ensaladas del almuerzo
field_ensaladas_format	int	Valor para filtrar el contenido del campo de texto ensaladas del almuerzo
field_vinos_value	varchar	Contiene al valor vinos del almuerzo
field_vinos_format	int	Valor para filtrar el contenido del campo de texto vinos del almuerzo
field_postre_value	varchar	Contiene al valor postre del almuerzo
field_postre_format	int	Valor para filtrar el contenido del campo de texto postre del almuerzo
field_aperitivo_0_value	varchar	Contiene al valor aperitivo de la cena
field_aperitivos_0_format	int	Valor para filtrar el contenido del campo de texto aperitivo de la cena
field_volante_value	varchar	Contiene al valor volante de la cena
field_volante_format	int	Valor para filtrar el contenido del campo de texto volante de la cena
field_plato_fuerte_0_value	varchar	Contiene al valor plato fuerte de la cena
field_plato_fuerte_0_format	int	Valor para filtrar el contenido del campo de texto plato fuerte de la cena
field_guarnicion_0_value	varchar	Contiene al valor guarnición de la cena
field_guarnicion_0_format	int	Valor para filtrar el contenido del campo de texto guarnición d ella cena
field_ensaladas_0_value	varchar	Contiene al valor ensaladas de la cena
field_ensaladas_0_format	int	Valor para filtrar el contenido del campo de texto ensaladas de la cena
field_vinos_0_value	varchar	Contiene al valor vinos de la cena
field_vinos_0_format	int	Valor para filtrar el contenido del campo de texto vinos de la cena
field_postre_0_value	varchar	Contiene al valor postre de la cena
field_postre_0_format	int	Valor para filtrar el contenido del campo de texto

		postre de la cena
--	--	-------------------

<b>Nombre:</b>	content_type_flora_fauna	
<b>Descripción:</b>	Contiene la descripción del entorno. Es una especialización de la tabla node	
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
vid	int	Identificador de vocabulario
nid	int	Identificador de la tabla node
field_description_value	varchar	Descripción del entorno

<b>Nombre:</b>	content_type_vinculo	
<b>Descripción:</b>	Contiene la Información de los enlaces a otros sitios. Es una especialización de la tabla node	
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
vid	int	Identificador de vocabulario
nid	int	Identificador de la tabla node
field_vinculo_value	varchar	Dirección del sitio
field_imagen_2_fid	int	Identificador de la imagen
field_imagen_2_title	varchar	Nombre de la imagen
field_imagen_2_alt	varchar	Texto alternativo de la imagen

<b>Nombre:</b>	content_type_obra	
<b>Descripción:</b>	Contiene información de las obras de arte que se encuentran en las residencias. Es una especialización de la tabla node	
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
vid	int	Identificador de vocabulario
nid	int	Identificador de la tabla node
field_autor_value	varchar	Nombre del autor de la obra

field_imagen_5_fid	int	Identificador de la imagen de la obra
field_imagen_5_title	varchar	Nombre de la imagen
field_imagen_5_alt	varchar	Texto alternativo de la imagen
field_ip_de_la_residencia_value	varchar	Dirección IP de la Maquina donde se encuentra la residencia
field_datos_del_autor_0_value	varchar	Breve descripción del autor
field_mas_informacion_value	varchar	Dirección web donde se puede encontrar más información del autor
field_imagen_del_autor_fid	int	Identificador de la imagen del autor
field_imagen_del_autor_title	varchar	Nombre de la imagen del autor
field_imagen_del_autor_alt	varchar	Texto alternativo de la imagen del autor

<b>Nombre:</b>	content_type_residencia	
<b>Descripción:</b>	Contiene Información de la residencia. Es una especialización de la tabla node	
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
vid	int	Identificador de vocabulario
nid	int	Identificador de la tabla node
field_nmero_de_ip_value	varchar	Dirección ip de la Maquina donde se encuentra la residencia
field_interiores_value	varchar	Descripción de los interiores
field_nivel_value	varchar	Nivel de la residencia
field_imagen_ubicacion_fid	int	Identificador de la imagen de la ubicación de la residencia
field_imagen_ubicacion_title	varchar	Nombre de la imagen de la ubicación de la residencia
field_imagen_ubicacion_alt	varchar	Texto alternativo de la imagen de la ubicación de la

		residencia
field_direccin_value	varchar	Dirección de la residencia

<b>Nombre:</b>	content_field_imagen_0	
<b>Descripción:</b>	Contiene información de las imágenes del entorno. Es una especialización de la tabla node.	
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
vid	int	Identificador de vocabulario
nid	int	Identificador de la tabla node
delta	int	Identificador de cada tupla de la tabla
field_imagen_fid	varchar	Identificador de la imagen del entorno
field_imagen_title	varchar	Nombre de la imagen del entorno
field_imagen_alt	int	Texto alternativo de la imagen del entorno

<b>Nombre:</b>	opiniones	
<b>Descripción:</b>	Contiene todas las opiniones enviados por los visitantes.	
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
id	int	Identificador de opinión
estado	int	Estado de lectura de la opinión
fecha	varchar	Fecha de envió de la opinión
ip	varchar	Ip de donde fue enviada la opinión
opinion	varchar	Cuerpo de la opinión
servicio	varchar	Servicio al que pertenece la opinión

<b>Nombre:</b>	users	
<b>Descripción:</b>	Contiene toda la información referente a los usuarios.	
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>

uid	int	Identificador de la tabla users
name	varchar	Nombre de usuario
pass	varchar	Contraseña de usuario
mail	varchar	Correo electrónico de usuario
mode	tinyint	Es utilizado para llevar el control de los usuarios, es como el campo status
sort	tinyint	Rol de usuario
theme	varchar	Tema especificado para el usuario
signaure	varchar	Firma del usuario
created	int	Fecha en que se creó el usuario
access	int	Fecha de último acceso
login	int	Es la fecha y hora de la última vez que se loguea el usuario
status	tinyint	Estado del usuario (activo o bloqueado)
timezone	varchar	Zona horaria
languaje	varchar	El idioma de la interfaz del sitio que se le presenta al usuario
picture	varchar	Avatar del usuario
init	varchar	Correo electrónico del usuario
data	longtext	Información sobre la cuenta (registro)

<b>Nombre:</b>	roles	
<b>Descripción:</b>	Guarda los roles de usuarios definidos en el sistema.	
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
rid	int	Identificador de la tabla roles
name	varchar	Nombre del rol

<b>Nombre:</b>	users_roles	
<b>Descripción:</b>	Esta tabla surge a partir de la relación de mucho a mucho de la tabla users y role.	
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
uid	int	Identificador de la tabla users
rid	int	Identificador de la tabla roles

<b>Nombre:</b>	i18n_node	
<b>Descripción:</b>	Esta tabla contiene todo lo referente a los idiomas en los que se puede mostrar el contenido del sitio.	
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
nid	int	Identificador de la tabla node
trid	int	Identificador de la tabla i18n_node
lenguaje	varchar	Lenguaje disponible del contenido
status	int	Estado de los contenidos

<b>Nombre:</b>	node_revisions	
<b>Descripción:</b>	Esta tabla se encarga de almacenar los datos completos de los contenidos.	
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
nid	int	Identificador de la tabla node
vid	int	Identificador de la tabla node_revisions
uid	int	Identificador de la tabla users
title	varchar	Título del contenido
body	longtext	Cuerpo del contenido
teaser	longtext	Resumen del contenido
timestamp	int	Fecha/hora de creación del contenido
format	int	Formato del contenido, puede ser en HTML o PHP

<b>Nombre:</b>	vocabulary	
<b>Descripción:</b>	Esta tabla contiene un listado de los términos de los vocabularios.	
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
vid	int	Identificador de la tabla vocabulary
name	varchar	Nombre del vocabulario
description	varchar	Descripción de vocabulario
help	varchar	Instrucciones para presentar al usuario cuando vaya a elegir un término
relations	int	Permite términos relacionados en este vocabulario
hierarchy	int	Permite una jerarquía de árboles entre los términos de un vocabulario
multiple	int	Permite a los nodos tener más de un término del vocabulario (siempre que estén activadas)
required	int	Requerido o no
tags	int	Etiquetas
module	varchar	Módulo de una categoría
weight	int	Peso del vocabulario
lenguaje	varchar	Lenguaje de los vocabularios

<b>Nombre:</b>	term_data	
<b>Descripción:</b>	Contiene los nombres de las categorías.	
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
tid	int	Identificador de la tabla tern_data
vid	int	Identificador de la tabla vocabulary
name	varchar	Nombre de la categoría
description	varchar	Descripción de la categoría

weight	int	Peso de la categoría
--------	-----	----------------------

<b>Nombre:</b>	term_node	
<b>Descripción:</b>	Esta tabla surge a partir de la relación de muchos a muchos de la tabla node y term_data.	
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
nid	int	Identificador de la tabla node
tid	int	Identificador de la tabla tern_data

<b>Nombre:</b>	files	
<b>Descripción:</b>	Contiene Información de los archivos del sitio.	
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
fid	int	Identificador de la tabla node
nid	int	Identificador de la tabla node
vid	int	Identificador de la tabla node_revisions
filename	varchar	Nombre del archivo
filepath	varchar	Dirección del archivo
filemime	varchar	Tipo del archivo
filesize	int	Tamaño del archivo

<b>Nombre:</b>	Poll	
<b>Descripción:</b>	Esta tabla contiene datos sobre las encuestas.	
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
nid	int	Identificador de la tabla node
runtime	int	Duración de la encuesta
active	int	Si esta activada la encuesta

## Glosario de Términos

**e-107:** es un Sistema de Manipulación de Contenidos (CMS) escrito en PHP, que usa MySQL como base de datos. Es completamente gratuito y está en constante desarrollo.

Se emplea en el Capítulo 1, epígrafe 1.3.1.

**GNU/GPL:** Acrónimo en inglés de General Public License (Licencia Pública General). Es una licencia que está orientada principalmente a proteger la libre distribución, modificación y uso de software.

Se emplea en el Capítulo 1, epígrafe 1.3.1.

**MAC OSX:** Macintosh Operating System X (Sistema operativo de Macintosh diez). Actual sistema operativo de los ordenadores Macintosh. Está basado en Unix pero con un gestor de ventanas llamado Aqua, totalmente desarrollado por Apple.

Se utiliza en el Capítulo 1, epígrafe 1.3.2.

**BSD:** Acrónimo de Berkeley Software Distribution (en español, Distribución de Software Berkeley) y se utiliza para identificar un sistema operativo derivado del sistema Unix nacido a partir de los aportes realizados a ese sistema por la Universidad de California en Berkeley.

Se emplea en el Capítulo 1, epígrafe 1.3.2.

**IIS:** Acrónimo en inglés de Internet Information Server. Es una serie de servicios para los ordenadores que funcionan con Windows. Los servicios que ofrece son: FTP, SMTP, NNTP y HTTP/HTTPS.

Se emplea en el Capítulo 1, epígrafe 1.3.2.

**ODBC:** Acrónimo en inglés de Open Data Base Connectivity. Es un estándar de acceso a bases de datos desarrollado por Microsoft Corporation, el objetivo de ODBC es hacer posible el acceso a cualquier dato de cualquier aplicación, sin importar qué sistema gestor de bases de datos almacene los datos.

Se utiliza en el Capítulo 1, epígrafe 1.4.1.

**DB2:** Es una marca comercial, propiedad de IBM, bajo la cual se comercializa el sistema de gestión de base de datos. DB2 versión 9 es un motor de base de datos relacional que integra XML de manera nativa, lo que IBM ha llamado pureXML, que permite almacenar documentos completos dentro del tipo de datos xml para realizar operaciones y búsquedas de manera jerárquica dentro de éste, e integrarlo con búsquedas relacionales.

Se emplea en el Capítulo 1, epígrafe 1.4.1.

**Frameworks:** Un framework, en el desarrollo de software, es una estructura de soporte definida, mediante la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros software para ayudar a desarrollar y unir los diferentes componentes de un proyecto.

Se utiliza en el Capítulo 1, epígrafe 1.4.1.

**W3C:** Acrónimo de World Wide Web Consortium, es un consorcio internacional que produce estándares para la World Wide Web. Está dirigida por Tim Berners-Lee, el creador original de URL (Uniform Resource Locator, Localizador Uniforme de Recursos), HTTP (HyperText Transfer Protocol, Protocolo de Transferencia de HiperTexto) y HTML (Lenguaje de Marcado de HiperTexto) que son las principales tecnologías sobre las que se basa la Web.

Se emplea en el Capítulo 1, epígrafe 1.4.1.

**WHATWG:** Sus siglas significan Web Hypertext Application Technology Working Group, es el nombre de una compañía que surge de la unión de grandes empresas como Apple, Mozilla y Opera, con vistas a trabajar en el futuro estándar de HTML 5.0.

Se emplea en el Capítulo 1, epígrafe 1.4.2.

**Tags:** Etiquetas o palabras clave, se usan para clasificar los mensajes.

Se emplea en el Capítulo 1, epígrafe 1.4.2.

**World Wide Web:** También llamada WWW o Telaraña mundial. Es un sistema de comunicación de hipertexto que funciona sobre Internet.

Se utiliza en el Capítulo 1, epígrafe 1.4.2.

**Ecmascript:** es una especificación de lenguaje de programación publicada por ECMA International. El desarrollo empezó en 1996 y estuvo basado en el popular lenguaje JavaScript propuesto como estándar por Netscape Communications Corporation.

Se emplea en el Capítulo 1, epígrafe 1.4.3.

**IDE:** Acrónimo en inglés de Integrated Drive Electronics (Entorno Integrado de Desarrollo). Estándar utilizado para crear la conexión de unidades de almacenamiento (discos duros, lectores de CD) a los ordenadores.

Se utiliza en el Capítulo 1, epígrafe 1.5.

**KDE:** K Desktop Environment o Entorno de Escritorio K, es un entorno de escritorio e infraestructura de desarrollo para sistemas Unix/Linux.

Se emplea en el Capítulo 1, epígrafe 1.5.1.

**SQL:** Acrónimo en inglés de Structured Query Language (Lenguaje de Consulta Estructurado). Es un lenguaje de consulta estructurado de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones sobre las mismas.

Se utiliza en el Capítulo 1, epígrafe 1.5.1.

**XML:** Es el acrónimo de eXtensible Markup Language (lenguaje de marcado extensible).

Se utiliza en el Capítulo 1, epígrafe 1.5.1.

**C++:** Potente lenguaje de programación para el desarrollo de aplicaciones.

Se emplea en el Capítulo 1, epígrafe 1.5.1.

**API:** Acrónimo en inglés de Application Programming Interface (Interfaces de Programación de Aplicaciones). Es una interfaz de programación de la aplicación que reúne un conjunto de especificaciones de comunicación entre componentes software.

Se utiliza en el Capítulo 1, epígrafe 1.6.2.

**Release plan:** Es una planificación donde los desarrolladores y clientes establecen los tiempos de implementación ideales de las historias de usuario, la prioridad con la que serán implementadas y las historias de usuario que serán implementadas en cada versión del programa.

Se utiliza en el Capítulo 1, epígrafe 1.8.1.

**CASE:** Sus siglas significan Computer-Aided Software Engineering, son herramientas generalmente aplicadas a cualquier sistema que ayudan a automatizar el proceso de diseño y desarrollo del software.

Se utiliza en el Capítulo 1, epígrafe 1.9.

**UML:** Lenguaje Unificado de Modelado (UML, por sus siglas en inglés, Unified Modelling Language) es el lenguaje de modelado de sistemas de software más conocido en la actualidad.

Se utiliza en el Capítulo 1, epígrafe 1.9.

**CU:** Se refiere a los Casos de Usos.

Se utiliza en el Capítulo 2, epígrafe 2.8.2.