

Universidad de las Ciencias Informáticas

Facultad 1



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

Título: Sistema para la Gestión de la Información de la línea Weblog
(infoBlog v2).

Autor: Nolberto Rojas Pérez

Tutores: Ing. Lilian Cid Escalona

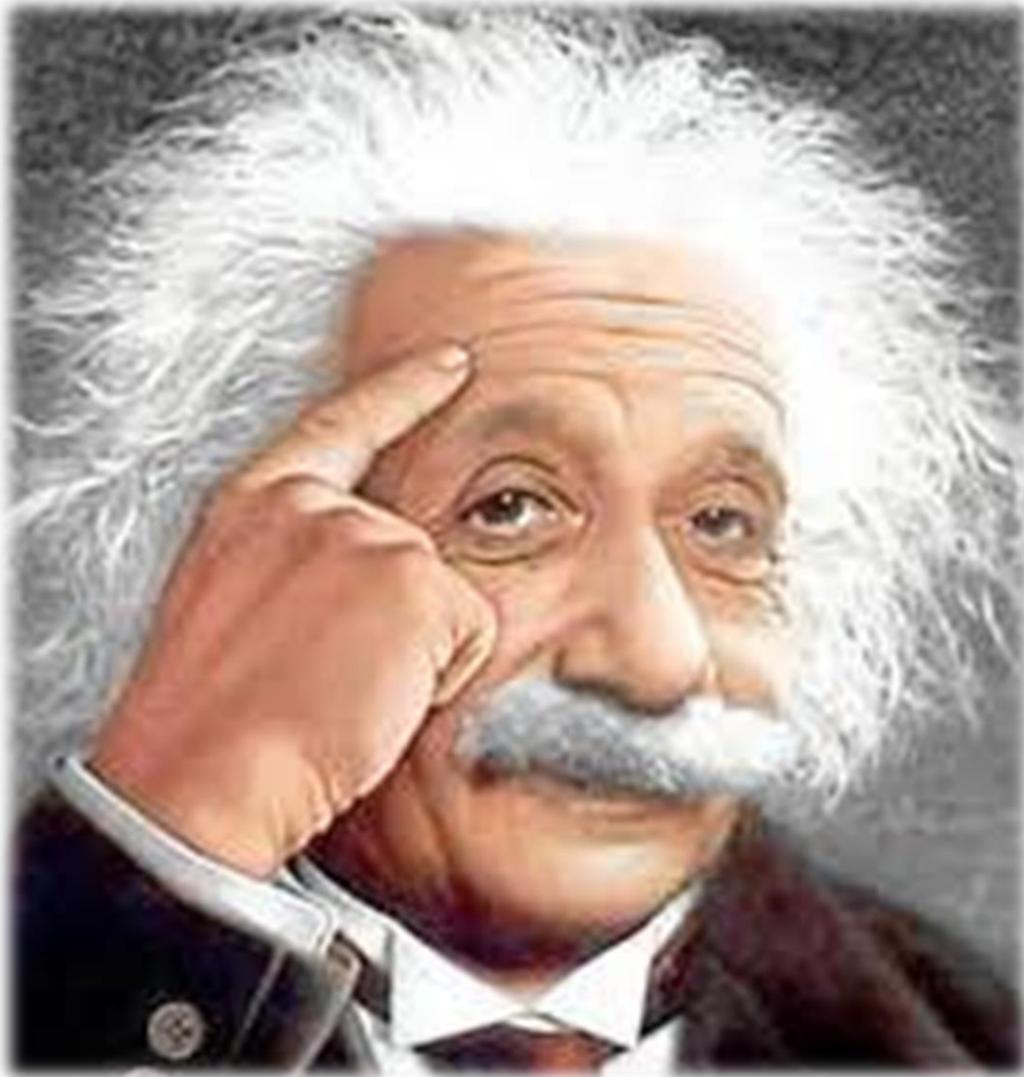
Ing. Disnayle Jorge Chacón

Consultantes: Lic. José Alberto Ponce Pérez

Ing. Randy Otero Jiménez

Ciudad de La Habana, junio 2014

“Año 56 de la Revolución”



“Si buscas resultados distintos no hagas siempre lo mismo.

El genio se hace con un 1% de talento, y un 99% de trabajo.

Nunca consideres el estudio como una obligación, sino como una oportunidad para penetrar en el bello y maravilloso mundo del saber”

Albert Einstein

DATOS DE CONTACTO

Autor: Nolberto Rojas Pérez. Estudiante en la Universidad de las Ciencias Informáticas. Estudiante de la Facultad 1, ingresando en el curso 2009-2010.

Facultad: Facultad1

Correo electrónico: nrojas@estudiantes.uci.cu

Tutor: Ing. Lilian Cid Escalona. Ingeniera en Ciencias Informáticas, UCI. Se ha desempeñado como profesor en la Disciplina de Ciencias Empresariales. Actualmente se desempeña como especialista general del departamento DOWAI de CIDI, específicamente vinculada a la línea de Weblog.

Facultad: Facultad 1

Correo electrónico: lescalona@uci.cu

Tutor: Ing. Disnayle Jorge Chacón. Ingeniera en Ciencias Informáticas, UCI. Graduada en Ciencias Informáticas del año 2010. Se ha desempeñado como Jefe de Línea. actualmente se desempeña como especialista general del departamento DOWAI del centro CIDI, específicamente vinculada a la línea de Weblog.

Facultad: Facultad 1

Correo electrónico: djorge@uci.cu

DECLARACIÓN DE AUTORÍA

Declaro ser autor único de la presente tesis que tiene por título: *Sistema para la Gestión de la Información de la línea Weblog (infoBlog v2)* y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo. Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

Ing. Lilian Cid Escalona

Firma del Tutor

Ing. Disnayle Jorge Chacón

Firma del Tutor

Nolberto Rojas Pérez

Firma del Tutor

Dedicatoria

Dedico esta tesis a mi abuela Julia López Pérez por creer siempre en mí y apoyarme en todo momento.

A mis padres Marisleidys Pérez López y Alberto Rojas Río por apoyarme en todos mis años de estudio y principalmente en estos 5 años de mi vida, y por brindarme todo lo que estaba a su alcance a cambio de nada.

A mi tío Julián Pérez López por ser uno de los que más me apoyó en todo este tiempo y sé que estará muy orgulloso de este título.

A toda mi familia, mi padrastro y amigos en general por su apoyo incondicional y a todos los que son como si fueran mi familia.

Agradecimientos

Primero que todo quiero agradecer a mis padres por todo el apoyo y consejos que me han brindado en toda mi vida.

A Eduardo Antonio Roque Díaz por guiarme, enseñarme y brindarme sus conocimientos en todo este 5to año de mi carrera.

A mis tutores y consultantes, Lilian Cid Escalona, Disnayle Jorge Chacón, José Alberto Ponce y Randy Otero Jimenez, que me han guiado y apoyado desde el primer momento que me asignaron este tema de tesis.

A mi novia Ilismay Echevarría Rodríguez por soportarme en mis altas y mis bajas, por brindarme todo el cariño y amor que necesité en todo el tiempo que llevamos juntos y por ayudarme en todo momento que necesité de ella.

A mi familia, padrastro, amistades y a los que no son mi familia que siempre me apoyaron de una u otra forma.

A mis compañeros del grupo 1503 por soportarme en todo este tiempo de estudio y brindarme su ayuda incondicional.

Al football por permitirme conocer muchas personas de todo el país y por brindarme la posibilidad de hacer nuevas amistades.

A mis compañeros del mundialito del equipo del Inter que este año somos los campeones.

RESUMEN

La gestión de la información es un proceso crítico en la línea Weblog del Departamento de Operaciones Web y Análisis de la Información del Centro de Ideoinformática de la UCI. Actualmente, en esta línea se utiliza una aplicación web llamada Info_Blog, la cual posee diversas funcionalidades que ayudan a gestionar la información en esta línea. En esta aplicación se pueden encontrar errores funcionales y la ausencia de requerimientos que afectan el trabajo del personal de la línea. Por lo que el propósito de la presente investigación consistió en desarrollar un sistema que gestione la información de la línea Weblog del Centro de Ideoinformática, para mejorar su trabajo productivo. Durante el desarrollo de la solución se analizaron los conceptos fundamentales relacionados con el tema como la gestión de la información y las aplicaciones web. Se decidió utilizar en la implementación de la solución el *framework* de desarrollo *Symfony2*, con los lenguajes, tecnologías y herramientas: HTML5, CSS3, *JavaScript*, PHP 5.4, *JQuery 1.9.1*, *Bootstrap 2.3.2*, *NetBeans 7.3*, como gestor de base de datos PostgreSQL 9.1 y como herramienta para administrar la base de datos PgAdmin3, y se utilizó como servidor web Apache 2.2.22. El proceso de desarrollo del software estuvo guiado por la metodología ágil *OpenUp*, apropiada para pequeños proyectos y enfocada en el cliente. El cumplimiento de las funcionalidades propuestas se verificó con varias pruebas realizadas al software. Como resultado se obtuvo un producto funcional con una documentación que sirve de base para futuras investigaciones o modificaciones a la propuesta de solución.

Palabras claves: aplicación web, gestión de la información, requerimientos, *framework* de desarrollo, metodología.

ÍNDICE

INTRODUCCIÓN	1
CAPÍTULO 1. FUNDAMENTOS TEÓRICOS Y ESTADO DEL ARTE DEL SISTEMA PARA LA GESTIÓN DE LA INFORMACIÓN DE LA LÍNEA WEBLOG.....	6
1.1. CONCEPTOS ASOCIADOS AL DOMINIO DEL PROBLEMA	6
1.1.1. <i>Gestión</i>	6
1.1.2. <i>Información</i>	7
1.1.3. <i>Gestión de la Información</i>	7
1.1.4. <i>Aplicación Web</i>	8
1.2. APLICACIONES WEB EXISTENTES QUE GESTIONAN INFORMACIÓN.....	8
1.2.1. <i>Aplicaciones extranjeras</i>	8
PANACEA.....	8
ÜberNote	8
HP Service Anywhere.....	9
1.2.2. <i>Aplicaciones nacionales</i>	9
PK-WEB versión 2.0.....	9
Control Sanitario Internacional (CSI)	9
Info_Blog.....	9
1.2.3. <i>Análisis crítico de los sistemas estudiados</i>	10
1.3. METODOLOGÍA DE DESARROLLO	10
1.3.1. <i>OpenUp</i>	11
1.4. ANÁLISIS DEL SOPORTE TECNOLÓGICO PARA EL DESARROLLO DEL SOFTWARE	12
1.4.1. <i>Lenguajes de programación</i>	12
PHP5.....	12
JAVASCRIPT	13
HTML5.....	13

CCS3	14
1.4.2. <i>Sistema gestor de base de datos y herramienta para su administración</i>	14
PostgreSQL.....	15
PgAdmin3.....	15
1.4.3. <i>Framework de desarrollo</i>	16
Symfony2	16
1.5. HERRAMIENTAS PARA EL DESARROLLO DEL SOFTWARE	16
1.5.1. <i>Sistema Operativo</i>	16
1.5.2. <i>Entorno de Desarrollo Integrado (IDE)</i>	17
Netbeans.....	17
1.5.3. <i>Servidor web</i>	18
Apache	18
1.5.4. <i>Herramienta CASE (Visual Paradigm para UML)</i>	18
1.5.5. <i>Librería JavaScript (jQuery)</i>	18
jQuery.....	19
1.5.6. <i>Librerías para el trabajo con tablas</i>	19
DataTable.....	19
TinyTable.....	20
1.5.7. <i>Framework para crear la interfaz de la aplicación web</i>	21
Bootstrap	21
1.5.8. <i>Librerías JQuery para graficar</i>	22
Flot.....	22
JqPlot.....	22
Chart.js.....	22
1.6. CONCLUSIONES PARCIALES.....	23
CAPÍTULO 2. CARACTERÍSTICAS, ANÁLISIS Y DISEÑO DEL SISTEMA PARA LA GESTIÓN DE LA INFORMACIÓN DE LA LÍNEA WEBLOG	25

2.1.	INTRODUCCIÓN	25
2.2.	DESCRIPCIÓN DE LA APLICACIÓN WEB.....	25
2.2.1.	<i>Modelado del dominio</i>	26
2.2.2.	<i>Descripción de clases del modelo del dominio</i>	26
2.3.	MODELADO DEL SISTEMA	27
2.3.1.	<i>Actores del sistema infoBlog v2</i>	27
2.3.2.	<i>Requisitos funcionales</i>	27
2.3.3.	<i>Requisitos no funcionales</i>	29
2.4.	DIAGRAMA DE CASOS DE USO DEL SISTEMA	30
2.4.1.	<i>Especificación de casos de uso (Descripción de casos de uso)</i>	31
2.5.	DESCRIPCIÓN DEL SISTEMA PROPUESTO	35
2.5.1.	<i>Patrones de diseño implementados</i>	35
2.5.2.	<i>Patrón arquitectónico implementado por Symfony2</i>	38
2.5.3.	<i>Estructura en bundles de la aplicación web</i>	39
2.5.4.	<i>Seguridad en el sistema</i>	40
2.6.	DIAGRAMAS DE CLASES DEL DISEÑO	40
2.7.	DIAGRAMAS DE INTERACCIÓN	41
2.7.1.	<i>Diagramas de Secuencia</i>	42
2.8.	DISEÑO DE LA BASE DE DATOS.....	42
2.9.	DIAGRAMA DE DESPLIEGUE	44
2.10.	CONCLUSIONES PARCIALES	44
 CAPÍTULO 3. IMPLEMENTACIÓN Y PRUEBAS DEL SISTEMA PARA LA GESTIÓN DE LA		
INFORMACIÓN DE LA LÍNEA WEBLOG		45
3.1.	INTRODUCCIÓN	45
3.2.	DIAGRAMA DE COMPONENTES.....	45

3.3. ESTÁNDARES DE CODIFICACIÓN	48
3.4. INTERFACES DE LA APLICACIÓN INFOBLOG v2	52
3.5. PRUEBAS DE SOFTWARE	53
3.5.1. Pruebas unitarias y de integración.....	53
3.5.2. Pruebas de funcionalidad.....	54
3.5.3. Pruebas de seguridad.....	56
3.6. CONCLUSIONES PARCIALES.....	57
CONCLUSIONES GENERALES	58
RECOMENDACIONES	59
REFERENCIAS BIBLIOGRÁFICAS.....	60
BIBLIOGRAFÍA	65
ANEXOS	69
Anexo A	69
Anexo B	73
Anexo C	96
Anexo D.....	114
Anexo E	118

ÍNDICE DE DIAGRAMAS Y FIGURAS

<i>Figura 1: Modelo del dominio.....</i>	26
<i>Figura 2: Diagrama de casos de uso del sistema infoBlog v2.....</i>	31
<i>Figura 3: Patrón arquitectónico MVC</i>	39
<i>Figura 4: DCDEW del CU Gestionar Informe Estadístico.....</i>	41
<i>Figura 5: Diagrama de Secuencia del CU Gestionar Informes Estadísticos</i>	42

<i>Figura 6: Modelo físico de la base de datos del sistema.....</i>	<i>43</i>
<i>Figura 7: Diagrama de despliegue</i>	<i>44</i>
<i>Figura 8: Diagrama de componentes de la aplicación infoBlog v2.....</i>	<i>46</i>
<i>Figura 9: Diagrama de componentes del paquete BackendBundle.....</i>	<i>47</i>
<i>Figura 10: Diagrama de componentes del paquete FrontendBundle</i>	<i>47</i>
<i>Figura 11: Diagrama de componentes del paquete ContenidoBundle</i>	<i>48</i>
<i>Figura 12: Imagen que muestra el nombre de las clases.....</i>	<i>49</i>
<i>Figura 13: Imagen que muestra el nombre de los métodos en una clase</i>	<i>49</i>
<i>Figura 14: Imagen que muestra los namespace y los use en una clase.....</i>	<i>50</i>
<i>Figura 15: Imagen que muestra los nombres de los atributos de las clases.....</i>	<i>50</i>
<i>Figura 16: Imagen que muestra el estilo de los métodos de una clase</i>	<i>51</i>
<i>Figura 17: Imagen que muestra los comentarios en las clases y en los métodos.....</i>	<i>52</i>
<i>Figura 18: Imagen de la portada de la aplicación web infoBlog v2.....</i>	<i>52</i>
<i>Figura 19: Imagen que muestra la interfaz de Mis Publicaciones de la aplicación web infoBlog v2.....</i>	<i>53</i>
<i>Figura 20: No conformidades encontradas en la aplicación web distribuidas en 3 iteraciones.....</i>	<i>56</i>
<i>Figura 21: Porcentaje de las No Conformidades encontradas por clasificación.....</i>	<i>56</i>

ÍNDICE DE TABLAS

<i>Tabla 1: Tabla comparativa de las aplicaciones web analizadas.</i>	<i>10</i>
<i>Tabla 2: Comparación entre las librerías JavaScript estudiadas.</i>	<i>23</i>
<i>Tabla 3: Descripción de clases del modelo de dominio.....</i>	<i>27</i>
<i>Tabla 4: Actores del sistema infoBlog v2</i>	<i>27</i>
<i>Tabla 5: Requisitos Funcionales.....</i>	<i>29</i>
<i>Tabla 6: Requisitos no Funcionales.....</i>	<i>30</i>
<i>Tabla 7: CU Gestionar Informe Estadístico</i>	<i>35</i>
<i>Tabla 8 : Descripción de las variables. Prueba de Funcionalidad para el CU Gestionar Comentario</i>	<i>54</i>
<i>Tabla 9: Prueba de Funcionalidad para el CU Gestionar Comentario</i>	<i>55</i>

INTRODUCCIÓN

Con el paso del tiempo, la información ha sido almacenada según la necesidad y el desarrollo de la sociedad, primero se almacenaba de forma manual, pero actualmente, debido a la gran cantidad de información que se acumula, se hace casi imposible la utilización de métodos tradicionales. El rápido y creciente desarrollo del software y el avance de las Tecnologías de la Información y las Comunicaciones (TIC), han proporcionado herramientas que permiten la realización de este trabajo. La mayoría de las empresas del mundo están haciendo uso de estos avances tecnológicos para informatizar y automatizar sus procesos, además de difundir la información de forma rápida, fácil y a bajo costo. Uno de los procesos que se está informatizando en distintas empresas, es el almacenamiento y administración de información valiosa para ayudar a la toma de decisiones, introduciéndose en este punto la gestión de la información.

Cuba, como país en desarrollo, realiza acciones encaminadas a informatizar los procesos que se llevan a cabo en los diferentes sectores de la sociedad. Para ayudar en el proceso de informatización, que se realiza en el país, se crea la Universidad de las Ciencias Informáticas (UCI). La UCI es una idea creada por Fidel, líder de la Revolución, para conjugar sus esfuerzos con los Jóvenes Club de Computación y Electrónica y las escuelas politécnicas e impulsar el proceso de informatización del país. En pos de ayudar en todo el proceso de informatización que se está llevando a cabo y realizar una mejor gestión de la información, la UCI, cuenta con una infraestructura productiva conformada por diversos centros dedicados al desarrollo de software y otras tareas de interés propio y para el país.

El Departamento de Operaciones Web y Análisis de Información (DOWAI), del Centro de Ideoinformática (CIDI), está dividido en líneas de trabajo, enfocadas a la recopilación, estudio y análisis de información extraída de Internet. Weblog constituye una de estas líneas y su objetivo radica en realizar estudios a través de la información extraída de Internet sobre la blogosfera cubana y mundial, brindar servicios de creación y administración de blogs, y de capacitación para desarrollar estas actividades, además se trabaja en la publicación de comentarios los cuales son creados por los estudiantes asociados al departamento.

En esta línea, la gestión de la información ocupa un papel decisivo para la toma de decisiones y la organización del trabajo con vista a lograr mejores resultados productivos. Actualmente, en la línea Weblog, existe una aplicación web para la gestión de la información llamada Info_Blog. La misma brinda un conjunto de funcionalidades entre las que se encuentran la gestión de blogs, comentarios, publicaciones, informes estadísticos, temáticas, posiciones y usuarios. Igualmente, la aplicación permite exportar a formato PDF

informes estadísticos sobre comentarios, publicaciones, blogs y un informe general. Sin embargo, en la aplicación se pueden encontrar errores funcionales y la ausencia de algunos elementos que se han ido agregando al flujo de trabajo de la línea durante su evolución y perfeccionamiento.

Existen una serie de funcionalidades y requisitos que no están creados ni son tomados en cuenta y que afectan el trabajo del personal de la línea y sus resultados, ya que la ausencia de algunos de estos indicadores atenta contra la calidad del informe que se realiza como resumen de cada período.

La aplicación web no permite la creación de gráficas que reflejen comparaciones entre los datos que se almacenan en los informes estadísticos de los blog, además de que no existe una categoría (semanal, mensual, trimestral y anual) definida para estos informes. Lo anterior afecta las decisiones que puedan emitirse al comparar el comportamiento de los datos estadísticos de los blog en un período de tiempo analizado. El sistema no permite crear gráficas que comparen la cantidad de comentarios registrados con el parámetro “cantidad de comentarios” que se almacena por cada blog; la cantidad de blogs insertados y actualizados; y la cantidad de blogs, publicaciones y comentarios insertados en la aplicación web.

La aplicación cuenta con una funcionalidad de autenticación, la cual está basada en roles, pero no permite asignar responsabilidades a un rol específico, ya que los permisos de estos roles están predefinidos y no pueden ser modificados. El sistema no garantiza insertar varios autores a una misma publicación, utilizando como variante almacenar la publicación a nombre de un solo autor o repetirla por la cantidad de autores que sean, lo que trae como consecuencia la duplicación de la información en la base de datos. Esta solución, en cualquiera de sus variantes, resta precisión a la información que se almacena, ya que al solicitar el informe de publicaciones que se genera en formato PDF, se pueden encontrar contenidos repetidos o información imprecisa.

Los informes que son generados en formato PDF contienen errores de contenido, ya que las funcionalidades que se encargan de realizar el procesamiento de la información contenida en dichos informes, no es precisa, igualmente estos no cuentan con la estructura definida por el centro. Esto trae consigo que para enviar estos informes haya que revisarlos varias veces corrigiendo los errores que presentan, provocando demora en el tiempo de entrega de los mismos.

La aplicación que actualmente se explota en esta línea, está desarrollada con tecnología fuera del marco tecnológico definido por el centro CIDI (MySQL, *phpMyAdmin* y *Codeigniter*) igualmente, carece de documentación que respalde su desarrollo. El sistema tampoco cuenta, en muchas de sus interfaces de

usuario, con filtros para la búsqueda de información y no brinda la posibilidad de evaluar a los estudiantes que trabajan en la línea publicando comentarios. A lo anterior se le suma que posee un bajo nivel de configuración que provoca que para realizar cualquier cambio en el sistema, es necesario hacerlo directamente en el código fuente.

Luego de analizar la situación problemática expuesta, se identifica como **problema de la investigación** ¿Cómo perfeccionar la gestión de la información asociada al trabajo productivo en la línea Weblog?

El **objeto de estudio** de esta investigación está orientado a las aplicaciones web para la gestión de la información y se enmarca en el **campo de acción**: las aplicaciones web para la gestión de la información de blogs.

Para darle solución a la problemática descrita se plantea como **objetivo general**: desarrollar un sistema que gestione la información de la línea Weblog del centro CIDI, para mejorar su trabajo productivo, el cual se desglosa en los siguientes **objetivos específicos**:

- Identificar cómo se realiza el proceso de gestión de la información que se desarrolla dentro de la línea Weblog.
- Desarrollar un estudio sobre la herramienta actualmente en uso, Info_Blog, para identificar todos sus aciertos y deficiencias.
- Definir las tecnologías, herramientas y metodología a utilizar para la implementación del Sistema para la Gestión de la Información de la línea Weblog.
- Implementar las funcionalidades del Sistema para la Gestión de la Información de la línea Weblog.
- Validar el Sistema para la Gestión de la Información de la línea Weblog.

Se define como **idea a defender**: el desarrollo del sistema para la gestión de la información de la línea Weblog del centro CIDI, mejorará el trabajo productivo de la línea.

Como **posible resultado** se espera obtener una aplicación web que gestione la información de la línea Weblog del centro CIDI, para mejorar su trabajo productivo.

Para el desarrollo de esta investigación se utilizaron como **métodos teóricos** el Histórico – Lógico y el Analítico – Sintético, y como **métodos empíricos** la Entrevista y la Observación.

Histórico – Lógico: Se aplica para realizar un seguimiento de la evolución del objeto de estudio y una predicción de lo que puede hacerse en el futuro. En este caso se realizó una investigación de las funcionalidades que posee la aplicación que actualmente está en funcionamiento (Info_Blog), para seleccionar los requerimientos que serían incorporados al nuevo sistema.

Analítico – Sintético: Se utilizó para estudiar la documentación existente sobre la gestión de la información y poder obtener de esta los conocimientos necesarios para el desarrollo del sistema.

Entrevista: Este tipo de método es muy importante para conocer detalladamente el funcionamiento de la nueva aplicación que se quiere elaborar. En este caso se realizó una entrevista a los clientes para conocer cómo funciona el proceso de gestión de la información en la línea Weblog, y posteriormente, definir todos los requerimientos necesarios para que el sistema satisfaga sus necesidades. Ver [Anexo E](#).

Observación: Este método es importante ya que permitió apreciar, a partir de la situación real que se estaba investigando, cómo se desarrolla el proceso que constituye el objeto de estudio. También se utilizó para analizar el sistema que está actualmente en funcionamiento y extraer las funcionalidades que serían incorporadas en la solución que se desarrolló.

El presente trabajo de diploma está estructurado en tres capítulos, además de Introducción, Conclusiones, Recomendaciones, Bibliografía y Anexos. A continuación se presenta la estructura capitular de la investigación:

Capítulo 1: Fundamentos teóricos y estado del arte del Sistema para la Gestión de la Información de la Línea Weblog.

En este capítulo se abordará toda la fundamentación teórica sobre los principales conceptos, la metodología y las herramientas para el desarrollo de la solución, lo cual aportará un mejor entendimiento del problema en cuestión y ayudará en el desarrollo de la solución.

Capítulo 2: Características, análisis y diseño del Sistema para la Gestión de la Información de la Línea Weblog.

En este capítulo se exponen las características que presentará la solución, se definen los requisitos funcionales y no funcionales que debe poseer la solución para satisfacer las necesidades del cliente. Se realiza el diagrama de caso de usos, de interacción, de despliegue, entre otros, para ayudar en la implementación de la solución, además se crean las descripciones de casos de usos del sistema y se

describe la arquitectura de la solución a desarrollar mostrando los patrones de diseño a emplear en la implementación de la misma.

Capítulo 3: Implementación y pruebas del Sistema para la Gestión de la Información de la Línea Weblog.

En este capítulo se describe cómo está implementado el sistema en base a la arquitectura definida en el capítulo anterior. También se muestran las pantallas principales del sistema desarrollado, así como el resultado de las pruebas realizadas en la validación del sistema para comprobar su correcto funcionamiento.

CAPÍTULO 1. FUNDAMENTOS TEÓRICOS Y ESTADO DEL ARTE DEL SISTEMA PARA LA GESTIÓN DE LA INFORMACIÓN DE LA LÍNEA WEBLOG

En el presente capítulo se establecen los fundamentos teóricos de la investigación y los conceptos asociados al dominio del problema a resolver. En el estudio del estado del arte se realiza un análisis sobre las aplicaciones web que gestionan información en el ámbito nacional e internacional para una mayor comprensión e inclusión de alguna funcionalidad en la solución a desarrollar. Se realiza un estudio de las diferentes técnicas y herramientas que se utilizarán para darle solución al problema planteado, sustentado así con su fundamentación. Se abordan los lenguajes de programación que se usarán, y se describen las tecnologías y *framework* necesarios para el desarrollo de la aplicación, así como el sistema gestor de bases de datos que se necesitará para lograr la persistencia de los datos.

1.1. Conceptos asociados al dominio del problema

Existen un conjunto de conceptos relacionados con el problema planteado, a los cuales se les dará una explicación a continuación. Estos en su conjunto forman el marco teórico de la investigación.

1.1.1. Gestión

La gestión es un proceso que comprende determinadas funciones y actividades laborales que los gestores deben llevar a cabo a fin de lograr los objetivos de la empresa. En la gestión los directivos utilizan ciertos principios que les sirven de guía en este proceso. Así pues, la gestión es una asignatura con principios, conceptos y teorías (1).

La gestión se refiere al proceso de "planear, organizar, dirigir, evaluar y controlar" (2) para garantizar una mayor calidad en un producto.

La gestión es dirigir las acciones que constituyan la puesta en marcha concreta de la política general de la empresa, es tomar decisiones orientadas a alcanzar los objetivos marcados, por otra parte la gestión de la producción es un conjunto de responsabilidades y de tareas que deben ser satisfechas para que las operaciones propiamente de la producción sean realizadas respetando las condiciones de calidad, de plazo y de costos que se desprenden de los objetivos y de las estrategias de la empresa (3).

Luego de analizar los conceptos asociados a la gestión, el autor de esta investigación la define como la dirección de las acciones que contribuyan a la toma de decisiones orientadas a alcanzar los objetivos trazados y medir los resultados obtenidos, para orientar finalmente las acciones hacia la mejora del sistema o la empresa.

1.1.2. Información

La información es el significado que otorgan las personas a las cosas. Los datos se perciben mediante los sentidos, estos integran y generan la información necesaria para el conocimiento quien permite tomar decisiones para realizar las acciones cotidianas que aseguran la existencia social (4).

Según Idalberto Chiavenato información es “un conjunto de datos con un significado, o sea, que reduce la incertidumbre o que aumenta el conocimiento de algo. En verdad, la información es un mensaje con significado en un determinado contexto, disponible para uso inmediato y que proporciona orientación a las acciones por el hecho de reducir el margen de incertidumbre con respecto a nuestras decisiones” (5).

Para Ferrell y Hirt, la información "comprende los datos y conocimientos que se usan en la toma de decisiones" (6).

Según Czinkota y Kotabe la información "consiste en datos seleccionados y ordenados con un propósito específico" (7).

Luego de analizar los conceptos asociados a la información, el autor de esta investigación la define como un conjunto de datos organizados con un propósito específico para ayudar en la toma de alguna decisión.

1.1.3. Gestión de la Información

Para Ponjuán cuando se menciona gestión de información se refiere a la gestión que se desarrolla en un Sistema de Información (si se trata de que el sistema tenga como propósito obtener salidas informacionales) y la define como: el proceso mediante el cual se obtienen, despliegan o utilizan recursos básicos (económicos, físicos, humanos, materiales) para manejar información dentro y para la sociedad a la que sirve. Tiene como elemento básico la gestión del ciclo de vida de este recurso y ocurre en cualquier organización. Es propia también de unidades especializadas que manejan este recurso en forma intensiva, llamadas unidades de información (8).

La gestión de la información se vincula con la generación y la aplicación de estrategias, el establecimiento de políticas, así como con el desarrollo de una cultura organizacional y social dirigida al uso racional, efectivo y eficiente de la información en función de los objetivos y metas de las compañías en materia de desempeño y de calidad (9).

Woodman (1985) plantea: “la gestión de información es todo lo que tiene que ver con obtener la información correcta, en la forma adecuada, para la persona indicada, al costo correcto, en el momento oportuno, en el lugar indicado para tomar la acción precisa” (10).

Luego de analizar los conceptos asociados a la gestión de la información, el autor de este trabajo la define

como un conjunto de estrategias y políticas para la obtención de un correcto producto final como un software, un hardware, un informe general, etc.

1.1.4. Aplicación Web

Una aplicación web es aquella que los usuarios usan accediendo a un servidor web a través de Internet o de una intranet (11). Se puede definir como una aplicación en la cual un usuario por medio de un navegador realiza peticiones a una aplicación remota accesible a través de Internet (o a través de una intranet) y que recibe una respuesta que se muestra en el propio navegador (12).

Es básicamente una manera de facilitar el logro de una tarea específica en la web. La aplicación web por lo tanto permite al usuario interactuar directamente con los datos del sistema, todo en forma personalizada, para llevar a cabo esa tarea específica (13).

1.2. Aplicaciones web existentes que gestionan información

El desarrollo de este epígrafe se centrará en el análisis de algunas aplicaciones web existentes en el ámbito nacional e internacional que son utilizadas para la gestión de información, con el objetivo de analizar el manejo de la información en ellas. Entre estos sistemas se encuentra la aplicación web que actualmente está utilizando la línea Weblog. Haciendo énfasis en el estudio de la gestión de información, se presentan a continuación las características de las aplicaciones que se analizaron.

1.2.1. Aplicaciones extranjeras

PANACEA

Es una aplicación web para la gestión de clínicas y hospitales, maneja ágilmente múltiples sedes y grupos empresariales, diferentes tipos de moneda, un alto nivel de integración de terceros a través de la *Web Services* con máxima seguridad para todos los módulos y una administración centralizada. Cuenta con 5 clasificaciones de módulos que poseen un conjunto de funcionalidades, los cuales son: Estratégicos, Asistenciales, Apoyo, Configuración y Administrativos. PANACEA se encuentra en la modalidad de software licenciado. Esta aplicación es comercializada a pequeñas, medianas y grandes empresas. Esta aplicación no posee una licencia de software libre.

ÜBERNOTE

Es una aplicación en línea para archivar y administrar notas. Se trata de un servicio que actualmente es gratuito y en su página oficial se indica que intentarán mantener gran parte de los servicios que brindan de forma libre. El objetivo fundamental de Üernote es permitir crear notas a través de una aplicación web para que luego sean accesibles desde cualquier ordenador. Por lo tanto, esta aplicación enfoca su uso a capturar, editar y gestionar fragmentos de información recogidos, preferentemente, a través del navegador

(14). No es una aplicación con una gran cantidad de funcionalidades y sólo permite la organización de la información a través de las etiquetas, pero puede resultar muy útil para la búsqueda y el archivo de información de Internet. Posee una amplia documentación y presenta una licencia libre.

HP SERVICE ANYWHERE

Es una solución de soporte técnico. Un servicio de atención al cliente basado en Software como Servicio (SaaS por sus siglas en inglés *Software as a Service*) que da soporte a procesos tales como Gestión de incidencias, Gestión de Problemas y Gestión de Cambios. También ofrece mejoras en el diseño de procesos, la creación de tareas, edición de tablas y formas para reducir el coste de la configuración, el mantenimiento continuo y simplificar la Gestión de Servicios de Tecnologías de Información (15).

1.2.2. Aplicaciones nacionales

PK-WEB VERSIÓN 2.0

PK-WEB en la versión 2.0 es una aplicación web del Centro de Isótopos, que permite gestionar y almacenar la información obtenida al realizar un estudio farmacocinético así como el cálculo de parámetros, además de la gestión de archivos referentes al plan de estudio, la implementación de un *Web Service* y el ajuste de curva, entre otras funcionalidades (16). Es una aplicación libre y cuenta con funcionalidades para realizar gráficas y exportar a PDF.

CONTROL SANITARIO INTERNACIONAL (CSI)

El Control Sanitario Internacional propuesto en Cuba permite el control y seguimiento de la importación de productos, enfermedades endémicas de otros países y las transmisibles por vectores. En la Universidad de las Ciencias Informáticas (UCI), se desarrolló el sistema "Control Sanitario Internacional" (CSI) y su objetivo fundamental es automatizar todos los procesos relacionados a la vigilancia, seguimiento de la higiene de los productos, control de foco y la detección de enfermedades (17).

INFO_BLOG

Info_Blog es la aplicación web que actualmente está desarrollada para gestionar la información sobre los datos estadísticos de la blogosfera cubana y mundial. Este sistema pertenece a la línea Weblog del departamento DOWAI, del centro CIDI de la UCI. La aplicación web fue creada para controlar el desarrollo de la actividad productiva de la línea de manera digital.

Esta aplicación no cuenta con algunos requerimientos funcionales atentando esto contra el trabajo del personal de la línea. Entre las funcionalidades que no están implementadas se encuentran: la creación de gráficas comparativas entre los informes estadísticos que se generan, la inserción de varios autores a una misma publicación y la generación de informes estructurados como está definido en el centro. La

aplicación está desarrollada con tecnología fuera del marco tecnológico del centro y no brinda la posibilidad de evaluar a los estudiantes que trabajan dentro de la línea. Posee un bajo nivel de configuración lo que condiciona que para realizar cualquier cambio es necesario hacerlo directamente sobre el código fuente. La aplicación posee una interfaz web fácil de utilizar, donde cualquier usuario que tenga acceso a ella puede navegar sin problema, pero las deficiencias anteriormente mencionadas atentan contra el personal de la línea lo que provoca que esta aplicación no sea una buena opción para utilizarla en la línea.

1.2.3. Análisis crítico de los sistemas estudiados

Los sistemas analizados no son los únicos existentes en el mundo, se realizó una investigación acerca de estos ya que la bibliografía a consultar se encuentra disponible. Sirvieron como muestra de aplicaciones web para el desarrollo del sistema, ya que dentro de sus características se encuentra la gestión de la información. Se hizo un análisis referente a los parámetros que muestra la **Tabla 1**, con el objetivo de estudiar algunas funcionalidades como los gráficos y la forma de exportar a PDF que presentan algunos de estos sistemas.

Una de las aplicaciones estudiadas fue, la que actualmente está en funcionamiento en la línea Weblog, Info_Blog, que se dedica a la gestión de la información sobre la blogosfera cubana y mundial, por lo que el estudio de esta se encaminó en analizar las funcionalidades que se incorporarían en la nueva solución, además de estudiar las deficiencias que presentaba y crear funcionalidades que las suplieran. El análisis de las otras aplicaciones se orientó en el diseño de las interfaces de usuarios, la forma de mostrar la información y la forma de exportar información a PDF.

Aplicaciones Web	Libre	Documentación	Tipo de información que gestiona	Gráficas	Exportar a PDF
PANACEA	No	Media	Gestión de clínicas y hospitales	Si	Si
ÜberNote	Sí.	Amplia	Almacenar Notas	No	No
HP Service Anywhere	No	Media	Incidencias, Problemas y Cambios del conjunto de mejores prácticas ITIL	No	-
PK-WEB en la versión 2.0	Si	Media	Información obtenida al realizar un estudio farmacocinético	Si	Si
Control Sanitario Internacional (CSI).	Si	Media	Permite el control y seguimiento de la importación de productos, enfermedades endémicas de otros países y las transmisibles por vectores.	No	Si
Info_Blog	Si	No existe	Blogs	No	Si

Tabla 1: Tabla comparativa de las aplicaciones web analizadas

1.3. Metodología de desarrollo

Desarrollar un buen software depende de un sinnúmero de actividades y etapas, donde el impacto de elegir la mejor metodología para un equipo, en un determinado proyecto es trascendental para el éxito del producto. El papel preponderante de las metodologías es sin duda esencial en un proyecto y en el paso inicial, que debe ajustar en el equipo, guiar y organizar actividades. Todo lo anterior conlleva al cumplimiento de las metas trazadas en el grupo. Las metodologías ágiles se han convertido en herramientas muy utilizadas por las empresas desarrolladoras de software debido al bajo costo en la producción de software y a la reducción en el tiempo de entrega de los productos.

1.3.1. *OpenUp*

Esta metodología es un proceso ágil y unificado, que contiene el conjunto mínimo de prácticas que ayudan a los equipos a ser más fuertes en el desarrollo de software. Se selecciona esta metodología para ser usada en el desarrollo del sistema ya que está definida como la metodología de trabajo en el centro CIDI, además de que establece una filosofía ágil que se centra en la naturaleza colaborativa del desarrollo de software. A continuación se exponen algunos de los beneficios que trae consigo el empleo de esta metodología.

Beneficios de esta metodología:

- Es apropiada para proyectos pequeños y de bajos recursos, permite disminuir las probabilidades de fracaso en los proyectos pequeños e incrementar las probabilidades de éxito.
- Permite detectar errores en etapas tempranas del desarrollo de un producto de software a través de un ciclo iterativo.
- Genera la documentación necesaria para el desarrollo del proyecto, lo que hace que se entreguen en el tiempo estimado.
- Por ser una metodología ágil tiene un enfoque centrado al cliente y con iteraciones cortas.

Esta metodología está compuesta por cuatro fases las cuales se exponen a continuación:

- **Concepción:** se enfoca en el entendimiento del propósito y objetivos del proyecto que se desarrollará, obteniendo suficiente información para confirmar que se debe hacer. En esta fase se debe entender qué construir, identificar los actores, quién está interesado en este sistema, etc.
- **Elaboración:** en esta fase se tratan los riesgos significativos para la arquitectura. El propósito de esta fase es establecer la base de elaboración de la arquitectura del sistema. Además obtener una comprensión más detallada de los requerimientos que debe cumplir el sistema.
- **Construcción:** fase enfocada al diseño, implementación y prueba de las funcionalidades para desarrollar un sistema completo. El propósito de esta fase es completar el desarrollo del sistema basado en la arquitectura definida.

- **Transición:** es la última fase, cuyo propósito es asegurar que el sistema es entregado a los usuarios, y evalúa la funcionalidad del último entregable de la fase de construcción.

OpenUp es un proceso mínimo, completo y extensible. Promueve técnicas ágiles y principios, mientras que tiene un ciclo de vida estructurado que hace hincapié en la continua entrega de software, que es valioso para los interesados en el desarrollo rápido de aplicaciones de calidad (18).

1.4. Análisis del soporte tecnológico para el desarrollo del software

La correcta selección de las tecnologías a emplear en el desarrollo de un software es uno de los pasos importantes debido a que influye directamente en la calidad del producto final que se entrega al cliente y en el esfuerzo del equipo de desarrollo para obtenerlo. Una de las características que debe presentar la nueva solución es que se desarrolle con tecnologías definidas en el marco de trabajo del centro CIDI. La aplicación que se está utilizando actualmente en la línea Weblog, está desarrollada con tecnologías fuera de este marco de trabajo. La selección de las tecnologías que se exponen a continuación para darle solución a la problemática planteada está sustentada por el grado de conocimiento que posee el cliente respecto a la informática, siendo este de vital importancia en la definición de las mismas.

1.4.1. Lenguajes de programación

Existen muchos lenguajes de programación para el desarrollo de aplicaciones web tales como: PHP, Java, ASP, entre otros, los cuales permiten interactuar con los usuarios y utilizar sistemas de base de datos para el tratamiento de datos.

PHP5

Se selecciona el lenguaje PHP que es “un lenguaje de programación del lado del servidor gratuito e independiente de plataforma, rápido, con una gran librería¹ de funciones y mucha documentación” (19), es de código abierto y utilizado para la generación de páginas dinámicas.

Ventajas (20):

- **Velocidad:** está escrito en C, por lo que se ejecuta rápidamente utilizando poca memoria.
- **Simplicidad:** los usuarios con experiencia en C y C++ podrán utilizar PHP rápidamente debido a que sus sintaxis es similares.
- **Conectividad:** dispone de una amplia gama de librerías, y agregarle extensiones es muy fácil. Esto le permite a PHP ser utilizado en muchas áreas diferentes, tales como encriptado, gráficos, XML y otras.

¹ Es un conjunto de subprogramas utilizados para desarrollar software, contienen código y datos, que proporcionan servicios a programas independientes.

- Puede interactuar con muchos motores de bases de datos tales como MySQL, MS SQL, Oracle, Informix, PostgreSQL y otros.

La selección de este lenguaje también viene unido al *framework* de desarrollo seleccionado. En la solución, este lenguaje se utilizará para el tratamiento de los datos del lado del servidor, es decir, para el trabajo con la información del lado del servidor.

JAVASCRIPT

JavaScript es un lenguaje de programación utilizado para crear pequeños programas encargados de realizar acciones dentro del ámbito de una página web.

Entre las acciones típicas que se pueden realizar en *JavaScript* se tienen dos vertientes. Por un lado los efectos especiales sobre páginas web, para crear contenidos dinámicos y elementos de la página que tengan movimiento, cambien de color o cualquier otro dinamismo. Por el otro permite ejecutar instrucciones como respuesta a las acciones del usuario, con lo que se pueden crear páginas interactivas con programas como calculadoras, agendas o tablas de cálculo.

JavaScript “se trata de un lenguaje de programación del lado del cliente, porque es el navegador el que soporta la carga de procesamiento. Gracias a su compatibilidad con la mayoría de los navegadores modernos, es el lenguaje de programación del lado del cliente más utilizado” (21).

Se decidió utilizar este lenguaje de programación porque es sencillo trabajar con él. Incluso las personas que no tengan una experiencia previa en la programación podrían aprender este lenguaje con facilidad con sólo un poco de práctica, garantizando optimizar el tiempo de desarrollo de las aplicaciones web. Además la selección de este lenguaje viene sustentada por la amplia documentación existente, la experiencia que poseen los desarrolladores con el mismo y porque el *framework* de diseño a utilizar en el desarrollo de la solución utiliza este lenguaje.

HTML5

El Lenguaje de Marcas de Hipertexto (HTML por sus siglas en inglés *HyperText Markup Language*), en la versión 5, es la quinta revisión del lenguaje básico de la *World Wide Web* (WWW por sus siglas en inglés). Este lenguaje de programación es usado por múltiples desarrolladores web por sus avances, mejoras y ventajas con respecto a las versiones anteriores. El principal criterio de diseño de HTML5 ha sido el de resolver problemas prácticos, y con este objetivo adopta soluciones dirigidas a facilitar el trabajo en situaciones reales.

HTML5 incluye varias novedades en diversos ámbitos (22):

- **Estructura del cuerpo:** la mayoría de las webs tienen un formato común, formado por elementos como cabecera, pie, navegadores, etc. HTML5 permite agrupar todas estas partes de una web en nuevas etiquetas que representarán cada una de las partes típicas de una página.
- **Etiquetas para contenido específico:** hasta ahora se utilizaba una única etiqueta para incorporar diversos tipos de contenido enriquecido, como animaciones *Flash* o vídeo. Ahora se utilizarán etiquetas específicas para cada tipo de contenido en particular, como audio, vídeo, etc.
- **Canvas:** es un nuevo componente que permitirá dibujar, por medio de las funciones de una Interfaz de Programación de Aplicaciones (API por sus siglas en inglés *Application Programming Interface*), en la página todo tipo de formas, que podrán estar animadas y responder a interacción del usuario.
- **Fin de las etiquetas de presentación:** todas las etiquetas que tienen que ver con la presentación del documento, es decir, que modifican estilos de la página, serán eliminadas. La responsabilidad de definir el aspecto de una web correrá a cargo únicamente de las Hojas de Estilo en Cascada (CSS por sus siglas en inglés *Cascading Style Sheet*).

CCS3

CCS3 es la nueva versión del CSS, es simplemente el lenguaje con el cual se puede dar diseño y apariencia a las páginas HTML. Es desarrollado y distribuido por la *World Wide Web Consortium* (W3C por sus siglas en inglés) con el fin especificado anteriormente. A continuación se muestran algunas características que facilitan su uso.

Características (23):

- CSS3 propone una navegación más rápida y menores tiempos de respuesta producidos por una reducción de imágenes, las cuales ya no serán requeridas para diseñar botones o efectos de texto.
- Deja atrás una excesiva dependencia de *JavaScript* para fines de representación visual, como lo son las animaciones, dando como resultado menos código y mejor rendimiento.
- Representa una futura menor dependencia de software para gráficos que resultan bastante caros, como es el caso de *Photoshop Illustrator* o *Corel*.

1.4.2. Sistema gestor de base de datos y herramienta para su administración

Un sistema gestor de base de datos (SGBD) es un conjunto de programas que administran y gestionan la información contenida en una base de datos ayudando a mantener la integridad, control de seguridad, privacidad y manipulación de los datos.

POSTGRESQL

PostgreSQL es un SGBD objeto-relacional, distribuido bajo licencia de Distribución de Software *Berkeley* (BSD por sus siglas en inglés *Berkeley Software Distribution*), que es una licencia de software libre y con su código fuente disponible libremente. Esto quiere decir que el código fuente del programa está disponible a cualquier persona, permitiéndole modificar el sistema para ajustarlo a sus necesidades. Algunas de las características que presentan los SGBD son las siguientes: “copias de seguridad en caliente, múltiples métodos de autenticación, completa documentación, disponible para Linux y *Windows* 32/64bit, acceso encriptado vía Capa de Conexión Segura (SSL por sus siglas en inglés *Secure Sockets Layer*) y completa documentación” (24).

PostgreSQL se caracteriza por ser un sistema estable, de alto rendimiento, gran flexibilidad ya que funciona en la mayoría de los sistemas Unix, además tiene características que permiten extender fácilmente el sistema. Puede ser integrada al ambiente *Windows* permitiendo de esta manera a los desarrolladores, generar nuevas aplicaciones o mantener las ya existentes (25).

Por lo expuesto, PostgreSQL se convierte en una gran alternativa al momento de decidirse por un SGBD a la hora de desarrollar aplicaciones de software que tengan que gestionar los datos en una base de datos. Por estas características el cliente decidió que se empleará PostgreSQL en la versión 9.1, que es de Apoyo a Largo Plazo (LTS por sus siglas en inglés *Long Term Support*), como SGBD.

PGADMIN3

PgAdmin3 es una aplicación de diseño y manejo de bases de datos para su uso con el gestor de bases de datos PostgreSQL. Es un software diseñado para responder a las necesidades de todos los usuarios, desde la escritura de simples consultas SQL hasta la elaboración de bases de datos complejas. A continuación se muestran algunas características de esta herramienta que hacen que sea una de las más usadas y completa.

Características (26):

- Soporte para todos los tipos de objetos de PostgreSQL.
- Incluye una interfaz gráfica de administración, una herramienta para el trabajo con SQL y un editor de código de procedimientos y funciones.
- La interfaz gráfica soporta todas las características presentes de PostgreSQL y se puede hacer la administración fácilmente.
- Está disponible en más de 30 lenguajes y para varios sistemas operativos.

Las características anteriores, la amplia documentación, la fácil interacción con la aplicación y la

experiencia que se tiene con el uso de la misma sustentan la selección de la herramienta PgAdmin3 en la versión 1.14 para gestionar la base de datos de la solución.

1.4.3. **Framework de desarrollo**

Un *framework* o marco de trabajo, en el desarrollo de software, es “una estructura conceptual y tecnológica de soporte definida, normalmente con artefactos o módulos de software concretos, en base a la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros programas para ayudar a desarrollar y unir los diferentes componentes de un proyecto” (27).

SYMFONY2

El uso de un *framework* para desarrollar un sistema facilita y agiliza todo el proceso de desarrollo. En el desarrollo del Sistema para la Gestión de la Información de la línea Weblog (infoBlog v2) el cliente decidió que se utilizará como marco de trabajo *Symfony2* en la versión 2.3.6 que es LTS, un *framework* para PHP con una amplia aceptación entre los usuarios y las empresas que se dedican a la creación de aplicaciones web. La elección de *Symfony2* para desarrollar la solución de esta investigación, facilitará el trabajo con algunos componentes para realizar gráficas comparativas que es uno de los requerimientos del sistema. Además permitirá mantener cierta homogeneidad con los demás componentes del sistema. A continuación se relacionan algunas características de *Symfony2* que fortifican su elección.

Características (28):

- Fácil de instalar y configurar en la mayoría de las plataformas (y con la garantía de que funciona correctamente en los sistemas *Windows* y *Unix* estándares).
- Independiente del sistema gestor de bases de datos.
- Sencillo de usar en la mayoría de los casos, pero lo suficientemente flexible como para adaptarse a los casos más complejos.
- Sigue la mayoría de las mejores prácticas y patrones de diseño para la web.
- Preparado para aplicaciones empresariales, además es lo suficientemente estable como para desarrollar aplicaciones a largo plazo.
- Fácil de extender, lo que permite su integración con librerías desarrolladas por terceros.

1.5. **Herramientas para el desarrollo del software**

1.5.1. **Sistema Operativo**

La plataforma seleccionada por los desarrolladores es Linux, específicamente el sistema operativo Ubuntu en su versión 12.04 el cual es LTS. Ubuntu es “un sistema operativo completo basado en GNU/Linux,

disponible de forma libre con soporte para la comunidad y los profesionales, que permite realizar todas las tareas diarias que actualmente llevas a cabo con tu maquina sin necesidad de reaprender todo lo que ya sabes” (29). Los sistemas UNIX ofrecen un conjunto de prestaciones en materia de seguridad, operatividad, disponibilidad y flexibilidad que dotan a los desarrolladores de un entorno de trabajo amigable y confiable lo cual facilita las condiciones de trabajo para el desarrollo del software. Los desarrolladores de aplicaciones web han optado por la selección de desarrollar sus aplicaciones sobre Linux y a continuación se muestran un conjunto de características que justifican la selección de muchas empresas y también la del desarrollador de la aplicación infoBlog v2.

Características (30):

- **Linux es gratis:** por lo que muchas empresas del mercado lo han visto como una excelente opción para bajar los costos de licenciamiento.
- **Linux es un sistema de código abierto:** puede crear su propia distribución del sistema operativo e incluir todo el software libre que crea necesario.
- **Servidores web basados en Linux:** la mayoría de servidores web que sirven páginas de Internet son Linux.
- **La seguridad y estabilidad de Linux:** le han convertido en una opción de primera mano para empresas que ofrecen servicios, como el hospedaje de sitios web.

1.5.2. *Entorno de Desarrollo Integrado (IDE)*

Un Entorno de Desarrollo Integrado (IDE por sus siglas en inglés *Integrated Development Environment*), es una herramienta informática que facilita el trabajo a los programadores, brindándoles un conjunto de herramientas de programación. Un concepto más amplio es el que se presenta a continuación:

“Es un entorno de programación que ha sido empaquetado como un programa de aplicación, o sea, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica. Los IDEs pueden ser aplicaciones por sí solas o pueden ser parte de aplicaciones existentes” (31).

NETBEANS

NetBeans es un IDE gratuito y de código abierto, utilizado en el desarrollo de aplicaciones web, de escritorio y para móviles. Se encuentra disponible para los sistemas operativos *Windows*, *Linux*, *Macintosh Operating System (MAC OS)* y *Solaris*. Para el desarrollo de la solución se utilizará la versión 7.3 de *NetBeans* como IDE, la base en la que se sustenta su elección es que permite desarrollar aplicaciones utilizando el *framework Symfony2* y ejecutar los comandos de este directamente desde la

interfaz del IDE. Por otra parte el desarrollador posee gran dominio y experiencia en el uso de esta herramienta de desarrollo.

1.5.3. **Servidor web**

Un servidor web es “un programa que atiende y responde demandas de información de los navegantes en protocolo HTTP. Repite una y otra vez la misma cadena de acciones que lo definen funcionalmente: espera peticiones, las recibe y busca el recurso necesario para satisfacerlas, envía el recurso en la misma conexión de abordaje de la petición y vuelve a esperar la siguiente demanda o petición” (32).

APACHE

Para el desarrollo de la solución se decide utilizar Apache como servidor web. “Apache es un servidor web HTTP de código abierto, para plataformas *Unix*, *Windows*, *MAC OS* y otras, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual, presenta entre otras características altamente configurables, bases de datos de autenticación y negociado de contenido, pero no posee una interfaz gráfica en su configuración” (33). El cliente decidió que se utilizaría como servidor web Apache en la versión 2.2.22 puesto que es un servidor de arquitectura modular el cual permite la instalación de diferentes módulos según la necesidad del usuario, lo cual facilita un mayor aprovechamiento de las prestaciones que ofrece el servidor y por la experiencia en el trabajo con el mismo tanto de él como del desarrollador.

1.5.4. **Herramienta CASE (Visual Paradigm para UML)**

Para el desarrollo de la solución se decide utilizar como herramienta de diseño *Visual Paradigm for UML 8.0*. “Es una herramienta de diseño multiplataforma diseñada para asistir el desarrollo de software. Soporta los principales estándares de la industria, tales como; UML, SysML, BPMN, XMI” (34). Se decide emplear esta herramienta para ayudar con el desarrollo de la solución por la experiencia que se posee con su utilización y ofrece un conjunto completo de herramientas que los equipos de desarrollo necesitan para la captura de requisitos, planificación de software, planificación de pruebas, modelación de clases, datos, prototipos y otras actividades.

1.5.5. **Librería JavaScript (jQuery)**

Los navegadores llevan integrado sus propios motores *JavaScript* con el objetivo de aprovechar al máximo sus características y ofrecer al usuario un mayor rendimiento en aplicaciones web. A continuación se exponen un conjunto de características que sustentan la selección de esta librería para el desarrollo de la solución.

JQUERY

JQuery es una librería *JavaScript* rápida, pequeña y rica en funciones que permite la manipulación, manejo de eventos y animación mucho más simple que funciona a través de una multitud de navegadores. Es la librería más utilizada por su fácil manejo en el tratamiento de los objetos del Modelo de Objetos del Documento (DOM por sus siglas en inglés *Document Object Modelo*). Es un software libre y de código abierto permitiendo su uso en proyectos libres y privativos. Además es compatible con los navegadores *Mozilla Firefox 2.0+*, *Internet Explorer 6+*, *Safari 3+*, *Opera 10.6+* y *Google Chrome 8+*. Muy aconsejable para la integración de efectos y animaciones personalizadas (35).

Para el desarrollo de la solución se decidió utilizar la librería de *JavaScript JQuery en su versión 1.9.1* puesto que el código se puede trabajar separado del HTML, haciendo más fácil de mantener y enriquecer, acelerando el desarrollo del proyecto. También se puede considerar que su sintaxis es liviana, teniendo en cuenta que la sencillez y poca extensión de código es fundamental para el desarrollo. A esto sumarle que existen cientos de *plugins*² adaptables como el que se utilizará en el desarrollo de la solución para mostrar las tablas. El uso de esta librería hará que el proyecto sea fácil de desarrollar. También se ha seleccionado puesto que es fácil de aprender, ahorrando tiempo en el desarrollo de la solución, además de la compatibilidad que posee con el *framework* de desarrollo a emplear. También que el amplio uso de esta librería hace que exista una gran comunidad de desarrolladores, con lo cual, el soporte, la documentación y los recursos se presentan de forma amplia.

1.5.6. Librerías para el trabajo con tablas

DATATABLE

DataTable es un *plugin* para la librería *JQuery* para mejorar tablas, añadiendo ordenación, la paginación y la capacidad de filtrado a las tablas HTML, con un mínimo esfuerzo. El objetivo declarado de *DataTable* es mejorar la accesibilidad de los datos en tablas HTML. Se encuentra bajo la licencia del Instituto Tecnológico de Massachusetts (MIT por sus siglas en inglés *Massachusetts Institute of Technology*). Usted es libre de usar, modificar y distribuir este software, siempre y cuando la cabecera de los derechos de autor se deja intacto (en concreto el bloque de comentarios que se inicia con */*!*) (36).

² Es un módulo aparte que se incluye opcionalmente en una aplicación. Es un complemento de una aplicación que se relaciona con otra para aportarle una función nueva y generalmente muy específica.

Es una herramienta muy flexible, en base a los fundamentos de la mejora progresiva, que se sumarán los controles avanzados de interacción a cualquier tabla HTML. Posee un conjunto de características las cuales se exponen a continuación.

Características (37):

- Paginación de longitud variable.
- Multi - columna de ordenación con la detección de tipo de datos.
- Mostrar datos de casi cualquier fuente de datos: arreglos Javascript, Ajax y el archivo en el servidor de procesamiento (PHP, C#, Perl, Ruby, AIR, Gears, etc).
- Es gratis.
- Creación dinámica de tablas.
- Filtrado de una sola columna.
- Tipos de paginación alternativos.

TINYTABLE

TinyTable es un script clasificador de tablas, es fácil de usar y con multitud de funciones en sólo 2,5 KB. Presenta grandes características como que incluyen resaltado de la columna, la paginación opcional, soporte para enlaces, análisis sintáctico, fecha/link, resaltado alterna fila, el reconocimiento automático de tipos de datos y la columna de clasificación selectiva. Además, el estilo de la tabla es completamente personalizable a través de CSS (38).

Algunas de las características que presenta este script es que permite la clasificación de las columnas de las tablas, la búsqueda en tiempo real de datos, filtrado, los promedios y totales de las columnas, la numeración de registros y el reconocimiento automático de tipos de datos. Es ligero, rápido y el único requisito para su correcto funcionamiento es tener *JavaScript* activado del lado del cliente. Esta desarrollado por Michael Leigeber bajo la licencia *OtherFree / OpenSourceLicense–CreativeCommons*. Trabaja en las plataformas *Windows*, *Linux*, *MAC OS* y *Solaris*.

Luego del análisis de estas dos herramientas para el manejo de los datos en tablas, se decide seleccionar para el desarrollo de la solución *DataTable*. La selección de esta herramienta se sustenta principalmente porque posee el diseño de tablas que el cliente definió en los requisitos, además de brindar un grupo de características que facilitará la visualización de los contenidos de la solución. Ejemplo de estas características es que facilitará los filtros dinámicos para la búsqueda de información en las tablas, igualmente permite ordenar por una o varias columnas. Estas características y las analizadas

anteriormente permitirán que la solución que se desea desarrollar brinde una interfaz de visualización de los datos acorde con lo especificado por el cliente.

1.5.7. *Framework para crear la interfaz de la aplicación web*

BOOTSTRAP

Es el *framework* de *Twitter* que permite crear interfaces web con CSS y *JavaScript*, que adaptan la interfaz dependiendo del tamaño del dispositivo en el que se visualice de forma nativa, es decir, automáticamente se adapta al tamaño de un ordenador o de una Tablet sin que el usuario tenga que hacer nada, esto se denomina diseño adaptativo o *Responsive Design*. Los diseños creados con *Bootstrap* son simples, limpios e intuitivos, esto les da agilidad a la hora de cargar y al adaptarse a otros dispositivos. El *framework* trae varios elementos con estilos predefinidos y fáciles de configurar: Botones, Menús desplegables, formularios incluyendo todos sus elementos e integración *JQuery* para ofrecer ventanas y *tooltips* dinámicos (39).

Bootstrap es un *framework* que simplifica el proceso de creación de diseños web combinando CSS y *JavaScript*. Ofrece una serie de plantillas CSS y ficheros *JavaScript* que permiten integrar el *framework* de forma sencilla y potente en los proyectos webs. Algunas características se presentan a continuación.

Características principales de *Bootstrap* (40):

- Permite crear interfaces que se adapten a los diferentes navegadores, tanto de escritorio como *tablets* y móviles a distintas escalas y resoluciones.
- Se integra perfectamente con las principales librerías *JavaScript*, por ejemplo *JQuery*.
- Ofrece un diseño sólido usando estándares como CSS3/HTML5.
- Es un *framework* ligero que se integra de forma limpia en el proyecto que se trabaja.
- Funciona con todos los navegadores, incluido *Internet Explorer*.

El cliente decidió utilizar como *framework* de diseño para el desarrollo de la solución, *Bootstrap* en la versión 2.3.2. La selección se sustenta en que este es un *framework* de fácil uso y no hay que hacer muchas modificaciones en los archivos CSS para lograr una buena apariencia de las plantillas que se van a crear, siendo de esta forma la elección de una plantilla, por parte del cliente, para el diseño de la solución que solo se le modificarán algunas partes. La selección es sustentada además en que *Bootstrap* se adapta bien en todos los navegadores y en los dispositivos móviles aunque estos últimos no son objetivo de esta investigación. También brinda las herramientas necesarias para que se extienda y se adapte a las necesidades que desea el cliente y es compatible con el *framework* de desarrollo que se seleccionó.

1.5.8. Librerías JQuery para graficar

FLOT

Flot es una librería muy difundida y que solo utiliza *JavaScript* para crear gráficas muy atractivas. Dentro de sus principales características se puede destacar su aspecto atractivo y su interacción teniendo la capacidad de realizar zoom en áreas determinadas o bien desplazamientos por medio del mouse. Es compatible con la mayoría de los navegadores: *Internet Explorer 6/7/8*, *Firefox 2.x+*, *Safari 3.0+*, *Opera 9.5+* y *Konqueror 4.x* (41). Es una librería de trazado de *JavaScript*, es pequeña, el rendimiento es bueno y es compatible con todo tipo de gráficos. Hay muchos tipos de gráficos disponibles, como gráfico de líneas, circular, de barras, de áreas, apilado; también soporta de actualización en tiempo real y de actualización Ajax. Si el conocimiento es poco acerca de *JavaScript* y *JQuery*, se podría empezar con *Flot* fácilmente (42).

JQPLOT

JqPlot es un trazado de gráficos y *plugin* para el *framework JQuery JavaScript*. *JqPlot* produce hermosas líneas, barras y gráficos circulares con muchas características como: numerosas opciones de estilo para el gráfico que dibuja, presenta hasta 9 ejes Y, el texto del eje girado, cálculo automático de la línea de tendencia, la información sobre herramientas y punto de datos resaltado. *JqPlot* es concebido, desarrollado y mantenido por Chris Leonello. Existe mucha documentación y fuentes para obtener ayuda sobre esta librería. Es un proyecto de código abierto dual bajo la Licencia Pública General (GPL por sus siglas en inglés *General Public License*) y MIT (43).

Algunas de las características de *JqPlot* (44):

- Realiza gráficos de líneas, de barra y circulares.
- Personalizar los colores, las sombras, los marcadores, las garrapatas y más opciones.
- Fecha de ejes con un formato personalizable.
- Parámetros por defecto para facilitar su uso.
- Es libre y de código abierto.

CHART.JS

Es una librería *JavaScript* que permite generar distintos tipos de gráficas.

Las librerías *Chart.js* proporcionan múltiples funciones (45):

- Se pueden combinar diferentes tipos de series y mostrarlas a través de diferentes paneles en la misma pantalla.
- Ejes secundarios múltiples de diferentes tipos son totalmente compatibles.

- El aspecto de los gráficos se puede cambiar por el cambio a un tema diferente. Usted es libre de crear temas personalizados según las necesidades.
- Aplicar la paleta de color que es el más apropiado para su serie de cartas. Elija entre una gama de colores predefinidos o definir una paleta personalizada.
- Características de interactividad de usuario incluyen selección de series y puntos, zoom y desplazamiento.

Para la selección de la librería a utilizar se tuvo en cuenta la siguiente tabla comparativa:

Parámetros	jqPlot	Chart.js	Flot
Año	2009	2013	2007
Navegadores compatibles	Internet Explorer (IE) 7, IE 8, Firefox, Safari, y Opera	Todas web y móviles de los navegadores modernos, incluyendo IE6 +	Internet Explorer 6 +, Chrome, Firefox2 +, Safari 3 + y Opera 9.5 +
Código fuente disponible para su descarga y modificación	Si	Si	Si
Licencias	GPL y MIT	Libre bajo licencia MIT	Libre
Animación	Si	Si	Código adicional requerido
Leyenda	Si	Si	Si
Zoom	Si	No	Si
Integración jQuery Nativo	Si	No	Si
Gráficos de columnas	Sí. En 2D	Sí. En 2D	Sí. En 2D
Gráficos de barras	Sí. En 2D	Sí. En 2D	Sí. En 2D
Gráficos de pastel y de anillos	Sí. En 2D	Sí. En 2D	Sí. En 2D
Gráficos de líneas	Sí. En 2D	Sí. En 2D	Sí. En 2D
Gráficos de área	Si	Si	Si
Gráficos apilados	Si	No	Si
Múltiples ejes	Si	No	Si

Tabla 2: Comparación entre las librerías JavaScript estudiadas

Después del estudio de estas librerías *jQuery* desarrolladas para hacer gráficos, se decide seleccionar para el desarrollo de la aplicación web la librería *JqPlot*, ya que brinda muchísimas variantes de los gráficos con diferentes funcionalidades y los estilos son muy fáciles para modificar. También permite la interacción con el usuario ya implementada la gráfica, proporcionando el manejo de leyendas, colores y tipos de gráficos. Es fácil de utilizar y brindará las funcionalidades que ayudarán en el desarrollo de los requisitos funcionales especificados por el cliente.

1.6. Conclusiones parciales

Luego de la investigación realizada se concluye que:

- Con el estudio de los sistemas existentes se tomaron ideas de funcionalidad y de interfaz gráfica para incorporarlas a la nueva solución.

- La selección de la herramienta para el modelado del sistema ayudó en el flujo de la implementación de la solución.
- La selección de lenguajes, tecnologías, herramientas y metodología de desarrollo permitieron desarrollar la aplicación web en el tiempo estimado.

CAPÍTULO 2. CARACTERÍSTICAS, ANÁLISIS Y DISEÑO DEL SISTEMA PARA LA GESTIÓN DE LA INFORMACIÓN DE LA LÍNEA WEBLOG

2.1. Introducción

En este capítulo se detallará el funcionamiento del sistema propuesto para dar solución a la problemática planteada. También se describirán los requerimientos que el sistema debe cumplir para satisfacer al cliente. Se estudiarán los requisitos identificados para estructurarlos y refinarlos, con el objetivo de entender el funcionamiento que debe cumplir cada uno para que el sistema funcione según las necesidades del cliente. Se describirá la arquitectura a emplear y se presentarán algunos diagramas como: el modelo del dominio, el de casos de uso, de clases del diseño, entre otros.

2.2. Descripción de la aplicación web

La aplicación web infoBlog v2 permitirá gestionar la información que se maneja en la línea Weblog, del departamento DOWAI, como: datos e informes estadísticos sobre los blog, las publicaciones, los comentarios, entre otros. Además de permitir exportar a formato PDF informes de tipo comentario, donde se mostrará un archivo con la cantidad de comentarios por blog donde son realizados. También exportará informes de blog, donde se mostrará un archivo en formato PDF que contendrá la cantidad de blogs insertados y actualizados en la aplicación. Igualmente exportará informes de publicaciones y un informe general. Brindará la posibilidad de generar una gráfica de comentarios donde se medirá la cantidad de los registrados en el sistema con la cantidad de los registrados en el parámetro “cantidad de comentarios” de los informes estadísticos semanales. En esta vista se mostrarán dos tablas, una con la cantidad de comentarios por blog y otra con la cantidad por usuarios.

Se generará una gráfica comparando la cantidad de blogs insertados y actualizados en el sistema y al mismo tiempo se mostrará una tabla con la cantidad de blogs insertados y actualizados por los usuarios. Se mostrarán varias gráficas comparando los parámetros que se registran (cantidad de comentarios, cantidad de visitas, cantidad de visitantes, etc.) en los informes estadísticos por cada blog. Por otra parte se visualizará una gráfica que compare la cantidad de comentarios, publicaciones y blogs registrados en el sistema. Esta aplicación brindará la posibilidad de gestionar usuarios a través del dominio UCI.CU, donde existirá una tabla en la base de datos que contendrá algunos datos de los usuarios (nombre, apellidos y usuario) y el estado de la cuenta que puede ser Activo o Inactivo. Se gestionarán los roles y se le asignarán

permisos a través del código, actualmente existen tres roles definidos (Comentarista, Editor y Administrador); cuando se inserte un nuevo usuario en el sistema, se le asignará un rol como mínimo. También permitirá listar los contenidos de tipo comentario y publicación a los usuarios que los crearon.

2.2.1. Modelado del dominio

Según Craig Larman en la 2da edición del libro UML y Patrones plantea que “Un modelo del dominio es una representación de las clases conceptuales del mundo real, no de componentes software. No se trata de un conjunto de diagramas que describen clases software, u objetos software con responsabilidades” (46). También se les denomina modelos conceptuales (término utilizado en la primera edición del libro de Larman), modelo de objetos del dominio y modelos de objetos de análisis.

En la **Figura 1** se muestra el diagrama de modelo de dominio que representa las relaciones que existen entre los principales conceptos.

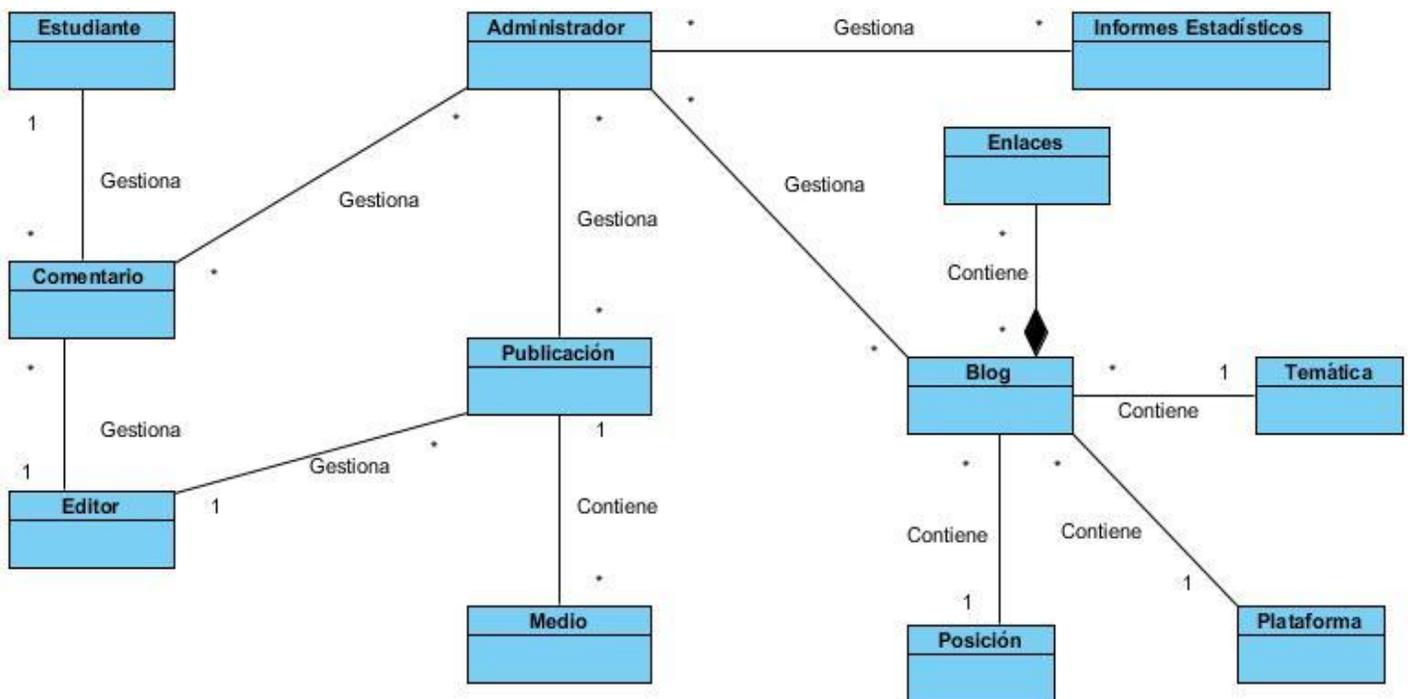


Figura 1: Modelo del dominio

2.2.2. Descripción de clases del modelo del dominio

La siguiente tabla muestra la descripción de los principales conceptos mostrados en el modelo del dominio.

Concepto	Descripción
Administrador	Persona encargada de gestionar todas las funcionalidades del sistema.
Estudiante	Persona encargada de gestionar Comentarios.
Editor	Persona encargada de gestionar las Publicaciones y los Comentarios.
Comentario	Almacena los comentarios que se desean enviar a una noticia.
Publicación	Almacena las publicaciones que se desean enviar a un sitio.
Blog	Almacena los datos de un blog específico.
Informes Estadísticos	Almacena los datos estadísticos de los blogs por semana, mes y año.
Enlaces	Almacena las URL de los enlaces de los blogs.
Medio	Almacena las URL de las páginas web donde se publican las noticias.
Posición	Almacena la posición ideológica del blog que se registra.
Plataforma	Almacena la plataforma donde está hosteado el blog.
Temática	Almacena el tema de lo que trata el blog (cultura, deporte, ciencia, etc.)

Tabla 3: Descripción de clases del modelo de dominio

2.3. Modelado del sistema

El sistema contendrá un conjunto de actores, los cuales serán capaz de interactuar con el mismo en dependencia de los permisos asignados a cada uno. En la **Tabla 4** se muestran los actores del sistema a desarrollar con sus respectivas descripciones según sus permisos. Hacer una correcta identificación de los requisitos que debe cumplir el sistema es muy importante, ya que un proyecto no puede ser exitoso sin una correcta especificación de los requerimientos que debe cumplir el sistema que se desea desarrollar. Una vez que se han definido los conceptos principales relacionados con el objeto de estudio y el dominio, se puede comenzar a analizar qué debe hacer el sistema para que cumpla con los objetivos planteados. Para ello, se enumeran a través de requerimientos funcionales y no funcionales, las acciones que el sistema deberá ser capaz de realizar.

2.3.1. Actores del sistema infoBlog v2

Actor	Descripción
Administrador	Es el rol encargado de realizar la gestión de los usuarios, roles, blogs, informes estadísticos, publicaciones, exportar informes, generar gráficas y velar por el funcionamiento del sistema.
Editor	Es el rol encargado de gestionar los tipos de contenido publicaciones y comentarios.
Comentarista	Es el rol encargado de gestionar el tipo de contenido comentario.

Tabla 4: Actores del sistema infoBlog v2

2.3.2. Requisitos funcionales

Ian Sommerville en la séptima edición del libro Ingeniería del Software plantea que “los requerimientos para un sistema son la descripción de los servicios proporcionados por el sistema y sus restricciones operativas.

Estos requerimientos reflejan las necesidades de los clientes de un sistema que ayude a resolver algún problema con el control de un dispositivo, hacer un pedido o encontrar información” (47). El propósito fundamental de la captura de los requisitos es guiar el desarrollo hacia el sistema correcto. Esto se consigue mediante una descripción de los requisitos del sistema que permita establecer un acuerdo entre el cliente y los desarrolladores sobre qué debe y qué no debe hacer el sistema.

1 – Autenticar Usuario	2 – Desactivar Usuario Automáticamente
3 – Ver Perfil de Usuario	4 – Asignar Permisos según los roles.
5 – Gestionar Usuarios 5.1 – Insertar Usuario 5.2 – Modificar Usuario 5.3 – Desactivar Usuario 5.4 – Listar Usuarios 5.5 – Ver Detalles	11 – Gestionar Nomencladores 11.1 – Temática 11.1.1 – Insertar Temática 11.1.2 – Modificar Temática 11.1.3 – Desactivar Temática 11.1.4 – Listar Temática
6 – Gestionar Publicaciones 6.1 – Insertar Publicación 6.2 – Modificar Publicación 6.3 – Ver Detalles de Publicación 6.4 – Eliminar Publicación 6.5 – Listar Mis Publicaciones 6.5.1 – Filtrar Mis Publicaciones por un rango de fecha	11.2 – Denominación 11.2.1 – Insertar Denominación 11.2.2 – Modificar Denominación 11.2.3 – Desactivar Denominación 11.2.4 – Listar Denominación 11.3 – Posición 11.3.1 – Insertar Posición 11.3.2 – Modificar Posición 11.3.3 – Desactivar Posición 11.3.4 – Listar Posición
7 – Gestionar Informes Estadísticos 7.1 – Insertar Informe Estadístico 7.2 – Modificar Informe Estadístico 7.3 – Ver Detalles de Informe Estadístico 7.4 – Eliminar Informe Estadístico 7.5 – Listar Informes Estadísticos 7.5.1 – Filtrar Informes Estadísticos por un rango de fecha	11.4 – Medios 11.4.1 – Insertar Medio 11.4.2 – Modificar Medio 11.4.3 – Desactivar Medio 11.4.4 – Listar Medios
8 – Gestionar Comentario 8.1 – Insertar Comentario 8.2 – Modificar Comentario 8.3 – Ver Detalles de Comentario 8.4 – Eliminar Comentario 8.5 – Listar Mis Comentarios 8.5.1 – Filtrar Mis Comentarios por un rango de fecha	11.5 – Autores 11.5.1 – Insertar Autor 11.5.2 – Modificar Autor 11.5.3 – Desactivar Autor 11.5.4 – Listar Autores
9 – Gestionar Blog 9.1 – Insertar Blog 9.2 – Modificar Blog 9.3 – Ver Detalles de Blog 9.4 – Eliminar Blog 9.5 – Listar Blogs.	11.6 – Plataforma 11.6.1 – Insertar Plataforma 11.6.2 – Modificar Plataforma 11.6.3 – Desactivar Plataforma 11.6.4 – Listar Plataforma 11.7 – Tipo de Informe 11.7.1 – Insertar Tipo de Informe

	11.7.2 – Modificar Tipo de Informe 11.7.3 – Desactivar Tipo de Informe 11.7.3 – Listar Tipo de Informe
10 – Gestionar Rol 10.1 – Insertar Rol 10.2 – Modificar Rol 10.3 – Desactivar Rol 10.4 – Eliminar Rol 10.5 – Listar Roles	12 – Gestionar Enlaces 12.1 – Insertar Enlaces 12.2 – Eliminar Enlaces 12.3 – Listar Enlaces de un Blog. 12.4 – Listar Todos los Enlaces.
13 – Exportar Informes 13.1 – Informe de Blogs 13.1.1 – Filtrar por un rango de fecha 13.1.2 – Filtrar por nomenclador Temática 13.1.3 – Filtrar por nomenclador Posición 13.1.4 – Filtrar por nomenclador Plataforma 13.1.5 – Filtrar por nomenclador Denominación 13.2 – Informe de Comentarios 13.2.1 – Filtrar por un rango de fecha 13.2.2 – Filtrar por usuario 13.3 – Informe de Publicaciones 13.3.1 – Filtrar por un rango de fecha 13.3.2 – Filtrar por nomenclador Medios 13.3.3 – Filtrar por nomenclador Autores 13.4 – Informe General 13.4.1 – Filtrar por un rango de fecha 13.4.2 – Filtrar por usuario	14 – Generar Gráficas 14.1 – Gráfica de Blog 14.1.1 – Filtrar por un rango de fecha 14.1.2 – Mostrar una tabla con la cantidad de blogs insertados y actualizados por usuarios. 14.2 – Gráfica de Comentarios 14.2.1 – Filtrar por un rango de fecha 14.2.2 – Mostrar una tabla con la cantidad de comentarios realizados por usuarios. 14.2.3 – Mostrar una tabla con la cantidad de comentarios por blog. 14.3 – Gráfica de Informes Estadísticos 14.3.1 – Filtrar por un rango de fecha 14.3.2 – Filtrar por el nomenclador Tipo de Informe 14.3.3 – Filtrar por un campo Blog. 14.4 – Gráfica de Com-Pub-Blog 14.4.1 – Filtrar por un rango de fecha
15 – Listar Contenidos 15.1 – Listar todos los comentarios de todos los usuarios 15.2 – Listar todos los comentarios de los usuarios activos 15.3 - Listar todas las publicaciones	

Tabla 5: Requisitos Funcionales

2.3.3. *Requisitos no funcionales*

Según Roger S. Pressman en la séptima edición del 2010 del libro Ingeniería de software define como requerimientos no funcionales, en la mayoría de los casos: “como las propiedades o cualidades que el sistema debe poseer. Sin embargo, desde otras aristas, pueden concebirse como las restricciones de las funcionalidades del sistema” (48). Los requerimientos no funcionales son las características que hacen el sistema usable, confiable y/o rápido. Estos requerimientos no aportan funcionalidades propias del sistema, pero son de gran importancia para el éxito del software y para lograr que este responda a las necesidades del cliente final.

Requerimientos de Usabilidad	
RNF 1	La aplicación solo podrá ser utilizada por los usuarios que estén registrados en el sistema.
RNF 2	El Sistema para la Gestión de la Información de la línea Weblog será una aplicación web.
RNF 3	El sistema debe permitir a usuarios con pocos conocimientos de informática poder interactuar con el mismo.
Requerimientos de Navegabilidad	
RNF 4	La aplicación web tendrá una interfaz fácil de navegar.
RNF 5	Todas las páginas de la aplicación web se encontrarán a no más de tres clic.
Requerimientos de Disponibilidad	
RNF 6	El sistema estará disponible las 24 horas del día los 7 días de la semana.
Requerimientos de Confiabilidad	
RNF 7	Los informes de comentarios, publicaciones, blogs y el informe general que se generarán serán con el formato definido por el centro CIDI.
RNF 8	La información no podrá ser modificada por usuarios no autorizados, protegiendo así la integridad de los datos.
Requerimientos de Licencia	
RNF 9	Se requiere el uso de herramientas y tecnologías de software libre, las cuales se podrán usar, modificar y distribuir libremente.
Requerimientos de Diseño e Implementación	
RNF 10	Se requiere el uso de PHP 5.4 en adelante como lenguaje de programación del lado del servidor.
RNF 11	Se requiere el uso de <i>Symfony</i> 2.3.6 como marco de trabajo.
RNF 12	Se requiere PostgreSQL 9.1 como gestor de bases de datos.
RNF 13	Se requiere del Mapeador de Objetos Relacionales (ORM por sus siglas en inglés <i>Object Relational Mapper</i>) <i>Doctrine2</i> para el manejo de la base de datos.
Requerimientos de Soporte	
RNF 14	El sistema debe dar la posibilidad de ser actualizado, así como de incorporarle nuevas funcionalidades en caso de ser necesarias.
Requerimientos de Hardware	
RNF 15	Para garantizar un buen funcionamiento del sistema, el servidor donde estará desplegado el sistema deberá ser Corei3, con una velocidad del procesador de 3.00 GHz, 3 GB de RAM y 160 GB de disco duro.
RNF 16	Para el cliente como requerimientos mínimos: Procesador Pentium III a 3.0GHz con 256 Mb de memoria RAM y una tarjeta de red.
Requerimiento de Software	
RNF 17	Las PC cliente deben tener instalado el Navegador Mozilla Firefox 25 en lo adelante y como Sistema operativo Linux o <i>Windows</i> .
RNF 18	Como servidor web Apache 2.2 o superior, con módulo PHP 5.4 disponible, este debe estar configurado con las extensiones "ldap ³ⁿ " y "pgsql"; un servidor de base de datos PostgreSQL 9.1; debe tener instalado como sistema operativo Linux; debe tener instalado el ORM <i>Doctrine2</i> . Se requiere que se configure el crontab del servidor donde se montará la aplicación con el siguiente comando: 00 17 * * * php /var/www/info_blog/app/console desactivar:usuarios.

Tabla 6: Requisitos no Funcionales

2.4. Diagrama de casos de uso del sistema

³ Protocolo Ligero de Acceso a Directorios (LDAP por sus siglas en inglés *Lightweight Directory Access Protocol*) que hacen referencia a un protocolo a nivel de aplicación que permite el acceso a un servicio de directorio ordenado y distribuido para buscar diversa información en un entorno de red. LDAP también se considera una base de datos (aunque su sistema de almacenamiento puede ser diferente) a la que pueden realizarse consultas.

Los casos de uso son una técnica para capturar y especificar los requisitos del sistema, así como para guiar su diseño, implementación y prueba. Cada caso de uso facilita un escenario sin ambigüedad en la interacción entre el actor y el software, así como describe un conjunto de funcionalidades que el sistema desarrollado debe poseer (49).

En la **Figura 2** se muestra el diagrama de casos de uso del sistema (DCUS).

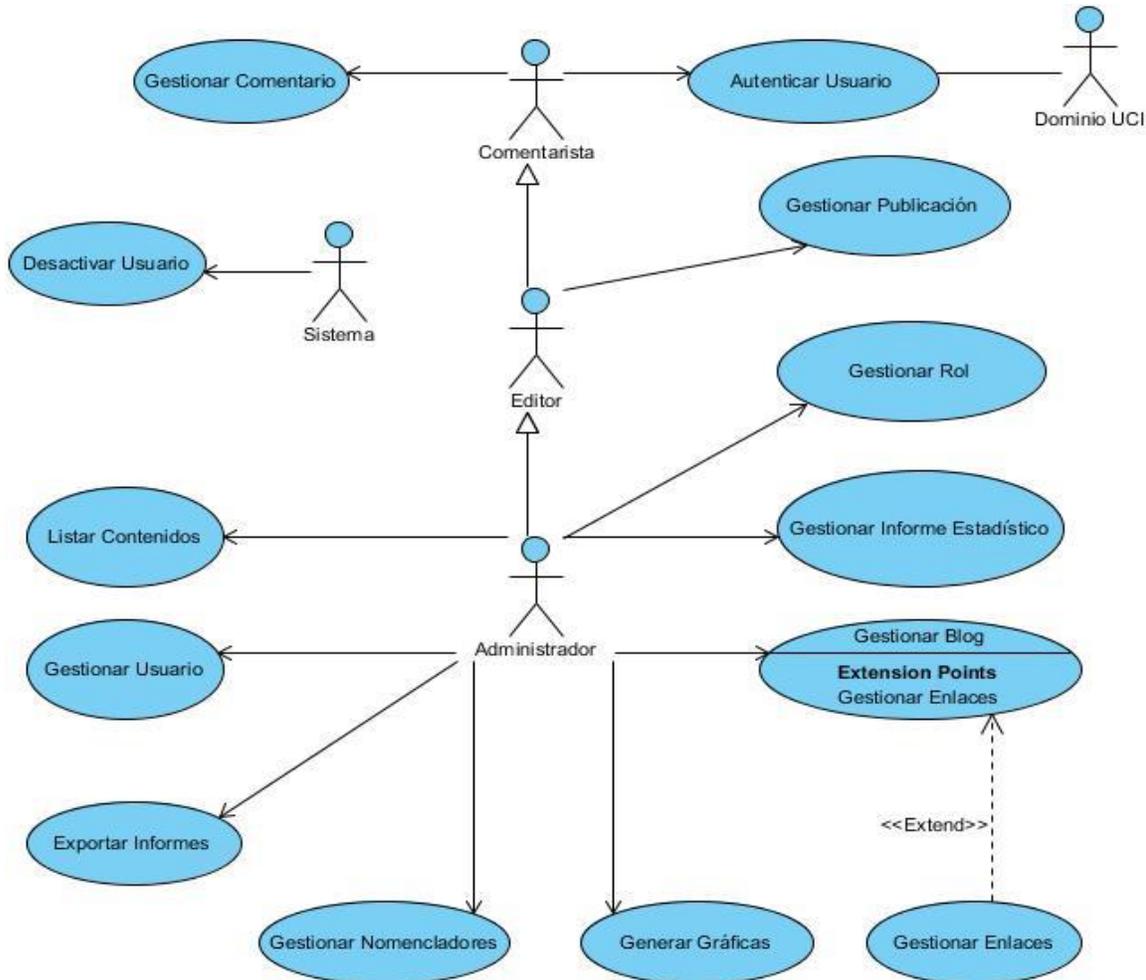


Figura 2: Diagrama de casos de uso del sistema infoBlog v2

2.4.1. Especificación de casos de uso (Descripción de casos de uso)

CU Gestionar Informe Estadístico

Objetivo	Gestionar contenidos de tipo Informe Estadístico
Actores	Administrador

Resumen	El CU se inicializa cuando el actor (en lo adelante usuario) quiere gestionar el contenido de tipo Informes Estadísticos.
Complejidad	Media
Prioridad	Alta (Critico)
Precondiciones	El usuario está autenticado en el sistema. El usuario pulsa la opción "Informes" del menú principal.
Postcondiciones	Se realizan satisfactoriamente cualquier acción sobre el tipo de contenido Informes Estadísticos.

Prototipo

Flujo de eventos

Flujo básico Gestionar Informes Estadísticos

	Actor	Sistema
1.	Pulsa la opción "Informe" del menú principal	
2.		Muestra un menú desplegable con las opciones siguientes: <ul style="list-style-type: none"> • Nuevo Informe. Ver Sección 1. • Listado de Informes Estadísticos. Ver Sección 2. • Exportar Informes. Ver CU Exportar Informes • Listado de Enlaces. Ver Sección 4 del CU Gestionar Enlaces.
3.	Selecciona una opción.	
4.		Muestra la vista referente a la opción que seleccionó.
5.		Termina CU

Sección 1: "Nuevo Informe" (Prototipo - 1)

Flujo básico Nuevo Informe

	Actor	Sistema
1.	Pulsa la opción "Nuevo Informe"	
2.		Muestra una vista con un formulario con los siguientes campos: <ul style="list-style-type: none"> • Blog

		<ul style="list-style-type: none"> • Fecha Inicial • Fecha Final • Mejores Publicaciones • Visitas por países • Tipo de Informe • Cantidad de visitas • Cantidad de Visitantes • Cantidad de publicaciones • Cantidad de comentarios • Seguidores del Blog • Seguidores comentarios • Seguidores twitter • Seguidores Facebook • Ranking Alexa
3.	Introduce los datos en los campos.	
4.	Pulsa la acción "Crear"	
5.		Valida que los datos sean correctos
6.		Almacena la información, redirecciona a la vista de "Insertar" y muestra un mensaje "Se ha creado el contenido satisfactoriamente"
7.		Termina Sección 1
Flujos alternos		
4 Datos Incorrectos		
	Actor	Sistema
1.		Verifica que los datos son incorrectos.
2.		Selecciona los campos incorrectos y muestra un mensaje.
3.		Ir al paso 3.
Sección 2: "Listado de Informes Estadísticos"		
Flujo básico Listado de Informes Estadísticos		
	Actor	Sistema
1.	Pulsa la opción "Listado de Informes Estadísticos"	
2.		<p>Muestra una vista con el listado de todos los Informes Estadísticos que se han creado. Además permite hacer las acciones siguientes:</p> <ul style="list-style-type: none"> • Insertar. Ver Sección 1. (Prototipo - 1) • Modificar. Ver Sección 3. (Prototipo - 3) • Ver Detalles. Ver Sección 4. (Prototipo - 2) • Eliminar. Ver Sección 5. (Prototipo - 4) <p>También muestra un filtro de búsqueda con los campos:</p> <ul style="list-style-type: none"> • Fecha Inicial • Fecha Final
3.	Selecciona una opción.	
4.		Termina Sección 2
Flujos alternos		
3 Filtrar la información.		
	Actor	Sistema
1.	Introduce los datos en los campos.	
2.	Pulsa la opción "Filtrar" (Prototipo - 5)	

3.		Hace una búsqueda según los datos introducidos y devuelve la información a la vista.
Sección 3: “Modificar Informe Estadístico” (Prototipo - 3)		
Flujo básico Modificar Informe Estadístico		
	Actor	Sistema
1.	Pulsa la opción “Editar”.	
2.		Muestra una vista con un formulario y los datos del Informe Estadístico que seleccionó en los siguientes campos: <ul style="list-style-type: none"> • Blog • Fecha Inicial • Fecha Final • Mejores Publicaciones • Visitas por países • Tipo de Informe • Cantidad de visitas • Cantidad de Visitantes • Cantidad de publicaciones • Cantidad de comentarios • Seguidores del Blog • Seguidores comentarios • Seguidores twitter • Seguidores Facebook • Ranking Alexa
3.	Modifica los datos.	
4.	Pulsa la opción “Actualizar”	
5.		Valida los datos entrados.
6.		Almacena la información, redirecciona a la vista de “Modificar” y muestra un mensaje “Se ha actualizado el contenido satisfactoriamente”
7.		Termina Sección 3
Flujos alternos		
4 Datos Incorrectos		
	Actor	Sistema
1.		Verifica que los datos son incorrectos.
2.		Selecciona los campos incorrectos y muestra un mensaje.
3.		Ir al paso 3.
Sección 4: “Ver Detalles” (Prototipo - 2)		
Flujo básico Ver Detalles		
	Actor	Sistema
1.	Pulsa la opción “Ver Detalles”.	
2.		Muestra una vista con la información de ese Informe Estadístico que seleccionó. Y la opción de “Modificar” Ver Sección 3.
3.		Termina Sección 4
Sección 5: “Eliminar Informe Estadístico” (Prototipo - 4)		
Flujo básico Eliminar Informe Estadístico		
	Actor	Sistema
1.	Pulsa la opción “Eliminar”.	

2.		Muestra un cuadro de diálogo con el mensaje “Desea eliminar este Informe Estadístico”.
3.	Pulsa la opción “Aceptar”	
4.		Elimina el Informe Estadístico y redirecciona a la vista de la lista de Informes Estadísticos en la que se encontraba y muestra un mensaje “Se ha eliminado el contenido satisfactoriamente”
5.		Termina Sección 5
Flujos alternos		
3 Pulsa opción “Cancelar”		
	Actor	Sistema
1.	Pulsa la opción “Cancelar”	
2.		Desaparece el cuadro de diálogo mostrado y redirecciona a la lista de Informes Estadísticos en la que se encontraba.
Relaciones	CU Incluidos	
	CU Extendidos	
Requisitos no funcionales		
Asuntos pendientes		

Tabla 7: CU Gestionar Informe Estadístico

En caso de necesitar consultar las descripciones de los demás casos de uso puede examinar el [Anexo A](#).

2.5. Descripción del sistema propuesto

2.5.1. Patrones de diseño implementados

Los patrones de diseño son el esqueleto de las soluciones a problemas comunes en el desarrollo de software. En otras palabras, brindan una solución ya probada y documentada a problemas de desarrollo de software que están sujetos a contextos similares. Se deben tener presente los siguientes elementos de un patrón: su nombre, el problema (cuando aplicar un patrón), la solución (descripción abstracta del problema) y las consecuencias (costos y beneficios) (50).

Los Patrones Generales de Software para Asignar Responsabilidades (GRASP por sus siglas en inglés *General Responsibility Assignment Software Patterns*) tienen una importante utilidad en el diseño de una aplicación al igual que los patrones Pandilla de los Cuatro (GoF por sus siglas en inglés *Gang of Four*). Los empleados en la solución de la problemática planteada se expondrán a continuación utilizando como referencia el libro de Craig Larman, UML y Patrones en la versión en español de la segunda edición (46):

Patrones GRASP:

Experto es un patrón para asignar una responsabilidad al experto en información: la clase que posee la información necesaria para cumplir con la responsabilidad. Con él no se pretende designar una idea oscura

ni extraña; expresa simplemente la intuición de que los objetos hacen cosas relacionadas con la información que poseen.

Symfony2 utiliza este patrón con la inclusión de *Doctrine2* para el mapeo de bases de datos. Se utiliza específicamente para crear una capa de abstracción en el modelo, encapsular toda la lógica de los datos y generar las clases entidades con todas las funcionalidades comunes (GET, SET y el constructor de la entidad); las clases de abstracción de datos poseen un grupo de funcionalidades que están relacionadas directamente con la entidad que representan y contienen la información necesaria de la tabla asociada.

En la implementación de la solución se pone de manifiesto en la clase *TbBlogController*, cuando se quiere eliminar un enlace de un blog, primero se necesita el blog del cual se quiere eliminar los enlaces, esta acción se realiza desde esta clase controladora, asignándole el permiso ya que es la que posee la información del blog que se seleccionó.

Creador es el patrón para asignar a la clase B la responsabilidad de crear una instancia de la clase A, si se cumple una se las siguientes condiciones: B contiene A, B agrega A, B tiene los datos de inicialización de A, B registra A, B utiliza A muy de cerca. Este patrón guía la asignación de responsabilidades relacionadas con la creación de objetos, tarea muy frecuente en los sistemas orientados a objetos. El propósito fundamental de este patrón es encontrar un creador que se debe conectar con el objeto producido en cualquier evento.

Este patrón es utilizado principalmente para el trabajo con las entidades que se almacenan en la base de datos, un ejemplo de su utilización es cuando se quiere crear una publicación y la clase *TbPublicacionController* es la encargada de instanciar los métodos de la entidad Publicación, que luego se almacenarán en la base de datos, debido a que obtiene todos los datos necesarios para la inicialización de los mismos.

Bajo Acoplamiento es el patrón para asignar las responsabilidades de modo que se mantenga bajo acoplamiento. El Bajo Acoplamiento estimula asignar una responsabilidad de modo que su colocación no incremente el acoplamiento tanto que produzca los resultados negativos propios de un alto acoplamiento. El patrón tiene la idea de tener las clases lo menos ligadas entre sí. De tal forma que en caso de producirse una modificación en alguna de ellas, se tenga la mínima repercusión posible en el resto de las clases disminuyendo la dependencia entre clases.

El Bajo Acoplamiento se evidencia en todo el proyecto ya que las clases controladoras como *TbBlogController*, *TbComentarioController*, *TbEnlacesController*, *TbPublicacionController*, etc, heredan únicamente de la clase controladora *Controller* para alcanzar un bajo acoplamiento de clases. Además de que ninguna clase controladora de la solución hereda de otra.

Alta Cohesión es el patrón para asignar las responsabilidades de modo que se mantenga bajo acoplamiento. Es un patrón evaluativo que el desarrollador aplica al valorar sus decisiones de diseño. Una clase de alta cohesión posee un número relativamente pequeño, con una importante funcionalidad relacionada y poco trabajo que hacer. Colabora con otros objetos para compartir el esfuerzo si la tarea es grande.

Este patrón se puede evidenciar en las siguientes clases controladoras: *TbComentarioController*, *TbPublicacionController*, *TnMediosController*, ya que estas clases solo se enfocan en manejar las responsabilidades correspondientes a las entidades con las que se relacionan manteniendo así la alta cohesión.

Controlador es un objeto de interfaz no destinada al usuario que se encarga de manejar un evento del sistema. Define además el método de su operación. Es el encargado de asignar la responsabilidad del manejo de un mensaje de los eventos de un sistema a una clase. La mayor parte de los sistemas reciben eventos de entrada externa, los cuales generalmente incluyen una interfaz gráfica para el usuario (IGU). Representa un escenario de caso de uso en el que tiene lugar el evento del sistema. Una de las opciones que brinda este patrón es que utiliza la misma clase controlador para todos los eventos del sistema en el mismo escenario de caso de uso.

Este patrón se pone de evidencia en todas las clases controladoras del sistema ya que cada uno de los eventos generados por el usuario es redirigido a una clase controladora que maneja estos eventos para luego realizar las operaciones solicitadas. Un ejemplo de esto es cuando el usuario crea un comentario y pulsa el botón Crear, el sistema dirige la petición a la clase controladora *TbComentarioController* y en esta se realiza la operación. Además, la mayoría de las clases de la solución se encargan de realizar los eventos del sistema referente a un caso de uso, ejemplo cuando se quiere gestionar una publicación, todos los eventos se dirigen a la clase controladora *TbPublicacionController*.

Patrones GoF:

Observador este patrón define, en sentido general, una dependencia de tipo uno-a-muchos entre objetos, de forma tal que cuando uno de los objetos cambia su estado, notifica su cambio a todos sus dependientes.

En la solución se evidencia este patrón en los objetos de tipo Blog, es decir en la clase entidad *TbBlog*, la cual presenta una relación uno-a-mucho con las entidades *TnPlataforma*, *TnPosicion*, *TnDenominacion* y *TnTematica*, de esta forma si se produce un cambio en alguna de estas cuatro entidades, la entidad Blog también cambia.

Decorador: *Symfony2* utiliza este patrón en la creación de plantillas generales para que otras las utilicen. Este patrón es utilizado en la solución con la creación de la plantilla *base.html.twig*, la cual poseerá en su contenido el código que será general para todas las interfaces de la solución, evitando de esta forma, tener que copiar este código en cada plantilla de la solución.

2.5.2. Patrón arquitectónico implementado por *Symfony2*

En la arquitectura de un software, los estilos definen la forma de organización arquitectónica de los subsistemas de información que conforman una aplicación. Estos indican los tipos de componentes y conectores involucrados, las restricciones de interconexión entre ellos, y el empleo de buenas prácticas para el diseño de un software. Según Javier Eguiluz en el libro Desarrollo web ágil con *Symfony2* plantea que “*Symfony2* basa su funcionamiento interno en la famosa arquitectura Modelo – Vista – Controlador (MVC) utilizada por la mayoría de *frameworks web*” (51).

El patrón arquitectónico MVC está formado por tres niveles:

- **Modelo:** representa la información con la que trabaja la aplicación, es decir, su lógica de negocio. En el caso de las aplicaciones desarrolladas con *Symfony2* se evidencia esta parte en las clases *Entity* (entidades). En la solución se evidencia en las clases como *TbBlog*, *TbPublicacion* y *TbComentario*, entre otras.
- **Vista:** transforma el modelo en una página web que permite al usuario interactuar con ella. En las aplicaciones desarrolladas con *Symfony2* se evidencia en los archivos *.html.twig* que son el formato de las vistas que proporciona. En la solución se ejemplifica con las vistas *edit.html.twig*, *new.html.twig* y *show.html.twig*, entre otras.
- **Controlador:** se encarga de procesar las interacciones del usuario y realiza los cambios apropiados en el modelo o en la vista. En la solución se pone de manifiesto esta parte en las clases *TbBlogController*, *TbPublicacionController* y *TbComentarioController* entre otras.

La arquitectura MVC separa la lógica del negocio (el modelo) y la presentación (la vista) por lo que se consigue un mantenimiento más sencillo de las aplicaciones. El controlador se encarga de aislar al modelo y a la vista de los detalles del protocolo utilizado para las peticiones (HTTP, consola de comandos, email, etc.). El modelo se encarga de la abstracción de la lógica relacionada con los datos, haciendo que la vista y las acciones sean independientes de, por ejemplo, el tipo de gestor de bases de datos utilizado por la aplicación (52).

En la siguiente figura se muestra el funcionamiento de *Symfony2* basado en el patrón arquitectónico Modelo – Vista – Controlador.

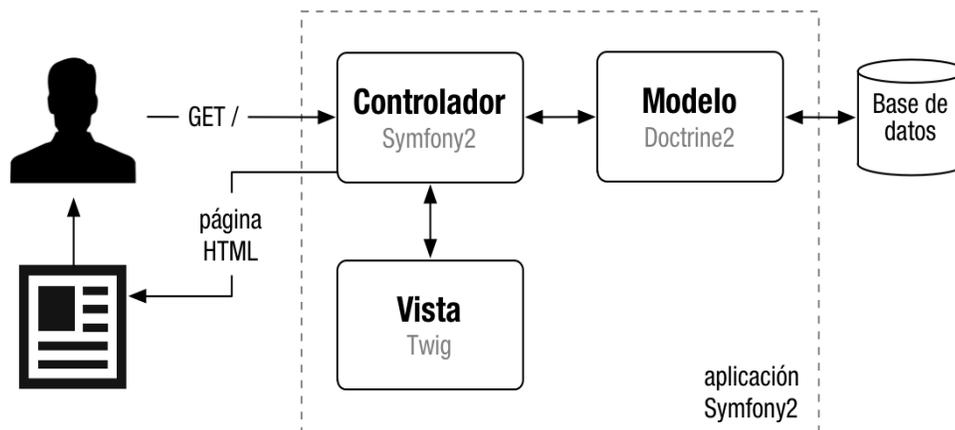


Figura 3: Patrón arquitectónico MVC⁴

2.5.3. Estructura en bundles de la aplicación web

El código de las aplicaciones y el propio código fuente de *Symfony2* se estructura mediante *bundles*⁵. Los *bundles* de las aplicaciones *Symfony2* suelen contener clases PHP y archivos web (*JavaScript*, *CSS* e imágenes). No obstante, no existe ninguna restricción sobre lo que puedes incluir dentro de un *bundle*. Tampoco existen límites técnicos sobre el tamaño que puede llegar a tener el mismo (51).

⁴ Tomado de Desarrollo web ágil con *Symfony2* de Javier Eguluz, 2013.

⁵ Los *bundles* son la base de la nueva filosofía de trabajo de *Symfony2*. Técnicamente, un *bundle* es un directorio que contiene todo tipo de archivos dentro una estructura jerarquizada de directorios.

Para el desarrollo del Sistema para la Gestión de la Información de la línea Weblog se decide crear un proyecto de *Symfony2* y utilizar tres *bundles*:

FrontendBundle: es donde se guardarán las páginas estáticas del sistema, como la página de Ayuda, la de Contacto y la página Sobre-Nosotros, además de la plantilla base de la cual las demás vistas que se mostrarán, heredarán de ella.

ContenidoBundle: es el encargado de gestionar todo el contenido de la aplicación web, los blog, las publicaciones, los comentarios, etc.

BackendBundle: es el que se encarga de gestionar los usuarios que se insertarán en el sistema, además de gestionar los roles.

2.5.4. **Seguridad en el sistema**

En el desarrollo de una aplicación web, la seguridad del sistema es uno de los aspectos más importantes, ya que es la garantía de la confidencialidad, integridad y disponibilidad de la información. Muchas aplicaciones web se ven afectadas en muchas ocasiones por diferentes vulnerabilidades del propio código fuente o muchas veces por descuido de los desarrolladores. En este sentido *Symfony2* implementa una serie de mecanismos para garantizar una aplicación segura, definiendo una lista de control de acceso, en el fichero *app/config/security.yml*.

Además de la seguridad que proporciona el *framework Symfony2*, tienen acceso a la aplicación solo las personas autorizadas, definiendo esto como un mecanismo de autenticación basado en un usuario y una contraseña que serán las mismas del dominio UCI. Además se crean roles para ser asignados a los usuarios y así definir los permisos, utilizando un modelo de control de acceso basado en roles. Con estos procedimientos se asegura que sólo un usuario correctamente identificado pueda acceder al sistema, garantizando la confidencialidad e integridad de los datos, y estos usuarios solo podrán accionar sobre la información precisa que por el rol establecido en el sistema pueden controlar.

2.6. Diagramas de Clases del Diseño

Los diagramas de clase del diseño se usan para describir la estructura de un sistema, formalizan el conocimiento del dominio de la aplicación. Las clases son abstracciones que especifican los atributos y comportamientos de un conjunto de objetos (53).

Las siguientes figuras muestran los Diagramas de Clases del Diseño con Estereotipos Web (DCDEW) de los casos de uso expuestos anteriormente en la especificación de casos de uso.

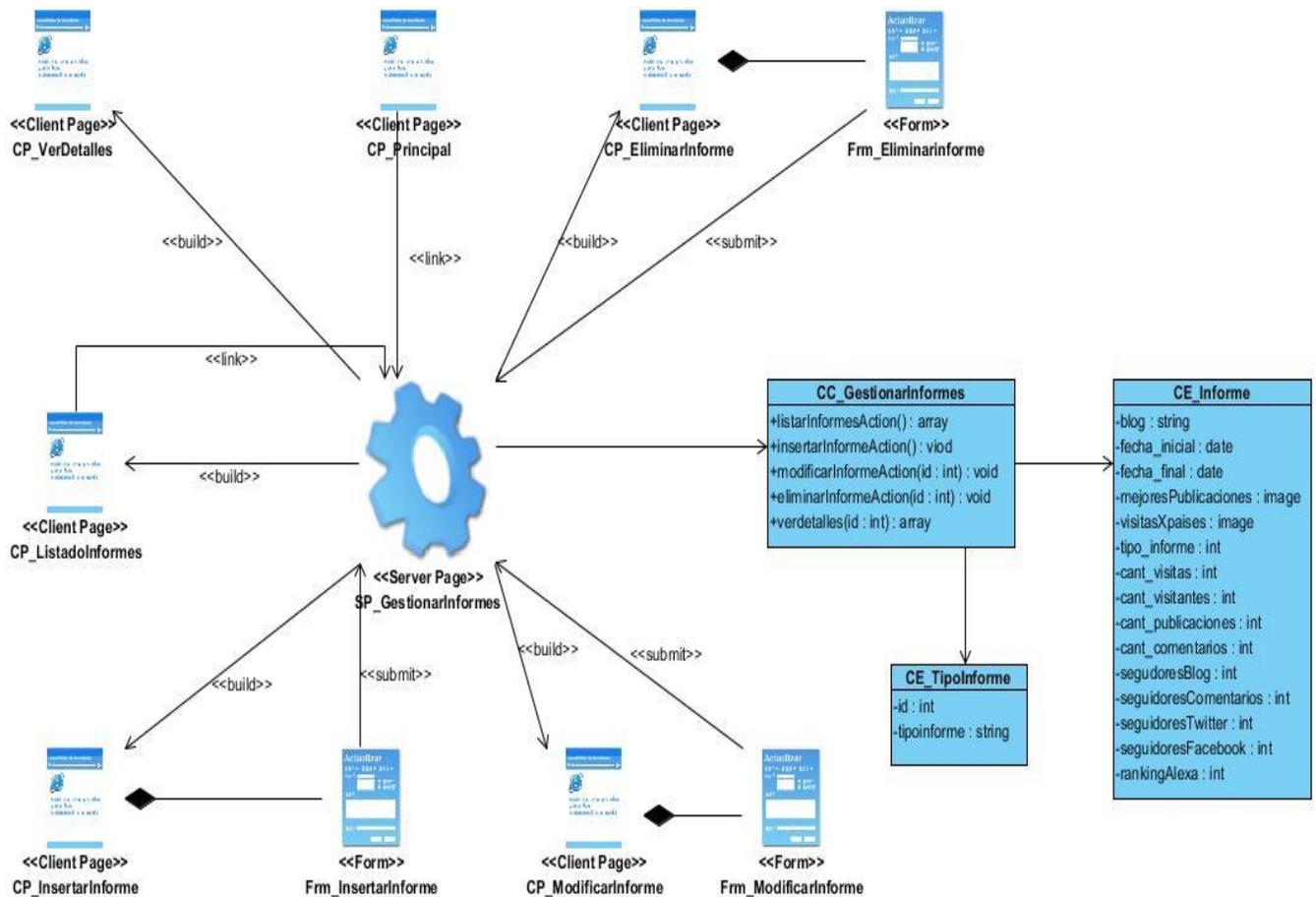


Figura 4: DCDEW del CU Gestionar Informe Estadístico

En caso de necesitar consultar los diagramas de clases del diseño de los demás casos de uso puede examinar el [Anexo B](#).

2.7. Diagramas de Interacción

Los diagramas de interacción representan la forma en que un cliente u objeto se comunica entre sí en petición a un evento. No son sólo importantes para modelar los aspectos dinámicos de un sistema, sino también para construir sistemas ejecutables por medio de ingeniería directa e inversa (54).

Los siguientes diagramas muestran la interacción entre las clases y los actores de los casos de uso expuestos anteriormente en la especificación de casos de uso.

2.7.1. Diagramas de Secuencia

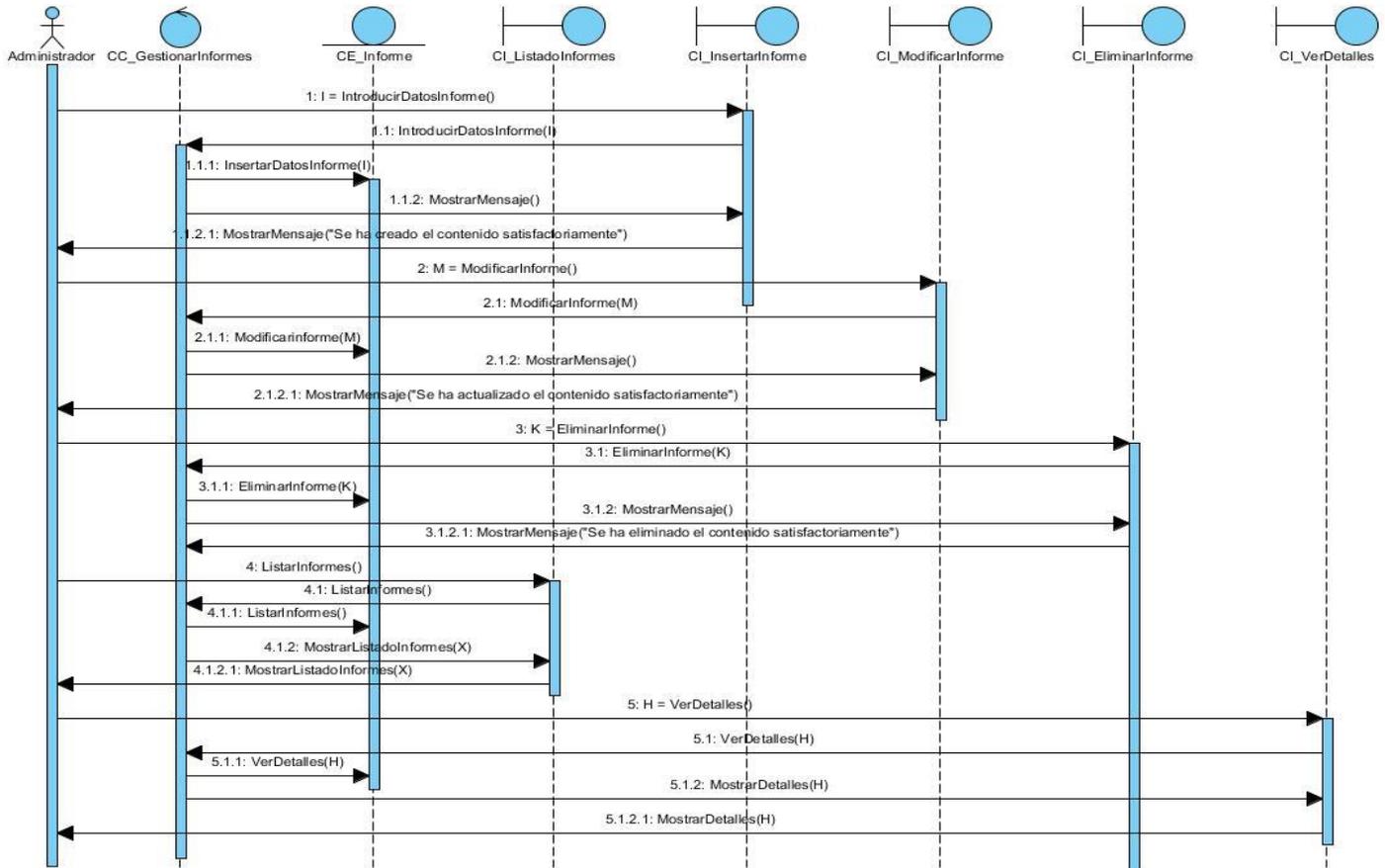


Figura 5: Diagrama de Secuencia del CU Gestionar Informes Estadísticos

En caso de necesitar consultar los diagramas de secuencia de los demás casos de uso puede examinar el [Anexo C](#).

2.8. Diseño de la Base de Datos

A continuación se muestra una figura que presenta el modelo de Base de Datos del sistema que se desea desarrollar, que de una manera física y estructurada, muestra cómo se almacenan, relacionan, organizan y manipulan los datos del sistema.

El modelo de base de datos que se muestra está compuesto por 19 tablas, de ellas 8 de nomencladores que son los que se relacionan con los contenidos y los usuarios en el caso del nomenclador rol y su nombre comienza por el identificador **tn_**, otras 4 que representan las relaciones mucho a mucho entre dos tablas y su nombre comienza con el identificador **tr_**, una que almacena las actualizaciones de los blog que su nombre es **tr_traza_user_blog**, y las demás tablas son de contenidos y su nombre comienza con el identificador **tb_**.

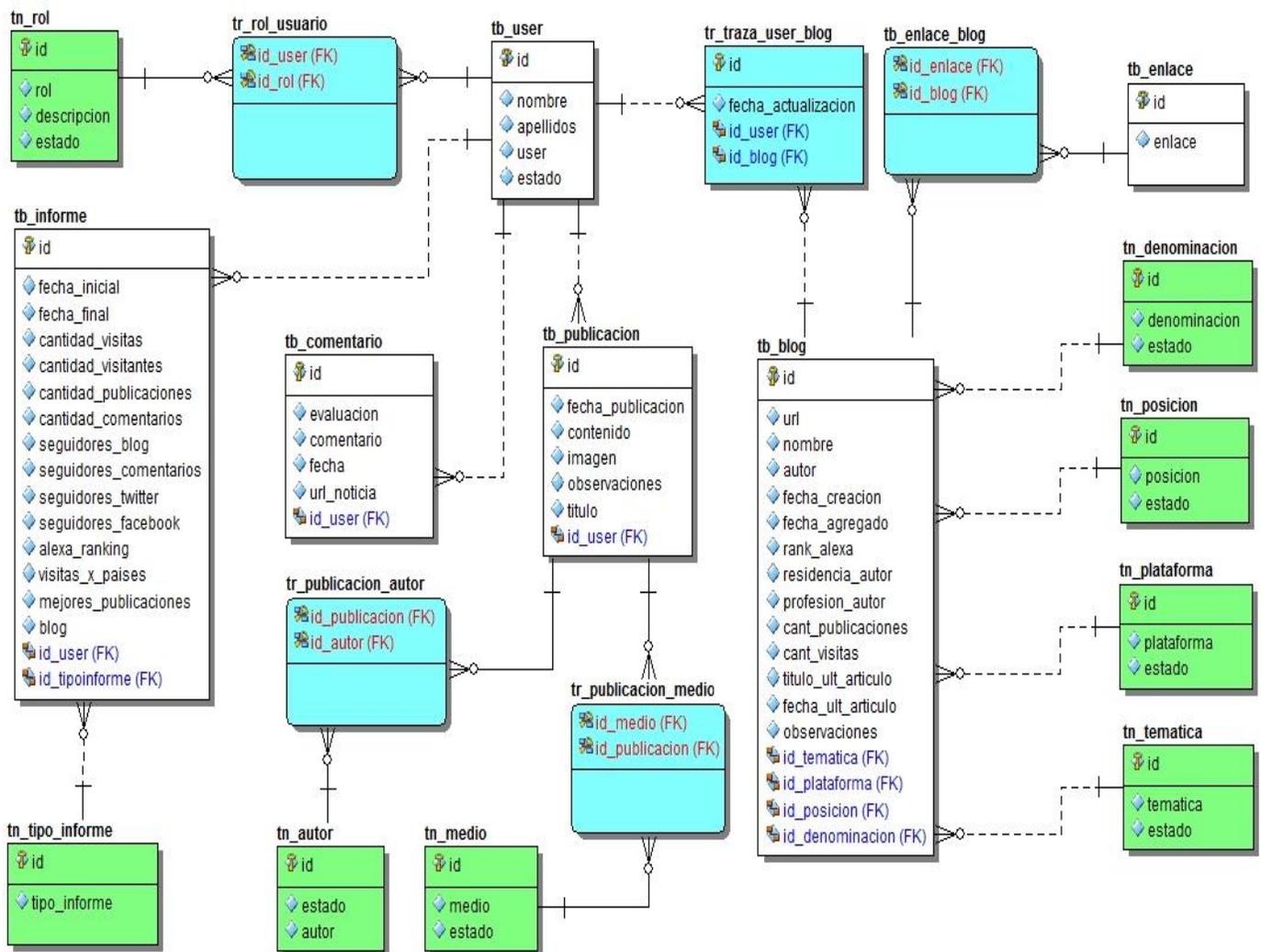


Figura 6: Modelo físico de la base de datos del sistema

2.9. Diagrama de despliegue

Un Diagrama de Despliegue modela la arquitectura en tiempo de ejecución de un sistema. Esto muestra la configuración de los elementos de hardware (nodos) y muestra cómo los elementos y artefactos del software se trazan en esos nodos.

En la siguiente figura se muestra de la forma que el usuario tendrá acceso a la aplicación, esto será a través de un navegador web en una PC Cliente, con una conexión por HTTPS. La aplicación se conecta al servidor de base de datos a través del protocolo TCP/IP y realiza las consultas a la BD de la aplicación.

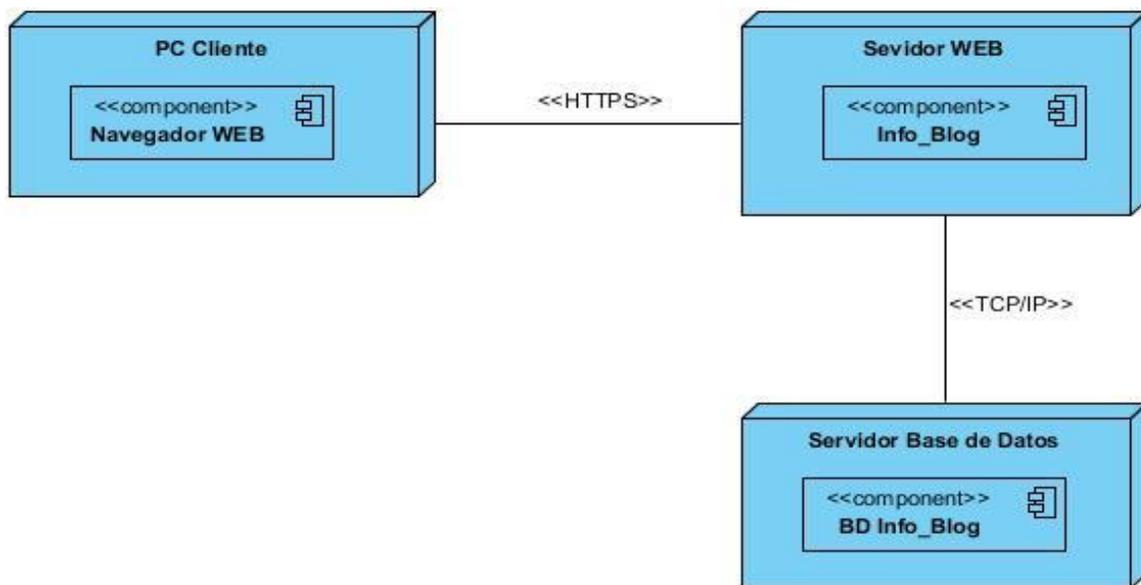


Figura 7: Diagrama de despliegue

2.10. Conclusiones parciales

- La modelación de los diferentes diagramas de clases del diseño, secuencia, modelo de la base de datos, despliegue, casos de uso, ayudó al desarrollador a implementar fácilmente la aplicación.
- Los patrones de diseño empleados en el desarrollo de la solución, propiciaron una arquitectura sólida y robusta al sistema.
- La estructura que presenta la solución permite una mayor organización en el desarrollo, además de una mejor distribución en la seguridad de la aplicación web.

CAPÍTULO 3. IMPLEMENTACIÓN Y PRUEBAS DEL SISTEMA PARA LA GESTIÓN DE LA INFORMACIÓN DE LA LÍNEA WEBLOG

3.1. Introducción

La implementación es el centro durante las iteraciones de la fase construcción, aunque también se lleva a cabo durante la fase de transición, para tratar otros defectos encontrados en dicha fase. En el flujo de trabajo de pruebas se verifica el resultado de la implementación, probando las funcionalidades del sistema. En el presente capítulo se presentará el diagrama de componentes del sistema, se expondrán los estándares de codificación empleados en el desarrollo, se mostrarán algunas interfaces de la aplicación web y se describirán algunas pruebas realizadas al sistema infoBlog v2. Las pruebas se realizan con el objetivo de validar el sistema desarrollado y detectar posibles no conformidades.

3.2. Diagrama de componentes

Un diagrama de componentes muestra los componentes de software, controladores embebidos, entre otros, que conformarán un sistema. Un diagrama de componentes tiene un nivel más alto de abstracción que un diagrama de clase, usualmente un componente se implementa por una o más clases (u objetos) en tiempo de ejecución (55). Para un mayor entendimiento los diagramas de componentes describen los elementos físicos del sistema y sus relaciones. Los componentes representan todos los tipos de elementos del software que intervienen en la fabricación de la aplicación informática. Pueden ser simples archivos, paquetes o librerías cargadas dinámicamente.

A continuación se muestra el diagrama de componentes generales que utiliza la aplicación web infoBlog v2. Para una mayor comprensión de los componentes desarrollados se muestran varias vistas detalladas de los principales paquetes que son *ContenidoBundle*, *BackendBundle* y *FroontendBundle*.

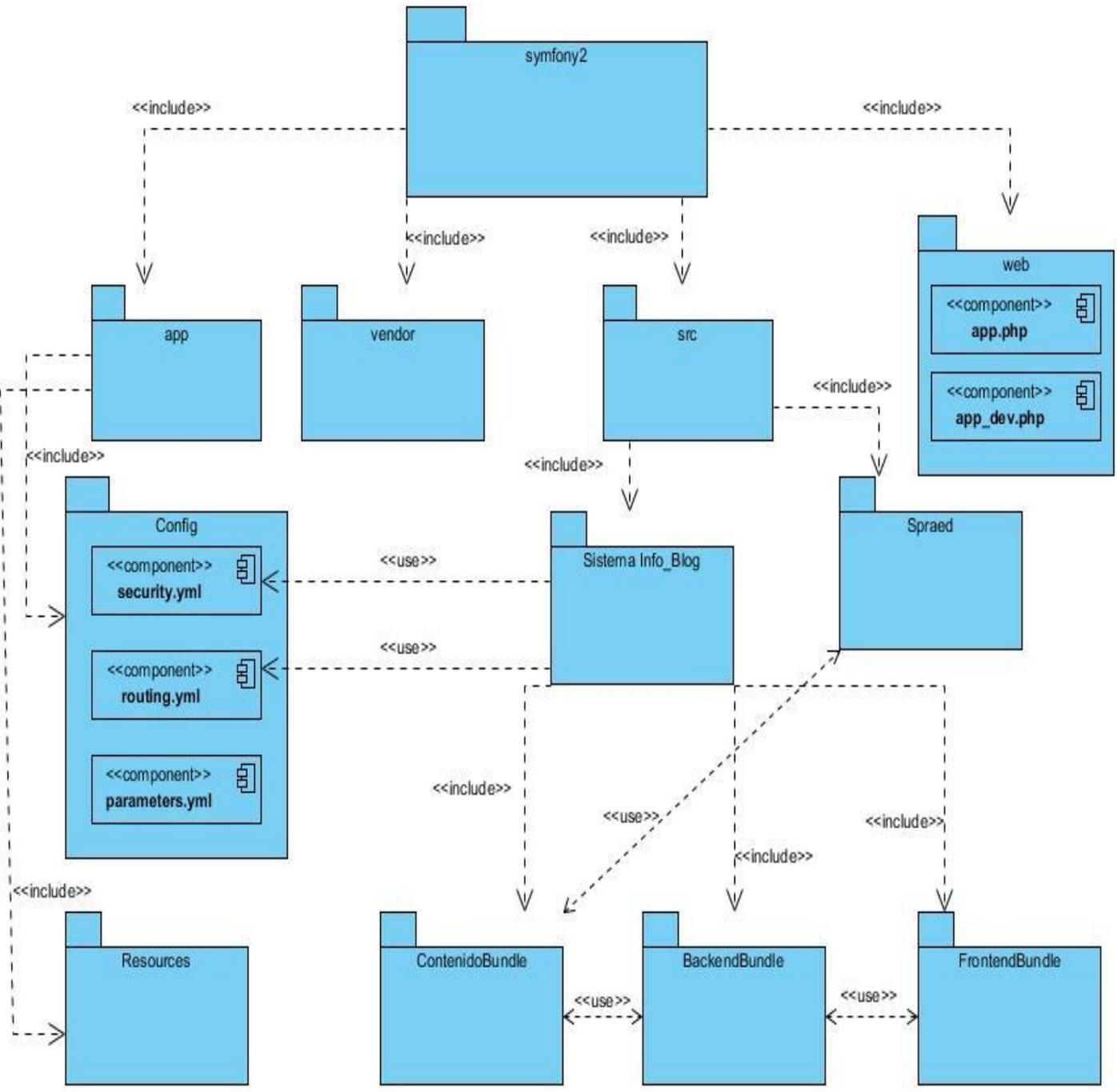


Figura 8: Diagrama de componentes de la aplicación infoBlog v2

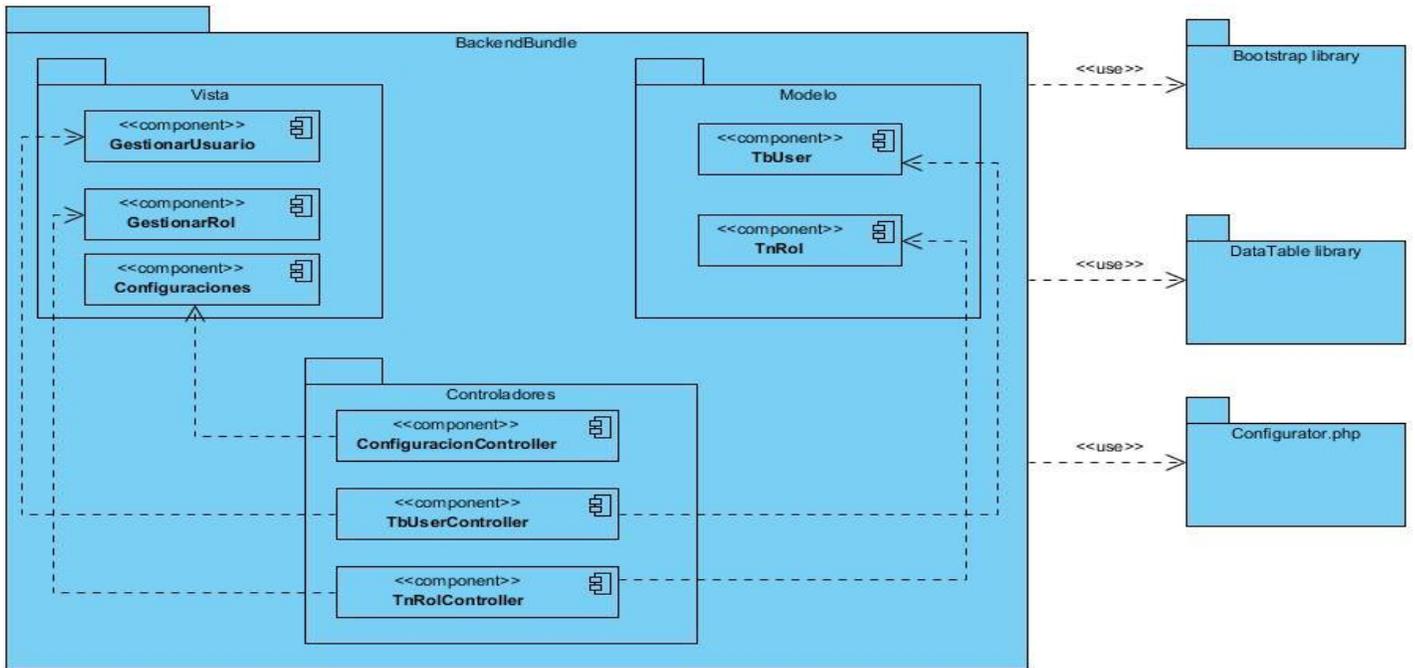


Figura 9: Diagrama de componentes del paquete *BackendBundle*

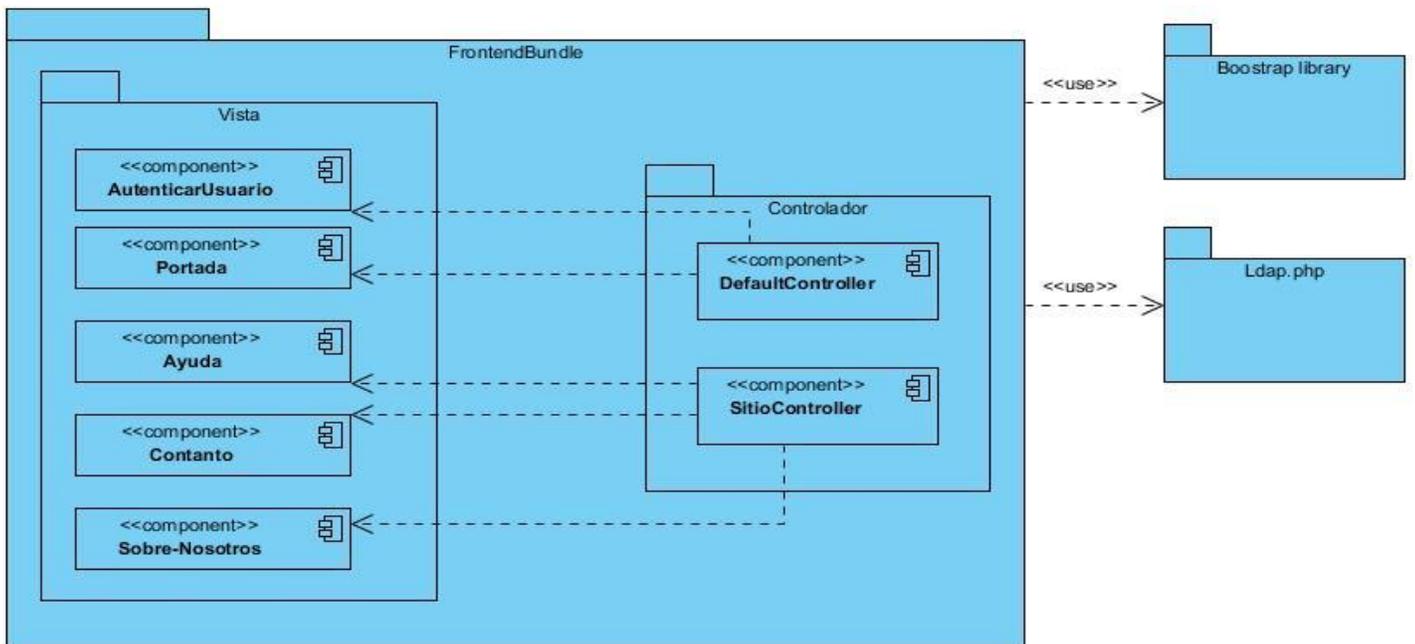


Figura 10: Diagrama de componentes del paquete *FrontendBundle*

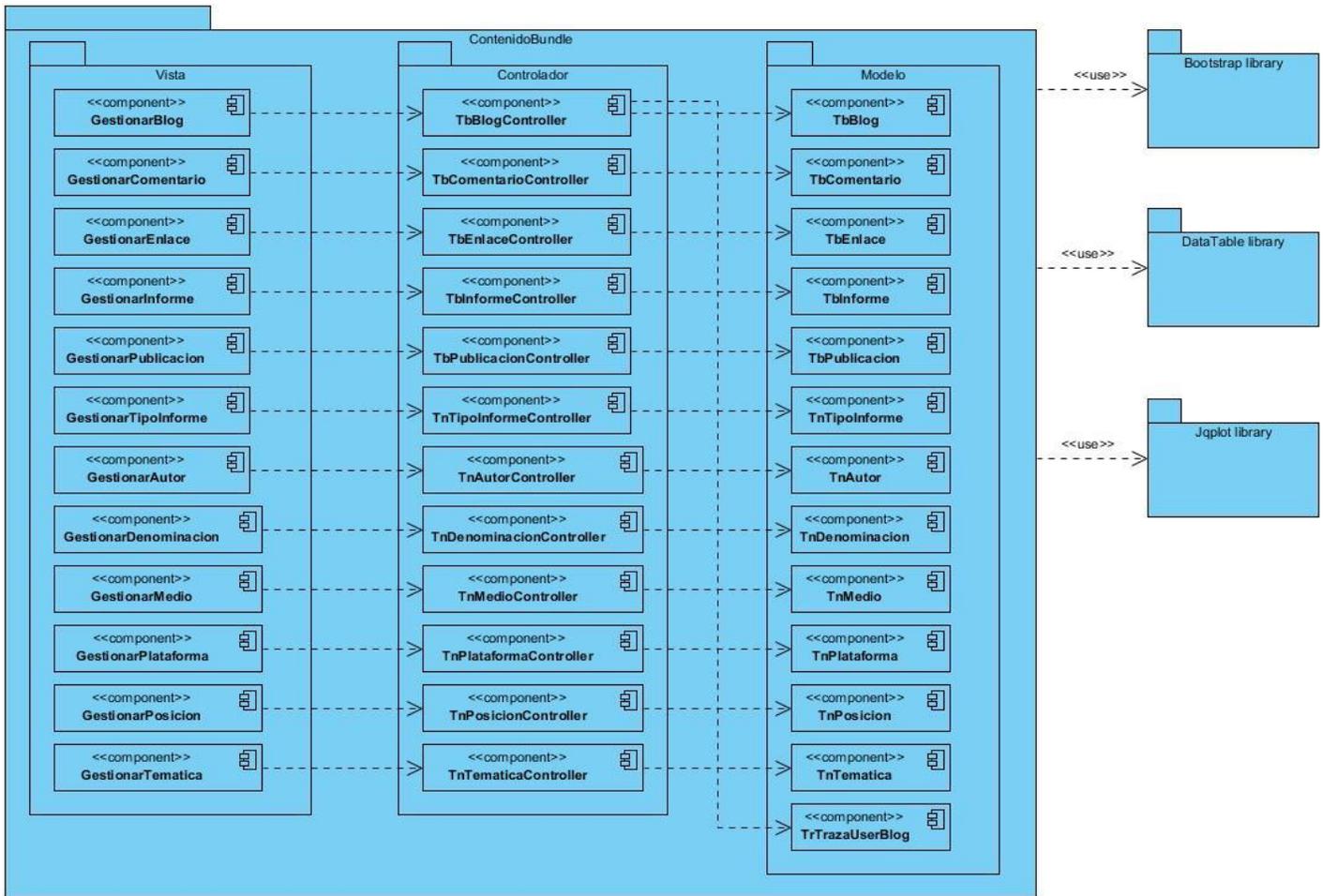


Figura 11: Diagrama de componentes del paquete *ContenidoBundle*

3.3. Estándares de codificación

Un estándar es un modelo, norma, patrón, referencia o la especificación que regula la realización de ciertos procesos o la fabricación de componentes para garantizar la interoperabilidad. Para la implementación de la aplicación web se estandarizará el código para que exista legibilidad y organización a la hora de darle mantenimiento y seguimiento a la aplicación. A continuación se especifican los estándares de código a utilizar en la construcción de la solución:

Camel Case: la notación "Camel Case" consiste en escribir frases o palabras compuestas eliminando los espacios intermedios y poniendo en mayúscula la primera letra de cada palabra.

Lower Camel Case: similar al *Camel Case* sólo que la primera letra de la primera palabra es también en minúscula.

Upper Camel Case: similar al *Camel Case* siendo la primera letra de la primera palabra en mayúscula.

El nombre de las clases debe ser declarado utilizando el estilo *Upper Camel Case*.

```
class TbComentarioController extends Controller
{
    /**
     * Lists all TbComentario entities.
     */
    public function indexAction()
    {
```

Figura 12: Imagen que muestra el nombre de las clases

El nombre de los procedimientos debe ser declarado utilizando el estilo *Lower Camel Case*.

```
public function graficarInformeAction(){
public function graficarInformeFullAction(Request $request){
public function graficarComPubBlogAction(){
```

Figura 13: Imagen que muestra el nombre de los métodos en una clase

Todo el código PHP debe estar encerrado entre las etiquetas `<?php...?>`, pero en caso de que el fichero solo contenga código escrito en PHP se omitirá la etiqueta de cierre `?>`.

1...//Código no PHP

2...<?php

3...//Código PHP

4...?>

1...<?php

2...//Código PHP

El código debe tener una indentación de cuatro espacios o tabulaciones y la longitud máxima que puede alcanzar una línea es de doscientos caracteres, las que excedan este tamaño deben ser divididas en varias líneas con un tamaño que no supere esta cantidad. Además no deben existir espacios en blanco al final de las líneas que no estén vacías.

1.//Código(longitud máxima **200 caracteres**)

2.//Código(longitud máxima **200 caracteres**)

Las constantes de PHP **true**, **false** y **null** deben estar escritas en minúsculas.

1.\$estaEIResultado = **true**;

Cuando en un fichero se define el espacio de nombre (*namespace*) debe existir una línea en blanco a continuación de la declaración. Además cuando están presentes todas las declaraciones de uso se escriben a continuación de la definición de espacio de nombre, se debe utilizar la palabra reservada **use** una vez por cada declaración y debe concluir con una línea en blanco.

```
<?php
namespace Info_blog\ContenidoBundle\Controller;

use Symfony\Component\HttpFoundation\Request;
use Symfony\Bundle\FrameworkBundle\Controller\Controller;
use Info_blog\ContenidoBundle\Entity\TbInforme;
use Info_blog\ContenidoBundle\Form\TbInformeType;
use Symfony\Component\HttpFoundation\Response;
use Symfony\Component\HttpFoundation\RedirectResponse;

/**
 * TbInforme controller.
 */
```

Figura 14: Imagen que muestra los *namespace* y los *use* en una clase

Los atributos de las clases deben ser privados, además de que no debe existir más de un atributo por línea y deben utilizar el estilo *Lower Camel Case*.

```
private $fechaPublicacion;

/**
 * @var string
 *
 * @ORM\Column(name="contenido", type="text", nullable=true)
 * @Assert\NotBlank()
 */
private $contenido;

/**
 * @var string
 *
 * @ORM\Column(name="imagen", type="string", nullable=true)
 * @Assert\Image()
 */
private $imagen;
```

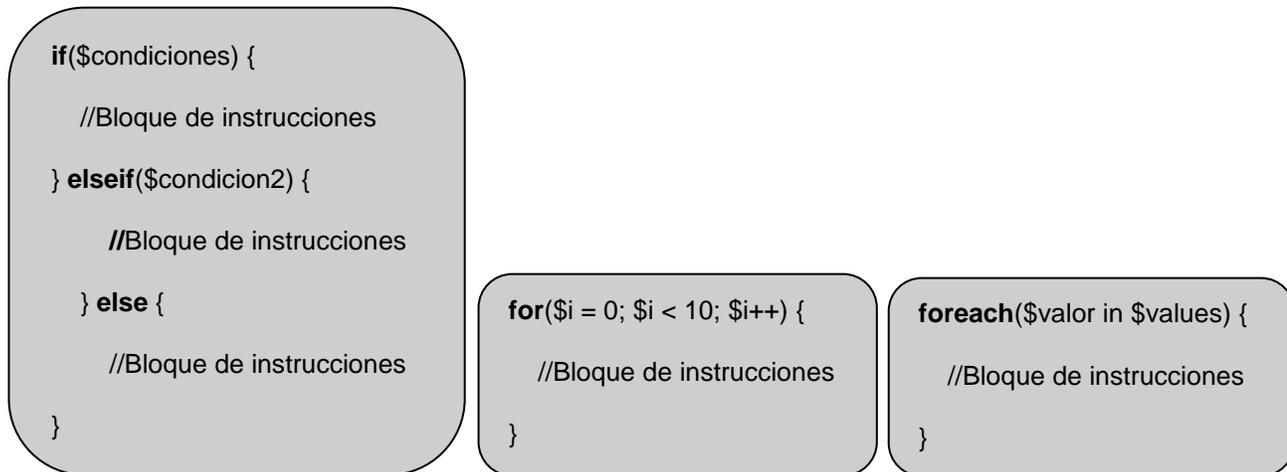
Figura 15: Imagen que muestra los nombres de los atributos de las clases

En los procedimientos o métodos de las clases, no deben existir espacios en blanco a continuación de los nombres de los mismos, ni después del paréntesis abierto, ni antes del paréntesis cerrado de los parámetros. Las llaves deben colocarse una en cada línea de forma que la que abre se encuentre a continuación de la declaración del método y la que cierra a continuación del bloque de instrucciones.

```
private function createDeleteForm($id)
{
    return $this->createFormBuilder()
        ->setAction($this->generateUrl('tblog_delete', array('id' => $id)))
        ->setMethod('DELETE')
        ->add('submit', 'submit', array('label' => 'Delete'))
        ->getForm()
    ;
}
```

Figura 16: Imagen que muestra el estilo de los métodos de una clase

Para las estructuras de control se define que no debe existir un espacio en blanco a continuación de la palabra reservada que define la estructura de control, no debe existir un espacio en blanco ni después del paréntesis abierto, ni antes del paréntesis cerrado y debe existir un espacio en blanco entre el paréntesis que cierra y la llave que abre el bloque de instrucciones.



Se deben añadir los bloques de documentación al principio de cada clase y procedimiento de la forma que se muestra a continuación.

```

/**
 * TbPublicacion controller.
 *
 */
class TbPublicacionController extends Controller
{
    /**
     * Lists all TbPublicacion entities.
     *
     */
    public function indexAction()
    {
        $em = $this->getDoctrine()->getManager();
    }
}

```

Figura 17: Imagen que muestra los comentarios en las clases y en los métodos

3.4. Interfaces de la aplicación infoBlog v2

A continuación se presentan varias interfaces de la aplicación web infoBlog v2, que se desarrolló. En el [Anexo D](#) aparecen otras imágenes del sistema.

La imagen que se muestra a continuación es la portada de la aplicación web infoBlog v2, donde se muestra un menú principal con las opciones a las que el usuario autenticado tiene acceso según el rol o los roles que posea. Este menú con las opciones se repite en todas las interfaces del sistema. La interfaz cuenta con un área para el contenido, que es donde se gestiona toda la información, y muestra un *footer*⁶ que también se repite en todas las interfaces, el cual muestra las páginas estáticas de Ayuda, Contacto y Sobre-Nosotros.



Figura 18: Imagen de la portada de la aplicación web infoBlog v2

⁶ Pie de página de la aplicación.

La imagen que se muestra a continuación es la de gestionar las publicaciones. La interfaz contiene el menú principal y el *footer*, y en el área de contenido muestra el listado de las publicaciones creadas por el usuario que esta autenticado, permitiéndole las opciones de Crear, Editar, Eliminar y Ver Detalles de una publicación, igualmente presenta un filtro para hacer una búsqueda por un rango de fechas.

The screenshot shows the 'Mis Publicaciones' (My Publications) page. At the top, there is a blue navigation bar with the title 'Info_Blog' and a user greeting 'Bienvenido(a) Nolberto Rojas Pérez'. Below the navigation bar, there is a search and filter section with input fields for 'Fecha Inicial' (2014-05-10) and 'Fecha Final' (2014-05-10), and a 'Filtrar' button. To the right, there is a '+ Nueva Publicación' button and a search box labeled 'Buscar:'. Below this, there is a table of publications with columns for 'Autores', 'Medio', 'Fecha', 'Título', and 'Acciones'. The table shows one publication by 'Ilismay Echevarria' with the URL 'http://firefoxmania.uci.cu', date '07-05-2014', and title 'asfasdgg'. At the bottom, there are pagination controls showing 'Mostrando desde 1 hasta 1 de 1 registros' and buttons for 'Primero', 'Anterior', '1', 'Siguiente', and 'Último'.

Figura 19: Imagen que muestra la interfaz de Mis Publicaciones de la aplicación web infoBlog v2

3.5. Pruebas de software

Las pruebas son un conjunto de actividades que se planean con anticipación y se realizan de manera sistemática. Por tanto, se debe definir una Plantilla para las Pruebas de Software⁷ (56). Las pruebas se realizan con el objetivo de encontrar errores cometidos al realizar el diseño y construcción de un producto de software. A la aplicación web infoBlog v2 se le hicieron un conjunto de pruebas que se explicarán en este epígrafe.

3.5.1. Pruebas unitarias y de integración

Las pruebas unitarias se realizan con el objetivo de comprobar que un módulo, entendido como una unidad funcional de un programa independiente, está correctamente codificado. En estas pruebas cada módulo se aprueba por separado y lo hará, generalmente, la persona que lo creó.

⁷ Un conjunto de pasos en el que se puedan incluir técnicas y métodos específicos del diseño de casos de pruebas.

En la aplicación web infoBlog v2, se probaron los *bundles* internos del sistema por separado, para analizar el correcto funcionamiento de las diferentes funcionalidades en las distintas clases. Mediante las pruebas de integración se verificó la operación conjunta de cada uno de estos componentes del sistema haciendo énfasis en la integración entre ellos, es decir que se comprobó el correcto funcionamiento de las funcionalidades que debe cumplir la aplicación.

3.5.2. Pruebas de funcionalidad

Este tipo de prueba se realizó con el objetivo de validar el correcto funcionamiento del software. Es decir, se aplicaron un conjunto de pruebas buscando fallas funcionales a la aplicación web, identificando situaciones en las que la respuesta de la aplicación, a una acción del usuario, no es la esperada. El tipo de prueba se le aplicó a todos los casos de uso definidos, con diferentes entradas de datos por los usuarios, para determinar que los resultados obtenidos fueran los esperados bajo cualquier situación, esto permitió comprobar el adecuado funcionamiento de los requerimientos funcionales definidos por el cliente.

A continuación se presenta una muestra de los casos de pruebas que se elaboraron para cada caso de uso, utilizando la técnica de Caja Negra⁸. Seguidamente se detallan las pruebas de funcionalidad realizadas al caso de uso Gestionar Publicación:

Descripción de las variables

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	Url de la Noticia	texto	No	Campo donde se puede escribir.
2	Comentario	textarea	No	Campo donde se puede escribir.
3	Fecha	date	No	Campo donde puedes escribir o seleccionar una fecha.
4	Evaluación	Select	No	Campo donde seleccionar una de las opciones.

Tabla 8 : Descripción de las variables. Prueba de Funcionalidad para el CU Gestionar Comentario

SC Insertar Comentario

Escenario	Descripción	Url de la noticia	Comentario	Fecha	Respuesta del sistema	Flujo central
EC 1.1 Insertar Comentario correctamente	El sistema inserta el Comentario correctamente	V http://inter nos.uci.cu	V Un comentario es válido si contiene más de 60 palabras en	V 2014-05-02	El sistema guarda los datos en la base de datos y muestra el mensaje " Se ha creado el contenido "	El usuario se autentica en el sistema con rol de Comentarista, Editor o Administrador. 1. El usuario presiona la opción Comentarios en el menú principal.

⁸ Las pruebas que se aplican a la interfaz de un sistema buscando errores de funcionalidad y demostrando que cada función es plenamente operacional.

			su contenido.		satisfactoriamente"	2. Se muestra un lista desplegable con las opciones Nuevo Comentario y Mis Comentarios. 3. Selecciona la opción Nuevo Comentario. 4. El sistema muestra la interfaz de insertar Nuevo Comentario.
EC 1.2 Existen campos obligatorios vacíos.	El sistema no inserta el comentario y muestra un mensaje.	I Vacío	I Vacío	V 2014-08-02	El sistema muestra el campo en color rojo y muestra un mensaje Rellene este campo si utiliza Firefox con el idioma en español.	
EC 1.3 Existen campos incorrectos.	El sistema no inserta el comentario y muestra un mensaje.	I internos.uc i.cu www.wordpress.com	I Un comentario es inválido si contiene menos de 60 palabras en su contenido.	I sfsfsfsafssf	Si el campo Url es inválido muestra el mensaje "Introduzca una URL" , si el campo comentario es incorrecto muestra el mensaje "Debe escribir como mínimo 60 palabras para que el comentario sea válido." y si el campo fecha es inválido se muestra el mensaje "Este valor no es válido"	

Tabla 9: Prueba de Funcionalidad para el CU Gestionar Comentario

Las celdas de la tabla contienen V, I, o N/A. V indica válido, I indica inválido, y N/A que no es necesario proporcionar un valor del dato en este caso, ya que es irrelevante.

Estas pruebas fueron realizadas a la aplicación web, arrojando como resultado un total de 99 no conformidades en 3 iteraciones distribuidas de la forma que se presenta en la **Figura 20**. Las no conformidades encontradas se agruparon fundamentalmente en cinco tipos: de correspondencia, de funcionalidad, de validación, de idioma y algunas recomendaciones. La **Figura 21** muestra el porcentaje que representaron las cinco clasificaciones de no conformidades con respecto al total.

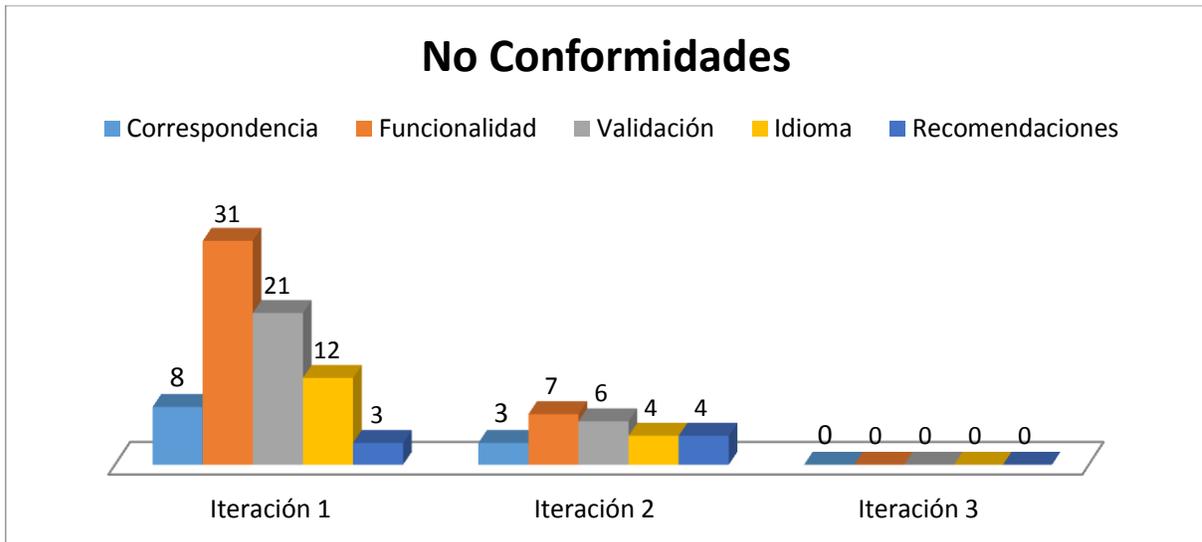


Figura 20: No conformidades encontradas en la aplicación web distribuidas en 3 iteraciones

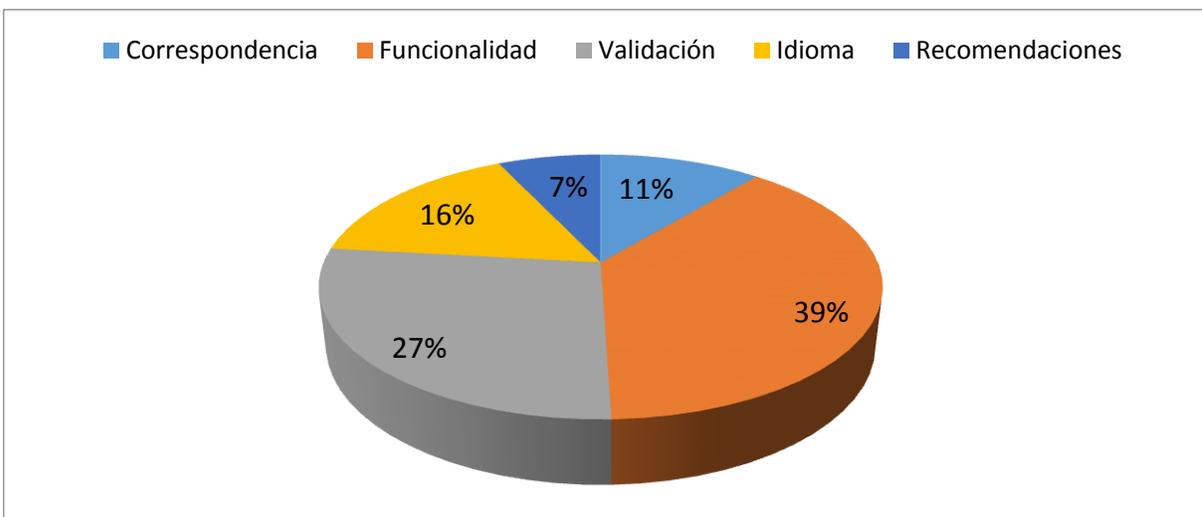


Figura 21: Porcentaje de las No Conformidades encontradas por clasificación

3.5.3. Pruebas de seguridad

Las pruebas de seguridad se realizaron para buscar vulnerabilidades en el ambiente del lado del cliente, en las comunicaciones al pasar datos del cliente al servidor y de vuelta, y el ambiente del servidor. Para realizar las pruebas de seguridad a la aplicación web infoBlog v2, se utilizó la herramienta *Acunetix Web Vulnerability Scanner 8*, la cual realiza un análisis a la aplicación buscando vulnerabilidades. Esta herramienta hace pruebas contra ataques como: Secuencias de comandos en sitios cruzados (XSS por sus

siglas en inglés *Cross-site scripting*) e Inyecciones SQL. Como resultados de estas pruebas realizadas, se obtuvieron deficiencias en 2 indicadores de impacto medio en la seguridad de la aplicación web. Luego de corregir estas deficiencias encontradas, la aplicación fue sometida nuevamente a un escaneo de vulnerabilidades para garantizar que habían sido resueltas, arrojando un resultado positivo.

3.6. Conclusiones parciales

Luego de lo analizado en este capítulo se concluye que:

- El desglose de la aplicación en componentes, mejoró la comprensión de cómo está integrada y las diferentes dependencias que presenta.
- La definición de los estándares de codificación a utilizar permitió lograr una mayor legibilidad, organización y limpieza al código que se implementó.
- Como resultado de la implementación se obtuvo una aplicación web completamente funcional.
- Las interfaces de la aplicación cumplen con los requisitos definidos por el cliente y brindan la posibilidad de gestionar la información con facilidad.
- La ejecución de las pruebas funcionales, de seguridad, las unitarias y las de integración, permitieron la detección de errores de diferentes tipos, que luego fueron solucionados, esto trajo como resultado un mejor funcionamiento del sistema.
- La realización de las pruebas al software desarrollado permitió validar la solución desarrollada y comprobar que cumple con los requerimientos definidos.

CONCLUSIONES GENERALES

- Con la realización de este trabajo se logró realizar un sistema que permite la gestión de la información en la línea Weblog del centro CIDI, dándole cumplimiento al objetivo general planteado, así como a los específicos, de la presente investigación.
- Mediante el estudio de diversos conceptos asociados al proceso de gestión de la información, se logró tener un mejor entendimiento de cómo funciona el mismo dentro de la línea Weblog.
- Con el estudio del estado del arte se analizaron las características que presentaba la aplicación que está actualmente en funcionamiento en la línea Weblog haciendo énfasis en sus funcionalidades, para extraer de ella, los requerimientos que serían incorporados en la nueva solución.
- La selección de las herramientas, lenguajes, tecnologías y metodología permitió desarrollar la solución en el tiempo estimado y con las funcionalidades definidas por el cliente.
- La declaración de los componentes para la gestión de la información de la solución, permitió una mayor organización en el desarrollo de la misma y una mejor distribución en la elaboración de la seguridad.
- El análisis y diseño para la implementación de la aplicación web, tuvo su base en las necesidades del cliente ya que las interfaces de la aplicación cumplen con los requisitos definidos y brindan la posibilidad de gestionar la información con facilidad.
- La ejecución de las pruebas realizadas, permitieron validar el correcto funcionamiento de la aplicación web infoBlog v2 simulando ambientes reales en diferentes interfaces.

RECOMENDACIONES

Los objetivos del trabajo de diploma fueron alcanzados pero se recomienda:

- Implementar una interfaz gráfica para la asignación de permisos a los diferentes roles de la aplicación infoBlog v2, ya que actualmente dichos permisos se configuran directamente en el código.
- Añadir una funcionalidad para monitorizar lo que se realiza en la aplicación.
- Agregar una funcionalidad para la creación de otros tipos de gráficos y mostrar el tipo de gráfico según la necesidad del usuario.

REFERENCIAS BIBLIOGRÁFICAS

1. **John M. Ivancevich, Peter Lorenzi y Steven J. Skinner.** *Gestión: calidad y competitividad.* Madrid, España : McGraw-Hill/Interamericana de Espana, S.A, 1997. ISBN 8448108485, 9788448108489.
2. **Heinz, Harold Koontz y Weihrich.** *Administración, una prospectiva global.* México : Editorial McGraw-Hill, 1998.
3. **Darromán Savigne, Velázquez Leyva.** "El proceso de gestión y la gestión económica en las empresas". *Observatorio de la Economía Latinoamericana.* [En línea] 2011. [Citado el: 10 de diciembre de 2013.] <http://www.eumed.net/cursecon/ecolat/cu/2011/dsvl.htm>. ISSN 1696-8352.
4. **Camejo, Ivis Goñi.** ACIMED. *Algunas reflexiones sobre el concepto de información y sus implicaciones para el desarrollo de las ciencias de la información.* [En línea] 27 de enero de 2000. [Citado el: 10 de diciembre de 2013.] http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S1024-94352000000300005&lng=es&nrm=iso. ISSN 1024-9435.
5. **Chiavenato, Idalberto.** *Introducción a la Teoría General de la Administración.* s.l. : McGraw-Hill Interamerica, 2006.
6. **O.C Ferrell, Hirt Geoffrey.** *Introducción a los Negocios en un Mundo Cambiante.* s.l. : McGraw-Hill Interamericana, 2004.
7. **Michael Czinkota, Masaaki Kotabe.** *Administración de Mercadotecnia.* s.l. : International Thomson Editores, 2001.
8. **Dante, Gloria Ponjuán.** *Gestión de información: dimensiones e implementación para el éxito.* s.l. : Rosario: Nuevo Parhadigma, 2004. ISBN:987-96536-6-1.
9. **Quiroga, Lic. Lourdes Aja.** ACIMED. *Gestión de información, gestión del conocimiento y gestión de la calidad en las organizaciones.* [En línea] octubre de 2002. [Citado el: 17 de diciembre de 2013.] http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S1024-94352002000500004&lng=es&nrm=iso. ISSN 1024-9435.
10. *Algunos aspectos teórico-conceptuales sobre el análisis documental y el análisis de información.* **Valdés, Marlery Sánchez Díaz y Juan Carlos Vega.** no. 2, La Habana, Cuba : s.n., 2003, Vol. XXXIV.

11. **Pereira, Jorge E.** GestioPolis. *Aplicaciones Web y ASPs*. [En línea] diciembre de 2006. [Citado el: 29 de enero de 2014.] <http://www.gestiopolis.com/canales6/mkt/mercadeopuntocom/aplicaciones-web-y-asps-relacion-cliente-servidor.htm>.
12. **Mora, Sergio Luján.** *Programación de aplicaciones web: historia, principios básicos y clientes web*. Alicante : Editorial Club Universitario, 2002. ISBN 978-84-8454-206-3.
13. **Asensio, Rafael Menéndez-Barzanallana.** Rafael Barzanallana. Universidad de Murcia. *Informática, Ciencia, Escepticismo*. [En línea] 10 de agosto de 2012. [Citado el: 29 de enero de 2014.] <http://www.um.es/docencia/barzana/DIVULGACION/INFORMATICA/Historia-desarrollo-aplicaciones-web.html>.
14. **Vallejo, Cesar.** Observatorio Tecnológico. *Gestión de la Información/Observatorio Tecnológico*. [En línea] 7 de septiembre de 2011. [Citado el: 10 de marzo de 2014.] <http://recursostic.educacion.es/observatorio/web/es/software/software-general/1009-gestion-de-la-informacion>.
15. **Hewlett-Packard Development Company, L. P.** HP Service Anywhere product / HP España. *IT/Change Managment/ HP Mexico*. [En línea] 2014. [Citado el: 10 de marzo de 2014.] <http://www8.hp.com/es/es/software-solutions/software.html?compURI=1363133>.
16. *Aplicación web para la realización de estudios farmacocinéticos, versión 2.0.* **Saavedra López, Dismey y Armentero Moreno, Yuniesky y Morales S Tabares, Zoila Esther.** no. 2, Ciudad de la Habana : s.n., 2013, Vol. 5. ISSN 1684-1859.
17. *Sistema automatizado para la gestión del programa de control sanitario internacional de Cuba.* **FERNANDEZ MARIN, Miguel Ángel y GONZALEZ TOLMO, Débora y VALDES DIAZ, Annia.** no. 1, Ciudad de La Habana : s.n., 2012, Vol. 4. ISSN 1684-1859.
18. **Eclipse.** OpenUP. *Oficial Site*. [En línea] 2008. [Citado el: 10 de diciembre de 2013.] <http://epf.eclipse.org/wikis/openup/>.
19. **Alvarez, Miguel Angel.** Desarrolloweb.com. *Que es PHP*. [En línea] 9 de mayo de 2001. [Citado el: 15 de diciembre de 2013.] <http://www.desarrolloweb.com/articulos/392.php>.

20. Castellano, Programación en. *¿Por qué elegir PHP?* [En línea] 2011. [Citado el: 15 de diciembre de 2013.] http://www.programacion.com/articulo/%20por_que_elegir_php_143.
21. **Alvarez, Miguel Angel.** Desarrolloweb.com. *Qué es Javascript.* [En línea] 1 de enero de 2001. [Citado el: 15 de diciembre de 2013.] <http://www.desarrolloweb.com/articulos/25.php>.
22. —. Desarrolloweb.com. *Qué es HTML 5.* [En línea] 14 de octubre de 2009. [Citado el: 15 de diciembre de 2013.] <http://www.desarrolloweb.com/articulos/que-es-html5.html>.
23. **Herrera, Daniel Ernesto Navarro.** LaWebera.es. *¿Cómo darle un uso apropiado a CSS3?* [En línea] 31 de octubre de 2011. [Citado el: 19 de diciembre de 2013.] <http://www.lawebera.es/maquetacion-web/darle-uso-apropiado-css3.php#>.
24. Portal en español sobre PostgreSQL. *PostgreSQL-es.* [En línea] 2 de octubre de 2010. [Citado el: 19 de diciembre de 2013.] http://www.postgresql.org.es/sobre_postgresql.
25. **Denzer, Patricio.** *PostgreSQL.* 2002.
26. PostgreSQL Wiki. *Guía de la Comunidad para las herramientas GUI de PostgreSQL.* [En línea] 18 de mayo de 2012. [Citado el: 16 de diciembre de 2013.] http://wiki.postgresql.org/wiki/Gu%C3%ADa_de_la_Comunidad_para_las_herramientas_GUI_de_PostgreSQL#pgAdmin_III.
27. Ecured. *Enciclopedia Colaborativa Cubana.* [En línea] 14 de diciembre de 2010. [Citado el: 16 de diciembre de 2013.] <http://www.ecured.cu/index.php/Framework>.
28. **Potencier, Fabien.** Symfony. *Oficial Site.* [En línea] [Citado el: 16 de diciembre de 2013.] <http://symfony.com/>.
29. Ubuntu. *Community website.* [En línea] 2010. [Citado el: 16 de diciembre de 2013.] <http://ubuntu.org.ar/?q=node/2>.
30. **Rojas, Cali.** LEWEBMONSTER.COM. *Linux para desarrollo web.* [En línea] 17 de junio de 2012. [Citado el: 15 de enero de 2014.] <http://www.lewebmonster.com/linux-para-desarrollo-web/>.
31. Ecured. *Enciclopedia Colaborativa Cubana.* [En línea] 14 de diciembre de 2010. [Citado el: 17 de enero de 2014.] http://www.ecured.cu/index.php/IDE_de_Programaci%C3%B3n.

32. Editum.org. *¿Qué Es Un Servidor Web?* [En línea] 27 de noviembre de 2007. [Citado el: 18 de enero de 2014.] <http://www.editum.org/Que-Es-Un-Servidor-Web-p-401.html>.
33. **The Apache Software Foundation.** Apache. [En línea] [Citado el: 18 de enero de 2014.] <http://httpd.apache.org/>.
34. Visual Paradigm. *Oficial Site.* [En línea] [Citado el: 18 de enero de 2014.] <http://www.visual-paradigm.com/product/vpuml>.
35. **Morgado, Jose Maria Acuña.** Reeditor.com. *Librerías javascript de gran utilidad para tu desarrollo web.* [En línea] 9 de octubre de 2013. [Citado el: 13 de enero de 2014.] <http://www.reeditor.com/columna/11084/17/informatica/librerias/javascript/gran/utilidad/tu/desarrollo/web>.
36. **GitHub.** GitHub. *Build software better, together.* [En línea] [Citado el: 27 de enero de 2014.] <https://github.com/DataTables/DataTables>.
37. **SpryMedia.** DataTables. [En línea] 2008. [Citado el: 27 de enero de 2014.] <http://www.datatables.net/>.
38. WEB APPERS. *Best Open Source Web Resources.* [En línea] [Citado el: 27 de enero de 2014.] <http://www.webappers.com/2009/03/09/tinytable-easy-to-use-javascript-table-sorter/#KzGjiYJDrFhvgTSz.99>.
39. **Fontela, Alvaro.** OpenWebCMS. *Gestores de contenido.* [En línea] 21 de mayo de 2013. [Citado el: 28 de enero de 2014.] <http://openwebcms.es/2013/que-es-bootstrap/>.
40. GENBETA: dev. *desarrollo y software.* [En línea] 16 de junio de 2012. [Citado el: 28 de enero de 2014.] <http://www.genbetadev.com/frameworks/bootstrap>.
41. **Suárez, Pablo Javier.** Kabytes. *4 Excelentes plugins para crear gráficas con jquery.* [En línea] 18 de octubre de 2010. [Citado el: 30 de enero de 2014.] <http://www.kabytes.com/programacion/4-excelentes-plugins-para-crear-graficas-con-jquery/>.
42. jQuery Flot Tutorial. [En línea] 2013. [Citado el: 30 de enero de 2014.] <http://www.jqueryflottutorial.com/what-is-jquery-flot.html>.
43. **jqPlot.** jqPlot. *pure javascript plotting.* [En línea] [Citado el: 29 de enero de 2014.] <http://www.jqplot.com/>.
44. **Bitbucket.** Bitbucket. [En línea] 29 de abril de 2013. [Citado el: 29 de enero de 2014.] <https://bitbucket.org/cleonello/jqplot/wiki/Home>.

45. **Developer Express Inc.** HTML5 JavaScript dynamic client side data visualization. *ChartJS*. [En línea] 2013. [Citado el: 11 de marzo de 2014.] <http://chartjs.devexpress.com/?gclid=CJWFsuGqlr4CFUpp7Aod7j0ADw>.
46. **Larman, Craig.** *UML y Patrones. 2da Edición.* s.l. : Prentice Hall, 2003. ISBN: 84-205-3438-2.
47. **Sommerville, Ian.** *Ingeniería del Software. Séptima edición.* Madrid : PEARSON EDUCACION S.A, 2005. ISBN 84-7829-074-5.
48. **PRESSMAN, Roger. S.** *Ingeniería de software.* España : McGraw-Hill Interamericana, 2010. ISBN 9786071503145.
49. **S. Pressman, Roger.** *Ingeniería del Software, un enfoque práctico.* McGraw-Hill : s.n., 2001.
50. **Tedeschi, Nicolás.** Microsoft. *Developer Network*. [En línea] [Citado el: 18 de marzo de 2014.] <http://msdn.microsoft.com/es-es/library/bb972240.aspx>.
51. **Eguiluz, Javier.** *Desarrollo web ágil con Symfony2.* Madrid : s.n., 2013.
52. **Fabien Potencier, François Zaninotto.** *Symfony la guía definitiva 1.2.* 2008.
53. **Godoy Jiménez, José Manuel.** Diseño de proyectos de software en código abierto. [En línea] 2002. [Citado el: 10 de marzo de 2014.] <http://www.ibiblio.org/pub/linux/docs/LuCaS/Tutoriales/doc-dise%F1o-software/doc-dise%F1o-software-parte-1.pdf>.
54. **Salinas Caro, Patricio.** Tutorial de UML. [En línea] [Citado el: 15 de marzo de 2014.] <http://www.dcc.uchile.cl/~psalinas/uml/>.
55. **Ltd, Sparx Systems Pty.** Sparx Systems- UML 2 Tutorial - Component Diagram. [En línea] 23 de abril de 2014. [Citado el: 28 de abril de 2014.] http://www.sparxsystems.com.au/resources/uml2_tutorial/uml2_componentdiagram.html.
56. **Pressman, Roger S.** *Ingeniería del Software. Un enfoque práctico. Sexta Edición.* México : MCGRAW-HILL / INTERAMERICANA DE MEXICO, 2005. ISBN 9789701054734.

BIBLIOGRAFÍA

1. *Algunos aspectos teórico-conceptuales sobre el análisis documental y el análisis de información.* **Valdés, Marlery Sánchez Díaz y Juan Carlos Vega.** no. 2, La Habana, Cuba : s.n., 2003, Vol. XXXIV.
2. **Alvarez, Miguel Angel.** Desarrolloweb.com. *Que es PHP.* [En línea] 9 de mayo de 2001. [Citado el: 15 de diciembre de 2013.] <http://www.desarrolloweb.com/articulos/392.php>.
3. **Alvarez, Miguel Angel.** Desarrolloweb.com. *Qué es Javascript.* [En línea] 1 de enero de 2001. [Citado el: 15 de diciembre de 2013.] <http://www.desarrolloweb.com/articulos/25.php>.
4. *Aplicación web para la realización de estudios farmacocinéticos, versión 2.0.* **Saavedra López, Dismey y Armentero Moreno, Yuniesky y Morales S Tabares, Zoila Esther.** no. 2, Ciudad de la Habana : s.n., 2013, Vol. 5. ISSN 1684-1859.
5. **Asensio, Rafael Menéndez-Barzanallana.** Rafael Barzanallana. Universidad de Murcia. *Informática, Ciencia, Escepticismo.* [En línea] 10 de agosto de 2012. [Citado el: 29 de enero de 2014.] <http://www.um.es/docencia/barzana/DIVULGACION/INFORMATICA/Historia-desarrollo-aplicaciones-web.html>.
6. **Bitbucket.** Bitbucket. [En línea] 29 de abril de 2013. [Citado el: 29 de enero de 2014.] <https://bitbucket.org/cleonello/jqplot/wiki/Home>.
7. **Camejo, Ivis Goñi.** ACIMED. *Algunas reflexiones sobre el concepto de información y sus implicaciones para el desarrollo de las ciencias de la información.* [En línea] 27 de enero de 2000. [Citado el: 10 de diciembre de 2013.] http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S1024-94352000000300005&lng=es&nrm=iso.ISSN 1024-9435.
8. Castellano, Programación en. *¿Por qué elegir PHP?* [En línea] 2011. [Citado el: 15 de diciembre de 2013.] http://www.programacion.com/articulo/%20por_que_elegir_php_143.
9. **Chiavenato, Idalberto.** *Introducción a la Teoría General de la Administración.* s.l. : McGraw-Hill Interamerica, 2006.
10. **Dante, Gloria Ponjuán.** *Gestión de información: dimensiones e implementación para el éxito.* s.l. : Rosario: Nuevo Parhadigma, 2004. ISBN:987-96536-6-1.
11. **Darromán Savigne, Velázquez Leyva.** "El proceso de gestión y la gestión económica en las empresas". *Observatorio de la Economía Latinoamericana.* [En línea] 2011. [Citado el: 10 de diciembre de 2013.] <http://www.eumed.net/cursecon/ecolat/cu/2011/dsvl.htm>.ISSN 1696-8352.
12. **Denzer, Patricio.** *PostgreSQL.* 2002.
13. Desarrolloweb.com. *Qué es HTML 5.* [En línea] 14 de octubre de 2009. [Citado el: 15 de diciembre de 2013.] <http://www.desarrolloweb.com/articulos/que-es-html5.html>.

14. **Developer Express Inc.** HTML5 JavaScript dynamic client side data visualization. *ChartJS*. [En línea] 2013. [Citado el: 11 de marzo de 2014.] <http://chartjs.devexpress.com/?gclid=CJWFsuGqlr4CFUpp7Aod7i0ADw>.
15. **Eclipse.** OpenUP. *Oficial Site*. [En línea] 2008. [Citado el: 10 de diciembre de 2013.] <http://epf.eclipse.org/wikis/openup/>.
16. **Ecured.** *Enciclopedia Colaborativa Cubana*. [En línea] 14 de diciembre de 2010. [Citado el: 16 de diciembre de 2013.] <http://www.ecured.cu/index.php/Framework>.
17. **Ecured.** *Enciclopedia Colaborativa Cubana*. [En línea] 14 de diciembre de 2010. [Citado el: 19 de diciembre de 2013.] http://www.ecured.cu/index.php/Gesti%C3%B3n_de_la_Informaci%C3%B3n.
18. **Ecured.** *Enciclopedia Colaborativa Cubana*. [En línea] 14 de diciembre de 2010. [Citado el: 17 de enero de 2014.] http://www.ecured.cu/index.php/IDE_de_Programaci%C3%B3n.
19. **Editum.org.** *¿Qué Es Un Servidor Web?* [En línea] 27 de noviembre de 2007. [Citado el: 18 de enero de 2014.] <http://www.editum.org/Que-Es-Un-Servidor-Web-p-401.html>.
20. **Eguiluz, Javier.** *Desarrollo web ágil con Symfony2*. Madrid : s.n., 2013.
21. **Fabien Potencier, François Zaninotto.** *Symfony la guía definitiva 1.2*. 2008.
22. **Fontela, Alvaro.** OpenWebCMS. *Gestores de contenido*. [En línea] 21 de mayo de 2013. [Citado el: 28 de enero de 2014.] <http://openwebcms.es/2013/que-es-bootstrap/>.
23. GENBETA: dev. *desarrollo y software*. [En línea] 16 de junio de 2012. [Citado el: 28 de enero de 2014.] <http://www.genbetadev.com/frameworks/bootstrap>.
24. **GitHub.** GitHub. *Build software better, together*. [En línea] [Citado el: 27 de enero de 2014.] <https://github.com/DataTables/DataTables>.
25. **Godoy Jiménez, José Manuel.** Diseño de proyectos de software en código abierto. [En línea] 2002. [Citado el: 10 de marzo de 2014.] <http://www.ibiblio.org/pub/linux/docs/LuCaS/Tutoriales/doc-dise%F1o-software/doc-dise%F1o-software-parte-1.pdf>.
26. **Heinz, Harold Koontz y Weihrich.** *Administración, una perspectiva global*. México : Editorial McGraw-Hill, 1998.
27. **Herrera, Daniel Ernesto Navarro.** LaWebera.es. *¿Cómo darle un uso apropiado a CSS3?* [En línea] 31 de octubre de 2011. [Citado el: 19 de diciembre de 2013.] <http://www.lawebera.es/maquetacion-web/darle-uso-apropiado-css3.php#>.
28. **Hewlett-Packard Development Company, L. P.** HP Service Anywhere product / HP España. *ITII Change Managment/ HP Mexico*. [En línea] 2014. [Citado el: 10 de marzo de 2014.] <http://www8.hp.com/es/es/software-solutions/software.html?compURI=1363133>.
29. **Hogan, Brian P.** *HTML5 and CSS3. Develop with Tomorrow's Standards Today*. Texas : Pragmatic Programmers, LLC, 2010. ISBN-10: 1-934356-68-9.
30. **jqPlot.** jqPlot. *pure javascript plotting*. [En línea] [Citado el: 29 de enero de 2014.] <http://www.jqplot.com/>.

31. jQuery Flot Tutorial. [En línea] 2013. [Citado el: 30 de enero de 2014.] <http://www.jqueryflottutorial.com/what-is-jquery-flot.html>.
32. **John M. Ivancevich, Peter Lorenzi y Steven J. Skinner.** *Gestión: calidad y competitividad*. Madrid, España : McGraw-Hill/Interamericana de España, S.A, 1997. ISBN 8448108485, 9788448108489.
33. **Larman, Craig.** *UML y Patrones. 2da Edición*. s.l. : Prentice Hall, 2003. ISBN: 84-205-3438-2.
34. **Ltd, Sparx Systems Pty.** Sparx Systems- UML 2 Tutorial - Component Diagram. [En línea] 23 de abril de 2014. [Citado el: 28 de abril de 2014.] http://www.sparxsystems.com.au/resources/uml2_tutorial/uml2_componentdiagram.html.
35. **Michael Czinkota, Masaaki Kotabe.** *Administración de Mercadotecnia*. s.l. : International Thomson Editores, 2001.
36. **Mora, Sergio Luján.** *Programación de aplicaciones web: historia, principios básicos y clientes web*. Alicante : Editorial Club Universitario, 2002. ISBN 978-84-8454-206-3.
37. **Morgado, Jose Maria Acuña.** Reeditor.com. *Librerías javascript de gran utilidad para tu desarrollo web*. [En línea] 9 de octubre de 2013. [Citado el: 13 de enero de 2014.] <http://www.reeditor.com/columna/11084/17/informatica/librerias/javascript/gran/utilidad/tu/desarrollo/web>.
38. **O.C Ferrell, Hirt Geoffrey.** *Introducción a los Negocios en un Mundo Cambiante*. s.l. : McGraw-Hill Interamericana, 2004.
39. **Pereira, Jorge E.** GestioPolis. *Aplicaciones Web y ASPs*. [En línea] diciembre de 2006. [Citado el: 29 de enero de 2014.] <http://www.gestiopolis.com/canales6/mkt/mercadeopuntocom/aplicaciones-web-y-asps-relacion-cliente-servidor.htm>.
40. Portal en español sobre PostgreSQL. *PostgreSQL-es*. [En línea] 2 de octubre de 2010. [Citado el: 19 de diciembre de 2013.] http://www.postgresql.org.es/sobre_postgresql.
41. PostgreSQL Wiki. *Guía de la Comunidad para las herramientas GUI de PostgreSQL*. [En línea] 18 de mayo de 2012. [Citado el: 16 de diciembre de 2013.] http://wiki.postgresql.org/wiki/Gu%C3%ADa_de_la_Comunidad_para_las_herramientas_GUI_de_PostgreSQL#pg_Admin_III.
42. **Potencier, Fabien.** Symfony. *Oficial Site*. [En línea] [Citado el: 16 de diciembre de 2013.] <http://symfony.com/>.
43. **Pressman, Roger S.** *Ingeniería del Software. Un enfoque práctico. Sexta Edición*. México : MCGRAW-HILL / INTERAMERICANA DE MEXICO, 2005. ISBN 9789701054734.
44. **PRESSMAN, Roger. S.** *Ingeniería de software*. España : McGraw-Hill Interamericana, 2010. ISBN 9786071503145.
45. **Quiroga, Lic. Lourdes Aja.** ACIMED. *Gestión de información, gestión del conocimiento y gestión de la calidad en las organizaciones*. [En línea] octubre de 2002. [Citado el: 17 de diciembre de 2013.]

- http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S1024-94352002000500004&lng=es&nrm=iso. ISSN 1024-9435
46. **Rojas, Cali**. LEWEBMOSTER.COM. *Linux para desarrollo web*. [En línea] 17 de junio de 2012. [Citado el: 15 de enero de 2014.] <http://www.lewebmonster.com/linux-para-desarrollo-web/>.
 47. **Saavedra, Gustavo Rebolledo**. Gestión, Calidad y Agregación de valor en información. [En línea] [Citado el: 5 de diciembre de 2013.] <http://b3.bibliotecologia.cl/ar-gestion.htm>.
 48. **Salinas Caro, Patricio**. Tutorial de UML. [En línea] [Citado el: 15 de marzo de 2014.] <http://www.dcc.uchile.cl/~psalinas/uml/>.
 49. *Sistema automatizado para la gestión del programa de control sanitario internacional de Cuba*. **FERNANDEZ MARIN, Miguel Ángel y GONZALEZ TOLMO, Débora y VALDES DIAZ, Annia**. no. 1, Ciudad de La Habana : s.n., 2012, Vol. 4. ISSN 1684-1859.
 50. **Sommerville, Ian**. *Ingeniería del Software. Séptima edición*. Madrid : PEARSON EDUCACION S.A, 2005. ISBN 84-7829-074-5.
 51. **SpryMedia**. DataTables. [En línea] 2008. [Citado el: 27 de enero de 2014.] <http://www.datatables.net/>.
 52. **Suárez, Pablo Javier**. Kabytes. *4 Excelentes plugins para crear gráficas con jquery*. [En línea] 18 de octubre de 2010. [Citado el: 30 de enero de 2014.] <http://www.kabytes.com/programacion/4-excelentes-plugins-para-crear-graficas-con-jquery/>.
 53. **Symfony**. *The Components Book for Symfony 2.3*. 2013.
 54. **S. Pressman, Roger**. *Ingeniería del Software, un enfoque práctico*. McGraw-Hill : s.n., 2001.
 55. **Tedeschi, Nicolás**. Microsoft. *Developer Network*. [En línea] [Citado el: 18 de marzo de 2014.] <http://msdn.microsoft.com/es-es/library/bb972240.aspx>.
 56. **The Apache Software Foundation**. Apache. [En línea] [Citado el: 18 de enero de 2014.] <http://httpd.apache.org/>.
 57. **Ubuntu**. *Community website*. [En línea] 2010. [Citado el: 16 de diciembre de 2013.] <http://ubuntu.org.ar/?q=node/2>.
 58. **Vallejo, Cesar**. Observatorio Tecnológico. *Gestión de la Información/Observatorio Tecnológico*. [En línea] 7 de septiembre de 2011. [Citado el: 10 de marzo de 2014.] <http://recursostic.educacion.es/observatorio/web/es/software/software-general/1009-gestion-de-la-informacion>.
 59. Visual Paradigm. *Oficial Site*. [En línea] [Citado el: 18 de enero de 2014.] <http://www.visual-paradigm.com/product/vpuml>.
 60. WEB APPERS. *Best Open Source Web Resources*. [En línea] [Citado el: 27 de enero de 2014.] <http://www.webappers.com/2009/03/09/tinytable-easy-to-use-javascript-table-sorter/#KzGjiYJDrFhvgTSz.99>.
 61. **Zazo, Juan Alberto Delgado**. *Porramática. Aplicación web para gestionar apuestas deportivas*. Madrid : Escuela Politécnica Superior de la Universidad, 2013.

ANEXOS

Anexo A

Especificación de Casos de Uso del Sistema (ECUS)

CU Gestionar Comentario

Objetivo	Gestionar contenidos de tipo Comentario
Actores	Usuario
Resumen	El CU se inicializa cuando el actor (en lo adelante usuario) quiere gestionar el contenido de tipo Comentario.
Complejidad	Media
Prioridad	Alta (Critico)
Precondiciones	El usuario está autenticado en el sistema. El usuario pulsa la opción "Comentarios" del menú principal.
Postcondiciones	Se realizan satisfactoriamente cualquier acción sobre el tipo de contenido Comentario.

Prototipo

Listado de Mis Comentarios

Fecha Inicial: Fecha Final: ← 5

1 →

Show entries Search:

Id	Hecho Por	Fecha	Url de la Noticia	Acciones
11	Nolberto Rojas Perez	03-04-2014	http://drupaleros.uci.cu/	
13	Nolberto Rojas Perez	07-04-2014	http://drupaleros.uci.cu/	

Showing 1 to 2 of 2 entries

◀ Previous Next ▶

2 3 4

Flujo de eventos

Flujo básico Gestionar Comentario

	Actor	Sistema
1.	Pulsa la opción "Comentarios" del menú principal	
2.		Muestra un menú desplegable con las opciones siguientes:

		<ul style="list-style-type: none"> • Nuevo Comentario. Ver Sección 1. • Mis Comentarios. Ver Sección 2.
3.	Selecciona una opción.	
4.		Muestra la vista referente a la opción que seleccionó.
5.		Termina CU
Sección 1: “Nuevo Comentario” (Prototipo - 1)		
Flujo básico Nuevo Comentario		
	Actor	Sistema
1.	Pulsa la opción “Nuevo Comentario”	Muestra una vista con un formulario con los siguientes campos: <ol style="list-style-type: none"> 1. Url de la noticia 2. Comentario 3. Fecha
2.	Introduce los datos en los campos.	
3.	Pulsa la acción “Crear”	
4.		Valida que los datos sean correctos
5.		Almacena la información, redirecciona a la vista de “Insertar” y muestra un mensaje “Se ha creado el contenido satisfactoriamente”
6.		Termina Sección 1
Flujos alternos		
4 Datos Incorrectos		
	Actor	Sistema
1.		Verifica que los datos son incorrectos.
2.		Selecciona los campos incorrectos y muestra un mensaje.
3.		Ir al paso 2.
Sección 2: “Mis Comentarios”		
Flujo básico Mis Comentarios		
	Actor	Sistema
1.	Pulsa la opción “Mis Comentarios”.	
2.		Muestra una vista el listado de todos los comentarios que ese usuario ha creado. Además permite hacer las acciones siguientes: <ol style="list-style-type: none"> 4. Insertar. Ver Sección 1. (Prototipo - 1) 5. Modificar. Ver Sección 3. (Prototipo - 3) 6. Ver Detalles. Ver Sección 4. (Prototipo - 2) 7. Eliminar. Ver Sección 5. (Prototipo - 4) También muestra un filtro de búsqueda con los campos: <ol style="list-style-type: none"> 1. Fecha Inicial 2. Fecha Final
3.	Selecciona una opción.	
4.		Termina Sección 2
Flujos alternos		
3 Filtrar la información.		
	Actor	Sistema

1.	Introduce los datos en los campos.	
2.	Pulsa la opción "Filtrar" (Prototipo - 5)	
3.		Hace una búsqueda según los datos introducidos y devuelve la información a la vista.
Sección 3: "Modificar Comentario" (Prototipo - 3)		
Flujo básico Modificar Comentario		
	Actor	Sistema
1.	Pulsa la opción "Editar".	
2.		Muestra una vista con un formulario y los datos del comentario que seleccionó en los siguientes campos: 3. Url de la noticia 4. Comentario 5. Fecha 6. Evaluación (aparece si es administrador)
3.	Modifica los datos.	
4.	Pulsa la opción "Actualizar"	
5.		Valida los datos entrados.
6.		Almacena la información, redirecciona a la vista de "Modificar" y muestra un mensaje "Se ha actualizado el contenido satisfactoriamente"
7.		Termina Sección 3
Flujos alternos		
4 Datos Incorrectos		
	Actor	Sistema
1.		Verifica que los datos son incorrectos.
2.		Selecciona los campos incorrectos y muestra un mensaje.
3.		Ir al paso 3.
Sección 4: "Ver Detalles" (Prototipo - 2)		
Flujo básico Ver Detalles		
	Actor	Sistema
1.	Pulsa la opción "Ver Detalles".	
2.		Muestra una vista con la información de ese comentario que seleccionó. Y la opción de "Modificar" Ver Sección 3.
3.		Termina Sección 4
Sección 5: "Eliminar Comentario" (Prototipo - 4)		
Flujo básico Eliminar Comentario		
	Actor	Sistema
1.	Pulsa la opción "Eliminar".	
2.		Muestra un cuadro de diálogo con el mensaje "Desea eliminar este comentario".
3.	Pulsa la opción "Aceptar"	
4.		Elimina el comentario y redirecciona a la vista de la lista de

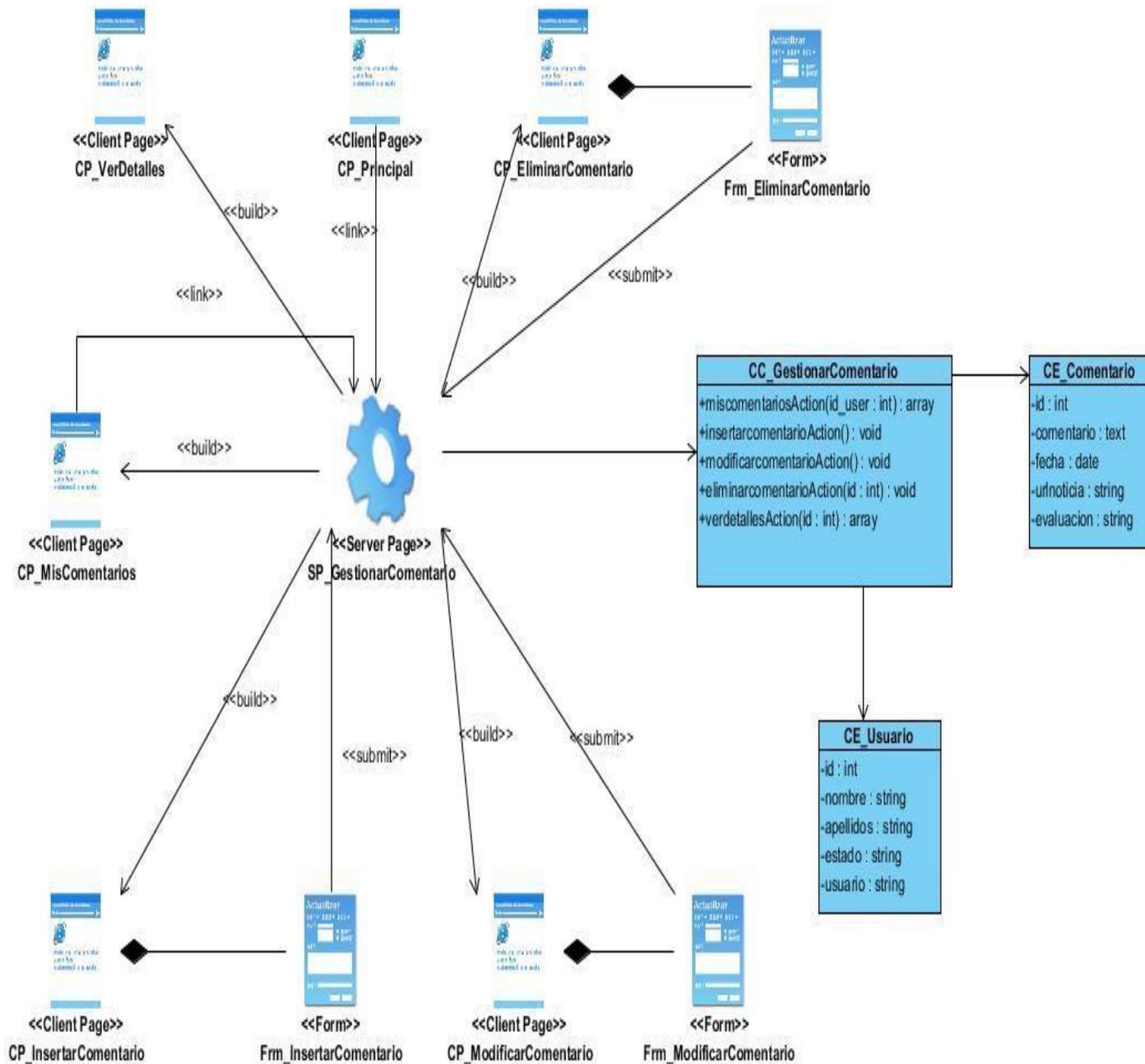
		comentarios en la que se encontraba y muestra un mensaje "Se ha eliminado el contenido satisfactoriamente"
5.		Termina Sección 5
Flujos alternos		
3 Pulsa opción "Cancelar"		
	Actor	Sistema
1.	Pulsa la opción "Cancelar"	
2.		Desaparece el cuadro de diálogo mostrado y redirecciona a la lista de comentarios en la que se encontraba.
Relaciones	CU Incluidos	
	CU Extendidos	
Requisitos no funcionales		
Asuntos pendientes		

Las demás descripciones de casos de uso se encuentran en el **Documento de Especificaciones de Casos de Uso del Sistema infoBlog v2**.

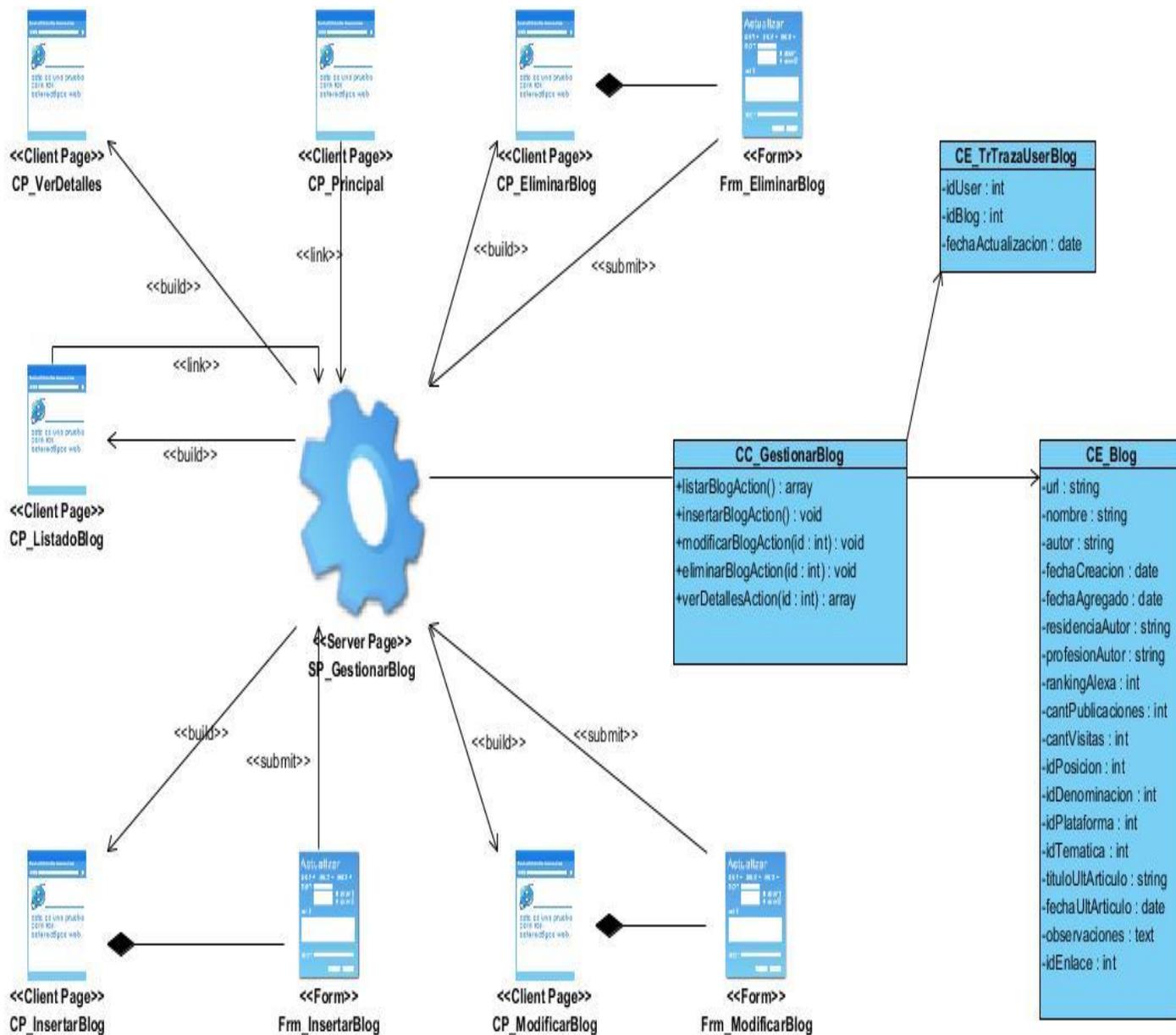
Anexo B

Diagramas de Clases de Diseño con Estereotipos Web (DCDEW)

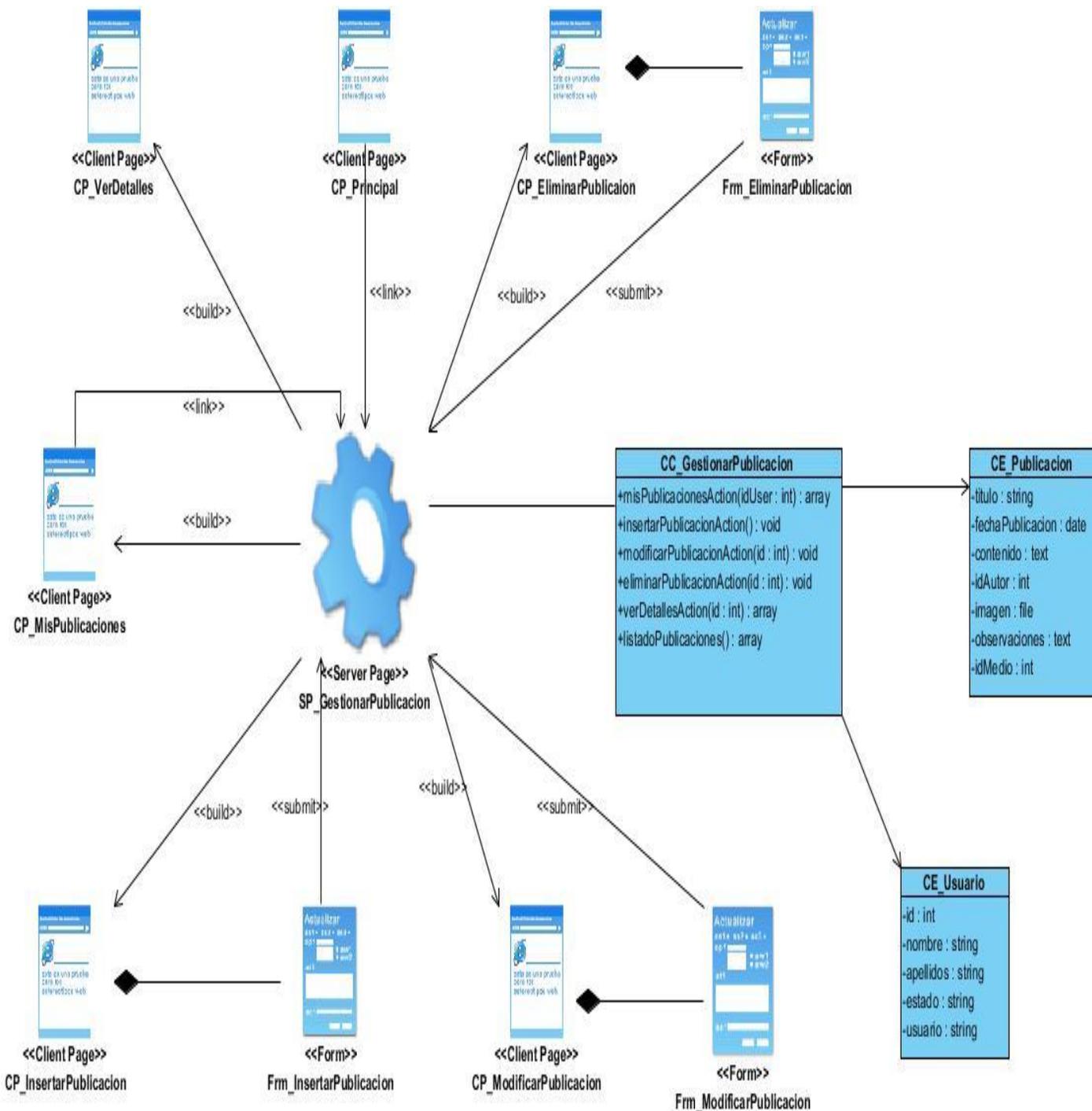
CU Gestionar Comentario



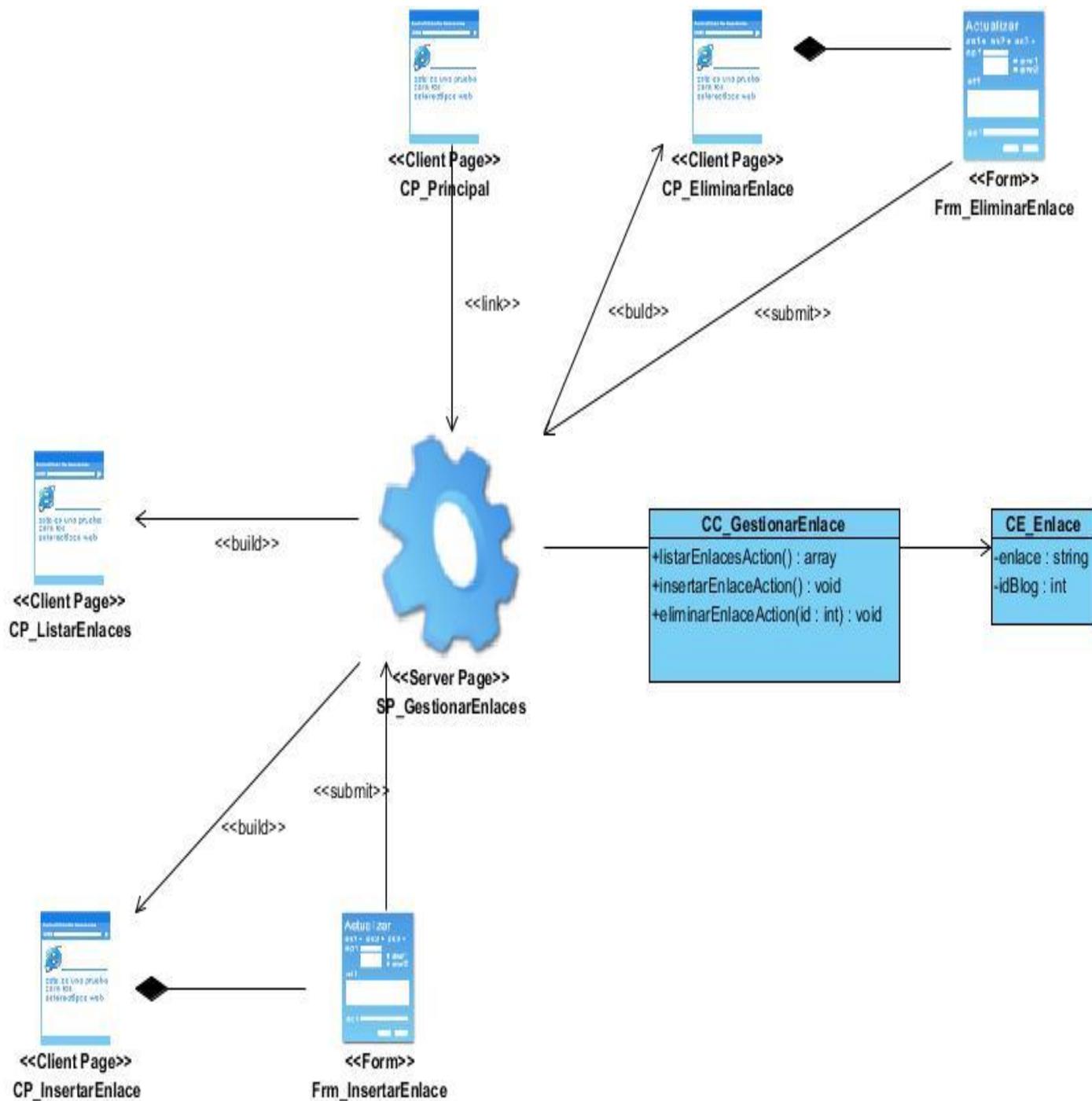
CU Gestionar Blog



CU Gestionar Publicación

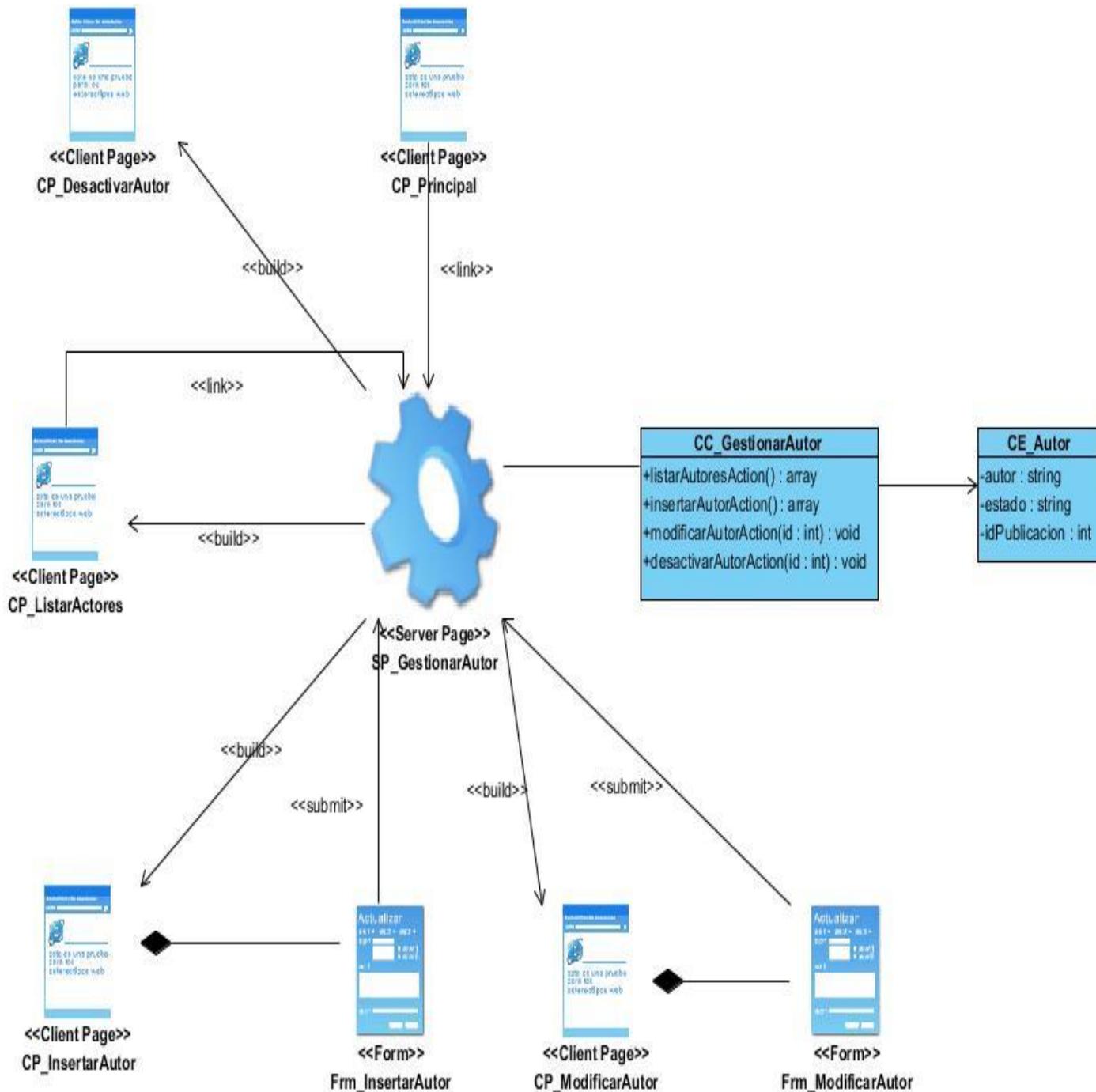


CU Gestionar Enlaces

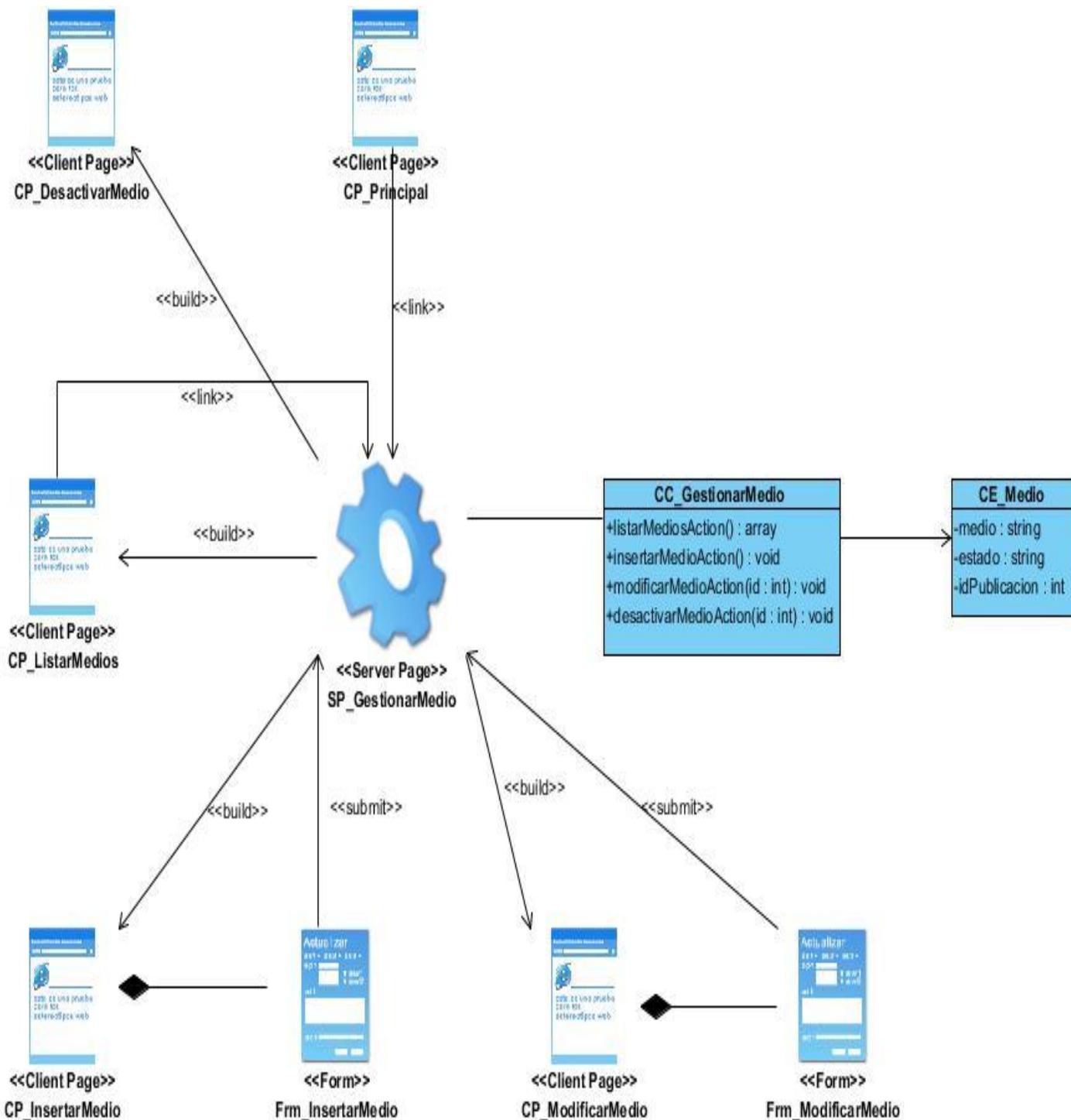


CU Gestionar Nomencladores

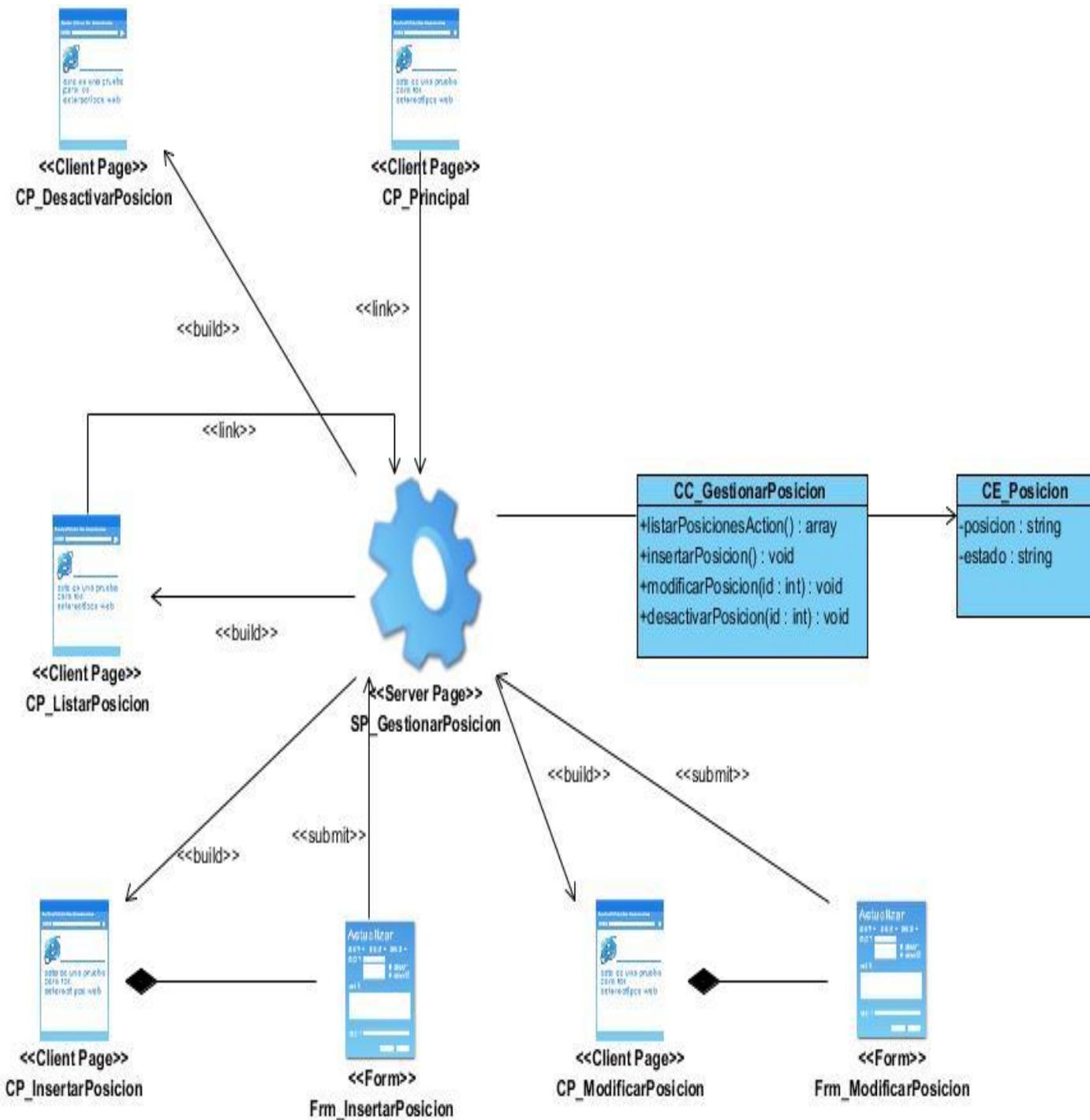
Sección Gestionar Nomenclador Autor



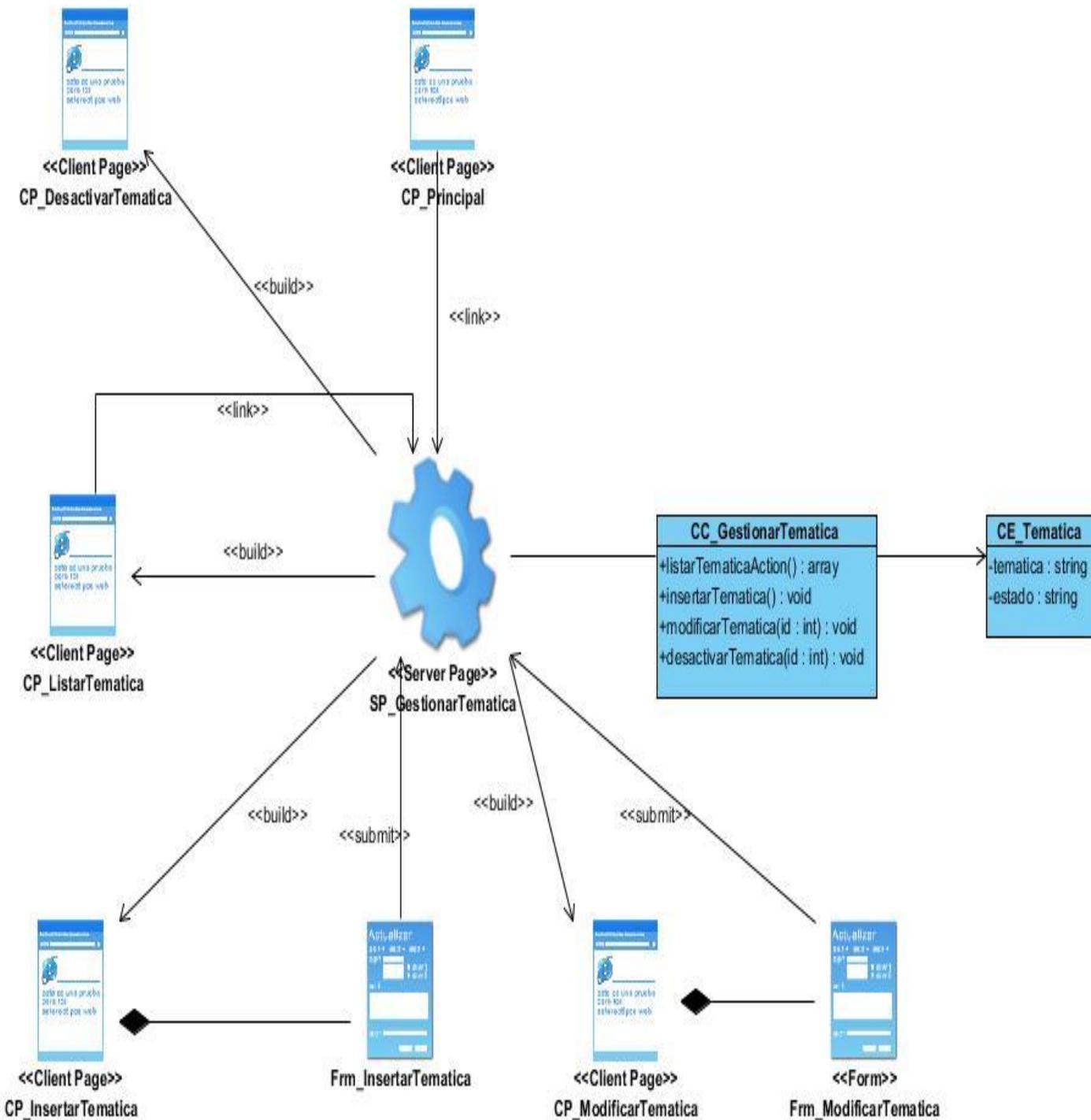
Sección Gestionar Nomenclador Medio



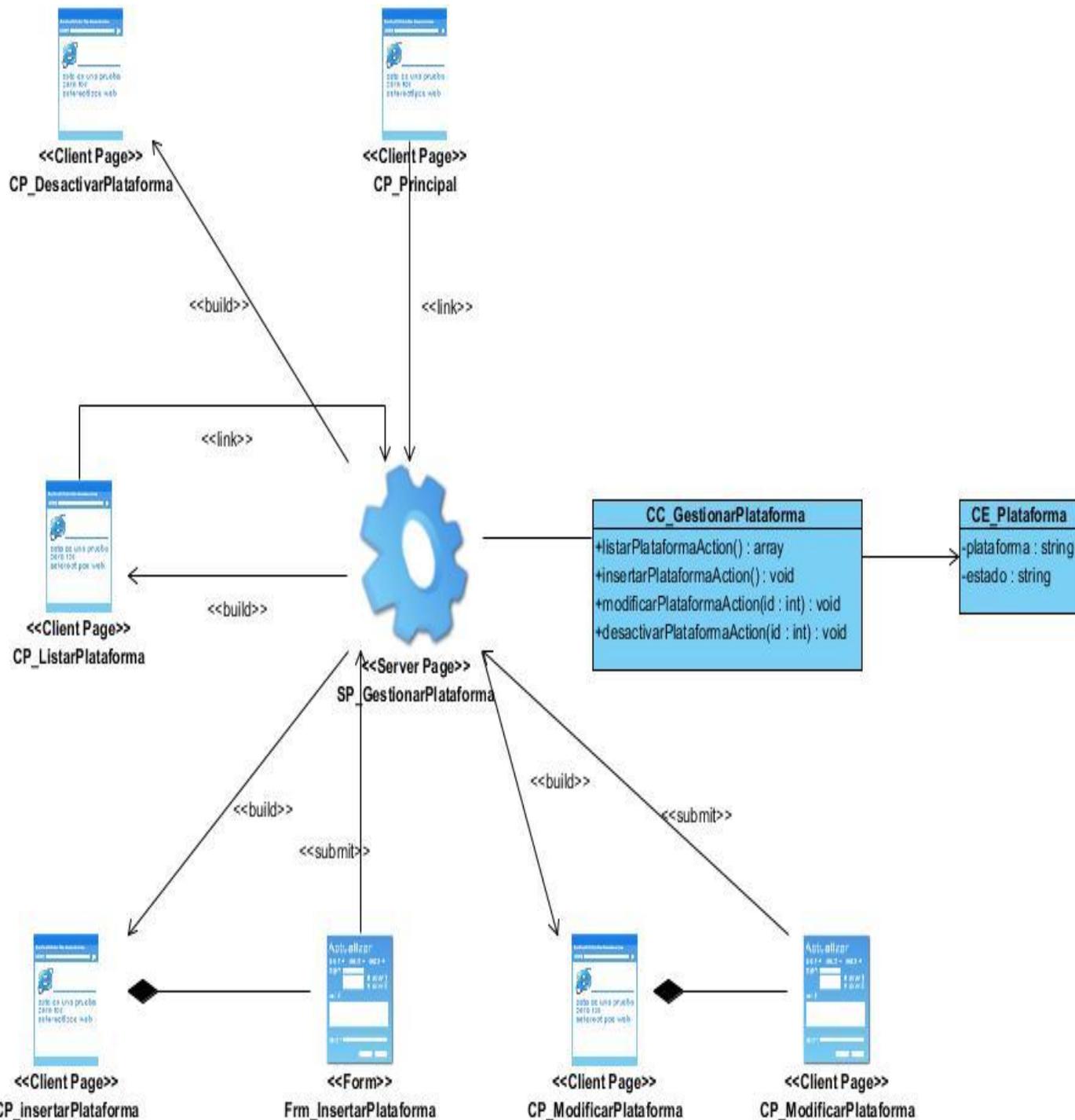
Sección Gestionar Nomenclador Posición



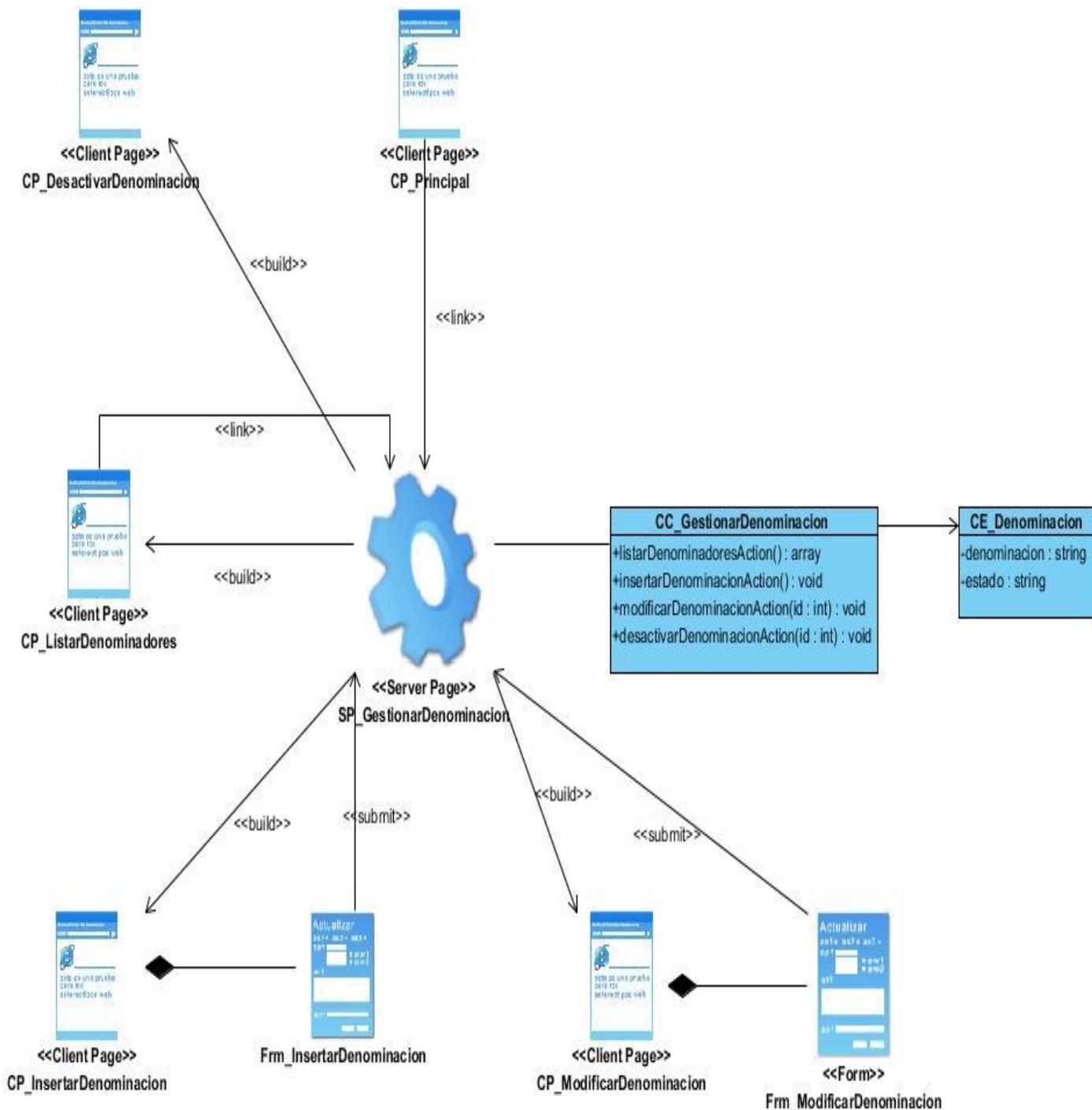
Sección Gestionar Nomenclador Temática



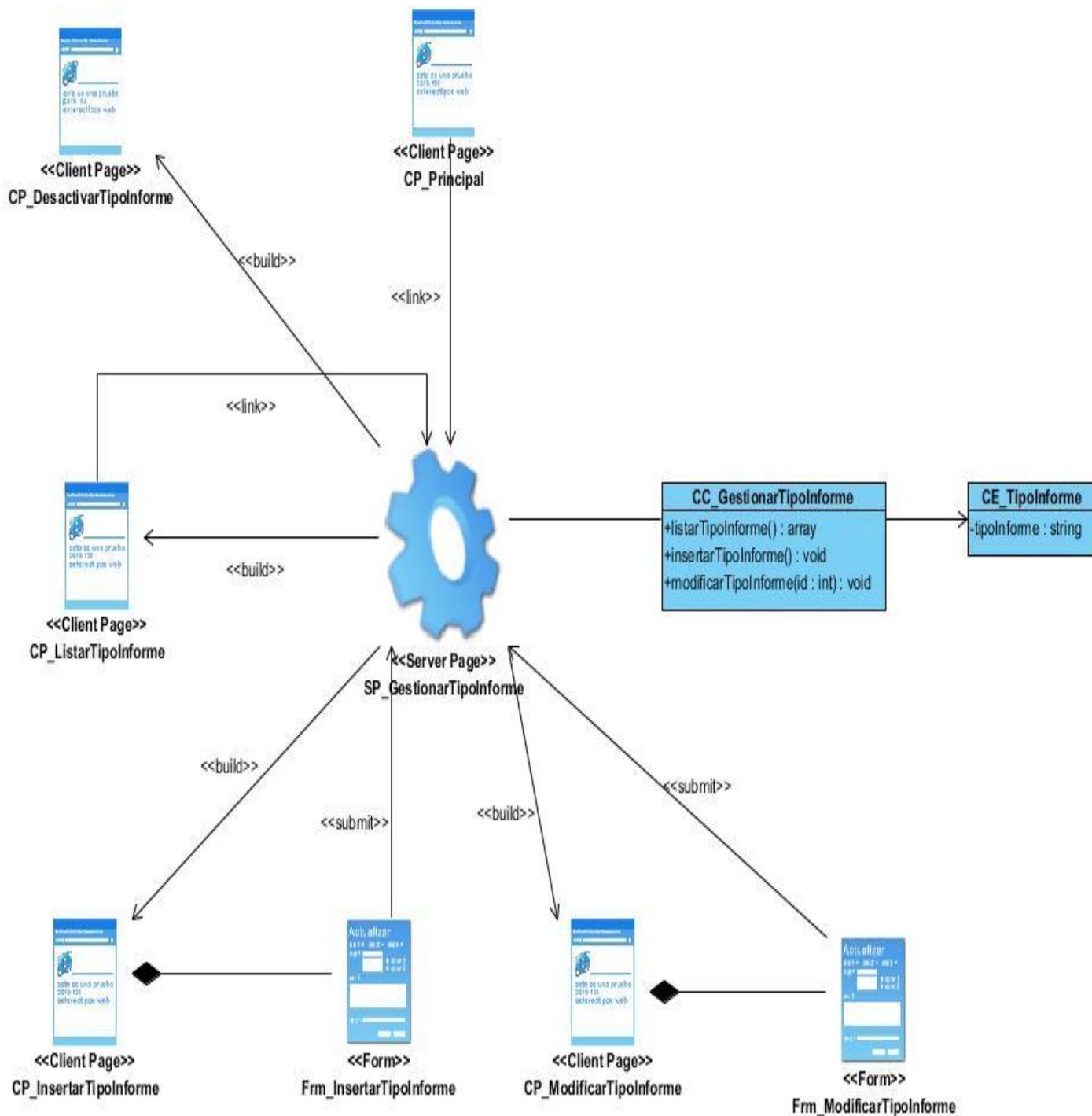
Sección Gestionar Nomenclador Plataforma



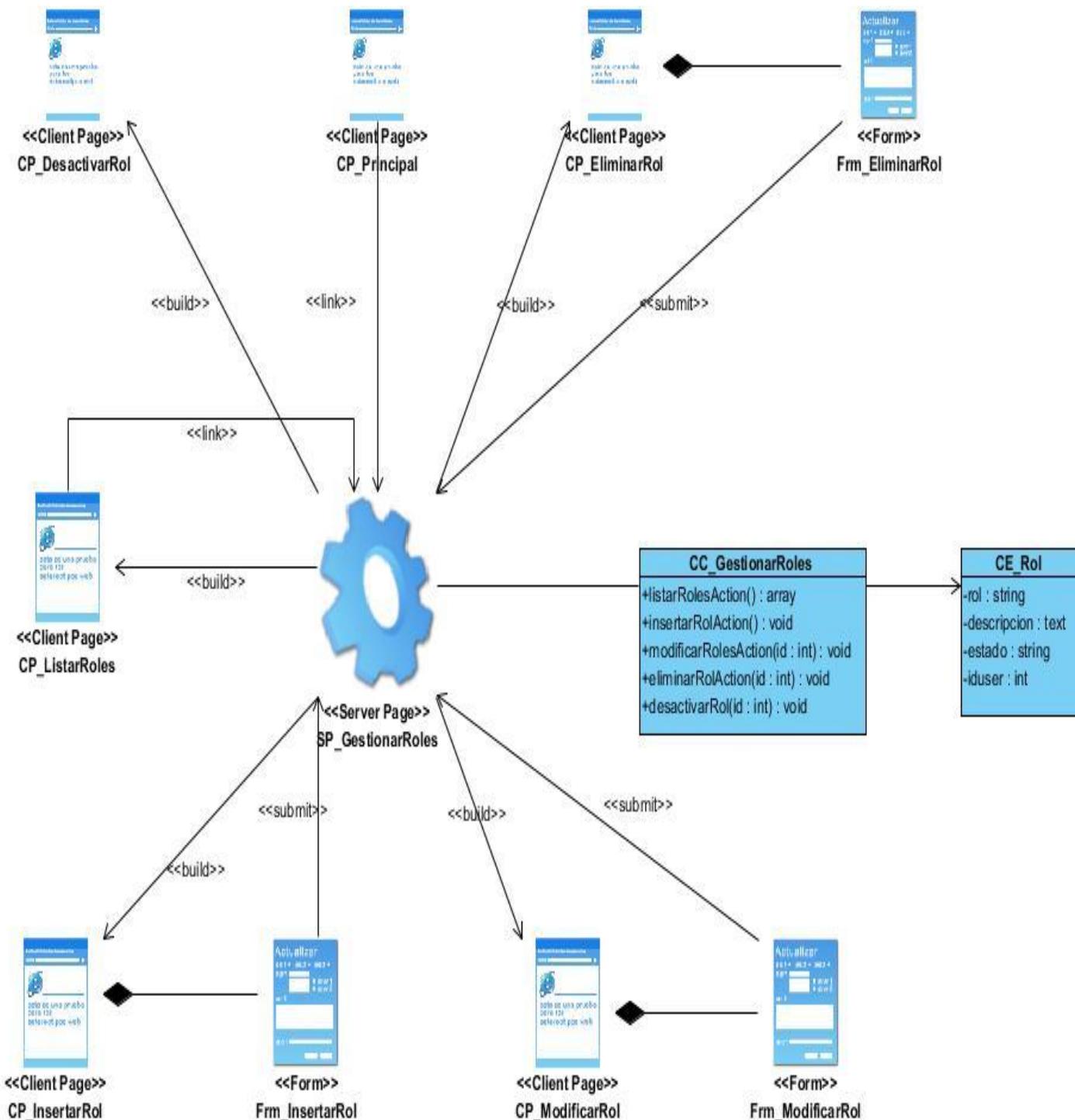
Sección Gestionar Nomenclador Denominación



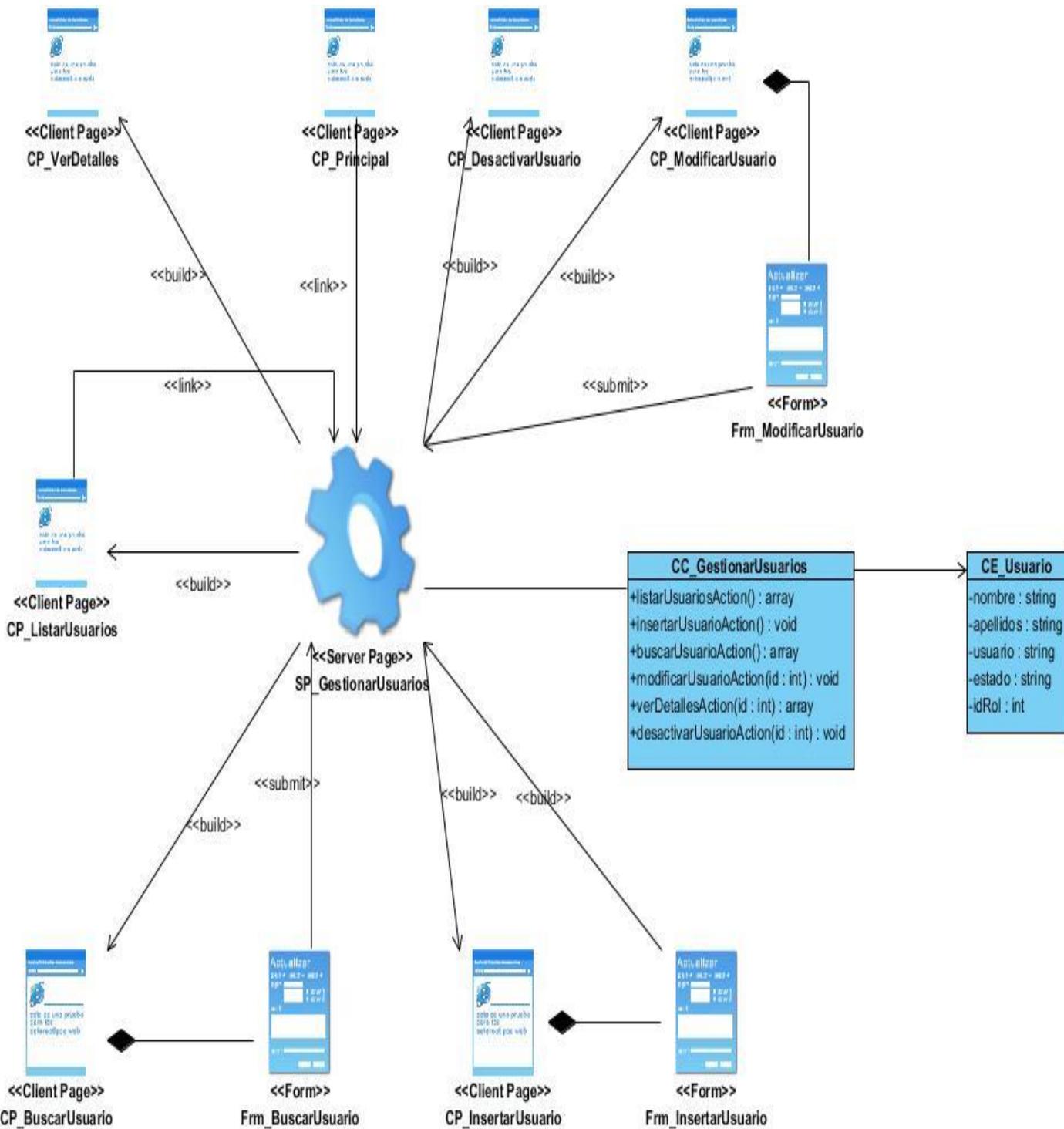
Sección Gestionar Nomenclador Tipo de Informe



CU Gestionar Rol

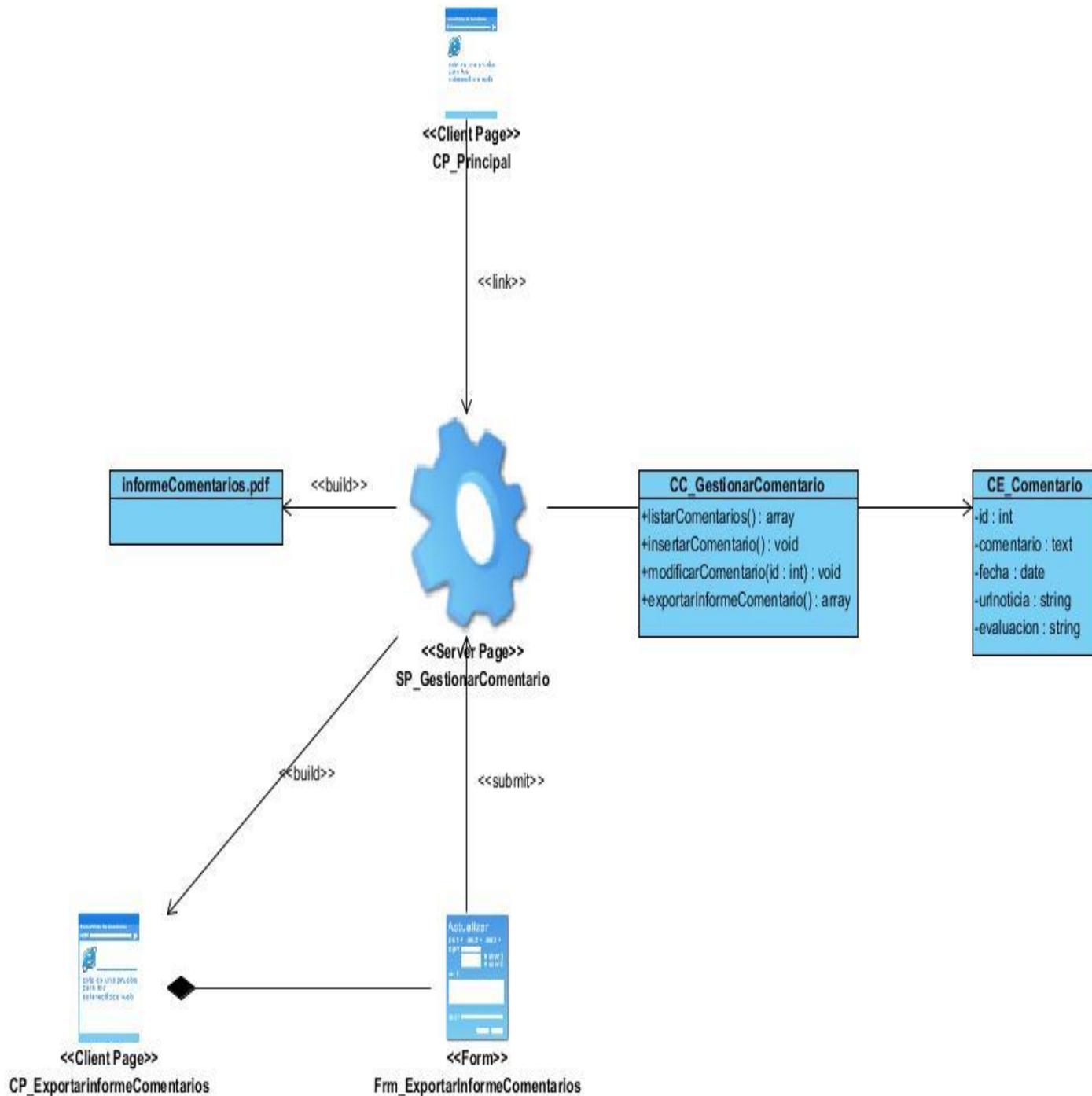


CU Gestionar Usuarios

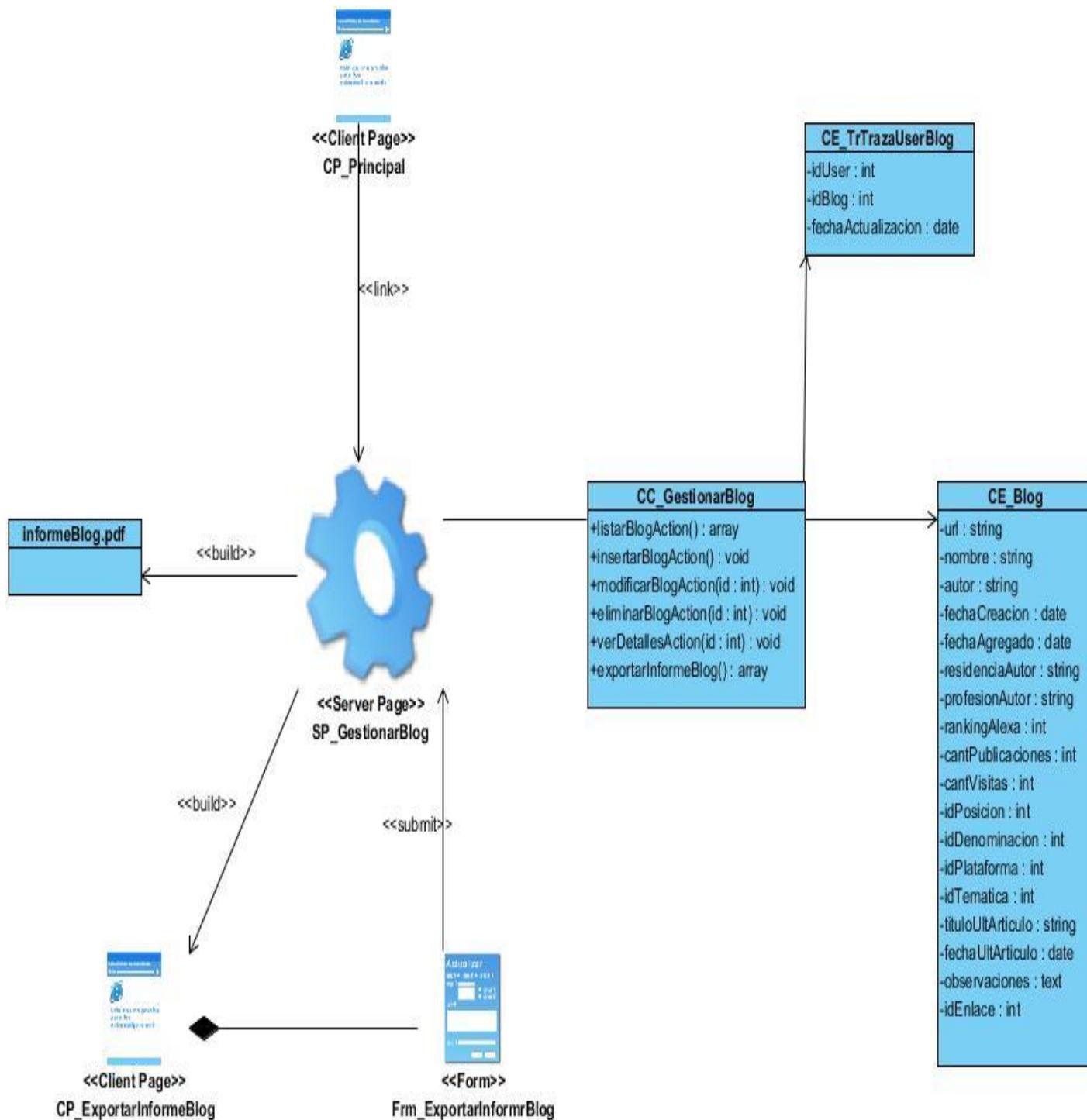


CU Exportar Informes

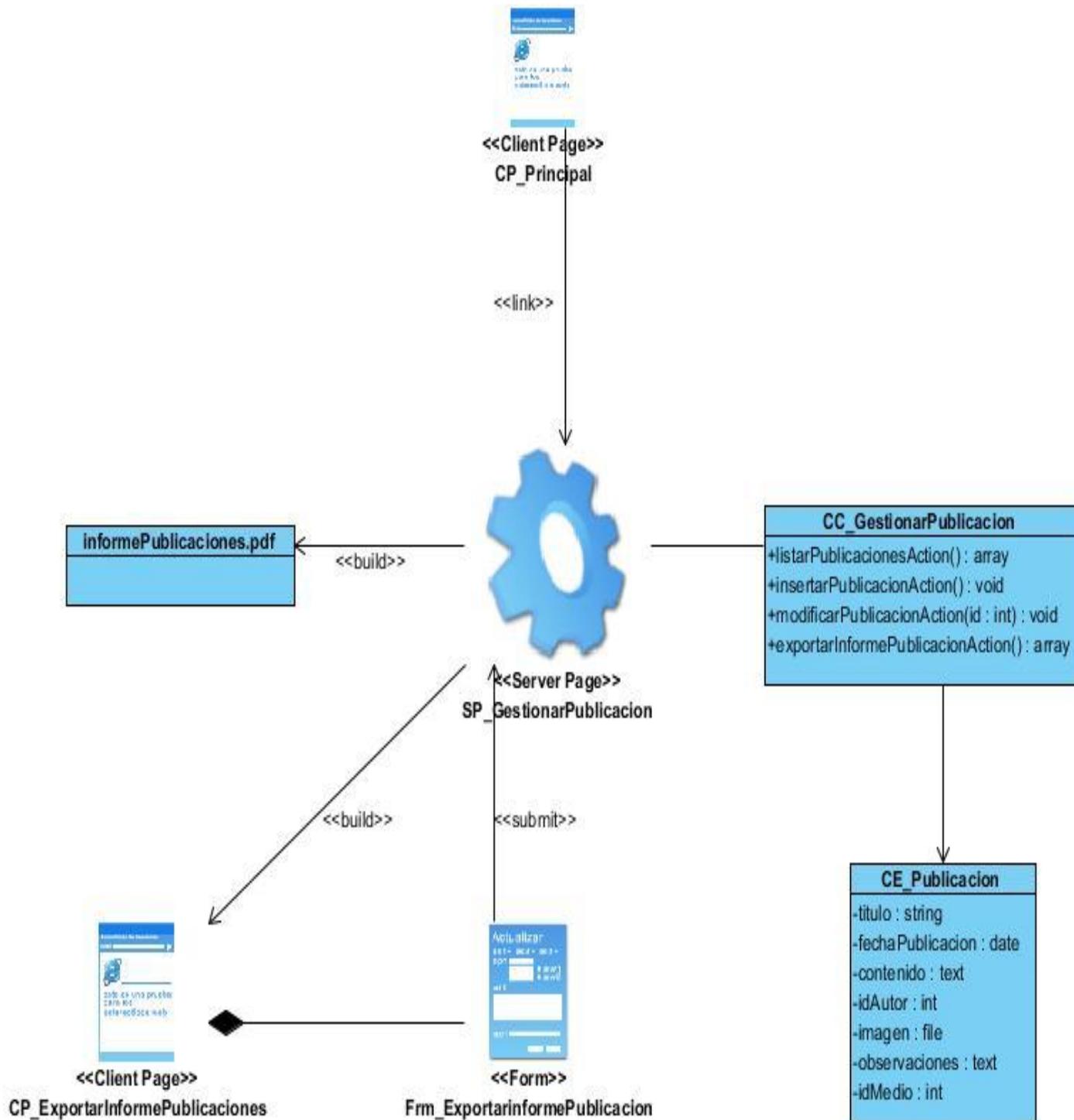
Sección Exportar Informe de Comentarios



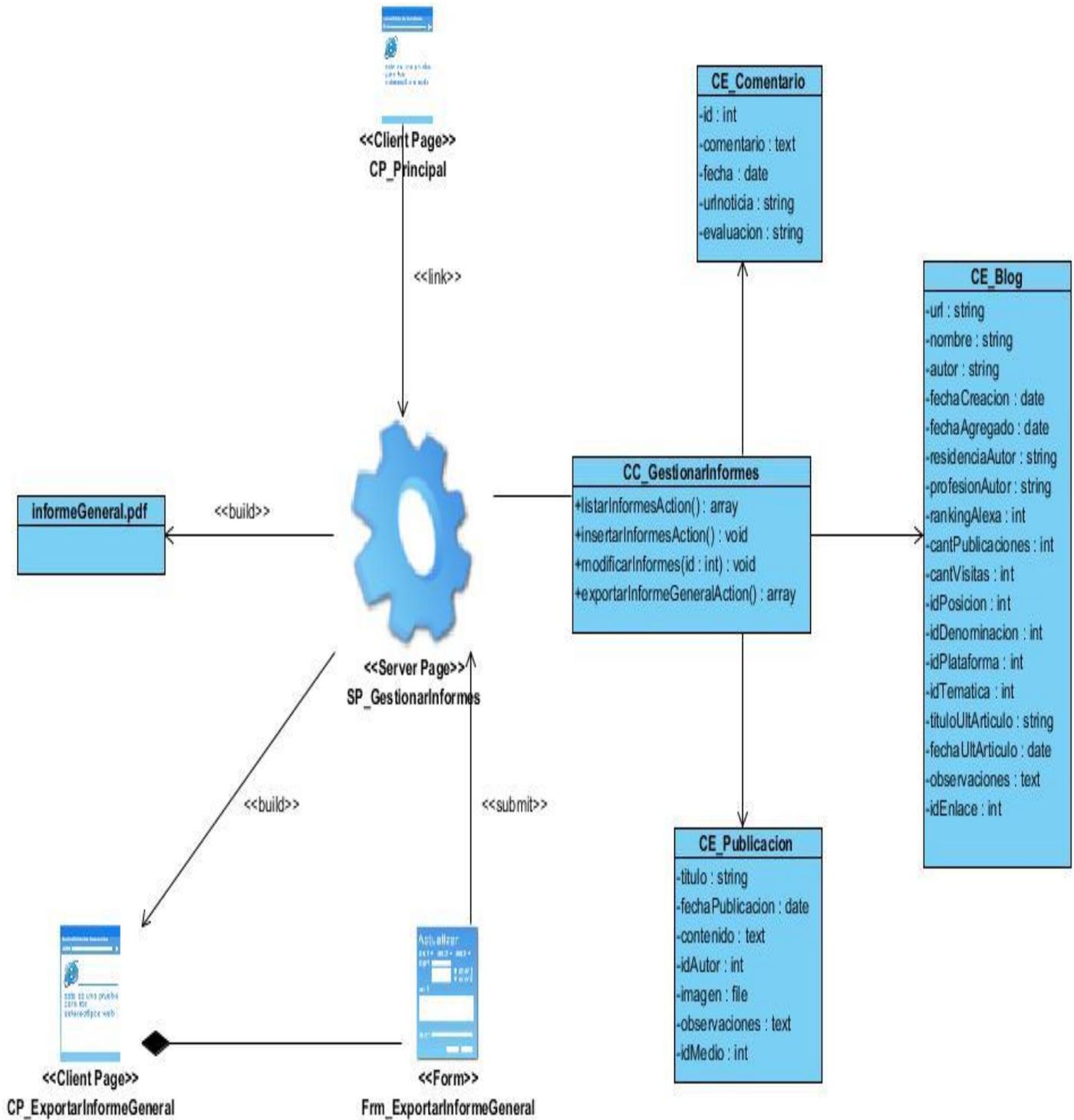
Sección Exportar Informe de Blogs



Sección Exportar Informe de Publicaciones

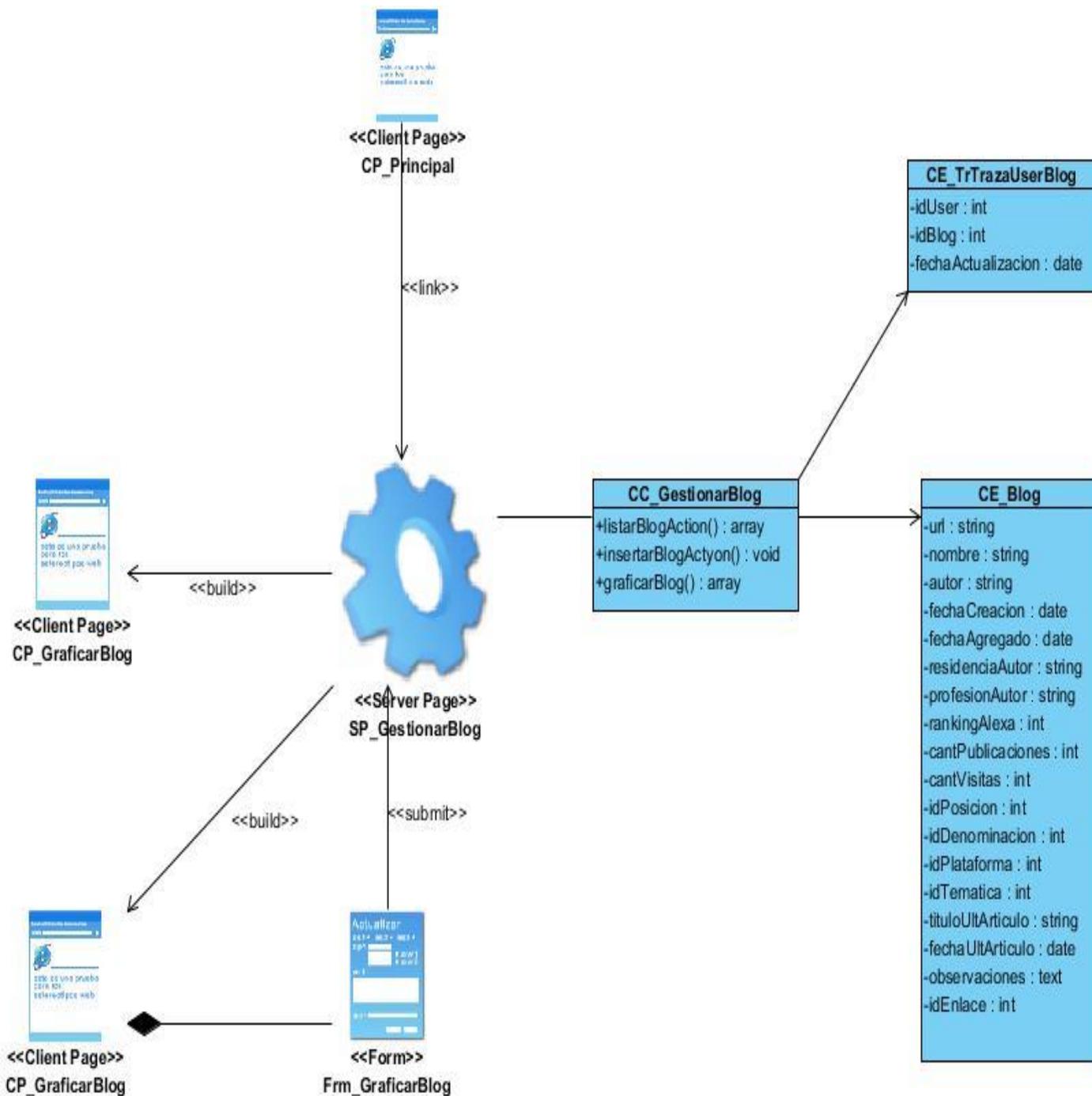


Sección Exportar Informe General

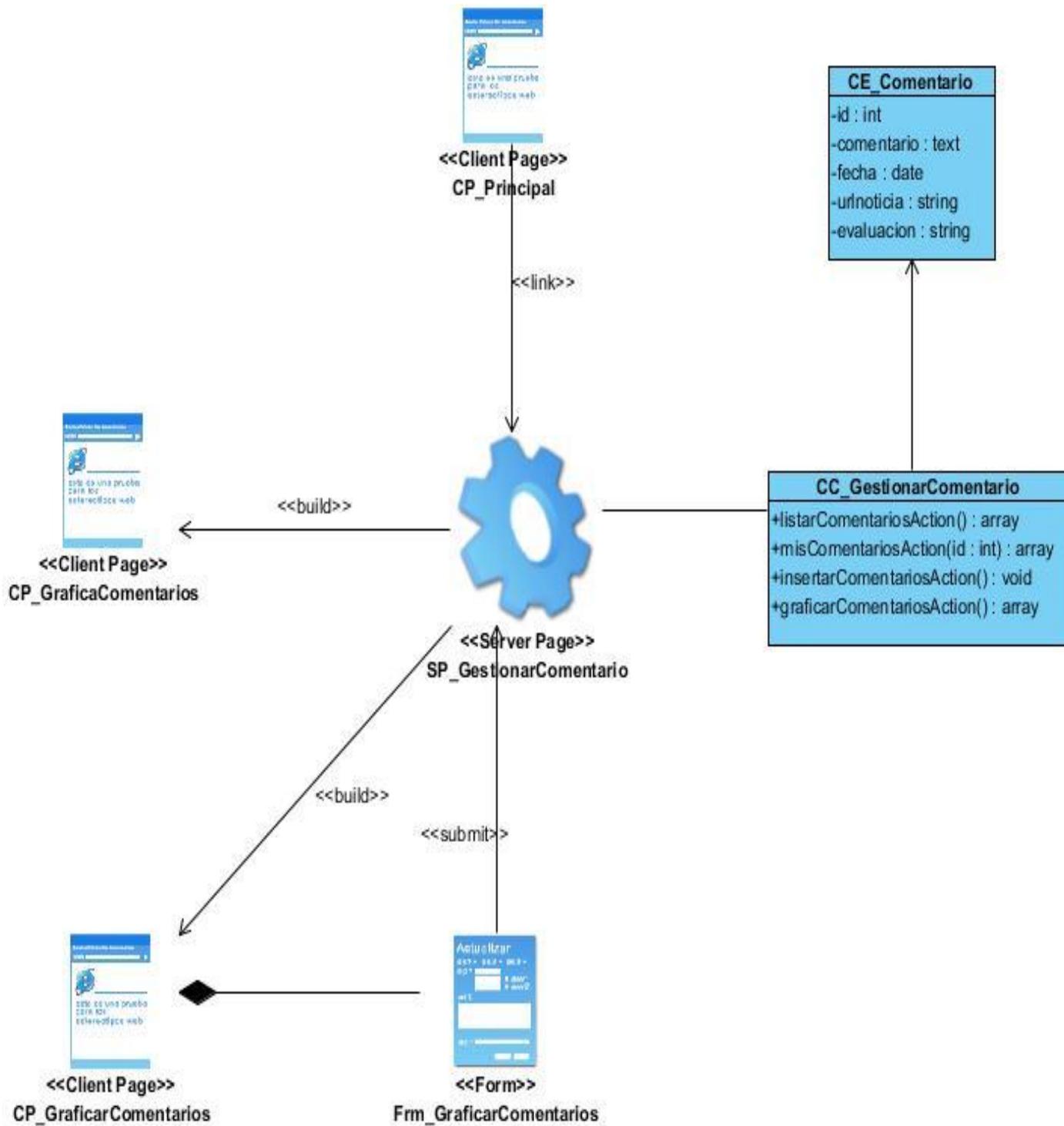


CU Generar Gráficas

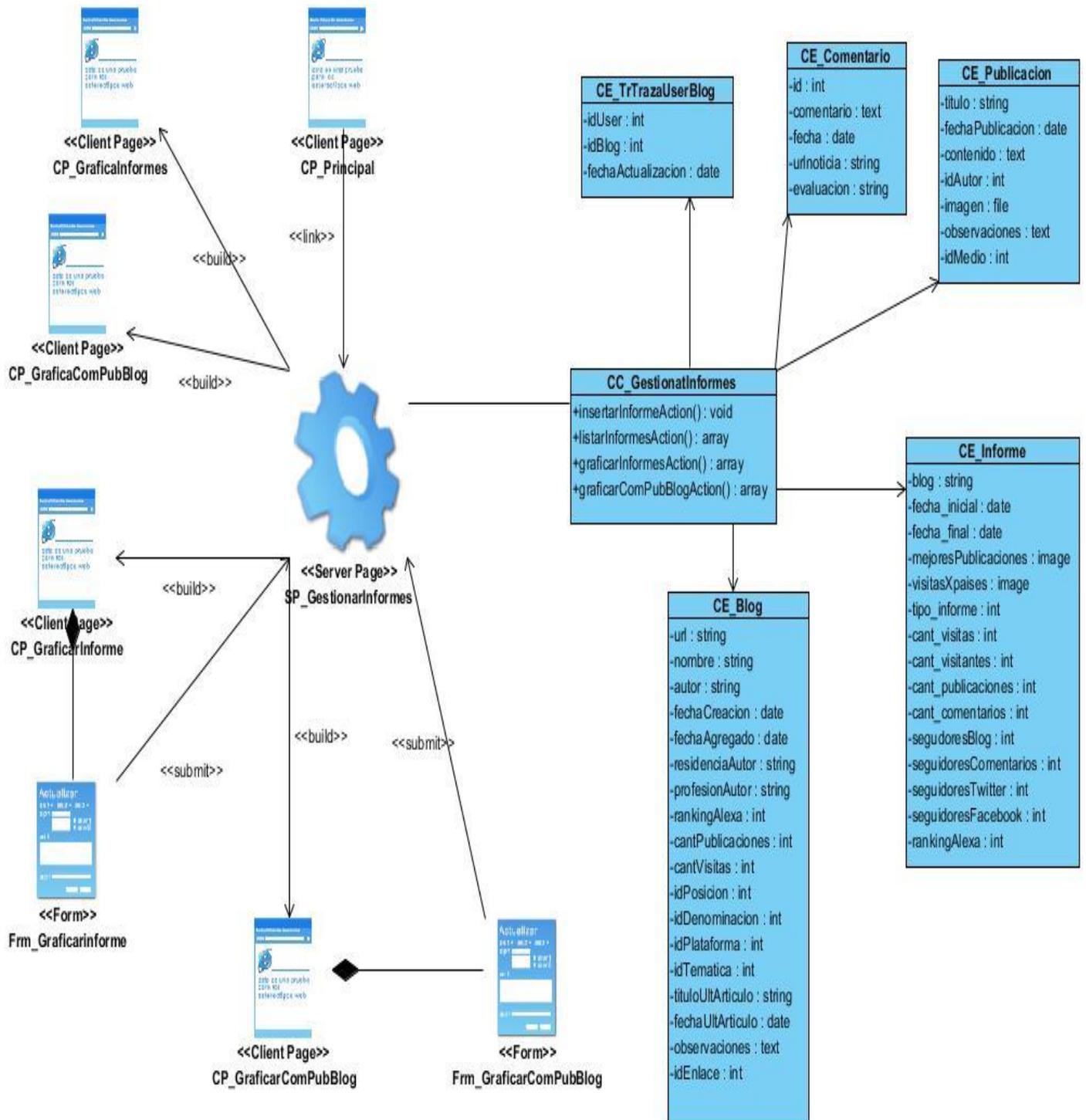
Sección Generar Gráfica de Blog



Sección Generar Gráfica de Comentario

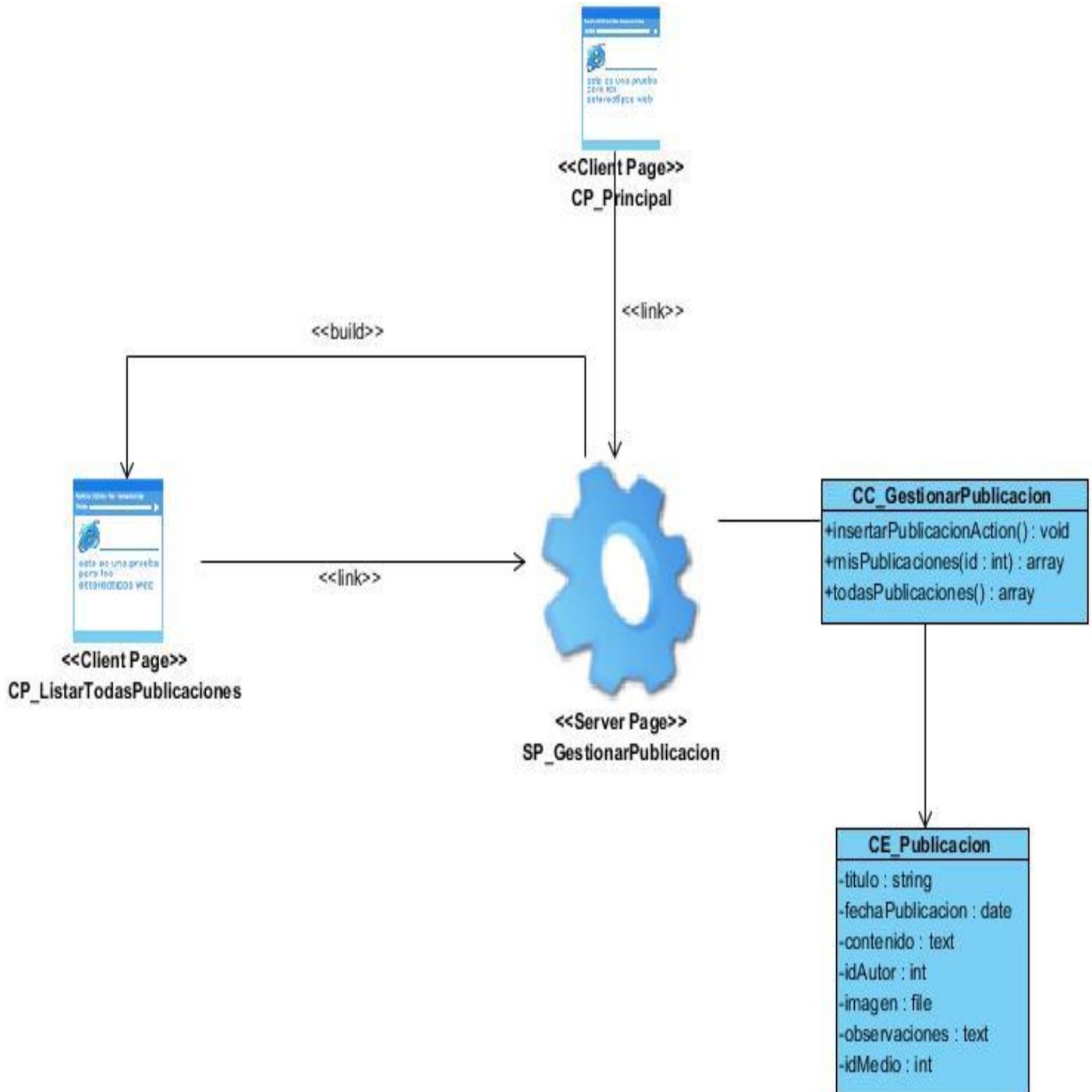


Secciones Generar Gráfica de Informes Estadísticos y Generar Gráfica ComPubBlog

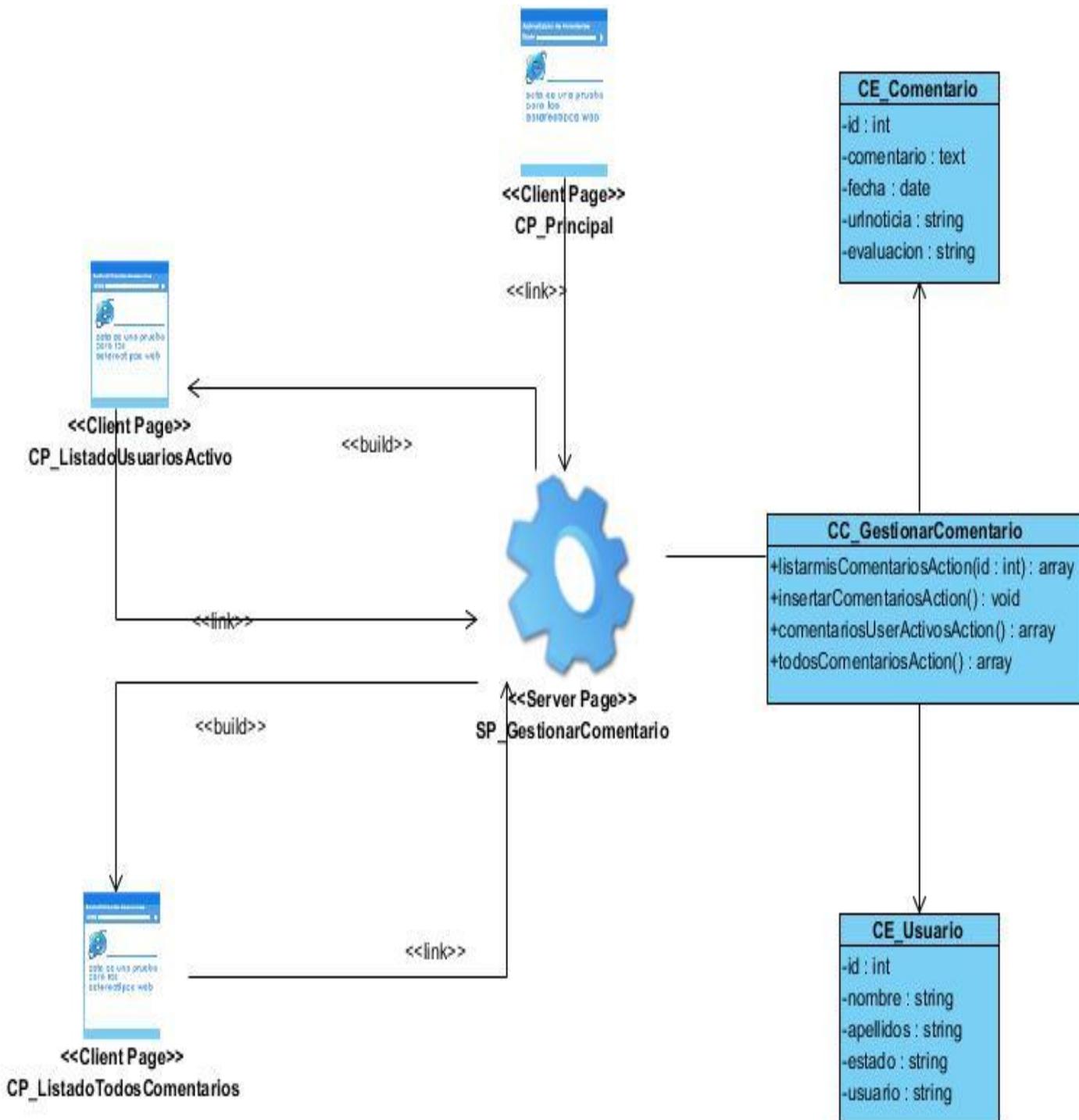


CU Listar Contenidos

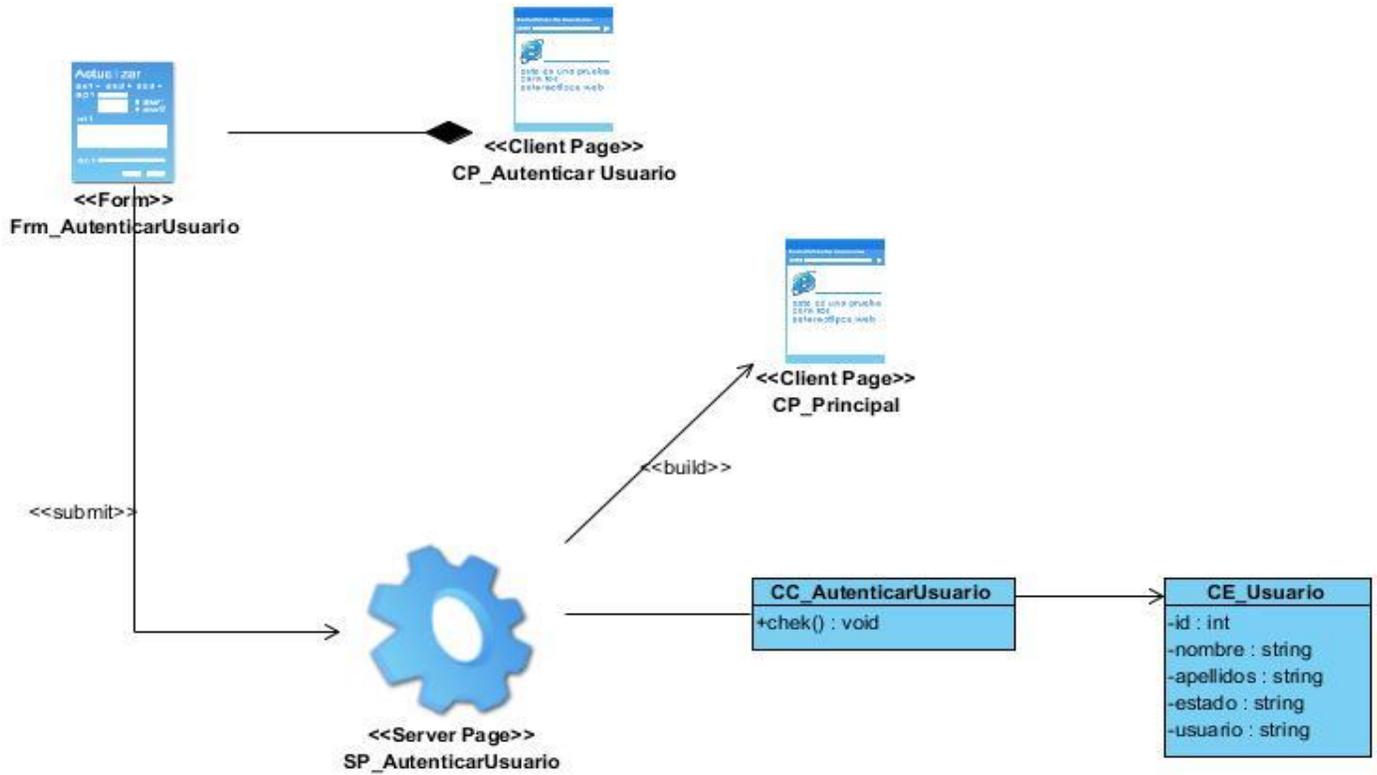
Sección Listar Publicaciones



Secciones Listado de Todos los Comentarios y Listado de los Comentarios de los Usuarios Activos



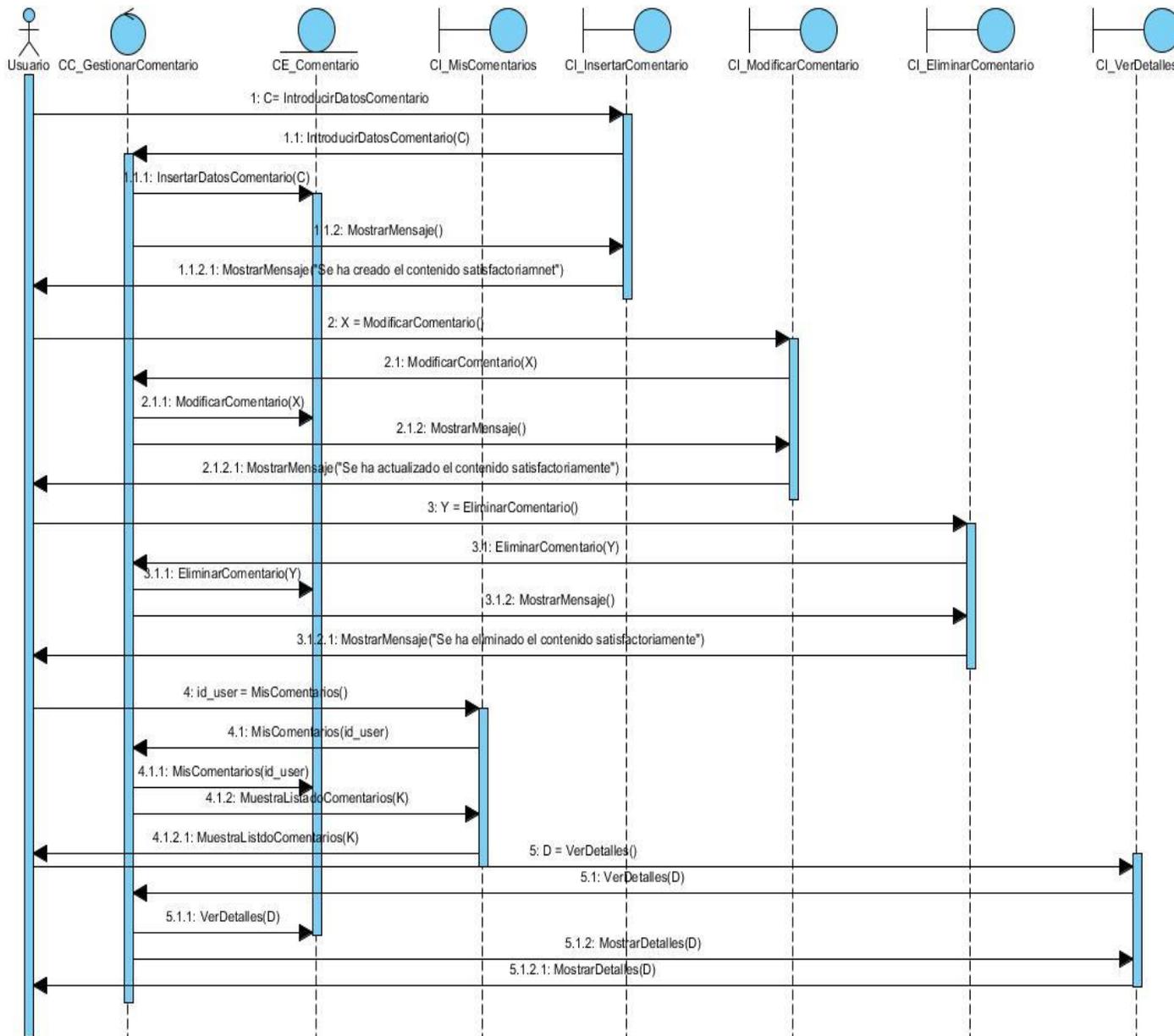
CU Autenticar Usuario



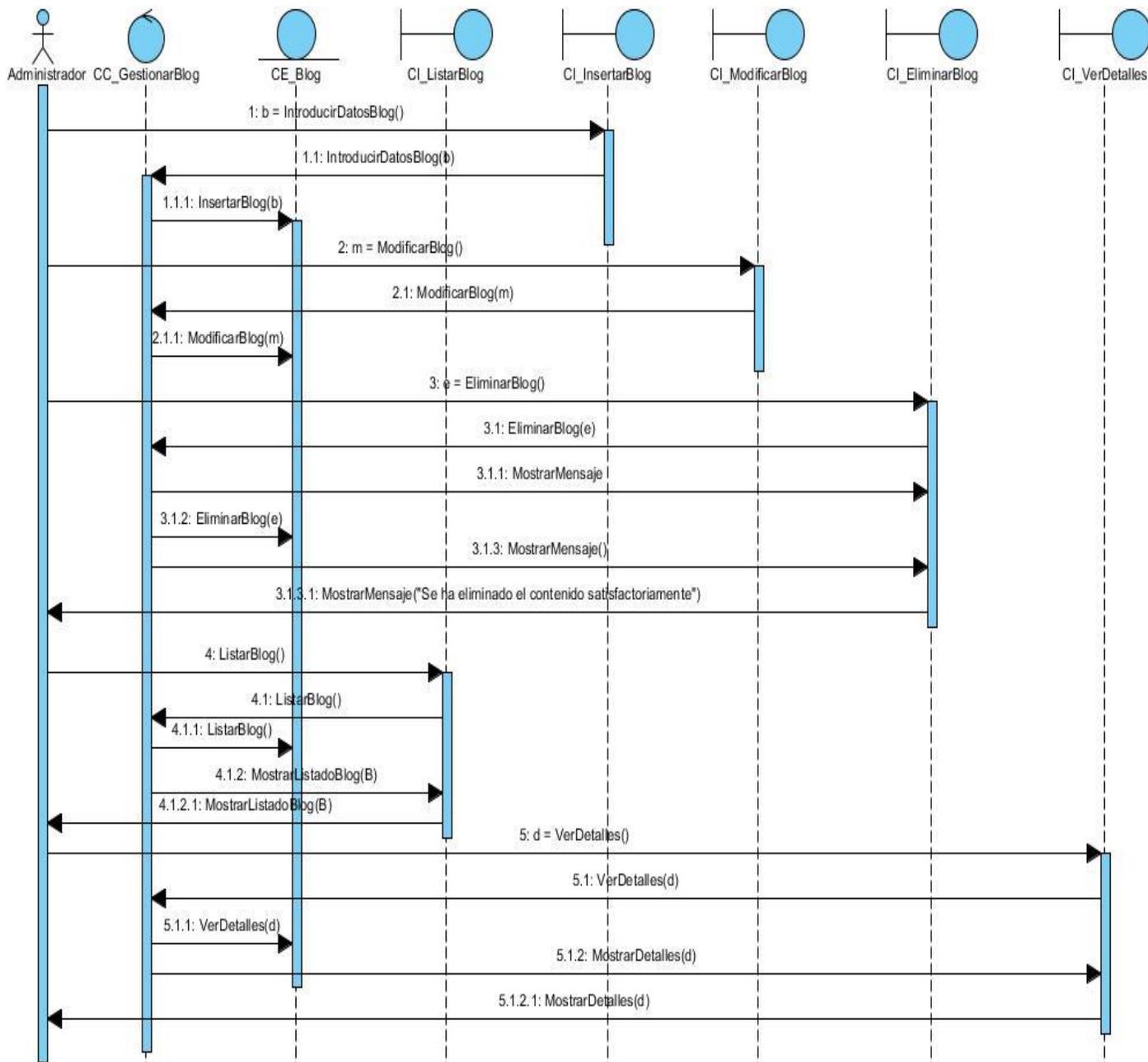
Anexo C

Diagramas de Secuencia

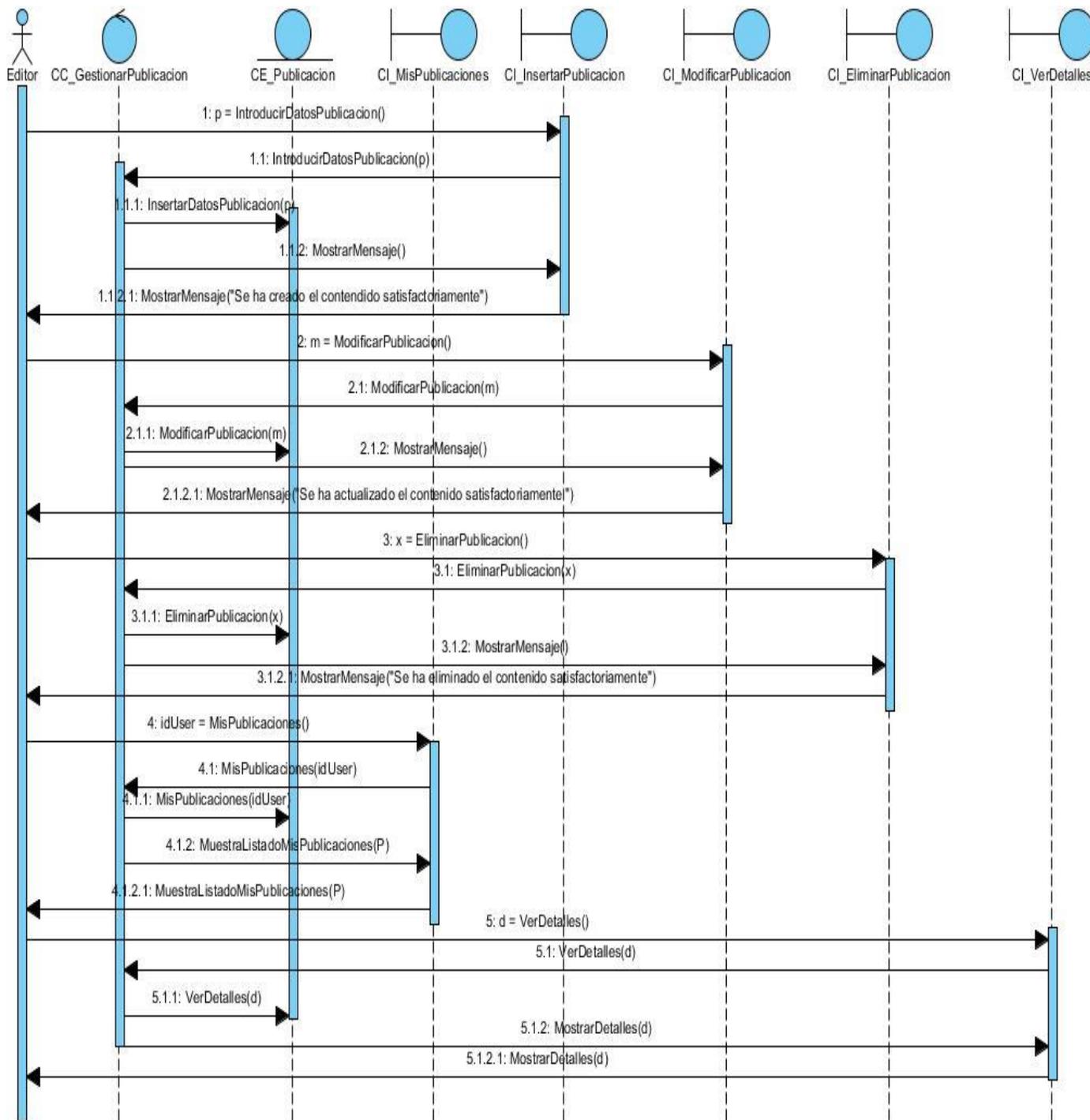
CU Gestionar Comentario



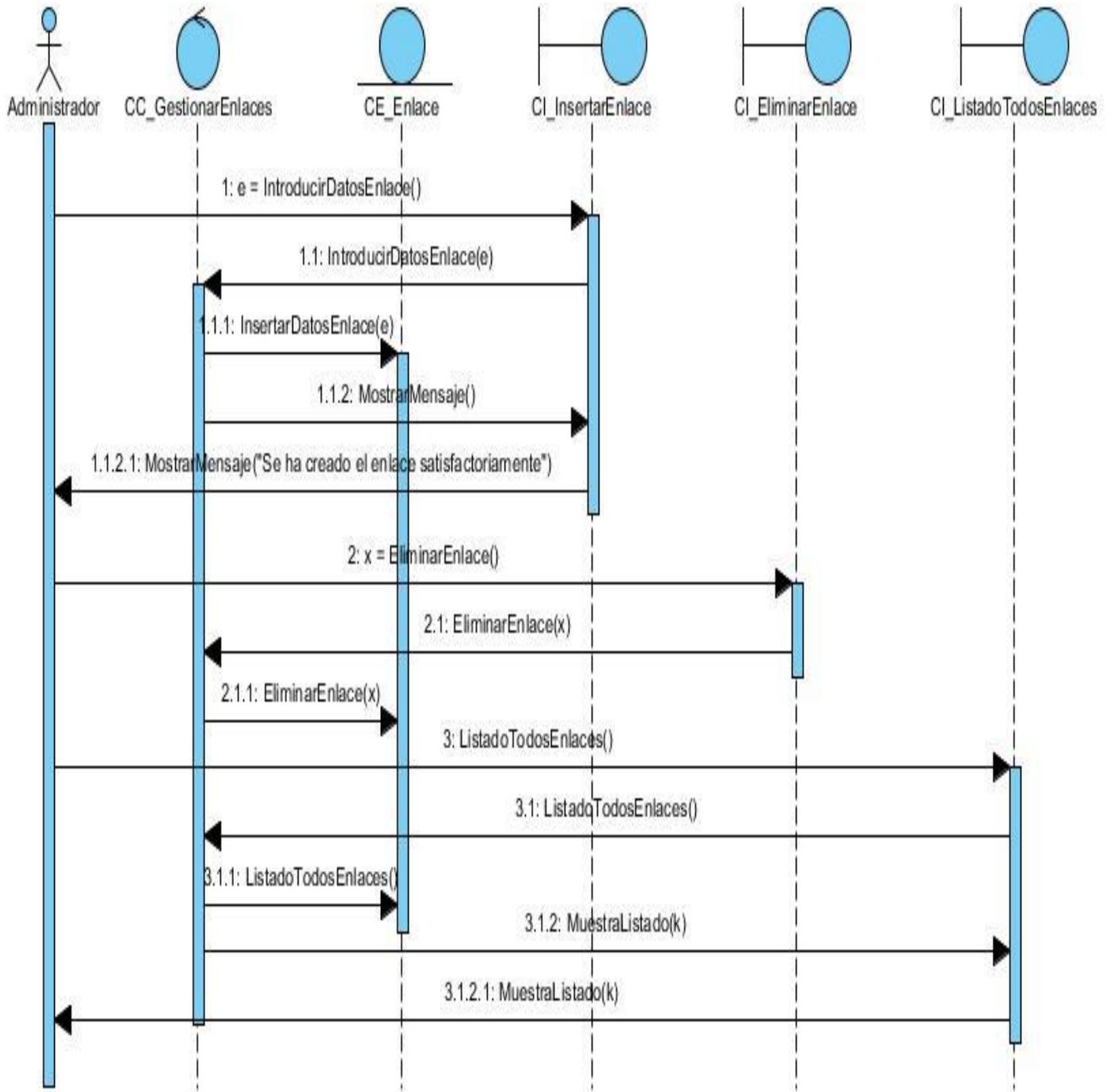
CU Gestionar Blog



CU Gestionar Publicación

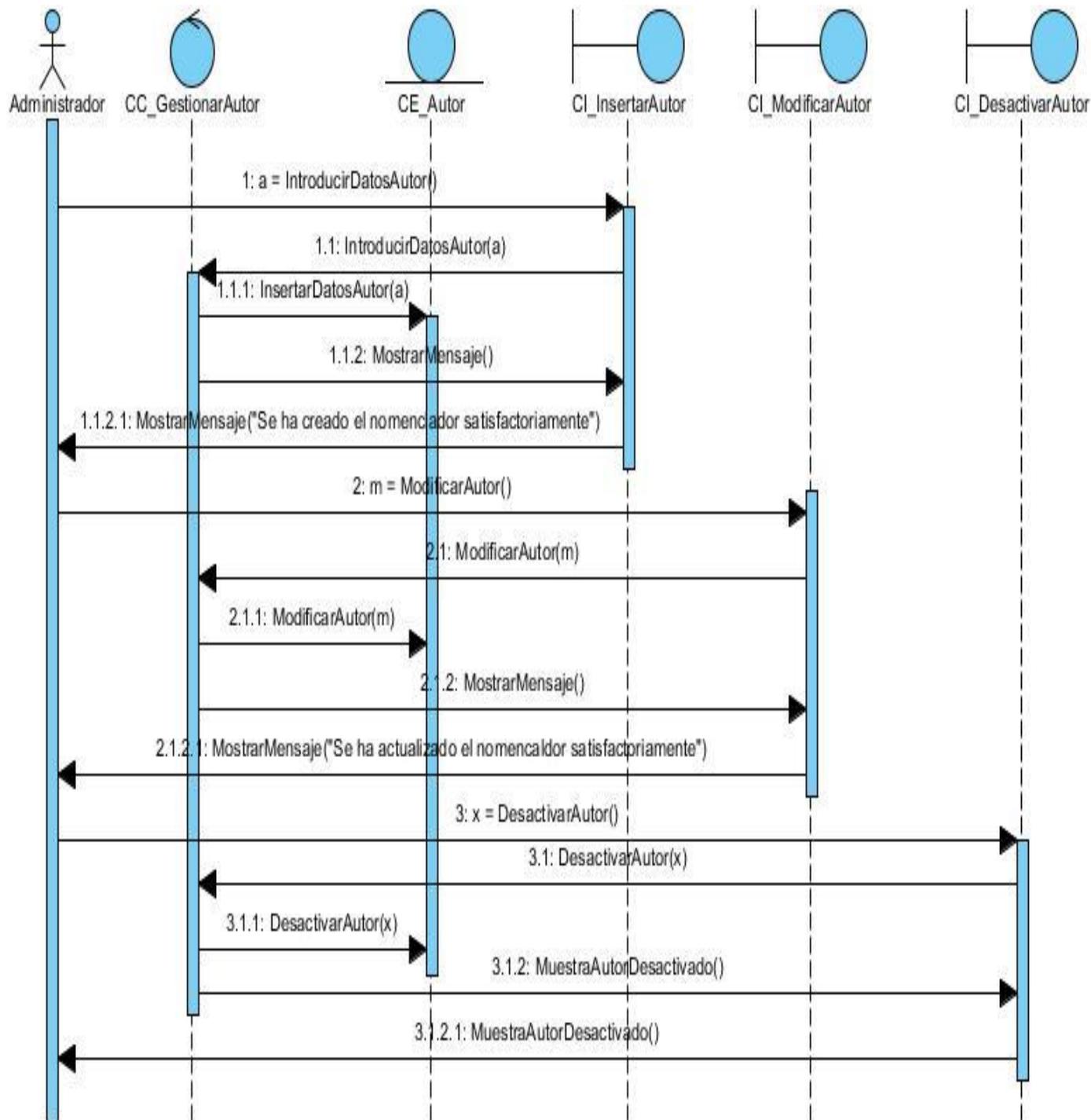


CU Gestionar Enlaces

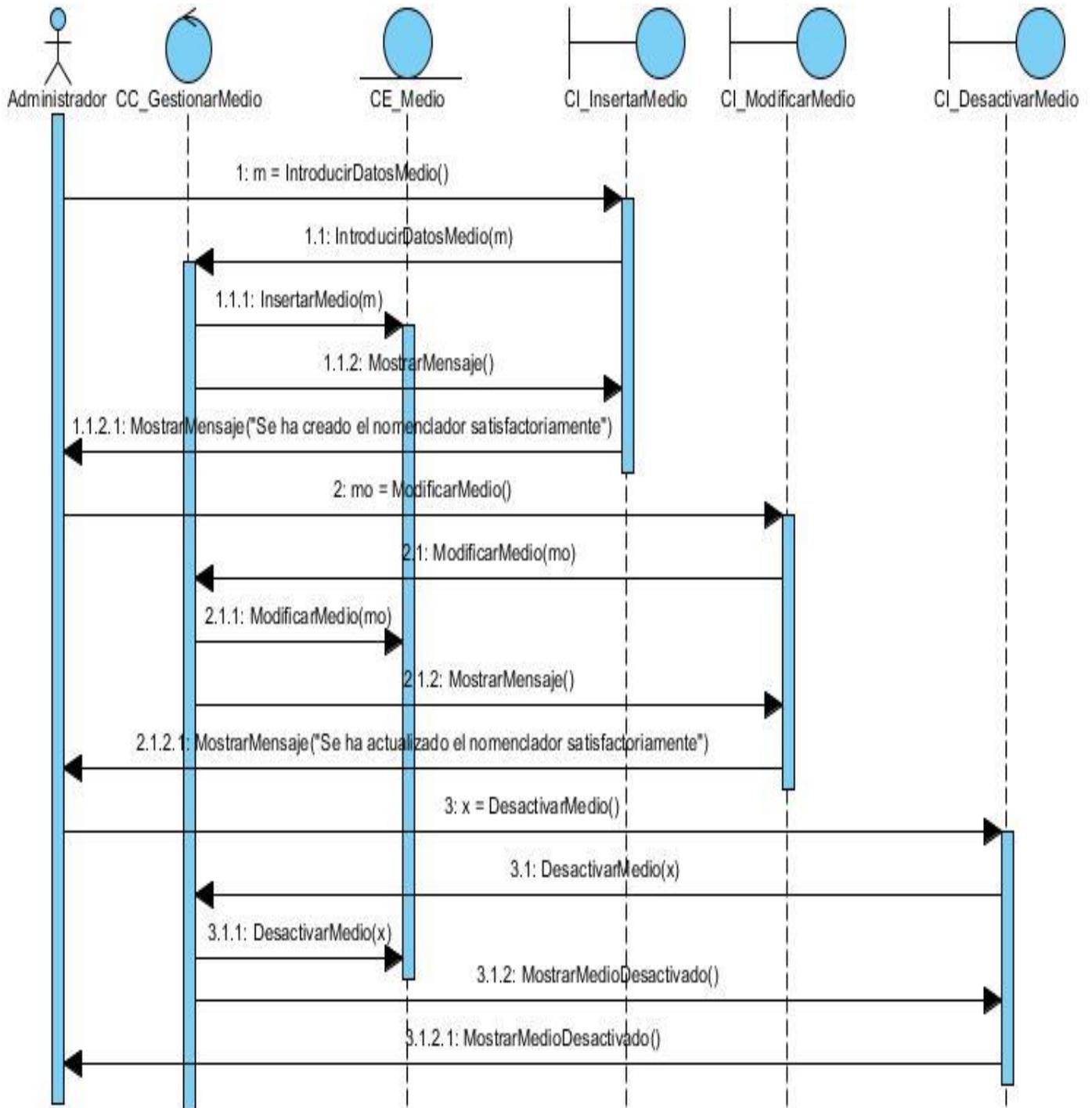


CU Gestionar Nomencladores

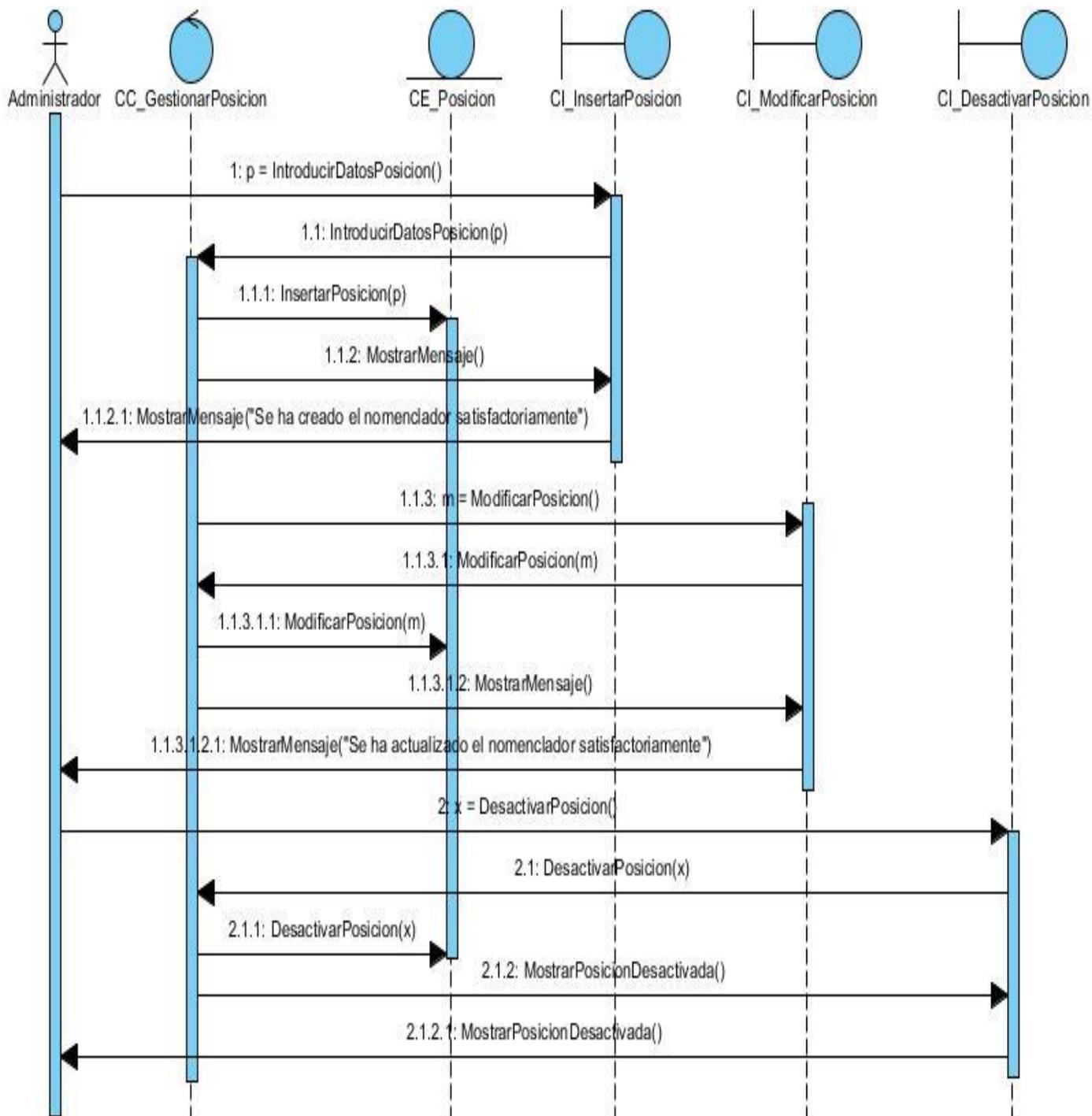
Sección Gestionar Nomenclador Autor



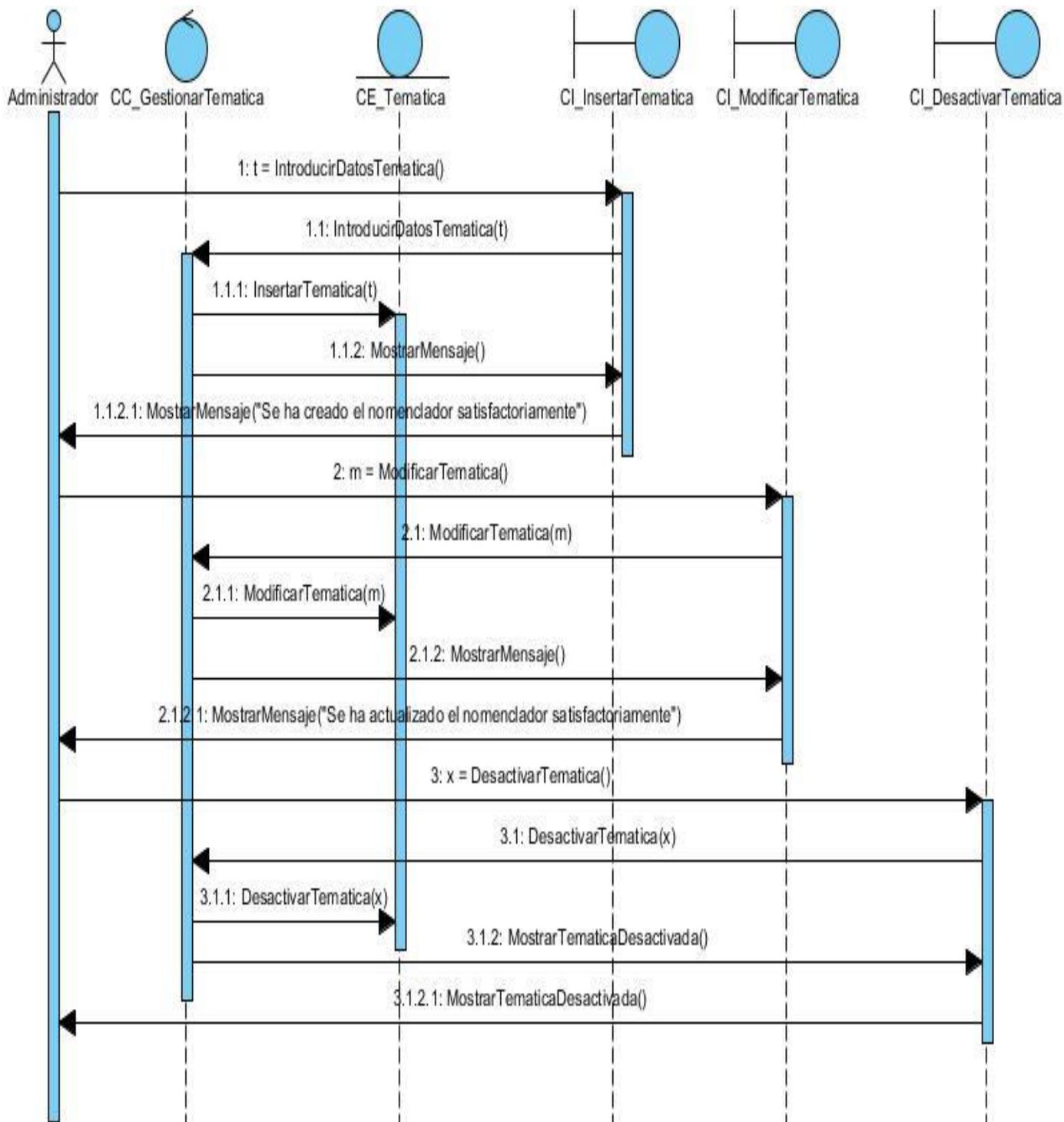
Sección Gestionar Nomenclador Medio



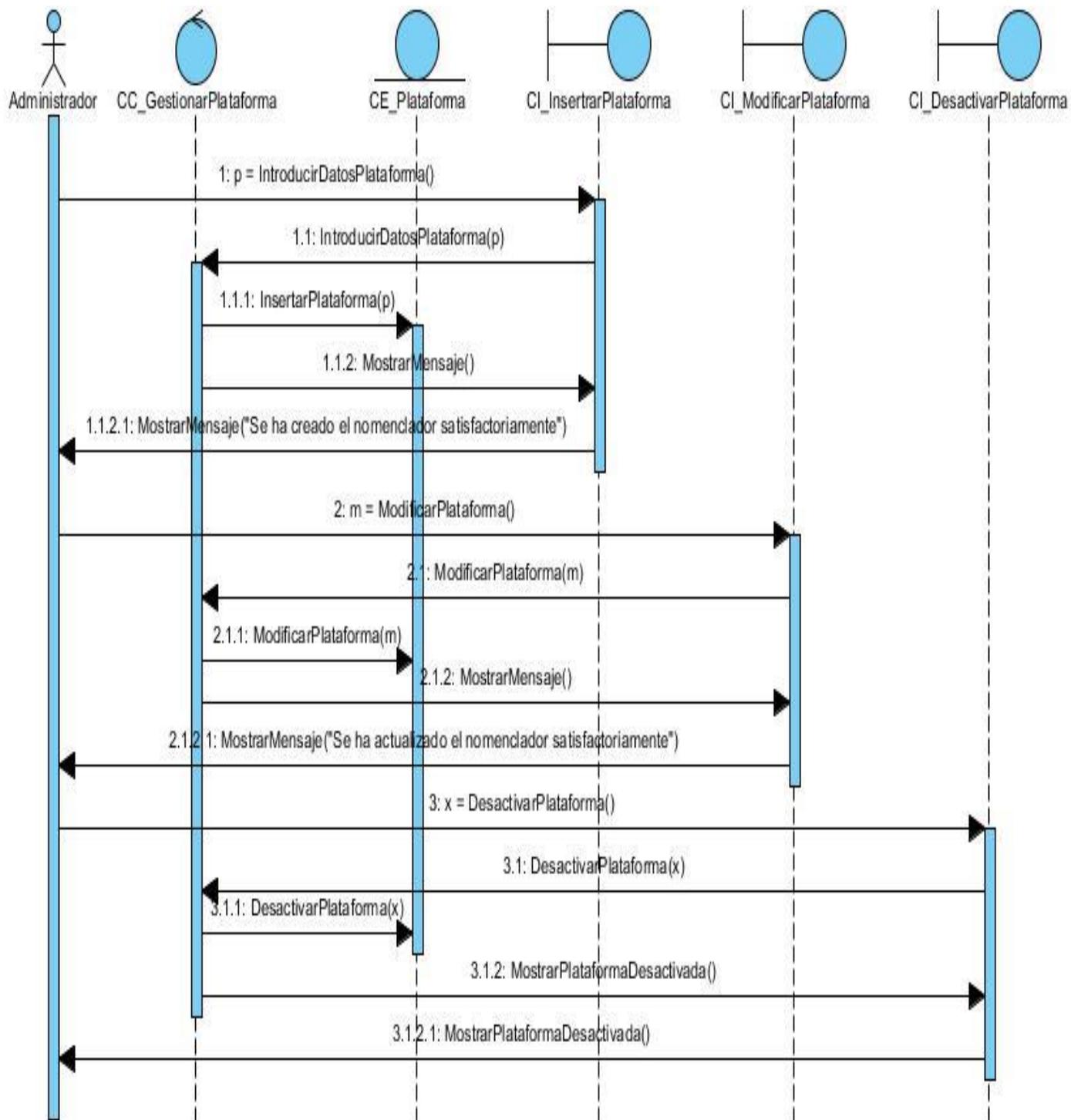
Sección Gestionar Nomenclador Posición



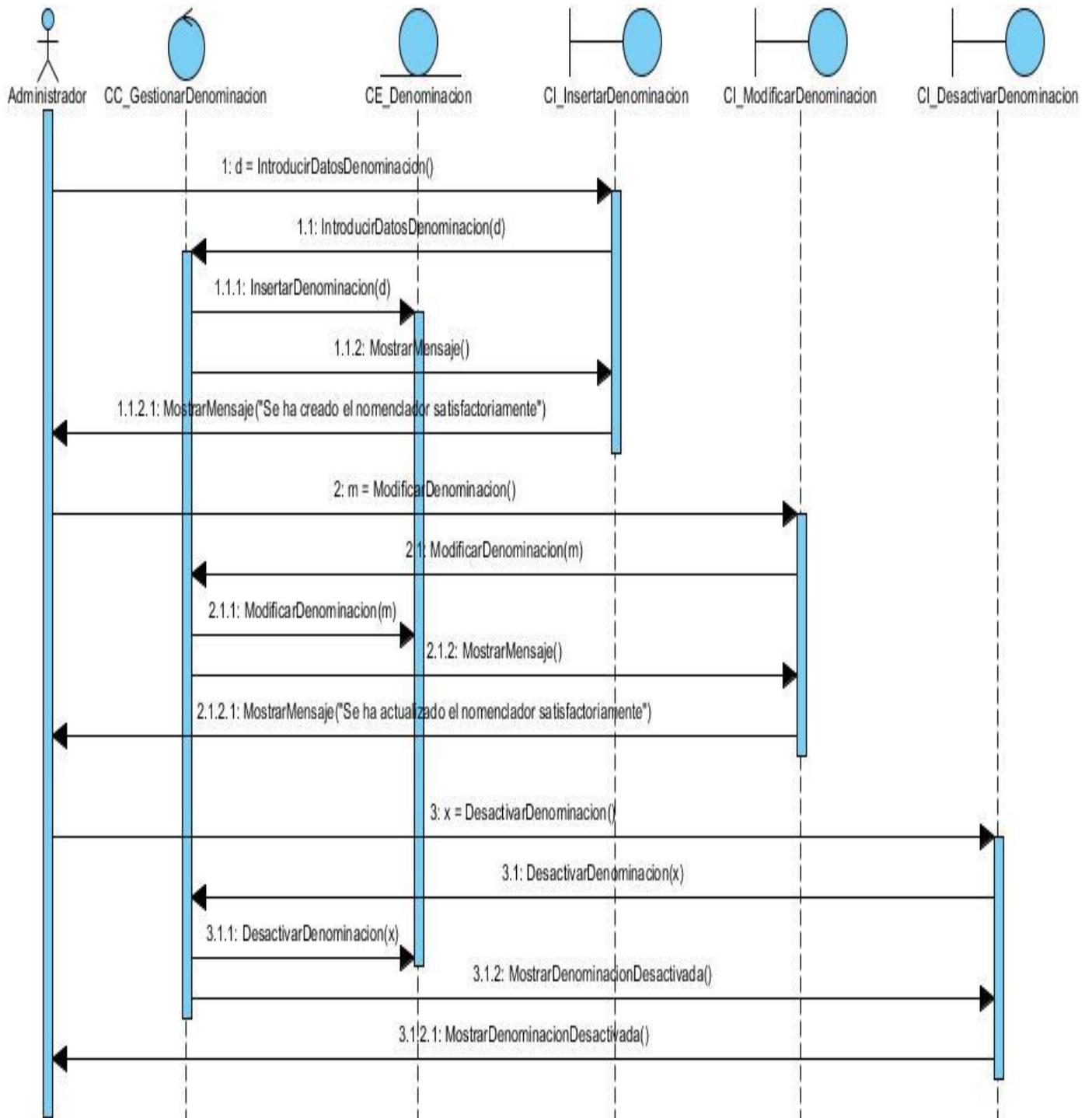
Sección Gestionar Nomenclador Temática



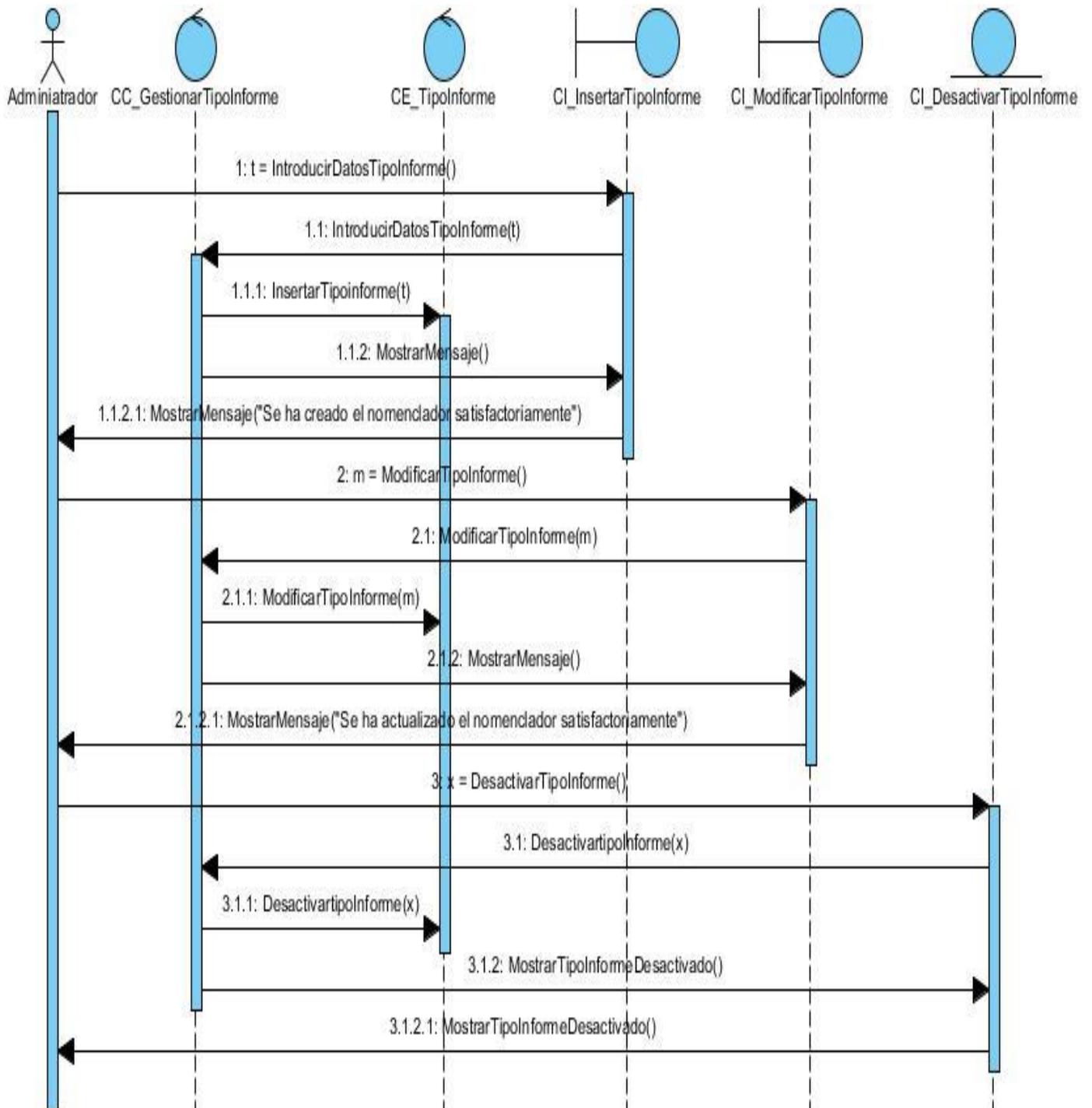
Sección Gestionar Nomenclador Plataforma



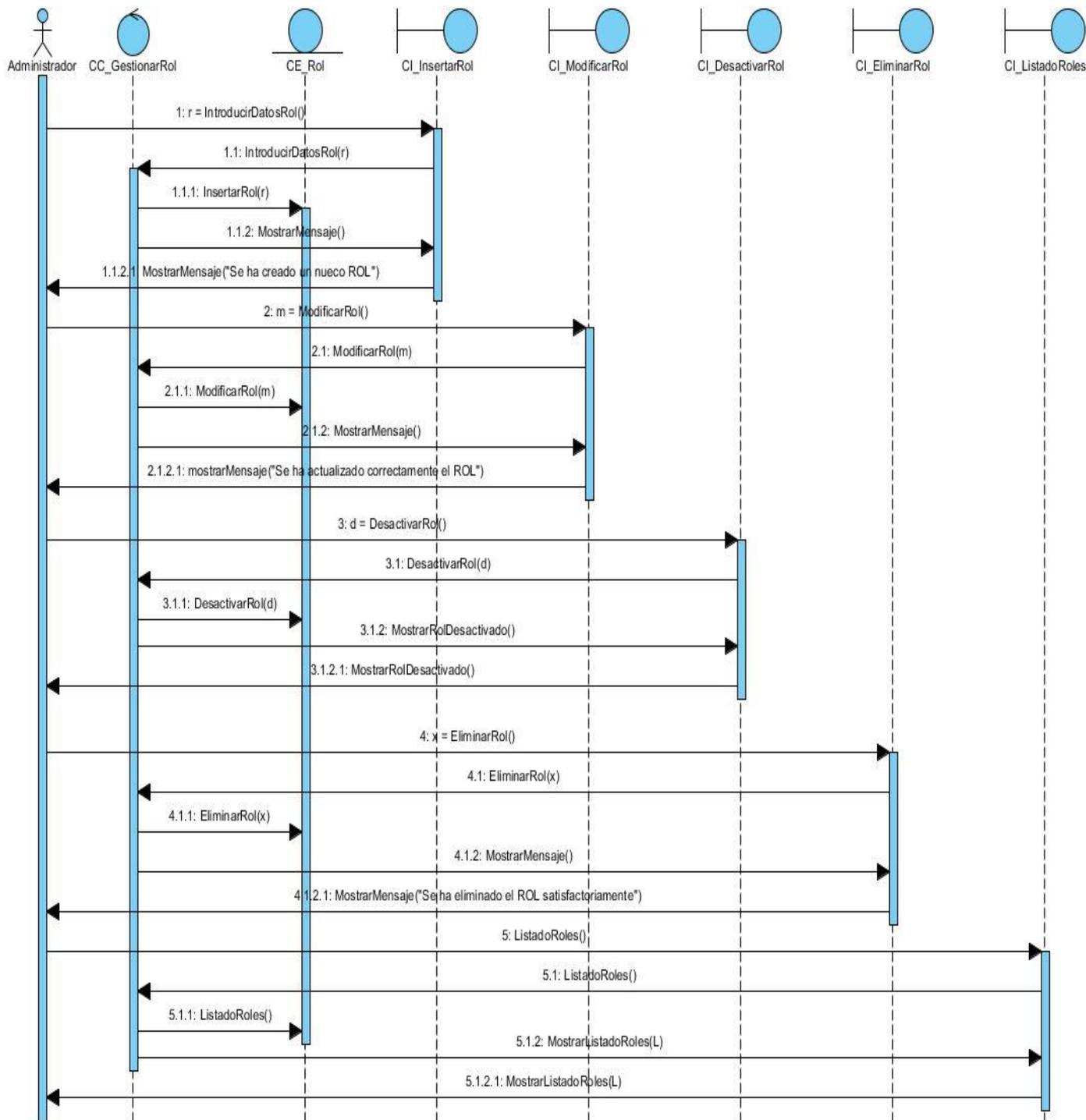
Sección Gestionar Nomenclador Denominación



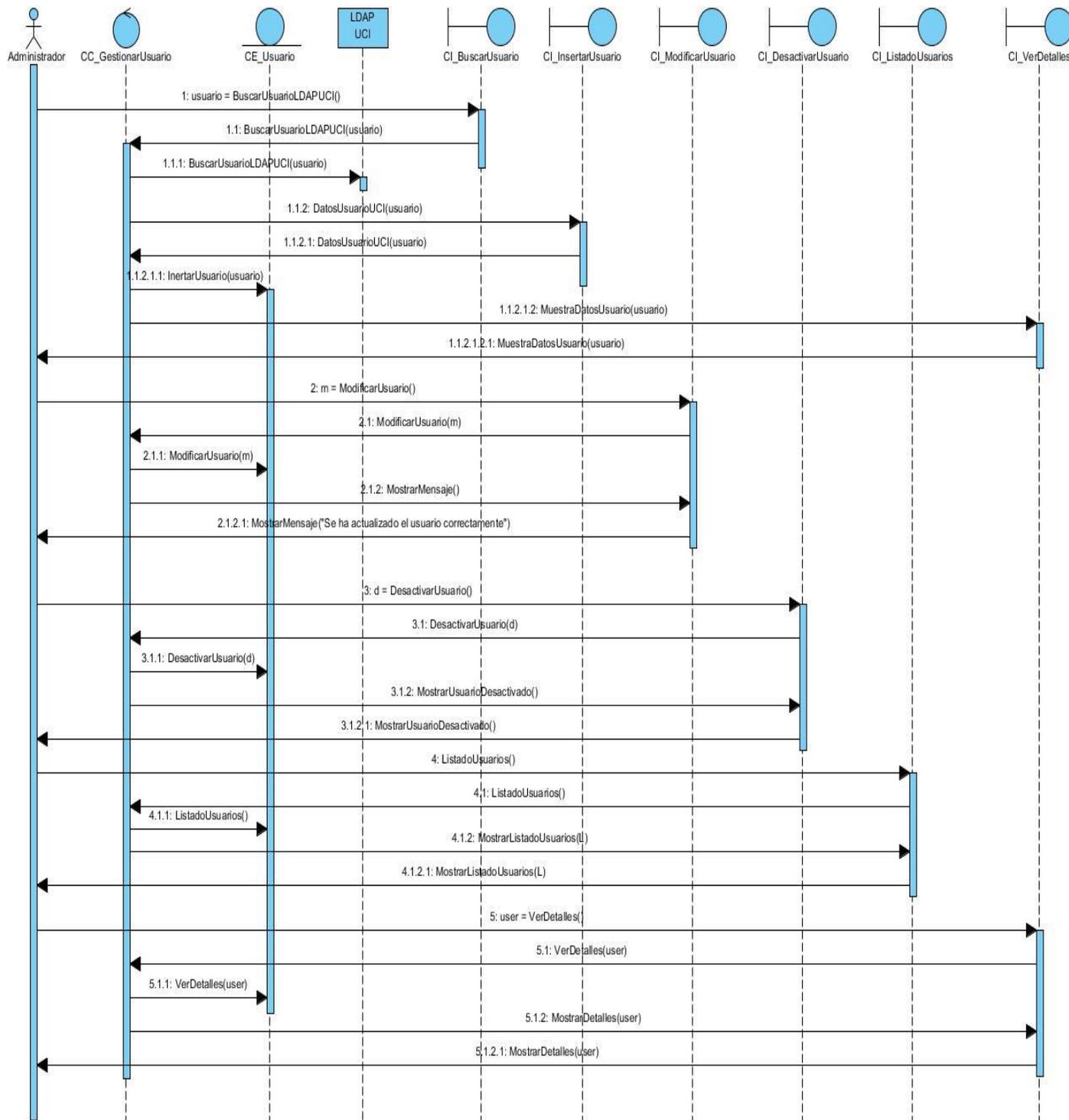
Sección Gestionar Nomenclador Tipo de Informe



CU Gestionar Rol

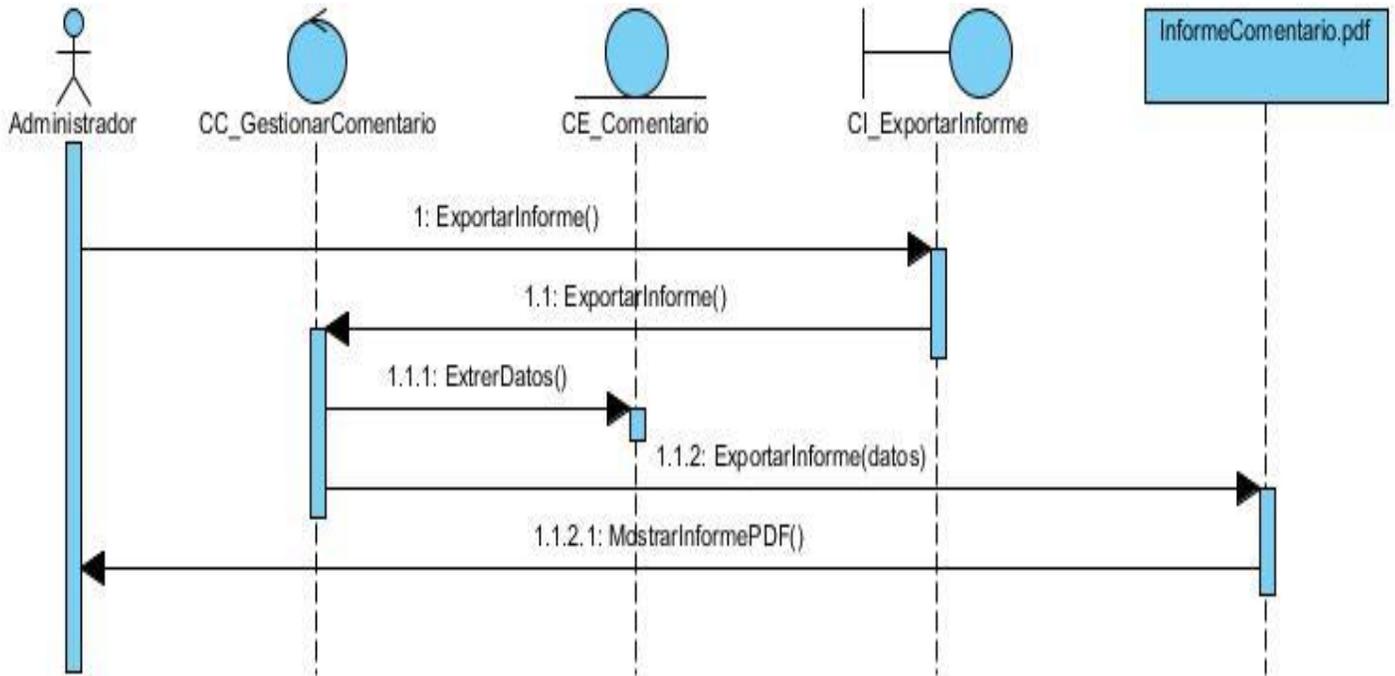


CU Gestionar Usuarios

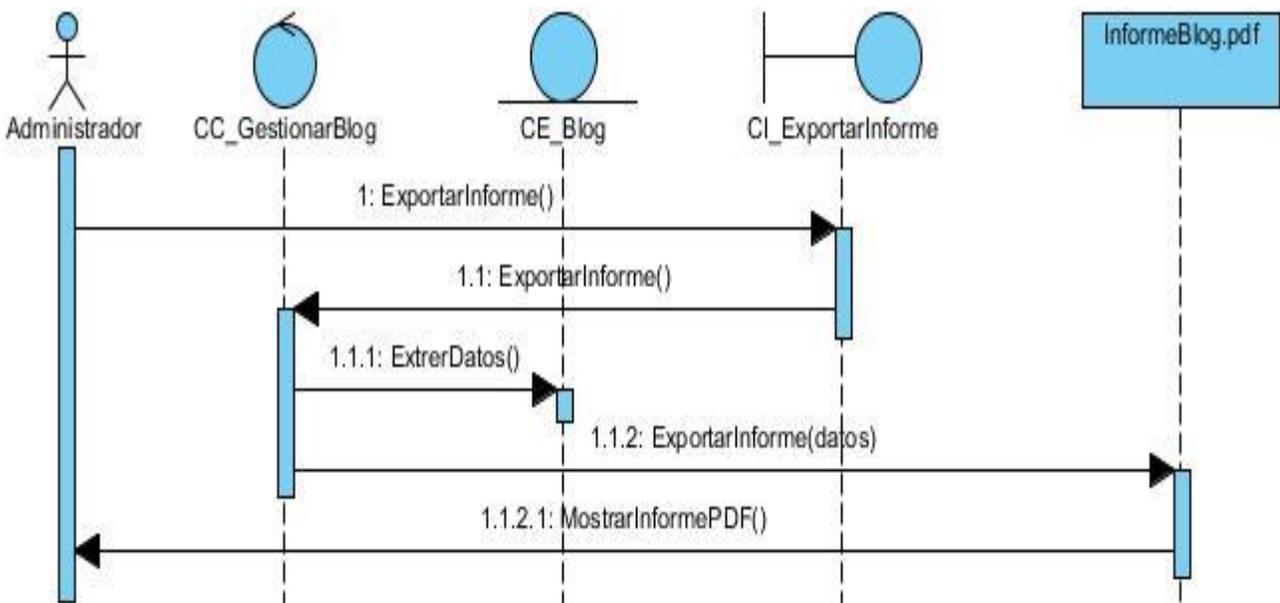


CU Exportar Informes

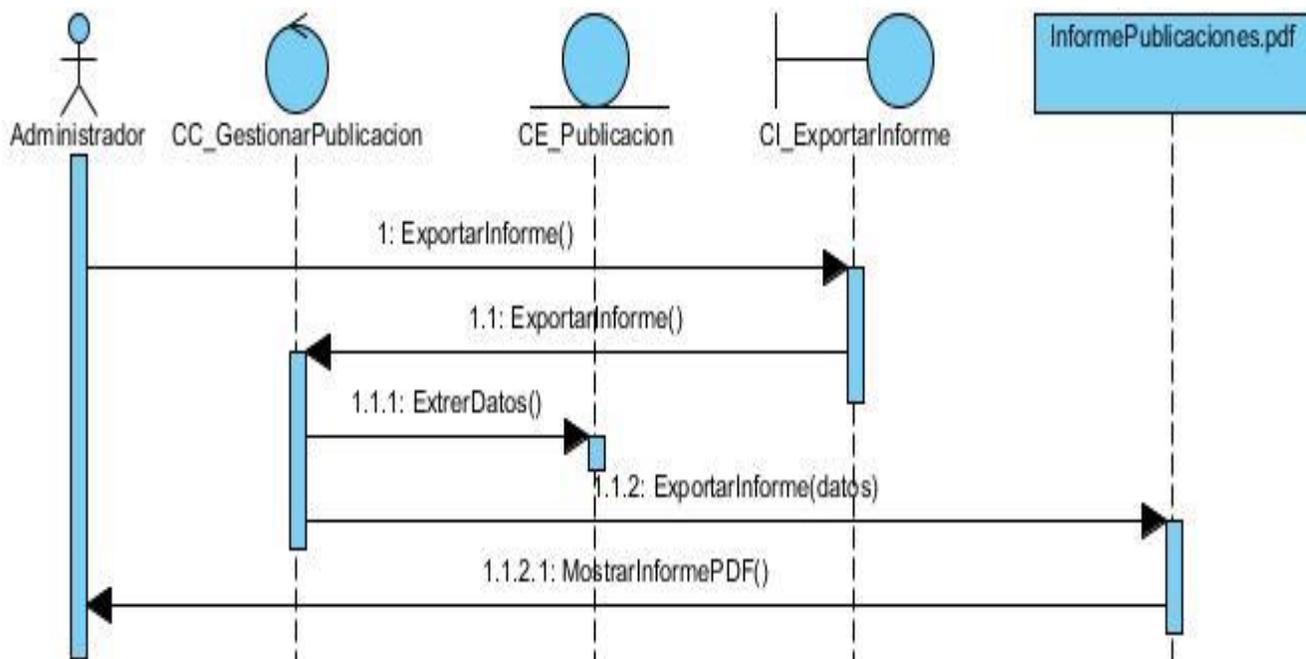
Sección Exportar Informe de Comentarios



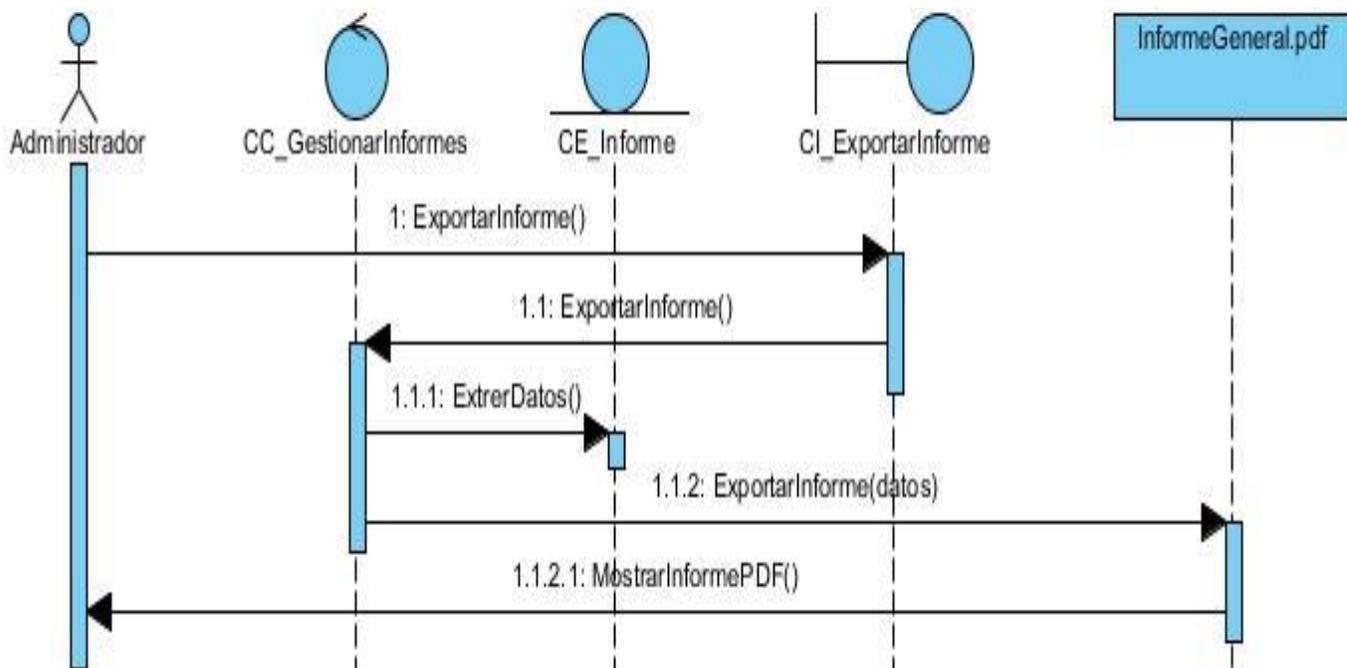
Sección Exportar Informe de Blogs



Sección Exportar Informe de Publicaciones

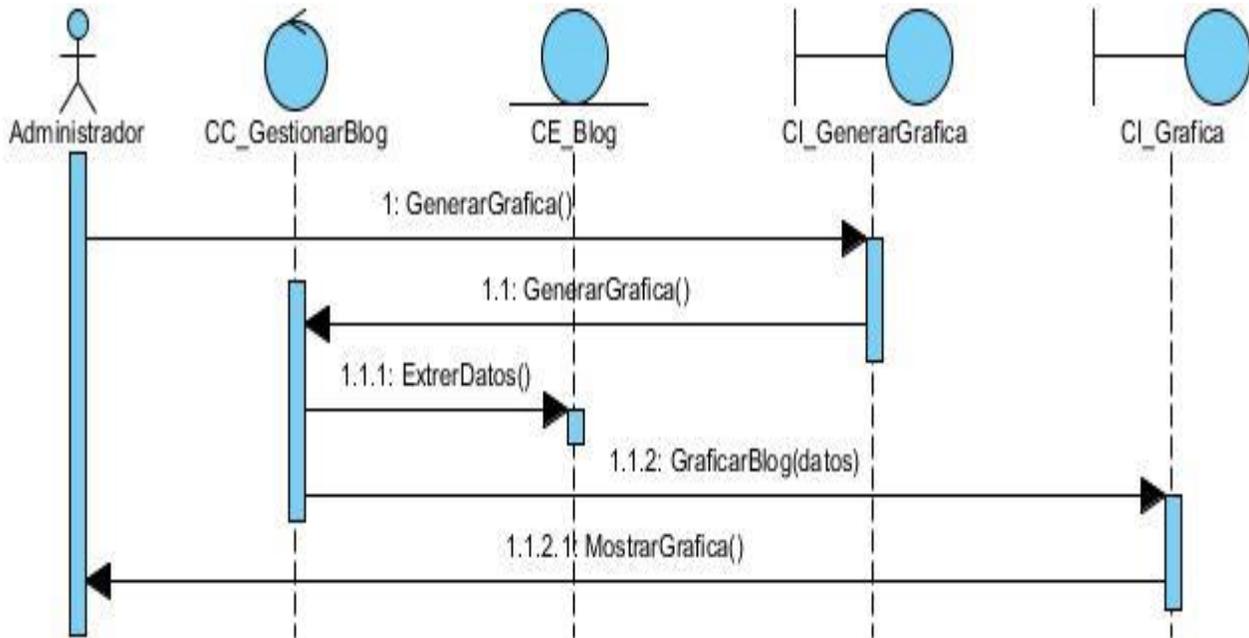


Sección Exportar Informe General

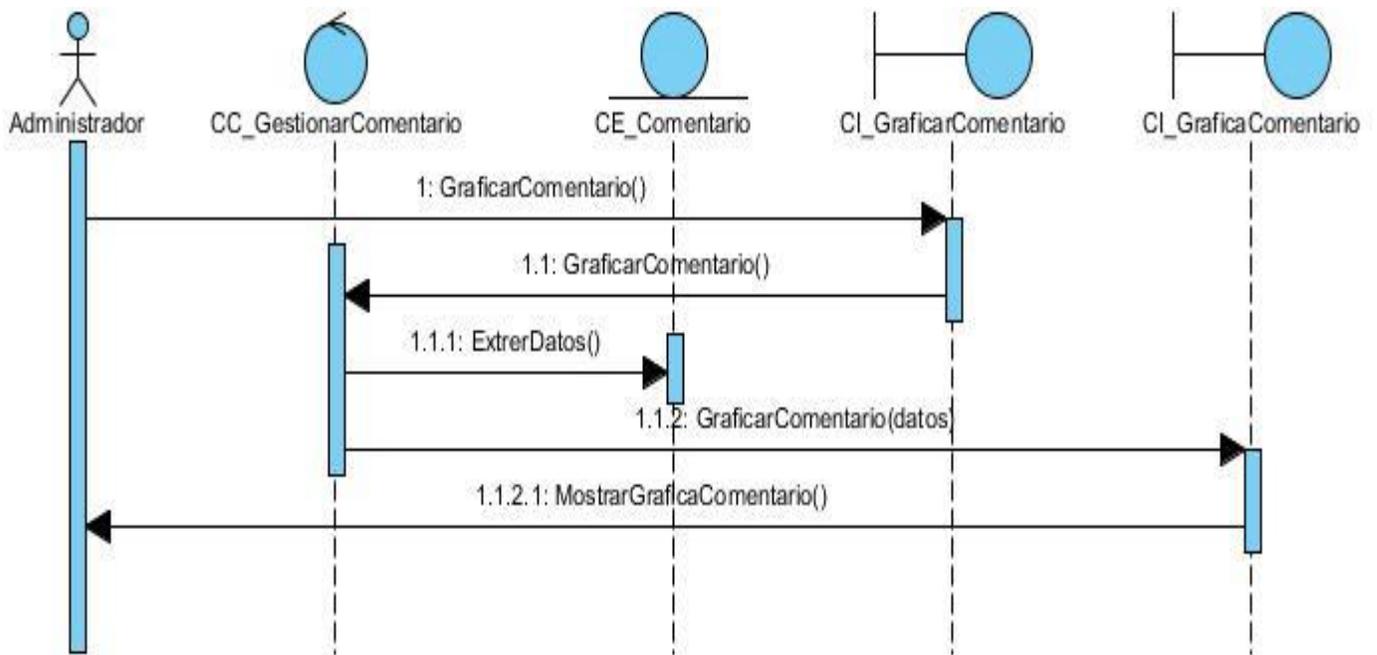


CU Generar Gráficas

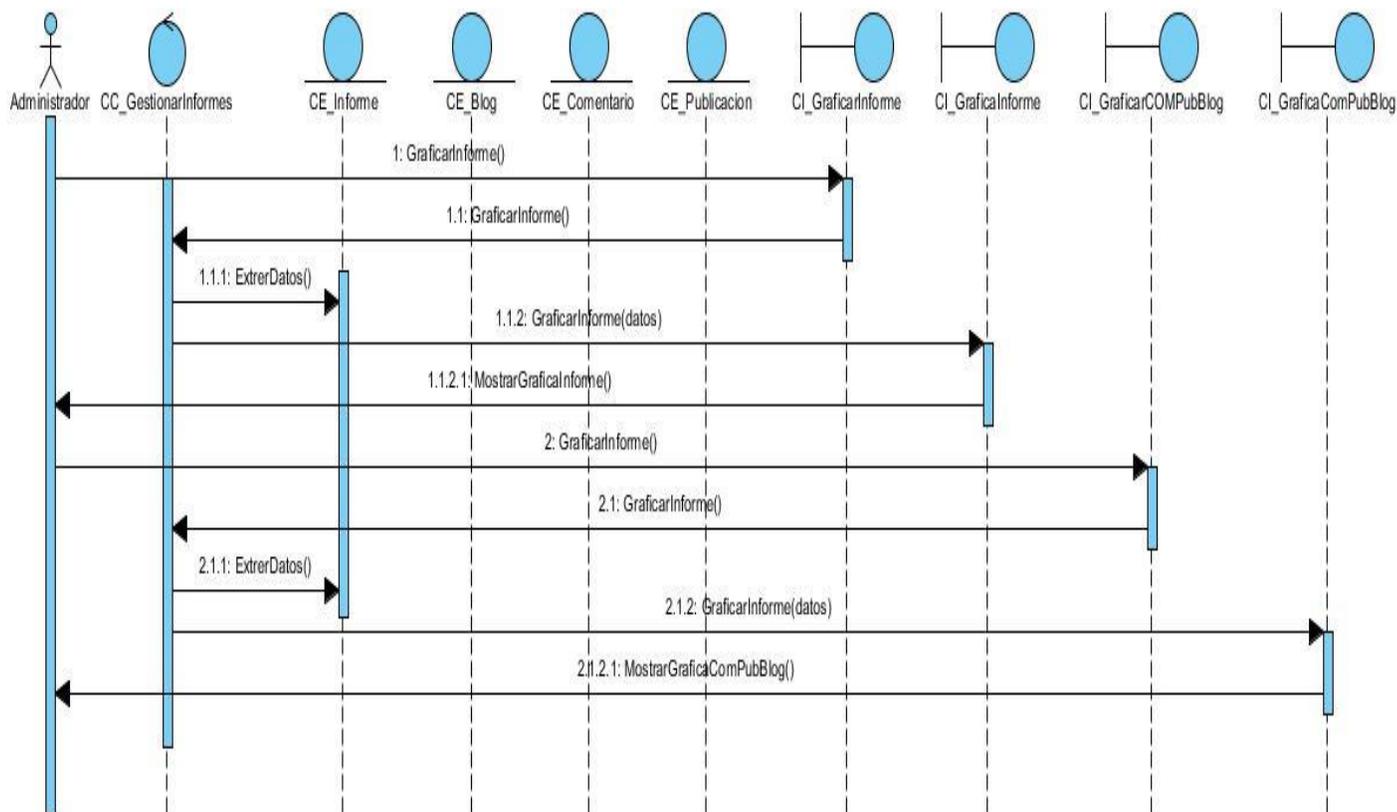
Sección Generar Gráfica de Blog



Sección Generar Gráfica de Comentario

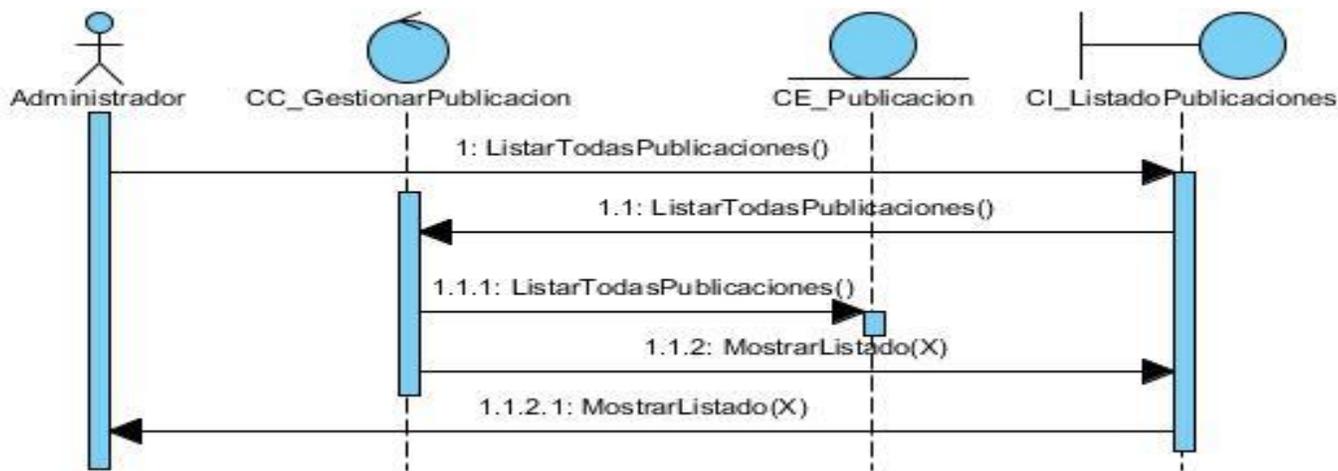


Secciones Generar Gráfica de Informes Estadísticos y Generar Gráfica ComPubBlog

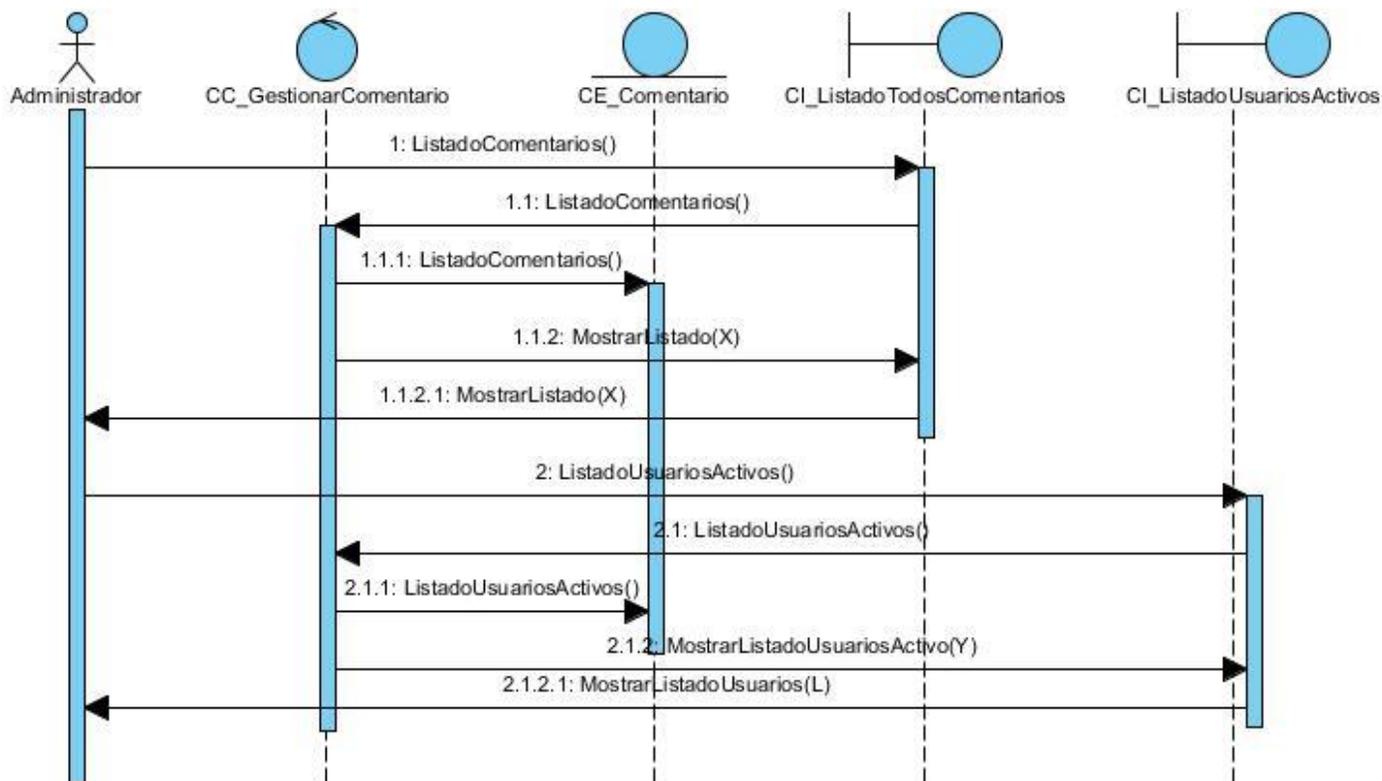


CU Listar Contenidos

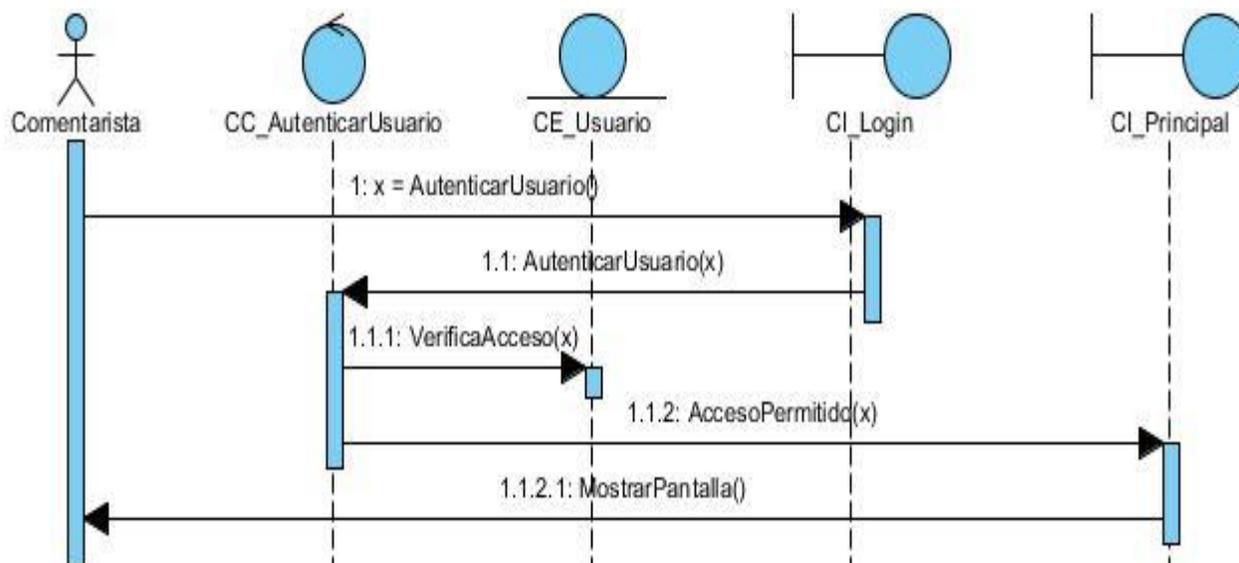
Sección Listar Publicaciones



Secciones Listado de Todos los Comentarios y Listado de los Comentarios de los Usuarios Activos



CU Autenticar Usuario



Anexo D

Interfaces de la aplicación web infoBlog v2.

Interfaz de la autenticación de la aplicación.



The screenshot shows the login page for 'Info_Blog'. At the top, there is a blue header with the text 'Info_Blog'. Below the header is a white login form with a blue border. The form contains the following elements: a title 'Info_Blog', a text input field for 'Usuario', a text input field for 'Contraseña', a checkbox labeled 'Recordar contraseña', and a blue button labeled 'Iniciar Sesión'.

Interfaz de configuración de la aplicación.



Configuración del sistema



Base de Datos



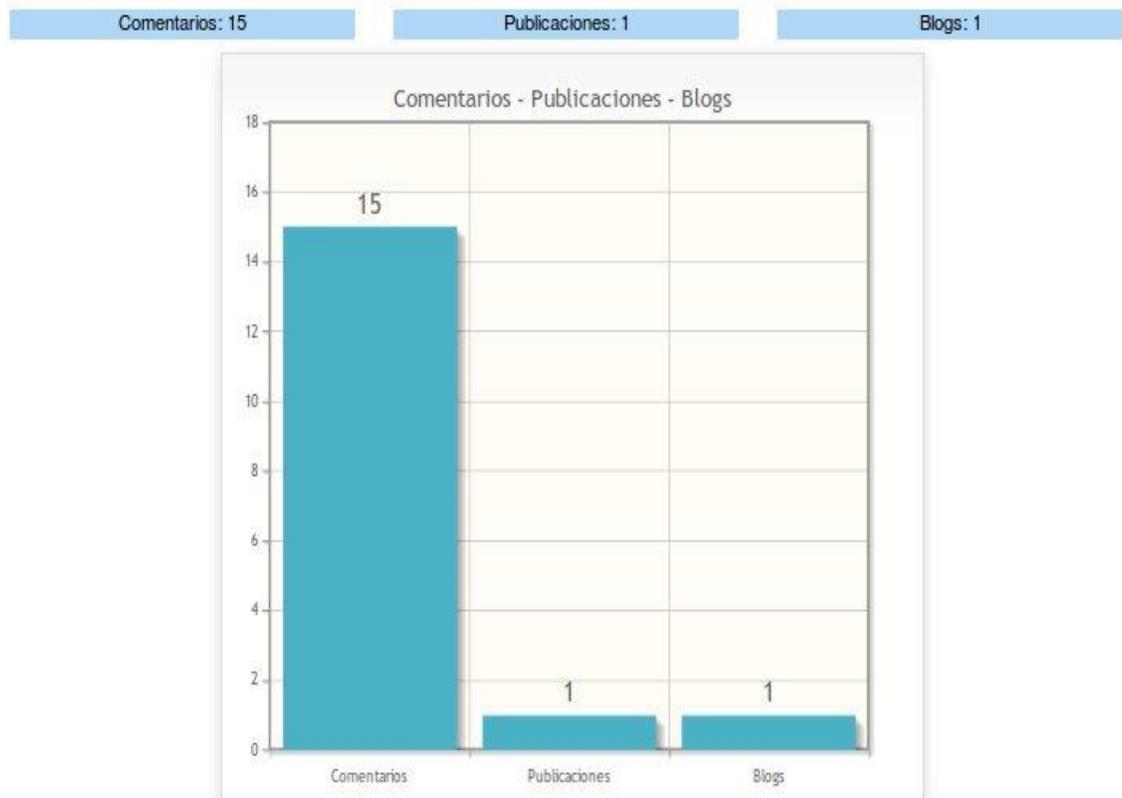
Desactivar Usuario Automáticamente

Interfaz de gráfica de Comentarios, Publicaciones y Blogs.

Info_Blog Bienvenido(a) Nolberto Rojas Pérez  Usuario ▾

Comentarios ▾ Publicaciones ▾ Blogs ▾ Informes ▾ Nomencladores ▾ Gráficas ▾ Listar Contenidos ▾ Administración ▾

Gráfica de Comentarios - Publicaciones - Blogs



Interfaz de gráfica de Comentarios.

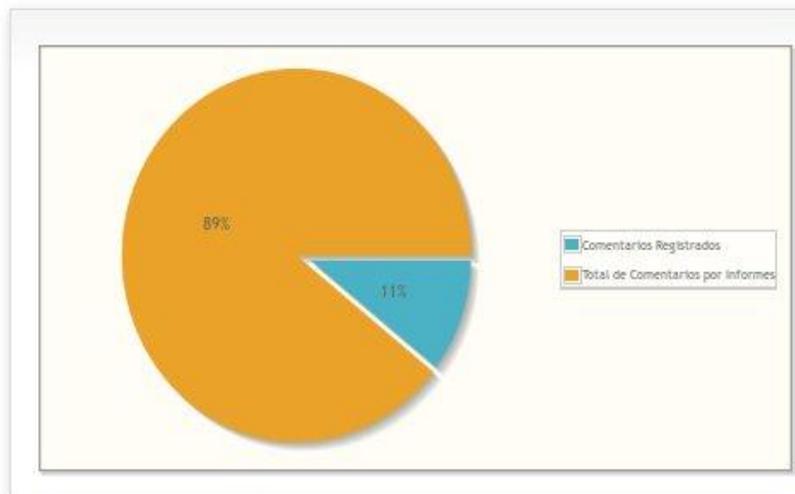
Info_Blog Bienvenido(a) Nolberto Rojas Pérez Usuario -

Comentarios - Publicaciones - Blogs - Informes - Nomencladores - Gráficas - Listar Contenidos - Administración -

Gráfica de Comentarios

Comentarios Registrados: 15

Total de Comentarios por Informes: 125



Cantidad de comentarios por Blogs

Blog	Comentarios
php.uci.cu	13
drupaleros.uci.cu	2

Cantidad de comentarios por Usuarios

Usuario	Comentarios
nrojasp	9
iechevarria	6

Interfaz que muestra los perfiles de usuarios.

Info_Blog Bienvenido(a) Nolberto Rojas Pérez Usuario ▾

Comentarios ▾ Publicaciones ▾ Blogs ▾ Informes ▾ Nomencladores ▾ Gráficas ▾ Listar Contenidos ▾ Administración ▾

Perfil del Usuario: nrojasp

Nombre	Nolberto
Apellidos	Rojas Perez
Solapín	E103628
Usuario	nrojasp
E-mail	nrojasp@estudiantes.uci.cu
Categoría	Estudiante
Facultad	F01A5
Roles	ROLE_ADMINISTRADOR
Cuenta	Activo



Interfaz que muestra la gestión de roles.

Info_Blog Bienvenido(a) Nolberto Rojas Pérez Usuario ▾

Comentarios ▾ Publicaciones ▾ Blogs ▾ Informes ▾ Nomencladores ▾ Gráficas ▾ Listar Contenidos ▾ Administración ▾

Listado de Roles

[+ Nuevo Rol](#)

Mostrar registros Buscar:

Id ▲	Estado	Nombre del Rol	Descripción	Acciones
1	Activo	ROLE_ADMINISTRADOR	Rol responsable de administrar el sistema	
2	Activo	ROLE_EDITOR	Rol responsable de gestionar comentarios y publicaciones	
3	Activo	ROLE_COMENTARISTA	Rol responsable de gestionar comentarios	

Mostrando desde 1 hasta 3 de 3 registros

Anexo E

Entrevista realizada a los clientes.

Preguntas	Respuestas
¿Cómo se realiza el proceso de gestión de la información de la línea Weblog en la actualidad?	
Actualmente existe una aplicación en la línea que realiza esta actividad. ¿Qué funcionalidades de esta aplicación le gustaría que estuviera en la nueva solución?	
¿Qué nuevas funcionalidades quisiera que incorpore la solución?	
¿Qué estructura tendrán los informes que se generarán?	
¿Cómo le gustaría que se presentara la información que se mostrará en las gráficas?	
¿Cómo le gustaría el diseño de las interfaces de la aplicación? Ya sea color de fondo, color de letras, menú principal.	