

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

FACULTAD 9

**Diseño de las nuevas funcionalidades del Módulo
de Redacción de la Plataforma de Televisión
Informativa PRIMICIA.**

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

AUTOR: Lisandra Delgado Cabrera

TUTOR: Ing. Yunior Montaner Hernández.

CONSULTOR: Ing. Rafael Lorente Miranda.

Ciudad Habana, 29 de Junio de 2010.

“Año 52 de la Revolución”

"El secreto del éxito es la constancia en el propósito."

Benjamin Disraeli.

DEDICATORIA

Dedico el resultado de esta investigación, con mucho amor:

A mis padres, ustedes son el mayor amor de mi vida.

A mis tíos María y Mario, que en estos años se han convertido en otros padres para mí.

A toda mi familia, los que tengo cerca y los que no lo están, ojalá estuviéramos todos juntos este día.

A mi novio Yunier, aunque estés lejos siempre estarás en mi corazón.

A todos mis amigos, los de aquí y los de Santiago.

AGRADECIMIENTOS

Agradezco de todo corazón:

A Dios, por iluminar mi camino y escucharme siempre.

A mis padres Sara y Manuel, por el amor, la confianza y el apoyo que me han dado toda la vida.

Ustedes son los mejores padres del mundo. Los amo con todo mi corazón.

A mis tíos María y Mario, por haberme querido siempre y haberme apoyado y ayudado como a una hija en todos estos años de universidad.

A toda mi familia, por haber confiado en mí. Sepan que siempre los tengo presente a pesar de las distancias.

A Yunier, por darme todo su amor, comprensión y apoyo en estos momentos de tanta tensión. Por ser guía de todos mis esfuerzos durante la carrera y por ser ese hombre que tanto admiro y quiero.

A los amigos que han estado conmigo durante toda la universidad, compartiendo cada momento de felicidad o tristeza. Gracias a Katy, Yeny, Yanita, Mariem, Elaine y Liset por haber sido mis mejores amigas en estos años y espero que lo sean para toda la vida.

A mi líder de proyecto, Ing. Rafael Lorente y a los tesisistas de PRIMICIA, mis compañeros en esta lucha; fue un placer compartir con ustedes los momentos de tensión y alegrías que tuvimos durante todo el transcurso de nuestras investigaciones. Ya somos ingenieros.

A mi tribunal, por haber puesto todo su empeño y dedicación en perfeccionar esta investigación hasta convertirla en lo que es hoy.

A Fidel y a la Revolución por haber creado esta Universidad, que ha sido casa y escuela para muchos de nosotros y que nos ha formado como los profesionales en que nos convertimos hoy.

DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor de este trabajo y autorizo a la Facultad 9 de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

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

Lisandra Delgado Cabrera

Ing. Yunior Montaner Hernández

RESUMEN

La televisión constituye un excelente medio de comunicación, pues brinda la posibilidad de difundir las más variadas informaciones en cuestión de minutos, sin importar lo dispersos o alejados que se encuentren los televidentes. Los progresos en la electrónica, la informática y las redes de telecomunicaciones influyeron de forma significativa en el surgimiento y evolución de plataformas de televisión informativa en todo el mundo.

PRIMICIA es una plataforma televisiva creada en Cuba, específicamente en la Universidad de las Ciencias Informáticas. Es un producto informático que permite la creación de noticias que son transmitidas por un canal de televisión. Actualmente el módulo de redacción de PRIMICIA es uno de los más importantes dentro de la plataforma, sin embargo no cuenta con un proceso que permita la configuración o animación de los elementos visuales de la noticia.

En el presente trabajo de diploma se describe cómo es el proceso de redacción en la plataforma hoy en día, además de una serie de funcionalidades que van a permitir realizar la redacción de noticias de forma personalizada. Se analizan soluciones informáticas que cuentan con características semejantes a las que se desean implementar en PRIMICIA y que son utilizadas en Cuba y en el resto del mundo. Además se realiza el diseño de las funcionalidades que se integrarán al módulo de redacción, generando en cada flujo de trabajo los artefactos correspondientes.

PALABRAS CLAVES: Plataforma de televisión informativa, redacción digital, redacción de noticias, elementos configurables.

SUMMARY

Television is an excellent mean of communication, because it offers the possibility of spreading around the world the most varied information in just minutes. Advances in electronics, computers and telecommunications networks significantly influenced the emergence and development of informative television platforms.

PRIMICIA is a television platform created in Cuba, specifically in the University of Informatics Sciences. It is a software product that allows the creation of news transmitted by a television channel. At present, PRIMICIA drafting module is one of the most important within the platform, but it does not have a process that allows the configuration or animation of the visual elements of the news.

This dissertation describes how drafting news process will be customized on the platform. It is made an analysis of informatics solutions that have similar characteristics to those which are about to be implemented in PRIMICIA and which are used in Cuba and the rest of the world. Also the design of the functionalities that are integrated into the drafting module is done, generating in each workflow the related artifacts.

KEYWORDS: Informative television platform, digital drafting, news drafting, configurable items.

ÍNDICE DE TABLAS

<i>Tabla 1: Elementos Configurables. (2).....</i>	<i>11</i>
<i>Tabla 2 Definición de los elementos configurables.....</i>	<i>24</i>
<i>Tabla 3: Requerimientos de Hardware para un (1) servidor.</i>	<i>29</i>
<i>Tabla 4: Requerimientos de hardware para dos servidores.</i>	<i>30</i>
<i>Tabla 5 Descripción textual del CUS Redactar Noticias Personalizadas.....</i>	<i>36</i>
<i>Tabla 6 Descripción textual del CUS Administrar Plantillas.</i>	<i>38</i>
<i>Tabla 7 Descripción textual del CUS Administrar Componente.....</i>	<i>42</i>
<i>Tabla 8 Descripción textual del CUS Administrar Acciones.....</i>	<i>45</i>
<i>Tabla 9 Descripción textual del CUS Administrar Efecto.....</i>	<i>68</i>
<i>Tabla 10: Descripción Textual del CUS Administrar Escena.</i>	<i>70</i>

ÍNDICE DE FIGURAS

<i>Figura 1: Diagrama de clases del dominio.</i>	<i>25</i>
<i>Figura 2 Diagrama de CUS de las nuevas funcionalidades del módulo de redacción de PRIMICIA.....</i>	<i>31</i>
<i>Figura 3: Ventana de configuración del tiempo de duración de la noticia.</i>	<i>35</i>
<i>Figura 4: Ventana de redacción de noticias personalizadas.</i>	<i>35</i>
<i>Figura 5: Ventana de confirmación de la opción Guardar Noticia.....</i>	<i>35</i>
<i>Figura 6: Ventana de confirmación para la opción Salir.</i>	<i>35</i>
<i>Figura 7: Ventana de selección de plantillas.....</i>	<i>37</i>
<i>Figura 8: Ventana de confirmación para la opción Eliminar Plantilla.</i>	<i>38</i>
<i>Figura 9: Ventana para asignación de recursos. (imagen, video y audio).....</i>	<i>41</i>
<i>Figura 10: Ventana de introducción de componentes de textos. (texto, infocinta)</i>	<i>41</i>
<i>Figura 11: Ventana de confirmación de la opción Borrar.</i>	<i>41</i>
<i>Figura 12: Ventana de acciones.</i>	<i>44</i>
<i>Figura 13: Ventana del timeline.</i>	<i>44</i>
<i>Figura 14: Diagrama de clases del análisis del CUS: Redactar Noticia Personalizada. .</i>	<i>47</i>
<i>Figura 15: Diagrama de clases del análisis del CUS: Administrar Plantilla.</i>	<i>47</i>
<i>Figura 16: Diagrama de clases del análisis del CUS: Administrar Acción.</i>	<i>47</i>
<i>Figura 17: Diagrama de clases del análisis CUS: Administrar Componente.</i>	<i>48</i>
<i>Figura 18 La implementación del MVC que realiza Symfony. (9)</i>	<i>52</i>
<i>Figura 19: Relación con los componentes de ExtJS.....</i>	<i>54</i>
<i>Figura 20: Clases relacionadas con los componentes de Symfony.</i>	<i>55</i>
<i>Figura 21: Diagrama de clases del diseño del CUS: Redactar Noticia Personalizada... </i>	<i>56</i>
<i>Figura 22: Diagrama de clases del diseño del CUS: Administrar Plantilla.</i>	<i>56</i>
<i>Figura 23: Diagrama de clases del diseño del CUS: Administrar Componente.....</i>	<i>57</i>
<i>Figura 24: Diagrama de clases del diseño del CUS: Administrar Acción.....</i>	<i>57</i>
<i>Figura 25: Estructura de las clases del modelo generadas por Symfony.</i>	<i>58</i>
<i>Figura 26: Diagrama de clases persistentes.</i>	<i>59</i>
<i>Figura 27: Diagrama entidad relación.</i>	<i>60</i>

<i>Figura 28: Formulario de creación de una escena.</i>	<i>69</i>
<i>Figura 29: Ventana de confirmación de la sección Eliminar Escena.</i>	<i>70</i>
<i>Figura 30: Diagrama de clases del análisis CUS: Administrar Efecto.</i>	<i>70</i>
<i>Figura 31: Diagrama de clases del análisis CUS: Administrar Escena.....</i>	<i>70</i>
<i>Figura 32: Diagrama de clases del diseño del CUS: Administrar Efecto.....</i>	<i>71</i>
<i>Figura 33: Diagrama de clases del diseño del CUS: Administrar Escena.</i>	<i>72</i>
<i>Figura 34: Relación entre redactarNPActions y las clases del modelo.</i>	<i>72</i>
<i>Figura 35: Relación entre admPlantillasActions y las clases del modelo.</i>	<i>72</i>
<i>Figura 36: Relación entre admCompActions y las clases del modelo.....</i>	<i>73</i>

ÍNDICE

Introducción	6
Capítulo I: “Fundamentación Teórica”	10
1.1 Introducción	10
1.2 Conceptos asociados al dominio del problema	10
1.2.1 PRIMICIA	10
1.2.2 Elementos Configurables	11
1.2.3 Teletexto	12
1.2.4 Redacción Digital Integrada	12
1.3 Módulo de redacción en PRIMICIA	12
1.4 Soluciones o sistemas similares existentes a nivel mundial	14
1.4.1 ENEWSROOM	14
1.4.2 Señal ACN	16
1.5 Metodología de desarrollo de software	17
1.6 Lenguaje Unificado de Modelado	18
1.7 Herramienta CASE	19
1.8 Conclusiones	20
Capítulo II: Análisis del Sistema	21
2.1 Introducción	21
2.2 Modelo de Dominio	21
2.2.1 Eventos principales del entorno	22
2.2.2 Glosario de términos del dominio	22
2.2.3 Diagrama de clases del modelo de dominio	25
2.3 Requerimientos	25
2.3.1 Estrategia de Captura de Requisitos	25
2.3.2 Requisitos Funcionales	26
2.3.3 Requisitos no Funcionales	28
2.4 Restricciones en el diseño y la implementación	30
2.5 Definición de los actores del sistema	31

2.6 Diagrama de casos de uso del sistema (CUS).....	31
2.7 Descripción textual de los casos de uso del sistema.....	32
2.7.1 CUS: Redactar Noticia Personalizada.....	32
2.7.2 CUS: Administrar plantillas.....	36
2.7.3 CUS: Administrar Componente.....	38
2.7.4 CUS: Administrar acciones.....	42
2.8 Conclusiones.....	45
Capítulo III: Construcción de la solución propuesta.....	46
3.1 Introducción.....	46
3.2 Modelo de análisis.....	46
3.2.1 Diagramas de clases del análisis.....	46
3.3 Patrones de diseño.....	48
3.3.1 Patrones de Principios Generales para Asignar Responsabilidades (GRASP) que implementa Symfony.....	49
3.3.2 Patrones Gang-of-Four (GOF) que implementa Symfony.....	50
3.4 Modelo de Diseño.....	51
3.4.1 Symfony y ExtJs.....	51
3.4.2 Diagramas de Clases del Diseño.....	53
3.5 Diseño de la base de Datos.....	58
3.5.1 Diagrama de clases persistentes.....	59
3.5.2 Modelo entidad relación.....	59
3.6 Conclusiones.....	60
Conclusiones.....	61
Recomendaciones.....	62
Bibliografía Citada.....	63
Bibliografía Consultada.....	64
Anexos.....	66
Anexo 1.....	66
Anexo 2.....	70
Anexo 3.....	70
Anexo 4.....	72

INTRODUCCIÓN

En una sociedad donde la información ha llegado a ser indispensable, la televisión constituye un medio de comunicación por excelencia. Su trascendencia aumenta día a día a pesar del surgimiento de nuevas tecnologías tales como Internet, que ha demostrado ser un potente vehículo en la comunicación. El hombre en su constante evolución ha vinculado la televisión a otras tecnologías tales como la informática y las telecomunicaciones, con el objetivo de hacer llegar las informaciones con más rapidez, calidad y a un mayor número de personas.

La revolución tecnológica que sacude al planeta hoy en día es inherente a los significativos avances que acontecen con respecto a las tecnologías de la información y las comunicaciones. La introducción de estas tecnologías llega actualmente a todos los aspectos de la vida del hombre, desde lo profesional hasta lo personal. En el mundo de las noticias la digitalización de la información ha traído grandes avances, pues ha permitido realizar la redacción de las noticias con un ambiente más atractivo y se ha logrado la ampliación del acceso a la información, la preservación, conservación, recuperación y la optimización del espacio de almacenamiento físico.

Por la gran importancia de introducir estos conceptos en Cuba se han desarrollado plataformas de televisión informativa en la Universidad de las Ciencias Informáticas (UCI). La facultad No. 9 fue la primera en incursionar en este campo con la creación de **Señal 3**, con el objetivo de llevar a la comunidad universitaria un canal de noticias que los mantuviera informados de forma dirigida y concreta.

Otra plataforma similar es **Señal ACN**, cuyo objetivo principal es hacer llegar noticias y otras informaciones de Cuba y el mundo a los colaboradores internacionalistas que cumplen misión en el extranjero.

Con la creación del Polo Video y Sonido Digital en la facultad 9 surgieron plataformas como **TV Energía**, plataforma creada para el Ministerio de Energía y Petróleo de Venezuela, y **PRIMICIA**, un producto adaptable a las necesidades de cualquier cliente y que hereda las mejores características de las soluciones desarrolladas anteriormente en la facultad 9.

Actualmente la utilización de plataformas de televisión se ha difundido por todo el mundo. Esto se debe a las enormes ventajas que poseen, pues mejoran la gestión de contenidos y archivos

audiovisuales provocando el ascenso de la productividad, producen un valioso ahorro de tiempo y una reducción sustancial de los costes.

En todo el mundo existen soluciones similares a la propuesta realizada en este trabajo de diploma, por ejemplo ENEWSROOM y SONAPS. ENEWSROOM es un sistema de producción de noticias y contenidos audiovisuales diseñado para facilitar el trabajo de periodistas y usuarios (1) y el sistema SONAPS de Sony es un entorno en red que integra todos los aspectos del proceso de producción de informativos, desde la planificación, captación y edición, hasta la emisión, distribución y archivo. (2)

En Cuba no se ha expandido el uso de estas tecnologías con tanta fuerza, sin embargo en la Universidad de las Ciencias Informáticas (UCI) se han desarrollado varias de estas plataformas, pues su comercialización significa un aporte considerable a la economía y el desarrollo del país.

El presente trabajo de diploma consiste en la realización de un informe de investigación práctico del desarrollo del rol de analista dentro del proyecto PRIMICIA¹. En la actualidad la plataforma cuenta con un módulo de redacción de noticias poco flexible, en cuanto a los intereses particulares de cada usuario y el carácter multidisciplinario de las informaciones que se desean transmitir.

La necesidad de hacer el proceso de redacción más personalizado y configurable hace que el presente trabajo de diploma busque darle solución al siguiente **problema científico**: El módulo de redacción de noticias de la Plataforma de Televisión Informativa PRIMICIA del polo Video y Sonido Digital de la Facultad 9 no permite realizar la redacción de forma personalizada.

Para el desarrollo de la investigación se define como **objeto de estudio** el módulo de redacción de PRIMICIA y la propuesta de mejoras del proceso de redacción de noticias de la plataforma. El **campo de acción** se define como el diseño de las nuevas funcionalidades propuestas para mejorar el proceso de redacción de PRIMICIA.

Se propone como **objetivo general** obtener mediante la realización del diseño de las nuevas funcionalidades del módulo de redacción de PRIMICIA, los artefactos y la documentación necesaria para realizar su implementación. Del mismo se derivan los siguientes **objetivos específicos**:

- Describir la propuesta de las nuevas funcionalidades del módulo de redacción de PRIMICIA.
- Caracterizar las soluciones para módulos de redacción configurables que brindan plataformas similares.

¹ El proyecto PRIMICIA es el que desarrolla la plataforma de televisión informativa PRIMICIA.

- Definir el análisis y diseño de las nuevas funcionalidades identificadas para el módulo de redacción de noticias de PRIMICIA.

Como guía de la investigación se plantea la siguiente **idea a defender**: “La realización del análisis y diseño de las nuevas funcionalidades del módulo de redacción de PRIMICIA proveerá los artefactos necesarios para su implementación.”

Para dar cumplimiento al objetivo general se trazaron las siguientes **tareas de la investigación**:

- Caracterizar la propuesta de mejoras del proceso de redacción de noticias de la plataforma de televisión informativa PRIMICIA.
- Caracterizar la existencia de otras aplicaciones o soluciones similares.
- Determinar las metodologías de desarrollo de software a utilizar.
- Caracterizar el lenguaje de modelado y la herramienta CASE a utilizar.
- Identificar los requisitos funcionales y no funcionales del módulo de redacción de noticias de PRIMICIA.
- Desarrollar el prototipo de interfaz de usuario.
- Determinar los patrones de diseño que resulten factibles utilizar.
- Generar los artefactos correspondientes al rol de analista en el desarrollo de las mejoras del proceso de redacción de PRIMICIA.
- Modelar los artefactos correspondientes al rol de analista de sistemas en el desarrollo de las mejoras del proceso de redacción de PRIMICIA.

A lo largo de la investigación se procura obtener información relevante mediante la aplicación de los siguientes **métodos científicos**:

Métodos Teóricos:

Se utiliza el método **analítico-sintético** en el estudio de la información referente al módulo de redacción actual de PRIMICIA, las nuevas funcionalidades que se proponen para mejorar el mismo y las soluciones similares existentes en el mundo.

La **modelación** se emplea en la creación de abstracciones con el objetivo de representar el entorno del problema. En el diseño del sistema se utiliza este método en la realización de los diagramas resultantes de los flujos de trabajo y que sirven de ayuda para comprender mejor el objeto de estudio.

La utilización del método **histórico-lógico** permite la obtención de información sobre el surgimiento, desarrollo y evolución de los módulos de redacción de noticias de las plataformas televisivas y sistemas de teletextos. La información obtenida sirve como apoyo al estudio de las nuevas funcionalidades que se van a integrar a la redacción de noticias en la plataforma PRIMICIA.

CAPÍTULO I: “FUNDAMENTACIÓN TEÓRICA”.

1.1 INTRODUCCIÓN.

El presente capítulo contiene los principales aspectos que conforman la fundamentación teórica de la investigación y que crean una visión de la misma, entre ellos los principales conceptos relacionados con el dominio del problema y la descripción de la situación problemática.

Se realiza un análisis del objeto de estudio, para caracterizar el módulo de redacción de la plataforma televisiva PRIMICIA y las nuevas funcionalidades que se le desean incorporar al mismo. Se analizan algunas plataformas similares para obtener una perspectiva de la utilización de funcionalidades afines a las que se proponen en la presente investigación.

Por último se caracterizan la metodología de desarrollo de software, el lenguaje de modelado y la herramienta CASE que se van a utilizar, poniendo especial atención en las ventajas que ofrecen para el desempeño del rol de analista en el diseño de las nuevas funcionalidades del módulo de redacción de PRIMICIA.

1.2 CONCEPTOS ASOCIADOS AL DOMINIO DEL PROBLEMA.

1.2.1 PRIMICIA.

PRIMICIA, es un producto informático creado en el año 2005 por la Universidad de las Ciencias Informáticas (UCI) en el Polo Productivo “Video y Sonido Digital” de la Facultad 9. Es una solución integral capaz de proveer un canal de televisión para la transmisión automática y constante de informaciones en distintos formatos. Una varios formatos como son: texto, imagen, video y audio, con el objetivo de mostrar noticias con la información editada de una forma más atractiva.

Esta solución puede ser ajustable a cualquier cliente que tenga la necesidad o interés de mantener informados de manera rápida y constante a un grupo significativo de personas. Se consideran clientes potenciales de la plataforma a las agencias de noticias, sedes ministeriales o empresariales, terminales de transporte, hoteles, universidades, centros de convenciones, entre otras.

Está estructurado en dos subsistemas:

- Subsistema de Administración.
- Subsistema de Transmisión.

El subsistema de administración permite la administración del canal y toda la gestión de las noticias y recursos multimedia, mientras que el subsistema de transmisión se encarga de visualizar las noticias y materiales publicados. (1)

1.2.2 Elementos Configurables.

Los elementos configurables son recursos multimedia que se pueden incorporar a las noticias. Son configurables desde la redacción, pues poseen diversas propiedades que pueden ser cambiadas para definir la forma o presencia que el elemento tendrá en la noticia.

Actualmente el módulo de redacción de PRIMICIA posee como recursos multimedia texto, imagen, sonido, video y la combinación de estos formatos. El redactor no tiene la posibilidad de configurarlos durante el proceso de redacción, o sea, no puede realizar cambios en sus propiedades ni asignarle acciones que hagan más dinámica la noticia.

Para las nuevas funcionalidades se propone agregar nuevos recursos multimedia, por lo que los elementos configurables disponibles serían los mostrados en la Tabla 1.

Elementos Configurables					
Imagen	Audio	Clima	Noticia	Tiempo-Restante	Plantilla
Video	Hora	Música	Máscara	Próxima-Noticia	Tabla
Texto	Fecha	Sección	Acción	Próxima-Sección	Infocinta ²

Tabla 1: Elementos Configurables. (2)

² Cintillo informativo que se muestra en las noticias con información para los televidentes.

1.2.3 Teletexto.

El teletexto es un servicio de información que muestra por la televisión las noticias en forma de texto combinado con imágenes, sonidos y videos. Su creación constituye el primer paso para el origen de las actuales plataformas de televisión informativa. El modelo original consistía en un conjunto de páginas que se eligen por el control remoto del televisor y fue utilizado por primera vez en BBC³ en el Reino Unido y más tarde en países como Austria y Alemania. Su principal objetivo era transmitir informaciones, pero también estaba dirigido a prestar servicios a las personas con discapacidad auditiva.

En la actualidad existen variantes del modelo original que ofrecen variados servicios, entre los más comunes están las noticias de última hora, resultados meteorológicos, deportivos, entre otros. Las plataformas de televisión basan su funcionamiento en los teletextos, pero agregan otros recursos para enriquecer las informaciones que transmiten.

1.2.4 Redacción Digital Integrada.

El desarrollo de las TICs⁴ ha influido de forma positiva en el mundo de las noticias. La redacción digital integrada permite capturar los textos, gráficos, fotografías, sonidos y videos de las informaciones y almacenarlas en servidores. Los periodistas y redactores pueden acceder a estos servidores desde su computadora personal para elaborar las noticias, por lo que el montaje y actualización de las noticias publicadas resulta cada vez más rápido.

La implantación de este sistema de redacción es de gran impacto, pues constituye un paso estratégico en la consolidación y la calidad de la producción de noticias. Numerosas televisoras de todo el mundo han adoptado sistemas de redacción digital integrada, como por ejemplo TELECINCO, CNN+ y Antena 3. (3)

1.3 MÓDULO DE REDACCIÓN EN PRIMICIA.

El módulo de redacción de noticias en PRIMICIA es uno de los más importantes dentro del subsistema de administración. En el mismo se realiza la redacción y edición de las informaciones y los recursos multimedia que van a conformar la noticia.

Actualmente el redactor sólo puede realizar dos actividades:

³ British Broadcasting Corporation (siglas de Corporación Británica de Radiodifusión)

⁴ Siglas de Tecnologías de la Informática y las Comunicaciones.

1. Redactar noticias: Proceso de edición de noticias que incluye ponerle un título, ubicarla en una sección temática, elegir el fondo musical y crear las pantallas que la conforman.
2. Gestionar noticias: Proceso que permite visualizar las noticias, modificarlas, eliminarlas y mostrar los cambios realizados a estas.

El módulo de redacción de PRIMICIA no brinda la posibilidad de adaptarse a los variables entornos⁵ en los que se planea usar la plataforma. Esto se debe a que el mismo posee ciertas limitaciones en el diseño de las noticias, pues no permite modificar las propiedades ni la estructura de los recursos multimedia que se emplean en la confección de las mismas.

Por estas razones no es posible la personalización de las noticias y el redactor no puede hacer llegar el mensaje de la manera que realmente desea. Para mejorar el módulo y hacer el proceso de redacción más atractivo y con una calidad superior se propone agregarle las siguientes funcionalidades:

- ❖ Edición de los elementos y recursos multimedia de la noticia.
- ❖ Incorporación de efectos de animación a las noticias.
- ❖ Pre-visualización en todo momento de la edición en curso. (2)

Estas modificaciones traen consigo la necesidad de realizar el diseño de las nuevas funcionalidades para su posterior implementación y puesta en práctica en la plataforma. De la propuesta de mejoras realizada por el Ing. Rafael Lorente Miranda (2) se tomaron las principales acciones que constituyen la guía para los cambios que se pretenden realizar.

Acciones a realizar para mejorar el módulo de redacción de Primicia.

- ❖ Agregar nuevas funcionalidades al módulo de redacción que permitan la confección de noticias que presenten elementos configurables. Las funcionalidades que posee el módulo en la actualidad se mantienen, para satisfacer las necesidades de un redactor poco diestro en la edición de noticias. (2)
- ❖ Crear las ventanas necesarias para la edición de la noticia. Se crearán las ventanas necesarias para el trabajo del editor en la plataforma, estas serán: área de trabajo, ventana de componentes, ventana de la línea de tiempo y ventana de propiedades. (2)
- ❖ Crear todos los componentes y herramientas necesarios para la creación de la noticia. (2)

⁵ Los entornos a los que se hace referencia son: agencias de noticias, sedes ministeriales o empresariales, terminales de transporte, hoteles, universidades, centros de convenciones, entre otras.

- ❖ Crear un banco de efectos, que sean aplicables a todos los elementos configurables y que se puedan configurar desde la ventana de edición. (2)

Es importante analizar algunos sistemas que aportan soluciones similares a la que se pretende desarrollar en esta investigación. El objetivo fundamental de este análisis es obtener experiencias que tributen al buen desarrollo de las nuevas funcionalidades que se incorporarán al módulo de redacción de PRIMICIA.

1.4 SOLUCIONES O SISTEMAS SIMILARES EXISTENTES A NIVEL MUNDIAL.

En el mundo entero se desarrollan plataformas de televisión que usan la redacción digital integrada en diversos procesos que son mejorados continuamente. Algunos ejemplos de sistemas que realizan el proceso de redacción de forma semejante al que se desea incorporar a PRIMICIA son:

1.4.1 ENEWSROOM.

ENEWSROOM es un sistema de producción de noticias y contenidos audiovisuales, de la plataforma propietaria Estructure Media Systems. (4) Integra las herramientas necesarias para la explotación de noticias y la introducción de flujos de trabajo más flexibles que permiten a los usuarios realizar su trabajo con grandes facilidades.

Sus características fundamentales son:

- Sistema sencillo, intuitivo, robusto y fiable. (4)
- Posee servidores multiformato que son capaces de digitalizar simultáneamente hasta tres calidades de vídeo⁶. (4)
- La base del sistema se conecta con otras herramientas externas⁷ de la plataforma Estructure que aportan sus funcionalidades a la redacción. (4)
- Utiliza metadatos personalizables donde los contenidos multimedia pueden ser etiquetados con términos descriptivos pertinentes para que los editores puedan localizarlos de forma rápida. (4)

El sistema cuenta con varios módulos de trabajo, que se agrupan de la siguiente forma:

Módulos de entrada:

⁶ Calidades soportadas: alta, baja y ultrabaja.

⁷ Herramientas externas: editores no lineales, tituladoras, prompter, sistemas de grafismo, etc.

- **Previsiones:** redactores y equipo técnico conforman un equipo para agilizar la entrada de información al sistema. (4)
- **Captura:** asocia los materiales audiovisuales con los metadatos correspondientes en el momento de su entrada en el sistema, clasificándolos para su posterior acceso por parte de los usuarios. (4)

Módulos de Organización:

- **Planificador:** organiza y coordina la producción de noticias, durante el proceso productivo y emisión. (4)
- **Búsqueda:** proporciona la visualización de toda la información que contiene el sistema y la interacción con otros módulos, llevando los contenidos seleccionados a éstos. (4)

Módulos de Producción:

- **Editor de noticias:** permite realizar ediciones del material audiovisual y del texto de la noticia. Selecciona los materiales desde cualquier bruto⁸, edita en el timeline el video, audio y gráficos mientras se desarrolla el texto de la noticia, la sinopsis, o el minutado. Permite la incorporación de voz en off, diferentes modos de edición, detección de planos, escucha rápida, entre otras. (4)
- **Documentación:** documenta noticias y material audiovisual bruto y esta documentación posteriormente es enviada al módulo de archivo. (4)

Módulos de Salida

- **Emisión:** gestiona la emisión a través de una escaleta dinámica controlando el video-servidor de emisión. (4)
- **Archivo:** gestiona el archivo del sistema mediante volcados automáticos o manuales a soportes de seguridad (Cintas, DVD,...), además de codificar indexando de nuevo el material audiovisual con la información contenida en el sistema. (4)
- **Internet:** La codificación a Windows Media File u otros formatos puede realizarse desde los módulos de editnews, compactador o búsqueda, o mediante la selección de un perfil de codificación en el módulo de Internet. (4)

Este sistema resuelve de forma parcial el problema planteado, pero para utilizarlo habría que pagar la licencia a la plataforma ESTRUCTURE y a su vez depender de varias herramientas externas de esta

⁸ El término bruto se refiere a los materiales audiovisuales que pueden ser utilizados en la edición de informaciones.

plataforma que también se distribuyen bajo licencia. Además la redacción en este sistema incluye muchísimas funcionalidades que ya la plataforma PRIMICIA posee y que por tanto no es necesario integrarle al módulo de redacción.

Por lo expresado anteriormente es evidente que el desarrollo de las funcionalidades que se proponen en la presente investigación resulta más factible que integrar ENEWSROOM con PRIMICIA.

1.4.2 Señal ACN

Señal ACN es uno de los principales precedentes de PRIMICIA, por tanto se hace necesario la valoración del proceso de redacción que posee esta plataforma.

El proceso de redacción en ACN se realiza de la siguiente forma:

- ❖ Se obtiene y sintetiza la información procedente de los principales medios de prensa (agencias de noticias, agencias cablegráficas, prensa escrita, etc.). (2)
- ❖ Se selecciona un título para la noticia. (2)
- ❖ Se ubica en la sección temática adecuada (Actualidad, Nacionales, Deportes, Culturales y otras). (2)
- ❖ Se escoge un fondo musical para la misma. (2)
- ❖ Se llena el cuerpo de la noticia creando cada una de las pantallas que la conforman, las cuales pueden ser tipo texto, texto-imagen, imagen y video. Si la pantalla es de tipo imagen o video se le asigna el recurso correspondiente. Cada una de las pantallas tiene una duración específica de acuerdo a la longitud del texto de la misma (10, 15 ó 25 segundos). (2)
- ❖ La conformación de la noticia termina cuando el corrector revisa la misma y enmienda cualquier error cometido por el redactor, teniendo la posibilidad de modificar el título, la sección, el tema musical de fondo y las pantallas de la noticia. (2)

Como se puede apreciar, en Señal ACN es posible el acceso, actualización y presentación de recursos multimedia en las noticias, pero como sucede actualmente en PRIMICIA no se da la posibilidad de configurar los componentes de las mismas. Otra desventaja que posee el módulo de redacción de Señal ACN es que no incorpora los nuevos componentes que se desean integrar a la redacción en PRIMICIA.

Luego de analizar algunos sistemas similares al que se propone en la investigación, se caracterizan las tecnologías utilizadas en el diseño del sistema, señalando en cada caso aspectos relevantes que justifican su elección.

1.5 METODOLOGÍA DE DESARROLLO DE SOFTWARE.

“Un proceso de desarrollo de software define **quién** está haciendo **qué**, **cuándo** y **cómo** para alcanzar un determinado objetivo.” (5)

En muchas ocasiones resulta complejo definir cómo se realizará el proceso de desarrollo de un software. Esto se debe a la necesidad de crear sistemas cada vez más sofisticados y mejor adaptados a las nuevas tecnologías y a las necesidades de los usuarios. Las metodologías de desarrollo de software guían todo el proceso de desarrollo para lograr mejorar los productos de una versión a otra y realizar todo el proceso de una forma más rápida.

Para la elección de una metodología de desarrollo para el proyecto PRIMICIA se realizó un análisis de las metodologías más usadas y conocidas, con el objetivo de determinar cual se ajusta más a sus características.

Se descartaron las metodologías ágiles, como EXTREME PROGRAMMING (XP), AGIL UNIFIED PROCESS (AUP), SCRUM y otras, pues se cuenta con un equipo de trabajo amplio y que no posee una vasta experiencia en el desarrollo de software. Además porque se requiere una buena documentación para garantizar la realización de nuevas versiones con integrantes nuevos en el proyecto.

Se determinó que es necesario usar una metodología robusta, pues estas confieren gran peso a la planificación, conceptualización y descripción del sistema que se va a diseñar. RATIONAL UNIFIED PROCESS (RUP) y MICROSOFT SOLUTION FRAMEWORK (MSF) son dos de las metodologías más conocidas y usadas entre las tradicionales.

La elección de una metodología depende principalmente de las características particulares del proyecto y de la experiencia que tenga el equipo de desarrollo en el uso de éstas. RUP y MSF son metodologías flexibles que pueden ser adaptadas a las necesidades de cada proyecto. Centran su atención en llevar la documentación del proyecto y cumplir con el plan definido para éste.

Para el desarrollo de PRIMICIA se utiliza como sistema operativo Linux, por tanto las tecnologías que se utilizan tienen que ser libres o al menos multiplataforma. MSF es una metodología desarrollada por

Microsoft que requiere el uso de Microsoft Visual Studio Team System en todo el modelo y muchas de las fases. Aunque es posible utilizar otra tecnología esto implicaría mayor complejidad en el proyecto. Con RUP se puede utilizar Visual Paradigm for UML, que es una herramienta multiplataforma que brinda excelentes resultados en el modelado y que ya ha sido utilizada en la modelación de los artefactos del proyecto.

Es importante resaltar que el proyecto PRIMICIA utilizó RUP durante el desarrollo de la primera versión de la plataforma. Se utiliza pues el equipo de desarrollo es joven, inexperto y está compuesto principalmente por estudiantes, esto implica además la posibilidad de que ocurran cambios de rol y la necesidad de incorporar nuevos miembros al proyecto, debido a las bajas por culminación de estudios u otras causas. RUP es la metodología que mejor se adapta a los proyectos a largo plazo como PRIMICIA y la documentación que genera es bien detallada en cuanto a cómo la tarea será ejecutada y los resultados que se deben obtener.

Como conclusiones del análisis se determinó que al ser RUP la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos resulta factible utilizar esta metodología, teniendo en cuenta además los resultados obtenidos del estudio realizado y la experiencia del equipo de trabajo en su uso.

1.6 LENGUAJE UNIFICADO DE MODELADO.

El Lenguaje Unificado de Modelado (UML siglas de Unified Modeling Language) es un lenguaje de modelado visual que se usa para especificar, visualizar, construir y documentar artefactos de un sistema de software. Captura decisiones y conocimiento sobre los sistemas que se deben construir. (6)

El Proceso Unificado utiliza el Lenguaje Unificado de Modelado para preparar todos los esquemas de un sistema de software. De hecho, UML es una parte esencial del Proceso Unificado, sus desarrollos fueron paralelos. (5)

Está pensado para usarse con todos los métodos de desarrollo, etapas del ciclo de vida, dominios de aplicación y medios. El lenguaje de modelado pretende unificar la experiencia pasada sobre técnicas de modelado e incorporar las mejores prácticas actuales en un acercamiento estándar. Pretende dar apoyo a la mayoría de los procesos de desarrollo orientados a objetos. (6)

Para el desarrollo de las nuevas funcionalidades del módulo de redacción de PRIMICIA, UML da la posibilidad de representar gráficamente los modelos de forma que puedan ser comprendidos por todos los desarrolladores. Permite especificar las características que tendrá el sistema, incluso antes de su construcción. A partir de los modelos que se vayan generando se puede construir el sistema diseñado y estos propios elementos gráficos sirven como documentación del sistema desarrollado para futuras revisiones y versiones.

Es un lenguaje libre y totalmente independiente del lenguaje de programación que se utilice. Tiene como fin modelar sistemas en base al paradigma orientado a objetos, por tanto los diseños realizados usando UML se pueden implementar en cualquier lenguaje orientado a objetos.

Para la modelación utilizando UML se han desarrollado varias herramientas CASE (Computer Aided Software Engineering) que sirven de gran ayuda a los desarrolladores de un proyecto.

1.7 HERRAMIENTA CASE.

Las herramientas CASE son aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software, mediante la realización del diseño del proyecto, el cálculo de los costos, la implementación de parte del código automáticamente, la documentación y detección de errores.

Actualmente existen varias herramientas CASE, entre ellas ARGO-UML, POSEIDON for UML, Rational Rose y Visual Paradigm. Lamentablemente la mayoría de estas herramientas requieren una licencia comercial para su uso. Sin embargo para la primera versión de PRIMICIA todos los artefactos se generaron utilizando Visual Paradigm, pues su uso no tiene trascendencia legal ya que no se incorpora, integra ni distribuye con el producto final.

Visual Paradigm es una herramienta multiplataforma pues posee la capacidad de ejecutarse sobre diferentes sistemas operativos. Es muy fácil de usar pues presenta un ambiente gráfico agradable e intuitivo para el usuario. Es muy poderosa pues permite la realización de ingeniería directa e inversa, el modelado de procesos de negocio, de requerimientos, de base de datos y la generación de código PHP. Admite la integración con varios entornos de desarrollo (IDEs). Es una herramienta colaborativa pues soporta múltiples usuarios trabajando sobre el mismo proyecto y a la vez permite realizar el control de versiones, característica que tiene un impacto positivo en proyectos que cuentan con amplios equipos de desarrollo como PRIMICIA. El uso de esta herramienta en el desarrollo de las funcionalidades propuestas en la presente investigación implica además un ahorro de tiempo en

capacitación, pues fue la herramienta CASE utilizada por el equipo de trabajo en el desarrollo de la primera versión de la plataforma.

1.8 CONCLUSIONES

Las nuevas funcionalidades que serán incorporadas a PRIMICIA recogen un poco de los avances de la ciencia y la tecnología que ocurren en el mundo de los canales “todo noticias”. Servirán para lograr que la plataforma sea más agradable para el redactor y proporcione informaciones mejor elaboradas, más actualizadas y que no tengan impedimentos en cuanto a su transmisión, almacenamiento y posterior uso.

El análisis de las nuevas funcionalidades sirve como base teórica para la realización del diseño de las mismas. La utilización de RUP como metodología de desarrollo de software, UML como lenguaje de modelado y Visual Paradigm como herramienta CASE trae consigo la posibilidad de obtener un producto bien documentado en un lenguaje común para todo el equipo de desarrollo.

Se puede concluir que PRIMICIA será un producto más adaptable y flexible una vez que permita la redacción personalizada de las noticias haciendo uso de elementos configurables.

CAPÍTULO II: ANÁLISIS DEL SISTEMA.

2.1 INTRODUCCIÓN.

En este capítulo se realiza el análisis del sistema que se desea construir. Se hace una conceptualización del entorno en un modelo de dominio y se realiza la captura de requisitos para obtener la especificación de los requerimientos funcionales y no funcionales que el sistema debe tener. Se realizan los diagramas de casos de uso del sistema y las descripciones de sus actores y casos de uso.

2.2 MODELO DE DOMINIO.

Las nuevas funcionalidades del módulo de redacción de PRIMICIA provocan que este proceso sea muy variable. Cada redactor puede confeccionar sus noticias de la forma que estime conveniente, por esto no existe un proceso de negocio visible y bien definido para estas funcionalidades. De ahí que no sea posible realizar el modelo de negocio y sólo se realizó modelo de dominio.

Un modelo de dominio captura los tipos más importantes de objetos en el contexto del sistema. Los objetos del dominio representan las "cosas" que existen o los eventos que suceden en el entorno en el que trabaja el sistema. (5)

Es un diccionario visual de términos o conceptos importantes en el dominio. Utiliza la notación UML de diagrama de estructura estática. Se enfoca en la semántica y proporciona un glosario de términos basado en sustantivos.

Para expresar todo el conocimiento que existe sobre las nuevas funcionalidades se desarrolló el siguiente modelo de dominio, que permite a los desarrolladores, usuarios y clientes manejar un lenguaje único y consistente, que posibilita compartir el conocimiento y evitar confusiones.

Éste modelo describe y enlaza los principales conceptos del entorno donde funcionará la plataforma, incluyendo sus principales eventos y el glosario de términos que detalla los conceptos más trascendentales que intervienen en el dominio del problema.

2.2.1 Eventos principales del entorno.

La redacción en la plataforma PRIMICIA comienza cuando el **redactor** le pone el título a la noticia, selecciona la **sección temática** a la que pertenece y selecciona el tipo de redacción que va a realizar (Pre-Elaborada o Personalizada). Si elige redactar una noticia Pre-Elaborada, deberá seleccionar un fondo musical y luego realizará el proceso actual de redacción.

Si elige la opción redactar noticias personalizadas la primera acción que realiza es configurar el tiempo total que va a durar la noticia en pantalla y seleccionar la **plantilla** que desea utilizar. En el área de trabajo se visualizan las regiones de la plantilla y el redactor configura los **componentes** pertenecientes a ésta o adiciona componentes al área de trabajo.

La noticia va a estar conformada por **escenas**, en las que el redactor puede configurar los componentes pertenecientes a cada escena, así como el orden de aparición que tendrán éstas en la noticia. El material que se está redactando podrá ser visualizado en cualquier instante de la edición en curso.

Los **elementos configurables** en cualquier momento pueden ser añadidos, eliminados o editados, realizando cambios en sus **propiedades**. También se les puede asignar **acciones** que definen el comportamiento del elemento en la noticia y a estas acciones se les puede aplicar **efectos**.

2.2.2 Glosario de términos del dominio.

Permisos: Definen las acciones que puede realizar un usuario dentro de la plataforma.

Usuario: Persona que se registra en la plataforma e interactúa con ella para la Administración o Transmisión de contenidos.

Redactor: Usuario registrado que tiene permisos para redactar las noticias que se transmitirán.

Sección Temática: Ubicación general que permite al redactor agrupar las noticias según su contenido.

Time-Line: Se presenta en forma de ventana y se utiliza para sincronizar en el tiempo todos los elementos de la noticia.

Área de trabajo: Se presenta en forma de ventana y se utiliza para insertar los componentes que conformarán la noticia.

Escena: Engloban las acciones presentes en un período de tiempo. Cuando se crea una nueva escena dentro de una noticia se realiza de nuevo el proceso de redacción de noticias personalizadas; pero las diferentes escenas que van creando pertenecen a la noticia que se está redactando.

Elementos Configurables: Componentes visuales o recursos multimedia que se pueden incorporar a las noticias. Éstos poseen propiedades que pueden ser modificadas para personalizar la noticia.

Elemento Configurable	Definición
Imagen	Es un componente que estará referenciando a un archivo de imagen. Sus principales propiedades son: posición, tamaño, visibilidad, superposición, opacidad, identificador y máscara.
Video	Es un componente que estará referenciando a un archivo de video. Sus principales propiedades son: posición, volumen, tamaño, superposición, intervalo, visibilidad, repetición, opacidad e identificador.
Texto	Es un componente que estará referenciando a un texto determinado. Sus principales propiedades son: posición, tamaño, margen, visibilidad, identificador, estilo, fuente, tamaño de fuente y color.
Audio	Es un componente que estará referenciando a un archivo de audio. Sus principales propiedades son: volumen, repetición, título, ícono, intervalo e identificador.
Infocinta	Este componente de la noticia tiene como objetivo mostrar informaciones alternativas durante la transmisión de la noticia como pudiera ser el parte del tiempo o las personas que están dentro de una imagen. Generalmente su aparición es de corta duración. Sus principales propiedades son: estilo (texto, climática, mixta), tipo de infocinta (inferior, superior, izquierdo, derecho), posición, tamaño, forma, texto, imagen e identificador.
Fecha	Este componente representa la fecha del día en que se está ejecutando la aplicación, puede tener un componente texto asociado. Sus principales propiedades son: tamaño, posición, formato, superposición, visibilidad, identificador, texto e idioma.

Hora	Este componente representa la hora en que se está ejecutando la aplicación, puede tener un componente texto asociado. Sus principales propiedades son: tamaño, posición, formato de hora, superposición, visibilidad, identificador, texto, imagen y tipo.
Clima	Este componente brinda informaciones sobre el estado del tiempo de una región determinada. Sus principales propiedades son: lugar, tipo, posición, tamaño, opacidad, superposición, visibilidad e identificador
Tiempo-Restante	Este elemento indica en todo momento el tiempo que falta para terminar la noticia. Sus principales propiedades son: tamaño, posición, superposición, visibilidad y formato.
Plantilla	Este componente es de los más importantes ya que brinda una distribución inicial de los componentes de la noticia, facilitando la edición de la misma. Sus principales propiedades son: nombre, logo (imagen), tipo, fecha, hora y color.
Sección	Este componente indica la sección temática a la que pertenece la noticia mostrada. Sus principales propiedades son: nombre, logo (imagen) y cantidad de noticias.
Próxima- Sección	Este componente indica la sección temática que le sigue a la sección en curso. Su principal propiedad es el nombre de la siguiente sección.
Próxima-Noticia	Indicará el título de la siguiente noticia. Su principal propiedad es el nombre de la siguiente noticia.
Máscara	Este componente se le aplicará a imágenes con el objetivo de visualizar una parte de la misma. Sus principales propiedades son: forma, posición, expansión y desenfocado.
Tabla	Este componente posibilitará la presentación de las noticias que requieran de tablas. Sus principales propiedades son: número de filas, número de columnas, bordes y sombreado.

Tabla 2 Definición de los elementos configurables.

Acciones: Los elementos configurables actúan en la noticia realizando acciones tales como mostrar, ocultar, desplazar, etc.

Efectos: Efectos de animación que se pueden aplicar a las acciones, tales como disolver, disolver-mover (ghost) y escalar.

2.2.3 Diagrama de clases del modelo de dominio.

En la figura 1 se muestran los principales conceptos asociados al dominio del problema y las relaciones que se establecen entre éstos.

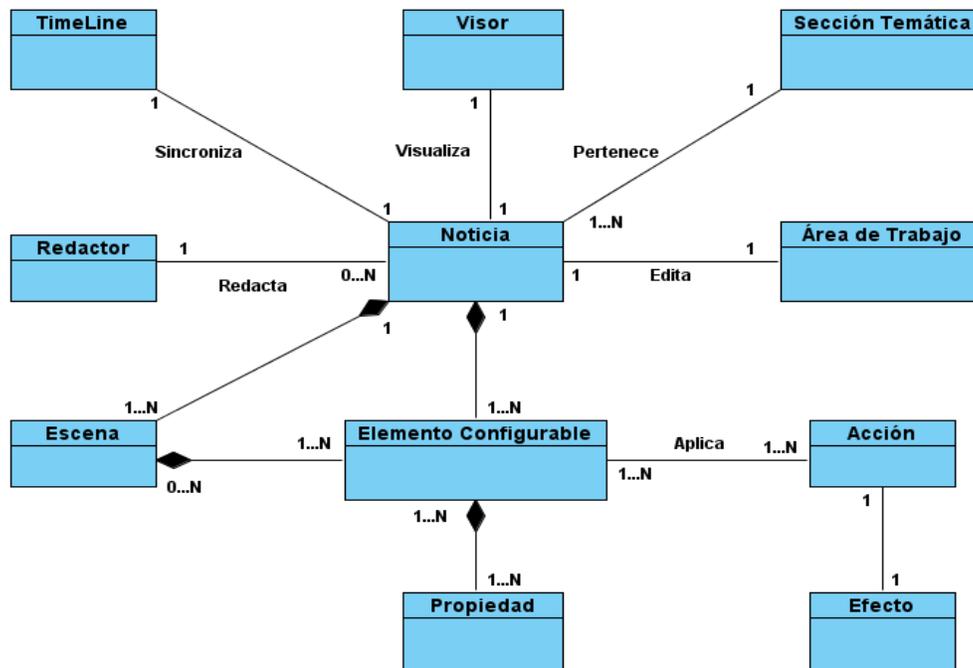


Figura 1: Diagrama de clases del dominio.

2.3 REQUERIMIENTOS.

Durante el proceso de captura de requisitos el propósito fundamental del equipo de desarrollo es guiar el desarrollo hacia el sistema que desea y necesita el cliente. Esto se consigue mediante una descripción de los requisitos funcionales y no funcionales que el sistema debe cumplir. La misma debe ser lo suficientemente buena, como para que pueda llegarse a un acuerdo entre clientes, usuarios y desarrolladores. Este proceso puede resultar complejo en productos adaptables como PRIMICIA, donde la captura de requisitos se realiza únicamente con los desarrolladores y el entorno de trabajo está en dependencia del cliente que se interese.

2.3.1 Estrategia de Captura de Requisitos.

Para realizar la captura de requisitos se propone utilizar algunas tareas de las planteadas en el modelo de Durán. Este es un modelo realizado por el Dr. Amador Durán Toro en su tesis doctoral “Un entorno metodológico de ingeniería de requisitos para sistemas de información” (7). En él se propone el uso de una serie de técnicas, tareas y plantillas durante las actividades de negociación, análisis y validación de los requerimientos que posibiliten obtener requisitos claros, consistentes y sin ambigüedades.

La presente investigación tiene como principal antecedente la conceptualización de mejoras del proceso de redacción de PRIMICIA, en la cual se realizaron varias de las tareas propuestas por el Dr. Durán para realizar la captura de requisitos. Como parte de la investigación actual se decidió realizar las siguientes tareas para obtener y refinar los requerimientos funcionales y no funcionales:

- Obtener información sobre el dominio del problema y el sistema actual.
- Identificar los requisitos funcionales y no funcionales, los actores y casos de uso del sistema a desarrollar.
- Revisar, en el caso de que haya conflictos, los requisitos previamente identificados.
- Priorizar los requisitos y casos de uso.

La plataforma constituye un producto adaptable y que puede ser personalizado a cualquier interesado, por esta razón en la captura de requisitos no intervino ningún cliente. Las tareas antes mencionadas se realizaron a través de talleres de proyecto con los desarrolladores de PRIMICIA, aplicando la técnica de tormenta de ideas. Como principal resultado se obtuvo una versión del listado de requisitos que luego fue refinada en otros talleres del proyecto. De esta forma se verificó que no hubiera omisión de ninguna funcionalidad y que los requisitos fueran consistentes.

2.3.2 Requisitos Funcionales.

RF 1: Redactar noticia personalizada. El sistema debe permitir redactar noticias haciendo uso de los elementos configurables definidos.

RF 1.1: Visualizar región de plantillas. Debe mostrar una ventana para que el usuario escoja la plantilla que desea utilizar.

RF 1.2: Visualizar región de componentes. Debe mostrar una ventana con los componentes que podrá utilizar el usuario en la redacción de la noticia.

RF 1.3: Visualizar ventana de propiedades. Cuando se escoja un componente debe mostrar una ventana con las propiedades del mismo.

RF 1.4: Visualizar ventana de acciones. Cuando se escoja un componente debe mostrar una ventana con las acciones que se le pueden aplicar, así como los efectos que se pueden aplicar a las acciones.

RF 1.5: Visualizar área de trabajo. Debe mostrar el área donde el redactor incluirá los componentes para la confección de la noticia.

RF 1.6: Visualizar el timeline. Debe mostrar la ventana del timeline.

RF 1.7: Guardar noticia. Debe permitir guardar en la base de datos los cambios realizados durante la confección de la noticia.

RF 1.8: Previsualizar noticia: El sistema debe permitir la pre visualización de la noticia en cualquier momento de la redacción.

RF 2: Configurar timeline. El sistema debe permitir configurar el tiempo de duración total de la noticia y el instante de tiempo en el que ocurrirá una acción.

RF 2.2: Visualizar acciones y efectos: Visualizar las acciones y efectos en los canales.

RF 3: Administrar componentes. El sistema debe permitir administrar los componentes que desea incluir en la noticia.

RF 3.1 Agregar componente. Debe permitir seleccionar un componente y desplazarlo hacia el área de trabajo, donde se visualizará con las propiedades predefinidas para este.

RF 3.2 Eliminar componente. Debe permitir eliminar un componente que haya sido agregado al área de trabajo.

RF 3.3 Ajustar propiedades de componente. Debe permitir realizar cambios en las propiedades predefinidas para cada componente.

RF 4: Administrar plantillas. El sistema debe permitir utilizar las plantillas definidas y eliminar su uso en la noticia.

RF 4.1: Seleccionar plantillas. Debe permitir seleccionar la plantilla que se va a visualizar en el área de trabajo.

RF 4.2 Eliminar plantillas. Debe permitir eliminar del área de trabajo la plantilla seleccionada.

RF 5. Administrar acción. El sistema debe permitir administrar las acciones que se le pueden aplicar a un componente.

RF 5.1: Seleccionar acción. Debe permitir aplicar una acción a un componente seleccionado.

RF 5.2: Eliminar acción. Debe permitir eliminar la acción aplicada a un componente.

RF 6. Administrar efecto. El sistema debe permitir gestionar los efectos aplicados a cada acción.

RF 6.1: Seleccionar efecto. Debe permitir aplicar un efecto para la acción seleccionada.

RF 6.2: Eliminar efecto. Debe permitir eliminar el efecto aplicado a una acción.

RF 6.3: Modificar propiedades de efecto. Debe permitir configurar las propiedades del efecto aplicado.

RF 7. Administrar escena. El sistema debe permitir gestionar las escenas que integran la noticia.

RF 7.1: Agregar escena. Debe permitir agregar escenas a la noticia que se está redactando.

RF 7.2: Eliminar escena. Debe permitir eliminar las escenas que se agreguen a una noticia.

RF 8. Asignar recursos. El sistema debe permitir asignar recursos multimedia a los componentes durante la edición de la noticia.

2.3.3 Requisitos no Funcionales.

2.3.3.1 SOFTWARE.

❖ Para la publicación de la aplicación se debe utilizar como servidor web el “Apache”.

2.3.3.2 HARDWARE.

La plataforma debe tener los siguientes requerimientos de hardware, separados en mínimos (azul claro) y óptimos (blanco):

Variante de un (1) servidor:

Servidor	Procesador	Memoria RAM	Disco Duro	Tarjeta de Red	Tarjeta de Video
Administración - Transmisión	Pentium IV 2.8 GHz	1 Gb	120 Gb	Ethernet 10/100 Mbps	Capturadora Hauppauge WinTV PVR 350 Exportadora ATI Radeon X300
	Dual-Core Xeon 2.33 GHz	4 Gb	500 Gb	Ethernet 10/100 Mbps	

Tabla 3: Requerimientos de Hardware para un (1) servidor.

Variante de dos (2) servidores:

Servidor	Procesador	Memoria RAM	Disco Duro	Tarjeta de Red	Tarjeta de Video
Administración	Pentium IV 2.8 GHz	1 Gb	120 Gb	2 puertos Ethernet 10/100 Mbps	Capturadora Hauppauge WinTV PVR 350
	Dual-Core Xeon 1.60 GHz	4 Gb	500 Gb	2 puertos Ethernet Gigabit	
Transmisión	Pentium IV 2.8 GHz	512 Mb	40 Gb	Ethernet 10/100 Mbps	Exportadora

	Dual-Core Xeon 1.60 Ghz	1 Gb	80 Gb	Ethernet Gigabit	ATI X300	Radeon
--	-------------------------------	------	-------	---------------------	-------------	--------

Tabla 4: Requerimientos de hardware para dos servidores.

2.3.3.3 APARIENCIA.

El sistema debe presentar una interfaz sencilla, amigable, intuitiva y de fácil comprensión para el usuario, que facilite su interacción con el sistema.

2.3.3.4 SEGURIDAD.

El sistema debe permitir la interacción con el módulo de redacción sólo a los usuarios registrados que tengan los permisos necesarios.

2.3.3.5 DISPONIBILIDAD

El sistema y los recursos que se emplearán en la confección de las noticias deben estar operativos las 24 horas del día, es decir, un total de 8760 horas al año. Considerando el tiempo que el sistema puede no estar accesible por un error de hardware, de software, de la red o de cualquier otro factor, el por ciento del tiempo que debe estar disponible para los usuarios es de un 99.99% del tiempo total.

La fórmula utilizada para obtener este resultado es la siguiente:

$$\text{Disponibilidad} = \text{tiempo de actividad} / (\text{tiempo de actividad} + \text{tiempo de inactividad}) * 100\%$$

Donde:

El tiempo de actividad es 8 760 horas y el tiempo de inactividad es aproximadamente 0.16 horas (10 min).

2.3.3.6 USABILIDAD.

El sistema debe poseer funcionalidades fáciles de entender y que muestren la información de forma lógica y bien estructurada.

2.4 RESTRICCIONES EN EL DISEÑO Y LA IMPLEMENTACIÓN.

Se definen las siguientes restricciones en el diseño y la implementación de las funcionalidades:

- ❖ El diseño y la implementación de una arquitectura flexible debe permitir la fácil integración o desintegración de componentes.
- ❖ La arquitectura debe soportar migrar la interfaz de usuario de forma rápida.
- ❖ Se requiere el uso de la arquitectura Modelo-Vista-Controlador (MVC) pues se va a utilizar Symfony para el desarrollo de la aplicación.
- ❖ Se deben usar como lenguajes de programación PHP y JavaScript, utilizando como IDE de desarrollo Eclipse.

2.5 DEFINICIÓN DE LOS ACTORES DEL SISTEMA.

Como parte del sistema se identificó como único actor al redactor. Este rol es uno de los más importantes dentro de la plataforma, pues de su trabajo depende todo el contenido que se va a mostrar en el canal.

Redactor: Usuario encargado de redactar las noticias que van a ser transmitidas en el canal.

2.6 DIAGRAMA DE CASOS DE USO DEL SISTEMA (CUS).

En la figura 2 se muestra la relación del actor con los CUS identificados durante la realización del modelo de sistema.

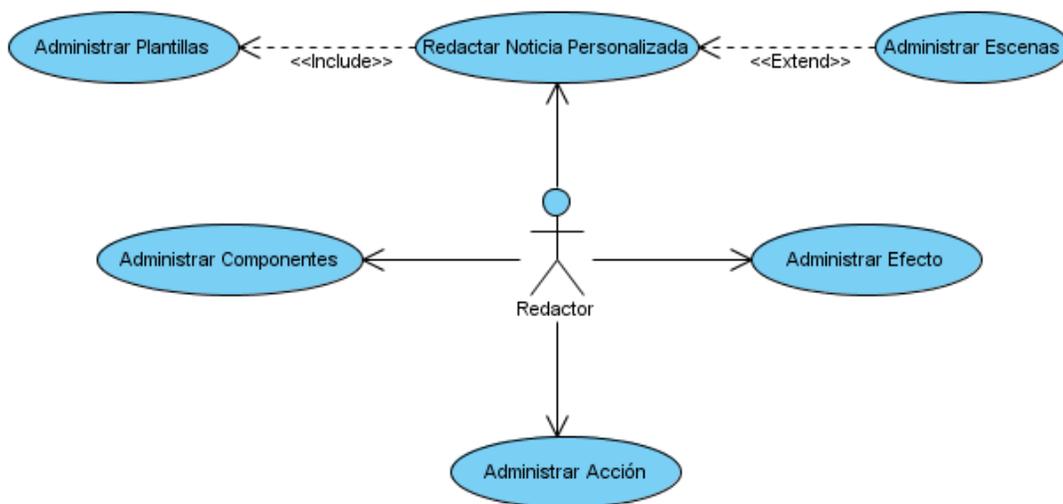


Figura 2 Diagrama de CUS de las nuevas funcionalidades del módulo de redacción de PRIMICIA.

2.7 DESCRIPCIÓN TEXTUAL DE LOS CASOS DE USO DEL SISTEMA.

Los CUS identificados se describen a continuación para ofrecer un mejor entendimiento de las funcionalidades que realizan.

2.7.1 CUS: Redactar Noticia Personalizada.

Caso de Uso:	Redactar noticia personalizada
Actores:	Redactor.
Resumen:	El CUS se inicia cuando el redactor selecciona la opción redactar noticia personalizada, el sistema muestra las ventanas y los elementos configurables que podrá escoger y configurar el redactor. El caso de uso termina cuando el redactor selecciona la opción “guardar noticia”.
Precondiciones:	El redactor debe estar autenticado en la aplicación y tener los permisos de redacción.
Referencias:	RF 1, RF 1.1, RF 1.2, RF 1.3, RF 1.4, RF 1.5, RF 1.6, RF 1.7, RF 2.
Prioridad	Crítico.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El redactor selecciona la opción redactar noticia personalizada.	2. El sistema muestra una ventana para la introducción del tiempo de duración total de la noticia.
3. El redactor introduce la duración de la noticia.	4. El sistema muestra la barra de tiempo de la ventana del timeline configurada para el tiempo de duración introducido.

	<p>5. Se visualiza el área de trabajo.</p> <p>6. Se visualiza la ventana de componentes.</p> <p>7. Se visualiza la ventana de propiedades con la pestaña de propiedades y la de acciones en estado inactivo.</p> <p>8. Se visualiza la ventana de escenas.</p> <p>9. Se visualiza la ventana de plantillas. Ir a CUS Administrar plantillas.</p>
<p>10. El redactor puede realizar las siguientes acciones:</p> <p>a) Agregar una escena a la noticia. Ir a CUS Administrar Escena</p> <p>b) Seleccionar la opción Pre Visualizar noticia (Ubicada en el Menú Archivo). Ir a Sección Pre Visualizar Noticia.</p> <p>c) Seleccionar la opción Guardar Noticia (Ubicada en el Menú Archivo).</p> <p>d) Seleccionar la opción Salir (Ubicada en el Menú Archivo).</p> <p>e) Seleccionar la opción configurar timeline. Ir a Sección Configurar Timeline.</p>	<p>11. Si el redactor realiza la acción:</p> <p>c) El sistema muestra en un mensaje de confirmación.</p> <p>d) El sistema muestra en un mensaje de confirmación.</p>
<p>12. c) El redactor confirma que</p>	<p>13. c) El sistema guarda la noticia redactada en la</p>

<p>desea guardar la noticia.</p> <p>d) El redactor confirma que desea salir de la redacción de noticias.</p>	<p>base de datos.</p> <p>d) El sistema sale de la redacción de noticias personalizadas sin guardar los cambios realizados.</p>
<p>Flujos Alternos de Eventos</p>	
<p>12. c) El redactor cancela la creación de la noticia.</p> <p>d) El redactor cancela la salida de la redacción de noticias.</p>	<p>13. El sistema muestra el área de trabajo con los elementos correspondientes a la noticia que se está redactando.</p>
<p>Sección Configurar Timeline.</p>	
<p>1. El redactor selecciona configurar duración.</p>	<p>2. El sistema muestra una ventana para introducir la nueva duración.</p>
<p>3. El redactor introduce el nuevo tiempo de duración de la noticia.</p>	<p>4. El sistema muestra el timeline configurado con la nueva duración.</p>
<p>Sección Pre Visualizar Noticia.</p>	
	<p>1. Se visualiza la noticia con los cambios realizados por el redactor hasta el momento en una ventana emergente.</p>
<p>2. El redactor cierra la ventana de pre visualización.</p>	<p>3. El sistema muestra el área de trabajo con los elementos correspondientes a la noticia que se está redactando.</p>

Prototipos de Interfaz.

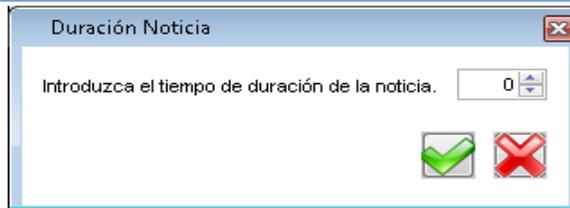


Figura 3: Ventana de configuración del tiempo de duración de la noticia.

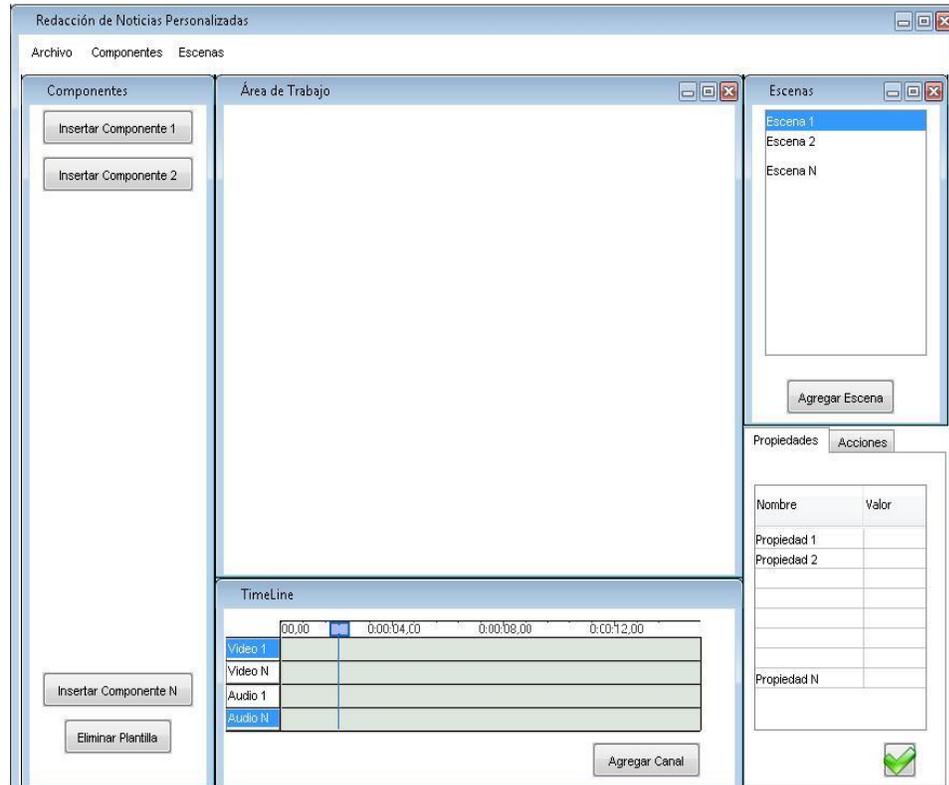


Figura 4: Ventana de redacción de noticias personalizadas.

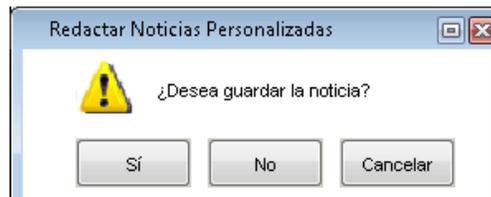


Figura 5: Ventana de confirmación de la opción Guardar Noticia.

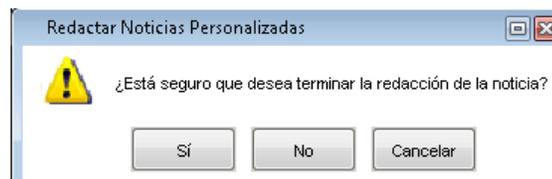


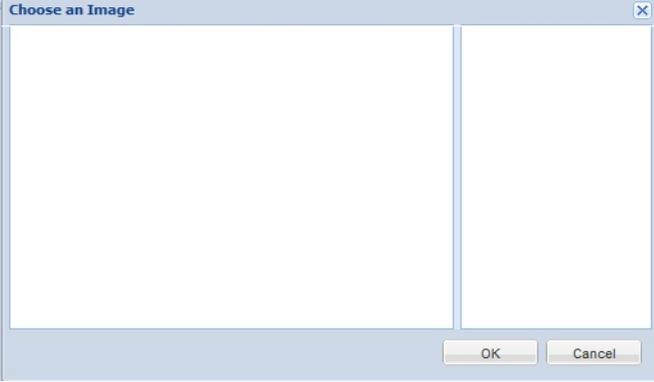
Figura 6: Ventana de confirmación para la opción Salir.

Pos condiciones:	Noticia personalizada guardada en la base de datos.
-------------------------	---

Tabla 5 Descripción textual del CUS Redactar Noticias Personalizadas.

2.7.2 CUS: Administrar plantillas.

Caso de Uso:	Administrar plantillas.
Actores:	Redactor.
Propósito:	Permite al redactor seleccionar la plantilla que desea utilizar, asignarle recursos y eliminarla del área de trabajo.
Resumen:	El CUS se inicia cuando el redactor necesita utilizar las plantillas predefinidas en la plataforma. Selecciona una plantilla, ésta visualiza en el área de trabajo los componentes que la integran. El actor puede asignarle un recurso a cada componente o eliminar la plantilla del área de trabajo para utilizar otra.
Referencias:	RF 4, RF 4.1, RF 4.2, RF 8.
Precondiciones:	El redactor debe estar autenticado en la aplicación y tener los permisos de redacción.
Prioridad	Alta.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El redactor selecciona la plantilla que desea utilizar. 2. Presiona el botón OK.	3. El sistema muestra en el área de trabajo los elementos que conforman la plantilla seleccionada.

<p>4. El redactor puede realizar las siguientes acciones:</p> <p>a) Eliminar del área de trabajo la plantilla que está usando. Ir a Sección Eliminar Plantilla.</p>	
<p>Prototipo de Interfaz</p>	 <p style="text-align: center;">Figura 7: Ventana de selección de plantillas.</p>
<p>Sección Eliminar Plantilla.</p>	
<p>1. El redactor presiona el botón Eliminar Plantilla.</p>	<p>2. El sistema muestra un mensaje para confirmar que el actor desea eliminar del área de trabajo la plantilla que está usando.</p>
<p>3. El redactor confirma que desea eliminar la plantilla.</p>	<p>4. El sistema elimina del área de trabajo la plantilla que estaba visualizándose.</p> <p>5. El sistema visualiza la ventana de plantillas. Ir a la acción 1 del flujo normal de eventos.</p>

Flujo Alternativo de Eventos	
2. El redactor cancela la eliminación de la plantilla.	3. El sistema envía a la acción 3 del flujo normal de eventos.
Prototipo de Interfaz	 <p>Figura 8: Ventana de confirmación para la opción Eliminar Plantilla.</p>
Pos condiciones:	Área de trabajo actualizada, mostrando la plantilla escogida y los cambios realizados a la misma.

Tabla 6 Descripción textual del CUS Administrar Plantillas.

2.7.3 CUS: Administrar Componente.

Caso de Uso:	Administrar Componente.
Actores:	Redactor.
Propósito:	Permite al redactor seleccionar componentes, así como eliminarlos, configurar sus propiedades, asignarles recursos y aplicarles acciones y efectos.
Resumen:	El CUS se inicia cuando el redactor selecciona un componente que se visualizará en el área de trabajo. Luego puede realizar cambios en las propiedades del mismo, asignarle recursos, aplicarle acciones y efectos y eliminarlos del área de trabajo.
Referencias:	RF 3, RF 3.1, RF 3.2, RF 8.
Precondiciones:	El redactor debe estar autenticado en la aplicación y tener los permisos

	de redacción. CUS Administrar Plantillas.
Prioridad	Alta.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El redactor selecciona el componente que va a utilizar.	2. El sistema visualiza la ventana de asignación de recursos al componente.
3. El redactor asigna un recurso al componente agregado según su tipo.	4. El sistema visualiza en el área de trabajo el recurso asignado.
5. El redactor selecciona el recurso y lo desplaza hacia la ubicación que desee.	6. El sistema visualiza el recurso en el lugar seleccionado.
7. El redactor puede realizar las siguientes acciones: a) Eliminar componente. Ir a Sección Eliminar Componente. b) Ajustar propiedades del componente. Ir a Sección Ajustar Propiedades de Componente. c) Aplicar acciones al componente. Ir a CUS Administrar Acciones.	
Sección Eliminar Componente.	

<p>1. El redactor selecciona en el área de trabajo el componente que va a eliminar.</p> <p>2. Presiona el clic derecho del mouse y elige la opción borrar.</p>	<p>3. El sistema muestra un mensaje para confirmar que se desea eliminar el componente.</p>
<p>4. El redactor confirma que desea eliminar el componente.</p>	<p>5. El sistema borra el componente del área de trabajo.</p>
<p>Flujos Alternos de Eventos</p>	
<p>4. El redactor cancela la eliminación del componente. Ir a la acción 7 del flujo normal de eventos.</p>	
<p>Sección Configurar Propiedades de Componente</p>	
<p>1. El redactor selecciona en el área de trabajo el componente al que va a ajustarle las propiedades.</p>	<p>2. El sistema activa la pestaña de propiedades, mostrando las propiedades del componente seleccionado.</p>
<p>3. El redactor introduce los valores de las propiedades que desea cambiar.</p>	<p>4. El sistema muestra el recurso con las nuevas propiedades.</p>

Prototipo de Interfaz

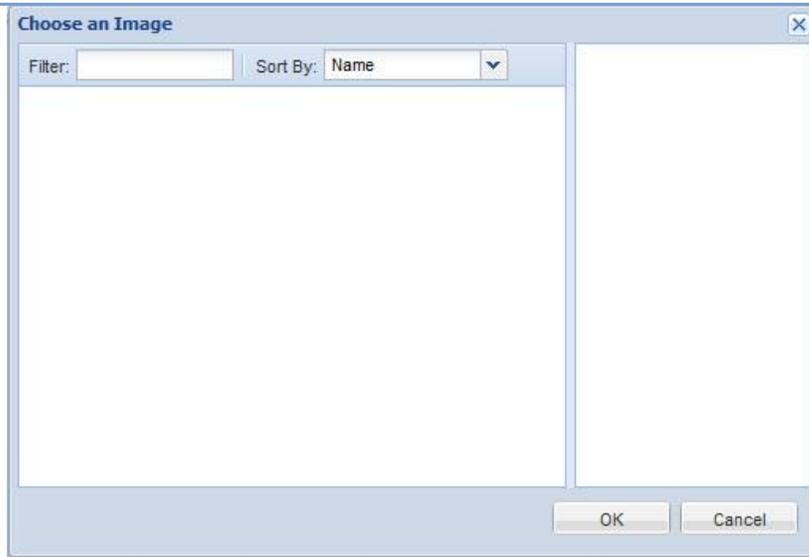


Figura 9: Ventana para asignación de recursos. (imagen, video y audio)

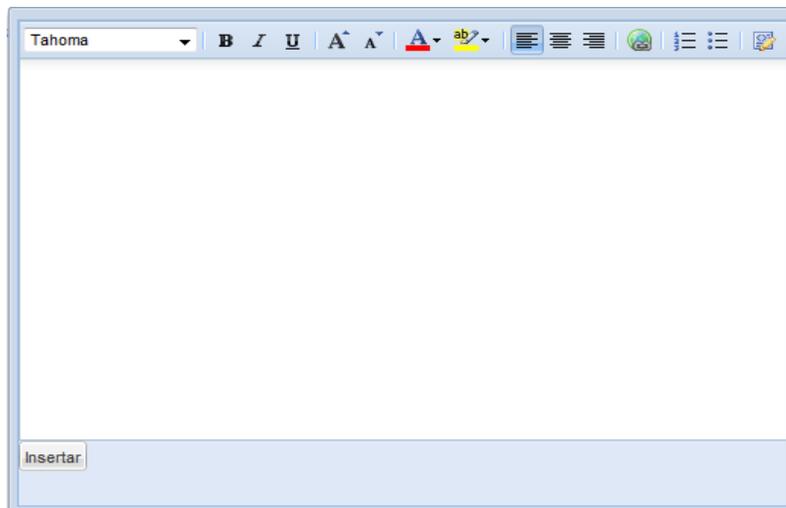


Figura 10: Ventana de introducción de componentes de textos. (texto, infocinta)



Figura 11: Ventana de confirmación de la opción Borrar.

Pos

1. Componente visualizado en el área de trabajo, con los cambios

condiciones:	<p>realizados en sus propiedades.</p> <p>2. Componente eliminado del área de trabajo.</p> <p>3. Propiedades del componente ajustadas.</p>
---------------------	---

Tabla 7 Descripción textual del CUS Administrar Componente.

2.7.4 CUS: Administrar acciones.

Caso de Uso:	Administrar acciones.
Actores:	Redactor.
Propósito:	Permite al redactor aplicar y configurar una acción a un componente determinado.
Resumen:	El CUS inicia cuando el redactor selecciona un componente y le aplica una acción. Permite luego asignarle un efecto a la acción asignada, eliminarla o configurarla en el timeline.
Referencias:	RF 2, RF 2.1, RF 5, RF 5.1, RF 5.2.
Precondiciones:	El redactor debe estar autenticado en la aplicación y tener los permisos de redacción.
	CUS Administrar Componentes.
Prioridad	Alta.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
<p>1. El redactor selecciona en el área de trabajo el componente al que desea aplicarle una acción.</p> <p>2. Selecciona en la ventana de</p>	<p>3. Se activa la pestaña de acciones mostrando las acciones definidas.</p>

<p>timeline el instante de tiempo en el que va a ocurrir la acción.</p>	
<p>4. El redactor selecciona la acción que desea aplicar.</p> <p>5. Oprime el botón Aceptar.</p>	<p>6. El sistema muestra en el canal video la acción que se le aplicó a un componente.</p> <p>7. En caso de que varias acciones coincidan en el mismo instante de tiempo, el sistema crea un canal para cada una.</p>
<p>8. El redactor puede:</p> <p>a) Asignarle un efecto a la acción seleccionada. Ir a CUS Administrar Efectos.</p> <p>b) Eliminar la acción aplicada. Ir a Sección Eliminar Acción.</p> <p>c) Configurar la acción en el timeline. Ir a Sección Configurar timeline.</p>	
<p>Sección Eliminar Acción.</p>	
<p>1. El redactor selecciona el componente al que desea eliminar la acción aplicada.</p>	<p>2. El sistema marca en el timeline las acciones asignadas al componente.</p>
<p>3. El redactor elige la acción que desea eliminar.</p> <p>4. Presiona el clic derecho del mouse y selecciona la opción Eliminar.</p>	<p>5. El sistema muestra un mensaje para confirmar que desea eliminar la acción.</p>
<p>6. El redactor confirma que desea eliminar la acción.</p>	<p>7. El sistema elimina la acción asignada al componente seleccionado. En caso de tener un</p>

	efecto asignado a la acción eliminada, éste también se eliminaría.
--	--

Flujos Alternos de Eventos

6. El redactor cancela la eliminación de la acción. Ir a la acción 8 del flujo normal de eventos.	
--	--

Sección Configurar Timeline

1. Desplaza por el canal la acción hasta el instante de tiempo que desee.	El sistema visualiza la acción en el instante de tiempo asignado por el actor.
---	--

Prototipo de Interfaz



Figura 12: Ventana de acciones.

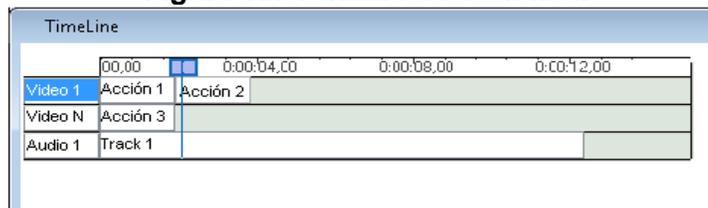


Figura 13: Ventana del timeline.

Pos-condiciones:	<ol style="list-style-type: none"> 1. Acción aplicada a un componente determinado. 2. Acción eliminada del área de trabajo.
-------------------------	---

	3. Acción configurada en el timeline.
--	---------------------------------------

Tabla 8 Descripción textual del CUS Administrar Acciones

2.8 CONCLUSIONES.

La realización del modelo de dominio permitió comprender mejor el funcionamiento del sistema, pues se identificaron y detallaron los principales eventos, términos y conceptos presentes en el entorno donde éste trabajará. Por su parte la realización del diagrama de clases del dominio hizo más comprensible la interrelación que se establece entre las mismas.

La realización del proceso de captura de requisitos concluyó con la obtención de los nuevos requerimientos que la redacción de noticias personalizadas debe brindar y las cualidades que el sistema debe cumplir para poder funcionar correctamente.

El diagrama de casos de uso del sistema tradujo los requisitos funcionales en interacciones del actor con el sistema. Los casos de uso se describieron de forma entendible para que fuera posible obtener un mejor entendimiento de la solución que se dará al problema científico planteado.

CAPÍTULO III: CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA.

3.1 INTRODUCCIÓN.

En el presente capítulo se presenta la solución propuesta, modelada en diagramas de clases del análisis y el diseño. Además se describen los elementos asociados al uso de los framework Symfony y ExtJs en los diagramas de diseño, para que se entiendan con mayor facilidad.

3.2 MODELO DE ANÁLISIS.

El modelo de análisis es una representación abstracta del sistema que se está modelando, suficiente para capturar la lógica esencial del sistema sin entrar en muchos detalles. En él se analizan los requisitos refinándolos y estructurándolos, utilizando el lenguaje de los desarrolladores para describir cómo debería diseñarse e implementarse el sistema.

Es importante resaltar que el modelo de análisis evita resolver algunos problemas y tratar algunos requisitos que es mejor posponer para el diseño. Por esta razón y porque en el diseño se debe considerar la plataforma de implementación, el lenguaje de programación, los framework y otros aspectos, no siempre en el diseño se puede conservar la estructura proporcionada por el análisis.

Los diagramas que se muestran a continuación son el resultado del análisis de los requisitos capturados en el flujo de trabajo anterior y el modelo de casos de uso del sistema, que servirán como primera entrada al diseño.

3.2.1 Diagramas de clases del análisis.

En los siguientes diagramas se representan las clases del análisis⁹ y las relaciones que se establecen entre ellas.

⁹ Clases asociadas al contexto del dominio del problema y que describen la realización de los casos de uso.

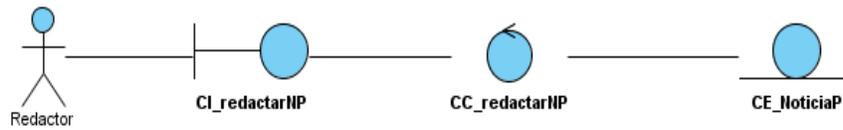


Figura 14: Diagrama de clases del análisis del CUS: Redactar Noticia Personalizada.

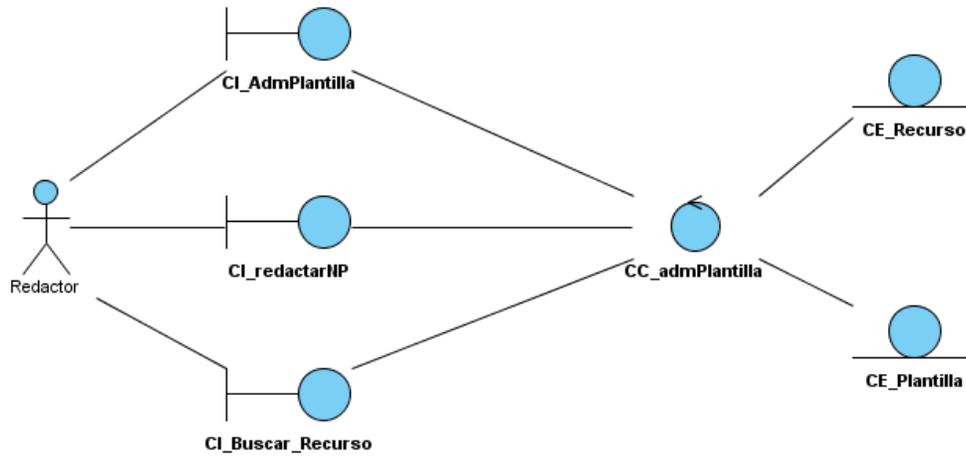


Figura 15: Diagrama de clases del análisis del CUS: Administrar Plantilla.

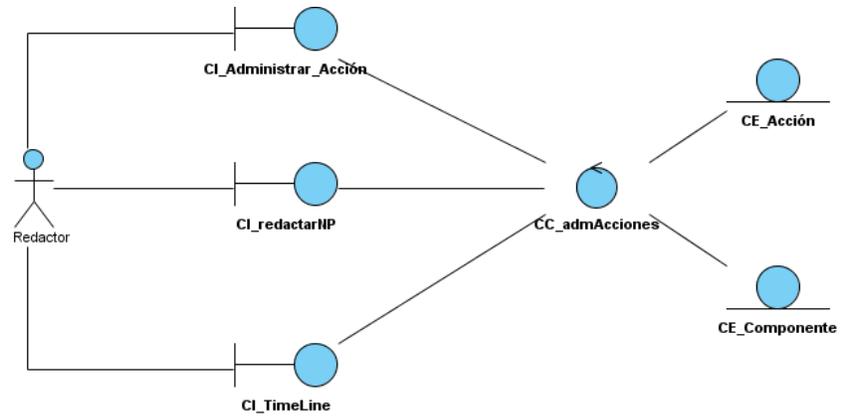


Figura 16: Diagrama de clases del análisis del CUS: Administrar Acción.

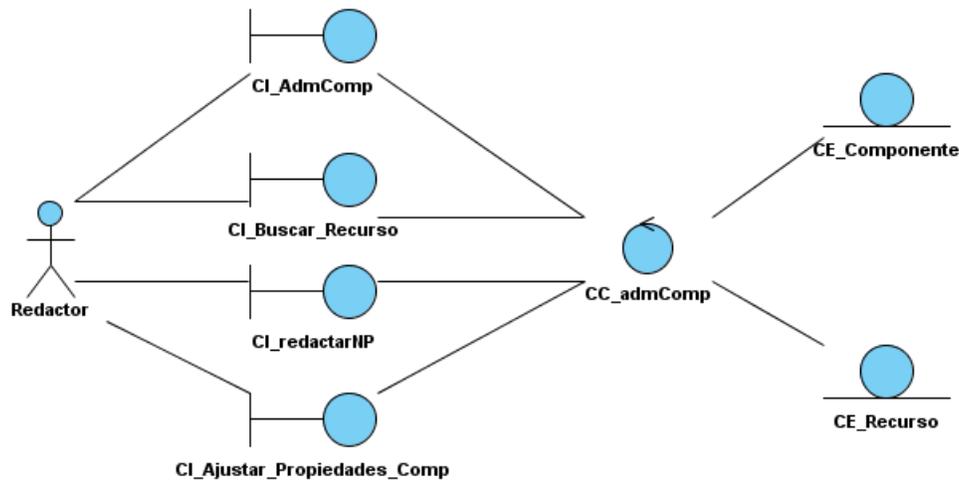


Figura 17: Diagrama de clases del análisis CUS: Administrar Componente.

Luego de realizados los diagramas de clases del análisis se realizan los diagramas de colaboración, para modelar los aspectos dinámicos del sistema. Estos muestran una interacción concreta, es decir, la organización estructural de un conjunto de objetos, sus relaciones y los mensajes que se envían entre ellos.

Los restantes diagramas de clases del análisis y los diagramas de colaboración correspondientes al modelo de análisis realizado se encuentran en el anexo 2 de esta investigación.

3.3 PATRONES DE DISEÑO.

Un patrón es una descripción de un problema y la solución, a la que se da un nombre, y que se puede aplicar a nuevos contextos; idealmente, proporciona consejos sobre el modo de aplicarlo en varias circunstancias, y considera los puntos fuertes y compromisos. (8)

Los patrones de diseño deben ser adaptados al software que se está desarrollando, pues constituyen una guía para resolver problemas comunes en programación. Su utilización ayuda a obtener un software de calidad, que sea reutilizable y que permita ampliar sus funcionalidades.

Los patrones de diseño son muchos y muy variados, resulta casi imposible revisar todos los que existen a la hora de desarrollar una aplicación. Por este motivo se recomienda el uso de los patrones que estén predeterminados en cada uno de los estilos que se seleccionen para la arquitectura.

3.3.1 Patrones de Principios Generales para Asignar Responsabilidades (GRASP) que implementa Symfony.

Los patrones GRASP describen los principios fundamentales del diseño de objetos y la asignación de responsabilidades.

Patrón Experto: Plantea que se debe asignar determinada responsabilidad al experto en información, o sea, a la clase que tiene la información necesaria para realizar la responsabilidad.

Este patrón es uno de los más utilizados en Symfony. Para realizar la capa de abstracción en el modelo Propel¹⁰ es quien tiene la información necesaria para encapsular toda la lógica de los datos y generar las clases con todas las funcionalidades comunes de las entidades.

Patrón Bajo Acoplamiento: Plantea que una clase no debe estar conectada con mucha fuerza ni depender de muchas otras, esto garantiza el aumento de la reutilización. El bajo acoplamiento no se debe llevar al extremo, se plantea que se deben crear sistemas orientados a objetos en los que las tareas se lleven a cabo mediante la colaboración de objetos conectados, evitando así tener objetos inconexos, saturados y con una actividad compleja.

En Symfony las clases Actions heredan solamente de sfActions, que es una clase estable en cuanto a su implementación por lo que se garantiza que exista un grado moderado de acoplamiento entre las clases.

Patrón Alta Cohesión: Plantea que las clases con responsabilidades estrechamente relacionadas no deben realizar un trabajo enorme, por lo que se debe asignar una responsabilidad garantizando que la cohesión siga siendo alta.

Symfony permite asignar responsabilidades a una clase con una alta cohesión pues plantea que debe existir una acción para cada una de las plantillas que producen la vista. Además las clases Actions colaboran con otras para realizar diferentes operaciones, instanciar objetos y acceder a las propiedades, proporcionando que el software sea flexible frente a grandes cambios.

Patrón Creador: Guía la asignación de responsabilidades relacionadas con la creación de objetos, una tarea muy común en los sistemas orientados a objetos. La correcta asignación de las

¹⁰ Propel es un ORM (Object-Relational Mapping) para PHP que se encarga de generar las clases de la capa del modelo en función de la estructura de datos de la aplicación.

responsabilidades de creación trae consigo un diseño que puede soportar un bajo acoplamiento, mayor claridad, encapsulación y reutilización.

En Symfony en la clase Actions se definen y ejecutan todas las acciones. En las acciones se crean los objetos de las clases que representan las entidades, evidenciando de este modo que la clase Actions es "creador" de dichas entidades.

Patrón Controlador: Plantea que se debe asignar la responsabilidad del manejo de un mensaje de los eventos de un sistema a un controlador. Éste concentra todas las solicitudes de la aplicación delegándolas en el manejador específico, que generalmente es un comando.

En Symfony todas las peticiones web son manejadas por el controlador frontal, que es el punto de entrada único de toda la aplicación en un entorno determinado.

3.3.2 Patrones Gang-of-Four (GOF) que implementa Symfony.

Patrones Creacionales:

Singleton (Instancia única): El patrón Singleton es uno de los más sencillos patrones de diseño, y es útil para limitar el máximo número de instancias de una clase en exactamente solo una. En este caso, si más de un objeto necesita utilizar una instancia de la clase Singleton, esos objetos comparten la misma instancia de la clase Singleton. En un uso más avanzado, este patrón puede ser utilizado también para administrar n instancias de una clase. (8)

Symfony en el controlador frontal tiene una llamada a `sfContext::getInstance()`. En una acción, el método `getContext()`, es un objeto muy útil que guarda una referencia a todos los objetos del núcleo de Symfony. La utilización de este patrón garantiza la existencia de una única instancia para una clase y la creación de un mecanismo de acceso global a dicha instancia.

Abstract Factory (Fábrica abstracta): Permite trabajar con objetos de distintas familias de manera que las familias no se mezclen entre sí haciendo transparente el tipo de familia concreta que se esté usando. (8)

Symfony utiliza las factorías en su funcionamiento interno, por ejemplo para los controladores y para las sesiones. Cuando el framework necesita por ejemplo crear un nuevo objeto para una petición, busca en la definición de la factoría el nombre de la clase que se debe utilizar para esta tarea. Como

la definición por defecto de la factoría para las peticiones es `sfWebRequest`, Symfony crea un objeto de esta clase para tratar con las peticiones. (9)

Patrones Estructurales:

Decorator (Decorador): Añade funcionalidad a una clase de una forma dinámica y transparente. En Symfony el archivo `layout.php` o plantilla global almacena el código HTML que es común a todas las páginas de la aplicación, para no tener que repetirlo en cada página. El contenido de las plantillas se integra en el layout, o si se mira desde el otro punto de vista, el layout decora el contenido de la plantilla. (9)

3.4 MODELO DE DISEÑO

El modelo de diseño es un modelo físico, concreto y un plano de la implementación; da forma al sistema y debe ser mantenido durante todo el ciclo de vida del software. El diseño de un software es el punto de partida para su implementación, por tanto a la hora de diseñar es necesario tener en cuenta los requisitos y las restricciones que imponen los lenguajes de programación y los framework que se utilizarán.

Para el desarrollo de las nuevas funcionalidades del módulo de redacción de PRIMICIA se utilizarán Symfony y ExtJs. A continuación se muestra una breve descripción del funcionamiento de estos framework.

3.4.1 Symfony y ExtJs.

Symfony está basado en un patrón clásico del diseño web conocido como arquitectura MVC. (9) Esta arquitectura separa las diferentes clases en dependencia de la función que realizan de modo tal que sea posible manejar dinámicamente la forma en que se procesan las solicitudes y se gestiona la manera en que se muestran resultados al usuario final.

La arquitectura MVC está formada por 3 niveles, que son independientes y la comunicación entre ellos es mediante interfaces que abstraen sus estructuras internas:

- ❖ El modelo, que representa la información con la que trabaja la aplicación, es decir, su lógica de negocio.

- ❖ La vista, que transforma el modelo en una página web que permite al usuario interactuar con ella.
- ❖ El controlador, que se encarga de procesar las interacciones del usuario y realiza los cambios apropiados en el modelo o en la vista.

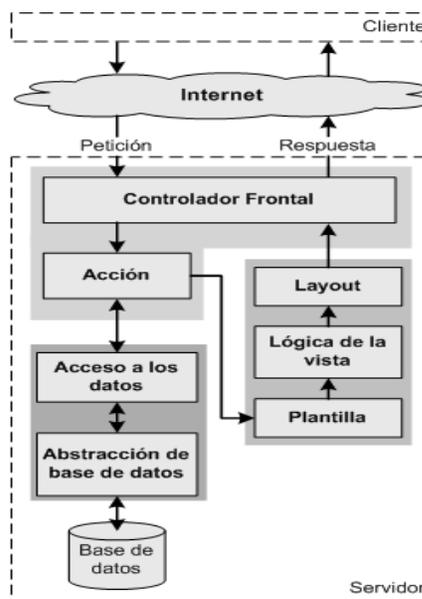


Figura 18 La implementación del MVC que realiza Symfony. (9)

Por lo general el controlador tiene una gran carga de trabajo, por esto Symfony divide este nivel en dos componentes: controlador frontal y acciones. El controlador frontal es el único punto de entrada a la aplicación, éste carga la configuración y determina la acción a ejecutarse (9). Las acciones por su parte contienen la lógica de la aplicación y preparan los datos requeridos por la vista.

En Symfony cuando el controlador frontal recibe una petición, utiliza el sistema de enrutamiento para asociar el nombre de una acción y el nombre de un módulo con la URL escrita o accedida por el usuario. Luego las acciones utilizan las clases del modelo para definir las variables que se van a mostrar en la vista.

La complejidad de las interfaces web dinámicas, que incluyen múltiples interacciones AJAX, ha provocado la necesidad del uso de ExtJS como framework especializado en el desarrollo de aplicaciones con JavaScript. Este framework permite construir aplicaciones complejas con mayor eficiencia y se caracteriza fundamentalmente por proveer un set de componentes que permite crear aplicaciones web 2.0 con interfaces muy similares a las de una aplicación de escritorio. (10) Su uso

en la vista de la aplicación permite que la carga de procesamiento se distribuya, pues el servidor al tener menor carga logra manejar un mayor número de clientes de forma simultánea.

En las aplicaciones web tradicionales, las acciones del usuario en la página desencadenan peticiones al servidor. Una vez procesada la petición del usuario, el servidor devuelve una nueva página HTML al navegador del usuario y este procede a actualizarse completamente.

En la aplicación diseñada las peticiones se envían de forma asíncrona al servidor web mediante AJAX.Request y las respuestas del servidor se devuelven como datos JSON (JavaScript Object Notation). Queda luego del lado de la vista de la aplicación la responsabilidad de actualizar únicamente la zona que corresponde. Utilizando esta tecnología hay muchas peticiones que son simples y no requieren la intervención del servidor, de modo que la respuesta es inmediata.

El uso de Symfony utilizando las ventajas que aporta ExtJS brinda la posibilidad de diseñar y desarrollar las funcionalidades propuestas con una excelente calidad. La interacción del usuario no se ve interrumpida por recargas de página o largas esperas por la respuesta del servidor por lo que se potencia la usabilidad¹¹ al mejorar la calidad de la interacción. Además resulta atractiva y novedosa para los usuarios la presentación de las interfaces web de manera semejante a aplicaciones de escritorio.

3.4.2 Diagramas de Clases del Diseño

Los diagramas de clases del diseño que se muestran a continuación servirán de apoyo al equipo de desarrollo encargado de implementar las funcionalidades. Constituyen la mayor aproximación al sistema que se está proponiendo, por lo que una buena modelación de las clases del diseño contribuye a que el software se implemente con buena calidad. El diseño propuesto se estructura en consecuencia con el patrón MVC implementado por Symfony.

Modelo: En las aplicaciones Symfony, el acceso y la modificación de los datos almacenados en la base de datos se realiza mediante objetos. Nunca se accede de forma explícita a la base de datos, lo que permite un alto nivel de abstracción y una fácil portabilidad. En Symfony la librería Propel genera las clases del modelo y Creole realiza la abstracción de la base de datos de una forma invisible para el programador. Se generan por cada tabla real de base de datos cuatro clases: NombreTabla, NombreTablaPeer, BaseNombreTabla, BaseNombreTablaPeer.

¹¹ Usabilidad se define en el estándar ISO 9241 como “el grado en el que un producto puede ser utilizado por usuarios específicos para conseguir objetivos específicos con efectividad, eficiencia y satisfacción en un determinado contexto de uso” (11)

En el presente diseño se representarán sólo las clases BaseNombreTabla con los atributos que intervienen en cada caso de uso, pues son las que proporcionan información acerca de los datos existentes en la base de datos.

Vista: Se encarga de producir las páginas que se muestran como resultado de las acciones. Generalmente está compuesta por las páginas cliente con sus formularios, el layout (plantilla global) y las plantillas (complemento de las acciones, nombradas para una acción executeNombreAction: NombreActionSuccess.php). En el diseño realizado el estereotipo `<<js class>>` se utiliza en el diseño propuesto para representar los objetos JavaScript, pues el concepto de clase no existe en la versión 3 de ECMA Script.

El siguiente diagrama muestra las relaciones que se establecen con el framework ExtJS.

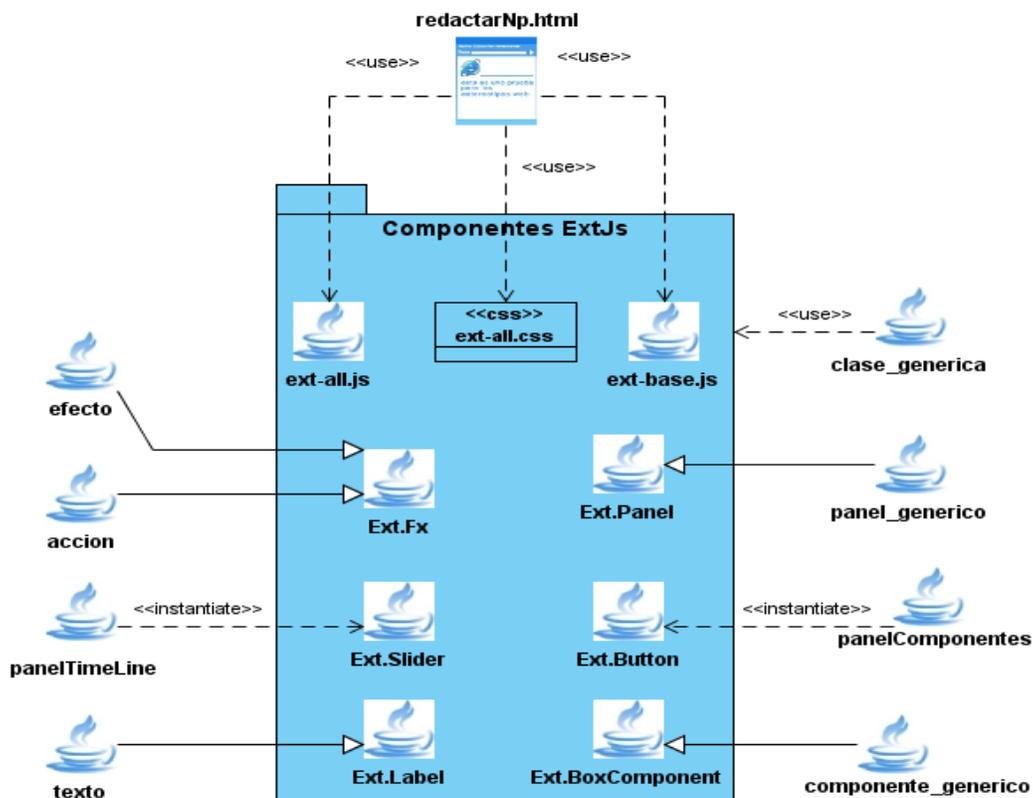


Figura 19: Relación con los componentes de ExtJS.

Controlador: En Symfony el controlador frontal es quien determina la acción a ejecutarse y produce la vista que será mostrada al usuario. Cuando se realiza una petición web en una aplicación Symfony, la URL define una acción y los parámetros de la petición. Las acciones se encuentran agrupadas por

módulos, son métodos con el nombre `executeNombreAccion` de una clase llamada `nombreModuloActions` que hereda de la clase `sfActions`. (9)

En el diagrama que se muestra a continuación se representan las clases más relevantes que describen el comportamiento interno del framework y sus relaciones con las clases de la vista y el controlador.

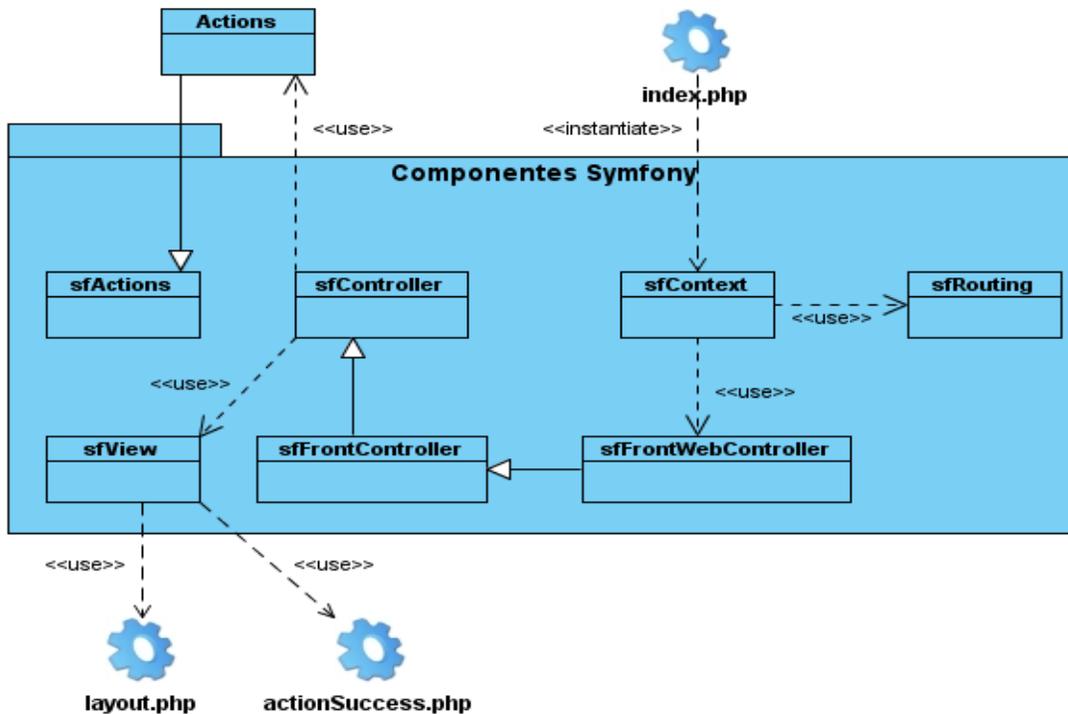


Figura 20: Clases relacionadas con los componentes de Symfony.

En el diseño que se propone a continuación existen muchas funcionalidades que no requieren el uso de una acción específica del controlador, pues se ejecutan completamente en la vista. Además en los paquetes Componentes Symfony y Componentes ExtJS no se muestran las clases que los componen, pues fueron mostradas anteriormente.

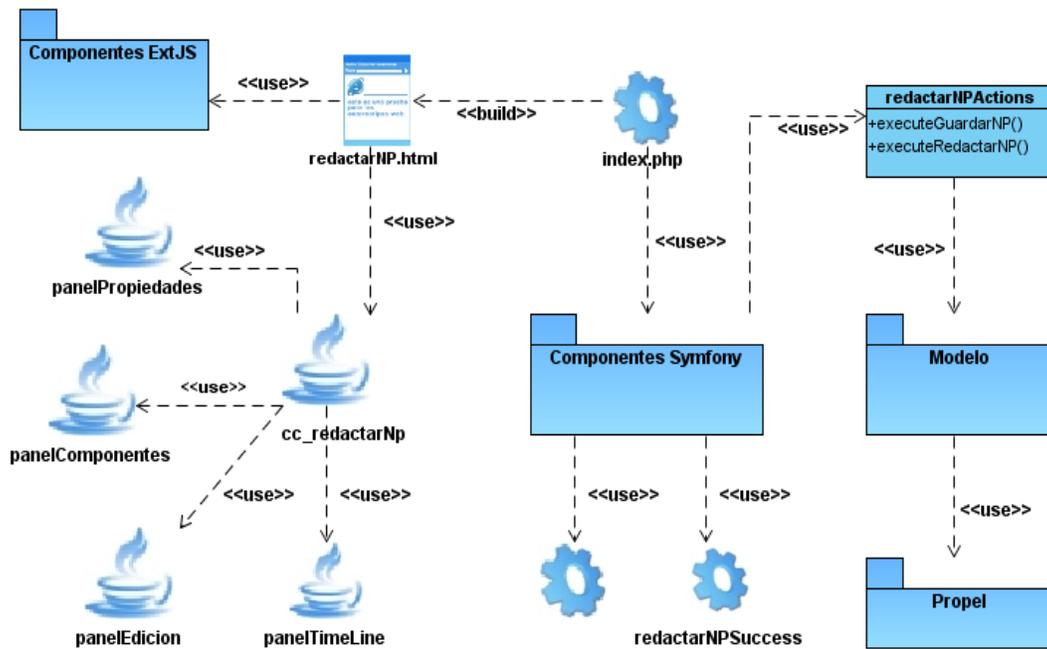


Figura 21: Diagrama de clases del diseño del CUS: Redactar Noticia Personalizada.

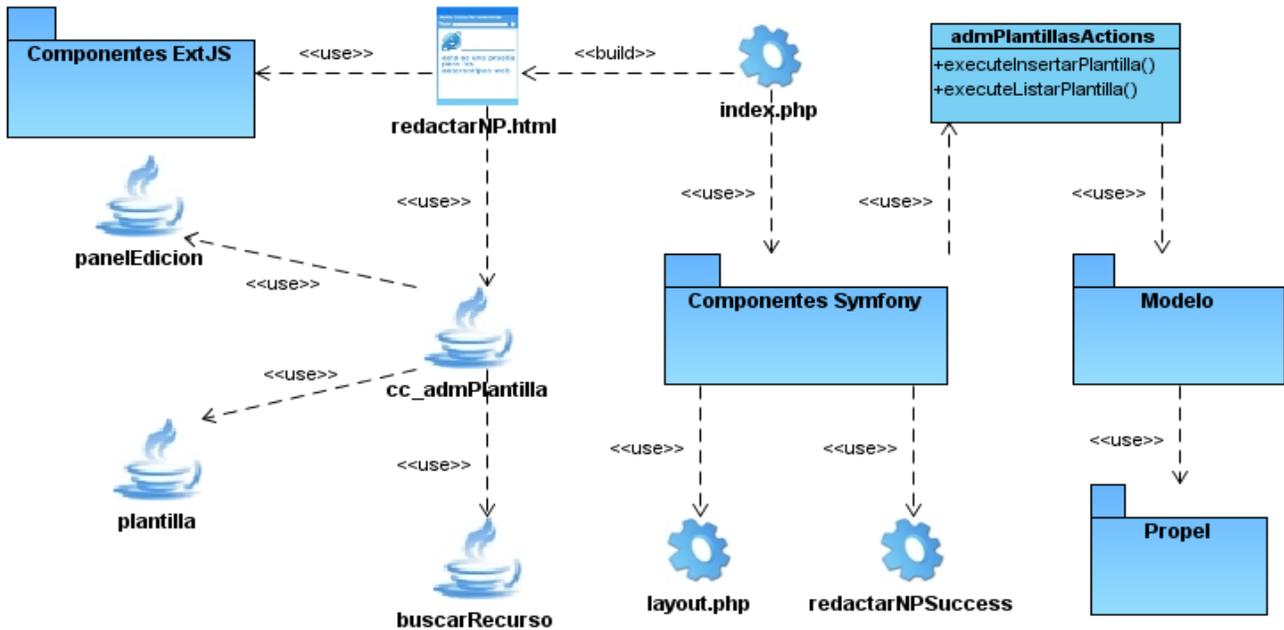


Figura 22: Diagrama de clases del diseño del CUS: Administrar Plantilla.

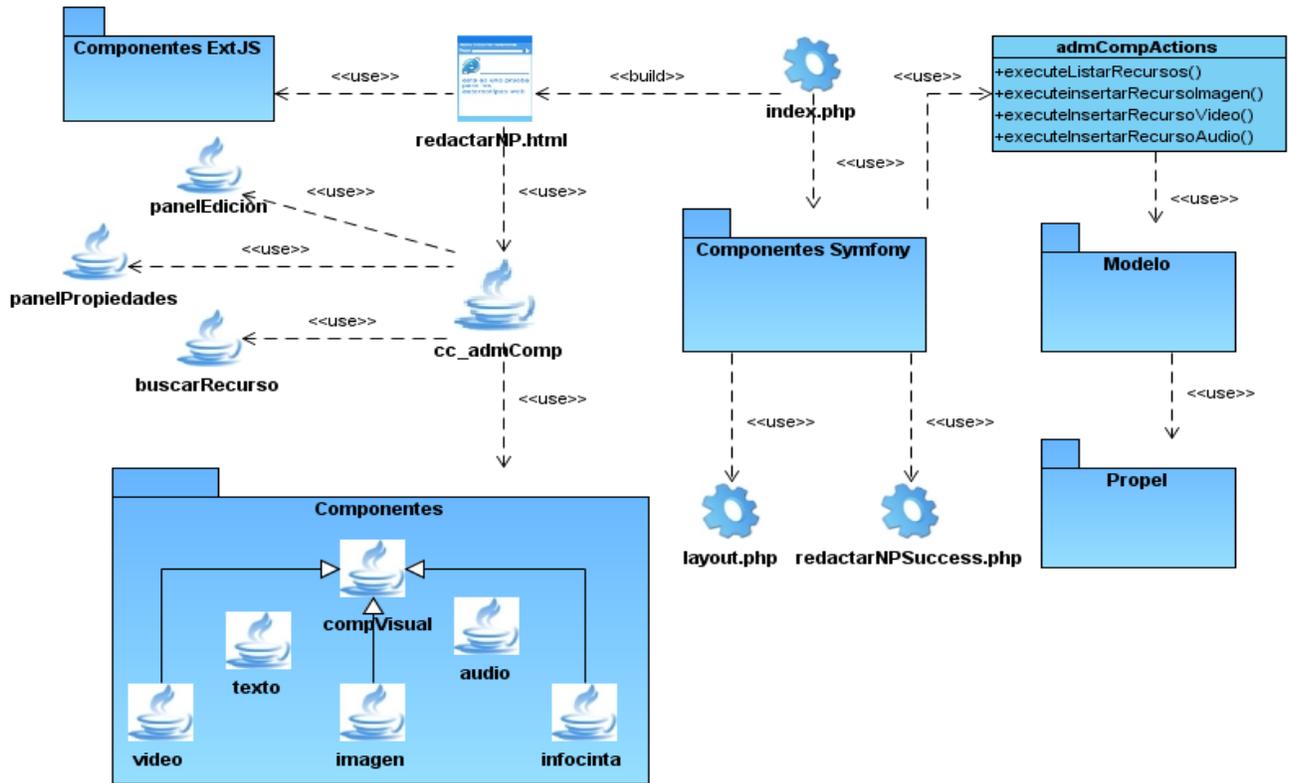


Figura 23: Diagrama de clases del diseño del CUS: Administrar Componente.

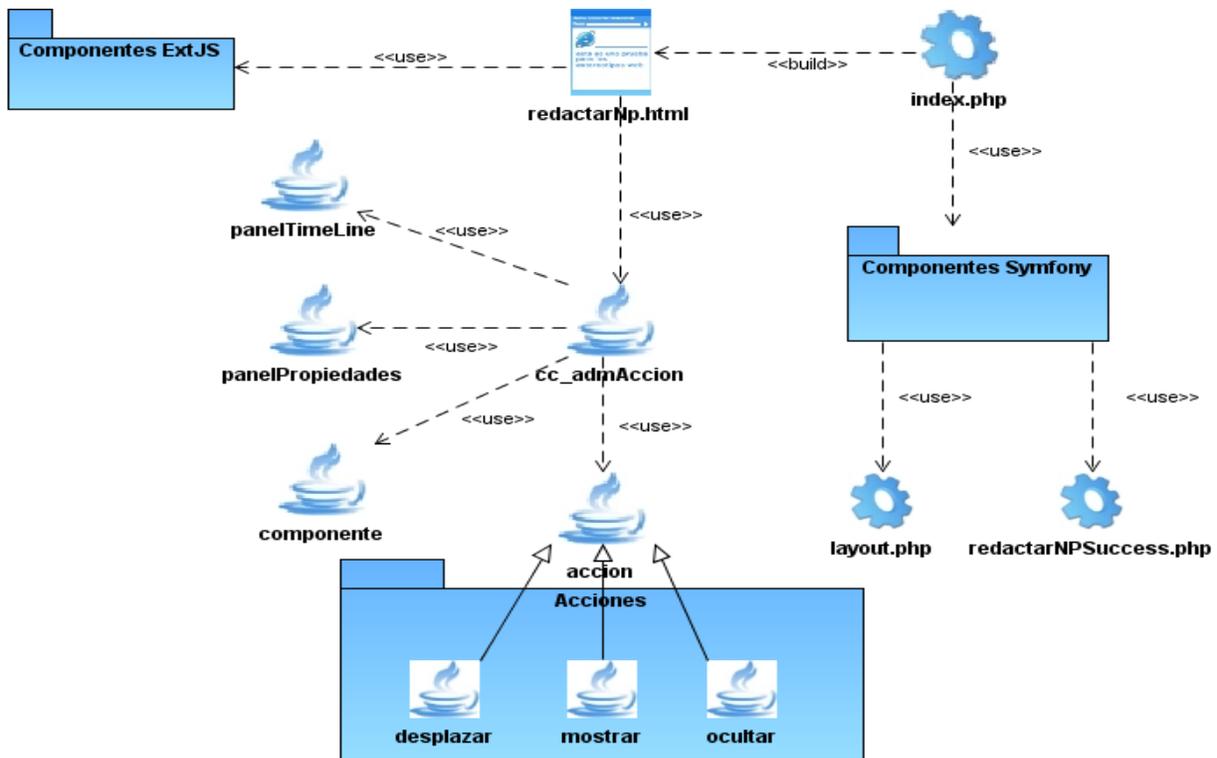


Figura 24: Diagrama de clases del diseño del CUS: Administrar Acción.

Los restantes diagramas de clases del diseño se encuentran en el anexo 3 y las relaciones entre las clases Actions y las clases del modelo se muestran en el anexo 4. Los diagramas de secuencia del diseño que se encuentran en el anexo 5 representan la parte dinámica de las relaciones entre las clases del diseño.

3.5 DISEÑO DE LA BASE DE DATOS.

El objetivo fundamental del diseño de la base de datos es garantizar que los datos persistentes se almacenen de forma coherente y eficaz. (12) Symfony genera de forma automática el modelo de datos de un proyecto. Este modelo está constituido por cuatro clases por cada tabla de la base de datos, estas son:

baseNombreTablaPeer: Son clases que tienen métodos estáticos para trabajar con las tablas de la base de datos. Proporcionan los medios necesarios para obtener los registros de las tablas y sus métodos devuelven normalmente un objeto o una colección de objetos de la clase objeto relacionada. Representan la parte del modelo que se encarga del acceso a los datos. (12)

baseNombreTabla: Son clases objeto que representan un registro de la base de datos. Permiten acceder a las columnas de un registro y a los registros relacionados. Representan la parte del modelo que se encarga de la abstracción de los datos. (12)

nombreTabla, nombreTablaPeer: Sirven para extender las funcionalidades de sus clases bases correspondientes. (12)

En la siguiente figura se muestra la representación genérica del modelo generado por Symfony:

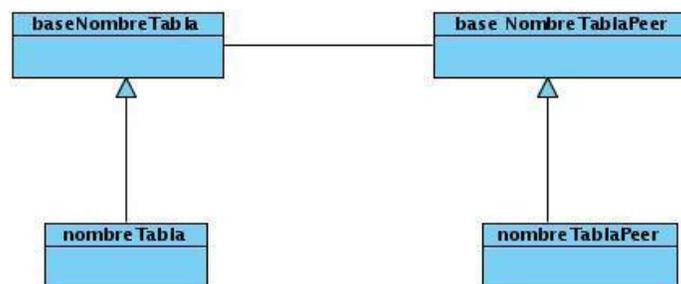


Figura 25: Estructura de las clases del modelo generadas por Symfony.

Teniendo en cuenta lo expuesto anteriormente y que son las clases baseNombreTabla las que mapean las tablas de la base de datos, se realizó el diagrama de clases persistentes que se muestra a continuación.

3.5.1 Diagrama de clases persistentes.

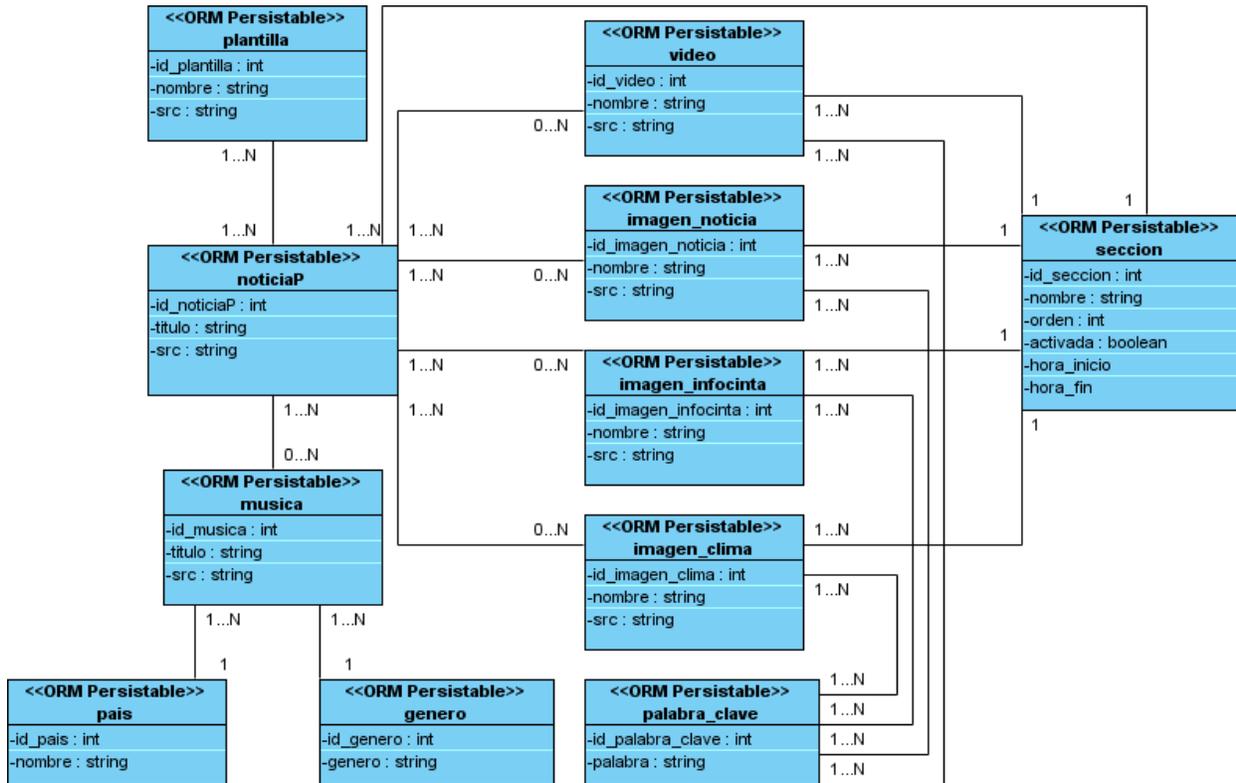


Figura 26: Diagrama de clases persistentes.

Sobre la base del diagrama de clases persistentes realizado se obtuvo el modelo de entidad relación que se muestra a continuación.

3.5.2 Modelo entidad relación.

En el siguiente diagrama entidad relación solo se muestra un fragmento de la base de datos de PRIMICIA, o sea, sólo se modelan las entidades que intervienen en la redacción de noticias personalizadas.

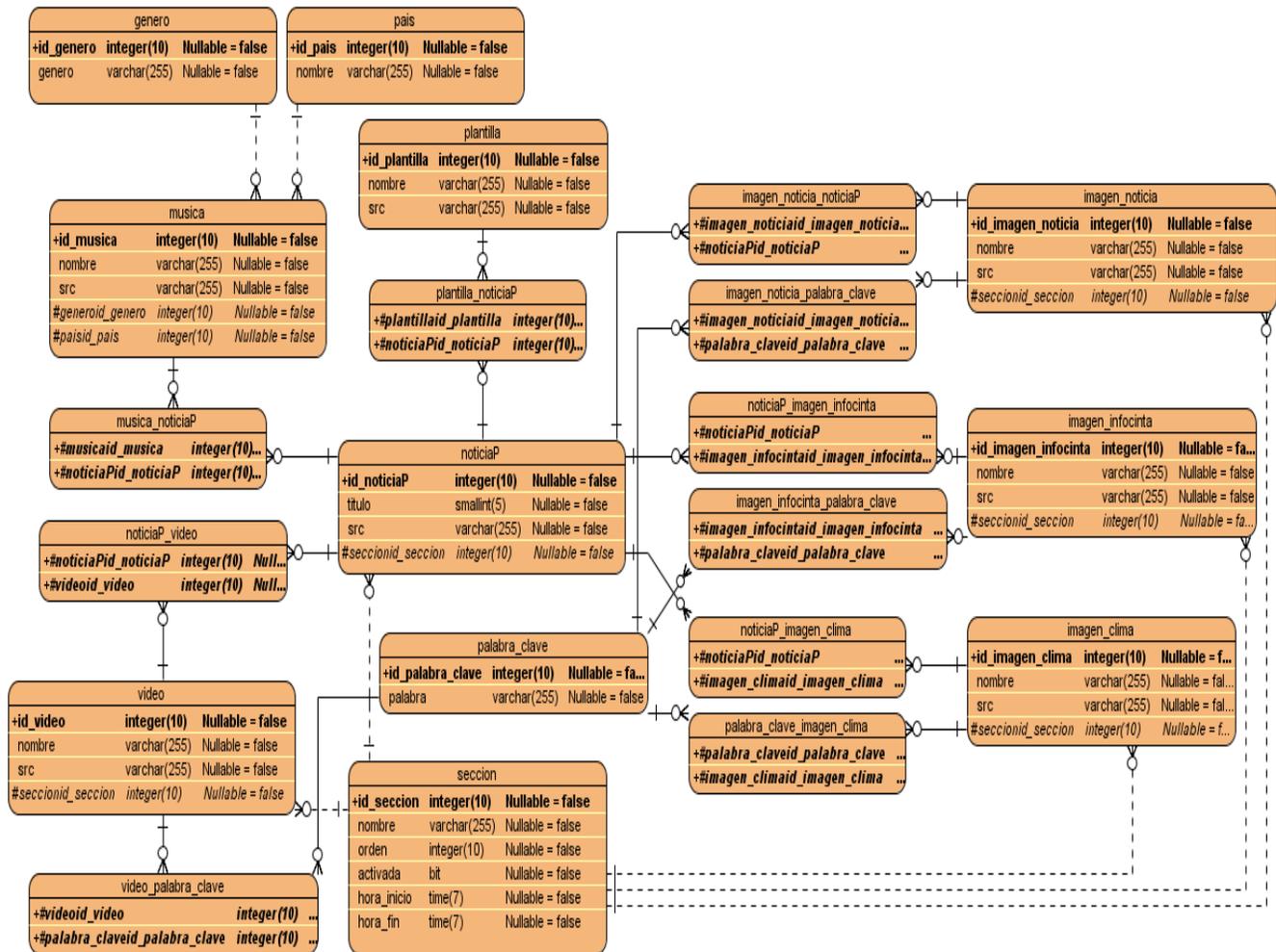


Figura 27: Diagrama entidad relación.

3.6 CONCLUSIONES

En el presente capítulo se muestran los diagramas de clases del diseño realizados durante el flujo de trabajo de análisis y diseño. Éstos constituyen la base para la implementación de las nuevas funcionalidades que se integrarán al módulo de redacción de noticias de PRIMICIA. El uso de Symfony integrado con ExtJs proporciona numerosas facilidades para la implementación, aunque introduce un cierto grado de complejidad en el diseño.

CONCLUSIONES

La investigación realizada permitió obtener una base teórica del proceso de redacción de noticias que posee actualmente PRIMICIA, revelando la importancia que tiene la incorporación de animaciones y elementos configurables. El estudio de soluciones similares existentes demostró que éstas no brindan la posibilidad de crear noticias tan personalizadas como las que se pretenden crear en la plataforma y que hacen de ésta un producto único.

La utilización de RUP como metodología de desarrollo de software y Visual Paradigm como herramienta CASE proporcionó una documentación detallada del sistema que se va a implementar. El uso de Symfony y ExtJS en el diseño busca mejorar la interacción del usuario con el sistema al crear aplicaciones interactivas (RIA, acrónimo de Rich Internet Applications). Las RIA brindan numerosos beneficios para la redacción en la plataforma pues permiten que los elementos sean configurados en el lado del cliente sin la necesidad de enviar de forma constante informaciones al servidor.

Por lo expuesto anteriormente se puede afirmar que el objetivo general de la investigación fue cumplido, pues los artefactos generados durante los flujos de trabajo servirán de apoyo a la automatización del proceso de redacción de noticias personalizadas en la plataforma de televisión informativa PRIMICIA.

RECOMENDACIONES

Teniendo en cuenta los resultados obtenidos con el desarrollo de esta investigación se recomienda:

- Implementar las funcionalidades concebidas para mejorar la redacción en la plataforma PRIMICIA, tomando como primera entrada para la implementación el modelo de diseño desarrollado.
- Incorporar nuevos componentes configurables y funcionalidades a la redacción de noticias personalizadas.
- Agregar nuevas propiedades, acciones y efectos a los elementos configurables propuestos en la presente investigación.

BIBLIOGRAFÍA CITADA

1. **Calzado, Frank Benítez.** *Procedimiento para el seguimiento y tratamiento de los errores de PRIMICIA.* Universidad de las Ciencias Informáticas. Ciudad de la Habana : s.n., 2009. Tesis.
2. **Miranda, Rafael Lorente.** *Propuesta de mejoras para el proceso de redacción de noticia de la Plataforma de Televisión Informativa PRIMICIA.* Universidad de las Ciencias Informáticas. Ciudad de la Habana : s.n., 2009. Tesis.
3. **Rafael Díaz Arias.** Bilbioteca.net Universia. [En línea]
<http://eprints.ucm.es/8029/1/Redacci%C3%B3nInfo.pdf>.
4. Estructure. [En línea] 2010. <http://www.estructuretv.com/enewsroom.htm>.
5. **James Rumbaugh, Ivar Jacobson, Grady Booch.** *El Proceso Unificado de Desarrollo de Software.* Madrid : s.n., 2000.
6. **Ivar Jacobson, Grady Booch, James Rumbaugh.** *El Lenguaje Unificado de Modelado. Manual de Referencia.* Madrid : s.n., 2000.
7. **Toro, Amador Durán.** *Un Entorno Metodológico de Ingeniería de Requisitos para Sistemas de Información.* Universidad de Sevilla. 2000. Tesis Doctoral.
8. **Larman, Craig.** *UML y Patrones. Introducción al análisis y diseño orientado a objetos.* México : s.n. 970-17-0261-1.
9. **Fabien Potencier, François Zaninotto.** *Symfony la guía definitiva.* 2007.
10. *Introducción a ExtJs. (Primera Parte).* **Barroso, Boris.** Revista de Software Libre ATIX., Vol. 5.
11. **9241, ISO.** Ergonomic requirements for office work with visual display terminals (VDTs) -- Part 11: Guidance on usability. 1998.
12. **Copyright IBM.** Ayuda RUP. 2006.
13. **Lisbeth López, Jose Luis Muñoz.** *Subsistema para la consulta de metadatos geológicos en la Oficina Nacional de Recursos Minerales.* Universidad de las Ciencias Informáticas. Ciudad de la Habana : s.n., 2008.

BIBLIOGRAFÍA CONSULTADA

- **ESTRUCTURE TV.** [En línea] [Citado el: 24 de Nov de 2009.]
<http://www.estructuretv.com/enewsroom.htm>.
- **Calzado, Frank Benítez.** Procedimiento para el seguimiento y tratamiento de los errores de PRIMICIA. Universidad de las Ciencias Informáticas. Ciudad de la Habana : s.n., 2009. Tesis.
- **Miranda, Rafael Lorente.** Propuesta de mejoras para el proceso de redacción de noticia de la Plataforma de Televisión Informativa PRIMICIA. Universidad de las Ciencias Informáticas. Ciudad de la Habana : s.n., 2009. Tesis.
- **James Rumbaugh, Ivar Jacobson, Grady Booch.** El Proceso Unificado de Desarrollo de Software. Madrid : s.n., 2000.
- **Ivar Jacobson, Grady Booch, James Rumbaugh.** El Lenguaje Unificado de Modelado. Manual de Referencia.
- **Pressman, Roger S.** Ingeniería de software: Un enfoque práctico. 4ta Edición. 970-10-5473-3.
- **Larman, Craig.** *UML y Patrones. Introducción al análisis y diseño orientado a objetos.* México : Prentice Hall : s.n. 970-17-0261-1.
- **Arias, Yuniel Eliades Proenza.** *Diseño Avanzado de Aplicaciones Web. Ext-Zend Framework y Doctrine.* 2009, Revista de Software Libre ATIX., Vol. 2.
- **Fabien Potencier, François Zaninotto.** *Symfony la guía definitiva.* 2007.
- **Barroso, Boris.** *Introducción a EXT JS.* 5, 2008, Revista de Software Libre ATIX.
- **Shea Frederick, Colin Ramsay, Steve 'Cutter' Blades.** *Learning Ext JS.* Mumbai : s.n., 2008.
- **Jesus D. Garcia, Jr.** *Ext Js in Action.* 2009.
- **Ivica Crnkovic, Magnus Larsson.** *Building Reliable Component-Based Software Systems.* 2002. 1-58053-327-2.
- **Lobo, Armando Robert.** *Arquitectura de Software para el Sistema Integrado de Gestión Estadística 2.0 Nuragas.* Ciudad de la Habana : s.n., Junio 2009.
- **Riesco, Prof. Daniel.** Objetos, UML Diagrama de Clases y de Objetos.
- **Juan Pablo Gómez Gallego, Ing. Jorge Galves.** *FUNDAMENTOS DE LA METODOLOGIA RUP. RATIONAL UNIFIED PROCESS.* 2007.
- ExtJS 3.2.1 API Documentation. [En línea] <http://www.extjs.com/deploy/dev/docs/>.

- **9241, ISO.** Ergonomic requirements for office work with visual display terminals (VDTs) -- Part 11: Guidance on usability. 1998.
- **Copyright IBM.** Ayuda de RUP en Español. 2006.
- **Lisbeth López, Jose Luis Muñoz.** *Subsistema para la consulta de metadatos geológicos en la Oficina Nacional de Recursos Minerales.* Universidad de las Ciencias Informáticas. Ciudad de la Habana : s.n., 2008.

ANEXOS

Anexo 1.

Descripciones Textuales CUS.

1. CUS: Administrar efectos.

Caso de Uso:	Administrar efectos.
Actores:	Redactor.
Propósito:	Permite al redactor aplicar un efecto a una determinada acción.
Resumen:	El CUS inicia cuando el redactor selecciona un efecto para aplicarlo a una acción determinada. Permite eliminarlo o cambiar las propiedades del efecto aplicado.
Referencias:	RF 6, RF 6.1, RF 6.2, RF 6.3.
Precondiciones:	El redactor debe estar autenticado en la aplicación y tener los permisos de redacción. CUS Administrar Acciones.
Prioridad	Alta.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El redactor selecciona en el TimeLine la acción a la que va a aplicarle el efecto.	2. Se activa la ventana de efectos.
3. El redactor selecciona el efecto que desea utilizar.	4. El sistema guarda la aplicación del efecto a la acción seleccionada.

<p>5. El redactor puede realizar las siguientes acciones:</p> <p>a) Configurar las propiedades del efecto. Ir a Sección Configurar Propiedades Efecto.</p> <p>b) Eliminar la aplicación del efecto. Ir a Sección Eliminar Efecto.</p>	
<p>Prototipo de Interfaz</p>	<p>Ver figura 12.</p>
<p>Sección Configurar Propiedades Efecto.</p>	
<p>1. El redactor selecciona una acción en el TimeLine.</p>	<p>2. El sistema muestra en la ventana de acciones el efecto que tiene aplicado.</p> <p>3. Se muestra las propiedades del efecto, con los valores por defecto.</p>
<p>4. El redactor introduce el valor de la propiedad del efecto que desea cambiar.</p> <p>5. Presiona el botón Aceptar.</p>	<p>6. El sistema guarda la aplicación del efecto con las propiedades configuradas.</p>
<p>Sección Eliminar Efecto.</p>	
<p>1. El redactor selecciona la acción a la que desea eliminar el efecto.</p>	<p>2. El sistema visualiza la ventana de efectos mostrando el efecto aplicado y las propiedades.</p>
<p>3. El redactor elige en la ventana de efectos la opción Ninguno.</p>	<p>5. El sistema elimina la aplicación del efecto a</p>

4. Presiona el botón Aceptar.	la acción.
Pos condiciones:	<ol style="list-style-type: none"> 1. Efecto aplicado a una acción, con las propiedades configuradas. 2. Efecto eliminado.

Tabla 9 Descripción textual del CUS Administrar Efecto.

2. CUS: Administrar Escena.

Caso de Uso:	Administrar Escena.
Actores:	Redactor.
Propósito:	Permite al redactor agregar o eliminar escenas de la noticia que está redactando.
Resumen:	El CUS se inicia cuando el redactor desea agregar una escena nueva a la noticia que está redactando. Luego puede eliminar la escena, que incluye eliminar todos los componentes que se encuentren dentro de ésta.
Referencias:	RF 7, RF 7.1, RF 7.2.
Precondiciones:	El redactor debe estar autenticado en la aplicación y tener los permisos de redacción.
Prioridad	Alta.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El redactor oprime el botón Agregar Escena.	2. El sistema muestra un formulario para introducir el nombre de la escena.

<p>3. El redactor introduce el nombre y presiona el botón OK.</p>	<p>4. El sistema agrega una escena a la noticia que está redactando.</p> <p>5. El sistema envía al CUS Administrar Plantillas.</p>
<p>6. El redactor puede eliminar la escena agregada a la noticia. Ir a Sección Eliminar Escena.</p>	
<p>Sección Eliminar Escena.</p>	
<p>1. El redactor selecciona la escena que desea eliminar.</p> <p>2. Oprime el botón Eliminar.</p>	<p>3. El sistema muestra un mensaje de confirmación sobre la eliminación de la escena.</p>
<p>4. El redactor confirma que desea eliminar la escena</p>	<p>5. El sistema elimina la escena con todos los elementos agregados a ésta.</p>
<p>Flujo Alternativo de Eventos.</p>	
<p>4. El redactor cancela la eliminación de la escena.</p>	<p>5. El sistema muestra la escena con los cambios realizados en ésta.</p>
<p>Prototipo de Interfaz</p>	<div data-bbox="630 1528 1101 1724" style="text-align: center;"> </div> <p style="text-align: center;">Figura 28: Formulario de creación de una escena.</p>

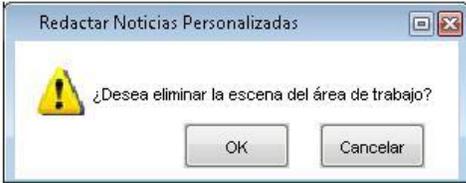
	 <p>Figura 29: Ventana de confirmación de la sección Eliminar Escena.</p>
<p>Pos condiciones:</p>	<ol style="list-style-type: none"> 1. Escena agregada a la noticia. 2. Escena eliminada de la noticia.

Tabla 10: Descripción Textual del CUS Administrar Escena.

Anexo 2.

A continuación se muestran los diagramas de clases del análisis restantes.

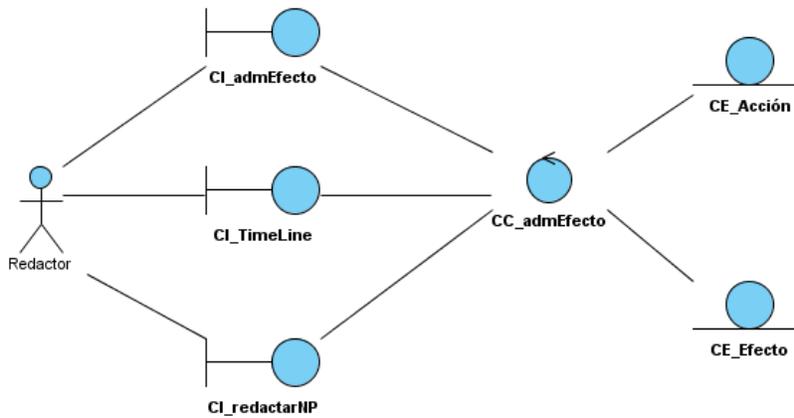


Figura 30: Diagrama de clases del análisis CUS: Administrar Efecto.

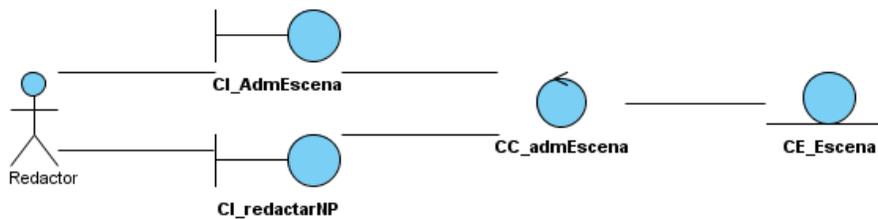


Figura 31: Diagrama de clases del análisis CUS: Administrar Escena.

Anexo 3.

Diagramas de clases del diseño.

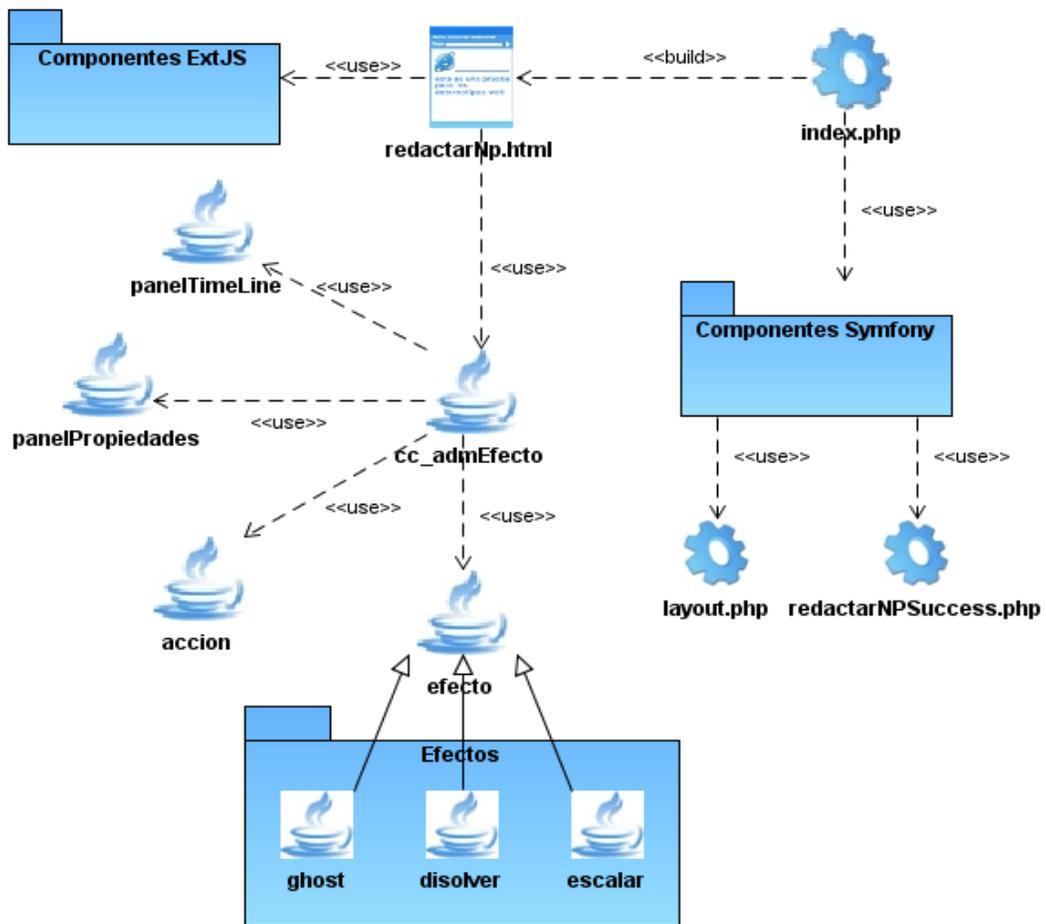


Figura 32: Diagrama de clases del diseño del CUS: Administrar Efecto

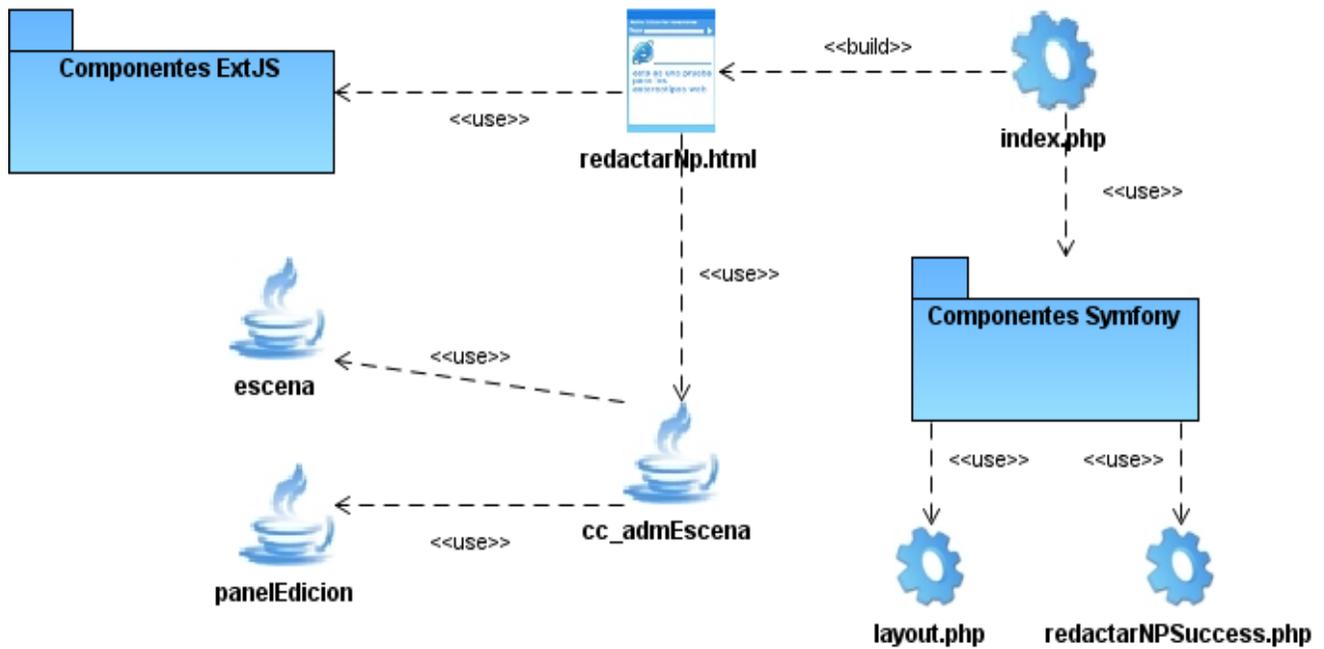


Figura 33: Diagrama de clases del diseño del CUS: Administrar Escena.

Anexo 4.

A continuación se muestran las relaciones entre las clases del modelo y las clases Actions.

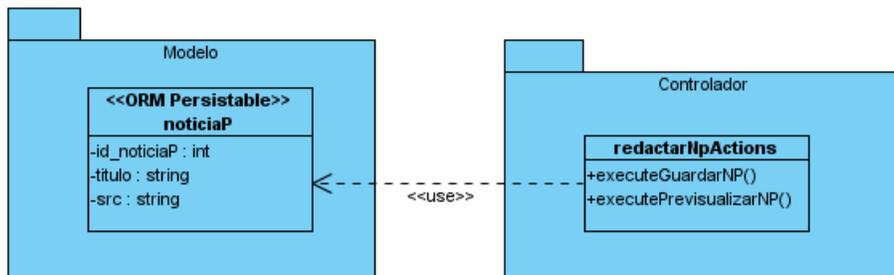


Figura 34: Relación entre redactarNPActions y las clases del modelo.

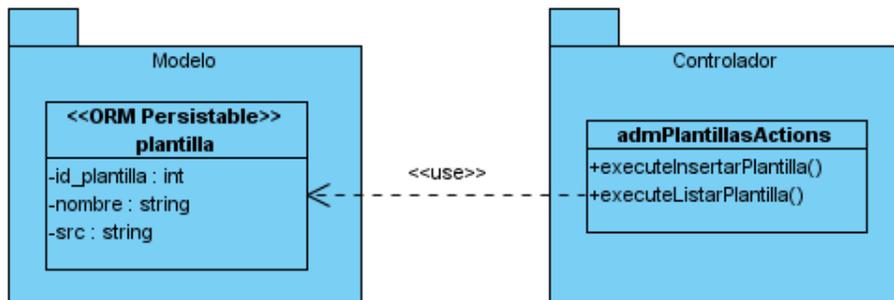


Figura 35: Relación entre admPlantillasActions y las clases del modelo.

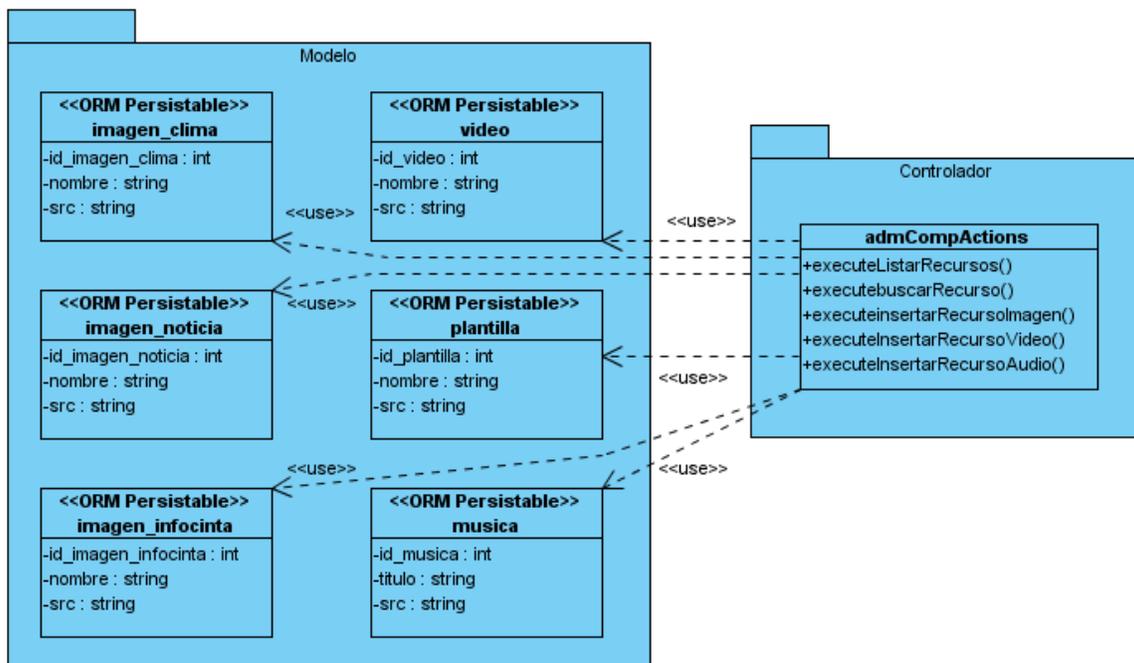


Figura 36: Relación entre admCompActions y las clases del modelo.