

# Plataforma de televisión informativa PRIMICIA: Implementación del módulo de Redacción

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

Autor: Felix Ivan Romero Rodríguez

Tutor: Ing. Rafael Lorente Miranda

Ciudad Habana, Junio de 2011.

“Año 53 de la Revolución”

“La ausencia de evidencia no evidencia la ausencia”

Vladímir Ilich Lenin

## **Dedicatoria**

**A mi Dios....**

**A mi familia...**

**A mis amigos...**

## **Agradecimientos**

Es difícil tener en cuenta a todos aquellos que brindaron su mano, su voz para brindar apoyo, para decir vas bien. Fueron muchos los que en muchos momentos dejaron salir palabras de cariño, de apoyo. Fueron muchos los que regalaron parte de su tiempo para lograr que llegara hasta aquí. Una cuartilla no alcanza para dejar en ella a todos aquellos que hicieron algo por mí. Pero quisiera navegar entre aquello que siempre he tenido más cerca. Mi costa empieza por mi fe, donde creer me ha hecho crecer y tener alguien mejor en mí por estos días. Mi familia, donde he tenido mi piedra angular, mi trampolín para ir hacia adelante. Quisiera incluir a mis hermanos, que desde primer año, me apoyaron con toda su confianza y cariño. Quienes rieron conmigo, se preocuparon, me cuidaron y me entendieron. Agradezco a los que ayudaron mi instrucción durante mi carrera, a todos que me enseñaron los pedacitos que arman un profesional. A mi tutor quien me guío en el desarrollo de este trabajo, todos los que me apoyaron a construir un pedacito de esta labor.

## Declaración de Autoría

Yo Felix Ivan Romero Rodríguez declaro que soy el único autor de este trabajo y autorizo al Departamento de Señales Digitales de la Facultad 6 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 \_\_\_\_\_.

Felix Ivan Romero Rodríguez

Ing. Rafael Lorente Miranda

---

---

Autor

Tutor

## Opinión del Tutor

## Opinión del Oponente

## **Datos de Contacto**

Datos de contacto del tutor:

Tutor: Ing. Rafael Miranda Lorente.

Categoría docente: Adiestrado.

Centro de trabajo: Universidad de las Ciencias Informáticas.

Título de la especialidad de graduado: Ingeniero en Ciencias Informáticas.

Año de graduación: 2009

Institución en la que se graduó: Universidad de las Ciencias Informáticas.

Correo electrónico: rlmiranda@uci.cu



## **Resumen**

Las comunicaciones han constituido a lo largo de la historia un elemento a considerar en el desarrollo y evolución social del hombre. En la actualidad sus avances tecnológicos han devenido en numerosas formas de uso. Es en este ámbito donde entran aplicaciones dentro del campo de las Tecnologías de la Información y las Comunicaciones (TIC).

Los sistemas automatizados de producción de noticias constituyen una forma de uso de las TIC, donde interactúan varias formas de comunicación como la informática y la televisión. La plataforma de televisión informativa PRIMICIA se destaca como la combinación de tecnologías. Mediante esta combinación se logra la gestión y transmisión de noticias en varios formatos a través de una red televisiva. En la actualidad el proceso de redacción de noticias no permite la configuración o animación de los elementos que conforman la noticia.

La presente investigación realizará una solución que garantice la integración de estas funcionalidades. Éstas serán a partir del diseño realizado para la implementación del módulo de redacción de noticias de la plataforma de televisión informativa PRIMICIA en su segunda versión. Para el desarrollo del presente se usa como tecnología de desarrollo RUP, como Entornos Integrados de Desarrollo NetBeans, Aptana, como frameworks de desarrollo ExtJs y Symfony.

### **Palabras Clave**

Implementación, módulo, plataforma de televisión informativa, redacción

## **Abstract**

Communications have formed throughout history a consideration in the development and social evolution of man. Currently, its technological advances have improved in many ways to use. The area of Information Technologies and Communications regards a high performance (ICT).

The automated news production system is a way of ICT, where they combine various forms of communication such as computer and television. The informative television platform PRIMICIA stands out as the combination of technologies. This combination achieves the management and transmission of news in various formats through a television network. Nowadays the news writing process does not allow the configuration and animation of the elements considered in the design developed for the platform. This research will make a solution that ensures the integration of these features. The start point would be the design for implementing a module of writing for PRIMICIA. For the development of this researching is used as RUP development methodology, Integrated Development Environments like NetBeans, Aptana, and development frameworks like Symfony and ExtJs.

**Keywords:** writing, module, information TV platform, implementation.

## Contenido

Dedicatoria.....	3
Agradecimientos .....	4
Declaración de Autoría.....	5
Opinión del Tutor .....	6
Opinión del Oponente .....	7
Datos de Contacto .....	8
Resumen .....	9
Abstract.....	10
Contenido .....	11
Índice de figuras.....	14
Índice de tablas.....	14
Introducción .....	3
Capítulo 1: Fundamentación teórica sobre el proceso de redacción y personalización de noticia e interacción de componentes visuales de la plataforma de televisión informativa PRIMICIA ....	7
1.1 Introducción .....	7
1.2 Conceptos asociados a la configuración y personalización en la implementación del módulo de redacción de noticias de PRIMICIA.....	7
1.2.1 Módulo .....	7

1.2.2 Configuración.....	8
1.2.3 Implementación.....	9
1.2.4 XML: Lenguaje de Marcas Extensible .....	9
1.2.5 Redacción Digital Integrada .....	10
1.2.7 Generador de formato XML.....	11
1.3 Análisis del objeto de estudio.....	12
1.3.1 Soluciones similares a PRIMICIA de ámbito internacional .....	12
1.3.3 Soluciones similares a PRIMICIA de ámbito nacional .....	14
1.3.4 Plataforma de televisión informativa PRIMICIA .....	14
1.3.5 Proceso de redacción de la noticia.....	15
1.3.6 Funciones para el módulo de redacción.....	16
1.4 Herramientas, Técnicas y tecnologías.....	17
1.4.1 Metodología de desarrollo de software RUP .....	17
1.4.2 UML .....	21
1.4.3 Técnicas del Implementador .....	21
1.4.4 Estándar de codificación .....	22
1.4.5 Herramientas y Lenguajes propuestos .....	24
1.5 Conclusiones .....	28
Capítulo 2 Descripción de la solución propuesta.....	29

2.1 Introducción .....	29
2.2 Requisitos Funcionales .....	29
2.3 Descripción del Diseño.....	31
2.4 Análisis de los frameworks de desarrollo Symfony y ExtJs .....	35
2.4.1 Symfony .....	35
2.4.2 ExtJs .....	36
2.5 Diagrama de clases del diseño .....	40
2.6 Modelo de Implementación .....	45
2.7 Conclusiones .....	53
Capítulo 3 Pruebas .....	54
3.1 Introducción .....	54
3.2 Diseño de casos de prueba.....	54
3.2.1 CUS: Redactar Noticia Personalizada .....	55
3.2.2 CUS: Administrar Plantilla .....	56
3.2.4 CUS: Administrar Acción.....	58
3.3 Conclusiones .....	60
Conclusiones .....	61
Recomendaciones .....	63
Bibliografía.....	64

Referencia Bibliográfica .....	68
Anexos.....	72
Glosario de términos.....	72

## Índice de figuras

Figura 1: Esfuerzo en actividades según fase del proyecto.....	19
Figura 2 Árbol de directorios de ExtJs.....	36
Figura 3 Ejemplo de herencia de ExtJs.....	39
Figura 4 Ejemplo de uso de DomQuery para obtener objetos del DOM.....	40
Figura 5 Diagrama de clases del diseño del CUS: Redactar Noticia Personalizada.....	41
Figura 6 Diagrama de clases del diseño del CUS: Administrar Plantilla.....	42
Figura 7 Diagrama de clases del diseño del CUS: Administrar Componente.....	43
Figura 8 Diagrama de clases del diseño del CUS: Administrar Acción.....	44
Figura 9 Diagrama de Componentes: Redactar Noticia Personalizada.....	47
Figura 10 Diagrama de Componentes: Administrar Plantilla.....	48
Figura 11 Diagrama de Componentes: Administrar Componente Visual.....	50
Figura 12 Diagrama de Componentes: Administrar Acción.....	51
Figura 13 Diagrama de Despliegue.....	52

## Índice de tablas

Tabla 1 CUS: Redactar Noticia Personalizada.....	55
Tabla 2 CUS: Administrar Plantilla .....	56

Tabla 3 CUS: Administrar Componente .....58

Tabla 4 CUS: Administrar Acción .....59

## Introducción

Las comunicaciones han sido desde siempre un elemento importante en el desarrollo de la sociedad. Han constituido un medio indispensable para llevar a cabo acciones y decisiones que han cambiado el curso de los eventos político-sociales. El acontecer tecnológico, ha venido aparejado con innumerables avances, que han transformado el mundo humano, principalmente con el uso de las Tecnologías de la Información y las Comunicaciones (TIC). Estas se han introducido en la mayoría de los espacios en la vida del hombre, haciendo que sus utilidades sean requeridas con más fuerza.

La televisión como medio de comunicación, se ha convertido en una herramienta para la mejora de las comunicaciones en la sociedad y ha alcanzado tanto auge que su expansión ha sido casi inevitable. La televisión se ha convertido en uno de los motores fundamentales en la construcción de la Sociedad de la Información. Su alto índice de implantación la convierte en el instrumento perfecto para canalizar el mercado de la información, aspecto necesario para consolidar este nuevo modelo social erigido en torno a la posibilidad que brinda la tecnología digital de una manipulación interactiva de la información. (1)

La digitalización de la información ha sido uno de los avances que han impactado con mayor fuerza en la época contemporánea. Su aplicación ha abarcado casi todas las esferas sociales, demostrando un aumento de eficiencia donde se aplique. Ello ha revelado un mayor acceso a la información, su integridad y la optimización de su almacenamiento. El uso de la televisión en conjunto con las nuevas técnicas para la digitalización de la información, ha creado una novedosa manera de trabajar en el ambiente mediático.

Cuba no ha pasado por alto la relevancia que presentan estos temas y es por ello que ha incursionado en el desarrollo de plataformas de televisión informativa. La Universidad de las Ciencias Informáticas ha obtenido un rol protagónico en las mismas, logrando como resultado diversos sistemas de producción de noticias, como Señal3, la cual fue creada con la misión de mantener informados a la comunidad universitaria residente en la misma. Experiencia que sirvió como base para realizar proyectos con un alcance mayor como fue la creación de Señal ACN diseñada para llevar el acontecer nacional e internacional a los colaboradores cubanos que prestan sus servicios en el extranjero y llegar a los lugares de silencio de la geografía cubana, donde las señales de radio y televisión no habían podido abrirse camino y la prensa se hacía presa de la desactualización, por lo que constituyó todo un éxito la



implantación de dicho sistema, marcando un nuevo hito del desarrollo de las tecnologías de la informática y las comunicaciones en el ámbito nacional.

Con las nuevas tendencias sobre la independencia tecnológica se decide crear una versión sobre plataformas libres. Cumpliendo dicha premisa se inicia el desarrollo de TV Energía, sistema destinado a automatizar los procesos que se realizan para visualizar la información en la red de televisión instalada en el Ministerio del Poder Popular para la Energía y Petróleo de Venezuela. Esta versión de la plataforma se caracteriza por haber sido desarrollada sobre un sistema operativo libre, GNU/Linux, además de que incorporaba nuevas funcionalidades no presentes en sus predecesoras.

A raíz de este éxito nace PRIMICIA, una plataforma de televisión informativa destinada a ser totalmente configurable. PRIMICIA incluye en su diseño objetivos y características de más alcance de manera que se logre una solución personalizable a diferentes gamas de requerimientos. Actualmente PRIMICIA va camino a su segunda versión y la misma cuenta con un módulo de redacción en el subsistema de Administración. Este subsistema posibilita crear las noticias que se transmiten a través del subsistema de transmisión que contiene dicha plataforma. Dicho módulo no permite una completa personalización en relación con las especificidades de cada usuario además que no presenta los suficientes elementos que permitan la configuración de las noticias en el sistema así como la diversidad de sus formas de visualización. Tomando como punto de partida dicha situación se plantea como **problema de investigación**: ¿Cómo lograr la configuración de los componentes visuales de la noticia en el módulo de redacción de la plataforma de televisión informativa PRIMICIA de manera que permita una completa personalización de la noticia?

En el proceso de desarrollo de la presente investigación se define como **objeto de estudio**: el proceso de redacción de la noticia en el subsistema de Administración de la plataforma de televisión informativa PRIMICIA. Para ello se señala como **campo de acción** el módulo de redacción de noticias de la plataforma de televisión informativa PRIMICIA v2.0. Se define como **objetivo general**: implementar el módulo de redacción de la plataforma de televisión informativa PRIMICIA, de manera que posibilite la configuración de los elementos visuales de la noticia, de esta forma se derivan los siguientes **objetivos específicos**:

- Implementación del módulo de redacción de PRIMICIA

- Integración con el generador de formato XML
- Obtención del modelo de Implementación
- Realización de las pruebas de sistema

Para la guía de esta investigación se plantea como **idea a defender** que con la implementación del módulo de redacción de noticias de la plataforma de televisión informativa PRIMICIA V2.0 se garantizará la configuración de los elementos visuales de la noticia.

Para dar cumplimiento a los objetivos específicos se definen como **tareas de investigación**:

- Caracterizar los componentes visuales de la noticia configurable a implementar.
- Caracterizar y seleccionar las herramientas y tecnologías de desarrollo.
- Implementar el módulo de redacción de PRIMICIA V2.0.
- Generar formato XML de las noticias.
- Obtener el modelo de Implementación referente al módulo de redacción de noticias de PRIMICIA V2.0.
- Realizar pruebas de Sistema al módulo de redacción implementado.

Dentro de métodos teóricos que se acoplan al desarrollo de la presente investigación se identifican:

- El analítico sintético: Este método permite que a partir de la documentación e información referente al presente módulo de redacción - del cual se caracterizaron los componentes - se garantice posteriormente generar una propuesta adecuada. Este método es puesto en práctica cuando se estudia el Análisis y Diseño del módulo a implementar. Mediante el mismo se fijan las características que conforman cada componente del módulo. También la manera en que interactúan y se relacionan. Es un estudio donde se toma en cuenta los elementos que se incluyen en el proceso de redacción así como también de sus características y cómo interactúan entre ellos.
- El método inductivo deductivo: Plantea a partir de los resultados anteriores cómo lograr resultados que amplíen el alcance del módulo de redacción de noticias de la plataforma de televisión informativa, de manera tal que garantice la personalización y configuración de la noticia. Este método se evidencia en el momento que se analizan las características que conforman los elementos a implementar. Éstas pueden ser desde atributos hasta funcionalidades, donde se recogen funciones y características específicas. Ello se lleva a cabo para desarrollar características

que generalicen una propiedad o funcionamiento, con el fin de establecer un funcionamiento común entre los componentes así como atributos generalizados en el sistema y decantación de aplicaciones en el mismo.

Entre los métodos empíricos que se utilizan se encuentra:

- Observación no Estructurada: Mediante este método se realiza una observación de los procesos de redacción de la noticia en los sistemas estudiados. A partir de los procesos observados se logra caracterizar de una manera más amena el flujo de la redacción de la noticia, identificando los puntos donde se identifican las partes de dicho proceso a modificar. Para la realización de este método no hizo falta la utilización de técnicas especiales.

# **Capítulo 1: Fundamentación teórica sobre el proceso de redacción y personalización de noticia e interacción de componentes visuales de la plataforma de televisión informativa PRIMICIA**

## **1.1 Introducción**

El presente capítulo está destinado a abarcar las principales cuestiones que conforman la fundamentación teórica de la presente investigación, donde se llega a formar de manera general una percepción futura de la misma. Se realiza un análisis y caracterización de la situación problémica, de manera que se exploren soluciones.

Se realiza un estudio sobre soluciones similares existentes en la literatura universal, nacional y local, que puedan responder a los requerimientos necesarios. Además se realiza un análisis sobre el proceso de personalización de la noticia. Finalmente se describe y caracteriza la metodología de desarrollo a usar, los entornos de desarrollo integrado (IDE por sus siglas en inglés) (2), herramientas de Ingeniería de Software Asistida por Ordenador (CASE por sus siglas en Inglés) (3). También se describe el desempeño del rol de implementador para la implementación de las nuevas funcionalidades del módulo de redacción de noticias de la plataforma de televisión informativa PRIMICIA.

## **1.2 Conceptos asociados a la configuración y personalización en la implementación del módulo de redacción de noticias de PRIMICIA.**

Al llevarse a cabo una tarea de investigación se manejan conceptos que son esenciales para la comprensión del desarrollo de la misma. Por la importancia que presenta para el presente trabajo de investigación se abordará lo referente al concepto de módulo, configuración, implementación entre otros.

### **1.2.1 Módulo**

Según el Diccionario de la Real Academia de la Lengua Española (DRAE) el término módulo, se define:

Un módulo es una pieza o un conjunto unitario de piezas que se repiten en una construcción de cualquier tipo, para hacerla más fácil, regular y económica. (4)

En cuestiones de programación:

- “Un módulo es un software que agrupa un conjunto de subprogramas y estructuras de datos. Los módulos son unidades que pueden ser compiladas por separado y los hace reusables y permite que múltiples programadores trabajen en diferentes módulos en forma simultánea, produciendo ahorro en los tiempos de desarrollo. Los módulos promueven la modularidad y el encapsulamiento, pudiendo generar programas complejos de fácil comprensión”. (5)
- El módulo por lo tanto forma parte de un sistema y mantiene algún tipo de relación o vínculo con el resto de los componentes. Lo modular es fácil de ensamblar y suele ofrecer una amplia flexibilidad no en sus componentes, sino en la manera de armado. Por otra parte, el producto final o sistema puede ser reparado, si se repara el módulo o componente que no funciona. Se conoce como modularidad a la capacidad de un sistema para ser entendido como la unión de varios componentes que interactúan entre sí y que son solidarios, donde cada uno cumple con una tarea en pos de un objetivo común. (6)
- Un módulo es una parte de un programa, que realizará una o varias tareas del total que debe realizar el programa para cumplir con sus objetivos. Normalmente suelen estar organizados jerárquicamente en niveles, de forma que los módulos de nivel superior realizan las llamadas a los de nivel inferior. Cuando un módulo es llamado, recibe como entrada los datos proporcionados por el de nivel superior que ha hecho la llamada, realiza su tarea, y cuando finaliza devuelve la salida pertinente al módulo superior que lo llamó, un módulo puede llamar a otro u otros módulos de nivel inferior o del mismo nivel, pero nunca uno superior. (7)

El módulo constituye una pieza importante en la implementación de aplicaciones, puesto que va a permitir la organización de las funcionalidades y junto con ellas la estructuración de las instrucciones que constituyen su código fuente. Es una pieza a considerar para la reutilización de código.

## **1.2.2 Configuración**

Según DRAE el término configuración se define como:

“la disposición de las partes que componen una cosa y le dan su peculiar forma y propiedades ajenas”. (8)

Una configuración personalizada es aquella que será definida por el usuario, para que el sistema o software se adapte mejor al entorno de trabajo donde será usado, esta es guardada generalmente en un archivo o en una base de datos, puede estar cifrada para que solo se pueda modificar por el programa a

configurar, o puede ser texto plano para que también se pueda modificar sin depender del programa. (7)

La configuración juega un papel muy importante puesto que implica la manera en que se va a organizar un conjunto de elementos, que pueden conformar la forma en que se desenvuelva un sistema. Mediante la configuración se permitirá modificar los valores de atributos, tales como el tamaño, el color entre otros.

### **1.2.3 Implementación**

Según DRAE el término implementación se define como:

“Acción y efecto de implementar.” (9)

La implementación es la realización de una especificación técnica o algoritmos como un programa, componente software, u otro sistema de cómputo. El resultado de la implementación se convierte en el cumplimiento de los requisitos definidos para la construcción de software. Muchas implementaciones son dadas según una especificación o un estándar.

La implementación constituye el ejercicio objetivo en la construcción de sistemas. Mediante la misma se llega a concretar los objetivos y requerimientos en la construcción de software. Con la ejecución de la misma se llegará a una solución que permitirá solventar el problema de investigación planteado. De la misma manera sostendrá como argumento la idea a defender expuesta con relación a la construcción del módulo de redacción.

### **1.2.4 XML: Lenguaje de Marcas Extensible**

El Extensible Markup Language (XML) o "Lenguaje de Marcas Extensible" es un metalenguaje. Es un lenguaje que describe los datos y cómo estos se estructuran, mediante el cual los desarrolladores pueden crear sus propios elementos para alcanzar sus propias necesidades de información. XML se usa fundamentalmente para crear meta vocabularios (conjuntos de etiquetas usados para representar elementos dentro de un documento XML) adaptados a las necesidades de cada empresa, entidad o industria. (10)

Como formato para el almacenamiento de la información que se genera en la creación de una noticia, se escogió el estándar XML. Dicho formato permite la integración entre aplicaciones a distintos niveles, además de ser un formato estandarizado para archivar información de cualquier índole. Como lenguaje provee extensibilidad y una extraordinaria capacidad de validación mediante las Definiciones de Tipo de

Documento (DTD) y XML esquemas. En el presente caso se hizo uso de la extensibilidad y flexibilidad del mismo para elaborar un estándar de noticias que permita la fácil integración y configuración de noticias entre los módulos fundamentales de la plataforma. (10)

Este lenguaje, es comúnmente usado para guardar mediante documentos XML configuraciones que se realizan en los procesos de sistemas. Esta característica es muy usada ya que permite que interactúen entre sí varias aplicaciones usando como una vía de comunicación este tipo de documento. Este lenguaje es usado para integrar, como un protocolo de comunicación, diferentes partes de una aplicación.

### **1.2.5 Redacción Digital Integrada**

Los medios viven hoy un proceso de transformación que atraviesa los flujos de trabajo, los modelos de negocios y los modos de vinculación con las audiencias. La tecnología impone cambios en las reglas de juego y las organizaciones periodísticas responden embarcándose en el desafío de la innovación.

La llegada de las computadoras ha marcado un nuevo hito en el desarrollo social de la humanidad, ha protagonizado un gran porcentaje de los nuevos avances científicos- tecnológicos que acontecen en la actualidad. La utilidad de las computadoras en el periodismo ha revolucionado una nueva forma de concebir el mismo, al tener la entrada de la misma en las salas de redacción y la combinación con nuevas tecnologías propicia el surgimiento de nuevos conceptos tales como redacción digital integrada. (11)

La redacción digital integrada permite capturar los textos, imágenes, sonidos y video de las informaciones y almacenarlas en servidores. Los profesionales de la información pueden consultar entre esta variedad de datos y de esta manera confeccionar y actualizar de manera dinámica las noticias. Lo cual constituye un aumento de la calidad y la eficiencia, en procesos relacionados con esta rama.

### **1.2.6 Módulo de redacción de 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:

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 entornos<sup>1</sup> 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. (12)

Es por ello que el módulo de redacción, en su versión actual no brinda la posibilidad de la personalización de las noticias, por lo que la misma no se transmite con todo lo que se requiere. En la segunda versión de la plataforma se proponen funcionalidades que permitirán la creación de una noticia personalizable. Una noticia personalizable es aquella a la que se le configura la disposición y visualización de los componentes que la forman. Este tipo de noticia trae consigo diversas características que son manejadas según las necesidades del redactor, entre las que se incluyen efectos a sus componentes entre otras. Estas fueron especificadas en el análisis y diseño para la implementación del módulo de redacción, entre estas se encuentran:

- Edición de los elementos y recursos multimedia de la noticia. (12)
- Incorporación de efectos de animación a las noticias. (12)
- Pre-visualización en todo momento de la edición en curso. (12)

### **1.2.7 Generador de formato XML**

En el proceso de redacción de la noticia es necesario exportar las configuraciones realizadas cuando se crea una noticia, de manera que esta personalización quede, de cierta manera, en un medio persistente. Una vía factible para ello lo constituye guardar dichas configuraciones en un archivo. A través de este se permitirá guardar futuros cambios a las noticias y su configuración. Para ello se propone el uso de una

---

<sup>1</sup> 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.



API<sup>2</sup> para la generación de archivos en formato XML que logre conservar dichos datos en el tiempo.

Para exportar el formato XML de noticias de la plataforma de televisión informativa PRIMICIA se hará uso de las facilidades que brinda el modelo de objetos de documento (DOM), mediante el lenguaje PHP<sup>3</sup>. El mismo cuenta con un conjunto de clases y métodos para el manejo del formato XML. (10)

De manera que se propone integrar el generador de formato XML con el módulo de redacción de la plataforma de televisión informativa, PRIMICIA.

### **1.3 Análisis del objeto de estudio**

Este epígrafe abordará temas acerca de las soluciones similares a PRIMICIA que han sido desarrolladas tanto en el ámbito nacional como internacional. Se realizará un estudio de las principales funcionalidades que presentará el módulo de redacción. Además de una descripción del proceso de redacción de la noticia.

#### **1.3.1 Soluciones similares a PRIMICIA de ámbito internacional**

En el orbe existen varias empresas como televisoras, que hacen uso de las utilidades que facilitan la tecnología de los sistemas de producción de noticias, confirmando con la experiencia la factibilidad que provee. Entre empresas que han incursionado en la misma se encuentran

BBC<sup>4</sup>: esta lanzó BBC News 24, esta logró una completa digitalización y una nueva estructura de los servicios informativos, transformando los modelos que conservaban la brecha que se sostenía entre las vías de redacción principal del canal, las de radio e Internet.

Esta plataforma cuenta con un sistema capaz de automatizar transmisiones en directo, la redacción dispone de cuarenta y siete terminales en las que el periodista puede editar. Este sistema permite transferir clips de noticias producidos por el canal de 24 horas a los servidores de la redacción.

SONAPS: es un entorno en red que integra todos los aspectos del proceso de producción de informativos,

---

<sup>2</sup> API: Application Programming Interface por sus siglas en inglés. Interfaz de programación de aplicaciones.

<sup>3</sup> PHP: PHP hypertext preprocessor por sus siglas en inglés. lenguaje de programación interpretado, diseñado originalmente para la creación de páginas Web dinámicas

<sup>4</sup> BBC: British Broadcasting Company por sus siglas en Inglés. Corporación Británica de Radiodifusión

desde la planificación, captación y edición, hasta la emisión, distribución y archivo. Permite que periodistas, editores profesionales y productores puedan compartir contenido y metadatos en un colaborativo flujo de trabajo en red basado en ficheros. El sistema Sonaps transforma el proceso de producción de noticias integrando todos los aspectos del proceso de producción en un completo y sencillo flujo de trabajo. Una vez ya se han acordado y planificado las historias, Sonaps se encarga de todo desde la búsqueda de contenido en el Archivo hasta la gestión de los datos de planificación para los reporteros (13)

En la sala de Redacción, Sonaps se integra por completo con los sistemas de informativos, desde la etapa de planificación pasando por la edición y emisión hasta el archivo. Los contenidos dentro de Sonaps y en sistemas de archivo profundos o near-line de terceras partes, pueden ser buscados y visionados fácilmente. (14)

ENEWSROOM: es un sistema de producción de noticias y contenidos audiovisuales diseñado para facilitar el trabajo de periodistas y usuarios. Este sistema permite una redacción desde varios clientes, los cuales son centralizados en un servidor central, de manera que se publican las noticias de cada redactor. Solución usada principalmente en empresas comerciales. (15)

Estos sistemas cumplen con algunas de las características requeridas para la plataforma pero su uso implicaría el pago de licencias y dependencias con otras herramientas.

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. (12)

Aunque abarcan una gama de funcionalidades ya implementadas en PRIMICIA, estos sistemas no cumplen con una completa personalización de los procesos de redacción, incluirlos aumentaría el costo de producción de la plataforma. Es por ello que la implementación del módulo de redacción de noticias, abarcaría estas funcionalidades agregando las que se definieron en el diseño para la segunda versión de la misma.

### **1.3.3 Soluciones similares a PRIMICIA de ámbito nacional**

Señal ACN<sup>5</sup>, sistema automatizado de teletexto para la plataforma de televisión digital satelital cubana, constituye una plataforma con el objetivo de acercar la información a aquellos colaboradores cubanos, que prestan servicios fuera de nuestras fronteras, además de llegar a las zonas de silencio, lugares donde la actualidad se abría paso paulatinamente.

Señal ACN no tiene concebido los nuevos elementos que se desean integrar al proceso de redacción de PRIMICIA. Este sistema no permite la configuración de los elementos que componen la noticia.

Señal3: Sistema de producción de noticias desarrollado en la Universidad de las Ciencias Informáticas. Diseñado para mantener informado a la comunidad universitaria. Este sistema fue desarrollado sobre tecnología privativa como base.

TV Energía: Resultado devenido de la investigación sobre el estudio de sistemas informáticos sobre tecnologías libre para la transmisión. Este sistema fue destinado a automatizar los procesos que se realizan para visualizar la información en la red de televisión instalada en el Ministerio del Poder Popular para la Energía y Petróleo de Venezuela.

Estos sistemas nacionales constituyen los precedentes de PRIMICIA, estos productos cuentan con la posibilidad de acceso, actualización y presentación de los recursos de multimedia en el proceso de redacción de la noticia, sin embargo carecen de la opción de configurar la misma, tampoco incorporan la capacidad de adicionar nuevos componentes a la noticia que se crea en el sistema.

### **1.3.4 Plataforma de televisión informativa PRIMICIA**

Tomando como experiencia las soluciones nacionales surge PRIMICIA. La plataforma de televisión informativa PRIMICIA, es un producto desarrollado en el centro productivo Geoinformática y Señales Digitales (GEYSED), específicamente en el Departamento de señales digitales. La cual confiere un canal de televisión con transmisión automática y regular de información en diversas clasificaciones de formatos, unificando videos, textos, imágenes con el propósito de mostrar de manera más amena la noticia. Esta plataforma puede ser ajustada a cualquier cliente que posea la necesidad de mantener informados

---

<sup>5</sup> ACN: Agencia Cubana de Noticias

constantemente a determinada cantidad de personas. 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.

### **1.3.5 Proceso de redacción de la noticia**

Mediante el proceso de redacción de la noticia, es posible redactar y conformar los elementos que se quieren transmitir en determinada entidad que esté usando la plataforma. Cada institución o empresa tiene características visuales que difieren para cada una, la transmisión de noticias. Esto acarrea que la presentación de la noticia así como el proceso de redacción sea diferente en cada entidad.

Sin embargo este proceso se caracteriza por ser poco flexible y esquemático ya que la plataforma no le permite al redactor interactuar con los elementos visuales de la noticia. Por ejemplo, mediante el proceso de redacción no se le permite al redactor definir nuevos elementos para la presentación de la noticia, o bien no puede cambiar ni eliminar los que vienen definidos por defecto. Esto es debido a que las noticias están compuestas por pantallas y estas son sólo cuatro tipo: texto, texto-imagen, imagen y video, por lo que cualquier noticia que combine otros dos recursos actualmente no se puede elaborar. Por último otra deficiencia es la imposibilidad por parte del redactor de configurar el tiempo que se visualizará la pantalla, este valor es estático y se define internamente durante el desarrollo de la aplicación. (16)

El subsistema de administración está compuesto por una variedad de módulos que en su conjunto, posibilitan el funcionamiento del mismo como un todo. Entre los módulos existentes es el módulo de redacción el que contiene mayor responsabilidad en el proceso de redacción de la noticia. El módulo de redacción está orientado a llevar a cabo la redacción y edición de las informaciones conjuntamente con los recursos de multimedia que conforman la noticia, así como su disposición en la misma. Debe tener la posibilidad de poder guardar un archivo persistente, las configuraciones de la noticia. Creando una

manera factible para su edición y visualización conjuntamente con las funciones del subsistema de transmisión.

El módulo de redacción de PRIMICIA no brinda la posibilidad de adaptarse a los variables entornos<sup>6</sup> 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.

### **1.3.6 Funciones para el módulo de redacción**

Con el propósito de mejorar la funcionalidad del módulo de redacción se concibieron varias características que pueden ser adecuadas a su perfil. Para lograr estas funcionalidades se tienen predefinidas varias acciones, de las que se mencionan:

- 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. (17)
- 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. (17)
- Crear todos los componentes y herramientas necesarios para la creación de la noticia. (17)
- Crear un banco de efectos, que sean aplicables a todos los elementos configurables y que se puedan configurar desde la ventana de edición. (17)

Estas acciones, integradas a las tareas de investigación, forman una gama de funcionalidades a tomar en cuenta en el momento de desarrollar dicho módulo.

---

<sup>6</sup> 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.

## **1.4 Herramientas, Técnicas y tecnologías**

Generalmente el desarrollo de un producto o proyecto es guiado por una metodología de desarrollo, lo cual constituye un marco de trabajo que es usado para estructurar, planear y controlar el proceso de desarrollo en sistemas de información.

No obstante, cada software cuenta con necesidades y entornos diferentes, que deben tenerse en consideración. Por otra parte cada metodología tiene características propias del ambiente de desarrollo, constituyendo estos los principales aspectos que rigen el proceso de selección de la metodología adecuada. Una metodología tiene que ser usada de forma simple y efectiva, sin hacer las tareas de diseño excesivamente complejas (18)

### **1.4.1 Metodología de desarrollo de software RUP<sup>7</sup>**

RUP es un proceso de desarrollo de software que obtiene las mejores prácticas del conocimiento de líderes en ingeniería de software y proporciona a los equipos de desarrollo guías, estándares y recomendaciones para la construcción de software de alta calidad.

Estas prácticas de desarrollo de software están documentadas como principios clave, en donde cada una de ellas corresponde a distintos aspectos del desarrollo de software que generalmente requieren habilidades específicas, esto se refleja en los roles y productos de trabajo de cada uno de ellos. Entre las principales características que presenta, está dirigido por los casos de uso, está centrado en la arquitectura y es iterativo e incremental.

Dirigido por los casos de uso

Los casos de uso son una técnica de captura de requisitos que fuerza a pensar en términos de importancia para el usuario y no sólo en términos de funciones que sería bueno contemplar. Se define un caso de uso como un fragmento de funcionalidad del sistema que proporciona al usuario un valor añadido. Los casos de uso representan los requisitos funcionales del sistema, en RUP los casos de uso no son sólo una herramienta para especificar los requisitos del sistema. También guían su diseño, implementación y prueba. Los casos de uso constituyen un elemento integrador y una guía del trabajo. (19) Es importante que los casos de uso, representen los conjuntos de procesos que intervienen en la confección de cierta

---

<sup>7</sup> RUP: Rational Unified Process por sus siglas en inglés. Proceso Unificado de Desarrollo

funcionalidad o funcionalidades con la que cuenta el sistema.

Los casos de uso no sólo inician el proceso de desarrollo sino que proporcionan un hilo conductor, permitiendo establecer trazabilidad entre los artefactos que son generados en las diferentes actividades del proceso de desarrollo. (19)

#### Centrado en la Arquitectura

En un sistema la arquitectura constituye la estructura fundamental del mismo, define los componentes más relevantes y conforma una visión más amena del sistema en cuestión.

La arquitectura involucra los aspectos estáticos y dinámicos más significativos del sistema, está relacionada con la toma de decisiones que indican cómo tiene que ser construido el sistema y ayuda a determinar en qué orden. Además la definición de la arquitectura debe tomar en consideración elementos de calidad del sistema, rendimiento, reutilización y capacidad de evolución por lo que debe ser flexible durante todo el proceso de desarrollo. La arquitectura se ve influenciada por la plataforma software<sup>8</sup>, sistema operativo, gestor de bases de datos, protocolos, consideraciones de desarrollo como sistemas heredados. Muchas de estas restricciones constituyen requisitos no funcionales del sistema. (19) En RUP, se decanta una gran fuerza en la construcción de la arquitectura, ya que esta implica la poca relevancia que puedan tener cambios futuros en el sistema.

#### Iterativo e incremental

El desarrollo de un producto, lleva consigo un gran esfuerzo, por lo que se considera práctico dividir el trabajo en partes más pequeñas, donde cada parte forma una iteración, donde las iteraciones hacen referencia al flujo de trabajo. En cada iteración los desarrolladores identifican los casos de uso críticos. Se implementa un diseño mediante componentes, verificando que satisfagan las especificaciones de los casos de uso. (19)

Una iteración puede realizarse por medio de una cascada. Se pasa por los flujos fundamentales (Requisitos, Análisis, Diseño, Implementación y Pruebas), también existe una planificación de la iteración, un análisis de la iteración y algunas actividades específicas de la iteración. Al finalizar se realiza una integración de los resultados con lo obtenido de las iteraciones anteriores. (19)

El proceso iterativo e incremental consta de una secuencia de iteraciones. Cada iteración aborda una

---

<sup>8</sup> Una plataforma software es un conjunto de artefactos (componentes o subsistemas) que forman una estructura común a partir de la cual podemos derivar (desarrollar, construir) sistemas de una forma eficiente

parte de la funcionalidad total, pasando por todos los flujos de trabajo relevantes y refinando la arquitectura. Cada iteración se analiza cuando termina. Se puede determinar si han aparecido nuevos requisitos o han cambiado los existentes, afectando a las iteraciones siguientes. Durante la planificación de los detalles de la siguiente iteración, el equipo también examina cómo afectarán los riesgos que aún quedan al trabajo en curso. Toda la retroalimentación de la iteración pasada permite reajustar los objetivos para las siguientes iteraciones. Se continúa con esta dinámica hasta que se haya finalizado por completo con la versión actual del producto. (19)

### Fases

RUP divide el proceso en cuatro fases, dentro de las cuales se realizan varias iteraciones en número variable según el proyecto y en las que se hace un mayor o menor hincapié en los distintas actividades.

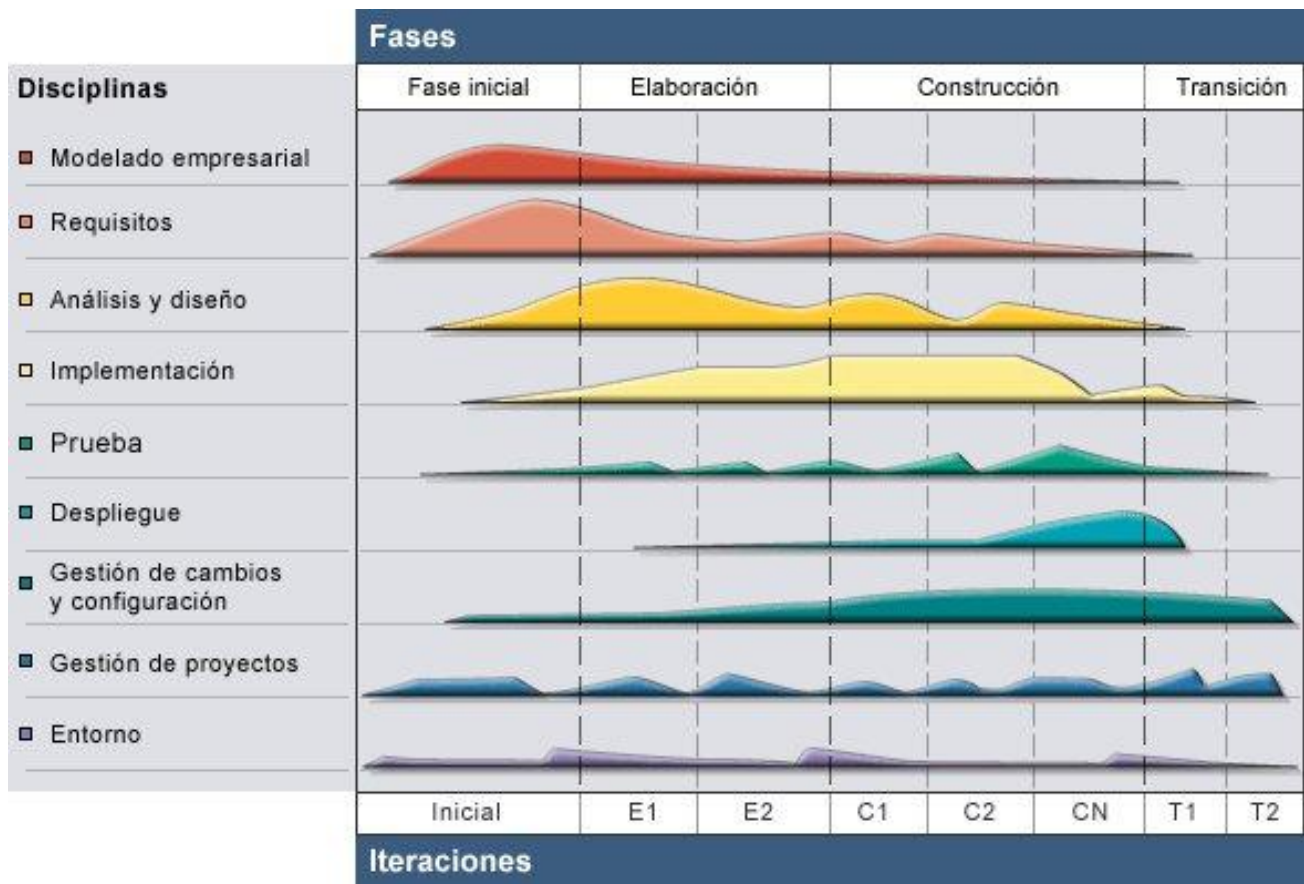


Figura 1: Esfuerzo en actividades según fase del proyecto



Durante la fase de inicio las iteraciones ponen mayor énfasis en actividades modelado del negocio y de requisitos.

En la fase de elaboración, las iteraciones se orientan al desarrollo de la línea base de la arquitectura, abarcan más los flujos de trabajo de requerimientos, modelo de negocios (refinamiento), análisis, diseño y una parte de implementación orientado a la línea base de la arquitectura.

En la fase de construcción, se lleva a cabo la construcción del producto por medio de una serie de iteraciones.

Para cada iteración se selecciona algunos casos de uso, se refina su análisis y diseño y se procede a su implementación y pruebas. Se realiza una pequeña cascada para cada ciclo. Se realizan tantas iteraciones hasta que se termine la implementación de la nueva versión del producto.

En la fase de transición se pretende garantizar que se tiene un producto preparado para su entrega a la comunidad de usuarios. (19)

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.

Para el desarrollo de la implementación del módulo de redacción de la plataforma de televisión informativa PRIMICIA, se utilizará RUP como metodología. Esta metodología no depende de sistema operativo alguno, lo que su materialización es multiplataforma. RUP permite utilizarse en el desarrollo de proyectos donde el personal no cuenta con la mayor experiencia en la producción de software además de permitir flexiblemente el cambio de roles.

RUP a diferencia de las metodologías ágiles no depende de la presencia constante del cliente, aspecto fundamental para el desarrollo de la implementación de dicho módulo. Es la metodología ideal para el desarrollo de proyectos a largo plazo, su documentación es lo suficientemente detallada para cumplir el seguimiento de proyectos desarrollados bajo dicha metodología.

## 1.4.2 UML<sup>9</sup>

El Lenguaje Unificado de Modelado 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. (20)

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. (20)

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. (12)

A través del uso de este lenguaje de modelado es posible obtener una abstracción del módulo a desarrollar. A través del modelado se generan artefactos que aseguran la continuidad de desarrollo, permitiendo que se entienda como está diseñadas las principales características del módulo a implementar. Facilita a integrantes de un equipo multidisciplinario participar e comunicarse fácilmente, estos integrantes siendo los analistas, diseñadores, especialistas de área y desde luego los programadores.

## 1.4.3 Técnicas del Implementador

Este rol desarrolla los componentes de software y efectúa las pruebas de desarrollador para la integración en subsistemas más grandes, de acuerdo con los estándares adoptados de proyecto. El mismo contiene varias tareas, entre las que se destacan:

- Analizar el comportamiento en tiempo de ejecución: En esta tarea se describe cómo analizar el comportamiento de un componente durante su ejecución para identificar las mejoras que se

---

<sup>9</sup> UML: Unified Modeling Language por sus siglas en inglés, Lenguaje Unificado de Modelado

pueden realizar.

- **Desarrollar productos de trabajo de instalación:** En esta tarea se describe cómo producir todo el software necesario para instalar y desinstalar el producto de forma rápida, sencilla y segura, sin afectar a las demás aplicaciones o características del sistema.
- **Ejecutar pruebas de desarrollador:** En esta tarea se describe cómo ejecutar y evaluar un conjunto de pruebas diseñadas para validar que el componente funciona correctamente antes de que se realicen más pruebas formales en el componente.
- **Implementar elementos de diseño:** En esta tarea se describe cómo producir una implementación para una parte del diseño (por ejemplo, una clase, una realización de guión de uso o una entidad de base de datos), o para solucionar uno o varios defectos. El resultado son archivos nuevos o modificados de datos y de código fuente, que se conocen generalmente como elementos de implementación.
- **Implementar la prueba de desarrollador:** En esta tarea se describe cómo crear un conjunto de pruebas para comprobar que el componente funciona correctamente antes de que se realicen más pruebas formales en él.
- **Implementar los elementos de comprobabilidad:** En esta tarea se describe cómo implementar funcionalidad especializada para dar soporte a los requisitos específicos de la prueba. (16)

#### **1.4.4 Estándar de codificación**

En el centro productivo GEYSED, se establece para el desarrollo de la implementación un estándar de codificación dado a las ventajas que brinda adoptar el mismo. Adoptar un estándar permite enfocarse más en el código que en el formateo del mismo porque a la larga se vuelve mecánico “Así se escribe el código y punto”. Asimismo, aporta consistencia pues las diferentes porciones de código (que pueden ser provenientes de diversos lenguajes) guarda un patrón que se puede reconocer. Mejora la legibilidad del código. Permite que otras personas puedan colaborar más eficazmente pues la redacción del código no

les resulta incómoda. En consecuencia, el mantenimiento es menos pesado como resultado de los aportes anteriormente mencionados. (21)

Los estándares definidos incluyen:

### *Principios generales*

- Los nombres de cada uno de los elementos del programa deben ser significativos.
- No manejar en los programas más de una instrucción por línea.
- Declarar las variables en líneas separadas.
- Añadir comentarios descriptivos junto a cada declaración de variables, si es necesario.
- La mayoría de los elementos se deben nombrar usando sustantivos.
- Los atributos deben comenzar con letra minúsculas.
- Los métodos deben comenzar con letra mayúsculas.

### *Sobre el nombramiento de clases, interfaces y archivos fuentes*

Los nombres deberán estar en sustantivo singular, con la primera letra en mayúscula y las demás en minúsculas. Si el nombre de la clase está comprimido en más de una palabra la primera letra debe de ser mayúscula. Letras mayúsculas seguidas no son permitidas.

Ejemplo: “Zend\_PDF” no es permitido mientras que “Zend\_Pdf” si lo es.

Las clases deben comenzar con un prefijo alegórico al componente que correspondan, evitando nombres duplicados.

Ejemplo: “Controles” no es un nombre permitido mientras que “Reproductor\_Controles” si lo es para el componente Reproductor.

Las API deben comenzar con un prefijo alegórico al componente que correspondan, evitando nombres duplicados y terminar en el sufijo `_Component`.

Ejemplo: “Controles” no es un nombre permitido mientras que

“Reproductor\_Controles\_Component” sí lo es.

Las variables comenzarán siempre con la primera letra minúscula correspondiente a su tipo de dato.

Para distinguir palabras dentro del nombre deberá emplearse un guión bajo (`_`).

Ejemplo: `formato_de_video`

Los comentarios representan el objetivo de la función y no descripción del procedimiento.

Se deberá presentar una explicación de uso de argumentos (parámetros). Además de una explicación de uso de valores devueltos (de retorno).

## **1.4.5 Herramientas y Lenguajes propuestos**

La selección de herramientas y lenguaje se basó principalmente en el documento de arquitectura actual del proyecto PRIMICIA. Los mismos se describen a continuación:

### JavaScript

Un lenguaje popular que es ampliamente apoyado en navegadores y otras herramientas Web. Agrega funciones interactivas de páginas HTML, que son otra cosa estática. JavaScript se ocupa principalmente del trabajo con los elementos en la página Web. En el cliente, JavaScript se mantiene como código fuente incrustado en una página HTML. (22)

JavaScript podría utilizarse para mostrar una entrada de datos y validar la misma. JavaScript se destaca como uno de los principales lenguajes del lado del cliente, desarrollando con gran fuerza la representación de interfaces gráficas de usuarios, es por ello que con este fin se selecciona este lenguaje para el desarrollo del módulo de redacción de la plataforma de redacción PRIMICIA.

## ExtJs

ExtJs es un API escrito en Java Script con la finalidad de asistir el desarrollo de aplicaciones enriquecidas para internet. Entre las principales características de ExtJs resalta un fuerte paradigma basado en componentes soportado por recursos para la programación orientada a objetos en Java Script que facilitan la implementación de extensiones y aplicaciones de gran complejidad, el poseer un amplio conjunto de componentes configurables de alta calidad y una implementación transparente y sencilla para el trabajo con AJAX. (23)

ExtJs es usado como framework de presentación en el desarrollo de proyectos Web que conforman las RIA<sup>10</sup>, dado que sus características tributan a que pueda ser utilizado con estas finalidades. Para el desarrollo del presente trabajo se usará ExtJs v3.0, versión estable de dicho producto. Esta versión está bajo la licencia de la GPL.

## PHP

(Pre-procesador Hipertexto) es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas Web dinámicas. La mayoría de su sintaxis está prestada de los lenguajes de programación C, Java y Perl, con la inclusión de algunos rasgos únicos de PHP. La meta del lenguaje es permitir a los que desarrollan sitios Web escribir rápidamente páginas generadas dinámicamente. (24) Este lenguaje constituye la base del subsistema de Administración de PRIMICIA, ya que casi todo su desarrollo concerniente a la tecnología del servidor responde a su uso.

## Symfony

Symfony es un framework para construir aplicaciones Web con PHP. En otras palabras, Symfony es un enorme conjunto de herramientas y utilidades que simplifican el desarrollo de las aplicaciones Web. Symfony emplea el tradicional patrón arquitectónico MVC (modelo-vista-controlador) para separar las distintas partes que forman una aplicación Web. (25) Symfony constituye una poderosa herramienta que facilita el trabajo para la construcción de aplicaciones Web. El subsistema de Administración como aplicación Web usa las ventajas de Symfony con el fin de reducir el tiempo de desarrollo. Symfony al emplear MVC favorece a un mantenimiento más sencillo de las aplicaciones.

---

<sup>10</sup>RIA: Rich Internet Application por sus siglas en Inglés, Aplicaciones enriquecidas para Internet

## Doctrine ORM<sup>11</sup>

Doctrine es un ORM para PHP 5.2.3 y posterior. Además de todas las ventajas que conlleva un ORM, uno de sus puntos fuertes es su lenguaje DQL (Doctrine Query Language) inspirado en el HQL de Hibernate. (30) Entre muchas otras cosas tienes la posibilidad de exportar una base de datos existente a sus clases correspondientes y también a la inversa, es decir convertir clases (convenientemente creadas siguiendo las pautas del ORM a tablas de una base de datos. (26) Con el uso de doctrine el control de datos podrá ser efectuado de manera fácil, permitiendo transformar el uso de consultas a base de datos a un lenguaje sencillo. El subsistema de Administración al usar Doctrine permite manejar consultas a un nivel muy sencillo.

## NetBeans

NetBeans es un entorno de desarrollo integrado (IDE) multiplataforma. El proyecto NetBeans está formado por un IDE de código abierto, libre, gratuito y sin restricciones de uso. Ha logrado un gran éxito con una inmensa base de usuarios. NetBeans provee una estructura para los proyectos que se pueden crear junto a este IDE. Propone un esqueleto para organizar código fuente, el editor conjuntamente integra los lenguajes como HTML, JavaScript y CSS. Con NetBeans se sustituye la consola de comandos de Symfony permitiendo centrarse en desarrollar en el IDE. Además se encuentra cargadas todas las clases, ayuda en línea, etc. Tiene la posibilidad de contar con la integración de sistemas de control de versiones, tales como SVN, CVS, Mercurial y Git.

## Aptana Studio

Aptana Studio, IDE basado en Eclipse constituye una herramienta que reúne las características más completas para el desarrollo de aplicaciones Web basadas principalmente basadas en HTML, DOM, JavaScript y CSS.

Entre las funciones más destacadas de Aptana Studio se puede mencionar:

---

<sup>11</sup> Un ORM o (Object Relation Mapper) es una técnica de programación que permite convertir datos entre el sistema de tipos utilizado en un lenguaje de programación orientado a objetos y el utilizado en una base de datos relacional, es decir, las tablas de la base de datos pasan a ser clases y los registros objetos se pueden manejar con facilidad. (26)

- Asistente de código para HTML y Java script.
- Librerías Ajax (jQuery, Prototype, Scriptaculous, ExtJs, Dojo, YUI y Spry entre otras).
- Conexión vía FTP, SFTP, FTPS y Aptana Cloud.
- Herramientas para trabajo con base de datos.
- Marcado de sintaxis mediante colores.
- Compatible con extensiones para Eclipse (existen más de 1000).

Aptana Studio 0.29 es gratuito, de código abierto y multiplataforma, por lo que puede instalarse en cualquier PC con sistemas operativos Windows, Linux o Mac OSX. (27)

Aptana brinda muchas funcionalidades para el desarrollo de aplicaciones Web. Entre las ventajas que posee Aptana para el desarrollo de Web se encuentra la alto completamiento de código JavaScript, logra identificar la compatibilidad del código con los navegadores más usados. Al incluir librerías Ajax, brinda la posibilidad de reutilización del código de las mismas. Estas características son importantes a considerar dado el hecho de que el subsistema de Administración de PRIMICIA es una aplicación Web, lo que representa muchas ventajas en este campo.

### Firebug

Es un depurador Web, embebido dentro de navegadores como Mozilla Firefox, cuya máxima funcionalidad reside en monitorear los eventos JavaScript, Ajax además del DOM, HTML y CSS, incluye una consola de comandos y variedad de inspectores Web. (28)

Esta herramienta es muy usada para realizar pruebas unitarias. A través de la misma se puede probar y confirmar la factibilidad del código realizado, principalmente el de las interfaces JavaScript con ExtJs.

### Visual Paradigm

Visual Paradigm es una herramienta profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El software de modelado UML ayuda a una más rápida construcción de aplicaciones de calidad y a un menor costo. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. (29)



Visual Paradigm 6.4, es una herramienta potente, ya que posibilita la realización tanto de ingeniería inversa como directa. Se integra con varios Entornos de Desarrollo Integrado. Funciona también como herramienta colaborativa, soportando un equipo de trabajo sobre un mismo proyecto, incluyendo la capacidad del control de versiones, elemento a considerar con proyectos con un vasto equipo de desarrollo como lo es PRIMICIA.

## 1.5 Conclusiones

En el capítulo se analizaron los conceptos que sustentan el problema científico planteado, donde intervienen conceptos tales como módulo, configuración e implementación. Los mismos constituyen conceptos que ayudan a comprender el desarrollo de la investigación. Se realizó un análisis del objeto de estudio, concluyendo que:

1. El análisis de las soluciones existentes tanto en el ámbito nacional e internacional permitió concluir que las características de dichos sistemas no cumplían con los requerimientos especificados para el proceso de redacción de la noticia. Debido a ello se determina la implementación del módulo de redacción.
2. El uso de RUP como metodología de desarrollo permite la generación de los artefactos necesarios en el flujo de Implementación, este proceso es vital para un mayor entendimiento de la aplicación a desarrollar.
3. Con el uso del lenguaje de modelado UML se crea una abstracción del sistema a implementar, permitiendo describir las funcionalidades básicas. Ello hace posible también la continuidad de desarrollo.
4. El uso de frameworks como ExtJs y Symfony permite una construcción bien estructurada de la aplicación, optimizando el uso de lenguajes como JavaScript y PHP. De la misma manera que el trabajo de los mismos a través de entornos de desarrollo y herramientas de depuración de código como el Firebug hacen más sencillo el acto de implementar.
5. Mediante los estándares de codificación se obtiene legibilidad en el código permitiendo reutilización del mismo así como la posibilidad de actualización de algoritmos y funcionalidades.

## **Capítulo 2 Descripción de la solución propuesta.**

### **2.1 Introducción**

En el siguiente capítulo realiza se una descripción del diseño que se implementa, perteneciente al módulo de redacción. Se hace referencia a los artefactos que constituyen la base para el flujo de la implementación. En los artefactos se encuentra el Diagrama de Clases y el Modelo de Implementación.

### **2.2 Requisitos Funcionales**

Los requisitos funcionales son características expresadas de lo que necesita el sistema, estos expresan funcionalidades del mismo. Básicamente los requisitos funcionales establecen los comportamientos del sistema. Un requisito funcional típico contiene un nombre y un número de serie único y un resumen. Esta información se utiliza para ayudar al lector a entender por qué el requisito es necesario, y para seguir al mismo durante el desarrollo del producto. Los requisitos funcionales definidos para el módulo de redacción son los siguientes:

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.

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: Guardar noticia. Debe permitir guardar en formato XML los cambios realizados durante la confección de la noticia.

RF 1.8: Pre visualizar 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 3: Visualizar recursos: Visualizar las acciones y efectos en los canales.

RF 4: Administrar componentes. El sistema debe permitir administrar los componentes que desea incluir en la noticia.

RF 4.1 Agregar componente. Debe permitir seleccionar un componente y desplazarlo hacia el área de trabajo, donde se visualizará con las propiedades predefinidas para este.

RF 4.2 Eliminar componente. Debe permitir eliminar un componente que haya sido agregado al área de trabajo.

RF 4.3 Ajustar propiedades de componente. Debe permitir realizar cambios en las propiedades predefinidas para cada componente.

RF 5: Administrar plantillas. El sistema debe permitir utilizar las plantillas definidas y eliminar su uso en la noticia.

RF 5.1: Seleccionar plantillas. Debe permitir seleccionar la plantilla que se va a visualizar en el área de trabajo.

RF 5.2 Eliminar plantillas. Debe permitir eliminar del área de trabajo la plantilla seleccionada.

RF 5. Administrar efecto. El sistema debe permitir gestionar los efectos aplicados a cada acción.

RF 5.1: Seleccionar efecto. Debe permitir aplicar un efecto para la acción seleccionada.

RF 5.2: Eliminar efecto. Debe permitir eliminar el efecto aplicado a una acción.

RF 5.3: Modificar propiedades de efecto. Debe permitir configurar las propiedades del efecto aplicado.

## 2.3 Descripción del Diseño

En el proceso de redacción de la noticia intervienen varios elementos, los componentes, que son los responsables de configurar sobre ellos las acciones y la forma en que la misma se conforma. Estos elementos, presentan ciertas propiedades que son las que se configuran mediante el proceso mencionado.

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. (12)

Estos elementos se describen a continuación:

### 1. Imagen

Es un componente que estará referenciando a un archivo de imagen. Sus principales propiedades son: posición, tamaño, visibilidad, superposición, opacidad, identificador y máscara. Este componente mostrará la imagen referenciada en la ventana de edición de la noticia, contiene además responsabilidades para su disposición, como cambiar de tamaño ó posición. Presenta efectos que le son aplicables durante la visualización de la noticia.

### 2. Vídeo

Es un componente que estará referenciando a un archivo de vídeo. Sus principales propiedades son: posición, volumen, tamaño, superposición, intervalo, visibilidad, repetición, opacidad e identificador. El mismo a través de la ventana de edición de la noticia mostrará al archivo referenciado, incluyendo los controles de ejecución.

### 3. 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. Dicho elemento es creado en el módulo mediante la participación del redactor. Tiene la capacidad de cambiar su ubicación en la escena, además se le pueden incorporar efectos de transición.

#### 4. Audio

Es un componente que estará referenciando a un archivo de audio. Sus principales propiedades son: volumen, repetición, título, icono, intervalo e identificador. Este elemento sólo es visible mediante un icono señalizador que indica su presencia.

#### 5. 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. Incluye en su composición la transición.

#### 6. Estado

Este componente está relacionado con los elementos de la noticia, puesto que el mismo representará los estados por los cuales transitarían todos los elementos de la noticia en el momento de su sincronización.

#### 7. Time-Line (Línea de Tiempo)

Este componente presenta la responsabilidad de sincronizar los estados de los demás componentes mediante transiciones a través del tiempo, mediante la inicialización y finalización de la noticia.

#### 8. 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.

#### 9. 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.

#### 10. Tiempo

Sus principales propiedades son: tamaño, posición, superposición, visibilidad y formato. Este componente puede indicar tiempo transcurrido o tiempo restante, según se seleccione.

#### 11. 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.

#### 12. 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.

#### 13. 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.

#### 14. Próxima-Noticia

Indicará el título de la siguiente noticia. Su principal propiedad es el nombre de la siguiente noticia.

#### 15. 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.

La combinación de los componentes, la manera en que se configuran y las acciones que se le apliquen son pasos a tener en cuenta en la personalización de la noticia. Para la implementación de los mismos se toman como base artefactos como los diagramas de clase del diseño.

## 2.4 Análisis de los frameworks de desarrollo Symfony y ExtJs

En la actualidad el uso de frameworks de desarrollo con el objetivo de construir aplicaciones se ha generalizado con una vertiginosa velocidad. Ello ha traído consigo un desarrollo rápido eficiente de aplicaciones, permitiendo la reutilización de código y actualización del mismo. En el ámbito del desarrollo Web son Symfony y ExtJs una combinación usada con el fin recrear de manera rápida y concisa aplicaciones de este tipo.

### 2.4.1 Symfony

Como se mencionaba en el epígrafe 1.4.5, Symfony es un framework para el desarrollo de aplicaciones Web escritas en PHP. Symfony usa el patrón arquitectónico MVC, para separar las distintas partes que forman la aplicación. Esta arquitectura separa la lógica de negocio (el modelo) y la presentación (la vista) por lo que se consigue un mantenimiento más sencillo de las aplicaciones. El controlador se encarga de aislar al modelo y a la vista de los detalles del protocolo utilizado para las peticiones (HTTP, consola de comandos, email, etc.). El modelo se encarga de la abstracción de la lógica relacionada con los datos, haciendo que la vista y las acciones sean independientes del tipo de gestor de bases de datos utilizado por la aplicación. Symfony realiza su propia implementación del MVC, para ello toma lo mejor de esta arquitectura logrando que el desarrollo de las aplicaciones sea más rápido y sencillo. (7)

Las acciones son el corazón de la aplicación, puesto que contienen toda la lógica de la aplicación. Las acciones utilizan el modelo y definen variables para la vista. Cuando se realiza una petición Web en una aplicación Symfony, la URL define una acción y los parámetros de la petición.

La vista se encarga de producir las páginas que se muestran como resultado de las acciones. La vista en Symfony está compuesta por diversas partes, estando cada una de ellas especialmente preparada para que pueda ser fácilmente modificable por la persona que normalmente trabaja con cada aspecto del diseño de las aplicaciones. (30)

Las bases de datos son relacionales. Symfony está orientado a objetos. Para acceder de forma efectiva a la base de datos desde un contexto orientado a objetos, es necesaria una interfaz que traduzca la lógica



de los objetos a la lógica relacional. Esta interfaz se llama ORM (object relational mapping) o “mapeo de objetos a bases de datos”, y está formada por objetos que permiten acceder a los datos y que contienen en sí mismos el código necesario para hacerlo.

La utilización de objetos en vez de registros y de clases en vez de tablas, tiene otra ventaja: permite añadir métodos de acceso en los objetos que no tienen relación directa con una tabla. (30)

El uso de Symfony en el desarrollo de la aplicación permitió una construcción sencilla y eficaz. A través del mismo fue posible centrarse de una manera más enfocada en las distintas partes de la aplicación. El uso de ORM facilitó el acceso a los datos. Mejoró el desarrollo de las aplicaciones mediante la automatización de muchas de las tareas comunes. La estructura del código fuente posibilitó que la escritura del código ganara en legibilidad para un futuro desarrollo de la aplicación.

## 2.4.2 ExtJs

Haciendo referencia a lo mencionado en el epígrafe 1.4.5, ExtJs es un framework escrito en JavaScript destinado a asistir el desarrollo de de aplicaciones enriquecidas para internet. El framework presenta el siguiente árbol de directorios:

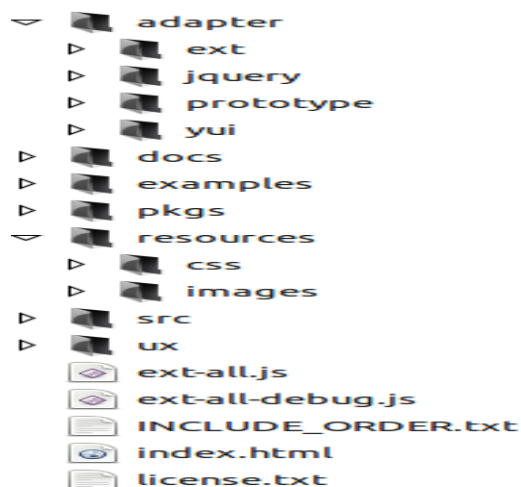


Figura 2 Árbol de directorios de ExtJs

La carpeta src contiene los archivos con el código fuente de cada uno de los componentes y clases de ExtJs agrupados por paquetes. La carpeta resources contiene a su vez las imágenes y las hojas de estilos necesarias de un proyecto. La carpeta pkgs resulta útil cuando no se utilizan completamente todos los recursos de ExtJs, aquí se encuentran comprimido y ofuscados en archivos cada uno de los paquetes de que componen el API. En el directorio de adapters se encuentran archivos para el trabajo con el framework así como para la integración con otros frameworks. En el directorio de docs y examples se encuentra ejemplos y documentación sobre las principales funcionalidades y componentes básicos de ExtJs.

## Arquitectura de ExtJs

ExtJs resalta un fuerte paradigma basado en componentes. Ello posibilita:

1. Múltiple uso: se refiere al hecho de que un componente es escrito dentro de un contexto que permita que su funcionalidad sea útil en la creación de distintas piezas de software.
2. Contexto no específico: en relación con la orientación conceptual de la especificación de un componente, debe estar planteada de una forma general que permita su adaptación en distintos sistemas, sin que el contexto tenga prioridad.
3. Encapsulación: se refiere a la especificación interna oculta o no investigable a través de la interface. Así se protege que el resto de componentes y piezas de software dentro de un sistema, no se vean afectados por cambios en el diseño de uno de los componentes.
4. Una unidad independiente de desarrollo con su propio control de versiones: este principio muy relacionado con la encapsulación, permite que un componente pueda ser desarrollado de manera independiente, cambiando el diseño o agregando nuevas funcionalidades, sin afectar significativamente el resto del sistema.

## Patrones de diseño:

ExtJs implementa varios patrones de diseños. “Los patrones de diseño son el esqueleto de las soluciones a problemas comunes en el desarrollo de software.” En otras palabras, brindan una solución ya probada y documentada a problemas de desarrollo de software que están sujetos a contextos similares. Es necesario tener presente los siguientes elementos de un patrón: su nombre, el problema (cuando aplicar

un patrón), la solución (descripción abstracta del problema) y las consecuencias (costos y beneficios). (31) Su desarrollo fue altamente inspirado en el libro de Head First Design Pattern por Freeman and Freeman. Entre los patrones de diseño que implementa se encuentran:

1. Flyweight: El patrón Flyweight sirve para eliminar o reducir la redundancia cuando se tiene una gran cantidad de objetos que contienen información idéntica, además de lograr un equilibrio entre flexibilidad y rendimiento (uso de recursos). El mismo es implementado a través de Ext.Fly, donde se usa como alternativa a Ext.get, método que obtiene elementos del DOM de una página, pero crea una instancia de cada objeto que llama, sin embargo Ext.fly crea los reemplazos de todos los objetos que se llaman en uno solo.
2. Observador: El patrón observador define una dependencia de uno-a-muchos entre objetos, de forma que cuando un objeto cambie de estado se notifique y actualicen automáticamente todos los objetos que dependen de él. Este patrón se ve reflejado en el manejador de evento de ExtJs, también en la clase base de ExtJs de Ext.util.Observable.
3. Singleton: Garantiza la existencia de una única instancia para una clase y la creación de un mecanismo de acceso global a dicha instancia. Una de las formas de uso se representa en la función init, la cual al llamarse asegura de que exista solo una instancia de un control de ventana en específico, y este pueda ser accedido desde cualquier lugar de la aplicación.

## Herencia

ExtJs provee de herramientas para llevar a cabo la extensión de propiedades y funcionalidades a sus componentes. De esta manera permite redefinir los mismos, cambiando su disposición a preferencia del desarrollador. Esta característica fue muy usada en la construcción del presente trabajo de diploma ya que permitió crear los componentes a partir de clases básicas de ExtJs.

La propiedad fundamental para la herencia se refleja a través de Ext.extend. A través de este método se especifica cómo se implementa la herencia así como las clases fuentes para llevar a cabo la misma.

```

1  Ext.ns("contenedor");
2  contenedor=Ext.extend(Ext.Panel, {
3      editable: "false",
4      father: "none",
5      border: true,
6      style:{
7          position: 'absolute',
8          width: 'auto',
9          height: 'auto'
10     },
11     draggable: true,
12
13     constructor:function(options){
14         Ext.apply(this,options || {});
15         contenedor.superclass.constructor.call(this);
16     },
17
18     getEditable:function(){
19         return this.editable;
20     },
21     setEditable:function(peditable){
22         this.editable = peditable;
23     },
24
25     getFather:function(){
26         return this.father;
27     },
28     setFather:function(pfather){
29         this.father = pfather;
30     }
31 });
32

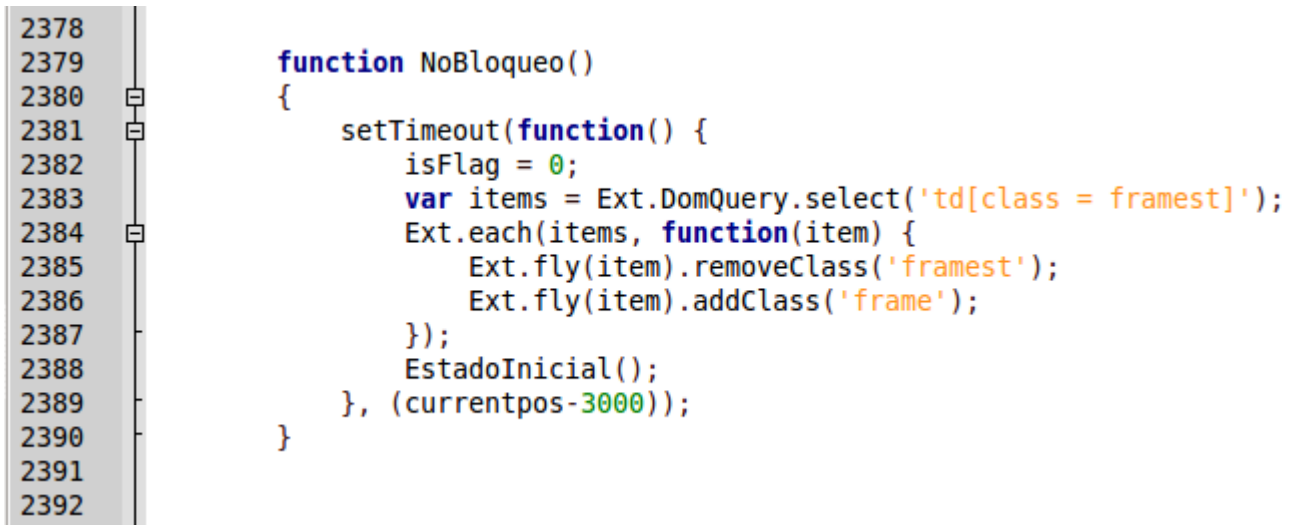
```

Figura 3 Ejemplo de herencia de ExtJs

### Consultas al DOM

ExtJs cuenta con un objeto que permite hacerle consultas al DOM para obtener de él objetos específicos con los cuales se quiere trabajar de forma específica. Ello se hace a través del objeto de DomQuery, el mismo de una manera muy sencilla y rápida, además de que provee varios métodos que regresan una

colección de nodos que satisfacen algún criterio de búsqueda o solamente el elemento que se necesita que sea seleccionado por su identificador, además podemos asignarle el nodo sobre el cual debe realizarse la búsqueda.



```
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392

function NoBloqueo()
{
    setTimeout(function() {
        isFlag = 0;
        var items = Ext.DomQuery.select('td[class = framest]');
        Ext.each(items, function(item) {
            Ext.fly(item).removeClass('framest');
            Ext.fly(item).addClass('frame');
        });
        EstadoInicial();
    }, (currentpos-3000));
}
```

Figura 4 Ejemplo de uso de DomQuery para obtener objetos del DOM

### Espacios de nombre

Los espacios de nombre en ExtJs permiten empaquetar el código de manera que dentro de una misma aplicación se reduzcan la cantidad de conflictos que puedan existir. Un espacio de nombre es un contenedor abstracto de que provee de un contexto a diferentes objetos. Con ello se evitan las ambigüedades y además de que previene que se sobrescriban objetos.

## 2.5 Diagrama de clases del diseño

A diferencias del modelo conceptual, un diagrama de este tipo contiene las definiciones de las entidades del software en vez de conceptos del mundo real. El UML no define concretamente un elemento denominado “diagrama de clases del diseño” sino que se sirve de un término más genérico “diagrama de clases”. (17)

Un diagrama de clases es un diagrama estático que describe la estructura de un sistema mostrando sus clases, atributos y las relaciones entre ellos. Los diagramas de clases son utilizados durante el proceso de análisis y diseño de los sistemas, donde se crea el diseño conceptual de la información que se manejará en el sistema, y los componentes que se encargarán del funcionamiento y la relación entre uno y otro.

Los diagramas de diseño representan una aproximación del proyecto que se quiere lograr, constituyen una base de apoyo para la implementación. A continuación se muestran los distintos diagramas de clases del diseño que se toman como punto de partida para la implementación del módulo de redacción de noticias.

La redacción de la noticia personalizada contiene las acciones que permiten crear una noticia a través de los elementos configurables, creándose según las especificaciones establecidas por el redactor.

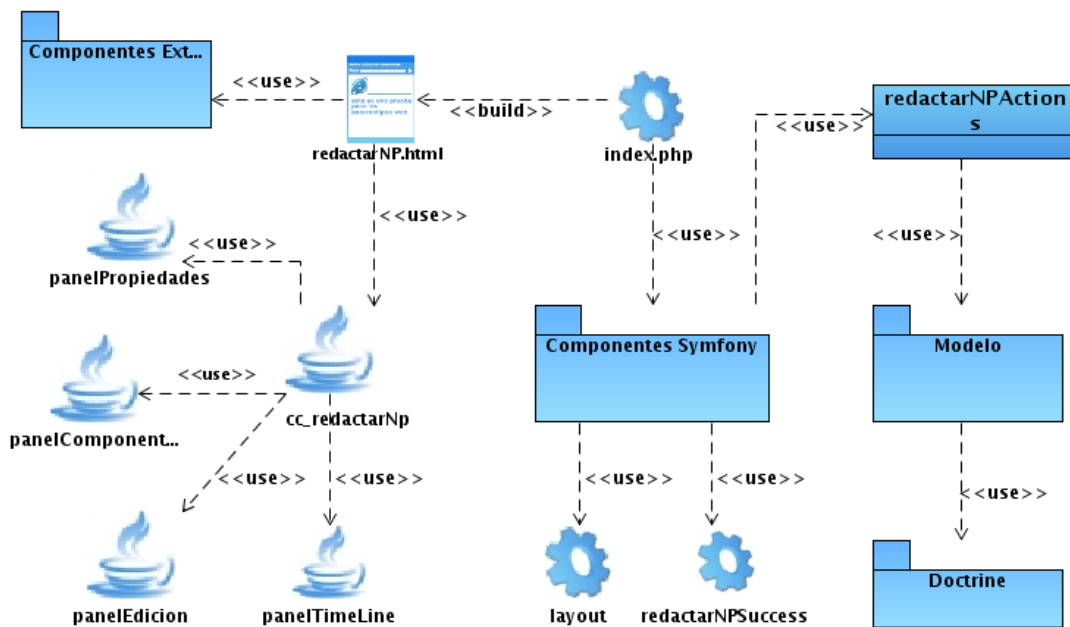


Figura 5 Diagrama de clases del diseño del CUS: Redactar Noticia Personalizada.

La plantilla representa un esquema base sobre el cual se irán agregando elementos para conformar una noticia. La plantilla es un marco predefinido, la cual presenta regiones editables sobre las que se insertan elementos. La plantilla es un canal para crear de manera más rápida una noticia.

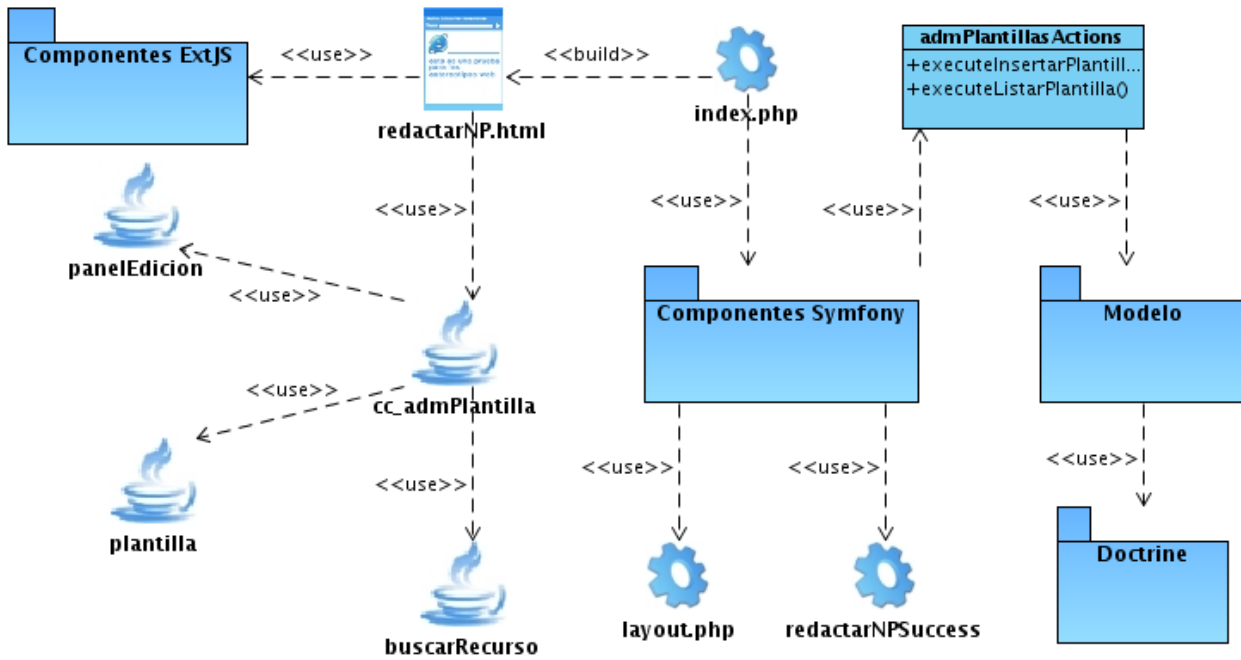


Figura 6 Diagrama de clases del diseño del CUS: Administrar Plantilla.

Los componentes son la base de la noticia. Estos combinados y sincronizados según la manera en que lo ejecute el redactor, conformarán la noticia. Los componentes reciben modificaciones a sus propiedades y es entonces que quedan listos para conformar una noticia.

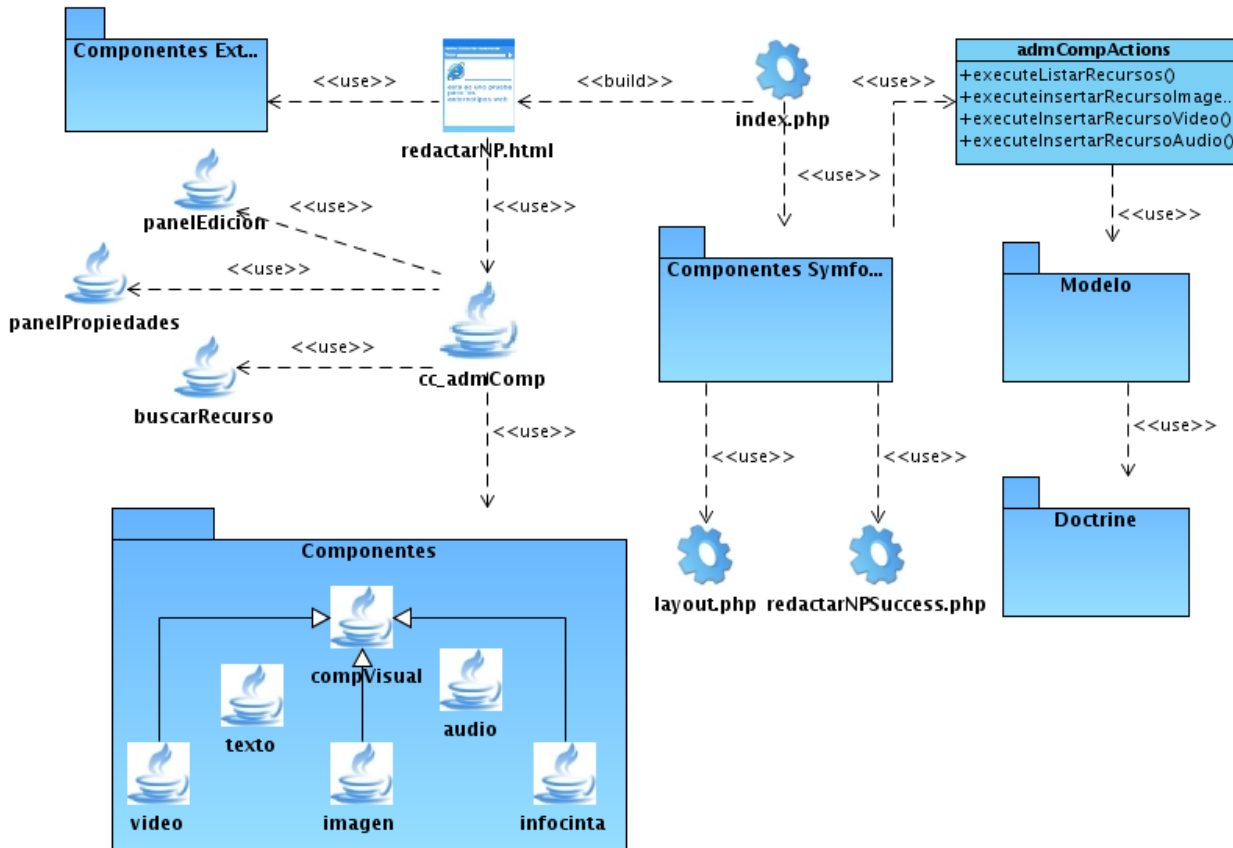


Figura 7 Diagrama de clases del diseño del CUS: Administrar Componente.



Las acciones van a constituir los efectos que se le pueden aplicar a los componentes. Éstas manifiestan la manera en que se van a visualizar los componentes una vez ya integrados en la noticia.

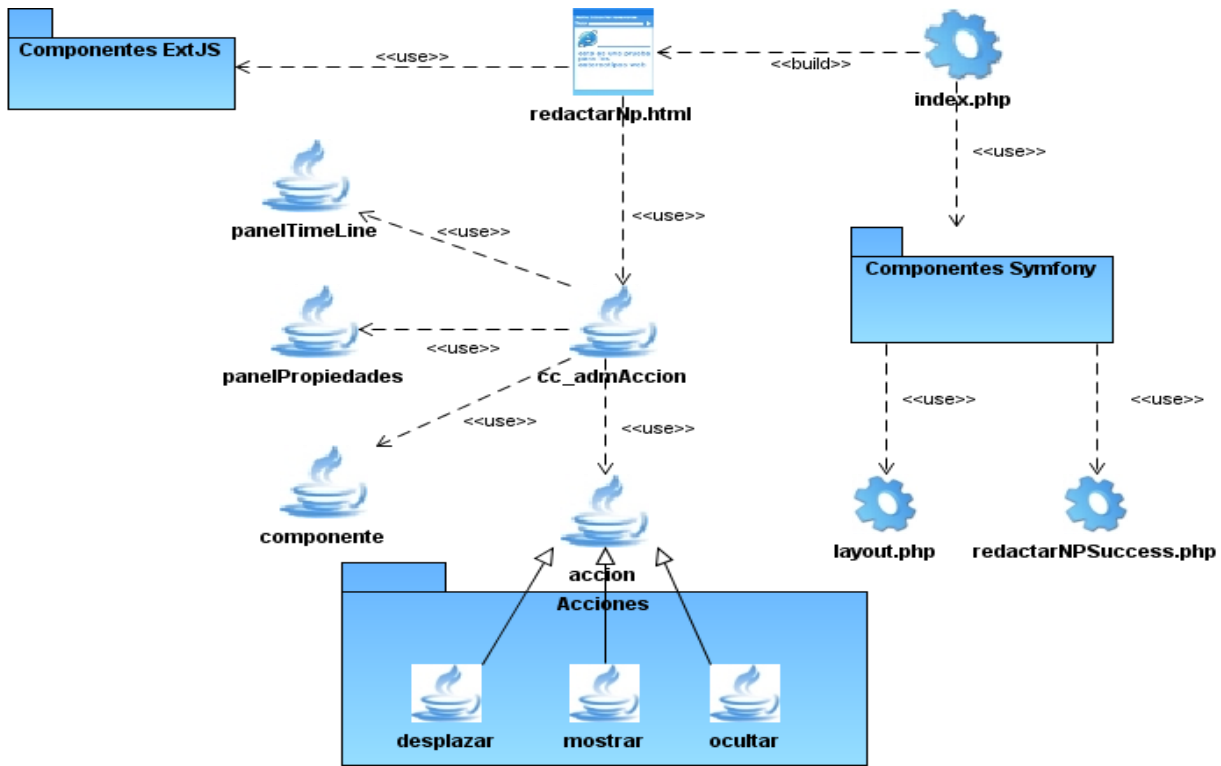


Figura 8 Diagrama de clases del diseño del CUS: Administrar Acción.

## **2.6 Modelo de Implementación**

El modelo de implementación describe cómo se establece la estructura de los componentes, los elementos de implementación. Se basa en las responsabilidades asignadas a subsistemas y módulos. El modelo de implementación describe cómo los elementos del modelo de diseño se implementan en términos de componentes. Describe también la forma en que se organizan los componentes de acuerdo con los mecanismos de estructuración y modularización disponibles en el entorno de implementación y en el lenguaje o lenguajes de programación utilizados además de las dependencias entre los componentes.

Los diagramas de componentes revelan la manera que se divide el software, mostrando las dependencias entre sus partes. Por otro lado el diagrama de despliegue muestra la disposición física en la que se va estacionar el sistema y las relaciones en la misma.

El diagrama de componentes Redactar Noticia Personalizada muestra una vista general de los componentes que presenta el módulo de redacción. Representado en el diagrama se encuentra:

1. El paquete del framework ExtJs, el cual presenta las funcionalidades los componentes y funcionalidades básicas para desarrollar con el mismo. Estas funcionalidades residen básicamente en las fuentes escritas en ext-all.js, ext-base.js y ext-all.css.
2. El paquete ux contiene componentes adicionales, entre los cuales se encuentra Ext.ux.ToastWindow.js, el cual constituye una ventana personalizada para mostrar mensajes. Estos mensajes pueden estar animados desde diferentes posiciones dentro de la ventana. También está el fichero cfg\_windows.js, el cual contiene configuraciones predeterminadas de las ventanas que se muestran en el sistema. Estas ventanas contienen diferentes formularios para la configuración de componentes.
3. El paquete componentes contiene dentro de sí los componentes que se pueden incorporar a una noticia personalizada. Se especifica dentro del mismo el fichero genérico componente.js, que contiene clases básicas para la creación de los componentes. Clases de las cuales heredan la mayoría de los componentes.
4. El paquete del framework de Symfony donde se recogen los ficheros que contienen la interacción del lado del servidor. La misma está desarrollada en PHP, lenguaje con el que trabaja dicho framework de desarrollo.
5. El fichero indexSuccess.php representa la plantilla donde se crea la interfaz de la aplicación.
6. El componente sala\_redacción.js es el encargado de crear toda la interfaz de la aplicación. En el mismo se crean componentes tales como:
  - a. panelTimeLine: representa el panel, o sección donde se mostrará la línea de tiempo de animación de la noticia.
  - b. panelPropiedades: representa el panel que muestra las propiedades del elemento que se selecciona en un momento dado.
  - c. panelEdicion: representa el panel donde se configuran los efectos a aplicar a determinado componente.

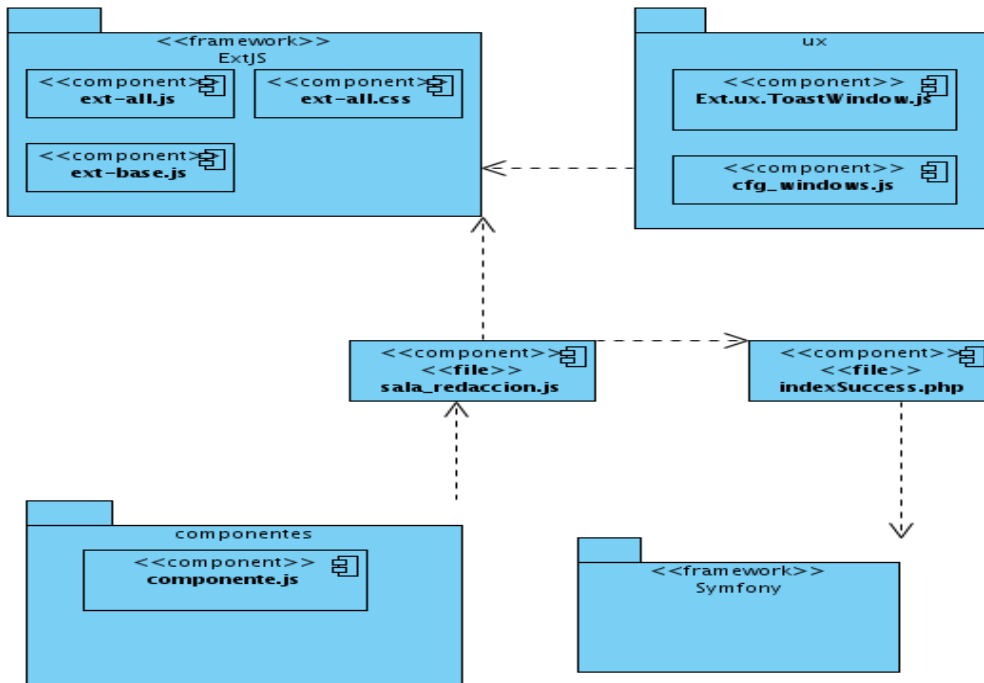


Figura 9 Diagrama de Componentes: Redactar Noticia Personalizada.

El diagrama de componentes Administrar Plantilla especifica las maneras en que se confeccionar una noticia. En esta sección es donde se incluyen las plantillas sobre la cual se creará una nueva noticia. Esta puede ser tanto desde una plantilla vacía como de una ya definida. Este diagrama de componentes que se muestra a continuación aprovecha las funcionalidades del Symfony para la creación y lectura de archivos XML, donde se guardan las configuraciones de las noticias personalizadas. Entre los ficheros que intervienen están:

1. XmlArchivesSuccess.php: Plantilla que responde a la acción executeXmlArchives, acción que se encarga de cargar las plantillas existentes para redactar una noticia personalizada.
2. PlantillaSuccess.php: Plantilla que responde a la acción executePlantilla, donde la misma realiza el envío de la información contenida en formato XML de la noticia. Esta información es procesada luego en la interfaz y construye la base de la plantilla.
3. LeerNoticiaSuccess.php: Plantilla que responde a la acción de executeLeerNoticia, que tiene la

responsabilidad de listar las configuraciones en formato XML de las noticias personalizadas que han sido creadas.

4. GenerarSuccess.php: Plantilla que responde a la acción executeGenerar, la misma posee la capacidad de guardar una noticia personalizada ya redactada en formato XML.
5. LeerSuccess.php: Plantilla que responde a la acción executeLeer, la cual se encarga de mostrar una noticia personalizada, previamente creada, con la posibilidad de ser modificada.

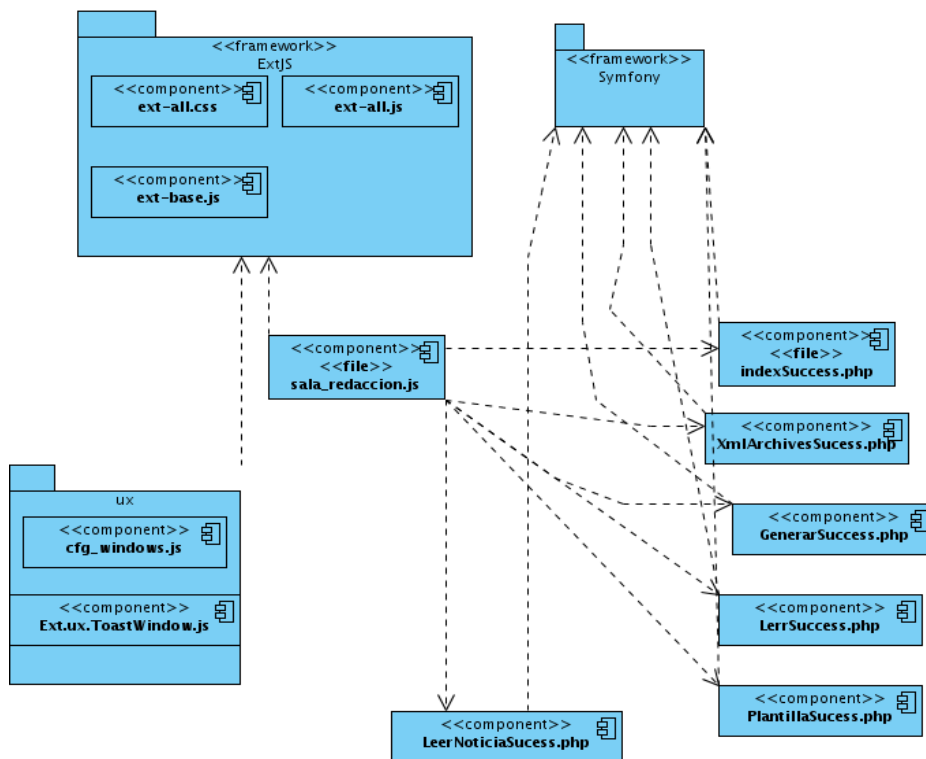


Figura 10 Diagrama de Componentes: Administrar Plantilla.

Mediante el diagrama de componentes Administrar Componente Visual se visualizan los componentes visuales que se construyen en la noticia. A ellos se les aplica configuraciones, cambiando sus atributos

como tamaño, posición, opacidad, además de su transcurso en la noticia. En este diagrama se representan explícitamente los archivos que integran el paquete de componentes que conforman la noticia, además de otros ficheros que contienen funcionalidades de apoyo, los mismos son:

1. `componente_imagen.js`: Representa el componente que crea imágenes en la noticia.
2. `componente_texto.js`: Representa el componente que crea textos en la noticia.
3. `c_texto_imagen.js`: Contiene la combinación del componente imagen y el componente texto.
4. `componente_infocinta.js`: Representa el componente que crea infocintas en la noticia.
5. `componente_video.js`: Representa el componente que crea vídeos en la noticia.
6. `componente_clima.js`: Representa el componente que crea las notificaciones del clima en la noticia.
7. `componente_fecha.js`: Representa el componente que configura la fecha y la hora en la noticia.
8. `propiedades.js`: Representa el componente que configura los elementos de Próxima-Sección, Próxima-Sección, Sección, Tiempo-Restante, Tiempo-Transcurrido y Título de la noticia.
9. `timeline.js`: Representa el componente que crea la línea de tiempo en la noticia.
10. `miDragDrop.js`: Contiene funcionalidades para el control de ubicación de elementos dentro de la noticia.
11. `menuContextual.js`: Contiene configuraciones de los menú contextuales usados en la noticia.
12. `timeline_core.js`: Contiene funcionalidades para el uso del timeline en la noticia.
13. `video_picker.js`: Contiene el componente que realiza la búsqueda de recursos de vídeos.
14. `chooser.js`: Contiene el componente que realiza la búsqueda de recursos de imágenes.
15. `ImagesInSuccess.php`: Plantilla que responde a la acción de `executelmagesIn`, la cual se encarga de listar las imágenes disponibles para crear una noticia personalizada.
16. `VideoInSuccess.php`: Plantilla que responde a la acción de `executeVideoIn`, la cual se encarga de listar los vídeos disponibles para crear una noticia personalizada.

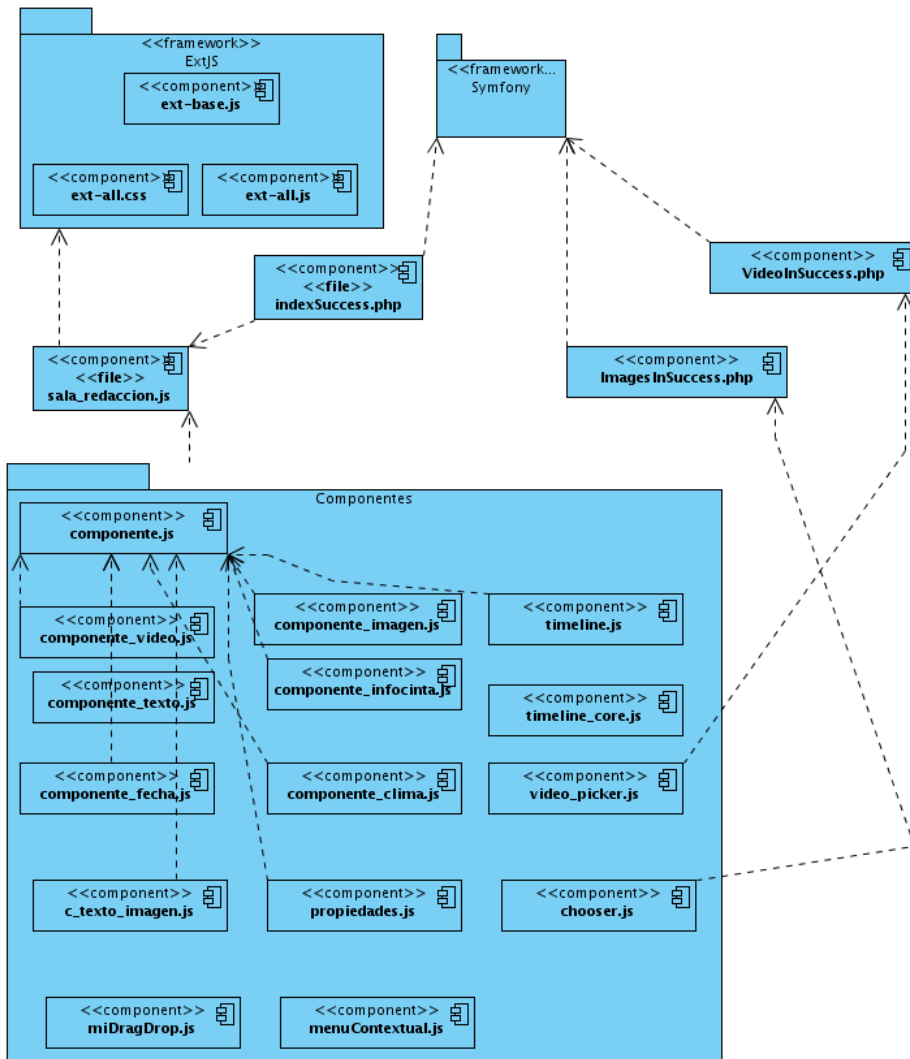


Figura 11 Diagrama de Componentes: Administrar Componente Visual.

El diagrama de componentes Administrar Acción representa las acciones que se le pueden aplicar a los componentes. Este diagrama contiene a los archivos que contienen las funcionalidades para administrar las acciones en una noticia personalizada.

1. El archivo de sala\_redaccion.js, además de las funcionalidades antes expuestas, permite listar y seleccionar los efectos que se le apliquen a los componentes.
2. Mediante menuContextual.js se acceden a configuraciones presentes en los componentes a utilizar en la noticia.
3. A través de timeline\_core.js se utilizan las funcionalidades para navegar entre los efectos configurados en los componentes. Permite configurar el tamaño del timeline entre otras funciones.

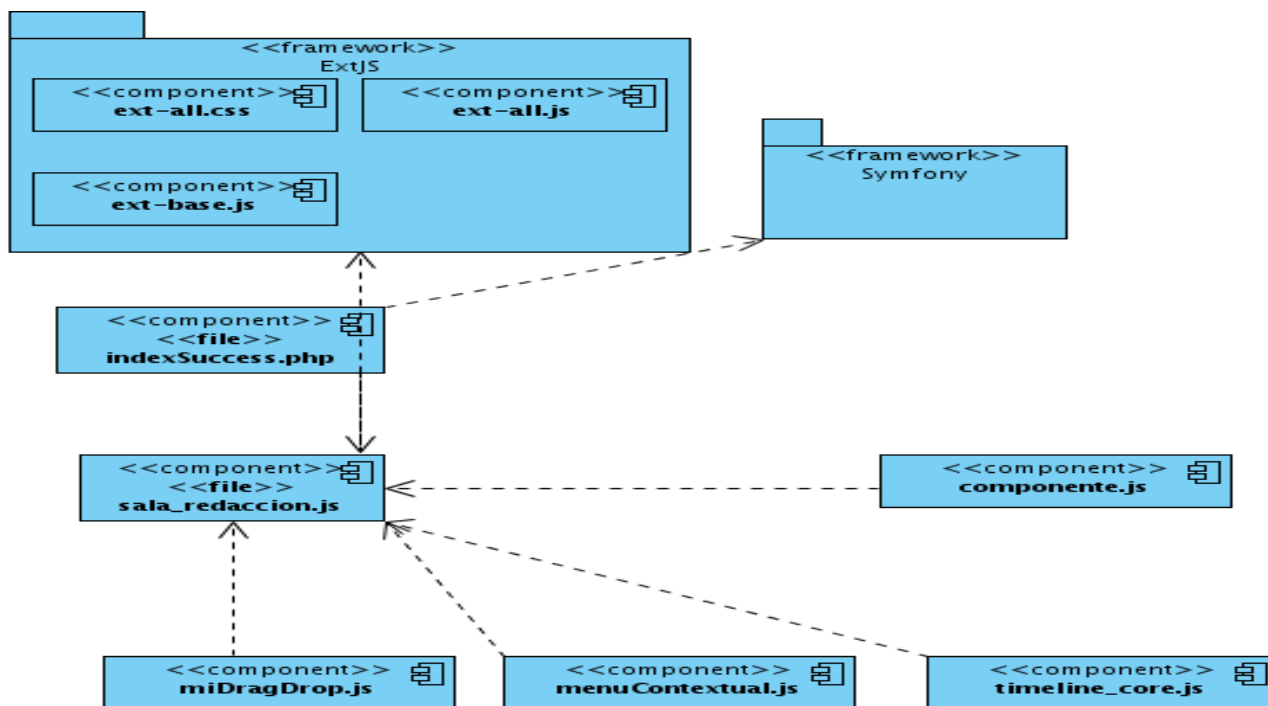


Figura 12 Diagrama de Componentes: Administrar Acción.



El diagrama de despliegue representa la distribución física del sistema. El mismo muestra el nodo donde se instala el servidor de Web, Base de Datos y servidor FTP. En éste es donde va a estar situado el subsistema de Administración, donde se encuentra el módulo de redacción. Al servidor se conecta una PC cliente vía http que tiene como navegador Mozilla Firefox versión 3.0 ó superior.

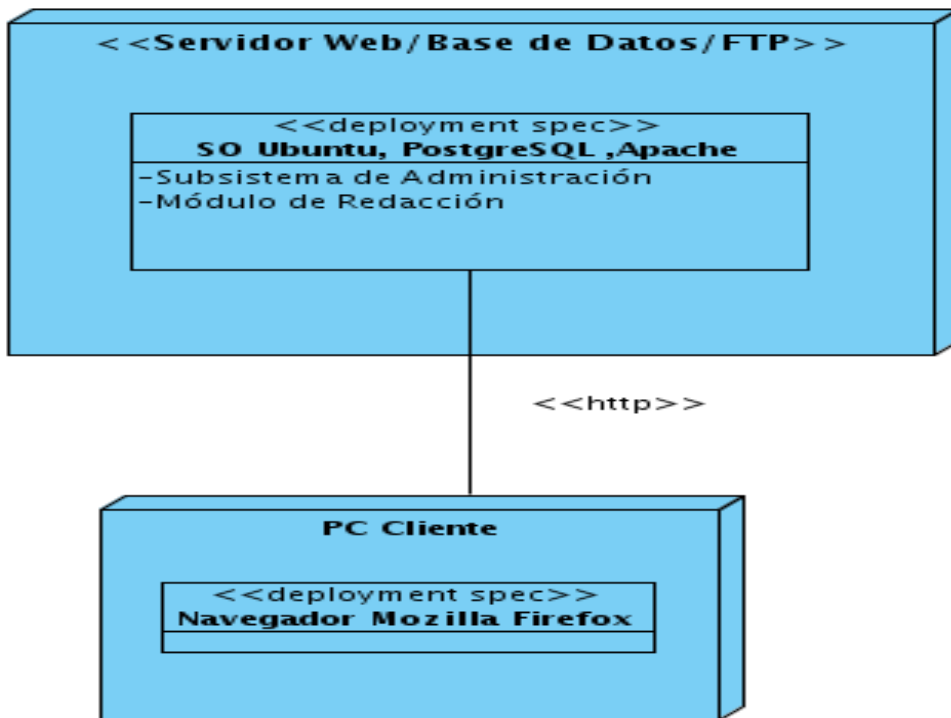


Figura 13 Diagrama de Despliegue

## 2.7 Conclusiones

En este capítulo se realizó una breve descripción del diseño a implementar. En el mismo se enfatizan en los componentes y elementos que intervienen en el proceso de redacción de la noticia. De manera que:

1. Se listan los requisitos funcionales, los cuales representen en ellos las principales funcionalidades a implementar para dar cumplimiento al objetivo general trazado para la investigación.
2. Se identifican los elementos configurables de la noticia a implementar, mediante la configuración de los mismos se crean las noticias personalizadas.
3. Se realiza un análisis de las características principales de los frameworks de desarrollo que se usan en el desarrollo de la aplicación. En este análisis se detallan las potencialidades usadas en la implementación del sistema.
4. Se realiza una breve descripción de los casos de uso del diseño, los cuales muestran una pauta de cómo va a estar constituido de manera genera el módulo con sus funcionalidades.
5. Se realiza una representación de los diagramas de componentes, reflejando en los mismos la interacción entre los componentes que conforman el sistema. Se muestran las dependencias que existen entre los componentes así como una breve explicación de su utilización e interacción dentro del módulo de redacción.
6. A través del diagrama de despliegue se especifica el nodo físico donde estará instalada la aplicación.

## Capítulo 3 Pruebas

### 3.1 Introducción

Las pruebas de software tienen el objetivo principal de detectar posibles errores que puedan ocurrir durante el flujo de trabajo de la aplicación. Las pruebas constituyen un instrumento para determinar el status de la calidad de un producto software. El proceso de prueba puede estar destinado a efectuarse sobre componentes de un sistema o sobre el sistema en sí.

Un proceso de prueba será exitoso cuando encuentre errores, los que no son siempre fruto de la negligencia del programador; por lo que de forma general las pruebas deben centrarse en 2 objetivos:

- Probar si el software no hace lo que debe hacer.
- Probar si el software hace lo que no debe hacer, es decir, si provoca efectos secundarios adversos.

A pesar de que las pruebas no pueden asegurar la ausencia de defectos, ya que sólo se puede demostrar que existen defectos en el software, pues este es hecho por humanos, son parte fundamental antes de entregar el software final. (32)

### 3.2 Diseño de casos de prueba

El método de prueba realizado al sistema son las pruebas de caja negra, las mismas se centran en los requisitos funcionales del software. Mediante las pruebas de caja negra se obtiene un conjunto de condiciones que transitan los requerimientos funcionales del software. Las pruebas de caja negra se centran en lo que se espera de un módulo, es decir, intentan encontrar casos en que el módulo no se atiene a su especificación. Por ello se denominan pruebas funcionales.

### 3.2.1 CUS: Redactar Noticia Personalizada

Nombre la sección	Escenario	Acción Realizada	Respuesta del sistema	Resultado de la prueba
SC1: Configurar Timeline	EC1: El redactor configura correctamente el Timeline.	El redactor selecciona la opción de configurar Timeline y el sistema muestra una ventana con las opciones necesarias para la configuración del Timeline.	El sistema muestra una ventana para configurar el Timeline.  El sistema muestra un mensaje que indica que el timeline ha sido configurado.	Satisfactoria.
	EC2: El redactor introduce datos inválidos.	El redactor introduce valores no válidos	El sistema muestra un mensaje indicando datos erróneos.	Satisfactoria.
SC2:Previsualizar Noticia	EC1: Previsualizar la Noticia.		Se visualiza la noticia con los cambios realizados por el redactor hasta el momento.	Satisfactoria.
	EC2: Guardar Noticia.	El redactor introduce el nombre de la noticia.	El sistema muestra un mensaje de indicando que se guardó la noticia.	Satisfactoria.

Tabla 1 CUS: Redactar Noticia Personalizada

### 3.2.2 CUS: Administrar Plantilla

Nombre la sección	Escenario	Acción Realizada	Respuesta del sistema	Resultado de la prueba
SC1: Seleccionar Plantilla	EC1: El redactor selecciona la opción de seleccionar plantilla.	El redactor selecciona la opción de seleccionar plantilla.	El sistema muestra una ventana para seleccionar la plantilla.	Satisfactoria.
	EC2: El redactor escoge determinada plantilla por su nombre.	El redactor acepta la plantilla seleccionada.	El sistema carga en área de edición la plantilla seleccionada.	Satisfactoria.

Tabla 2 CUS: Administrar Plantilla

### 3.2.3 CUS: Administrar Componente

Nombre la sección	Escenario	Acción Realizada	Respuesta del sistema	Resultado de la prueba
SC1: Insertar Componente	EC1: El redactor selecciona el componente a insertar.	El redactor selecciona un componente y selecciona la opción de insertar.	El sistema muestra en el área de edición el componente seleccionado.	Satisfactoria.

			El sistema crea un canal de efecto al componente seleccionado	
SC2: Modificar propiedades	EC1: El redactor escoge el componente a modificar.	El redactor selecciona un componente previamente insertado.	El sistema actualiza el panel de propiedades y efectos según el componente seleccionado.	Satisfactoria.
	EC2: El redactor modifica las propiedades del componente seleccionado	EL redactor cambia las propiedades a través del panel de propiedades.	El sistema actualiza las propiedades del componente seleccionado.	Satisfactoria.
	EC3: El redactor modifica las propiedades del componente por el menú contextual.	El redactor da clic derecho sobre el componente.	El sistema muestra un menú contextual para cambiar las propiedades del componente.	Satisfactoria.
	EC4: El redactor selecciona opción en el menú contextual.	El redactor da clic sobre la opción seleccionada.	El sistema muestra una ventana de configuración en caso de que sea necesario.	Satisfactoria.
SC3: Eliminar Componente	EC1: El redactor elimina un componente.	El redactor selecciona un componente, da	El sistema elimina el componente del área de edición y	Satisfactoria.

		clic derecho y selecciona la opción de borrar.	borra el canal del Timeline destinado.	
--	--	--	--	--

Tabla 3 CUS: Administrar Componente

### 3.2.4 CUS: Administrar Acción

Nombre la sección	Escenario	Acción Realizada	Respuesta del sistema	Resultado de la prueba
SC1: Aplicar Acción	EC1: El redactor selecciona el componente.	El redactor selecciona el componente en el área de trabajo.	El sistema muestra las acciones que se le pueden aplicar.	Satisfactoria.
	EC2: El redactor escoge cambia las propiedades del componente.	El redactor selecciona dos momentos en los frames cambiando las propiedades del componente seleccionado. Además selecciona la duración del cambio.	El sistema actualiza las modificaciones en los diferentes momentos.	Satisfactoria.
	EC3: El redactor escoge aplicar efecto.	El redactor selecciona del panel de efectos los que estén disponibles para ese elemento y lo aplica	El sistema actualiza las modificaciones en los diferentes momentos.	Satisfactoria.

		seleccionando una duración del mismo.		
SC2:Modificar Acción	EC1: El redactor selecciona el momento de la acción a modificar.	El redactor da clic en el frame a modificar.	El sistema muestra la posición y propiedades de todos los componentes en ese frame.	Satisfactoria.
	EC2: El redactor modifica propiedades.	El redactor selecciona un componente y modifica las propiedades en el panel de propiedades.	El sistema actualiza las propiedades del componente.	Satisfactoria.
	EC3: El redactor modifica un efecto.	El redactor selecciona un efecto y modifica el que tiene definido el componente. Selecciona una nueva duración.	El sistema modifica el efecto seleccionado junto con la duración especificada.	Satisfactoria.
SC3: Visualizar Animación	EC1: El redactor reproduce la animación.	El redactor da clic en el botón que inicia la reproducción.	El sistema comienza a reproducir la animación.	Satisfactoria.
	EC2: Detener animación	El redactor da clic en el botón que detiene la reproducción.	El sistema comienza a reproducir la animación.	Satisfactoria.

Tabla 4 CUS: Administrar Acción



Una vez realizadas las pruebas a las funcionalidades del módulo en diferentes iteraciones, la cantidad de errores fueron en descenso. Ello refleja que el resultado de la aplicación de las pruebas fue exitoso. Obteniéndose, para la última iteración un producto más completo cumpliendo con los requerimientos definidos, de manera que se valida la solución propuesta.

### **3.3 Conclusiones**

La ejecución de las pruebas de software no sólo repercute en el área de desarrollo sino que también se dejan notar en el resto de las áreas de negocio y especialmente en la satisfacción del cliente final. El presente capítulo analiza objetivos y alcance las pruebas que les son realizadas a sistemas informáticos. Para ello se diseñan casos de prueba para comprobar el correcto funcionamiento del flujo de trabajo de la aplicación. Después de realizarse varias iteraciones de pruebas arrojaron que las acciones fueron validadas correctamente.

## Conclusiones

A modo de conclusiones el presente trabajo de diploma confirma el cumplimiento del objetivo general de implementar el módulo de redacción de la Plataforma de Televisión Informativa PRIMICIA. Es por ello que con la implementación del módulo de redacción de noticias de la plataforma de televisión informativa PRIMICIA V2.0 se garantiza la configuración de los elementos visuales de la noticia por lo que se concluye de la siguiente manera:

- Se analizaron soluciones similares, estudiando ventajas que pudieran ser adaptadas para la mejora del proceso de redacción de noticias, determinando que no poseían características suficientes para lograr estos requerimientos referentes a la configuración de la noticia.
- Mediante el uso de la metodología de desarrollo de software RUP, se obtuvo la documentación necesaria para llevar a cabo la implementación de las funcionalidades. Estos artefactos generados servirán como base de futuras actualizaciones al módulo.
- Acorde a las políticas de migración en la Universidad de las Ciencias Informáticas, el desarrollo del presente trabajo de diploma se realizó completamente sobre tecnologías libres.
- Con el uso de diagramas, modelos técnicos y descripciones se muestra una abstracción de lo que funcionalmente representará el módulo de redacción. Todo ello sustentado en la base del Análisis y Diseño previamente definido.
- El módulo implementado garantiza la creación de elementos configurables para la configuración de la noticia
- La aplicación permite la sincronización de las animaciones en la línea de tiempo permitiendo diversidad de visualización de la noticia.
- Se logró la selección de las plantillas para la creación de noticias a partir de la misma.
- Se logró la generación de noticias para su posterior visualización o modificación.
- Con la ejecución de las pruebas y la validación de la solución se comprueba la calidad del

software. La aplicación de pruebas de caja negra, sobre las funcionalidades de la solución propuesta obtuvo resultados positivos.

- Con la implementación del módulo de redacción de noticias se garantiza la configuración de los elementos visuales de la noticia.

Es por ello que se evidencia el cumplimiento de los objetivos propuestos en el presente trabajo de diploma lo cual conlleva al cumplimiento del objetivo general.

## Recomendaciones

Para mantener continuidad al presente trabajo se recomienda:

- Ampliar la creación de componentes, entre ellos el componente tabla y el componente máscara.
- Continuar aplicando procesos de prueba más profundos que permitan refinar las funcionalidades existentes.
- Permitir divulgación del presente documento, permitiendo que sea usado como guía para futuras actualizaciones de la plataforma.
- Ampliar las funcionalidades de manera que permite dimensionar su objeto social ante nuevos requerimientos que surjan.

## Bibliografía

1. **García, Alberto García.** *La implantación técnica de la televisión digital terrestre en España. Modelos Europeos.* Madrid : s.n., 2005.
2. **TeachTarget.** Definición de IDE. <http://searchsoftwarequality.techtarget.com/>. [En línea] TeachTarget, 2010. <http://searchsoftwarequality.techtarget.com/>.
3. **Pressman, Roger S.** Definición de CASE. [En línea] <http://www.rspa.com/spi/CASE.html>.
4. **Española, Real Academia.** Real Academia Española. [En línea] Real Academia Española. [http://buscon.rae.es/draeI/SrvltConsulta?TIPO\\_BUS=3&LEMA=modulo..](http://buscon.rae.es/draeI/SrvltConsulta?TIPO_BUS=3&LEMA=modulo..)
5. Definición de módulo. *Definicion.de.* [En línea] 2008. <http://definicion.de/modulo..>
6. **ALEGSA.** Diccionario Informático. [En línea] ALEGSA. , 2009. <http://www.alegsa.com.ar/Dic/modulo.php..>
7. **González, Carlos Andrés de Jesús.** *Diseño del Subsistema de Configuración de la Plataforma de Televisión Informativa Primicia.* Ciudad de La Habana : s.n., 2010.
8. **Española, Real Academia.** Real Academia Española Definición Configuración. *Real Academia Española Definición Configuración.* [En línea] [http://buscon.rae.es/draeI/SrvltConsulta?TIPO\\_BUS=3&LEMA=configuraci%C3%B3n..](http://buscon.rae.es/draeI/SrvltConsulta?TIPO_BUS=3&LEMA=configuraci%C3%B3n..)
9. Real Academia Española Definición de implementación. *Definición de implementación.* [En línea] [http://buscon.rae.es/draeI/SrvltConsulta?TIPO\\_BUS=3&LEMA=implementaci%C3%B3n..](http://buscon.rae.es/draeI/SrvltConsulta?TIPO_BUS=3&LEMA=implementaci%C3%B3n..)
10. **Daniel Barrios Cardoso, Marisleidy Torres Camejo.** *Desarrollo del prototipo funcional encargado de generar el formato XML de la Plataforma de Televisión Informativa PRIMICIA.* Ciudad Habana : s.n., 2009.
11. **Avilés, José Alberto García.** *Influencia de la Digitalización en los canales "Todo Noticias" de las televisiones públicas.* s.l. : Universidad Miguel Hernández.
12. **Cabrera, Lisandra Delgado.** *Diseño de las nuevas funcionalidades del Módulo de redacción de la Plataforma de televisión Informativa Primicia.* . Ciudad de La Habana : s.n., 2010.
13. **Sony.** Sonaps. *Sonaps.* [En línea] 2010. <http://www.sony.fi/biz/lang/en/fi/product/network-production/sonaps/overview>.
14. —. SONAPS Sistema de Producción en Red. *Sistema de Producción en Red.* [En línea] 2007. <http://www.scribd.com/doc/8328747/Sonaps-Sistema-de-Produccion-en-Red>.

15. **Enterprises, MECK.** eNewsRoom. *eNewsRoom*. [En línea] 2010. <http://www.ereleases.com/pr/free-online-eneewsroom>.
16. **Corp, IBM.** *Classic RUP for SOMA*. 2006.
17. **Miranda, Rafael Lorente.** *Propuesta de mejoras para el proceso de redacción de noticia de la Plataforma de Televisión Informativa PRIMICIA*. . Ciudad de la Habana : s.n., 2009.
18. **Ivar Jacobson, Grady Booch, James Rumbaugh.** *Proceso Unificado de Desarrollo de Software*. s.l. : Addison Wesley, 1999.
19. **Valencia, Universidad Politécnica de.** *Introducción a RUP*.
20. **Ivar Jacobson, Grady Booch, James Rumbaugh.** *El Lenguaje Unificado de Modelado. Manual de Referencia*. Madrid : s.n., 2000.
21. **Motta, Eduardo De la.** ¿Porqué usar estándares de codificación? ¿Porqué usar estándares de codificación? [En línea] 2008. <http://eduardodelamotta.com/2008/por-que-usar-estandares-en-la-codificacion.html>.
22. **Complete, Computers & Applied Sciences.** *JavaScript. Computer Desktop Encyclopedia*. 2010 .
23. **Lobo, Armando Robert.** *Tutorial Básico de ExtJS*. Ciudad de la Habana : s.n., 2006.
24. **Inc., Hostito.** Hostito Inc. *Glosario de Términos*. [En línea] 2010. [http://www.hostito.com/es/faq/index.php?section=glossary&title\\_section=Glosario%20de%20T%E9rminos#30](http://www.hostito.com/es/faq/index.php?section=glossary&title_section=Glosario%20de%20T%E9rminos#30).
25. **Eguiluz, Javier.** El framework Symfony, una introducción práctica. *El framework Symfony, una introducción práctica*. [En línea] 2010. <http://www.maestrosdelWeb.com/editorial/el-framework-symfony-una-introduccion-practica-i-parte/>.
26. **Mata, Manuel Pérez.** ¿Qué es Doctrine ORM? ¿Qué es Doctrine ORM? [En línea] 2009. <http://www.tecnoretas.com/programacion/que-es-doctrine-orm/>.
27. **Vera, Pablo.** Aptana Studio: excelente entorno gratuito para desarrollo Web. *Aptana Studio: excelente entorno gratuito para desarrollo Web*. [En línea] 2009. <http://www.adictosalared.com/aptana-studio-excelente-entorno-gratuito-para-desarrollo-Web/> .
28. **Hewitt, Joe.** Firebug, Web debugging envolved. *Firebug, Web debugging envolved*. [En línea] 2010. <http://joehewitt.com/software/firebug/>.
29. **Visual-paradigm.** Visual-paradigm. *Visual-paradigm*. [En línea] 2010. <http://www.visual-paradigm.com/>.

30. **Potencier, Fabien y Zaninotto, François.** *Symfony la guía definitiva*. 2008.
31. **Tedeschi, Nicolás.** ¿Qué es un patrón de diseño? *MSDN*. [En línea] 2011. <http://msdn.microsoft.com/es-es/library/bb972240.aspx>.
32. **Garnet, Joan.** Doctrine Open Source para PHP 5.2. *Doctrine Open Source para PHP 5.2*. [En línea] 2009. <http://joan.garnet.io/?p=415>.
33. **Calzado, Frank Benítez.** *Procedimiento para el seguimiento y tratamiento de los errores de PRIMICIA*. . Ciudad de la Habana : s.n., 2009.
34. Real Academia Española. *Definición de configuración*. [En línea] Real Academia Española. [http://buscon.rae.es/draeI/SrvltConsulta?TIPO\\_BUS=3&LEMA=configuraci%C3%B3n..](http://buscon.rae.es/draeI/SrvltConsulta?TIPO_BUS=3&LEMA=configuraci%C3%B3n..)
35. **The jQuery Project.** JQuery. [En línea] 2010. [http://docs.jquery.com/Main\\_Page](http://docs.jquery.com/Main_Page).
36. **NetBeans Team.** NetBeans. [En línea] Oracle Corporation, 2011. [http://netbeans.org/community/releases/69/index\\_es.html](http://netbeans.org/community/releases/69/index_es.html).
37. **Prototype Core Team.** Prototype. [En línea] 2006-2007. <http://prototypejs.org/>.
38. **IDG Communications.** *Técnicas y modelos de personalización de sitios Web*. 2001. iWorld, La revista de la tecnología y estrategia de negocio en Internet.
39. **Visual Paradigm.** Visual Paradigm. [En línea] 1999-2011. <http://www.visual-paradigm.com/>.
40. **Zend Technologies Ltd.** Zend Framework. [En línea] 2006-2011. [Citado el: 1 de Marzo de 2011.] <http://framework.zend.com/manual/en/coding-standard.html>.
41. **Alonso García, Yanoisis.** *Diseño del Módulo de Personalización de la interfaz gráfica de la Plataforma de Televisión Informativa PRIMICIA*. Ciudad de la Habana : UCI, 2010.
42. **Bermúdez Valdés, Geovanys.** *Propuesta de Sistema Operativo con kernel Linux para la Plataforma de Televisión Informativa, PRIMICIA*. Ciudad de la Habana : UCI, 2010.
43. **Booch, Rumbaugh y Jacobson.** *El Lenguaje Unificado de Modelado*. s.l. : Addison Wesley, 2000.
44. **Cachero Castro, Cristina.** *Una extensión a los métodos OO para el modelado y generación automática de interfaces hipermediales*. 2001.
45. **Chamba, Julio.** Herramientas CASE. [En línea] 2010. [Citado el: 22 de Octubre de 2010.] <http://www.scribd.com/doc/40173039/Herramientas-Case>.

46. **Frederick, Shea, Ramsay, Colin y Blades, Steve 'Cutter'**. *Learning Ext JS*. Birmingham - Mumbai : Packt Publishing Ltd., 2008.
47. **Garcia, Jesus**. *EXT JS in Action*. 2010.
48. **Hernández García, Ruber y Montaner Hernandez, Yuniór**. *Primicia: Plataforma de Televisión Informativa*. Ciudad de la Habana : UCI, 2008.
49. **Ivar Jacobson, Grady Booch, James Rumbaugh**. *El proceso unificado de desarrollo de software*. s.l. : Addison Wesley, 2000.
50. **Molpeceres, Alberto**. *Procesos de desarrollo: RUP, XP y FDD*. 2002.
51. **Van Der Henst S., Christian**. Maestros del Web. [En línea] Maestros del Web Corp., 2011. <http://www.maestrosdelWeb.com/editorial/comparacion-frameworks-javascript/>.
52. **Agencia de diseño y comunicación gráfica**. Almas de papel. [En línea] 2009. <http://www.almasdepapel.com/conceptos/identidad-e-identidad-visual.html>.
53. **Revista RED**. CiberHábitat. [En línea] Febrero de 2003. <http://www.inegi.gob.mx/inegi/contenidos/espanol/ciberhabitat/museo/cerquita/redes/medios/intro.htm>.
54. **Estructure Media Systems**. Estructure. [En línea] 2001-2011. <http://www.estructuretv.com/enewsroom.htm>.
55. **UMSNH**. Facultad de Ciencias Físicos-Matemáticas. [En línea] 2000. Universidad Michoacana de San Nicolás de Hidalgo, 2010. <http://www.fismat.umich.mx/~crivera/tesis/node11.html>.
56. **Hooping Publicidad S.L.** Hooping.net. [En línea] 1995-2008. <http://www.hooping.net/glossary/aplicaciones-Web-146.aspx>.
57. **Eguíluz Pérez, Javier**. *Introduccion a XHTML*. 2008.
58. —. *Introducción a Javascript*. 2008.
59. —. *Introduccion a CSS*. 2008.
60. **Potencier, Fabien**. *Symfony, La guía definitiva*. 2007.



## Referencia Bibliográfica

1. **García, Alberto García.** *La implantación técnica de la televisión digital terrestre en España. Modelos Europeos.* Madrid : s.n., 2005.
2. **TeachTarget.** Definición de IDE. <http://searchsoftwarequality.techtarget.com/>. [En línea] TeachTarget, 2010. <http://searchsoftwarequality.techtarget.com/>.
3. **Pressman, Roger S.** Definición de CASE. [En línea] <http://www.rspa.com/spi/CASE.html>.
4. **Española, Real Academia.** Real Academia Española. [En línea] Real Academia Española. [http://buscon.rae.es/drael/SrvltConsulta?TIPO\\_BUS=3&LEMA=modulo..](http://buscon.rae.es/drael/SrvltConsulta?TIPO_BUS=3&LEMA=modulo..)
5. Definición de módulo. *Definicion.de.* [En línea] 2008. <http://definicion.de/modulo..>
6. **ALEGSA.** Diccionario Informático. [En línea] ALEGSA. , 2009. <http://www.alegsa.com.ar/Dic/modulo.php..>
7. **González, Carlos Andrés de Jesús.** *Diseño del Subsistema de Configuración de la Plataforma de Televisión Informativa Primicia.* Ciudad de La Habana : s.n., 2010.
8. **Española, Real Academia.** Real Academia Española Definición Configuración. *Real Academia Española Definición Configuración.* [En línea] [http://buscon.rae.es/drael/SrvltConsulta?TIPO\\_BUS=3&LEMA=configuraci%C3%B3n..](http://buscon.rae.es/drael/SrvltConsulta?TIPO_BUS=3&LEMA=configuraci%C3%B3n..)
9. Real Academia Española Definición de implementación. *Definición de implementación.* [En línea] [http://buscon.rae.es/drael/SrvltConsulta?TIPO\\_BUS=3&LEMA=implementaci%C3%B3n..](http://buscon.rae.es/drael/SrvltConsulta?TIPO_BUS=3&LEMA=implementaci%C3%B3n..)
10. **Daniel Barrios Cardoso, Marisleidy Torres Camejo.** *Desarrollo del prototipo funcional encargado de generar el formato XML de la Plataforma de Televisión Informativa PRIMICIA.* Ciudad Habana : s.n., 2009.
11. **Avilés, José Alberto García.** *Influencia de la Digitalización en los canales "Todo Noticias" de las televisiones públicas.* s.l. : Universidad Miguel Hernández.
12. **Cabrera, Lisandra Delgado.** *Diseño de las nuevas funcionalidades del Módulo de redacción de la Plataforma de televisión informativa Primicia.* . Ciudad de La Habana : s.n., 2010.
13. **Sony.** Sonaps. *Sonaps.* [En línea] 2010. <http://www.sony.fi/biz/lang/en/fi/product/network-production/sonaps/overview>.
14. —. SONAPS Sistema de Producción en Red. *Sistema de Producción en Red.* [En línea] 2007. <http://www.scribd.com/doc/8328747/Sonaps-Sistema-de-Produccion-en-Red>.

15. **Enterprises, MECK.** eNewsRoom. *eNewsRoom*. [En línea] 2010. <http://www.ereleases.com/pr/free-online-eneewsroom>.
16. **Corp, IBM.** *Classic RUP for SOMA*. 2006.
17. **Miranda, Rafael Lorente.** *Propuesta de mejoras para el proceso de redacción de noticia de la Plataforma de Televisión Informativa PRIMICIA*. . Ciudad de la Habana : s.n., 2009.
18. **Ivar Jacobson, Grady Booch, James Rumbaugh.** *Proceso Unificado de Desarrollo de Software*. s.l. : Addison Wesley, 1999.
19. **Valencia, Universidad Politécnica de.** *Introducción a RUP*.
20. **Ivar Jacobson, Grady Booch, James Rumbaugh.** *El Lenguaje Unificado de Modelado. Manual de Referencia*. Madrid : s.n., 2000.
21. **Motta, Eduardo De la.** ¿Porqué usar estándares de codificación? ¿Porqué usar estándares de codificación? [En línea] 2008. <http://eduardodelamotta.com/2008/por-que-usar-estandares-en-la-codificacion.html>.
22. **Complete, Computers & Applied Sciences.** *JavaScript. Computer Desktop Encyclopedia*. 2010 .
23. **Lobo, Armando Robert.** *Tutorial Básico de ExtJS*. Ciudad de la Habana : s.n., 2006.
24. **Inc., Hostito.** Hostito Inc. *Glosario de Términos*. [En línea] 2010. [http://www.hostito.com/es/faq/index.php?section=glossary&title\\_section=Glosario%20de%20T%E9rminos#30](http://www.hostito.com/es/faq/index.php?section=glossary&title_section=Glosario%20de%20T%E9rminos#30).
25. **Eguiluz, Javier.** El framework Symfony, una introducción práctica. *El framework Symfony, una introducción práctica*. [En línea] 2010. <http://www.maestrosdelWeb.com/editorial/el-framework-symfony-una-introduccion-practica-i-parte/>.
26. **Mata, Manuel Pérez.** ¿Qué es Doctrine ORM? ¿Qué es Doctrine ORM? [En línea] 2009. <http://www.tecnoretas.com/programacion/que-es-doctrine-orm/>.
27. **Vera, Pablo.** Aptana Studio: excelente entorno gratuito para desarrollo Web. *Aptana Studio: excelente entorno gratuito para desarrollo Web*. [En línea] 2009. <http://www.adictosalared.com/aptana-studio-excelente-entorno-gratuito-para-desarrollo-Web/> .
28. **Hewitt, Joe.** Firebug, Web debugging envolved. *Firebug, Web debugging envolved*. [En línea] 2010. <http://joehewitt.com/software/firebug/>.
29. **Visual-paradigm.** Visual-paradigm. *Visual-paradigm*. [En línea] 2010. <http://www.visual-paradigm.com/>.

30. **Potencier, Fabien y Zaninotto, François.** *Symfony la guía definitiva*. 2008.
31. **Tedeschi, Nicolás.** ¿Qué es un patrón de diseño? *MSDN*. [En línea] 2011. <http://msdn.microsoft.com/es-es/library/bb972240.aspx>.
32. **Garnet, Joan.** Doctrine Open Source para PHP 5.2. *Doctrine Open Source para PHP 5.2*. [En línea] 2009. <http://joan.garnet.io/?p=415>.
33. **Calzado, Frank Benítez.** *Procedimiento para el seguimiento y tratamiento de los errores de PRIMICIA*. . Ciudad de la Habana : s.n., 2009.
34. Real Academia Española. *Definición de configuración*. [En línea] Real Academia Española. [http://buscon.rae.es/draeI/SrvltConsulta?TIPO\\_BUS=3&LEMA=configuraci%C3%B3n..](http://buscon.rae.es/draeI/SrvltConsulta?TIPO_BUS=3&LEMA=configuraci%C3%B3n..)
35. **The jQuery Project.** JQuery. [En línea] 2010. [http://docs.jquery.com/Main\\_Page](http://docs.jquery.com/Main_Page).
36. **NetBeans Team.** NetBeans. [En línea] Oracle Corporation, 2011. [http://netbeans.org/community/releases/69/index\\_es.html](http://netbeans.org/community/releases/69/index_es.html).
37. **Prototype Core Team.** Prototype. [En línea] 2006-2007. <http://prototypejs.org/>.
38. **IDG Communications.** *Técnicas y modelos de personalización de sitios Web*. 2001. iWorld, La revista de la tecnología y estrategia de negocio en Internet.
39. **Visual Paradigm.** Visual Paradigm. [En línea] 1999-2011. <http://www.visual-paradigm.com/>.
40. **Zend Technologies Ltd.** Zend Framework. [En línea] 2006-2011. [Citado el: 1 de Marzo de 2011.] <http://framework.zend.com/manual/en/coding-standard.html>.
41. **Alonso García, Yanoisis.** *Diseño del Módulo de Personalización de la interfaz gráfica de la Plataforma de Televisión Informativa PRIMICIA*. Ciudad de la Habana : UCI, 2010.
42. **Bermúdez Valdés, Geovanys.** *Propuesta de Sistema Operativo con kernel Linux para la Plataforma de Televisión Informativa, PRIMICIA*. Ciudad de la Habana : UCI, 2010.
43. **Booch, Rumbaugh y Jacobson.** *El Lenguaje Unificado de Modelado*. s.l. : Addison Wesley, 2000.
44. **Cachero Castro, Cristina.** *Una extensión a los métodos OO para el modelado y generación automática de interfaces hipermediales*. 2001.
45. **Chamba, Julio.** Herramientas CASE. [En línea] 2010. [Citado el: 22 de Octubre de 2010.] <http://www.scribd.com/doc/40173039/Herramientas-Case>.

46. **Frederick, Shea, Ramsay, Colin y Blades, Steve 'Cutter'**. *Learning Ext JS*. Birmingham - Mumbai : Packt Publishing Ltd., 2008.
47. **Garcia, Jesus**. *EXT JS in Action*. 2010.
48. **Hernández García, Ruber y Montaner Hernandez, Yuniór**. *Primicia: Plataforma de Televisión Informativa*. Ciudad de la Habana : UCI, 2008.
49. **Ivar Jacobson, Grady Booch, James Rumbaugh**. *El proceso unificado de desarrollo de software*. s.l. : Addison Wesley, 2000.
50. **Molpeceres, Alberto**. *Procesos de desarrollo: RUP, XP y FDD*. 2002.
51. **Van Der Henst S., Christian**. Maestros del Web. [En línea] Maestros del Web Corp., 2011. <http://www.maestrosdelWeb.com/editorial/comparacion-frameworks-javascript/>.
52. **Agencia de diseño y comunicación gráfica**. Almas de papel. [En línea] 2009. <http://www.almasdepapel.com/conceptos/identidad-e-identidad-visual.html>.
53. **Revista RED**. CiberHábitat. [En línea] Febrero de 2003. <http://www.inegi.gob.mx/inegi/contenidos/espanol/ciberhabitat/museo/cerquita/redes/medios/intro.htm>.
54. **Estructure Media Systems**. Estructure. [En línea] 2001-2011. <http://www.estructuretv.com/enewsroom.htm>.
55. **UMSNH**. Facultad de Ciencias Físicos-Matemáticas. [En línea] 2000. Universidad Michoacana de San Nicolás de Hidalgo, 2010. <http://www.fismat.umich.mx/~crivera/tesis/node11.html>.
56. **Hooping Publicidad S.L.** Hooping.net. [En línea] 1995-2008. <http://www.hooping.net/glossary/aplicaciones-Web-146.aspx>.
57. **Eguíluz Pérez, Javier**. *Introduccion a XHTML*. 2008.
58. —. *Introducción a Javascript*. 2008.
59. —. *Introduccion a CSS*. 2008.
60. **Potencier, Fabien**. *Symfony, La guía definitiva*. 2007.

## **Anexos**

### **Glosario de términos.**

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.

TIC: Las Tecnologías de la Información y la Comunicación, también conocidas como TIC, son el conjunto de tecnologías desarrolladas para gestionar información y enviarla de un lugar a otro. Abarcan un abanico de soluciones muy amplio. Incluyen las tecnologías para almacenar información y recuperarla después, enviar y recibir información de un sitio a otro, o procesar información para poder calcular resultados y elaborar informes.

GPL: La Licencia Pública General de GNU o más conocida por su nombre en inglés GNU General Public License o simplemente sus siglas del inglés GNU GPL, es una licencia creada por la Free Software Foundation en 1989 y está orientada principalmente a proteger la libre distribución, modificación y uso de software.

Metalenguaje: Lenguaje utilizado para describir un sistema de lenguaje de programación.

DTD: Una definición de tipo de documento o DTD es una descripción de estructura y sintaxis de un documento XML o SGML. Su función básica es la descripción del formato de datos, para usar un formato común y mantener la consistencia entre todos los documentos que utilicen la misma DTD. De esta forma,

dichos documentos, pueden ser validados, conocen la estructura de los elementos y la descripción de los datos que trae consigo cada documento, y pueden además compartir la misma descripción y forma de validación dentro de un grupo de trabajo que usa el mismo tipo de información.

**Framework:** Conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular, que sirve como referencia para enfrentar y resolver nuevos problemas de índole similar. En el desarrollo de software, un framework es una estructura conceptual y tecnológica de soporte definida, normalmente con artefactos o módulos de software concretos, con base en la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros programas para ayudar a desarrollar y unir los diferentes componentes de un proyecto.

**DOM:** Es esencialmente una interfaz de programación de aplicaciones (API) que proporciona un conjunto estándar de objetos para representar documentos HTML y XML, un modelo estándar sobre cómo pueden combinarse dichos objetos, y una interfaz estándar para acceder a ellos y manipularlos. A través del DOM, los programas pueden acceder y modificar el contenido, estructura y estilo de los documentos HTML y XML, que es para lo que se diseñó principalmente.