

**Universidad de las Ciencias Informáticas**

**FACULTAD 6**



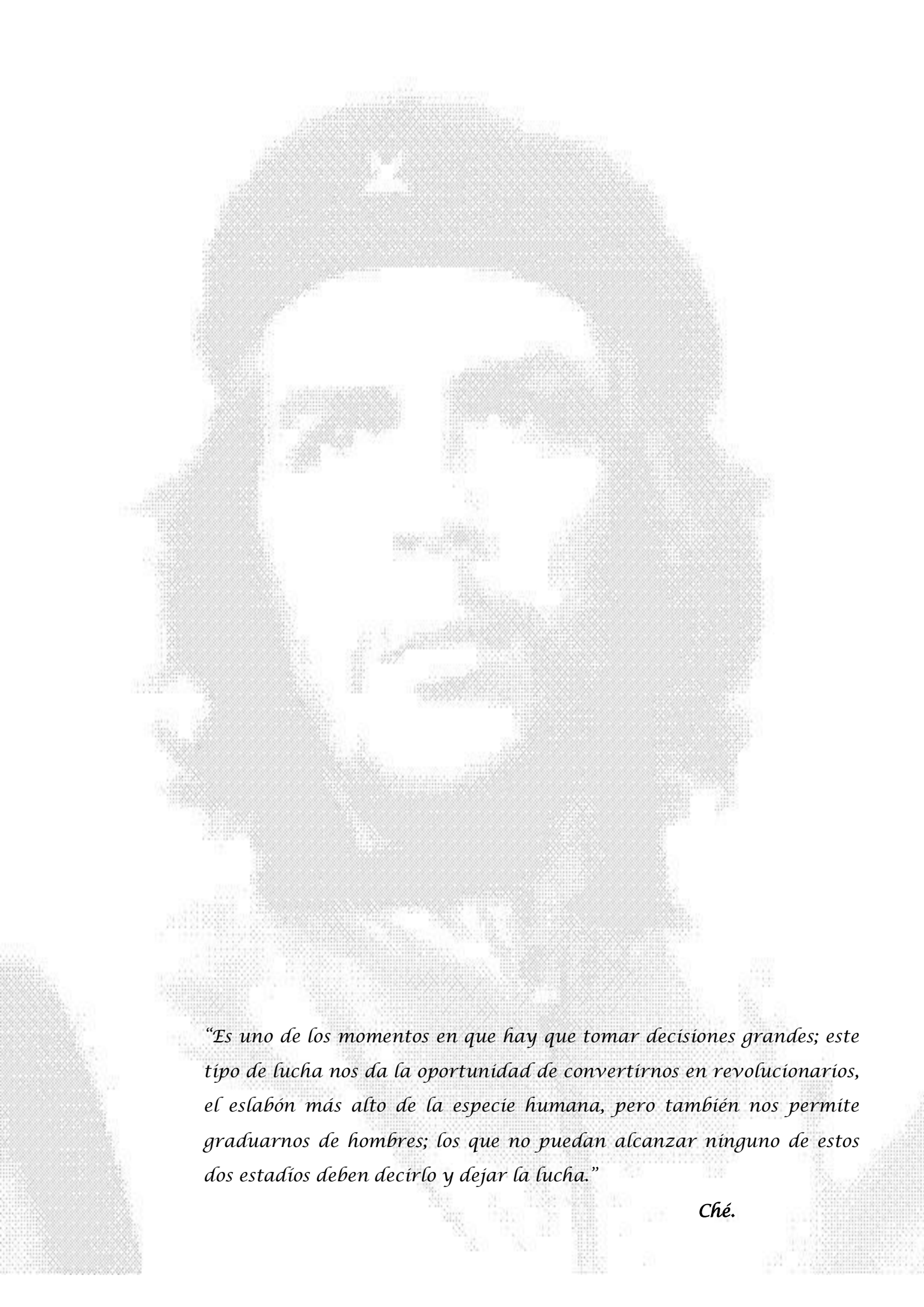
**Título: Desarrollo del módulo de Redacción de Noticias para la Plataforma de Televisión Informativa PRIMICIA v2.0.**

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

**Autor:** Dorgis Montes de Oca Gómez

**Tutor:** Ing. Félix Iván Romero Rodríguez

Junio de 2012



*“Es uno de los momentos en que hay que tomar decisiones grandes; este tipo de lucha nos da la oportunidad de convertirnos en revolucionarios, el eslabón más alto de la especie humana, pero también nos permite graduarnos de hombres; los que no puedan alcanzar ninguno de estos dos estadios deben decirlo y dejar la lucha.”*

*Ché.*

## *Declaración de Autoría*

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma con carácter exclusivo.

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

Dorgis Montes de Oca Gómez

Ing. Félix Iván Romero Rodríguez

---

---

Firma del Autor

Firma del Tutor

**Tutor:** Ing. Félix Iván Romero Rodríguez

**Categoría docente:** Adiestrado

**Centro de trabajo:** Universidad de las Ciencias Informáticas

**Título de la especialidad de graduado:** Ingeniero en ciencias Informáticas

**Año de graduación:** 2011

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

**Correo electrónico:** firomero@uci.cu

**Teléfono particular:** 837 337

## *Dedicatoria*

A mis padres Odalis y Ramón por conducirme siempre por el sendero correcto y más seguro, por ayudarme siempre y por brindarme tanto amor y cariño. Ustedes han hecho realidad este sueño.

A la memoria de mi abuela Reinalda y mi abuelo Tomás a los cuales extraño mucho.

A toda mi familia por apoyarme y exigirme siempre ser cada vez mejor.

A mi novia Cheila por estar siempre a mi lado en los momentos difíciles.

A los amigos de todo el mundo que considero como hermanos.

## *Agradecimientos*

A mi madre Odalis por crear en mí ese espíritu de sacrificio y de lucha, por brindarme ese hombro en el cual apoyarme cuando lo necesitaba y por darme tanto amor incondicional. Estoy muy orgulloso de ti. Gracias mami.

A mi padre Ramón por ser siempre un ejemplo para mí de honestidad, integridad, decoro, rectitud, respeto, decencia y dignidad. Eres mi paradigma de hombre y de padre.

A mi novia y más cercana colaboradora Cheila por ser mi brazo derecho durante este tiempo, por estar siempre ahí cuando la he necesitado y por tanto cariño que me ha entregado.

A mis hermanos Roberto, Dainovy, Yoandri, José Antonio, Wilfredo, Alejandro por estar siempre al lado de uno hombro con hombro.

A Lázaro que es para mí como un padre y que tanto sostén siempre me ha dado incluso cuando ya no estamos tan cerca.

A mi hermano Jesús Da Rocha y todos mis amigos de Venezuela que tanto me ayudaron cuando no tenía a nadie a mi lado.

A mi primo Aliomar que es el hermano que nunca tuve y que siempre me ha querido tanto.

A mis hermanitos Beatriz y Kevin que tanto quiero y que siempre tengo presente en todo momento.

A tía María y tío Gustavo por estar siempre al tanto de mis estudios y por su apoyo incondicional.

A mi abuela Claudina y mis tíos Omar y Madeline por impulsarme a seguir hacia adelante.

A mis segundos padres Maribel y Alberto por su atención.

A Dunia por esa frase de “lo que te haga falta”, por estar siempre pendiente de mí trabajo y por su ayuda infinita.

A Yordany y Eridniel que tanto me ayudaron con sus revisiones.

A todos aquellos que me ayudaron y no mencioné sin querer.

A aquellos que alguna vez me preguntaron ¿y la tesis?

En la actualidad el acelerado avance de las Tecnologías de la investigación y las comunicaciones ha devenido en diversidad de formas de emitir informaciones y a su vez en disímiles vías de trasmisión. La televisión constituye un atractivo medio de comunicación gracias a su inmediatez y a las facilidades que muestra para los televidentes, pues es capaz de emitir variedad de noticias hasta largas distancias. Es por ello que en el mundo se exige la necesidad de crear plataformas de televisión informativas.

En nuestro país PRIMICIA es una de las plataformas de televisión que se basa en la combinación de tecnologías para dar como fruto la creación de noticias e informaciones que posteriormente serán difundidas por canales televisivos. Dentro de ella el módulo de Redacción de Noticias constituye de vital importancia pues es el que interactúa directamente con el redactor y crea la noticia a transmitir, sin embargo el mismo no permite la configuración y animación de todos sus componentes.

El presente trabajo de investigación construye una solución capaz de suprimir las deficiencias en cuanto a configuración de componentes de la versión anterior del módulo de Redacción de Noticias, dando como resultado una noticia personalizada. Se hizo un estudio de las soluciones análogas a las implementadas en PRIMICIA tanto en Cuba como en el resto del mundo demostrando así la necesidad de desarrollar el nuevo módulo de Redacción de Noticias para PRIMICIA v2.0.

### **Palabras claves:**

Implementación, módulo, plataforma de televisión informativa, redacción, componentes, efectos, transiciones

Introducción .....	1
Capítulo I: .....	5
Fundamentación Teórica .....	5
1.1    Introducción .....	5
1.2    Términos asociados al problema .....	5
1.2.1    Televisión digital satelital .....	5
1.2.2    Módulo.....	5
1.2.3    Componentes .....	6
1.2.4    Implementación .....	7
1.2.5    Lenguaje de Marcas Extensible .....	7
1.3    Análisis del objeto de estudio.....	8
1.3.1    Soluciones similares a PRIMICIA.....	8
1.3.2    Plataforma de televisión informativa PRIMICIA .....	12
1.3.3    Proceso de redacción de la noticia en PRIMICIA .....	13
1.3.4    Caracterización de los componentes visuales de la noticia.....	14
1.4    Metodologías, herramientas y tecnologías a utilizar para el desarrollo de la solución .....	14
1.4.1    Metodología de desarrollo RUP .....	15
1.4.2    Lenguaje Unificado de Modelado.....	18
1.4.3    Herramientas CASE Visual Paradigm.....	19
1.4.4    Lenguaje del lado del servidor PHP .....	19
1.4.5    Lenguaje del lado del cliente Javascript.....	20
1.4.6    Framework para PHP Symfony2.....	20
1.4.7    Framework de presentación jQuery .....	21
1.4.8    Entorno Integrado de Desarrollo NetBeans.....	22
1.4.9    Servidor web Apache.....	22



1.4.10	Sistema Gestor de Base de Datos PostgreSQL.....	23
1.5	Conclusiones .....	23
Capítulo II: .....		25
Presentación de la Solución Propuesta.....		25
2.1	Introducción .....	25
2.2	Modelo del dominio.....	25
2.2.1	Eventos principales del entorno .....	25
2.2.2	Diagrama de clases del modelo del dominio.....	26
2.3	Requisitos.....	26
2.3.1	Requisitos funcionales.....	26
2.3.2	Requisitos no funcionales .....	28
2.4	Modelo de casos de uso del sistema .....	30
2.4.1	Definición de los actores del sistema .....	30
2.4.2	Diagrama de casos de uso del sistema .....	30
2.5	Descripción textual de los casos de uso del sistema.....	31
2.6	Conclusiones .....	31
Capítulo III: .....		33
Construcción de la Solución Propuesta.....		33
3.1	Introducción .....	33
3.2	Modelo de análisis .....	33
3.3	Symfony2 y jQuery.....	34
3.3.1	Arquitectura de Symfony2.....	34
3.3.2	Patrones de diseño de Symfony2 y jQuery .....	35
3.4	Modelo de diseño .....	38
3.5	Diseño de la base de datos.....	38
3.5.1	Diagrama de clases persistentes .....	39

3.5.2	Modelo entidad-relación.....	40
3.6	Modelo de despliegue.....	40
3.7	Modelo de Implementación.....	41
3.8	Validación del sistema.....	41
3.8.1	Prueba de aceptación final del cliente.....	42
3.9	Conclusiones.....	45
	Conclusiones.....	46
	Recomendaciones.....	47
	Bibliografía.....	48
	Glosario de Términos.....	55
	Anexos.....	56
	Anexo I. Descripción textual de los casos de uso del sistema.....	56
	Anexo II. Diagramas de clases del diseño.....	67

Tabla 1. Caracterización de los componentes visuales de la noticia (Lorente Miranda, 2009) .....	14
Tabla 2. Especialistas y rol de las personas participantes en las pruebas de aceptación.....	43
Tabla 3. Recopilación y valoración de no conformidades.....	44
Tabla 4. Descripción textual del CU: Administrar noticia .....	59
Tabla 5. Descripción textual del CU: Crear noticia .....	62
Tabla 8. Descripción textual del CU: Administrar pantalla.....	66

## *Índice de Figuras*

Figura 1. Proceso de desarrollo Iterativo de la Metodología RUP .....	16
Figura 2. Esfuerzo en actividades según fase del proyecto.....	17
Figura 3. Diagrama de clases del dominio .....	26
Figura 4. Diagrama de casos de uso del sistema.....	31
Figura 5. Funcionamiento interno de Symfony2 .....	34
Figura 6. Diagrama de clases persistentes .....	39
Figura 7. Diagrama entidad-relación .....	40
Figura 8. Diagrama de despliegue .....	40
Figura 9. Diagrama de componentes .....	41
Figura 10. Flujo de trabajo de las pruebas de aceptación .....	43
Figura 11. Diagrama de clases del diseño del CU: Administrar noticia.....	67
Figura 12. Diagrama de clases del diseño del CU: Crear noticia.....	67
Figura 15. Diagrama de clases del diseño del CU: Administrar pantalla .....	68

## INTRODUCCIÓN

La necesidad del hombre de estar siempre informado ha proporcionado un vertiginoso incremento en el desarrollo de las Tecnologías de la Información y las Comunicaciones (TIC), la información se considera como un elemento indispensable para cualquier proceso de la actividad humana. La televisión es en este momento uno de los medios informativos más importantes producto de su inmediatez, ha alcanzado un profundo auge dentro de la sociedad y su expansión ha sido inevitable. Su índice de inserción la convierte en el medio perfecto para encaminar el mercado de la información, punto importante para fortalecer este nuevo modelo social creado por las posibilidades que ofrece la tecnología digital.

A partir del 2005 diversas entidades nacionales del Ministerio de la Informática y las Telecomunicaciones comenzaron a trabajar para montar y poner en marcha la Plataforma de Televisión Digital Satelital Cubana. Dicha plataforma posibilita capturar las señales de radio y televisión cubanas en todo el territorio nacional, así como otras señales que se encuentran fuera de las fronteras del país.

La existencia en Cuba de la primera plataforma para la transmisión de radio y televisión directa vía satélite permitirá eliminar las llamadas zonas de silencio<sup>1</sup>. Mediante la instalación de receptores satelitales y antenas parabólicas en centros y entidades sociales, se logra la recepción de las señales a la población de esas regiones, donde viven hoy alrededor de 70 mil cubanos. Además se han instalado cientos de estos equipos en los centros médicos cubanos en el exterior, así como en las residencias de los cooperantes, lo cual posibilita la recepción de los canales Cubavisión, Cubavisión Internacional, Tele-Rebelde, Canal Educativo, Canal Educativo 2, Canal Habana, Radio Rebelde, Radio Progreso, Radio Taíno, Radio Habana Cuba y Radio Reloj. (Hernández, y otros, 2007)

La Universidad de las Ciencias Informáticas ha jugado un papel importante en el desarrollo de plataformas informativas. Señal3, la cual fue construida con el objetivo de mantener informados a la comunidad universitaria fue el primer logro de este sistema de fabricación de noticias.

Esta experiencia sirvió como plataforma para realizar proyectos con un mayor alcance, como es el caso de Señal ACN, aplicación elaborada para mantener informados del acontecer nacional y extranjero a los miles de internacionalistas cubanos que participan en duras y variadas misiones en el exterior del país.

---

<sup>1</sup> Zonas de silencio: regiones montañosas de difícil acceso a la Televisión Terrestre y la Prensa Escrita.

Con la nueva tendencia al uso del software libre y para lograr una independencia tecnológica se crea una versión del producto sobre plataformas libres. Cumpliendo esta condición se construye 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.

Fruto de este triunfo surge PRIMICIA, una plataforma de televisión informativa sentenciada a ser plenamente adaptable. Dicha plataforma contiene en su diseño objetivos de importancia de modo que se obtenga una solución personalizable a disímiles grados de requerimientos. Actualmente PRIMICIA va camino a su segunda versión y la misma posee un módulo de Redacción de Noticias en el subsistema de Administración. Este módulo viabiliza crear las noticias que se transmiten a través del subsistema de transmisión que contiene dicha plataforma. El módulo no admite una completa caracterización en correspondencia con las especificidades de cada beneficiario, además que no exhibe elementos que permitan la configuración de las noticias en la aplicación así como la pluralidad de sus formas de visualización. La poca flexibilidad y alta resistencia reflejada por el actual módulo de Redacción de Noticias propicia que surja la necesidad de alterar su proceso de redacción con la meta de perfeccionar el mismo.

Teniendo en cuenta lo previamente explicado se traza como **problema de investigación**: ¿Cómo lograr la configuración de todos los elementos de la noticia en el módulo de Redacción de Noticias de la plataforma de televisión informativa PRIMICIA? Para el desarrollo de la investigación y teniendo como antecedente el problema anterior se precisa como **objeto de estudio** el proceso de redacción de noticias en los sistemas de información, se plantea como **campo de acción** el módulo de Redacción de Noticias de la plataforma de televisión informativa PRIMICIA v2.0. Se define como **objetivo general**: Desarrollar un editor de noticias que viabilice la configuración de todos los componentes visuales de las noticias concebidas en el módulo de Redacción de Noticias. Del mismo se desprenden los siguientes **objetivos específicos**:

1. Diseñar el módulo de Redacción de Noticias de PRIMICIA v2.0.
2. Implementar el módulo de Redacción de Noticias de PRIMICIA v2.0.
3. Validar el módulo desarrollado mediante pruebas de aceptación.

Como guía de la investigación se plantea la siguiente **idea a defender**: Con el desarrollo del módulo de Redacción de Noticias de la plataforma de televisión informativa PRIMICIA v2.0 se garantizará la configuración de todos los elementos visuales de la noticia.

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

1. Definir de los requisitos funcionales y no funcionales del sistema.
2. Caracterizar de las soluciones existentes en el ámbito nacional o internacional.
3. Desarrollar el sistema haciendo uso de las Herramientas y Tecnologías de desarrollo.
4. Validar la solución propuesta.

Se utilizaron diferentes métodos que permitieron obtener información valiosa para el desarrollo de la investigación:

Se utilizó la **modelación** en la creación de abstracciones con la meta de graficar el ambiente del problema. En el diseño del sistema se usa este método en la elaboración de los diagramas consecuentes de los flujos de trabajo y que proveen de auxilio para entender mejor el objeto de estudio.

La utilización del método **histórico-lógico** permitió la obtención de información relativa al nacimiento, desarrollo y progreso de los módulos de redacción de noticias de las plataformas televisivas y sistemas de teletextos. La información conseguida se aprovecha como sustento al estudio de las novedosas funcionalidades que se van a integrar a la redacción de noticias en la plataforma PRIMICIA.

La investigación incluye la actual introducción, tres capítulos, conclusiones, recomendaciones, bibliografía, glosario de términos y anexos que completan la información contenida en la tesis.

El capítulo 1 basa su desarrollo en el estudio de las soluciones similares a la Plataforma de Televisión Informativa PRIMICIA que de forma análoga se encuentran en el campo del problema planteado. Para la comprensión de los lectores se definen términos que serán usados en reiteradas ocasiones. Se realiza un análisis del objeto de estudio de la investigación y por último se seleccionan y describen las herramientas y tecnologías que serán utilizadas para el desarrollo del producto.

El principal objetivo del capítulo 2 es la selección de los Requisitos Funcionales y no Funcionales del sistema, la posterior descripción de los casos de uso que forman parte del mismo y se definen los actores que interactúan con la aplicación.

El capítulo 3 trata sobre la elaboración de la solución propuesta para el problema en cuestión. Se explica la arquitectura Modelo Vista Controlador que implementa Symfony2 y los patrones de diseño utilizados. Por último se valida el producto a través de las pruebas de aceptación realizadas en un entorno real.

Para culminar esta investigación se arriban a un grupo de conclusiones y recomendaciones, así como la bibliografía consultada para la confección de la tesis. Cuenta además con gran cantidad de anexos que complementan el trabajo efectuado.



#### 1.1 Introducción

En este capítulo se puntualizan algunos de los conceptos fundamentales que permitirán un mejor entendimiento del dominio del problema: Televisión Digital Satelital, PRIMICIA, módulo, Elementos Configurables, Implementación y Lenguaje de Marcas Extensible. De igual modo se realiza un análisis del objeto de estudio para definir el módulo de Redacción de Noticias de la plataforma televisiva PRIMICIA y las novedosas funcionalidades que se anhelan agregar al mismo.

Por último se examinarán soluciones análogas a nivel mundial así como las oriundas de nuestra nación. También se determina la metodología de desarrollo de software, el lenguaje de modelado y la herramienta CASE que se va a utilizar durante todo el flujo de trabajo.

#### 1.2 Términos asociados al problema

##### 1.2.1 Televisión digital satelital

La televisión digital vía satélite es el resultado de la aplicación de la tecnología digital a la señal de televisión para luego transmitirla a una amplia zona geográfica por medio de satélites de comunicaciones, en contraste con la televisión terrestre cuyas ondas no salen de la atmósfera, o la televisión por cable, basada en la transmisión a través de redes de fibra óptica y cable coaxial.

La transmisión de televisión digital vía satélite se divide en dos tramos claramente diferenciados:

- El enlace ascendente o *uplink*, mediante el cual el centro emisor envía las señales de televisión al satélite utilizando grandes antenas parabólicas (de 9 a 12 metros de diámetro)
- Y el enlace descendente, o *downlink*, por medio del cual el satélite retransmite la señal de televisión recibida hacia su zona de cobertura sobre la superficie de la tierra, utilizando una banda de frecuencias diferente a la del enlace ascendente, para evitar interferencias. (Televisión Digital, 2011)

##### 1.2.2 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. (Real Academia Española, 2011)

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”. (Definición, 2011)

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. (Alegsa, 2011)

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. (Andrés, 2010)

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.

### **1.2.3 Componentes**

Los componentes son recursos que se pueden incorporar en la redacción de las noticias. Tienen diversas propiedades las cuales se pueden cambiar, configurando la presentación de dichos componentes en la noticia. Actualmente el módulo de Redacción de Noticias de PRIMICIA posee componentes como el texto, la imagen, el audio, el video y la combinación de estos formatos. El

redactor no tiene la posibilidad de configurarlos durante el proceso de redacción, o sea, no puede realizar cambios en sus propiedades ni asignarle acciones que hagan dinámica la noticia.

### **1.2.4 Implementación**

Según RAE el término implementación se define como:

“Acción y efecto de implementar.” (Real Academia Española, 2011)

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.

El autor Félix Iván Romero Rodríguez, el cual ha realizado estudios sobre el tema, en su Tesis de Diploma define implementación como:

La implementación constituye el ejercicio objetivo en la construcción de sistemas. Mediante la misma se llega a concretar los objetivos y Requisitos 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 de Noticias. (Romero, 2010)

### **1.2.5 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. (Barrios, y otros, 2009)

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. (Barrios, y otros, 2009)

Este lenguaje, es comúnmente usado para guardar mediante documentos XML configuraciones que se realizan en los procesos de sistemas. Esta característica es 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.

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

En este epígrafe se abordarán temas interesantes como la identificación de varios software similares, los cuales han sido analizados como resultado de un profundo estudio de materiales bibliográficos. Además se hará una breve descripción de la Plataforma de Televisión Informativa PRIMICIA así como de su proceso de redacción. Finalmente se hará una pequeña caracterización de los componentes visuales de la noticia.

#### **1.3.1 Soluciones similares a PRIMICIA**

Tanto en el planeta como en el país existen varias compañías como televisoras, que hacen uso de los beneficios que proporcionan la tecnología de los sistemas de producción de noticias, ratificando con la experiencia la factibilidad que proporciona. Entre empresas que han irrumpido en la misma se encuentran:

##### **1.3.1.1 Enewsroom**

ENEWSROOM es un sistema de producción de noticias y contenidos audiovisuales, de la plataforma propietaria Estructure Media Systems. (Estructure, 2011) Integra las herramientas necesarias para la explotación de noticias y la introducción de flujos de trabajo flexibles que permiten a los usuarios realizar su trabajo con grandes facilidades.

Sus características fundamentales son: (Estructure, 2011)

- Sistema sencillo, intuitivo, robusto y fiable.

## Capítulo I: Fundamentación Teórica

- Posee servidores multiformato que son capaces de digitalizar simultáneamente hasta tres calidades de vídeo<sup>2</sup>.
- La base del sistema se conecta con otras herramientas externas<sup>3</sup> de la plataforma Estructure que aportan sus funcionalidades a la redacción.
- Utiliza metadatos personalizables donde los contenidos multimedia pueden ser etiquetados con términos descriptivos pertinentes para que los editores puedan localizarlos de forma rápida.

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

### **Módulos de entrada:** (Estructure, 2011)

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

### **Módulos de Organización:** (Estructure, 2011)

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

### **Módulos de Producción:** (Estructure, 2011)

- Editor de noticias: permite realizar ediciones del material audiovisual y del texto de la noticia. Selecciona los materiales desde cualquier bruto<sup>4</sup>, edita en la línea de tiempo<sup>5</sup> el video, audio y gráficos mientras se desarrolla el texto de la noticia, la sinopsis, o el minutado<sup>6</sup>. Permite la incorporación de voz en off<sup>7</sup>, diferentes modos de edición, detección de planos, escucha rápida, entre otras.

---

<sup>2</sup> Calidades soportadas: alta, baja y ultra baja.

<sup>3</sup> Herramientas externas: editores no lineales, tituladores, prompter, sistemas de grafismo.

<sup>4</sup> El término bruto se refiere a los materiales audiovisuales que pueden ser utilizados en la edición de informaciones.

<sup>5</sup> La línea de tiempo permite ordenar una secuencia de eventos de tal forma que se visualice con claridad la relación temporal entre ellos.

<sup>6</sup> Minutado es distribuir el tiempo correspondiente a una determinada actividad, como programas de televisión y actos públicos.

<sup>7</sup> Se refiere a la técnica de producción donde se retransmite una voz no pronunciada visualmente delante de la cámara.

- Documentación: documenta noticias y material audiovisual bruto y esta documentación posteriormente es enviada al módulo de archivo.

### **Módulos de Salida:** (Estructure, 2011)

- Emisión: gestiona la emisión a través de una escaleta dinámica controlando el video-servidor de emisión.
- Archivo: gestiona el archivo del sistema mediante volcados automáticos o manuales a soportes de seguridad (Cintas, DVD), además de codificar indexando<sup>8</sup> de nuevo el material audiovisual con la información contenida en el sistema.

Internet: La codificación a Windows Media File u otros formatos puede realizarse desde los módulos de editnews, compactador o búsqueda, o mediante la selección de un perfil de codificación en el módulo de Internet. (Estructure, 2011)

### **1.3.1.2 Sonaps**

Es un entorno en red que integra todos los aspectos del proceso de producción de informativos, desde la planificación, captación y edición, hasta la emisión, distribución y archivo. 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 que se han acordado y planificado las noticias, Sonaps se encarga desde la búsqueda de contenido en el archivo hasta la gestión de los datos de planificación para los reporteros. (Sony, 2011)

En la sala de redacción, Sonaps se integra 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. (Sonaps, 2007)

Estos sistemas solventan de manera parcial el inconveniente planteado, pero en contraste la redacción en dichas plataformas incluyen funcionalidades que ya la plataforma PRIMICIA posee y que por ende no es forzoso integrarlas con el módulo de Redacción de Noticias. Además la manipulación de estas

---

<sup>8</sup> Registrar ordenadamente datos e informaciones, para elaborar su índice.

acarrearía el pago de licencias y a su vez crear una dependencia de varias herramientas externas de estas plataformas que de igual modo se comercializan sujetas a licencias.

Por lo formulado precedentemente es incuestionable que el desarrollo de las funcionalidades que se exponen en la vigente investigación resulta más viable que la integración de estas plataformas con PRIMICIA.

### **1.3.1.3 Señal ACN**

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

El proceso de redacción en ACN se realiza de la siguiente forma: (Lorente, 2009)

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

Como se puede estimar, en Señal ACN es permisible el acceso, actualización y presentación de recursos multimedia en las noticias, sin embargo el módulo de Redacción de Señal ACN no incorpora los novedosos componentes que se desean añadir a la redacción en PRIMICIA. Otra desventaja con la que consta esta plataforma es que como acontece en PRIMICIA se niega la posibilidad de configurar los componentes existentes.

### **1.3.1.4 TV Energía**

Este sistema fue destinado a computarizar los procesos que se efectúan para visualizar la información en la red de televisión emplazada en el Ministerio del Poder Popular para la Energía y Petróleo de Venezuela. Es la consecuencia devenida de la investigación sobre el estudio de sistemas informáticos sobre tecnologías libre para la transmisión.

Estos sistemas constituyen los antecedentes de PRIMICIA, estas utilidades describen la posibilidad de acceso, actualización y presentación de los recursos de multimedia en el proceso de redacción de la noticia, no obstante necesitan de la elección de configurar la misma, tampoco concentran la capacidad de añadir novedosos elementos a la noticia que surge en el sistema.

### **1.3.2 Plataforma de televisión informativa PRIMICIA**

PRIMICIA, es un producto informático creado por la Universidad de las Ciencias Informáticas (UCI) en el Departamento de Señales Digitales del Centro GEYSED de la Facultad 6. Es una solución integral capaz de proveer un canal de televisión para la transmisión automática y constante de informaciones en distintos formatos. Une varios formatos como son: texto, imagen, video y audio, con el objetivo de mostrar noticias con la información editada de una forma más atractiva. (Delgado, 2010)

Esta solución puede ser ajustable a cualquier cliente que tenga la necesidad o interés de mantener informados de manera rápida y constante a un grupo significativo de personas. Se consideran clientes potenciales de la plataforma a las agencias de noticias, sedes ministeriales o empresariales, terminales de transporte, hoteles, universidades, centros de convenciones, entre otras. (Delgado, 2010)

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 transmitir las noticias y materiales publicados. (Benítez, 2009)



### 1.3.3 Proceso de redacción de la noticia en PRIMICIA

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. (Romero, 2010)

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. (Lorente, 2009)

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 de Noticias el que contiene mayor responsabilidad en el proceso de redacción de la noticia. El módulo de Redacción de Noticias 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. (Romero, 2010)

El módulo de Redacción de Noticias de PRIMICIA no brinda la posibilidad de adaptarse a los variables entornos<sup>9</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. (Romero, 2010)

---

<sup>9</sup> Variables entornos: agencias de noticias, sedes ministeriales o empresariales, terminales de transporte, hoteles, universidades, centros de convenciones.

Sin embargo para la versión 2.0 ya no existirán los tipos de pantalla sino que las estas serán genéricas, dándole la posibilidad al redactor de combinar los componentes según su criterio. Además posibilitará definir previamente la duración exacta para cada una de las pantallas de manera independiente permitiendo enfatizar en las importantes. En la nueva iteración se prioriza la flexibilidad de las propiedades y la estructura de los componentes de la noticia posibilitando un despliegue en entornos con disimiles características. Obteniéndose un software eficiente y adaptable a las necesidades del cliente.

### 1.3.4 Caracterización de los componentes visuales de la noticia

Componente	Características
<b>Texto</b>	Es un componente que hace referencia a un texto determinado. Puede estar referenciando al texto de un componente específico.
<b>Imagen</b>	Es un elemento que hace referencia a un archivo de imagen cuya dirección esta almacenada en la base de datos.
<b>Video</b>	Componente referenciado a un archivo de tipo video que tiene como característica que su dirección se encuentra en la base de datos.

Tabla 1. Caracterización de los componentes visuales de la noticia (Lorente, 2009)

### 1.4 Metodologías, herramientas y tecnologías a utilizar para el desarrollo de la solución

En este epígrafe se realiza un estudio minucioso de los principales conceptos y las tecnologías que se ajustan a la solución propuesta. Se analiza la metodología a aplicar para el análisis y diseño del módulo teniendo en cuenta las prestaciones que pueda aportar. Se consideran los lenguajes de programación y frameworks de desarrollo apropiados para el desarrollo de la aplicación así como el Sistema de Gestión de Bases de Datos (SGBD) empleado.

## 1.4.1 Metodología de desarrollo RUP<sup>10</sup>.

La complejidad, la globalización y la flexibilidad de los mercados, han conllevado a definir nuevas formas de realizar los negocios. Por otra parte los equipos de desarrollo han tenido que adaptarse a las nuevas tendencias del comercio con el fin de sobrevivir y ser competitivas.

RUP es una guía que permitirá definir un conjunto de actividades indispensables para convertir los Requisitos Funcionales (qué debe hacer el sistema) del usuario en un conjunto de elementos necesarios en la construcción de productos. (Gil, 2008)

La metodología utiliza el Lenguaje de Modelado Unificado (UML) para preparar todos los diagramas de un sistema de software y está fundamentado en tres (3) principios básicos: dirigido por casos de uso, centrado en la arquitectura e iterativo, e incremental. (Gil, 2008)

### 1.4.1.1 Dirigido por casos de uso

Un sistema de software debe brindar servicios a sus usuarios, por lo que se debe conocer qué se necesita y desea para el futuro. Un caso de uso es una fracción de la funcionalidad del sistema. Estos representan los requisitos funcionales (¿qué debe hacer el sistema?) y especifican una secuencia de acciones que el sistema puede llevar a cabo. Tomando el modelo de casos de uso como base, los desarrolladores crean una serie de modelos de diseño e implementación que llevan a cabo los casos de uso. (Gil, 2008)

Es importante que los casos de uso representen los conjuntos de procesos que intervienen en la confección de cierta 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. (Letelier, 1998)

### 1.4.1.2 Centrado en la arquitectura

La arquitectura en un sistema de software es descrita por medio de diferentes vistas del sistema en proceso de desarrollo. La arquitectura surge de las necesidades de la organización, de cómo la perciben los usuarios y de cómo se reflejan en los casos de uso. Pero hay otros factores que también influyen tales como la plataforma en la que funcionará el software, los bloques de construcción de que

---

<sup>10</sup> RUP: Rational Unified Process.

se dispondrá, consideraciones de implementación, sistemas heredados y requisitos no funcionales. (Gil, 2008)

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 de software, sistema operativo, gestor de bases de datos, protocolos, consideraciones de desarrollo como sistemas heredados. Muchas de estas restricciones constituyen requisitos no funcionales del sistema. (Letelier, 1998)

### 1.4.1.3 Iterativo e incremental

Se recomienda dividir el trabajo en partes más pequeñas (Principio divide y vencerás) o subsistemas; donde cada subsistema es una iteración que resulta en un incremento. Las iteraciones hacen referencia a pasos en el flujo de trabajo, y los incrementos, al crecimiento del producto. En cada iteración, los desarrolladores identifican y especifican los casos de uso relevantes, crean un diseño utilizando la arquitectura seleccionada como guía, implementan el diseño mediante componentes y verifican que los componentes satisfagan los casos de uso. (Gil, 2008)

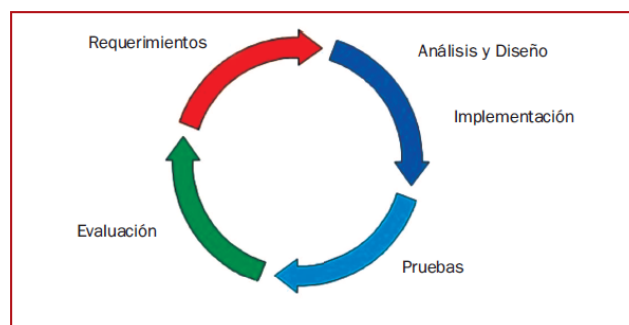


Figura 1. Proceso de desarrollo Iterativo de la Metodología RUP

### 1.4.1.4 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 las distintas actividades. (Letelier, 1998)

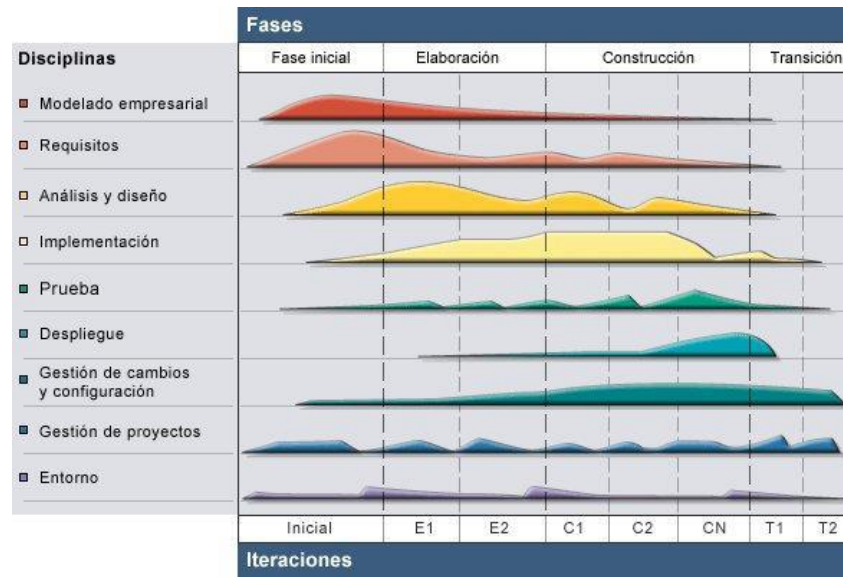


Figura 2. Esfuerzo en actividades según fase del proyecto

#### 1.4.1.4.1 Fase de comienzo o inicio

Está principalmente dirigida al entendimiento de los requerimientos y determinar el alcance del esfuerzo de desarrollo. Se define la idea, la visión y el alcance del proyecto. Esta fase incluye la fase de análisis y diseño que menciona Galvis (2000) en su método de desarrollo de materiales multimedia interactivos. Se incluye un análisis de las necesidades educativas y del entorno educativo, utilizando el modelo de Galvis (2000) y de Lee y Owen (2000), así como el diseño de instrucciones del proyecto. Los requerimientos de diseño gráfico se satisfacen a través del desarrollo del plan creativo de la interfaz (Ward, 1999). En este plan creativo se integra el trabajo de diseñadores gráficos, analistas de sistemas y docentes especialistas en el área. Esta fase se culmina con los objetivos del ciclo de vida. (Díaz-Antón, 2003)

#### 1.4.1.4.2 Fase de elaboración

Planificar las actividades necesarias y los recursos requeridos, especificando las características y el diseño de la arquitectura del software. Esta fase culmina con la arquitectura del ciclo de vida. (Díaz-Antón, 2003)

#### 1.4.1.4.3 Fase de construcción

Desarrollar el producto y evolucionar la visión; la arquitectura y los planes hasta que el producto en una primera versión esté listo para ser enviado a la comunidad de usuarios. Esta fase culmina con la capacidad inicial de operación. (Díaz-Antón, 2003)

#### **1.4.1.4.4 Fase de transición**

Realizar la transición del producto a los usuarios, lo cual incluye: manufactura, envío, entrenamiento, soporte y mantenimiento del producto hasta que el cliente esté satisfecho. Esta fase culmina con la versión de producto, la cual a su vez concluye el ciclo. (Díaz-Antón, 2003)

Para el desarrollo del módulo de Redacción de Noticias de la Plataforma de Televisión Informativa PRIMICIA en su segunda versión es conveniente la implantación de RUP como Metodología de Desarrollo. Esta metodología garantiza la calidad del producto y mayor eficacia en el desarrollo del mismo, es útil para el desarrollo del sistema porque el equipo de desarrollo no cuenta con la máxima experiencia en el desarrollo de software.

#### **1.4.2 Lenguaje Unificado de Modelado**

El proceso de modelado es una base esencial en para la elaboración de cualquier software ya que el mismo ayuda a entender mejor lo que estamos haciendo, comunica en un sistema complejo toda su estructura y descubre oportunidades de reutilización y simplificación en un sistema.

El Lenguaje Unificado de Modelado (UML siglas de Unified Modeling Language) es un lenguaje de modelado visual que se usa para especificar, visualizar, construir y documentar artefactos de un sistema de software. Captura decisiones y conocimiento sobre los sistemas que se deben construir. (Booch, y otros, 1999)

El Proceso Unificado utiliza el Lenguaje Unificado de Modelado para preparar todos los esquemas de un sistema de software. De hecho, UML es una parte esencial del Proceso Unificado, sus desarrollos fueron paralelos. (Booch, y otros, 1999)

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. (Booch, y otros, 1999)

En el desarrollo del módulo Redacción de Noticias de PRIMICIA se utilizará UML porque permite la comprensión por los desarrolladores, pues se encarga de modelar, construir y documentar los elementos que forman este módulo. Se basa en un lenguaje totalmente independiente del lenguaje de programación que se utiliza. Modela los sistemas basados en la programación orientada a objetos, por

tanto los diseños realizados usando UML se pueden implementar en cualquier lenguaje orientado a objetos.

### 1.4.3 Herramientas CASE<sup>11</sup> Visual Paradigm

En la actualidad con el auge de las tecnologías los sistemas de información han ido sufriendo numerosos cambios para lograr cumplir los resultados trazados. Para mantener el camino correcto, satisfacer los objetivos definidos y que el mismo mantenga el equilibrio necesario es importante que cumpla con eficiencia y eficacia, en ocasiones se vuelve un trabajo engorroso. Es por ello la importancia de una correcta selección, adquisición, inserción de las herramientas Case sin embargo esto no constituye una tarea sencilla aún si se toma en cuenta la diversidad de herramientas de este tipo que existen como por ejemplo ARGO-UML, POSEIDON for UML, Rational Rose y 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 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. (Visual-paradigm, 2012)

En la elaboración del módulo de Redacción de Noticias de PRIMICIA se usará para la modelación de todos los artefactos que se generan el Visual Paradigm ya que su uso no se integra con el producto final obtenido y por lo cual no tendría trascendencia legal. El mismo es multiplataforma y permite su uso en cualquier sistema operativo, es una herramienta potente y permite la integración con varios (IDEs).

### 1.4.4 Lenguaje del lado del servidor PHP<sup>12</sup>

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. (Hostito, 2012)

---

<sup>11</sup> CASE: Ingeniería de Software Asistida por Computadora.

<sup>12</sup> PHP: Pre-procesador de Hipertexto.

El uso de PHP en la implementación del módulo en su versión 2.0 tiene como base principal las ventajas de este lenguaje, pues admite la programación orientada a objeto permitiendo programar el software de forma que esté organizado en la misma manera que el problema que se trata de modelar. Existe un amplio conocimiento del mismo, por lo que tiene aceptación por el grupo de desarrollo. Contiene una amplia documentación situada en su página oficial y está respaldado por una activa y numerosa comunidad de desarrolladores.

### **1.4.5 Lenguaje del lado del cliente Javascript**

Javascript es un lenguaje de programación que se utiliza principalmente para crear páginas web dinámicas. Una página web dinámica es aquella que incorpora efectos como texto que aparece y desaparece, animaciones, acciones que se activan al pulsar botones y ventanas con mensajes de aviso al usuario. (Eguíluz, 2009)

Técnicamente, Javascript es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos. En otras palabras, los programas escritos con Javascript se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios. (Eguíluz, 2009)

Gracias a las ventajas de Javascript en cuanto a la representación de opciones gráficas ya que brinda una gran diversidad de elementos visuales es que se hace indispensable su uso para la elaboración del módulo Redacción de Noticias, pues muestra una gama de efectos que son usados en su desarrollo.

### **1.4.6 Framework para PHP Symfony2**

Symfony2 es un framework para PHP que integra una interfaz visual para probar la configuración del servidor, muy útil para solucionar problemas relacionados con el servidor Web o una incorrecta configuración de PHP, permite desarrollar software más rápido y mejor que con PHP simple. Puede rescatar de las tareas cotidianas y te permite recuperar el control de tu código. Symfony2 encamina la petición al código que la maneja tratando de hacer coincidir la URL solicitada contra algunos patrones configurados. También es compatible con XML, PHP y anotaciones nativas. Los diferentes formatos son compatibles y se pueden utilizar indistintamente en una aplicación. Además, el rendimiento de tu aplicación no depende del formato de configuración que elijas, ya que todo se memoriza en caché en la primer petición. (Pacheco, 2011)



En la implementación del módulo Redacción de Noticias de PRIMICIA se decide comenzar la utilización de Symfony2. Symfony 1.4.6 estaba considerado como el framework de PHP favorito ya que posee una clara documentación y gran extensibilidad, pero existen algunas limitaciones en las capas de vistas, en los plug-ins, en la cantidad de archivos que posee, en la configuración, en los formularios y en la rapidez. Symfony2 es un framework completamente nuevo, el cual soluciona estos problemas y en el que todo se traduce a bundle. En Symfony 1.4.6 se habla de frontend y backend como aplicaciones, módulos, plug-ins y librerías, ahora todo esto se debe encapsular en bundles logrando así el desacoplamiento de estos factible. Por último Symfony 1.x es una tecnología más antigua que tiene mantenimiento hasta noviembre de 2012 y no toma ventajas de los espacios de nombres ni de PHP 5.3.

### 1.4.7 Framework de presentación jQuery

jQuery es un framework Javascript que implementa una serie de clases (de programación orientada a objetos) que permiten programar sin preocuparnos del navegador con el que está visitando el usuario, ya que funcionan de exacta forma en todas las plataformas más habituales. (Alvarez, 2010)

Así pues, este framework Javascript, ofrece una infraestructura con la que se tendrá mayor facilidad para la creación de aplicaciones complejas del lado del cliente. Con jQuery se obtendrá ayuda en la creación de interfaces de usuario, efectos dinámicos, aplicaciones que hacen uso de Ajax. (Alvarez, 2010)

Cuando se programa Javascript con jQuery se tendrá a disposición una interfaz para programación que permitirá hacer cosas con el navegador que se estará seguro que funcionará para todos los visitantes. Simplemente se deberán conocer las librerías del framework y programar utilizando las clases, sus propiedades y métodos para la consecución de los objetivos. (Alvarez, 2010)

jQuery es un producto con una aceptación por parte de los programadores muy buena y un grado de penetración en el mercado muy amplio, lo que hace suponer que es una de las mejores opciones. Además, es un producto serio, estable, bien documentado y con un gran equipo de desarrolladores a cargo de la mejora y actualización del framework. (Alvarez, 2010)

En PRIMICIA anteriormente no se utilizaba ningún Framework de presentación. No se decidió escoger ExtJs<sup>13</sup> debido a que no es conveniente para el proyecto PRIMICIA utilizar su licencia GNU GPL V3<sup>14</sup>,

---

<sup>13</sup> Framework para Javascript con el que se pueden crear aplicaciones con estilo de aplicación de escritorio.

<sup>14</sup> Licencia de software que incluye cláusulas que defienden a la comunidad de software libre del uso de las patentes de software

tampoco lo es comprar la licencia comercial que este framework posee. Tampoco es conveniente utilizar Dojo Toolkit <sup>15</sup> ya que este está diseñado para programadores de avanzada experiencia en el tema. jQuery es un framework javascript que ha tenido éxito debido a que está orientado para programadores con poca experiencia. Además posee licenciamiento bajo la Licencia MIT <sup>16</sup>, la cual posibilita utilizarlo y liberarlo bajo BSD <sup>17</sup> la cual es conveniente para PRIMICIA. Por último posee ventajas graficas, es multiplataforma y brinda una gran cantidad de efectos.

### 1.4.8 Entorno Integrado de Desarrollo NetBeans

NetBeans es una herramienta para que los programadores puedan escribir, compilar, depurar y ejecutar programas. Está escrito en Java - pero puede servir para cualquier otro lenguaje de programación. Existe además un número importante de módulos para extender el NetBeans IDE. NetBeans IDE es un producto libre y gratuito sin restricciones de uso. (NetBeans, 2012)

En el desarrollo del módulo utilizamos este IDE ya que además de la ventaja de que es Multiplataforma es capaz de soportar todas las tecnologías con las que se necesitará trabajar como JQuery, Symfony2, Javascript, PHP, CSS, TWIG y HTML. Se podrá ejecutar comandos de Symfony2 desde el Netbeans con solo añadir un script al proyecto y además con la adición de un plugins será reconocido el lenguaje TWIG por este potente Entorno Integrado de Desarrollo.

### 1.4.9 Servidor web Apache

El servidor HTTP Apache es un servidor HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etcétera), Windows y otras, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual. (Guía documentada para Ubuntu, 2012)

Apache es seleccionado como servidor web para el desarrollo del módulo de Redacción de Noticias de PRIMICIA debido a sus potencialidades de servidor web de código abierto que permite modificar el código de fuente según las necesidades del sistema, también porque es multiplataforma y un sistema robusto. Se utilizará un virtual host para poder montar el sitio en una carpeta local de la PC.

---

<sup>15</sup> Framework para Javascript diseñado para programadores con experiencia.

<sup>16</sup> Licencia que permite reutilizar el software así licenciado tanto para hacer software libre como privativo.

<sup>17</sup> Licencia que mantiene la protección de los derechos de autor, pero permite la libre redistribución y modificación, permitiéndose el uso en software comercial.

## 1.4.10 Sistema Gestor de Base de Datos PostgreSQL

Es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD y con su código fuente disponible libremente. Es el sistema de gestión de bases de datos de código abierto más potente del mercado y en sus últimas versiones no tiene nada que envidiarle a otras bases de datos comerciales. Además se ejecuta en la mayoría de los Sistemas Operativos más utilizados en el mundo incluyendo, Linux, varias versiones de UNIX y por supuesto Windows. (PostgreSQL, 2012)

PostgreSQL utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando. (PostgreSQL, 2012)

PostgreSQL constituye un excelente motor de bases de datos, es un sistema fácil de administrar y robusto. Posee la facilidad como las anteriores herramientas del módulo de ser multiplataforma o sea que se podrá usar en cualquier sistema operativo, estas características han permitido que se haya escogido su uso para la realización del módulo de Redacción de Noticias de PRIMICIA.

## 1.5 Conclusiones

El auge de las nuevas tecnologías acentúa la necesidad actual de un perfeccionamiento de las herramientas de trabajo para la elaboración de un software. Las cuales servirán para lograr que el módulo de Redacción de Noticias de PRIMICIA v2.0 cuente con una interfaz que permita que el redactor de noticias pueda elaborar informaciones con mayor calidad y actualizadas. Además, un análisis del objeto de estudio permite arribar al siguiente grupo de conclusiones:

- El estudio de las soluciones anteriores lleva a la necesidad de alterar el proceso de redacción de la noticia con la meta de perfeccionar el mismo, la urgencia de evolucionar el módulo de Redacción de Noticias de PRIMICIA.
- La utilización de RUP como metodología de desarrollo de software garantiza la calidad del producto y una mayor eficacia en el desarrollo del mismo. Además avala la calidad de la aplicación y posibilita el exitoso desarrollo del proyecto.
- El uso de jQuery, posibilita utilizarlo y liberarlo bajo licencia BSD la cual es conveniente para el proyecto PRIMICIA, además posee ventajas graficas, es multiplataforma y brinda una gran cantidad de efectos.

## *Capítulo I: Fundamentación Teórica*

- El uso de Symfony2 como framework de PHP posibilita la construcción de un sistema robusto con una definida arquitectura MVC. Su utilización proporciona trabajar con la actualidad novedosa en cuanto a herramientas de desarrollo de software.

### PRESENTACIÓN DE LA SOLUCIÓN PROPUESTA

#### 2.1 Introducción

En este capítulo se ofrece una descripción de la solución que se propone. Se enuncian los Requisitos Funcionales del sistema, se realizan las descripciones de los casos de usos generados a partir de los requisitos anteriormente mencionados. Para el desarrollo del presente capítulo se mantendrán los elementos definidos en la Fundamentación Teórica anteriormente expuestos.

#### 2.2 Modelo del dominio

Las nuevas funcionalidades del módulo de Redacción de Noticias de PRIMICIA provocan que este proceso sea variable. Cada redactor puede confeccionar sus noticias de la forma que estime conveniente, por esto no existe un proceso de negocio visible y bien definido para estas funcionalidades. De ahí que no sea posible realizar el modelo de negocio y sólo se realizó modelo de dominio. (Delgado, 2010)

Un modelo de dominio captura los tipos más importantes de objetos en el contexto del sistema. Los objetos del dominio representan las "cosas" que existen o los eventos que suceden en el entorno en el que trabaja el sistema. (Rumbaugh, y otros, 2010)

Es un diccionario visual de términos o conceptos importantes en el dominio. Utiliza la notación UML de diagrama de estructura estática. Se enfoca en la semántica y proporciona un glosario de términos basado en sustantivos. (Delgado, 2010)

Todo el conocimiento de las nuevas funcionalidades que se desean desarrollar se presenta en el siguiente diagrama. Este diagrama posibilita a los clientes, usuarios finales del sistema y desarrolladores un estándar para el entendimiento del problema. En él se expresan y vinculan los conceptos primarios del medio donde operará el módulo.

##### 2.2.1 Eventos principales del entorno

El proceso de redacción en la Plataforma de Televisión Informativa PRIMICIA se inicia cuando el **redactor** selecciona la opción Adicionar Noticia. A continuación el sistema mostrará el **área de trabajo** y el redactor le pondrá título a la noticia, elegirá un audio y seleccionará la **sección temática** a la que pertenecerá dicha noticia. Luego el redactor irá agregando **pantallas** y **componentes** a la noticia

pudiendo definir un tiempo para la visualización de la pantalla y una **transición** de entrada y salida. También podrá aplicarle un **efecto** de entrada y de salida a los componentes así como ajustarle sus **propiedades**. Por último cuando el redactor considere que la noticia esté terminada deberá presionar la opción Guardar Noticia.

### 2.2.2 Diagrama de clases del modelo del dominio

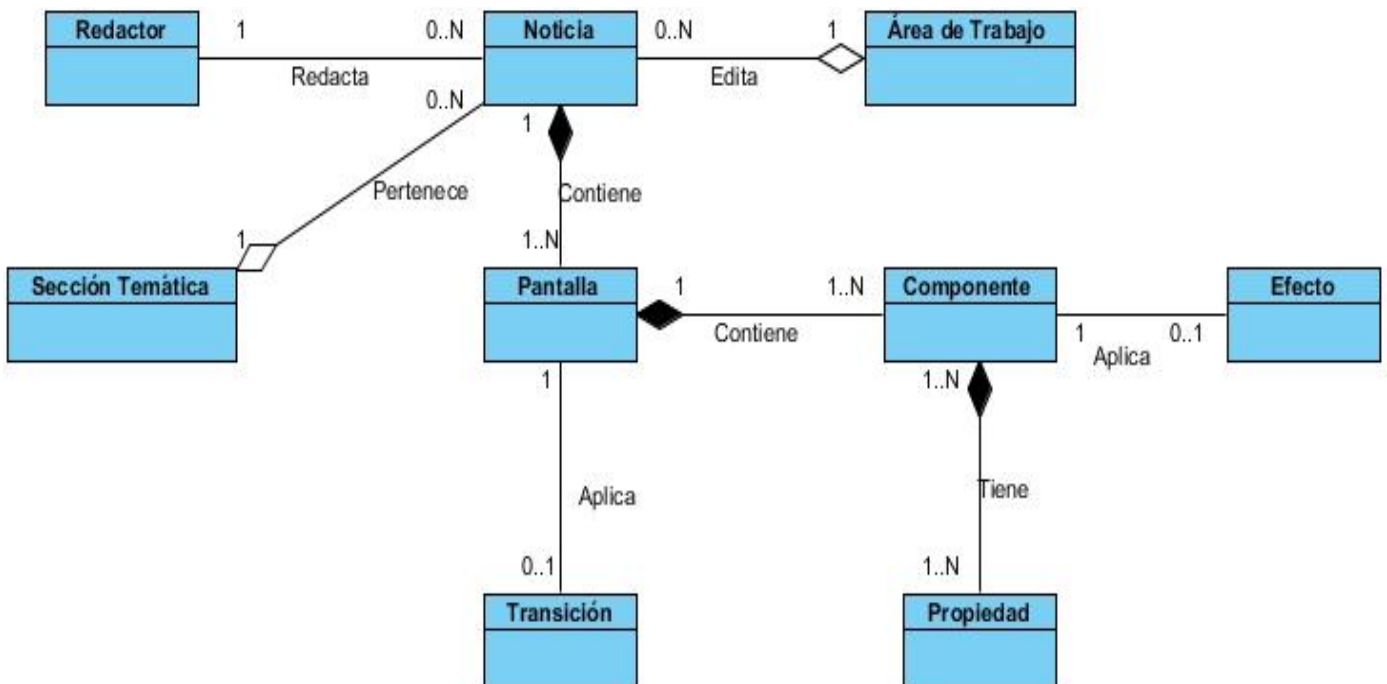


Figura 3. Diagrama de clases del dominio

## 2.3 Requisitos

La ingeniería de requisitos es una de las tareas difíciles que puede enfrentar el ingeniero de software. Ella ayuda a los profesionales a comprender mejor las dificultades para lograr una solución que satisfaga tanto a los clientes como a los usuarios finales.

### 2.3.1 Requisitos funcionales

- RF 1 Listar noticias. El sistema debe brindar un listado que contenga todas las noticias almacenadas en la base de datos.
- RF 2 Crear noticia. El sistema debe permitir la creación de noticias haciendo uso de los componentes definidos.

## *Capítulo II: Presentación de la Solución Propuesta*

- RF 3 Editar Noticia. Debe permitir la edición de las noticias creadas utilizando los elementos configurables.
- RF 4 Mostrar noticia. Debe permitir visualizar las noticias que se localicen en el listado de noticias.
- RF 5 Filtrar noticias. Debe permitir filtrar un listado de noticias de acuerdo a diferentes criterios (título, sección).
- RF 6 Eliminar noticia. El sistema debe admitir la eliminación de una o varias noticias que se encuentren en el listado de noticias.
- RF 7 Validar noticia. El sistema debe permitir validar las noticias listas para publicar una vez que hayan sido corregidas.
- RF 8 Aplicar transición de entrada a pantalla. El sistema debe permitir aplicarle una transición de entrada a las pantallas de las noticias.
- RF 9 Aplicar transición de salida a pantalla. El sistema debe permitir aplicarle una transición de salida a las pantallas de las noticias.
- RF 10 Eliminar transición de entrada a pantalla. Debe permitir eliminar la transición de entrada aplicada a una pantalla.
- RF 11 Eliminar transición de salida a pantalla. Debe permitir eliminar la transición de salida aplicada a una pantalla.
- RF 12 Aplicar efecto de entrada a componente. El sistema debe admitir aplicar un efecto de entrada a los componentes de la noticia.
- RF 13 Aplicar efecto de salida a componente. El sistema debe acceder a aplicar un efecto de salida a un componente de la noticia.
- RF 14 Eliminar efecto de entrada a componente. Debe permitir eliminar el efecto de entrada aplicado a un componente
- RF 15 Eliminar efecto de salida a componente. El sistema de admitir eliminar un efecto de salida aplicado a un componente de la noticia.
- RF 16 Crear pantalla. El sistema debe admitir adicionar una pantalla a la noticia.
- RF 17 Editar pantalla. El sistema debe acceder a editar una pantalla de la noticia.
- RF 18 Ordenar pantallas. El sistema debe permitir Ordenar las pantallas de una noticia de acuerdo a su gusto.
- RF 19 Eliminar pantalla. Debe permitir eliminar una pantalla de la noticia.
- RF 20 Insertar componente. El sistema deberá permitir la inserción de los componentes de la noticia (texto, imagen y video).

## *Capítulo II: Presentación de la Solución Propuesta*

- RF 21 Redimensionar texto o imagen. El sistema debe facilitar redimensionar cualquier componente texto o imagen de la noticia.
- RF 22 Editar texto. Debe acceder a editar un texto de alguna de las pantallas de la noticia del sistema.
- RF 23 Eliminar componente. El sistema de permitir eliminar un componente de la pantalla.
- RF 24 Ajustar propiedades. El sistema debe admitir ajustar las propiedades de un componente de la noticia.

### **2.3.2 Requisitos no funcionales**

Los requisitos no funcionales son propiedades o cualidades que el producto debe tener, además son aspectos importantes que el producto debe cumplir para lograr un producto atractivo, usable, rápido o confiable. (Pressman, 2005)

A continuación se exponen, separados por categorías, los diferentes requisitos no funcionales que el sistema debe tener.

#### **2.3.2.1 Requisitos de Software**

- Para la publicación de la aplicación web se debe utilizar como servidor web Apache v2.2.
- Como Sistema Gestor de Base de Datos en el servidor se debe utilizar PostgreSQL 9.0 superior.
- En el servidor como sistema operativo se debe usar Ubuntu 10.10 o superior.
- Las PC clientes para acceder al sistema deben utilizar como navegador Mozilla Firefox 3.5 o superior.

#### **2.3.2.2 Requisitos mínimos de Hardware**

##### **Cliente**

- Procesador Dual Core 1.60 GHz.
- Disco Duro de 160 GB.
- Memoria RAM de 1 GB.
- Dispositivo de audio.
- Soporte de video que admita resolución de al menos 800 x 600.
- Dispositivo de red con una velocidad de 100 Mbps.



## Capítulo II: Presentación de la Solución Propuesta

### Servidor

- Procesador Core I3 2.80 Ghz.
- Disco Duro de 1TB.
- Memoria RAM de 4GB.
- Dispositivo de red con una velocidad de 100 Mbps.

#### 2.3.2.3 Requisitos de restricciones en el diseño y la implementación

- JQuery 1.7
- Symfony 2.0
- PHP 5.3
- Apache 2.2
- PostgreSql 9.2

#### 2.3.2.4 Requisitos de apariencia e interfaz externa

El sistema deberá contar de manera general, con una interfaz sencilla, que sea intuitiva y de fácil comprensión para el redactor y el corrector de la noticia garantizando la correcta interacción con el sistema. El área de trabajo debe ser de al menos el 60% de la página, el logo siempre debe de estar visible desde cualquier parte del sistema y el menú debe de estar en la parte superior de las páginas.

#### 2.3.2.5 Requisitos legales

Cada herramienta que se utiliza debe ser multiplataforma y de código abierto.

Cada una de las medias (imágenes, videos, sonidos) que se utilicen en el producto deben tener el permiso legal de sus autores y su aprobación para hacer uso de ellas.

#### 2.3.2.6 Seguridad

El software solo debe permitir a cada usuario realizar las acciones para las que tenga permisos es decir será necesario la autenticación del redactor o corrector previamente.

#### 2.3.2.7 Disponibilidad

La disponibilidad del sistema es un aspecto fundamental en la confección de las noticias, para ello es necesario que esté activo las 24 horas del día garantizando que la información fluya de manera que cada usuario pueda acceder al sistema en el momento que lo necesite.

### **2.3.2.8 Usabilidad**

Los enlaces deben estar detallados con el contenido temático, que permita acceder a cualquiera de las páginas o secciones, con un número mínimo de clic. El sistema debe poder ser usado por cualquier persona que posea conocimientos básicos en el manejo de la computadora y de un ambiente Web en sentido general. El sistema no deberá tener ningún enlace defectuoso que impida un rápido acceso a una página o a una funcionalidad del sistema.

## **2.4 Modelo de casos de uso del sistema**

En este epígrafe se definirán los actores del sistema que deberán interactuar con el módulo de Redacción de Noticias de la Plataforma de Televisión Informativa PRIMICIA, además se confeccionará el Diagrama de casos de uso del sistema que mostrará las asociaciones actor/caso de uso.

### **2.4.1 Definición de los actores del sistema**

Cada trabajador del negocio que tiene actividades a automatizar es un candidato a actor del sistema. Si algún actor del negocio va a interactuar con el sistema, entonces también será un actor del sistema. (Pressman, 2005)

Se detectan como actores del sistema al redactor y al corrector de la noticia ya son los que deberán interactuar de manera directa con el módulo Redacción de Noticias.

Redactor: Se encarga de redactar cada una de las noticias que se transmitirán y difundirán.

Corrector: Se relaciona con la noticia una vez creada, se encarga de dar el visto bueno de la noticia. Es el actor que decide cuando la noticia esta lista para publicarla.

### **2.4.2 Diagrama de casos de uso del sistema**

Un diagrama de Casos de Uso del Sistema describe parte del modelo de casos de uso y muestra un conjunto de casos de uso y actores con una asociación entre cada par actor/caso de uso que interactúan (Rumbaugh, y otros, 2010)

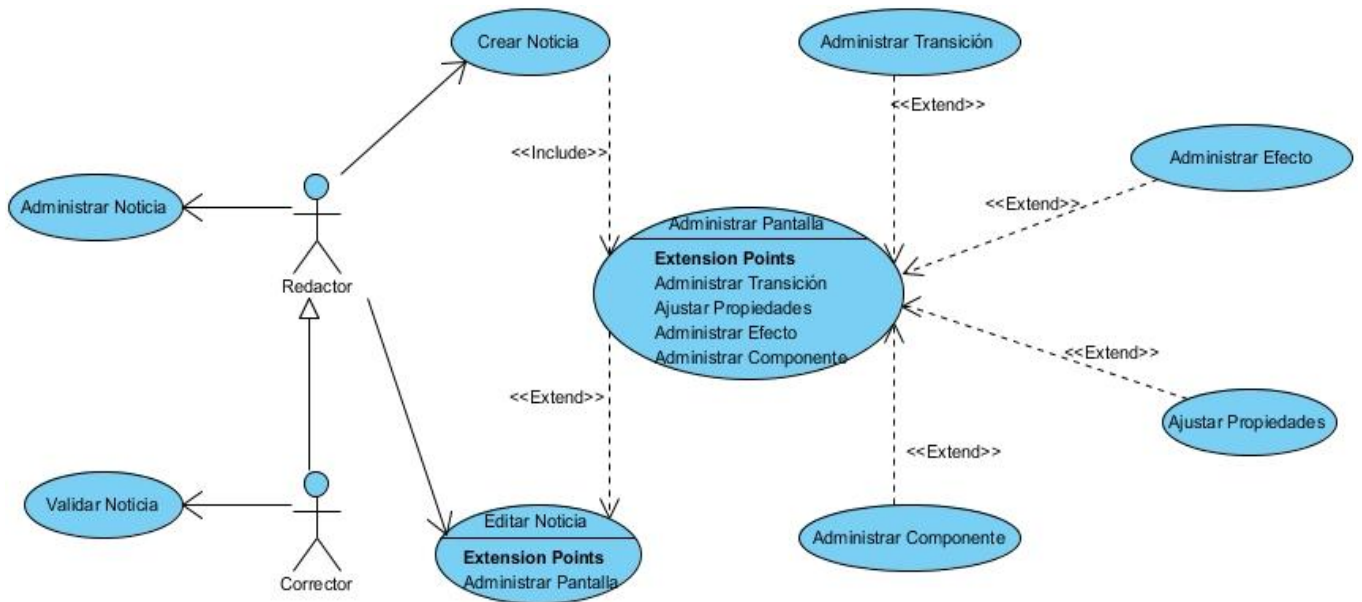


Figura 4. Diagrama de casos de uso del sistema

### 2.5 Descripción textual de los casos de uso del sistema

Un caso de uso es una unidad coherente de funcionalidad, externamente visible, proporcionada por una unidad del sistema y expresada por secuencias de mensajes intercambiados por la unidad del sistema y uno o más actores. El propósito de un caso de uso es definir una pieza de comportamiento coherente, sin revelar la estructura interna del sistema. Desde el punto de vista de los usuarios, éstas pueden ser situaciones anormales. Desde el punto de vista de los sistemas, son las variaciones adicionales que deben ser descritas y manejadas. (Booch, y otros, 1999)

Los casos de uso se usan para recopilar información acerca de cómo debe trabajar el sistema. Describen bajo la forma acción-reacción el comportamiento del sistema desde el punto de vista del usuario. En los Anexos I se realiza una profunda y detallada descripción de los casos de uso significativos del sistema.

### 2.6 Conclusiones

En este capítulo se procede a la confección de diferentes diagramas que posibilitan un mayor entendimiento del sistema en cuestión. Se realizan el Modelo de Dominio para comprender el negocio y el Modelo de casos de uso del sistema para tener un mayor entendimiento del sistema a desarrollar. También se realiza la captura de requisitos y se describen los casos de uso del sistema lo que posibilita el siguiente grupo de conclusiones. Permite

## *Capítulo II: Presentación de la Solución Propuesta*

- La confección del diagrama de clases del dominio brindó la interrelación existente entre cada una de las clases, esto permitió hacer más comprensible el negocio para su posterior implementación.
- La elaboración del diagrama de caso de uso del sistema permitió mostrar de manera explícita cómo funciona el software y la relación de cada uno de sus actores con el sistema. Permitió un mayor entendimiento al equipo de desarrollo del sistema a construir.
- La captura de los requisitos y la posterior descripción de los Casos de Uso aseguró que el sistema cumpla con el objetivo inicialmente trazado. Contribuyó a certificar que su desarrollo proceda de manera eficiente y eficaz, logrando la calidad en las prestaciones del mismo.

## CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA

### **3.1 Introducción**

La base de este capítulo está en la confección de la solución propuesta para el problema planteado, se muestra un modelado de las clases que se han implementado para lograr una visualización de los elementos a realizar. Los diagramas de clases del diseño elaborados a continuación describen la estructura que tendrá el módulo de Redacción de Noticias al concluir la etapa de implementación. Se muestran desglosados todos los patrones que han sido necesarios utilizar para lograr un mayor entendimiento de los componentes mostrados.

### **3.2 Modelo de análisis**

El modelado del análisis constituye un proceso de vital importancia en el desarrollo de un software, representa una abstracción del sistema que se está modelando. Ayuda a comprender con una mayor profundidad acerca de los procesos que internamente se desarrollan.

Fundamentalmente se basa en la comprensión de los requisitos funcionales identificados refinándolos y estructurándolos, por lo cual se facilita la visión de las funciones reales del sistema basándose en las actividades que se realizan en las fases de Diseño y la Implementación. Sin embargo este paso no es obligatorio, ya que la metodología es altamente configurable y que por tanto hay artefactos que pueden obviarse durante el desarrollo.

El modelo del análisis es opcional. En él, se describen un conjunto de Clases del Análisis que se utilizan para realizar una descripción abstracta de la realización de los casos de uso del modelo de casos de uso. Las clases del análisis luego evolucionan hacia otras clases más detalladas en el Modelo del Diseño. (Torossi, 2008)

Teniendo en cuenta lo anterior se decide hacer una transición del flujo de trabajo de Modelo de casos de uso al Modelo de diseño sin necesidad de realizar el Modelo de análisis. Teniendo en cuenta que Symfony2 tiene sus características relacionado con el diseño y la arquitectura de software, que los requisitos son bien conocidos, el hecho de que los lenguajes de programación, los frameworks y en

general las tecnologías sobre las que se estará desarrollando el sistema sean ya elementos dominados, se prescindirá del flujo de trabajo Modelo de Análisis.

### 3.3 Symfony2 y jQuery

Symfony2 y jQuery son dos frameworks para las aplicaciones web populares por su amplia documentación y por su rapidez, hecho este que los convierte en dos usados frameworks. A continuación se detallará la Arquitectura Modelo-Vista-Controlador utilizada por Symfony2 así como una breve descripción de los patrones importantes usados por estos dos frameworks.

#### 3.3.1 Arquitectura de Symfony2

La arquitectura de Symfony2 se centra en la popular arquitectura Modelo-Vista-Controlador (MVC), esta es la usada por la mayor parte de los frameworks para php. Sin embargo para su creador:

Symfony2 no es un framework MVC. Symfony2 sólo proporciona herramientas para la parte del Controlador y de la Vista. La parte del Modelo es responsabilidad tuya, aunque existen librerías para integrar fácilmente los ORM más conocidos, como Doctrine y Propel. (Potencier, 2011)

En cualquier caso, resulta esencial conocer cómo se aplican los principios fundamentales de la arquitectura MVC a las aplicaciones Symfony2. Observa el siguiente esquema simplificado del funcionamiento interno de Symfony2 (Eguiluz, 2011)

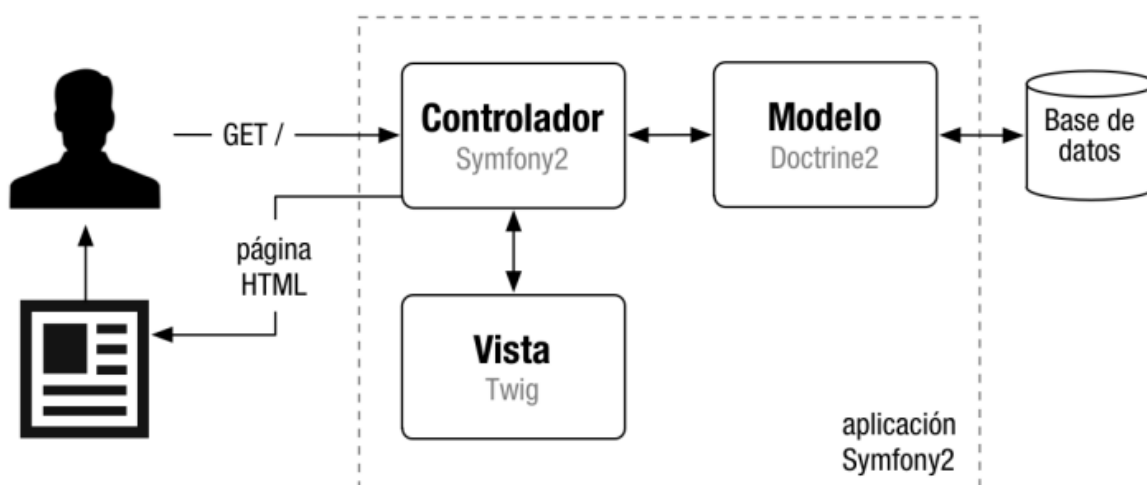


Figura 5. Funcionamiento interno de Symfony2

## Capítulo III: Construcción de la Solución Propuesta

Cuando el usuario solicita ver la portada del sitio, internamente sucede lo siguiente: (Eguiluz, 2011)

- El sistema de enrutamiento determina qué Controlador está asociado con la página de la portada.
- Symfony2 ejecuta el Controlador asociado a la portada. Un controlador no es más que una clase PHP en la que puedes ejecutar cualquier código que quieras.
- El Controlador solicita al Modelo los datos de la oferta del día. El modelo no es más que una clase PHP especializada en obtener información, normalmente de una base de datos.
- Con los datos devueltos por el Modelo, el Controlador solicita a la Vista que cree una página mediante una plantilla y que inserte los datos del Modelo.
- El Controlador entrega al servidor la página creada por la Vista.

A pesar de llegar a hacer cosas muy complejas con Symfony2, el funcionamiento interno siempre es el mismo: 1) el Controlador manda y ordena, 2) el Modelo busca la información que se le pide, 3) la Vista crea páginas con plantillas y datos. (Eguiluz, 2011)

### 3.3.2 Patrones de diseño de Symfony2 y jQuery

Los patrones de diseño son principios generales de soluciones que aplican ciertos estilos que ayudan a la creación de software. Un patrón es un par/problema solución con nombre que se puede aplicar en nuevos contextos, con consejos acerca de cómo aplicarlo en nuevas situaciones y discusiones sobre sus compromisos. Es una descripción de un problema y la solución a la que le da el nombre y que se puede aplicar en nuevos contextos. Muchos patrones ayudan a asignar responsabilidades a los objetos. (Olivares, 2007)

Un patrón de diseño es el cimiento para buscar una solución a un problema común en el desarrollo de un software. Un patrón sugiere algo repetitivo, algún problema conocido y su respectiva solución. A continuación se mencionarán algunos de los importantes patrones de diseño usados en Symfony2 y jQuery.

#### 3.3.2.1 Patrón de bajo acoplamiento

El Bajo Acoplamiento es un principio que debemos recordar durante las decisiones de diseño: es la meta principal que es preciso tener siempre. Es un patrón evaluativo que el diseñador aplica al juzgar sus decisiones de diseño. (Larman, 2003)

## *Capítulo III: Construcción de la Solución Propuesta*

En la aplicación las clases `DefaultController` y `CrearController` heredan solamente de `Controller`, que es una clase estable en cuanto a su implementación por lo que se garantiza que exista un grado moderado de acoplamiento entre las clases.

### **3.3.2.2 Patrón creador**

El patrón creador guía la asignación de responsabilidades relacionadas con la creación de objetos, tarea muy frecuente en los sistemas orientados a objetos. El propósito fundamental de este patrón es encontrar un creador que debemos conectar con el objeto producido en cualquier evento. Al escogerlo como creador, se da soporte al bajo acoplamiento. (Larman, 2003)

En `Symfony2` en la clase `Controller` se definen y ejecutan todas las acciones. En las acciones se crean los objetos de las clases que representan las entidades, evidenciando de este modo que la clase `Controller` es "creador" de dichas entidades.

### **3.3.2.3 Patrón controlador**

La mayor parte de los sistemas reciben eventos de entrada externa, los cuales generalmente incluyen una interfaz gráfica para el usuario (IGU) operado por una persona. Otros medios de entrada son los mensajes externos –entre ellos un conmutador de telecomunicaciones para procesar llamadas- o las señales precedentes de sensores como sucede en los sistemas de control de procesos. (Larman, 2003)

En todos los casos si se recurre a un diseño orientado a objetos, hay que elegir los controladores que manejen esos eventos de entrada. Este patrón ofrece una guía para tomar decisiones apropiadas que generalmente se aceptan. (Larman, 2003)

En `Symfony2` todas las peticiones web son manejadas por el controlador frontal, que es el punto de entrada único de toda la aplicación en un entorno determinado.

### **3.3.2.4 Patrón decorador**

Propósito: Añadir responsabilidades adicionales a un objeto dinámicamente, proporcionando una alternativa flexible a la especialización mediante herencia, cuando se trata de añadir funcionalidades. (Facultad de informática - Universidad Politécnica de Madrid, 2005)

Este patrón añade funcionalidad a una clase de una forma dinámica y transparente. En la aplicación el archivo `layout.html.twig` o plantilla global almacena el código HTML que es común a todas las páginas



## Capítulo III: Construcción de la Solución Propuesta

de la aplicación, para no tener que repetirlo en cada página. El contenido de las plantillas se integra en los archivos `index.html.twig` y `crear.html.twig`.

### 3.3.2.5 Patrón adaptador

Propósito: Convertir la interfaz de una clase para que se adapte a lo que el cliente que la usa necesita, permitiendo así que trabajen juntas clases cuyas interfaces son incompatibles. (Facultad de informática - Universidad Politécnica de Madrid, 2005)

Este patrón se utiliza en los archivos `imagen.js`, `texto.js` y `video.js` cuando se le establece al componente el valor de la opacidad en 1.

Ejemplo: `imagenSeleccionada.css('opacity',1);`

### 3.3.2.6 Patrón observador

Propósito: Un efecto lateral muy frecuente en aquellos sistemas que se fragmentan en un conjunto de clases que cooperan es la necesidad de mantener la consistencia entre los distintos objetos interrelacionados. Para no recurrir a soluciones fuertemente acopladas (que reducen la posibilidad de reutilización), este patrón define una dependencia “uno-a-muchos” entre objetos, para que, cuando uno de ellos cambie su estado, todos los que dependan de él sean avisados y puedan actualizarse convenientemente. (Facultad de informática - Universidad Politécnica de Madrid, 2005)

La utilización de este patrón de diseño se ve reflejada en el archivo `eventos.js` cuando queremos capturar los eventos que se activan cuando el usuario presiona Clic Izquierdo o Clic Derecho sobre un elemento determinado.

Ejemplo: `$("#previsualizarId").live('click', function(){...});`

`$(".Imagen").live("mouseup", function(event){...});`

### 3.3.2.7 Patrón fachada

Propósito: Proporcionar una interfaz unificada de alto nivel que, representando a todo un subsistema, facilite su uso. La “fachada” satisface a la mayoría de los clientes, sin ocultar las funciones de menor nivel a aquellos que necesiten acceder a ellas. (Facultad de informática - Universidad Politécnica de Madrid, 2005)

El uso de este patrón GoF se puede observar en la utilización de las funciones de jQuery `$get()` y `$post()` de los archivos `noticia.js` y `noticias.js`

## Capítulo III: Construcción de la Solución Propuesta

Ejemplo: `$.get("/noticia/crear/guardar/", {...}, json);`

`$.post("/noticia/eliminarVarias/", {...}, json);`

### 3.3.2.8 Patrón constructor

Propósito: Separar la construcción y la representación de un objeto complejo, para así permitir que el mismo proceso de construcción sirva para crear diferentes representaciones. (Facultad de informática - Universidad Politécnica de Madrid, 2005)

Este patrón creacional se usa en la función Previsualizar del archivo noticia.js cuando se crean objetos para la previsualización de la noticia.

Ejemplo: `var previo = $('<div id="previo" style="position:relative"></div>');`

## 3.4 Modelo de diseño

En la elaboración y confección de un software es importante realizar previamente un correcto diseño para facilitar la implementación del mismo, ya que las bases siempre estarán fomentadas en gran medida en la eficacia de lo que se ha diseñado anteriormente. Un modelo de diseño es básicamente un modelo físico que proporcionará la estructura y la forma que tendrá el sistema que se está construyendo. Es de vital importancia que se estudien anteriormente todos los requisitos que se hayan detectado para sobre esa base y la de las limitaciones de los framework con los que se estén programando trazar todas las estrategias a desarrollar. Para un mejor entendimiento en los Anexos II se pueden apreciar los Diagramas de clases del diseño relevantes del sistema de una manera detallada.

## 3.5 Diseño de la base de datos

El objetivo fundamental del diseño de la base de datos es garantizar que los datos persistentes se almacenen de forma coherente y eficaz. ( Copyright IBM, 2006) Para diseñar la base de datos se utilizan el diagrama de clases persistentes y el diagrama entidad-relación. El diseño de la base de datos está compuesto por seis tablas debidamente normalizadas para evitar la repetición de datos. A continuación se presentan el Diagrama de clases persistentes y el Diagrama entidad-relación.

### 3.5.1 Diagrama de clases persistentes

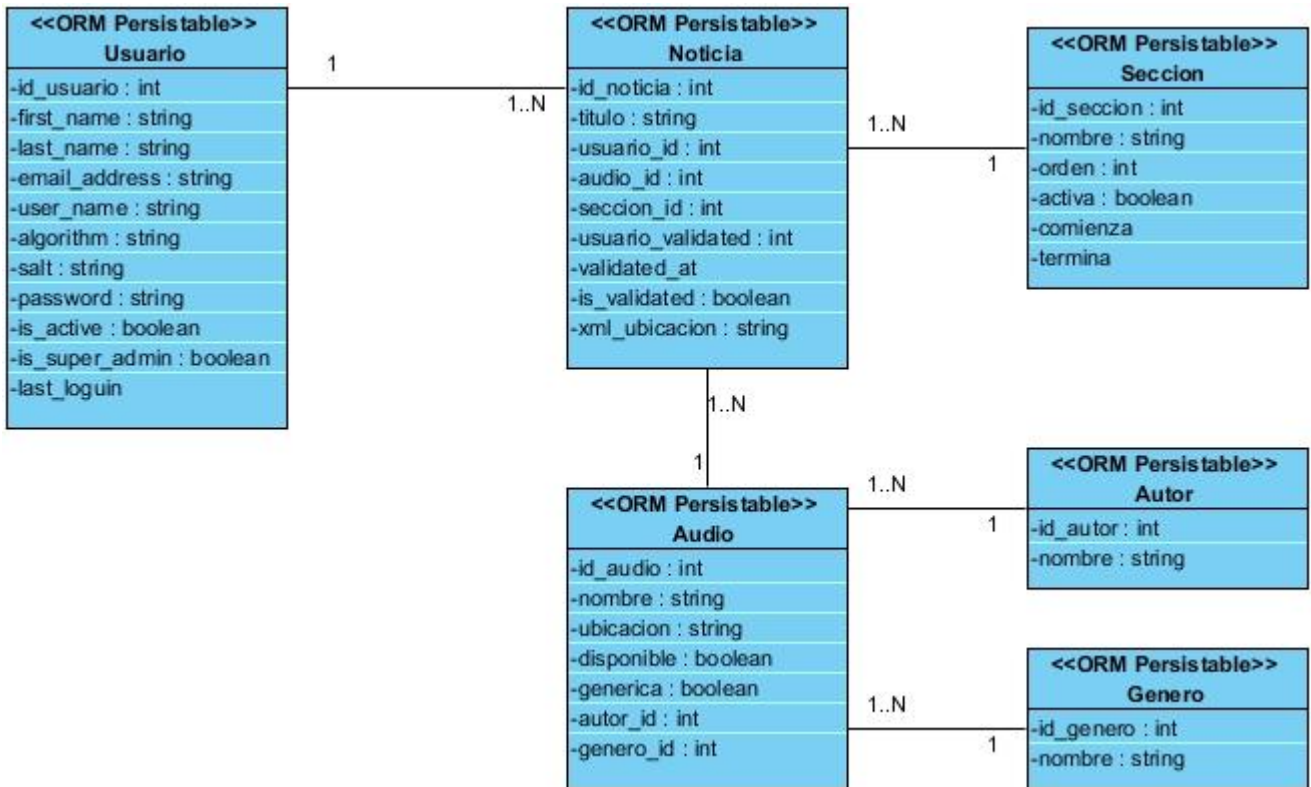


Figura 6. Diagrama de clases persistentes

### 3.5.2 Modelo entidad-relación

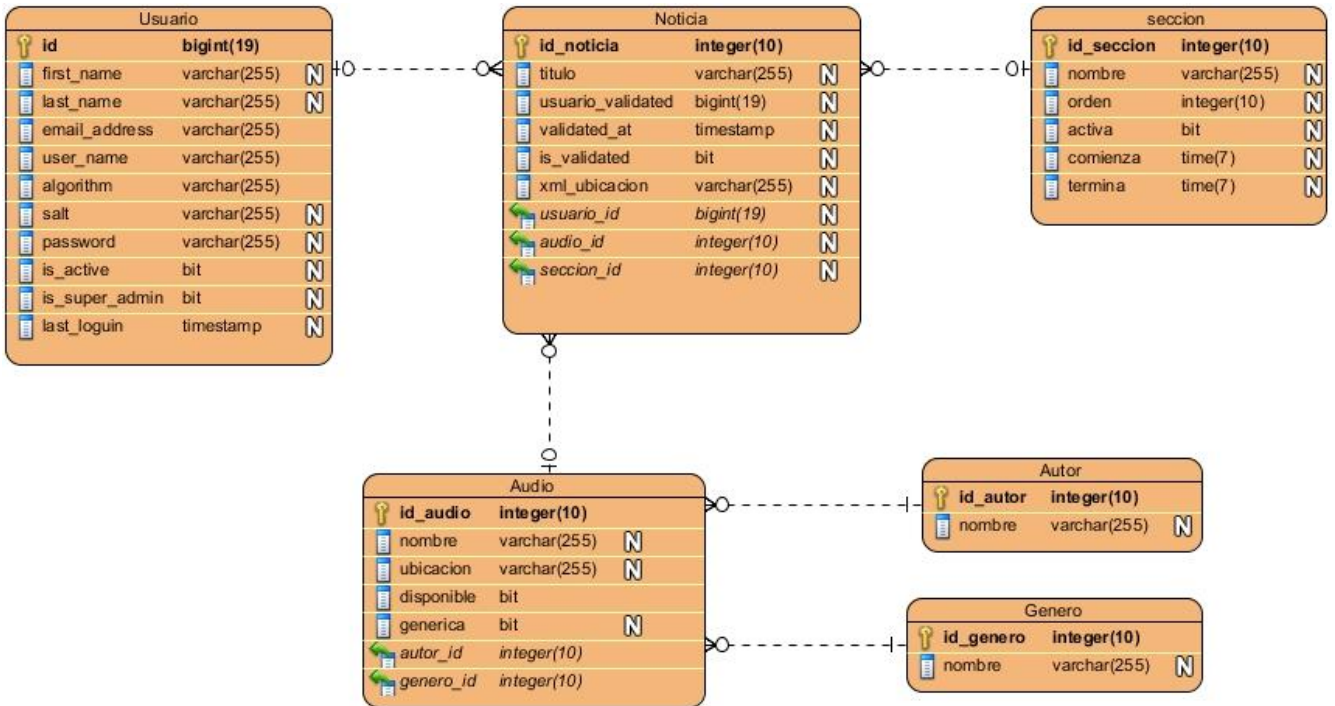


Figura 7. Diagrama entidad-relación

### 3.6 Modelo de despliegue

El modelo de despliegue describe la distribución física del sistema, muestra como están distribuidos los componentes de software entre los distintos nodos de cómputo. Permite comprender la correspondencia entre la arquitectura software y la arquitectura hardware. (Hernández, y otros, 2007)

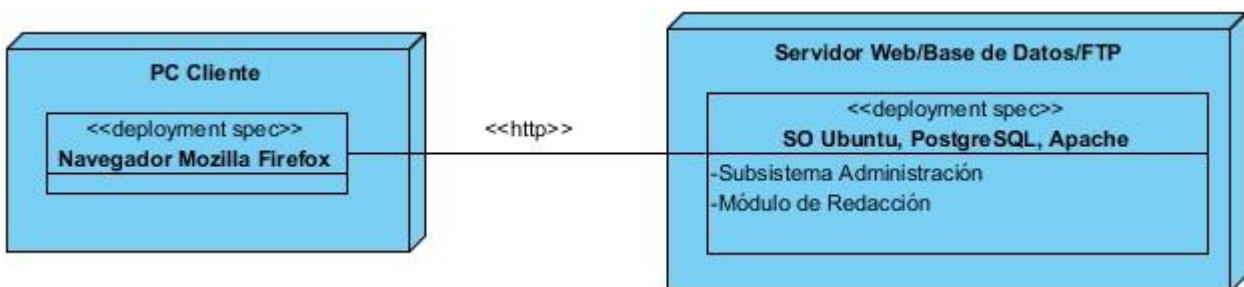


Figura 8. Diagrama de despliegue

## 3.7 Modelo de Implementación

Los diagramas de componentes son usados para estructurar el modelo de implementación en términos de subsistemas de implementación y mostrar las relaciones entre los elementos de implementación. También se utilizan para mostrar las dependencias de compilación de ficheros, relaciones de derivación entre los ficheros que son resultado de la compilación, dependencias entre elementos de implementación y los correspondientes elementos de diseños que son implementados. (Booch, y otros, 1999)

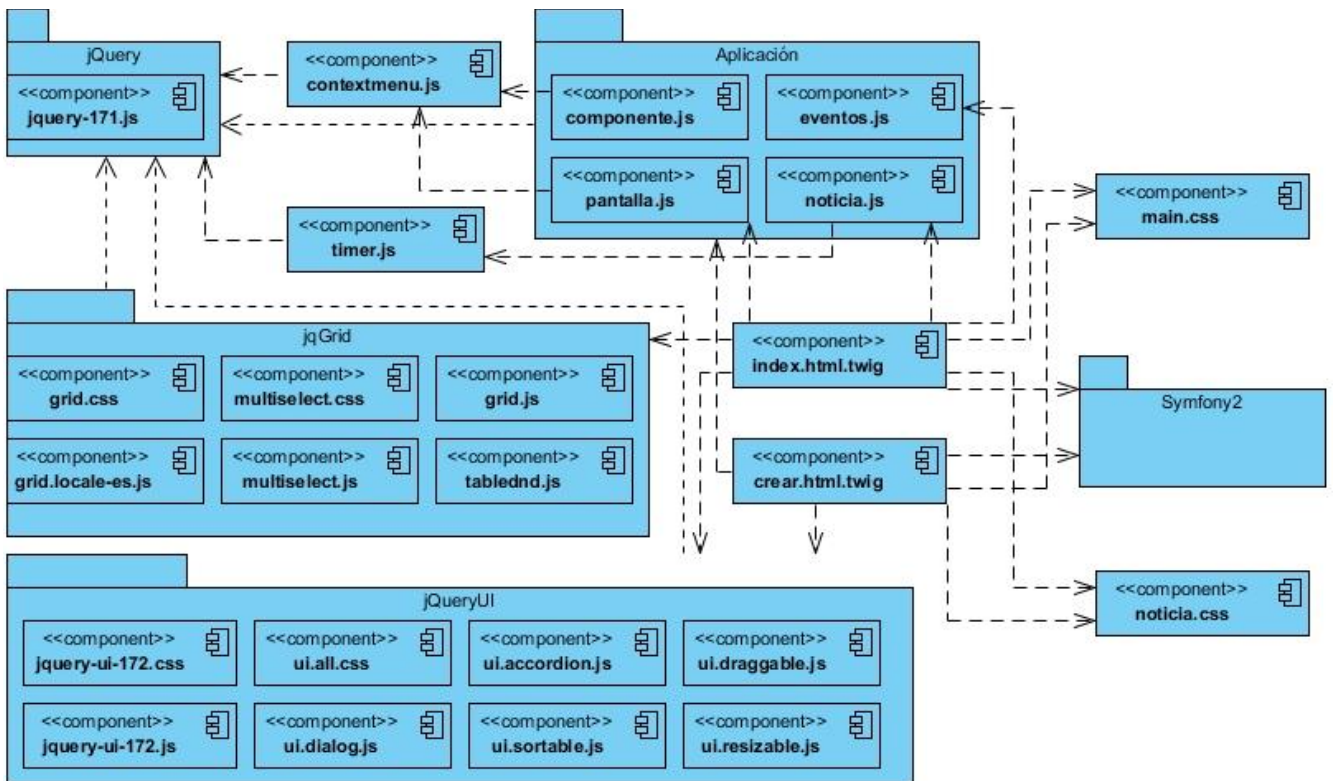


Figura 9. Diagrama de componentes

## 3.8 Validación del sistema

En este epígrafe trataremos las pruebas de aceptación del cliente realizadas para la validación del módulo de Redacción de Noticias para la Plataforma de Televisión Informativa PRIMICIA v2.0. Las pruebas que se desarrollarán serán del tipo: pruebas de aceptación final del cliente, al efectuarse la entrega de la versión correspondiente. Los artefactos, condiciones necesarias y documentos rectores de esta actividad, así como todo lo relacionado a su completo y efectivo cumplimiento.

## *Capítulo III: Construcción de la Solución Propuesta*

A continuación se abordan los aspectos significativos de estas pruebas y la recolección y valoración de no conformidades y solicitudes de cambio de las pruebas de aceptación final del cliente.

### **3.8.1 Prueba de aceptación final del cliente**

Las pruebas de aceptación final del cliente son las encargadas de asegurar el cumplimiento de los Procesos Elementales del Negocio, los Casos de Uso, lo aceptado en los prototipos y lo planteado en el proyecto técnico.

#### **3.8.2.1 Objetivos generales de las pruebas**

- Ejecutar pruebas de software en un ambiente real y bajo la supervisión del equipo de desarrollo.
- Probar el rendimiento de la aplicación.
- Detectar casos críticos.
- Determinar la factibilidad de despliegue del software desarrollado.

#### **3.8.2.2 Fases para la realización de las pruebas**

Todas las pruebas se realizarán de forma manual y se probarán todas las funcionalidades implementadas en el módulo según la metodología del cliente.

Estas pruebas se harán teniendo en cuenta los siguientes pasos: (Rivera, y otros, 2010)

- Organización de los escenarios de pruebas y capacitación del equipo de pruebas.
- Realización de pruebas de funcionalidad.
- Realización de las pruebas de interfaz de usuario.
- Se realizarán pruebas de regresión para chequear que los errores ya corregidos no se vuelvan a repetir y/o no se dañen funcionalidades que funcionaban correctamente.
- Se realizarán Pruebas de Seguridad y Control de Acceso con el objetivo de validar la protección de la aplicación sensible a entradas no deseadas.
- Realizar las pruebas de integridad de los datos y la base de datos.
- Se realizarán las pruebas de diseño informacional
- Realizar las pruebas de referencia cruzada
- Aplicación de la lista de chequeo y los principios de calidad.
- Conciliación de los documentos entregables resultados de las pruebas, aprobación y firma de los mismos.

## Capítulo III: Construcción de la Solución Propuesta

### 3.8.2.3 Participantes en las pruebas de validación

Especialista	Rol
Ing. Carlos de Jesús Andrés González	Líder de Proyecto
Ing. Lisandra Delgado Cabrera	Analista del Proyecto
Ing. Félix Iván Romero Rodríguez	Jefe del Subsistema de Administración

Tabla 2. Especialistas y rol de las personas participantes en las pruebas de aceptación

### 3.8.2.4 Flujo de trabajo

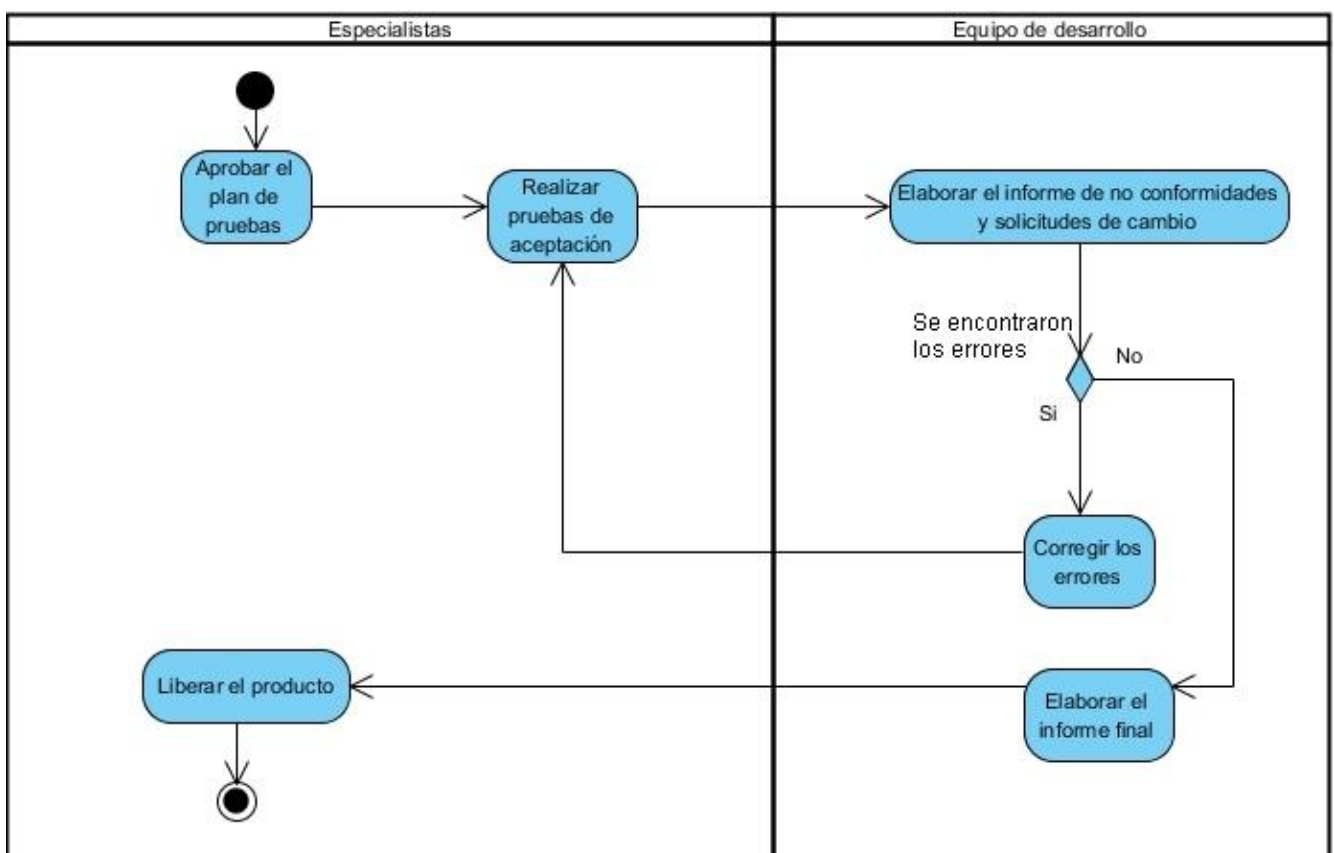


Figura 10. Flujo de trabajo de las pruebas de aceptación

### 3.8.2.5 Descripción del flujo de trabajo

Las pruebas de aceptación final del cliente se inician con la aprobación y firma del Plan de Pruebas que incluye el plan de trabajo detallado, la relación de los datos del personal que intervendrán en las pruebas y la selección de los funcionalidades a probar. De manera opcional se puede asumir el paso donde se introduce a los especialistas funcionales al uso de las funcionalidades a probar. Garantizadas

## Capítulo III: Construcción de la Solución Propuesta

todas las condiciones técnicas y verificadas por la dirección representante de la parte especialista, se puede dar inicio al desarrollo de las pruebas de las funcionalidades planificadas.

Para ello, intervienen el Líder de Proyecto, la Analista del Proyecto, el Arquitecto del Proyecto y el Jefe del Subsistema de Administración de forma directa con el software y como apoyo, el equipo de desarrollo, esclareciendo las dudas de las funcionalidades que lo requieran. Durante las pruebas se verificará el cumplimiento de los Procesos Elementales del Negocio, los Requisitos Funcionales y los Casos de Uso.

Las no conformidades que surjan durante las pruebas serán analizadas y valorada su solución, que le será asignada al equipo de desarrollo para ser ejecutada. La respuesta se realiza de manera inmediata, brindando una nueva solución al cliente. Las solicitudes de cambios que aparezcan serán analizadas y llevadas a la mesa de negociaciones entre ambas partes; las aceptadas y pactadas pasarán a ser nuevos requisitos.

Al concluir cada ciclo se reiniciará con una nueva revisión de la aplicación, partiendo de las No Conformidades pendientes a solución y de las solicitudes aceptadas, verificándose la solución dada y la integridad del sistema. Terminada la revisión se elabora un informe final sobre el resultado de las pruebas y se libera el sistema.

### 3.8.2.6 Acción de recolección y valoración de no conformidades y solicitudes de cambio

Descripción	Tipo de observación	Estado
Las pantallas que contengan algún video deben tener una duración mayor a la del tiempo total del video.	No Conformidad	Resuelta
La interfaz de usuario de la aplicación no se adecua al diseño del proyecto	No Conformidad	Resuelta
El sistema en algunas ocasiones no previsualiza la noticia correctamente.	No Conformidad	Resuelta
Algunos efectos no logran la animación visual esperada por el usuario.	No Conformidad	Resuelta
Al reordenar las pantallas el sistema no actualiza de manera correcta la Región Previo así como las pantallas de la misma.	No Conformidad	Resuelta

Tabla 3. Recopilación y valoración de no conformidades



## *Capítulo III: Construcción de la Solución Propuesta*

Los resultados expuestos en la tabla anterior corresponden a las pruebas de aceptación realizadas al módulo de Redacción de Noticias de la Plataforma de Televisión Informativa PRIMICIA una vez concluida la implementación de su versión 2.0. Las inquietudes encontradas fueron de tipo No Conformidad ya que se originaron por una incorrecta implementación de un requisito previamente pactado. No se encontraron inquietudes de tipo Solicitud de Cambio. En general se encontraron 5 No Conformidad dentro del sistema, las cuales fueron correctamente resueltas por el equipo de desarrollo encargado. Los resultados fueron satisfactorios y quedaron satisfechos los usuarios, especialistas e informáticos de la institución.

### **3.9 Conclusiones**

Durante la construcción de la solución propuesta y su posterior validación se arribaron a varias conclusiones, a continuación se detallan algunas:

- Durante la etapa de análisis y diseño del módulo Redacción de Noticias de PRIMICIA v2.0 se realiza la confección de los diagramas de clases del diseño, los que han sido presentados en el actual capítulo. Los mismos representan la visión de todos los requisitos detectados que han sido materializados como las funcionalidades del problema planteado.
- La correcta integración de Symfony2 con JQuery ha constituido la base fundamental para lograr el éxito en la solución del problema inicial. Estos frameworks han posibilitado un sistema rápido, con una arquitectura MVC y una variedad de efectos para garantizar las prestaciones del sistema.
- El diseño de la base de datos así como el Modelo de despliegue y el Modelo de implementación permite completar la construcción del software. Se garantiza que los datos se almacenen de manera vinculada y efectiva, permite describir la distribución física del sistema y posibilita mostrar las relaciones entre los elementos de implementación.
- Durante la validación del sistema mediante las pruebas de aceptación se pudo ejecutar pruebas de software en un ambiente real y bajo la supervisión del equipo de desarrollo, probar el rendimiento de la aplicación, detectar casos críticos y determinar la factibilidad de despliegue del software desarrollado.

## CONCLUSIONES

El sistema que se obtuvo como resultado de esta investigación, el módulo de Redacción de Noticias para la Plataforma de Televisión Informativa PRIMICIA v2.0 es hoy una realidad que permite al redactor de la plataforma confeccionar una noticia agradable a la vista del televidente. Esta afirmación se debe a las opciones de configuración que tiene el redactor al crear la noticia, logrando modificar las características de los componentes que contiene así como la aplicación de una notable cantidad de efectos para hacer dinámica y atractiva la visualización de las noticias. Partiendo de este hecho se puede concluir que:

- La elaboración de los diferentes diagramas ayudó a tener un mayor entendimiento del sistema en cuestión al equipo de desarrollo. Además estos diagramas formarán parte de los documentos que serán entregados al proyecto una vez finalizado el producto.
- El uso de las herramientas y tecnologías seleccionadas brindaron las prestaciones necesarias para la elaboración de un robusto sistema. También permitió cumplir con la política de soberanía tecnológica por la que proclama el país.
- La utilización de la metodología escogida garantizó el éxito en el desarrollo del producto. El tránsito sus fases permitió generar los artefactos y documentación de Ingeniería de Software necesarios. Dicha documentación proporcionada podrá ser utilizada en versiones futuras del módulo.
- La implementación del nuevo módulo de Redacción de Noticias de la Plataforma Informativa PRIMICIA contribuyó a cumplir con las expectativas del cliente. Permitted obtener como resultado un producto fiable y de calidad avalado por las pruebas de aceptación.

Por todo lo expuesto anteriormente se puede afirmar que con el Desarrollo del módulo de Redacción de Noticias para la Plataforma de Televisión Informativa PRIMICIA v2.0 se garantizará la configuración completa de los elementos visuales de la noticia, por lo que se han alcanzado satisfactoriamente los objetivos y las tareas propuestas para el presente trabajo de diploma.

### **RECOMENDACIONES**

Al término de este trabajo se recomienda:

- Ampliar la creación de componentes, entre ellos el componente clima, el componente tabla y el componente infocinta.
- Continuar aplicando procesos de prueba más profundos que permitan refinar las funcionalidades existentes.
- Divulgar el 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 requisitos que surjan.
- Garantizar la disponibilidad del sistema durante las 24 horas del día para que la información fluya de manera que cada usuario pueda acceder al sistema en el momento que lo necesite.

## BIBLIOGRAFÍA

**Copyright IBM. 2006.** *Ayuda RUP.* 2006.

**Alegsa. 2011.** [En línea] 2011. <http://www.alegsa.com.ar/Dic/modulo.php>.

—. Diccionario Informático. *Diccionario Informático.* [En línea] Alegsa. [Citado el: 18 de Noviembre de 2011.] <http://www.alegsa.com.ar/Dic/uml.php>.

**Alvarez, Miguel Angel. 2010.** *Manual de jQuery.* 2010.

**Andrés, González Carlos de Jesús. 2010.** *Diseño del Subsistema de Configuración de la Plataforma de Televisión.* Ciudad de La Habana : s.n., 2010.

**Barrios, Cardozo Daniel y Torres Camejo, Marisleidy. 2009.** *Desarrollo del prototipo funcional encargado de generar el formato XML de la Plataforma de Televisión Informativa PRIMICIA.* Ciudad Habana : s.n., 2009.

**Bass, L., Clements, P. y Kasman, R. 1998.** *Architecture in practice.* s.l. : Addison-Wesley, 1998. 0201199300.

**Benítez, Calzado Frank. 2009.** *Procedimiento para el seguimiento y tratamiento de los errores de PRIMICIA.* Ciudad de la Habana : s.n., 2009.

**Benítez, Carlos. 2012.** Etnassoft. *Etnassoft.* [En línea] 2012. <http://www.etnassoft.com/2011/05/26/patrones-de-diseno-utilizados-por-jquery/>.

**Bestwebframeworks. 2012.** Bestwebframeworks. *Bestwebframeworks.* [En línea] 2012. <http://www.bestwebframeworks.com/compare-web-frameworks/php/>.

**Booch, Grady, Rumbaugh, James y Jacobson, Ivar. 1999.** *El Lenguaje Unificado de Modelado.* Madrid : s.n., 1999.

**Buschmann, F., y otros. 1996.** *Pattern – Oriented Software Architecture. A System of Patterns.* 1996. 978-0471958697.

**Camacho, Erika, Cardeso, Fabio y Nuñez, Gabriel. 2004.** *Arquitecturas de software.* s.l. : Universidad Simón Bolívar, 2004.

**Cano, Zárate José Carlos. 2007.** *Apuntes de Tecnología Educativa para las NT Cátedras de Comunicación Educativa Universidad Marista y Apuntes de Diseño Instruccional.* México : s.n., 2007.

**Centro Nacional de Tecnologías de Información. 2010.** MeRinde. *MeRinde.* [En línea] Centro Nacional de Tecnologías de Información, 2010. [Citado el: 15 de Febrero de 2012.] [http://merinde.net/index.php?option=com\\_content&task=view&id=96&Itemid=297](http://merinde.net/index.php?option=com_content&task=view&id=96&Itemid=297).

**Cibertenia.** Cibertenia. *Cibertenia.* [En línea] UOC. [Citado el: 30 de Abril de 2012.] [http://www.cibertenia.com/manuales/introduccion\\_aplicaciones\\_web/3\\_historia\\_aplicaciones\\_web.php](http://www.cibertenia.com/manuales/introduccion_aplicaciones_web/3_historia_aplicaciones_web.php).

**Definicion. 2011.** [En línea] 2011. <http://definicion.de/modulo/>.

**Definicion.de. 2008.** Definicion.de. *Definicion.de.* [En línea] 2008. [Citado el: 29 de Abril de 2012.] <http://definicion.de/web/>.

**definición.org.** definición.org. *definición.org.* [En línea] [Citado el: 17 de Noviembre de 2011.] <http://www.definicion.org/lenguaje-de-programacion>.

**Delgado, Cabrera Lisandra. 2010.** *Diseño de las nuevas funcionalidades del Módulo de Redacción de la Plataforma de Televisión Informativa PRIMICIA.* Ciudad de la Habana : s.n., 2010.

**Díaz-Antón, María Gabriela. 2003.** *Propuesta de una metodología de desarrollo de software educativo bajo un enfoque de calidad sistémica.* 2003.

**Eguiluz, Pérez Javier. 2011.** *Desarrollo web ágil con Symfony2.* 2011.

**Eguíluz, Pérez Javier. 2005.** *Introducción a AJAX.* s.l. : Autoedición, 2005. 1909-2520.

—. **2009.** *Introducción a JavaScript.* 2009.

*El documentalista audiovisual del Centro de Documentación de Barcelona Televisión.* **Martínez, Virginia y Mas, Tomeu. 2010.** 24, Barcelona : Facultad de Biblioteconomía y Documentación, Universidad de Barcelona, 2010, Vol. I. 1575-5886.

**Estructure. 2011.** [En línea] 2011. <http://www.estructure.es/enewsroom.htm>.

**Fabien Potencier, François Zaninotto. 2009.** *Symfony la guía definitiva Doctrine.* s.l. : Apress, 2009. 978-1590597866.

- Facultad de informática - Universidad Politécnica de Madrid. 2005.** *Patrones del "Gang of Four"*. 2005.
- Fernández Vilas, Ana.** Laboratorio de Televisión Digital Interactiva. *Laboratorio de Televisión Digital Interactiva*. [En línea] <http://tvdi.det.uvigo.es/~avilas/UML/node50.html>.
- Frómata Lescay, Roberto. 2011.** *Subsistema de Control de Ingesta del Sistema*. La Habana : Universidad de las Ciencias Informáticas, 2011.
- . 2011. *Subsistema de Control de Ingesta del Sistema*. La Habana : s.n., 2011.
- Gil, Aros Celio. 2008.** *RUP: Metodología en los sistemas y aplicaciones basadas en la web*. 2008.
- Gómez Ballester, Eva y Martínez Barco, Patricio.** Universidad de Alicante. *Universidad de Alicante*. [En línea] [Citado el: 18 de Noviembre de 2011.] <http://www.alu.ua.es/j/jmr36/Conectate/Base%20Datos/Apuntes2006.pdf>.
- Gómez, Fernando J. 2012.** Instituto de Matemática de Bahía Blanca. *Instituto de Matemática de Bahía Blanca*. [En línea] INMAB-CONICET, 11 de Agosto de 2012. [Citado el: 29 de Abril de 2012.] <http://inmabb.criba.edu.ar/catalis/catalis.php?p=main>.
- González Arencibia, Mario. 2006.** Entorno virtual de aprendizaje. *Entorno virtual de aprendizaje*. [En línea] Enero de 2006. [Citado el: 17 de Enero de 2011.] <http://eva.uci.cu/mod/resource/view.php?id=10794>.
- Guía documentada para Ubuntu. 2012.** [En línea] 2012. [http://www.guia-ubuntu.org/index.php?title=Servidor\\_web#Servidor\\_Web\\_Apache](http://www.guia-ubuntu.org/index.php?title=Servidor_web#Servidor_Web_Apache).
- Hernández Moya, Iván. 2011.** *Migración del Módulo de Catalogación de materiales audiovisuales del producto Captura y catalogación de medias*. La Habana : Universidad de las Ciencias Informáticas, 2011.
- Hernández Sampieri, Roberto, Fernández Collado, Carlos y Baptista Lucio, Pilar. 1991.** *Metodología de la investigación*. s.l. : MCGRAW-HILL, 1991. 968-422-931-3.
- Hernández, García Ruber y Montaner Hernández, Yunior. 2007.** *Sistema Automatizado de Teletexto para la Plataforma de Televisión Digital Satelital Cubana*. Ciudad de la Habana : s.n., 2007.

- Hostito. 2012.** [En línea] 2012. [http://www.hostito.com/es/faq/index.php?section=glossary&title\\_section=Glosario%20de%20T%C3%A9rminos#30](http://www.hostito.com/es/faq/index.php?section=glossary&title_section=Glosario%20de%20T%C3%A9rminos#30).
- Larman, Craig. 2003.** *UML y patrones*. 2003.
- Learning Technology Standards Committee. 2002.** *IEEE Standard for Learning Object Metadata*. New York : IEEE, 2002. 0-7381-3297-7.
- Leopoldo, Carlos. 2007.** *techtastico*. [En línea] *techtastico*, 27 de Noviembre de 2007. [Citado el: 1 de abril de 2012.] <http://techtastico.com/post/zend-framework-una-introduccion/>.
- Letelier, Torres Patricio Orlando. 1998.** *Introducción a RUP*. 1998.
- Lorente, Miranda Rafael. 2009.** *Propuesta de mejoras para el proceso de redacción de noticia de la Plataforma de Televisión Informativa PRIMICIA*. Ciudad de la Habana : s.n., 2009.
- Lycos. 2012.** [En línea] 2012. [cine.lycos.es/glosario2.php?orden2=Voz-en-off](http://cine.lycos.es/glosario2.php?orden2=Voz-en-off).
- Marín Abreu, Angel Dayán. 2009.** *Desarrollo de la Plataforma VideoWeb. Rol Analista*. La Habana : Universidad de las Ciencias Informáticas, 2009.
- MultiMania. 2003.** *Temas para opositores a Ayudantes de Bibliotecas*. [En línea] 2003. [Citado el: 7 de Octubre de 2011.] [http://usuarios.multimania.es/obib/doc/doc\\_catalogación\\_2004.doc](http://usuarios.multimania.es/obib/doc/doc_catalogación_2004.doc).
- NetBeans. 2012.** [En línea] 2012. <http://www.netbeans.org>.
- Olivares Rojas, Juan Carlos.** Instituto Tecnológico de Morelia. *Instituto Tecnológico de Morelia*. [En línea] [Citado el: 30 de Noviembre de 2011.] [antares.itmorelia.edu.mx/~jcolivar/courses/dp07a/patrones.pdf](http://antares.itmorelia.edu.mx/~jcolivar/courses/dp07a/patrones.pdf).
- Olivares, Rojas Juan Carlos. 2007.** *Patrones de Diseño*. 2007.
- Oré, Alexander. 2009.** *CalidadSoftware.com*. [En línea] *CalidadSoftware.com*, 2009. [Citado el: 17 de Febrero de 2012.] [http://www.calidadsoftware.com/testing/pruebas\\_funcionales.php](http://www.calidadsoftware.com/testing/pruebas_funcionales.php).

**Ortiz Fidalgo, Jackeline y Rios Morales, Fredy Hector. 2010.** *Desarrollo de un componente de software para importar reportes desde documentos Excel a Sistemas Gestores de Bases de Datos en el Centro de Investigaciones del Petróleo (CEINPET).* 2010.

**Pacheco, Nacho. 2011.** *Manual de Symfony2.* 2011.

**Pérez, Javier Eguíluz. 2009.** *Introducción a CSS.* s.l. : Autoedición, 2009. 0188-7297.

**PostgreSql. 2012.** [En línea] 2012. [http://www.postgresql.org.es/sobre\\_postgresql](http://www.postgresql.org.es/sobre_postgresql).

**Potencier, Fabien. 2009.** *El tutorial Jobeet.* s.l. : librosweb-es, 2009. 291839016X.

—. **2011.** *What is Symfony2?* 2011.

**Pressman, Roger S. 2005.** *Ingeniería de Software (6ta Edición).* 2005.

**Pressman, Roger S. 2010.** *Ingeniería de Software: un enfoque práctico.* s.l. : Higher Education, 2010. 978-0-07-337 597-7.

**Pwned's blog. 2007.** Pwned's blog - Desarrollo de Tecnología. *Pwned's blog - Desarrollo de Tecnología.* [En línea] Pwned's blog, 2 de Noviembre de 2007. [Citado el: 1 de abril de 2012.] <http://pwneds.blogspot.com/2007/11/comparacion-de-10-frameworks-de-php.html>.

**Quintanilla Ramírez, Olga María. 2005.** Sistema de Bibliotecas - Universidad Nacional Mayor de San Marcos. *Sistema de Bibliotecas - Universidad Nacional Mayor de San Marcos.* [En línea] 21 de Febrero de 2005. [Citado el: 8 de Octubre de 2011.] [sisbib.unmsm.edu.pe/publicacion/presenta/pdf/cata\\_prin\\_inc.pdf](http://sisbib.unmsm.edu.pe/publicacion/presenta/pdf/cata_prin_inc.pdf).

**Real Academia Española. 2011.** [En línea] 2011. [http://buscon.rae.es/drae/SrvltConsulta?TIPO\\_BUS=3&LEMA=modulo](http://buscon.rae.es/drae/SrvltConsulta?TIPO_BUS=3&LEMA=modulo).

—. **2011.** [En línea] 2011. [http://buscon.rae.es/drae/SrvltConsulta?TIPO\\_BUS=3&LEMA=implementaci%C3%B3n](http://buscon.rae.es/drae/SrvltConsulta?TIPO_BUS=3&LEMA=implementaci%C3%B3n).

—. **2001.** Diccionario de la Lengua Española - Vigésima segunda edición. *Diccionario de la Lengua Española - Vigésima segunda edición.* [En línea] Real Academia Española, Octubre de 2001. [Citado el: 7 de Octubre de 2011.] <http://buscon.rae.es/drae/>.



**Rivera, Carbonell Nelson y Fernández López, Cristian. 2010.** *Diseño, implementación y validación del módulo Control de armamento, equipos y medios de seguridad del Sistema de Gestión Penitenciaria.* 2010.

**Rodríguez, Elad. 2010.** CAMON. CAMON. [En línea] 16 de Febrero de 2010. [Citado el: 1 de abril de 2012.] <http://www.tucamon.es/contenido/ventajas-de-usar-frameworks-en-php>.

**Romero, Rodríguez Félix Ivan. 2010.** *Implementación del módulo de Redacción de la PLataforma de Televisión Informativa PRIMICIA.* Ciudad de la Habana : s.n., 2010.

**Rumbaugh, James, Jacobson, Ivar y Booch, Grady. 2010.** *El Proceso Unificado de Desarrollo de Software.* Madrid : s.n., 2010.

**SensioLabs. 2009.** *Doctrine ORM for PHP Doctrine 1.1.* s.l. : Autoedición, 2009. 2918390038.

**Sonaps. 2007.** *Sistema de Producción en Red. Sistema de Producción en Red.* 2007.

**Sony. 2011.** Sonaps. Sonaps. [En línea] 2011. <http://www.sony.fi/biz/lang/en/fi/product/networkprod/sonaps/overview>.

**Tedial. Tedial. Tedial.** [En línea] Tedial. [Citado el: 30 de Abril de 2012.] <http://www.tedial.com/en/about-us/introduction>.

—. Tedial. Tedial. [En línea] Tedial. [Citado el: 30 de Abril de 2012.] <http://www.tedial.com/en/products/tarsys/features>.

—. 2011. Tedial. Workflow optimisation and media asset management for the broadcast industry. *Tedial. Workflow optimisation and media asset management for the broadcast industry.* [En línea] Tedial, 2011. [Citado el: 30 de Abril de 2012.] <http://www.tedial.com/en/about-us/introduction>.

—. 2011. Tedial. Workflow optimisation and media asset management for the broadcast industry. *Tedial. Workflow optimisation and media asset management for the broadcast industry.* [En línea] Tedial, 2011. [Citado el: 30 de Abril de 2012.] <http://www.tedial.com/en/products/tarsys/features>.

**Televisión Digital. 2011.** [En línea] 2011. <http://www.televisiondigital.es/Satelite/Paginas/TVSatelite.aspx>.

**Torossi, Gustavo. 2008.** *El Proceso Unificado de Desarrollo de Software.* 2008.

**Universidad de las Ciencias Informáticas.** Teleformación. *Teleformación*. [En línea] Universidad de las Ciencias Informáticas. <http://teleformacion.uci.cu>.

**Universidad de Murcia.** Universidad de Murcia. *Universidad de Murcia*. [En línea] [Citado el: 18 de Noviembre de 2011.] <http://www.um.es/docencia/barzana/IAGP/lagp2.html>.

**Varela Ureña, Edith. 2007.** *Sistema web para el manejo de facturas y garantías de bienes institucionales*. Puebla de Zaragoza : BUAP, 2007.

**Visual-paradigm. 2012.** [En línea] 2012. <http://www.visual-paradigm.com>.

**Worsley, John C y Drake, Joshua D. 2002.** *Practical PostgreSQL*. Sebastopol : O'Reilly & Associates, 2002. 1565928466 9781565928466.

### **GLOSARIO DE TÉRMINOS**

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

**Corrector:** Usuario registrado que tiene permisos para redactar y corregir las noticias que posteriormente será transmitida. Es el actor que decide cuando la noticia esta lista para publicarla.

**Área de trabajo:** Es básicamente el área de la ventana que se usa para insertar, modificar y eliminar componentes, los que posteriormente formarán parte de la noticia.

**Pantalla:** Se utiliza para definir un intervalo de la noticia de tiempo de duración configurable, donde se insertan elementos ya sean imágenes, textos o videos.

**Sección temática:** Criterio que posibilita al redactor acumular las noticias de acuerdo a su contenido. Puede ser Nacionales, Internacionales, Deportivas entre otras.

**Propiedades:** Atributos de los componentes que pueden ser modificados de acuerdo al gusto del redactor. Estas características varían de acuerdo al componente seleccionado.

**Transición:** Acción que se emplea entre una pantalla y otra logrando un efecto visual agradable al usuario. Puede ser de entrada o de salida.

**Efectos:** Son acciones que se le aplican a los componentes de forma individual garantizando en cada noticia una mejor visualización.

## Anexo I. Descripción textual de los casos de uso del sistema

<b>Caso de Uso:</b>	<b>Administrar noticia.</b>
<b>Propósito:</b>	Manipular las noticias del sistema.
<b>Actores:</b>	Redactor.
<b>Resumen:</b>	El Caso de Uso se inicia cuando el redactor necesita listar, mostrar, buscar o eliminar una noticia del sistema. Para ello accede al menú principal y selecciona la opción Noticias.
<b>Precondiciones:</b>	El actor debe estar autenticado en la aplicación y tener los permisos de redacción.
<b>Pos condiciones:</b>	Se administran las noticias del sistema.
<b>Referencias:</b>	RF 1, RF 4, RF, RF 5 y RF 6.
<b>Prioridad</b>	Medio.
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. El redactor selecciona la opción Noticia del menú principal.	2. El sistema muestra una página donde aparece un listado con todas las noticias que contiene. El sistema muestra las opciones (Crear, Editar, Validar, Mostrar, Filtrar y Eliminar) para la gestión

	de las noticias.
3. El actor selecciona una opción.	<p>4. Si elige:</p> <ul style="list-style-type: none"> <li>a. Adicionar Noticia (A), ir al <b>Caso de Uso Crear noticia.</b></li> <li>b. Mostrar Noticia (B), ver sección “Mostrar noticia”.</li> <li>c. Editar Noticia (C), ir al <b>Caso de Uso Editar Noticia.</b></li> <li>d. Eliminar Noticia (D), ver sección “Eliminar noticia”.</li> <li>e. Buscar Noticia (E), ver sección “Filtrar noticia”.</li> <li>f. Validar Noticia (F) ver sección “Validar noticias”.</li> </ul>
<b>Sección Mostrar noticia.</b>	
	1. El sistema visualiza la noticia donde se logran ver los detalles de la misma y luego de finalizar la reproducción de la noticia muestra el listado de noticias.
<b>Flujo Alternativo de Eventos</b>	
1. El actor presiona el botón cerrar.	2. Ir a la <b>acción 2 del Flujo Normal de Eventos del Caso de Uso Redactar noticia.</b>
<b>Sección Eliminar noticia.</b>	

	1. El sistema verifica que la noticia no esté publicada y muestra un mensaje de confirmación para eliminar la noticia.
2. El redactor confirma que desea eliminar la noticia del sistema.	3. El sistema elimina la noticia de la base de datos y el XML quedando actualizado el listado de noticias.
<b>Flujo Alternativo de Eventos</b>	
	1. En caso de que la noticia esté publicada el sistema muestra un mensaje indicándole al actor que no puede eliminar la noticia.
2. El redactor cancela la confirmación de que desea eliminar la noticia.	3. Ir a la <b>acción 2 del Flujo Normal de Eventos del Caso de Uso Redactar noticia.</b>
<b>Sección Filtrar noticia.</b>	
1. El actor presiona el botón Buscar información.	2. El sistema muestra una ventana donde el actor podrá introducir los criterios de búsqueda.
3. El actor introduce los criterios de filtrado y presiona el botón Buscar.	4. El sistema muestra un listado de noticias que cumplen con los criterios establecidos por el actor.
<b>Flujo Alternativo de Eventos</b>	
3. El actor introduce los criterios de filtrado y presiona el botón Limpiar.	4. El sistema elimina los criterios establecidos por el actor para la búsqueda.
5. El actor presiona el botón Recargar.	6. El sistema muestra un listado con todas las noticias de la base de datos.



Tabla 4. Descripción textual del CU: Administrar noticia

<b>Caso de Uso:</b>	<b>Crear Noticia.</b>
<b>Propósito:</b>	Crear las noticias que contendrá el sistema.
<b>Actores:</b>	Redactor.
<b>Resumen:</b>	El Caso de Uso se inicia cuando el redactor desea agregar una noticia al sistema. Para esto el redactor presiona el botón Adicionar noticia.
<b>Precondiciones:</b>	El actor debe estar autenticado en la aplicación y tener los permisos de redacción.
<b>Pos condiciones:</b>	Se adicionan noticias al sistema.
<b>Referencias:</b>	RF 2.

<b>Prioridad</b>	Crítico.
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. El redactor presiona el botón Adicionar Noticia	2. El sistema muestra el área de trabajo.
3. El actor ingresa el título de la noticia.	4. El sistema muestra el título en el área de trabajo.
5. El actor selecciona el audio para la noticia.	6. El sistema muestra el título del audio seleccionado.
7. El actor selecciona la sección de la noticia.	8. El sistema muestra el logo y el nombre de la sección en el área de trabajo.
9. El actor presiona el botón Adicionar pantalla para agregar una pantalla.	10. Ir al <b>Caso de Uso Administrar pantalla.</b>
11. El redactor presiona el botón de guardar noticia.	12. El sistema guarda la noticia en la base de datos y en el XML y redirecciona al listado de noticias.
<b>Flujo Alterno de Eventos</b>	
	12. En caso que el redactor deje vacío el campo "Título" o ingrese un título no válido. El sistema muestra un mensaje diciendo "Debe ingresar un título para la noticia" u otro que diga que debe ingresar un título válido.



	12. En caso que el actor no seleccione un audio para la noticia. El sistema muestra un mensaje que dice que debe seleccionar un titulo para la noticia.
	12. En caso que el redactor no seleccione una sección para la noticia. El sistema muestra un mensaje diciendo que debe seleccionar una sección para la noticia.
11. En caso que el redactor presione el botón Cancelar.	12. El sistema cancela la creación de la noticia y redirecciona al listado de noticias.
11. El actor presiona el botón Previsualizar Noticia.	12. Ir a la <b>Sección Previsualizar noticia del Caso de Uso Crear Noticia.</b>
<b>Sección Previsualizar noticia</b>	
	1. El sistema visualiza la noticia donde se logran ver los detalles de la misma y luego muestra el área de trabajo.
<b>Flujo Alternativo de Eventos</b>	
2. El actor presiona el botón Cerrar durante la previsualización.	3. Se muestra el área de trabajo con las pantallas de la noticia que se está creando.

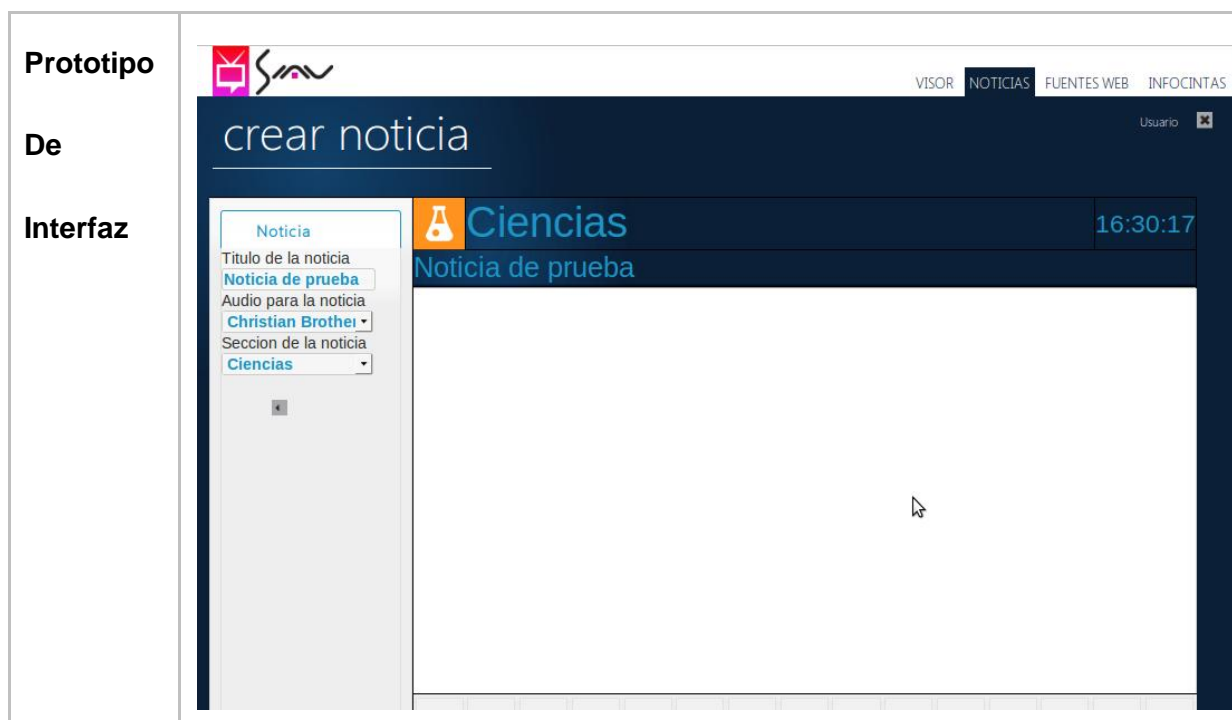


Tabla 5. Descripción textual del CU: Crear noticia

<b>Caso de Uso:</b>	<b>Administrar pantalla.</b>
<b>Propósito:</b>	Administrar las pantallas de las noticias del sistema.
<b>Actores:</b>	Redactor.
<b>Resumen:</b>	El Caso de Uso se inicia cuando el redactor necesita crear, eliminar, reajustar tiempo u ordenar las pantallas de una determinada noticia del sistema.
<b>Precondiciones:</b>	El actor debe estar autenticado en la aplicación y tener los permisos de redacción.

<b>Pos condiciones:</b>	Quedan actualizadas las pantallas de una noticia del sistema.
<b>Referencias:</b>	RF 16, RF 17, RF 18 y RF 19.
<b>Prioridad</b>	Crítico.
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	1. El sistema muestra el formulario para introducir el tiempo de duración de la pantalla en segundos.
2. El redactor introduce el tiempo de duración de la pantalla.	3. Se muestra el área de trabajo en blanco y se activan las opciones para insertar componentes (texto, imagen y video), el área de las propiedades de los componentes así como los botones “Previsualizar” y “Guardar”.
4. El actor hace clic en el botón Insertar del componente que desee (texto, imagen o video).	5. Ir al <b>Caso de Uso Administrar componente.</b>
6. El redactor puede realizar las siguientes acciones a los componentes insertados:	7. Si el redactor selecciona la opción: <ul style="list-style-type: none"> <li>a. Aplicar efectos de entrada o salida (A), ir al <b>Flujo Normal de Eventos del Caso de Uso Administrar Efecto.</b></li> <li>b. Eliminar efectos de entrada o salida (B), ir a la <b>Sección Eliminar efecto del Caso de Uso Administrar efecto.</b></li> <li>c. Ajustar propiedades (D), ir al <b>Caso de Uso</b></li> </ul>

	<p><b>Ajustar propiedades</b></p> <p>d. Eliminar componente (E), ir a la <b>Sección Eliminar componente del Caso de Uso Administrar componente.</b></p> <p>e. Redimensionar, de los componentes texto e imagen (F), ir a la <b>Sección Redimensionar del Caso de Uso Administrar componente.</b></p> <p>f. Editar, del componente texto (G), ir a la <b>Sección Editar texto del Caso de Uso Administrar componente.</b></p>
<p>8. El redactor a las pantallas creadas le puede realizar diferentes acciones.</p>	<p>9. Si el usuario elige la opción:</p> <p>a. Adicionar pantalla (A), ir a la <b>acción 1 del Flujo Normal de Eventos del Caso de Uso Administrar pantalla.</b></p> <p>b. Reajustar tiempo (B), ir a la <b>Sección Reajustar tiempo del Caso de Uso Administrar pantalla.</b></p> <p>c. Ordenar pantallas (C), ir a la <b>Sección Ordenar pantallas del Caso de Uso Administrar pantalla.</b></p> <p>d. Editar pantalla (D), ir a la sección <b>Editar pantalla del Caso de Uso Administrar pantalla.</b></p> <p>e. Aplicar transición de entrada o salida (E), ir al <b>Flujo Normal de Eventos del Caso de Uso Administrar transición.</b></p> <p>f. Eliminar transición de entrada o salida (F), ir a la <b>Sección Eliminar transición del Caso de Uso Administrar transición.</b></p> <p>g. Eliminar pantalla (G), ir a la <b>Sección</b></p>

	<b>Eliminar pantalla del Caso de Uso Administrar pantalla.</b>
<b>Sección Reajustar tiempo</b>	
	1. El sistema muestra un campo para introducir el nuevo tiempo de duración de la pantalla.
2. El actor introduce el nuevo tiempo de duración y presiona el botón Aceptar.	3. El sistema actualiza el tiempo de duración de la pantalla.
<b>Flujo Alternativo de Eventos</b>	
2. El actor presiona el botón Cancelar.	3. El sistema muestra el área de trabajo con las pantallas y los componentes de la noticia que se está redactando.
<b>Sección Eliminar pantalla</b>	
	1. El sistema muestra un mensaje de confirmación para eliminar la pantalla.
2. El redactor confirma que desea eliminar la pantalla.	3. El sistema elimina la pantalla de la noticia actualizando la región del listado de pantallas y el área de trabajo.
<b>Flujo Alternativo de Eventos</b>	
2. El redactor cancela la confirmación de que desea eliminar la pantalla.	3. El sistema muestra el área de trabajo con las pantallas y los componentes de la noticia que se está redactando.
<b>Sección Ordenar pantallas</b>	

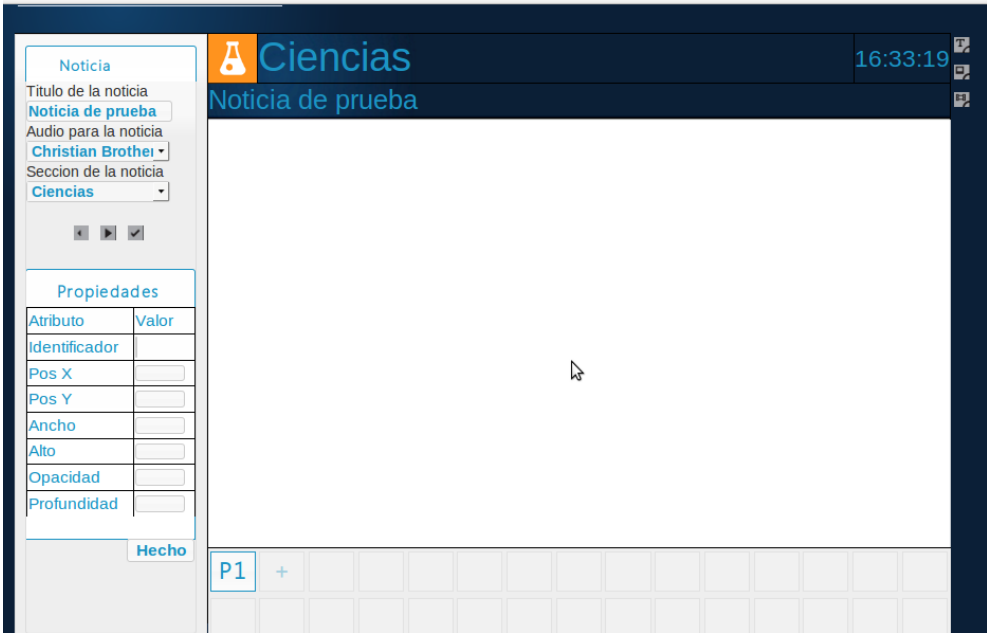
<p>1. El redactor selecciona la pantalla que desea ordenar e indica el lugar para donde desea moverla.</p>	<p>2. El sistema actualiza el orden las pantallas visualizándolas en el listado.</p>
<p><b>Sección Editar pantalla</b></p>	
<p>1. El redactor selecciona la pantalla que desea editar.</p>	<p>2. El sistema muestra en el área de trabajo la pantalla seleccionada con los componentes que contiene.</p>
<p>3. El actor edita la pantalla realizando las modificaciones que sean necesarias</p>	<p>4. El sistema actualiza el estado de la pantalla así como el de los componentes de dicha pantalla</p>
<p><b>Prototipo</b> <b>De</b> <b>Interfaz</b></p>	

Tabla 6. Descripción textual del CU: Administrar pantalla

## Anexo II. Diagramas de clases del diseño

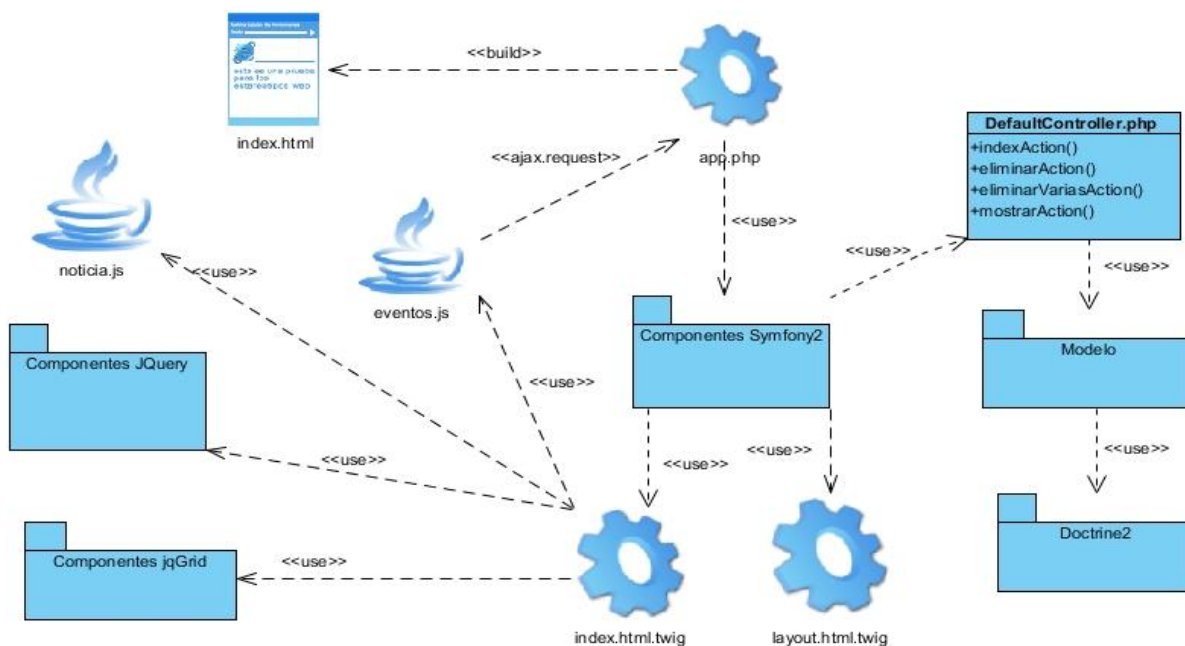


Figura 11. Diagrama de clases del diseño del CU: Administrar noticia

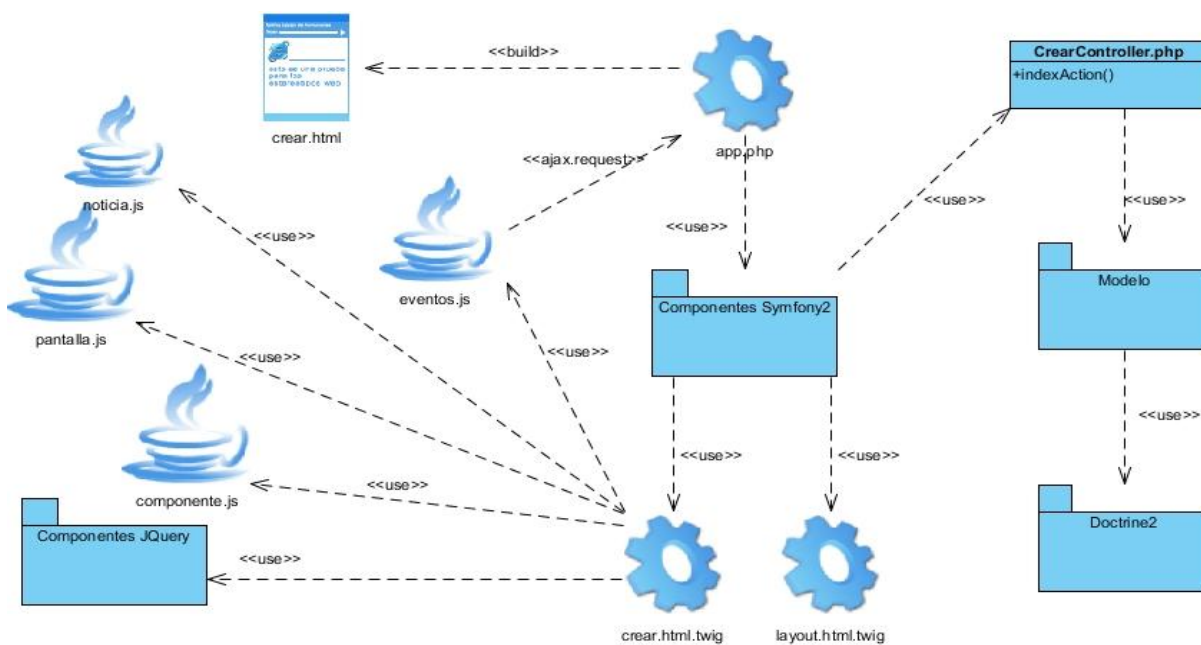


Figura 12. Diagrama de clases del diseño del CU: Crear noticia

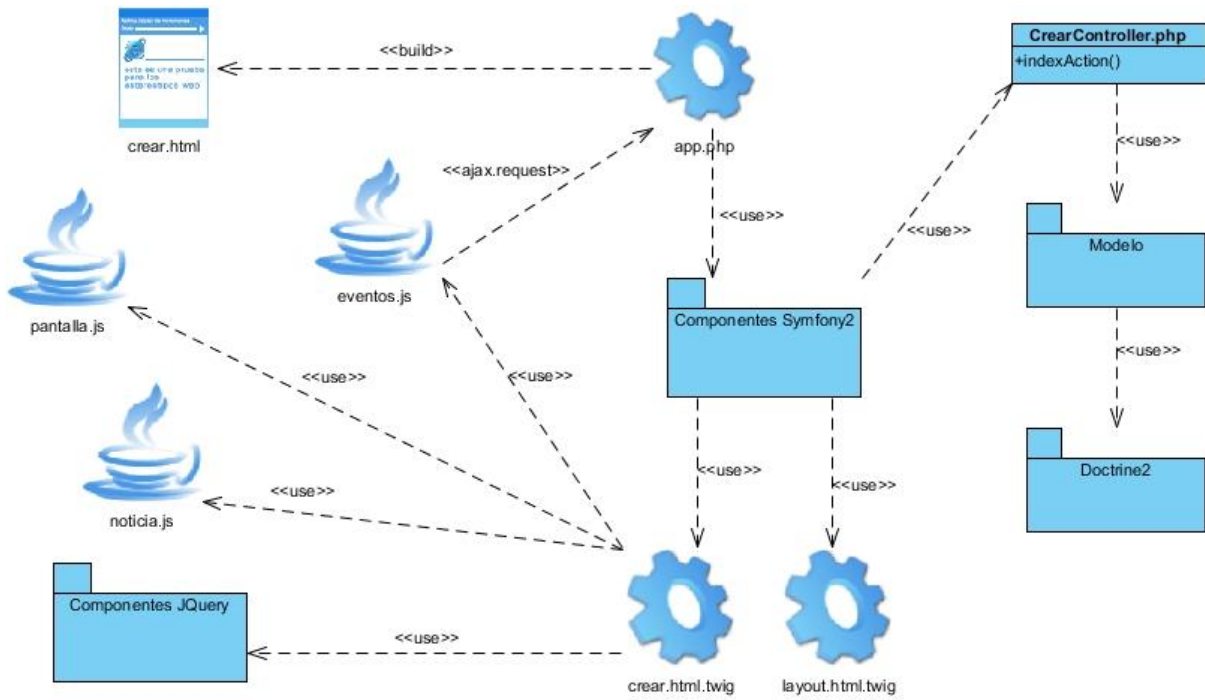


Figura 13. Diagrama de clases del diseño del CU: Administrar pantalla