



UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

Facultad 6

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

Título: Subsistema de Administración para el canal Señal ACN.

Autor: Orlando Maresma Londres.

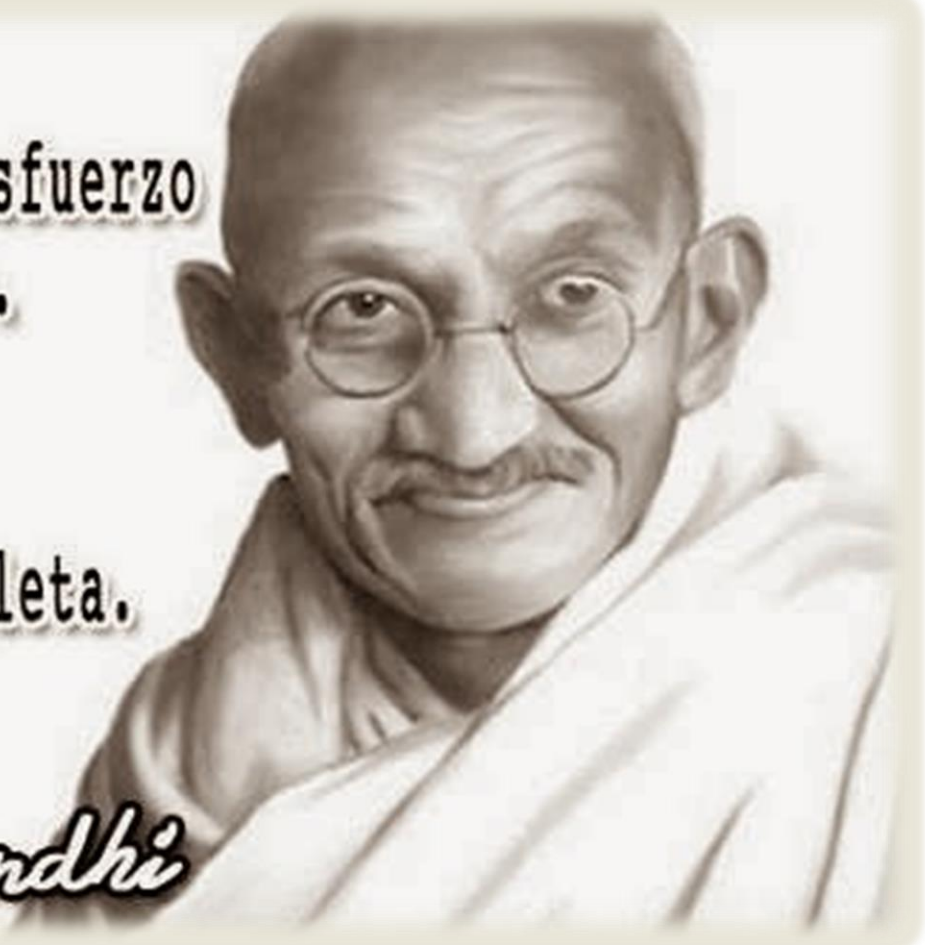
**Tutor (es): Ing. Felix Ivan Romero Rodríguez.
Ing. Mairena Arzuaga Limonta.**

La Habana, Julio, 2015.

Nuestra recompensa
se encuentra en el esfuerzo
y no en el resultado.

Un esfuerzo total
es una victoria completa.

Mahatma Gandhi



AGRADECIMIENTOS

Agradezco principalmente a mi familia los Maresma-Londres en Baracoa a mi madre Estela, mi padre Orlando, mi hermano Osmani por todo el apoyo que me brindaron durante todos estos años de universidad que confiaron en mí a pesar de todo, es para ellos esta tesis. A mi abuelo que a pesar de fallecer hace dos semanas siempre se preocupó por mí, gracias abu siempre te recordaré. Para mi segunda familia los Brooks de Guantánamo en especial a mi hermano Rafael que a pesar de no encontrarse entre nosotros siempre me apoyó en todo, ha sido la fuente de mi inspiración por todo lo que luchó contra su cáncer y logró el objetivo que siempre tuvo en mente que fue regalarle a su madre un título de ingeniero, es para ti mi hermano del alma este título donde quiera que estés, siempre estarás en mi corazón y espíritu. Para Soraya gracias por los tirones de orejas que me diste y por el apoyo durante todo este tiempo, sé que tu corazón sufre por la pérdida de tu hijo pero recuerda que aún te queda uno que no te olvida y siempre estará a tu lado.

A todas las personas que de una forma u otra siempre estuvieron ahí e hicieron de mí una mejor persona, no menciono nombre para que no se me quede ninguno por lo que mil gracias por su amistad. A los kneks el mejor piquete que ha pasado en la historia de la UCI y que a pesar de ser de diferentes lugares nunca nadie olvidará ese piquete, aquellos que son una leyenda viviente llamados los kneks.

A mis tutores por todo el apoyo y la paciencia que me brindaron, sé que les hice pasar un mal rato por lo que me disculpo y les agradezco de todo corazón. A los dirigentes de la facultad que me permitieron defender mi tesis y poder graduarme.

De forma general gracias a todas las personas que de una forma u otra son responsables de que hoy en días pueda gritar a los cuatro vientos!!!!!!SOY INGENIERO!!!!!!.

DECLARACIÓN DE AUTORÍA

Declaro por este medio que yo, Orlando Maresma Londres, soy el autor de este trabajo y que autorizo a la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste, firmamos la presente declaración de autoría en La Habana a los ____ días del mes de _____ del año _____.

Orlando Maresma Londres

Autor

Ing. Felix Ivan Romero Rodríguez

Tutor

Ing. Mairena Arzuaga Limonta.

Tutor

RESUMEN

La Plataforma de Televisión Informativa (PRIMICIA) es una plataforma de televisión desarrollada en Cuba que se basa en la combinación de tecnologías, para dar como resultado la creación de noticias e informaciones que posteriormente serán difundidas por canales televisivos. Cuenta con un Subsistema de Administración compuesto por una variedad de módulos que en su conjunto posibilitan su funcionamiento como un todo, garantizando así la gestión del canal. Sin embargo el Subsistema de Administración no constituye una solución recomendable para algunos escenarios.

El objetivo del presente trabajo es desarrollar un Subsistema de Administración para el canal Señal ACN que se adapte a las necesidades de la Agencia Información Nacional. Se encuentra estructurado en tres capítulos que recogen las diferentes etapas de desarrollo del sistema, las cuales van desde la fundamentación teórica de la investigación a través de la propuesta y el modelado de la solución hasta llegar a la implementación y validación de la aplicación. Se presentan los resultados de la investigación realizada para la elaboración del prototipo funcional, demostrando así la necesidad de desarrollar dicho subsistema de administración para el canal Señal ACN.

Palabras Clave:

Administración, PRIMICIA, subsistema.

ABSTRACT

The Informative Television Platform (PRIMICIA) is a television platforms based on the combination of technologies to give as a result the creation of news and information that will later be broadcast by television channels. It has a Management Subsystem consists of a variety of modules which together enable the operation thereof as a whole ensuring channel management. However, it's not a recommended solution for some scenarios.

The aim of this study is to develop a management subsystem for the ACN Signal channel that suits the needs of the National Information Agency. The same is divided into three chapters that reflect the different stages of system development, ranging from the theoretical foundation of the research through the proposal and modeling of the solution up to the implementation and validation of the application. The results of the research for the development of functional prototype are presented, demonstrating the need to development a management subsystem for ACN signal channel.

Keywords: management, PRIMICIA, subsystem.

ÍNDICE

INTRODUCCIÓN.....	9
CAPÍTULO 1: Fundamentación Teórica.....	13
1.1 Conceptos asociados al dominio del problema.....	13
1.2 Sistema de Producción de Noticias.....	13
1.3 Análisis de Soluciones Existentes.....	17
1.4 Metodología de Desarrollo.....	18
1.5 Lenguaje Unificado de Modelado.....	19
1.6 Herramienta Case.....	20
1.7 Sistema de gestión de Base de Datos.....	20
1.8 Marcos de trabajo.....	21
1.9 Tecnologías de desarrollo.....	22
1.10 Lenguaje de Programación.....	22
1.11 Entorno de Desarrollo Integrado.....	23
1.12 Servidor de aplicaciones web.....	24
1.13 Conclusiones del capítulo.....	24
CAPÍTULO 2: Características del sistema.....	25
2.1 Modelo de Negocio.....	25
2.1.1 Actores del negocio.....	25
2.1.2 Trabajadores del negocio.....	25
2.1.1 Modelo de Casos de Uso del Negocio.....	26
2.1.2 Descripción de Caso de Uso del Negocio.....	26
Tabla 1 Descripción del Caso de Uso. Realizar Noticia.....	26
2.2 Requisitos funcionales.....	28
2.3 Requisitos no funcionales.....	29
2.3.1 Diagramas de Casos de Uso.....	30
2.3.2 Descripción de los Casos de Usos.....	31

2.4	Patrones	34
2.4.1	Patrones de casos de uso.....	35
2.4.2	Patrones de arquitectura.....	35
2.4.3	Diagrama de Clase del Diseño.	37
2.4.4	Patrones de diseño.....	38
2.5	Modelo de Datos.....	43
2.5.1	Descripción del Modelo de Datos.....	43
2.6	Conclusiones del capítulo.	44
CAPÍTULO 3 Implementación y prueba.		45
3.1	Modelo de despliegue.....	45
3.2	Modelo de Implementación.	45
3.3	Código fuente.	46
3.3.1	Estándares de codificación.	47
3.3.2	Ejemplos de código fuente.....	48
3.4	Estrategia de prueba de software.	48
3.5	Conclusiones parciales.	53
CONCLUSIONES GENERALES.....		54
RECOMENDACIONES.		55
REFERENCIAS BIBLIOGRÁFICAS.		56
GLOSARIO DE TÉRMINOS.....		59

INTRODUCCIÓN.

El uso de las Tecnologías de la Información y las Comunicaciones (TIC's) trae consigo un desarrollo acelerado en cuanto a informática, internet y comunicaciones se refiere. Las TIC's han proporcionado un cambio positivo en la sociedad brindando resultados rápidos, eficientes y de mayor calidad, conformando un conjunto de recursos necesarios para el manejo de la información. El avance tecnológico proporciona a la sociedad herramientas que brindan una mejor comunicación, facilidad de asociación, reducción del tiempo empleado en el desarrollo de actividades e incluso mejoran la propia tecnología. Uno de los medios de comunicación más usados es la televisión y es propenso a constantes actualizaciones que van desde sus procesos, contenidos y hasta las herramientas que se utilizan para llevar la información hacia los telerreceptores.

La integración entre la televisión e internet ha hecho posible la transmisión de materiales audiovisuales de mayor información y calidad. Es un hecho que la divulgación de la noticia se ha convertido en una industria digital debido a que favorece su difusión. Para la producción de noticias en la industria digital se emplean aplicaciones informáticas que administran, conforman y transmiten informaciones llamadas Sistemas de Producción de Noticias.

En Cuba existe un Sistema Productor de Noticias llamado Señal ACN que se encuentra en la Agencia de Información Nacional (AIN). Señal ACN es un canal de teletexto desarrollado y dedicado a transmitir informaciones en diferentes formatos a colaboradores cubanos y para los habitantes de las zonas de silencio de la geografía cubana. Dicho sistema fue desarrollado por la Universidad de las Ciencias Informáticas (UCI) como resultado de la generalización de un conjunto de soluciones informáticas que se personalizaron para la AIN, a la cual le interesó poseer un canal de teletexto para su centro o televisora.

Entre los principales antecedentes del canal Señal ACN se encuentran Señal 3, TV Energía y PRIMICIA. Señal 3 era un sistema informativo creado para la red de televisión interna de la UCI que permitía mantener informada la comunidad universitaria. TV Energía es un sistema personalizado creado con el propósito de mantener informados a trabajadores y visitantes de la sede central del Ministerio del Poder Popular para la Energía y Petróleo de Venezuela (MENPET). PRIMICIA es una aplicación informática creada en el año 2005 por la UCI, desarrollada y soportada completamente en software libre. Se encuentra estructurada en dos subsistemas que se relacionan entre sí y actúan como un todo para brindar un resultado final eficiente y acorde a las necesidades de los usuarios. En el Subsistema de Administración se

realiza la administración del canal y toda la gestión de las informaciones y recursos multimedia que se transmiten y el Subsistema de Transmisión es el encargado de la visualización de las noticias y materiales publicados.

La Agencia de Información Nacional (AIN) es una institución que ofrece información actualizada sobre lo que acontece en el país a partir de una red nacional de corresponsales en los campos económico, político, social, cultural, científico, tecnológico y deportivo. Tiene como principal objetivo la venta de noticias a sus colegas nacionales e internacionales, brindando información por diferentes medios como televisión y radio, cuenta también con un banco de fotografías para la comercialización.

La AIN ha sido uno de los clientes que más ha utilizado las potencialidades de las versiones desarrolladas del producto PRIMICIA. Sin embargo han surgido nuevas necesidades que Señal ACN (versión de PRIMICIA que actualmente se encuentra desplegada en la institución) no satisface en su totalidad. Por ejemplo: en la edición o creación de la noticia no se puede subir un nuevo archivo multimedia, lo que hace al proceso más engorroso para el usuario del sistema. Los procesos de publicación y de gestión de medias son excluyentes lo que provoca que cuando se publique no se puedan modificar las medias. Actualmente las noticias se archivan con el orden de creación y no se puede definir un orden de prioridad entre ellas para la visualización.

También es necesario obtener una bitácora de todos los procesos por los que pasa la noticia y por quiénes son manejados dichos procesos, así como poder establecer prioridad de publicación. Además, los reportes que se generan actualmente son los de roles y los de música, no obteniendo reportes relacionados a las noticias y a los otros procesos que se realizan en la aplicación. Otro problema es que los archivos de mayor antigüedad no son borrados en su totalidad lo que genera una acumulación de archivos en el servidor pudiendo ocasionar un colapso en el sistema. También es necesaria una migración tecnológica a nuevas herramientas que mejore el funcionamiento del sistema. Por otro lado la gestión instantánea de los contenidos audiovisuales no pasa por el proceso de gestión de medias, ni por la catalogación.

Con el propósito de convertir Señal ACN en un producto que pueda ser adecuado a las necesidades de la AIN en la actualidad y teniendo como antecedente trabajos de análisis y diseño anteriores elaborados en el proyecto de desarrollo de la plataforma, se ha definido el siguiente **problema a resolver**: los procesos del subsistema de administración del canal Señal ACN no se adaptan a las nuevas necesidades presentadas por los usuarios del sistema.

Para darle solución a dicho problema se plantea como **objetivo general**: desarrollar una personalización del subsistema de administración del canal Señal ACN. Definiéndose como **objeto de estudio**: los procesos de administración en sistemas informáticos de producción de noticias. Enmarcando en el **campo de acción**: los procesos de administración de noticias en Señal ACN.

Con el fin de obtener una guía para el desarrollo de la investigación se definen las siguientes preguntas de la investigación:

¿Qué procesos intervienen en el subsistema de administración de señal ACN?

¿Qué herramientas y tecnologías son factibles para el desarrollo del subsistema de administración?

¿Qué técnicas de prueba de software verifican el correcto funcionamiento del subsistema?

Como guía para dar cumplimiento al objetivo propuesto, se definieron las siguientes tareas de la investigación:

1. Analizar los Sistemas Productores de Noticias.
2. Definir y caracterizar las herramientas y metodología a utilizar durante el desarrollo del subsistema.
3. Definir los requisitos funcionales y no funcionales del subsistema de administración.
4. Modelar el subsistema de administración para Señal ACN.
5. Implementar el subsistema de administración para Señal ACN.
6. Realizar pruebas a la solución propuesta.

En el transcurso de la investigación se utilizaron varios métodos científicos, clasificados en teóricos y empíricos. Entre los métodos teóricos empleados se encuentran:

- **Analítico – Sintético**: Brinda la posibilidad de buscar la esencia de los fenómenos, los rasgos que caracterizan y distinguen los procesos relacionados con la administración de PRIMICIA v1.7. Dentro de la investigación ha permitido la extracción de los elementos más importantes que se relacionan con la plataforma.
- **Inductivo – Deductivo**: Permitió llegar a un grupo de conocimientos generalizadores acerca de los posibles elementos de administración en la Plataforma, así como determinar aquellos procesos más específicos implicados en los nuevos cambios que se necesitan para una correcta administración.

Se ha emplearon como métodos empíricos:

- **Observación**: Se utiliza este método con la finalidad de caracterizar y analizar detalladamente cómo funciona actualmente el subsistema de administración, identificando los mecanismos para desarrollar un producto acorde con las necesidades del cliente.

La presente investigación incluye los siguientes capítulos:

Capítulo 1. Fundamentación Teórica: se enuncian conceptos que posibilitan un adecuado entendimiento de la situación problemática y el marco teórico en sentido general, se realiza una descripción del objeto de estudio de la presente investigación y se definen los elementos teóricos que la sustentan. Se exponen y argumentan otras aplicaciones que sirven de guía para dar solución al problema al resolver del presente trabajo. Por último se seleccionan y describen la metodología de desarrollo, lenguajes, herramientas y tecnologías utilizadas para el desarrollo del producto.

Capítulo 2. Características del sistema: se realiza el modelo de negocio necesario para conocer la estructura y los roles de las personas dentro de la AIN, se seleccionan los requerimientos funcionales y no funcionales del sistema, posteriormente se describen los casos de uso que forman parte del mismo y se definen los actores que interactúan con la aplicación. Por último se describen los patrones de arquitectura y los patrones de diseño utilizados así como el modelo de datos.

Capítulo 3. Implementación y prueba: se abordan aspectos referentes a la implementación del sistema, como los diagramas de componente y de despliegue. Además, de exponer los resultados arrojados por las pruebas realizadas para comprobar el correcto funcionamiento del sistema y se definen los estándar de codificación que se utilizarán.

CAPÍTULO 1: Fundamentación Teórica.

El presente capítulo contiene los principales aspectos que conforman la fundamentación teórica de la investigación. Se analizan algunas plataformas similares para obtener una perspectiva de la utilización de funcionalidades afines a las que se proponen en la investigación. Por último se caracterizan la metodología de desarrollo de software y las herramientas que se van a utilizar en el desarrollo del sistema.

1.1 Conceptos asociados al dominio del problema.

En este acápite se relacionan los conceptos que desde el punto de vista teórico permiten un mejor entendimiento de lo planteado en la situación problemática o el marco del problema en sentido general.

Subsistema: Cada uno de los componentes principales de un sistema se llama subsistema, cada subsistema abarca aspectos del sistema que comparten alguna propiedad común. Es un conjunto de elementos ordenados o funciones relacionados para cumplir con un propósito o fin determinado y cuyas partes deben reunir ciertas condiciones de tal manera que se complementen formando el sistema (1).

Catalogación: Según la Real Academia Española catalogación no es más que la acción y efecto de catalogar. Catalogar es clasificar o encasillar dentro de una clase o grupo. Específicamente para el sistema sería la clasificación de las medias en temporal e intemporal. La catalogación temporal no es más que el archivo o media catalogada como temporal, una vez utilizado por el sistema es eliminado automáticamente del servidor. La catalogación intemporal es utilizada cuando la media es almacenada en el servidor y no es eliminada.

Infocintas: Cintillo informativo que se muestra en las pantallas de tipo texto o texto-imagen promocionando eventos de importancia para la institución, noticias relevantes que se transmitirán o información en general para los televidentes (2).

1.2 Sistema de Producción de Noticias.

Se define como Sistema de Producción de Noticias (SPN) un sistema integrado de recepción, almacenamiento, composición, pre-edición, supervisión, edición y postproducción de un conjunto de señales constituidas por textos, videos, imágenes, sonidos y voz en "off".

Las principales características de un SPN son:

- Registro automático en servidores de las señales que se reciben en el centro, bien sean textos de agencia, imágenes, videos o sonidos de cualquier origen.
- Almacenamiento en dos tipos de servidores: servidor de textos para recepción de agencias y servidor central de video y audio de alta calidad para imágenes y sonidos recibidos del exterior o volcados internamente y almacenamiento de las noticias montadas listas para emitir.
- Catalogación, indexación y etiquetado en una base de datos con el mayor número posible de niveles de búsqueda por filtros, palabras claves de todas las informaciones recibidas.
- Integración en un único terminal de periodista con las siguientes características:
 1. Búsqueda y acceso a los textos almacenados en el servidor por medio del sistema de recepción de agencias.
 2. Composición, generación y edición de textos con teclado, ratón, pantalla táctil, lápiz óptico, etc.
 3. Búsqueda y acceso a las imágenes y sonidos almacenados en el servidor con calidad de baja resolución (servidor réplica) pero suficiente para seleccionar con facilidad y sin dudas las informaciones que se deseen.
 4. Edición sencilla de las imágenes y sonidos seleccionados con capacidades de mezcla, cortinas, incrustaciones, generación de caracteres, etc.
 5. Composición y edición de los textos, las imágenes y los sonidos dentro de la misma aplicación trabajando en la misma pantalla, es decir gestión integrada de noticias.
 6. Visualización de noticias redactadas.
 7. Conexión con unidades de edición avanzada.
- Capacidad de Edición Avanzada, con acceso al servidor central y facilidades de mezcla y efectos de audio y video, que acepten listados de edición a través de red.
- Emisión de las noticias montadas desde el servidor central o desde un servidor dedicado de emisión con envío directo al control de emisión de informativos en su mezclador de salida.
- Existencia de una estación de supervisión y gestión. Desde esta estación de supervisión y gestión se puede tomar la decisión final sobre la noticia con capacidad de borrado, alteración y envío a emisión y a otras redes (3).

Estos sistemas cuentan con procesos de administración con el objetivo de lograr el correcto funcionamiento de sus actividades y de esta forma obtener los resultados esperados. Los procesos hacen referencia al conjunto de pasos o etapas necesarias para llevar a cabo una actividad. La administración por su parte es el proceso de planificar, organizar, dirigir y controlar las actividades con el propósito de alcanzar las metas establecidas para la organización. Los

procesos de administración se encargan de gestionar las configuraciones del sistema y de los usuarios que acceden al mismo.

En Cuba existe el canal Señal ACN (versión de PRIMICIA) que se encuentra desplegado en la AIN y PRIMICIA v2.0 que se está desarrollando en la UCI. Ambos sistemas están compuesto por dos subsistemas, el subsistema de Transmisión y el subsistema de Administración.

Procesos de administración en Señal ACN.

El Subsistema de Administración que posee el canal Señal ACN es una aplicación web desarrollada en Symfony v1.0.5 que permite la gestión de las noticias y recursos multimedia según los requerimientos de la AIN. Propone un flujo de trabajo donde se definen los operadores del canal y su rol dentro del sistema, los cuales pueden realizar las siguientes acciones según su nivel de acceso:

- redactar noticias con toda la información asociada (título, cantidad de pantallas y música de fondo).
- corregir las noticias que han sido introducidas al sistema pudiendo modificar cada campo de la noticia.

Además se gestiona la programación del canal publicando las noticias según la fecha y el horario deseado, se especifican las secciones que se mostrarán en el canal; así como todo el banco de temas musicales que servirán de fondo al canal. El proceso de redacción de Señal ACN se realiza de la siguiente forma:

- Se obtiene y sintetiza la información procedente de los principales medios de prensa (agencias de noticias, agencias cablegráficas y prensa escrita).
- Se selecciona un título para la noticia.
- Se ubica en la sección temática adecuada (Actualidad, Nacionales, Deportes y Culturales).
- Se escoge un fondo musical.
- 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 (10, 15 o 25 segundos).
- La conformación de la noticia termina cuando el corrector la revisa 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.

En Señal ACN es permisible el acceso, actualización y presentación de recursos multimedia en las noticias, sin embargo se niega la posibilidad de configurar los componentes existentes para su redacción (4).

PRIMICIA es una aplicación informática, creado en el año 2005 por la UCI desarrollado y soportado completamente en software libre. Se encuentra estructurado en dos subsistemas que se relacionan entre sí y actúan como un todo para brindar un resultado final eficiente y acorde a las necesidades de los usuarios. En el subsistema de administración se realiza la administración del canal y toda la gestión de las informaciones y recursos multimedia que se transmiten y el subsistema de transmisión es el encargado de la visualización de las noticias y materiales publicados.

Su arquitectura física puede ser adaptada en dependencia del tipo de red de televisión que se desee utilizar para la transmisión del canal y las condiciones del entorno donde se instale, las que pueden ser más sofisticadas o en algún caso estar restringidos por un bajo presupuesto. El canal informativo que soporta la plataforma muestra de forma automática ciclos de noticias constantes y repetitivas, condicionadas por las informaciones publicadas para determinados períodos de tiempo, logrando integrar en sus transmisiones textos combinados con imágenes y video.

Durante las emisiones permite la reproducción de un fondo musical que puede ser personalizado según la noticia que se muestra, además es posible la utilización de cintillos informativos o infocintas que permitan el adelanto de breves asuntos de carácter relevante o promocional. Como elemento adicional es posible mostrar la fecha y hora, tiempo restante de la noticia o pantalla y titular de la próxima. Es importante señalar que el desarrollo de PRIMICIA se ha enfocado hacia la obtención de funcionalidades genéricas fácilmente escalables que no dependan de un entorno dado, ni de un diseño gráfico específico, pudiendo acordar con los clientes la personalización o eliminación específica de alguna funcionalidad (2).

Los procesos de administración son diferentes con respecto a los de Señal ACN analizados posteriormente. A continuación se realiza una breve descripción de estos procesos.

Procesos de administración en PRIMICIA v1.7.

El Subsistema de Administración que posee PRIMICIA v1.7 está compuesto por una variedad de módulos que en su conjunto posibilitan su funcionamiento como un todo. Permite administrar los usuarios de la plataforma, la transmisión del canal, los recursos multimedia para la confección de las noticias, además de llevar a cabo los procesos de redacción y publicación. El módulo de Redacción de Noticias es el que contiene mayor responsabilidad en el proceso de redacción y está orientado a llevar a cabo la redacción y edición de las informaciones conjuntamente con los recursos multimedia que la conforman.

El Subsistema de Administración tiene las siguientes prestaciones generales:

- **Gestión de los usuarios del sistema:** permite adicionar y eliminar usuarios, así como establecer las responsabilidades de cada usuario en el sistema.
- **Gestionar las secciones temáticas del canal:** permite establecer el orden de las secciones, horario en que serán mostradas y habilitarlas o deshabilitarlas.
- **Funcionalidades para la redacción de noticias:** permite según los formatos definidos para las pantallas; la publicación de las noticias teniendo en cuenta fecha de inicio y fin de la publicación. Posibilita la gestión de las noticias del canal permitiendo modificar, eliminar y archivar las noticias; además de la administración del archivo de noticias del canal permitiendo reutilizarlas y eliminarlas.
- **Gestionar recursos multimedia:** posibilita el almacenamiento, administración y reproducción de recursos multimedia como imágenes, música y video.
- **Funcionalidades para la creación y administración de cintillos informativos o infocintas:** la administración de los cintillos establece el orden de prioridad de muestra y la habilitación o deshabilitación de los mismos.
- **Generación de reportes sobre la actividad del sistema:** los reportes se realizarán sobre la actividad de los trabajadores del sistema, permite realizar búsquedas de noticias publicadas atendiendo a distintos criterios como: fecha de publicación, temática, palabras clave y título. Ofrece facilidades para la impresión de los reportes y la exportación de los reportes a formato digital.
- **Administración de la señal del canal:** lo cual permite cambiar entre la señal del canal y la televisión en vivo de un canal externo (5).

El análisis de los procesos de administración en Señal ACN y PRIMICIA v1.7 permitió definir que el subsistema de administración que mejor se adapta al SPN que se desea desarrollar es el de PRIMICIA v1.7.

1.3 Análisis de Soluciones Existentes.

En la actualidad existen varios SPN por lo que su estudio ayudan a comprender el funcionamiento de la administración. Entre los SPN existentes se encuentra:

VSN.

Es una compañía de alta tecnología dedicada al desarrollo de software y fabricación de sistemas complejos de alta fiabilidad y rendimiento para la gestión de contenidos multimedia. VSN ofrece más de 50 módulos de software desarrollados y fabricados sobre una arquitectura abierta. Basada en estándares de la industria y pensada para permitir la máxima personalización del flujo de trabajo de un canal de TV, la plataforma VSN facilita operaciones

con entornos de trabajo remotos y distribución multiplataforma de contenidos y una perfecta integración entre todas las áreas y con tecnología de terceros fabricantes. Las soluciones de VSN garantizan la escalabilidad del sistema, la rentabilidad de la inversión realizada y la consecución de flujos de trabajos robustos, completos e integrados y permiten desplegar un sistema completo de automatización en la producción y emisión de noticias y directos (6).

ENEWSROOM.

Es el sistema de producción de noticias y contenidos audiovisuales de la plataforma propietaria *Estructure Media Systems*. Integra las herramientas necesarias para la explotación de noticias y la introducción de flujos de trabajo más flexibles que permiten a los usuarios realizar su trabajo con grandes facilidades. Dentro de sus características fundamentales se encuentra que es un sistema sencillo, intuitivo, robusto y fiable. También posee servidores multiformato que son capaces de digitalizar simultáneamente hasta tres calidades de vídeo. La base del sistema se conecta con otras herramientas externas 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 (7).

En general el análisis de las soluciones existentes provee una visión de cómo se maneja la noticia en el mundo mediante los SPN. La VSN y el Enewrsoom son sistemas privativos que no se adaptan a las necesidades específicas de AIN. Además de todas las soluciones existentes la que se enfoca a la investigación es Señal ACN que está desplegado en la AIN y es a la cual se le va a realizar una actualización en cuanto a funcionalidades y rendimiento se refiere.

1.4 Metodología de Desarrollo.

El Proceso Unificado de Rational (RUP, por sus siglas del inglés *Rational Unified Process*) es un proceso de Ingeniería de Software. Provee un acercamiento disciplinado a la asignación de tareas y responsabilidades dentro de una organización de desarrollo. Su meta es asegurar la producción de software de alta calidad que satisfaga la necesidad de sus usuarios finales, dentro de límites de tiempo y presupuesto predecibles. Esta metodología, más allá de enfocarse en la producción de grandes cantidades de documentos, enfatiza en crear y mantener modelos que son representaciones, lógicamente ricas, del sistema de software en desarrollo.

En ella se efectúan un conjunto de actividades que transforman los requisitos de un usuario o cliente en un sistema de software. Una de las características por las que es ampliamente aceptada es por su empleo del Lenguaje Unificado de Modelado (UML por sus siglas en inglés)

para la representación de sus artefactos. Además posee una amplia documentación tanto en idioma inglés como en español. El ciclo de vida de RUP se identifica por tres características fundamentales; guiado por casos de uso, centrado en la arquitectura, iterativo e incremental, está dividido en 4 fases durante las cuales se transita por 9 flujos de trabajo.

Por las características que ha presentado desde sus inicios el proyecto PRIMICIA v1.7 tales como su envergadura, las condiciones legales de su concepción, el nivel de documentación requerido y las posibilidades de continuar su mejora en el futuro, la metodología RUP se presenta como la más indicada para la implementación del Subsistema de Administración de la plataforma (8). RUP no solo cubre las necesidades presentadas por las características antes mencionadas, sino que de manera general sustenta el desarrollo del proyecto a través de un efectivo sistema de roles y responsabilidades bien distribuidas, garantizando que se generen todos los artefactos que permitirán la continuidad del trabajo en el proyecto.

Además, habiendo sido la metodología empleada desde la creación de Señal ACN resulta muy conveniente realizar los siguientes pasos en la implementación del subsistema bajo su cobertura. El hecho de que los líderes y el equipo de desarrollo estén familiarizados con el modo de trabajo de RUP, por su empleo desde los inicios del proyecto, apoya la decisión de utilizarla para su aplicación en la presente investigación.

1.5 Lenguaje Unificado de Modelado.

El Lenguaje Unificado de Modelado (UML, por sus siglas del inglés *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. 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.

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 (9).

Es un lenguaje libre y totalmente independiente del lenguaje de programación que se utilice, tiene como fin modelar sistemas en base al paradigma orientado a objetos, por tanto los diseños realizados usando UML se pueden implementar en cualquier lenguaje orientado a objetos. Para el desarrollo de las nuevas funcionalidades de Señal ACN UML brinda la

posibilidad de representar gráficamente los modelos de forma que puedan ser comprendidos. Permite especificar las características que tendrá el sistema, incluso antes de su construcción. A partir de los modelos que se vayan generando se puede construir el sistema diseñado y estos propios elementos gráficos sirven como documentación del sistema desarrollado para futuras revisiones y versiones.

1.6 Herramienta Case.

Las herramientas de ingeniería de software asistida por computadoras (CASE, por sus siglas del inglés *Computer Assisted Software Engineer*) son aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software mediante la realización del diseño del proyecto, el cálculo de los costos, la implementación de parte del código automáticamente, la documentación y detección de errores. Sin embargo para la primera versión de PRIMICIA V1.7 todos los artefactos se generaron utilizando Visual Paradigm. El Visual Paradigm es una herramienta multiplataforma, es muy fácil de usar pues presenta un ambiente gráfico agradable e intuitivo para el usuario.

Es muy poderosa la misma permite la realización de ingeniería directa e inversa, el modelado de procesos de negocio, de requisitos, de base de datos y la generación de código PHP. Admite la integración con varios entornos de desarrollo (IDEs) (10). Soporta múltiples usuarios trabajando sobre el mismo proyecto y a la vez permite realizar el control de versiones, característica que tiene un impacto positivo para diversos proyectos.

El uso de esta herramienta en el desarrollo de las funcionalidades propuestas en la presente investigación implica además un ahorro de tiempo en capacitación, pues fue la herramienta CASE utilizada por el equipo de trabajo en el desarrollo de la primera versión de la plataforma.

1.7 Sistema de gestión de Base de Datos.

Un sistema de gestión de bases de datos (SGBD) es un conjunto de programas que permiten el almacenamiento, modificación y extracción de la información en una base de datos; además de proporcionar herramientas para añadir, borrar, modificar y analizar los datos. Los SGBD proporcionan métodos para mantener la integridad de los datos, administrar el acceso de usuarios y recuperar la información si el sistema se corrompe.

Dentro de los distintos SGBD se escoge para el desarrollo del subsistema de administración es PostgreSQL v9.4

PostgreSQL: Es considerado entre los SGBD como uno de los más completos. La versión 9.4 contiene una gran cantidad de mejoras que hacen la administración, consulta y programación

en PostgreSQL mucho más fácil con respecto a sus versiones anteriores. Es multiplataforma y posee una documentación organizada, detallada y pública. Además es altamente configurable y extensible para muchas aplicaciones y posee un sofisticado analizador/optimizador de consultas.

A partir de las características expuestas se concluye que PostgreSQL 9.4 es aplicable a la investigación por ser altamente escalable, tanto en la enorme cantidad de datos que puede manejar como en el número de usuarios concurrentes. Puede mantenerse en línea por períodos prolongados y requiere de poco (o en muchos casos de ningún) mantenimiento.

1.8 Marcos de trabajo.

Un marco de trabajo simplifica el desarrollo de una aplicación mediante la automatización de algunos de los patrones utilizados para resolver las tareas comunes. Además, un framework proporciona estructura al código fuente forzando al desarrollador a crear código más legible y más fácil de mantener. Facilita la programación de aplicaciones ya que encapsula operaciones complejas en instrucciones sencillas.

Symfony v1.4.10: Es un framework multiplataforma diseñado para optimizar el desarrollo de las aplicaciones web. Separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja. Además automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación. Está desarrollado completamente con PHP 5 y es compatible con la mayoría de gestores de bases de datos (11).

Symfony fue seleccionado porque la arquitectura y patrones de diseños que emplea dotan a los sistemas que lo utilizan de una robustez y seguridad, especialmente para proyectos de la envergadura como Señal ACN e incluso más grandes. Además cuenta con una gran comunidad internacional que ofrece soporte y solución a problemas que puedan surgir durante la codificación de una aplicación web.

Bootstrap¹ 3: Es un framework diseñado para ayudar a diseñadores y desarrolladores en el proceso de creación de sitios web y aplicaciones de una manera rápida y fácil. Es de código abierto que está basado y pensado para la web móvil “mobile-first approach” o enfoque móvil de primera. Entre las principales ventajas que presenta se encuentra:

- Aporta un estilo base a todos los elementos HTML

¹ <http://getbootstrap.com/>

- Posee una documentación muy detallada y abundante, cosa que no ocurre con otros frameworks.
- Incluye una lista extensa de componentes que incluye: dropdowns, botones, barras de navegación, alertas, barras de progreso.
- Contiene en el conjunto de herramientas y componentes de interfaz que trae este sistema para crear diseños flexibles.
- posee características de respuestas que hacen que sus diseños aparezcan de la forma más apropiada en los diferentes dispositivos y resoluciones de pantallas.

1.9 Tecnologías de desarrollo.

HTML 5²(del inglés Hyper Text Markup Language; Lenguaje de Marcado de Hipertexto): describe la estructura y el contenido en forma de texto a la vez que permite complementar texto y objetos. La razón principal por la que se emplea HTML5, es que esta versión posibilita desarrollar páginas compatibles con la mayoría de los navegadores web, algo que no realizan sus versiones precedentes.

jQuery³ **1.9:** Es una biblioteca de JavaScript que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web. Es software libre y de código abierto que al igual que otras bibliotecas, ofrece una serie de funcionalidades basadas en JavaScript que de otra manera requerirían de mucho más código, es decir, con las funciones propias de esta biblioteca se logran grandes resultados en menos tiempo y espacio.

CSS3⁴(del inglés Cascading Style Sheets; Hojas de Estilo en Cascada): es un lenguaje utilizado para dar estilo a documentos HTML y XML⁵ (del inglés eXtensible Markup Language; Lenguaje de Marcas Extensible), separando el contenido de la presentación.

1.10 Lenguaje de Programación.

Un lenguaje de programación es diseñado para describir el conjunto de acciones consecutivas que un equipo debe ejecutar; estos se componen de un conjunto de reglas sintácticas y semánticas que permiten expresar instrucciones que luego serán interpretadas. El programador

² <http://www.w3schools.com>

³ <http://jquery.com/>

⁴ <http://www.css3.info/>

⁵ XML es un lenguaje muy simple que permite la lectura de datos a través de diferentes aplicaciones

es el encargado de utilizarlo para crear un conjunto de instrucciones que constituirán un programa informático.

1.10.1 Lenguajes del lado del cliente.

JavaScript⁶: es un lenguaje de programación interpretado que permite el desarrollo de interfaces de usuario mejoradas y páginas web dinámicas. Actualmente todos los navegadores modernos interpretan el código JavaScript integrado en las páginas web. Dentro de sus principales ventajas se encuentra el ser seguro, no requerir de compilación ya que funciona del lado del cliente y el ser independiente de la plataforma.

1.10.2 Lenguajes del lado del servidor.

PHP 5.4: Es un lenguaje de alto nivel, especialmente pensado para desarrollos web y el cual puede ser incrustado en páginas HTML. Fue creado con el objetivo de construir páginas web dinámicas en las que exista procesamiento del lado del servidor. Permite a las aplicaciones web ejecutadas por un servidor generar contenido de manera dinámica, por ejemplo, información contenida en una base de datos para su alteración.

Es un lenguaje que no solo es rápido al ser ejecutado sino que no genera retrasos en la máquina. También se integra muy bien junto a otras aplicaciones. Utiliza su propio sistema de administración de recursos y posee de un sofisticado método de manejo de variables, conformando un sistema robusto y estable. Además maneja distintos niveles de seguridad y los usuarios con experiencia en C y C++ podrán utilizar PHP rápidamente. Además PHP dispone de una amplia gama de librerías y permite la posibilidad de agregarle extensiones (12).

1.11 Entorno de Desarrollo Integrado

NetBeans IDE v8.4 es un entorno de desarrollo gratuito y de código abierto. Permite el uso de un amplio rango de tecnologías de desarrollo tanto para escritorio, aplicaciones Web o para dispositivos móviles. Además puede instalarse en varios sistemas operativos: Windows, Linux, MacOS. (13). Debido a la alta integración que posee con el lenguaje PHP y Symfony, con un eficiente completamiento automático de código, generación de funcionalidades menores y ejecución de los comandos que brinda Symfony para la configuración de cualquier aplicación web, cuyas características son indispensables para la implementación del Subsistema de Administración de Señal ACN.

⁶ <https://www.javascript.com>

1.12 Servidor de aplicaciones web.

Se selecciona el Servidor web **Apache⁷ 2.4** por ser flexible, rápido y eficiente, continuamente actualizado y adaptado a los nuevos protocolos, por ser multiplataforma y adaptarse a diferentes entornos y necesidades. Es de código abierto y distribuido bajo la Licencia Pública General Reducida de GNU, o más conocida por su nombre en inglés GNU Lesser General Public License.

Además Apache 2.4 proporciona diferentes módulos de apoyo para el desarrollo de otros módulos más específicos y extensibles, característica que adquiere gracias a ser modular lo que ha permitido el desarrollo de diversas extensiones entre las que sobresale PHP, lenguaje de programación del lado del servidor. La versión 2.4 de Apache cuenta con una comunidad de más de 4.000 contribuyente y mejora el rendimiento con respecto a sus versiones precedentes.

1.13 Conclusiones del capítulo.

En este capítulo se realizó una descripción del objeto de estudio donde se define los SPN, así como los diferentes procesos de administración que se realizan en el Señal ACN y PRIMICIA v1.7. El análisis de las soluciones existentes en el ámbito internacional permitió concluir que las características de dichos sistemas no cumplen con los requerimientos especificados, por lo que se toma como referencia el subsistema de administración de Señal ACN. El análisis de las herramientas y la metodología sirve como base para la realización del diseño e implementación del subsistema.

⁷ <http://www.apache.org>

CAPÍTULO 2: Características del sistema.

En el presente capítulo se realiza la descripción de los requisitos funcionales y no funcionales que deberá cumplir el subsistema de administración para dar solución al problema. Así como los patrones utilizados en la implementación del sistema. Además presenta la descripción del modelo de negocio así como lo diagramas relacionados.

2.1 Modelo de Negocio.

El modelo de negocio es una técnica para modelar el funcionamiento de una organización a través de sus procesos de negocio (14). Una vez que se logra determinar en el proceso de negocio quienes son las personas que lo inician cada uno de estos procesos y además quienes son los desarrolladores de las actividades dentro de cada uno de los procesos entonces se podrá realizar una correcta modelación del negocio de los clientes.

2.1.1 Actores del negocio.

En el modelado de negocio un actor es un rol o papel que juega una persona u otro sistema en algún proceso de negocio de una organización (14).

Actor	Descripción
Jefe de Prensa	El Jefe de Prensa es la persona que dirige, supervisa y controla todos los procesos informacionales del Departamento de Prensa de la Oficina de Comunicación y Relaciones Institucionales del ministerio.

2.1.2 Trabajadores del negocio.

Define el comportamiento y responsabilidades (rol) de un individuo, grupo de individuos, sistema automatizado o máquina, que trabajan en conjunto como un equipo. Ellos realizan las actividades y son propietarios de elementos.

Trabajador	Descripción
Redactor	Persona encargada de elaborar las noticias que se publican en los medios de difusión que tiene el ministerio, escogen las medias, que pueden estar acompañando a dichas noticias.
Editor	Encargado de aprobar las noticias realizadas por los redactores y mandar a publicarla, esta persona puede cambiar toda la

		información referente a una noticia.
Periodista		Persona encargada de elaborar los guiones de los videos que se van a confeccionar para ser visualizados en la red de televisión interna, además de escoger los videos e imágenes que se utilizarán para su construcción.
Administrador del Archivo		Persona encargada de toda la gestión de los materiales audiovisuales de interés ministerial.

2.1.1 Modelo de Casos de Uso del Negocio.



Fig. 1 Modelo de Casos de Uso del Negocio.

2.1.2 Descripción de Caso de Uso del Negocio.

Tabla 1 Descripción del Caso de Uso. Realizar Noticia.

Caso de uso	Realizar Noticias.
Actores	Agencia Cubana de Noticias (ACN/AIN).
Resumen	Este caso de uso comienza cuando se va confeccionar una noticia, posteriormente designa a un grupo de redactores que se encargarán de la redacción de la noticia. Después que la noticia es redactada pasa a manos de un editor que es el encargado de revisarla y corregir los errores que pueda encontrar dejándola lista para que sea

Acción del actor	Respuesta del proceso del negocio.
1. El caso de uso se inicia cuando va a realizar la redacción de una noticia.	publicada. 2. Los redactores consultan las fuentes de prensa que le pueden servir para la confección de la noticia.
	3. Los redactores redactan la noticia. Durante la redacción de la noticia esta puede estar acompañada por archivos de multimedia (imágenes o videos).
	4. El editor corrige los errores que detecta en la redacción de la noticia y la deja lista para ser publicada.
5. Se realiza una noticia de video e imágenes.	6. Los periodistas se encarga de realizar los videos ya sean de un reportaje o de un documental, así como la captura de imágenes que puede ser utilizadas también en la confección de un video. Estos archivos de multimedia también pueden ser utilizados en la realización de una noticia.
	7. Terminada la grabación del video los periodistas se encargan de realizar un guion para los videos que serán confeccionados.
	8. Una vez realizado los archivos de multimedia (videos o imágenes), estos pasan por una edición para ser confeccionados.
	9. Una vez confeccionado el archivo de multimedia pasa por un equipo que los revisa y lo aprueba.
	10. Si la multimedia es aprobada se guarda para ser utilizado.

Flujo alternativo.	
9a. El archivo es rechazado	Regresa al paso 8 del flujo central de eventos.

Tabla 2 Descripción de Caso de Uso del Negocio. Confeccionar reportes.

Caso de uso	Confeccionar reportes.
Actores	Agencia Cubana de Noticias (ACN/AIN).
Resumen	Este caso de uso comienza cuando se va confeccionar los reportes para conocer como se ha estado trabajando en la institución.
Acción del actor	Respuesta del proceso del negocio.
11. El caso de uso se inicia cuando van a realizar los reportes.	12. El Jefe de Redacción realiza los reportes para conocer como se ha ido trabajando en la institución en un período de tiempo determinado. Los reportes sirven para conocer el trabajo de los de los empleados de la agencia, para poder realizar el pago del salario por resultados.

2.2 Requisitos funcionales.

Los requisitos funcionales son capacidades o condiciones que el sistema debe cumplir. Se mantienen invariables sin importar con que propiedades o cualidades se relacionen (15). Son los que definen las acciones que debe realizar el sistema desde el punto de vista de las necesidades del usuario. Durante el estudio de los procesos actuales, así como el análisis de algunos de los gestores de contenidos más importantes en la actualidad, se determinaron los requisitos con que debía contar el Subsistema de Administración. A continuación se relacionan dichos requisitos:

RF1 Ordenar noticias.

Descripción: El sistema debe ser capaz de ordenar las noticias por relevancia dentro de cada sección.

RF2 Buscar recursos.

Descripción: El sistema debe ser capaz de realizar una búsqueda de las fotos, noticias, video y música teniendo en cuenta autor, título, palabra clave, editor, tema.

RF3 Eliminar recursos.

Descripción: El sistema debe ser capaz de eliminar los videos, las fotos, las noticias y la música que el usuario especifique.

RF4 Encolar recurso.

Descripción: El sistema debe ser capaz de ordenar las noticias por fecha de elaboración.

RF5 Posicionar fotos.

Descripción: A las noticias que contengan fotos el sistema debe permitir posicionar la foto donde el usuario especifique.

RF6 Registrar las trazas de los materiales que se elaboren.

Descripción: El sistema debe ser capaz de almacenar todos los cambios que se realicen en la plataforma.

RF7 Subir el video, audio e imagen junto con la noticia.

Descripción: El sistema debe permitirle al usuario subir video, audio e imagen junto con la noticia en la misma página, sin necesidad de enviar el archivo.

RF8 Generar reportes.

Descripción: El sistema debe ser capaz de generar reportes dependiendo de su tipo (redactores, correctores, temáticos, música, video, imagen).

RF9 Visualizar reportes.

Descripción: El sistema debe ser capaz de visualizar reportes que se generaron

RF10 Imprimir reportes.

Descripción: El sistema debe ser capaz de imprimir reportes que se generaron.

RF11 Catalogación de las medias en temporal e intemporal.

Descripción: El sistema debe permitir catalogar las medias en temporal e intemporal.

2.3 Requisitos no funcionales.

Los requisitos no funcionales son propiedades o cualidades que el producto debe tener. Son características que hacen al producto atractivo, usable, rápido y confiable. Son importantes para que clientes y usuarios puedan valorar las características no funcionales del producto. Las propiedades no funcionales como cuán usable, seguro y agradable pueden marcar la diferencia entre un producto bien aceptado y uno con poca aceptación (15).

Los requisitos no funcionales se clasifican en múltiples categorías, a continuación se mencionan:

Requisitos de apariencia o interfaz externa: El sistema deberá contar de manera general, con una interfaz sencilla, que sea intuitiva y de fácil comprensión, garantizando la correcta interacción con el sistema.

Requisitos de portabilidad: El sistema estará realizado para funcionar en los sistemas operativos GNU/Linux o Windows garantizando un producto multiplataforma.

Requisitos de Software:

En el servidor:

Sistema operativo GNU/Linux Ubuntu, servidor de aplicaciones Apache 2, servidor de bases de datos PostgreSQL 9.4, PHP 5.4.

En los clientes:

Navegador web Mozilla Firefox 30.0 o superior y Chrome 38.

Requisitos de Hardware:

En el servidor:

Procesador Dual-Core Xeon 2.33 GHz, 4Gb Memoria RAM, 500 GB Disco Duro, Tarjeta de Red Ethernet Gigabit con 2 puertos, Tarjeta de Video Hauppauge WinTV PVR 350.

En los clientes:

Procesador P-IV a 2.0 GHz o superior, 512 Mb de RAM, Tarjeta de Red Ethernet Gigabit.

Restricciones en el diseño y la implementación: Para la modelación del sistema se utilizará el lenguaje UML y a través de la herramienta Visual Paradigm. Se requiere el uso de la arquitectura Modelo-Vista-Controlador implementada por el framework Symfony v1.4.10 de PHP. La arquitectura debe soportar migrar la interfaz de usuario de forma rápida, para lograr visualizar cualquiera de los cambios que se produzcan.

2.3.1 Diagramas de Casos de Uso.

El diagrama de casos de uso de sistema muestra la interacción entre los actores y las funcionalidades de un sistema de software agrupadas en casos de uso. Los casos de uso (CU) agrupan un conjunto de requisitos funcionales que indican qué hará el sistema, mientras que los actores no son más que individuos que interactúan con estos (16). A continuación en la Fig. 2 se muestra el diagrama de CU del sistema.

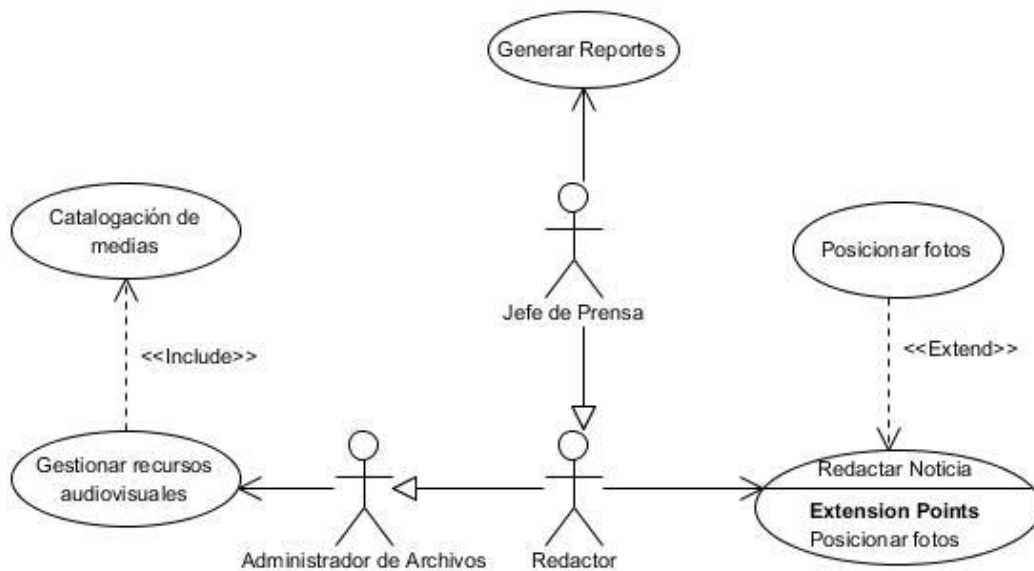


Fig. 2 Diagramas de casos de usos del sistema.

2.3.2 Descripción de los Casos de Usos.

Tabla 3 Descripción de los Casos de Usos. Generar Reportes.

Caso de uso	Generar Reportes.
Actores	Jefe de Prensa
Resumen	El caso de uso inicia cuando el actor desea conocer el desempeño de sus trabajadores o realizar una búsqueda atendiendo a diferentes criterios.
Acción del actor	Sistema
1. El actor selecciona la opción generar reportes según el criterio (redactor, corrector, temático, música, videos, imagen).	2. Muestra los formularios según el criterio de los reportes.
3. El actor llena los campos de los datos requeridos para la generación del reporte y da la opción de mostrar reportes	4. El sistema visualiza los reportes según los datos introducidos en los formularios.
Flujo alterno	
3a. El actor deja campos vacíos	

.	<ol style="list-style-type: none"> 1. El sistema muestra varios mensajes: <ul style="list-style-type: none"> ➤ La fecha de fin ("") debe ser mayor que la fecha de inicio (""). ➤ Debe al menos especificar una fecha. ➤ Requerido.
2. Completa los campos vacíos.	3. Regresa al paso 4 del flujo central de eventos.
3b. El actor selecciona mal los campos de fecha de inicio y fin.	
	<ol style="list-style-type: none"> 1. El sistema muestra el mensaje: <ul style="list-style-type: none"> ➤ La fecha de fin ("") debe ser mayor que la fecha de inicio ("").
2. El actor selecciona correctamente los datos de fecha de inicio y fin.	3. Regresa al paso 4 del flujo central de eventos.

Tabla 4 Descripción de los Casos de Usos. Posicionar fotos.

Caso de uso	Posicionar fotos
Actores	Jefe de Prensa, Redactor
Resumen	El caso de uso inicia cuando el actor accede al módulo de redactar noticia, el sistema le mostrará un formulario para que entre el título, seleccione la sección y el fondo musical de la noticia y después entrará el cuerpo de la noticia dividido en pantallas que pueden ser de varios tipos (texto, imagen, texto-imagen y video). Selecciona la opción de texto-imagen, cuando el usuario decide que terminó la redacción de la noticia, escoge la opción de posicionar fotos dentro de la noticia redactada.
Acción del actor	Sistema
1. El actor selecciona la opción de redactar noticia.	2. Muestra las noticias redactadas y la opción de agregar una nueva noticia.

3. El actor debe seleccionar la opción de adicionar noticia.	4. Muestra los formularios para que introduzca el título. la sección a la que pertenece y el fondo musical de la noticia.
5. Debe seleccionar la opción de guardar la noticia.	6. Muestra un formulario para seleccionar el tipo de noticia que se desea redactar.
7. El usuario escoge la opción de texto-imagen.	8. Muestra un formulario con un campo de texto, para redactar la noticia y un campo seleccionable para seleccionar la imagen.
9. El usuario escoge la opción para posicionar la foto en cualquier parte del texto	10. El sistema posiciona la foto en el lugar seleccionado.
11. Escoge la opción de guardar noticia.	12. Muestra la noticia redactada.
Flujo alternativo	
4a. El actor deja los campos vacíos.	
	1. El sistema muestra un mensaje de Requerido.
2. El actor completa los campos.	3. Regresa al paso 6 del flujo central de eventos.
7a. El actor no escoge el tipo de noticia que desea redactar.	
	1. El sistema muestra un mensaje de Seleccione el tipo de pantalla.
2. El actor escoge texto-imagen.	Regresa al paso 8 del flujo central de eventos.
8a. El actor no redacta la noticia ni selecciona la imagen que desea insertar.	
	1. El sistema muestra un mensaje de Requerido.
2. El actor redacta la noticia y selecciona la imagen que desea insertar.	3. Regresa al paso 9 del flujo central de eventos.

Tabla 5 Descripción de los Casos de Usos. Catalogación de medias.

Caso de uso	Catalogación de medias.
Actores	Jefe de Prensa, Redactor, Administrador de Archivos.

Resumen	Este caso de uso comienza cuando los actores desean catalogar fichero del repositorio de medias de la entidad. Para ello accede al menú principal y selecciona la opción Medias.
Acción del actor	Sistema
1. El actor selecciona la opción de medias.	2. Muestra un formulario con los tipos de medias (imagen, video y audio) además del género y autor listados todos en una tabla.
3. Escoge el tipo de media (imagen, video o audio) al que desee realizar la operación.	4. Muestra las que se encuentran el servidor.
5. Escoge la opción de adicionar según el tipo de medias.	6. Muestra los formularios de nombre, autor, género, ubicación y la clasificación de temporal e intemporal
7. El actor escoge la opción de guardar el archivo multimedia seleccionado.	8. El sistema agrega la media.
Flujo alterno.	
6a. El actor deja vacío los campos.	
	1. El sistema muestra mensaje de Requerido.
2. El actor completa los campos.	3. Regresa al paso 7 del flujo central de eventos.
6b. El actor no escoge la opción de clasificar las medias.	
	1. El sistema no clasifica las medias.
2. El actor escoge la opción de clasificar las medias.	3. El sistema clasifica las medias.

2.4 Patrones.

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 (17). En desarrollo del sistema se ven evidenciado el uso de patrones de los cuales a continuación se realiza una descripción.

2.4.1 Patrones de casos de uso.

Los patrones de casos de uso tienen como objetivo representar las funcionalidades del sistema de manera tal que el modelo sea reutilizable y entendible. Para el desarrollo del Subsistema de Administración para canal Señal ACN se hizo empleo de los patrones de CU que se enumeran a continuación:

Múltiples Actores con Rol Común: Este patrón indica que dos actores juegan el mismo rol sobre un caso de uso. Evidenciado en la relación de herencia entre Jefe de Prensa y Redactor.

Extensión Concreta: Este patrón se aplica cuando un flujo puede extender del flujo de otro caso de uso. Evidenciado en la relación entre el CU Redactar Noticia y el CU Posicionar fotos, donde en la redacción de la noticia el usuario puede posicionar una imagen de cualquier parte de la noticia.

Inclusión: Se incluye una relación del caso de uso base al caso de uso de inclusión. El último puede ser instalado en sí mismo. El caso de uso base puede ser concreto o abstracto. Como ejemplo de su aplicación se puede ver la CU Gestionar recursos audiovisuales y CU Catalogación de medias, donde dentro la gestión de media se incluye la catalogación en temporal e intemporal (17).

2.4.2 Patrones de arquitectura.

Los patrones arquitectónicos o patrones de arquitectura, expresan un esquema de organización estructural esencial para un sistema de software que consta de subsistemas, sus responsabilidades e interrelaciones. En comparación con los patrones de diseño, los patrones arquitectónicos tienen un nivel de abstracción mayor (17). Para la implementación de Symfony se utilizan varios patrones situándolos en las capas de Modelo y Control que plantea el patrón arquitectónico MVC.

Modelo Vista Controlador MVC.

El patrón de arquitectura MVC conocido por sus siglas en inglés Model View Controller que significa *Modelo Vista Controlador*, permite realizar la programación multicapa separando en tres componentes distintos los datos de una aplicación, la interfaz del usuario y la lógica de control. Este patrón se ve usualmente en aplicaciones web donde la vista es la página Html y el código que provee de datos dinámicos a la página, el modelo es el sistema de gestión de base de datos y el controlador representa la lógica del negocio (17).

La Fig.3 se muestra el patrón MVC como lo implementa el marco de trabajo Symfony.

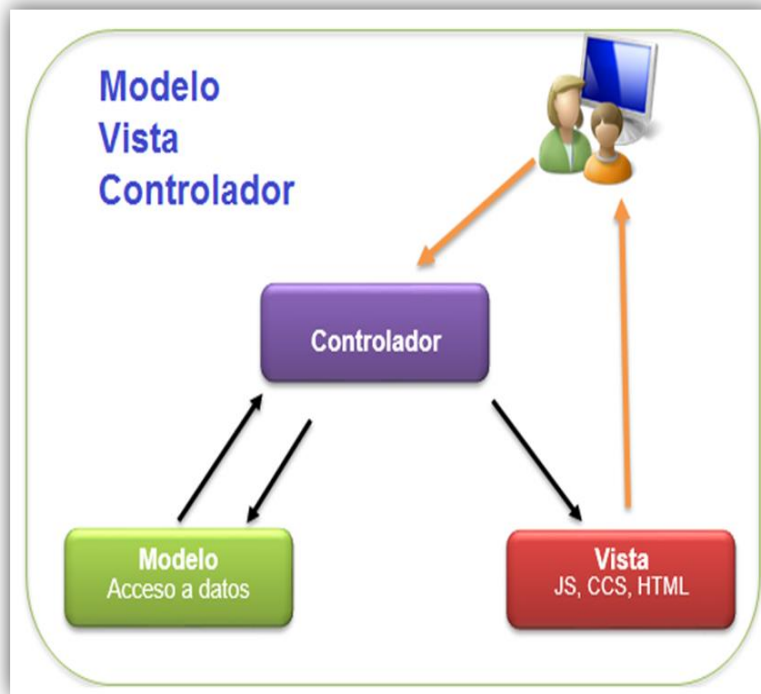


Fig. 3 Patrón MVC para el marco de trabajo Symfony. Fuente de elaboración: propia.

Symfony está basado en un patrón clásico del diseño web conocido como arquitectura MVC que está formado por 3 niveles:

- El modelo representa la información con la que trabaja la aplicación, es decir, su lógica de negocio.
- La vista transforma el modelo en una página web que permite al usuario interactuar con ella.
- El controlador se encarga de procesar las interacciones del usuario y realiza los cambios apropiados en el modelo o en la vista.

La arquitectura MVC separa la lógica de negocio (el modelo) y la presentación (la vista) consiguiendo un mantenimiento más sencillo de las aplicaciones. Por ejemplo, si una aplicación debe ejecutarse tanto en un navegador estándar como un navegador de un dispositivo móvil solamente es necesario crear una vista nueva, manteniendo el controlador y el modelo original. El controlador se encarga de aislar al modelo y a la vista de los detalles del protocolo utilizado para las peticiones. El modelo se encarga de la abstracción de la lógica relacionada con los datos haciendo que la vista y las acciones sean independientes de, por ejemplo, el tipo de gestor de bases de datos utilizados por la aplicación.

La ventaja de utilizar MVC es que realiza una separación total entre lógica de negocio y presentación. A esto se le pueden aplicar opciones como el multilinguaje, distintos diseños de

presentación, etc. sin alterar la lógica de negocio. La separación de capas como presentación, lógica de negocio, acceso a datos es fundamental para el desarrollo de arquitecturas consistentes, reutilizables y de fácil mantenimiento lo que al final resulta en un ahorro de tiempo en desarrollo en posteriores proyectos (18).

2.4.3 Diagrama de Clase del Diseño.

El diagrama de clases del diseño describe gráficamente las especificaciones de las clases de software y de las interfaces en una aplicación. Generalmente contiene las clases, asociaciones, atributos, interfaces, métodos y dependencias (19). A continuación en la Fig. 4 se presenta el diagrama de clases del diseño correspondiente al caso de uso Administrar Reportes.

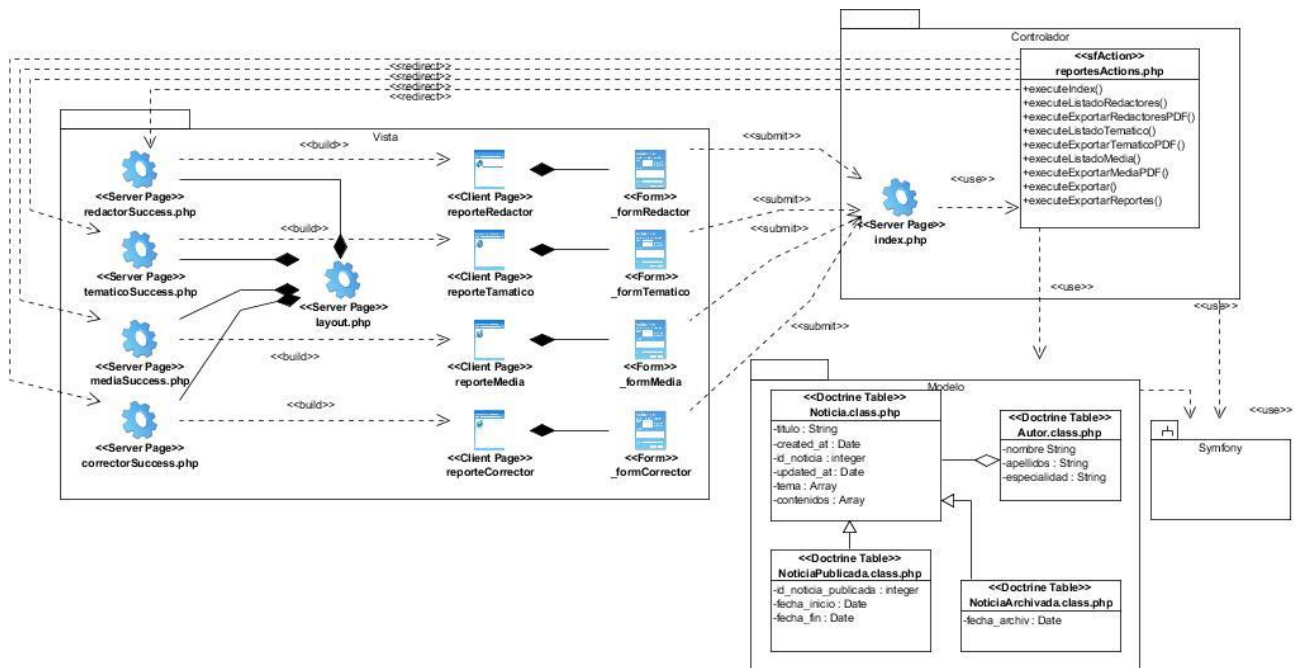


Fig. 4 Diagrama de Clase del Diseño de CU Administrar Reportes.

El diagrama de clases representado anteriormente contiene las principales clases que intervienen en el proceso de administración de reportes con sus principales métodos, atributos y relaciones. Para lograr un mejor entendimiento se han agrupado las clases pertenecientes al negocio en el Modelo, Vista y Controlador.

A continuación se explica cada una de las clases.

El Modelo contiene las clases encargadas del acceso a datos, la cuales están definidas según la estructura que propone Symfony. La clase Noticia que representa la entidad que contiene los atributos principales de una noticia así como los métodos para acceder y modificar estos

atributos. Las clases *NoticiaPublicada* y *NoticiaArchivada* heredan de *Noticia*, *NoticiaPublicadas* contiene los atributos de una noticia que ha sido publicada y *NoticiasArchivadas* los atributos de una noticia archivada. Por otra parte la clase *Autor* contiene los métodos que permiten saber el autor de una noticia.

En la Vista se encuentran las clases encargadas de presentar la información al usuario. Las clases *redactorSuccess.php*, *tematicoSuccess.php*, *mediaSuccess.php*, *correctorSuccess.php*, *imprimirSuccess.php*, son clases que construyen las interfaces del sistema. La clase *layout.php* es la que define la estructura de la interfaz gráfica del sistema. El *layoutImprimir.php* define la interfaz para imprimir los reportes generados. Los *ClientPage* son los que contienen el código HTML que definen las estructuras de la vista.

Los Form son donde se muestran y capturan los datos por ejemplo *formRedactor* se muestran los datos del Redactor. El Controlador posee la clase *reportesActions.php* que contiene los métodos solicitados por la vista para realizar acciones sobre el modelo.

2.4.4 Patrones de diseño.

Los patrones de diseño son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de software y otros ámbitos referentes al diseño de interacción o interfaces. Un patrón de diseño resulta ser una solución a un problema de diseño. Para que una solución sea considerada un patrón debe poseer ciertas características. Una de ellas es que debe haber comprobado su efectividad resolviendo problemas similares en ocasiones anteriores. Otra es que debe ser reutilizable, lo que significa que es aplicable a diferentes problemas de diseño en distintas circunstancias (17).

Patrones GRASP.

GRASP es un acrónimo de General Responsibility Assignment Software Patterns (patrones generales de software para asignar responsabilidades). Describen los principios fundamentales del diseño de objetos y la asignación de responsabilidades expresados como patrones. A continuación se presentan los patrones GRASP que se utilizan para la creación del subsistema de administración para el canal Señal ACN:

Experto: Uno de los más utilizados, puesto que Doctrine (librería externa que utiliza Symfony para realizar su capa de abstracción en el modelo), encapsula toda la lógica de los datos y son generadas las clases con todas las funcionalidades comunes de las entidades. Por ejemplo en la siguiente figura se muestra el uso de Doctrine para acceder a los datos.

```

public function executeIndex(sfWebRequest $request)
{
    $this->control_var = "imagen/filter";
    $this->control_var1 = "imagen/deletemany";

    $this->filter = new ImagenFormFilter();

    $query = Doctrine::getTable('Imagen')->createQuery('a')->orderBy('ci

```

Fig. 5 Ejemplo del uso del patrón Experto.

Creador: En la clase sfActions se encuentran las acciones definidas para la aplicación y se ejecutan cada una de ellas. En las acciones se crean los objetos de las clases que representan las entidades, evidenciando de este modo que la clase sfActions es "creador" de dichas entidades. Por ejemplo en la siguiente figura se crea el objeto de la clase BuscarForm().

```

/
class buscarActions extends sfActions
{
    /**
     * Executes index action
     *
     * @param sfRequest $request A request object
     */
    public function executeIndex(sfWebRequest $request)
    {
        $this->form = new BuscarForm();
    }

    public function executeBusqueda(sfWebRequest $request)
    {
        $this->form = new BuscarForm();
        $this->recurso = $this->processbuscarForm($request, $this->form);
        $this->setTemplate('index');
    }
}

```

Fig. 6 Ejemplo del uso del patrón Creador.

Alta Cohesión: Symfony permite la organización del trabajo en cuanto a la estructura del proyecto y la asignación de responsabilidades con una alta cohesión. Un ejemplo de ello es la clase Actions, la cual está formada por varias funcionalidades que están estrechamente relacionadas, siendo la responsable de definir las acciones para las plantillas y colaborar con otras para realizar diferentes operaciones, instanciar objetos y acceder a las propiedades.

Bajo Acoplamiento: Su principal objetivo es lograr poca dependencia entre clases, con esto se reduce el impacto de los cambios en otros componentes. Además es conveniente para la reutilización, ya que las clases están lo menos ligadas posible (17). Por ejemplo en la siguiente figura las clases que están en la raíz de la carpeta Doctrine pueden modificarse sin alterar la estructura de las clases base, que son las que se comunican directamente con la base de datos.

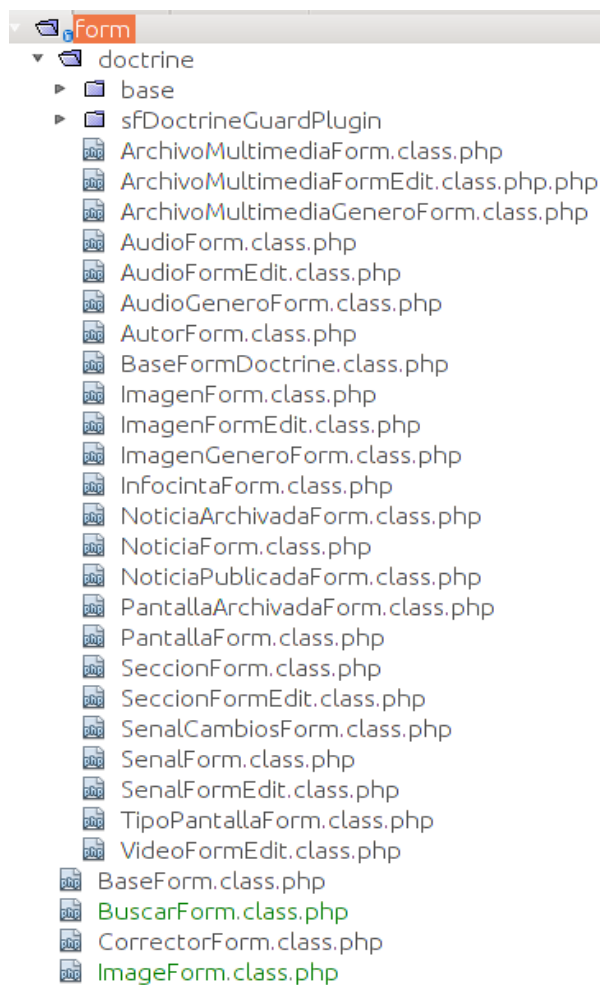


Fig. 7 Ejemplo del uso del patón Bajo Acoplamiento.

Controlador: Todas las peticiones Web son manejadas por un solo controlador frontal (sfActions), que es el punto de entrada único de toda la aplicación en un entorno determinado. Cuando el controlador frontal recibe una petición, utiliza el sistema de enrutamiento para asociar el nombre de una acción y el nombre de un módulo con la URL entrada por el usuario.

Patrones de diseño GOF.

Los patrones GoF (Gang of Four), en español Banda de los Cuatro se clasifican en 3 categorías basadas en su propósito: creacionales, estructurales y de comportamiento.

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

```
$configuration = ProjectConfiguration::getApplicationConfiguration('primicia',  
sfContext::getInstance($configuration)->dispatch());
```

Fig. 8 Ejemplo del uso del patrón Singleton.

Decorator: Añade funcionalidad a una clase, dinámicamente. El archivo `layont.php`, que también se denomina plantilla global, almacena el código HTML que es común a todas las páginas de la aplicación, para no tener que repetirlo en cada página. El contenido de la plantilla se integra en el layout, o si se mira desde el otro punto de vista, el layout decora la plantilla.

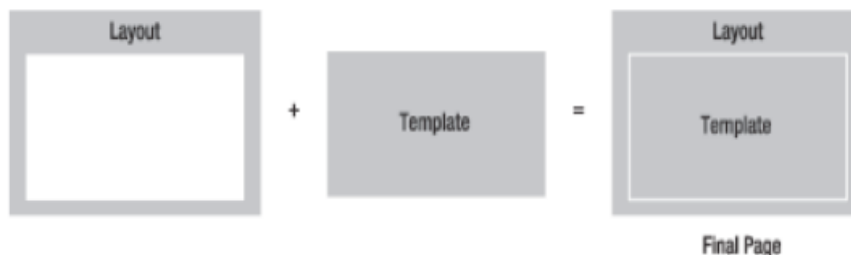


Fig. 9 Ejemplo del uso del patrón Decorator.

Composite: Permite tratar objetos compuestos como si de uno simple se tratase. Sirve para construir objetos complejos a partir de otros más simples y similares entre sí, gracias a la composición recursiva y a una estructura en forma de árbol. Esto simplifica el tratamiento de los objetos creados, ya que al poseer todos ellos una interfaz común, se tratan todos de la misma manera.

```

td></tr>
>getResults() as $noticia): ?>
    <h1lo">
        $noticia->Slug() ?>
    </h1ion">
        Doctrine::getTable('Seccion')->find($noticia->getSeccionId())->getNombre()
    </h1ate-update">
        format_date($noticia->getCreatedAt(), 'y/MM/dd | HH:mm:ss'); ?>
    </h1ate-update">
        format_date($noticia->getUpdatedAt(), 'y/MM/dd | HH:mm:ss'); ?>
    </h1iones">
        link_to(image_tag('ShowNews.jpg', array('title' => 'Mostrar', '&aux=index'
        link_to(image_tag('Edit.jpg', array('title' => 'Editar')), 'noticia/editn?
        link_to(image_tag('Delete-Logout.jpg', array('title' => 'Eliminar')), 'not

```

Fig. 10 Ejemplo del uso del patón Composite.

Facade: Que proporciona una interfaz unificada para un conjunto de interfaces de un subsistema. Define una interfaz de alto nivel que hace que el subsistema sea más fácil de usar. Como es el caso del acceso a base de datos donde existe una interfaz común para los diferentes gestores de bases de datos soportados por el ORM **(18)**.

2.5 Modelo de Datos.

El modelado de datos persigue el objetivo de identificar y representar gráficamente las entidades que participan en el sistema. En él se especifican las relaciones entre las entidades y sus atributos para brindar una información más completa al equipo de desarrollo. En la Fig. 11 se muestra el modelo de datos del Subsistema de Administración del canal Señal ACN.

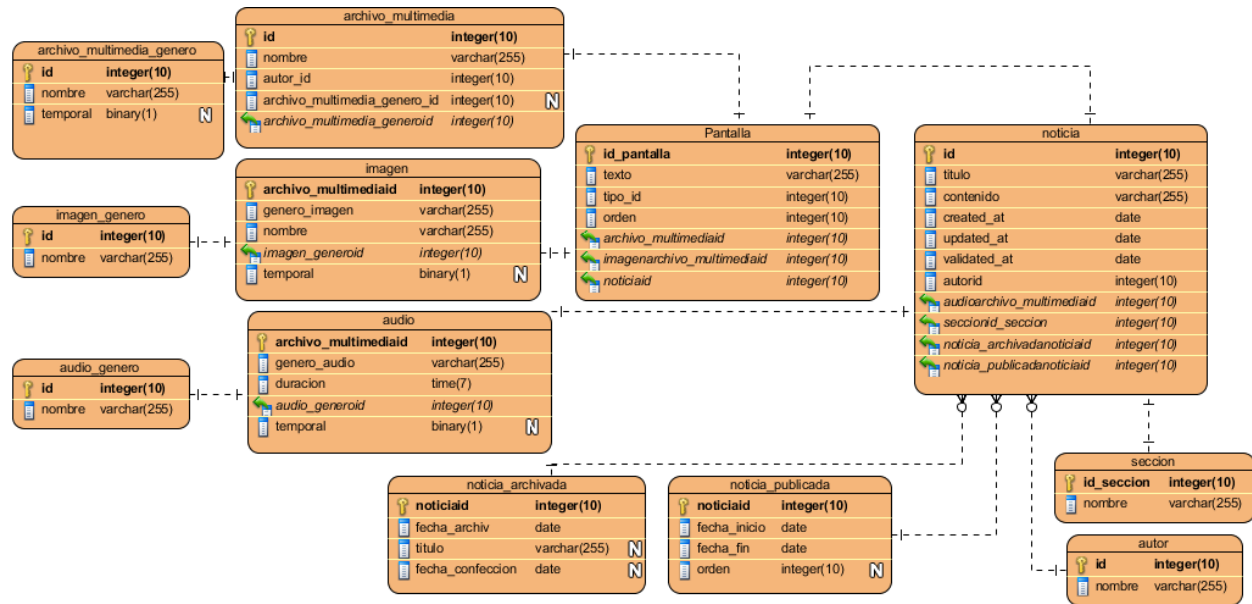


Fig. 11 Modelo de Datos del subsistema de administración.

2.5.1 Descripción del Modelo de Datos.

El modelo de datos cuenta con 12 entidades, imagen, audio, archivo_multimedia, noticia, noticia_archivada, noticia_publicada, autor, pantalla, sección, archivo_multimedia_genero, imagen_genero y audio_genero. En las entidades imagen, audio, archivo_multimedia se guardan la información de las imágenes, audios y videos respectivamente. La entidad noticia_publicada es la encargada de guardar la información de las noticias que se van a publicar por el canal, mientras que noticia_archivada es donde se guardan las noticias que fueron publicadas.

En la tabla sección se guarda la sección a la que pertenece la noticia que se va a redactar y en la tabla autor el nombre de los autores de los recursos multimedia. En las tablas imagen_genero, audio_genero y archivo_multimedia_genero se guardan los géneros de las medias. En la entidad pantalla, además de contener la información de archivo_multimedia o imagen, también va la información de la noticia a la que está asociada. En la entidad noticia se guarda la información del título de la noticia y el audio asociado.

2.6 Conclusiones del capítulo.

Las tareas de la investigación asociadas a las fases de análisis y diseño permitieron la captura de los requisitos del sistema. Identificándose cinco requisitos no funcionales y 11 requisitos funcionales agrupados en cinco casos de uso representados en el diagrama de casos de uso del sistema. Se modeló el diagrama de clases del diseño con el fin de representar la distribución lógica de los principales objetos y clases que intervienen en el desarrollo de la aplicación. Además se argumentó el uso de los patrones de casos de uso, arquitectónicos y de diseño que garantizan las buenas prácticas de diseño e implementación de la solución. En el modelo de datos se definió la estructura de la base de datos y la relación entre las entidades.

CAPÍTULO 3 Implementación y prueba.

En el presente capítulo se presenta el modelo de despliegue para la distribución del sistema, el modelo de implementación elaborado a partir de los resultados obtenidos en la fase de análisis y diseño llevada a cabo en el capítulo anterior. Se establecen el modelo de implementación y las pruebas de las funcionalidades del sistema. Las pruebas constituyen un elemento clave para la garantizar la calidad del sistema, y representan además una revisión final de todo el proceso de desarrollo.

3.1 Modelo de despliegue.

El Modelo de Despliegue describe la distribución física del sistema, muestra cómo están distribuidos los componentes de software entre los distintos nodos de cómputo. Permite comprender la correspondencia entre la arquitectura software y la arquitectura hardware (17).

A continuación se muestra el Modelo de Despliegue de la solución propuesta.

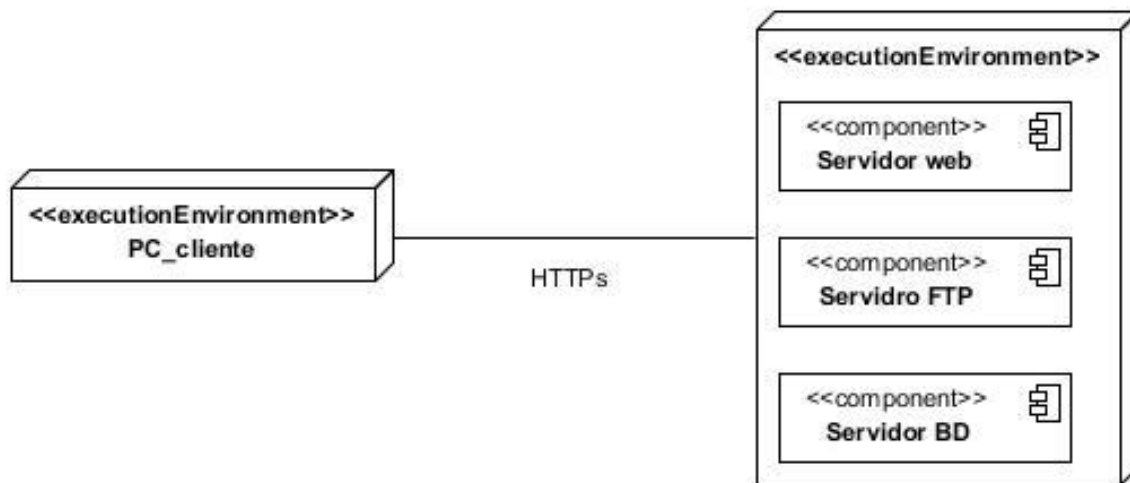


Fig. 12 Modelo de despliegue.

3.2 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 (17).

En el Diagrama de Componentes que se muestra a continuación en la Fig. 12, confeccionado durante el desarrollo de este trabajo, se visualiza la relación física existente entre los componentes considerados necesarios para la programación de los CU definidos con anterioridad. Además es utilizado con el fin de estructurar y controlar los componentes en los sistemas de software, fundamentalmente mostrando las dependencias existentes entre estos.

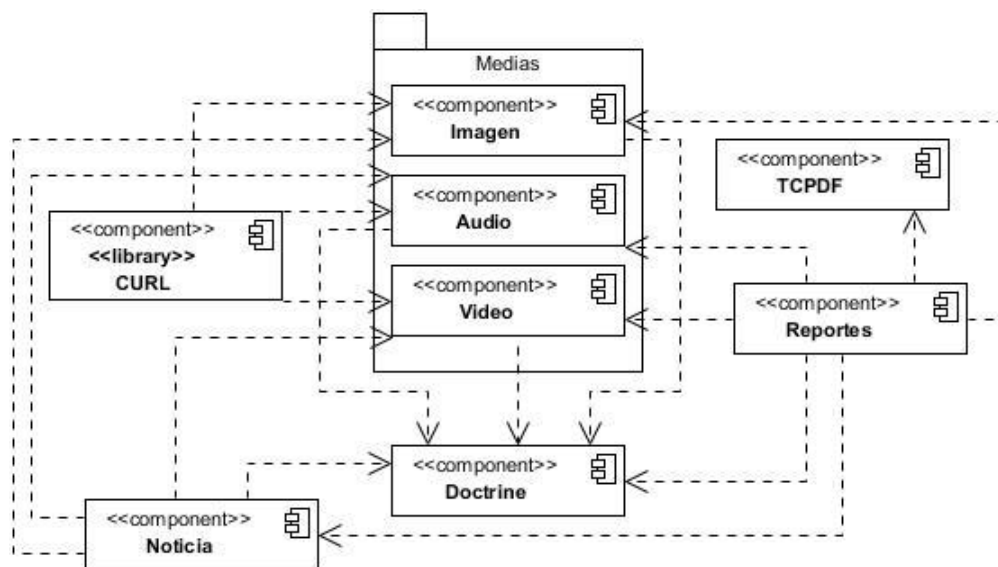


Fig. 13 Diagrama de Componentes.

En el siguiente modelo de implementación se representan componentes que hacen referencia a librerías y aplicaciones externas al propio framework de desarrollo utilizadas para la implementación de algunas funcionalidades de la aplicación; tal es el caso de **TCPDF**⁸ para la generación de reportes, **Doctrine** es un **ORM**⁹ para PHP que facilita la interacción con la Base de Datos y **CURL** es una herramienta para usar en un intérprete de comandos para transferir archivos con sintaxis URL.

3.3 Código fuente.

El código fuente de un sistema informático no es más que el lenguaje en que se representan las instrucciones que el ordenador debe ejecutar para llevar a cabo las funcionalidades del software. La presentación y estructura del código fuente varía según el lenguaje de programación empleado. Para almacenarlo de una manera organizada y comprensible los desarrolladores hacen empleo de estándares de codificación que facilitan la lectura del código

⁸ **TCPDF** es una clase de PHP5 para la generación de documentos PDF que no requiere extensiones externas.

⁹ **ORM** (Object-Relational Mapping, en español, Mapeo Relacional de Objetos) es una técnica de programación para convertir datos entre el sistema de tipos utilizado en un lenguaje de programación orientado a objetos y el utilizado en una base de datos relacional.

por parte de otros desarrolladores. La siguiente sección está destinada a explicar los estándares de codificación empleados durante el desarrollo del Subsistema de Administración del canal Señal ACN.

3.3.1 Estándares de codificación.

Un estándar de codificación es un conjunto de reglas de notación y nomenclatura, específicas de cada lenguaje de programación, que se usan y se siguen durante la fase de implementación (codificación) de una aplicación y reducen perceptiblemente el riesgo de que los desarrolladores introduzcan errores que no son detectados por los compiladores. Reduciendo el tiempo y coste de las actividades de depuración y pruebas necesarias para la detección y corrección de los mismos (20). La elaboración y el empleo de estándares de codificación ofrecen numerosas ventajas a la hora de elaborar cualquier tipo de producto software y entre sus principales ventajas se puede mencionar:

- Se reduce la posibilidad de cometer errores.
- Se obtiene un código legible y comprensible.
- Mejora la comunicación entre los miembros del grupo de programadores del equipo de desarrollo.

Para la elaboración de un estándar de codificación se debe tener en cuenta, entre otros aspectos, las especificaciones y características del framework de desarrollo y los lenguajes de programación escogidos.

A continuación se enumeran algunas de las normas de codificación empleadas:

- Las variables locales serán inicializadas en el momento de su declaración o en caso contrario cuando su valor inicial dependa de una llamada a otro método en un determinado momento.
- Se utilizarán llaves de apertura y cierre en todos los casos de condiciones y ciclos para aumentar la legibilidad del código fuente.
- Se hará empleo de la nomenclatura CamelCase. Las variables y métodos estarán escritos con el estándar lowerCamelCase, mientras que la declaración de clases hará empleo del estándar UpperCamelCase.
- Cada función debe tener un comentario, antes de su declaración, explicando qué hace. Tanto el nombre como el comentario que acompañe a la función deben bastar para ello.
- Todo el código implementado tendrá una indentación de 4 espacios, facilitando la legibilidad y la comprensión del código.
- El límite máximo de longitud de una línea de código debe ser de 80 caracteres.

3.3.2 Ejemplos de código fuente.

A continuación se presenta en la Fig. 13 un fragmento de código donde se muestra el empleo de los estándares de codificación descritos en la sección anterior. Donde `executeBusqueda()` y `executeListado()` muestran el uso de `lowerCamelCase`, mientras `BuscarForm()` el uso de `UpperCamelCase`.

```
public function executeBusqueda(sfWebRequest $request)
{
    $this->form = new BuscarForm();
    $this->recurso = $this->processBuscarForm($request, $this->form);
    $this->setTemplate('index');
}

public function executeListado(sfWebRequest $request)
{
    $this->termino=$this->getUser()->getAttribute('term');
    $this->filtro=$this->getUser()->getAttribute('fil');

    if($this->filtro == 0){
        $this->tabla = ImagenTable::getInstance();
        $this->result=$this->tabla->Reporte($this->termino);
    }
}
```

Fig. 14 Ejemplo de Estándares de Codificación.

3.4 Estrategia de prueba de software.

El flujo de trabajo de prueba consiste en la ejecución de un programa informático con la intención de descubrir errores. Las pruebas de software constituyen un elemento crítico para la garantía de calidad de un software. Estas permiten validar que el sistema implementado cumple con los requisitos definidos durante la fase de análisis y diseño (16). Las pruebas son aplicadas en diferentes momentos del ciclo de vida del software con el fin de comprobar hasta qué punto el programa desempeña un buen funcionamiento. Se considera que una prueba tiene éxito si detecta un error no identificado hasta el momento.

3.4.1 Pruebas unitarias y funcionales.

Dependiendo del modelo de desarrollo de software empleado para el desarrollo de aplicaciones y su implementación las pruebas unitarias y funcionales pueden jugar un papel más o menos central, pero siempre importante.

En las pruebas unitarias se analiza una porción de código que pueda ser analizada de manera aislada, por ejemplo funciones y métodos, a los que después de pasarle unos parámetros de entrada se deben obtener otros parámetros de salida claramente definidos. Validan la forma en la que las funciones y métodos trabajan en cada caso particular. Las pruebas unitarias se encargan de un único caso cada vez, lo que significa que un único método puede necesitar varias pruebas unitarias si su funcionamiento varía en función del contexto.

Existen numerosos frameworks para crear pruebas unitarias, dentro de los que se destacan PHPUnit y SimpleTest. Symfony incluye su propio framework para pruebas llamado Lime, el cual se basa en la librería Test: More de Perl y es compatible con TAP. Esto significa que los resultados de las pruebas se muestran con el formato definido en el “Test Anything Protocol”, creado para facilitar la lectura de los resultados de las pruebas. Además de proporcionar soporte para pruebas unitarias, Lime tiene las siguientes ventajas:

- Las pruebas de Lime son fáciles de leer y sus resultados también lo son. En los sistemas operativos que lo soportan, los resultados de Lime utilizan diferentes colores para mostrar de forma clara la información más importante.
- El núcleo de Lime se valida mediante pruebas unitarias.
- Está escrito con PHP, es muy rápido y está bien diseñado internamente. Consta únicamente de un archivo, llamado lime.php, y no tiene ninguna dependencia.

En las pruebas funcionales se evalúa la aplicación como un todo y se comprueba que se alcanzan determinados resultados tras realizar una serie de acciones sobre la aplicación. Estas pruebas simulan la navegación del usuario, realizan peticiones y comprueban los elementos de la respuesta, tal y como lo haría manualmente un usuario para validar que una determinada acción hace lo que se supone que tiene que hacer. En las pruebas funcionales, se ejecuta un escenario correspondiente a lo que se denomina un “caso de uso”.

Symfony dispone de un objeto especial, llamado sfBrowser, que actúa como un navegador que está accediendo a una aplicación, pero sin necesidad de utilizar un servidor web real. Este objeto permite el acceso directo a los objetos que forman cada petición (el objeto petición, el objeto sesión, el objeto contexto y el objeto respuesta). Symfony también dispone de una extensión de esta clase llamada sfTestBrowser, que está especialmente diseñada para las pruebas funcionales y que tiene todas las características de sfBrowser, además de algunos métodos muy útiles para los asertos.

3.4.2 Prueba de caja negra o funcional.

Las pruebas de caja negra, también denominadas pruebas de comportamiento, se concentran en los requisitos funcionales del software. Estas pruebas permiten obtener un conjunto de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un programa. En ellas se ignora la estructura de control, concentrándose en los requisitos funcionales del sistema y ejercitándolos.

La prueba de Caja Negra no es una alternativa a las técnicas de prueba de la Caja Blanca, sino un enfoque complementario que intenta descubrir diferentes tipos de errores a los encontrados en los métodos de la Caja Blanca.

Para preparar los casos de pruebas hacen falta un número de datos que ayuden en su ejecución y que permitan que el sistema se ejecute en todas sus variantes, pueden ser datos válidos o inválidos para el programa según si lo que se desea es hallar un error o probar una funcionalidad. Los datos se escogen atendiendo a las especificaciones del problema, sin importar los detalles internos del programa, a fin de verificar que el programa corra bien.

Para desarrollar la prueba de caja negra existen varias técnicas, entre ellas están:

Técnica de la Partición de Equivalencia: esta técnica divide el campo de entrada en clases de datos que tienden a ejercitar determinadas funciones del software.

Técnica del Análisis de Valores Límites: esta técnica prueba la habilidad del programa para manejar datos que se encuentran en los límites aceptables.

Técnica de Grafos de Causa-Efecto: es una técnica que permite al encargado de la prueba validar complejos conjuntos de acciones y condiciones.

Dentro del método de Caja Negra la técnica de la Partición de Equivalencia es una de las más efectivas pues permite examinar los valores válidos e inválidos de las entradas existentes en el software, descubre de forma inmediata una clase de errores que, de otro modo, requerirían la ejecución de muchos casos antes de detectar el error genérico. La partición equivalente se dirige a la definición de casos de pruebas que descubran clases de errores, reduciendo así en número de clases de prueba que hay que desarrollar (16).

Tabla 6 Pruebas de Caso de Uso Administrar Reportes.

Nombre del Caso de Uso	Nombre de Sección	Escenario.	Descripción.	Respuesta del sistema.	Resultado.
Administrar Reportes	SC1: Generar Reportes	EC1: El Jefe de Prensa selecciona el tipo de reporte que desea generar y llena los campos que se muestran	El Jefe de Prensa selecciona la opción de reportes, despliega un menú donde se muestran los tipos de reportes que se pueden generar	Muestra los formularios según el criterio de los reportes. El sistema muestra los reportes.	Satisfactorio.
		EC2: El Jefe	El Jefe de	El sistema	Satisfactorio.

		de Prensa deja campos vacíos.	Prensa no completa los campos.	muestra un mensaje Requerido.	
		EC3: EL Jefe de Prensa escoge mal las fechas de inicio y fin.	El Jefe de Prensa selecciona mal las fechas de inicio y de fin de los reportes que se desean generar.	El sistema muestra un mensaje La fecha de fin () debe ser mayor que la fecha de inicio ().	Satisfactorio.
	SC2: Visualizar Reportes.			El sistema muestra los repostes generados.	Satisfactorio.
	SC3: Imprimir Reportes.	EC1: El Jefe de Prensa desea imprimir los reportes.	El Jefe de Prensa selecciona la opción de exportar los reportes a PDF para imprimirlos.	El sistema muestra la opción de exportar los reportes generados a PDF para su impresión.	Satisfactorio.

Para la realización de los casos de pruebas se ejecutaron tres iteraciones en las que se detectaron 25 no conformidades (NC). La primera iteración arrojó 17 NC, la segunda iteración 8 NC y en la tercera no se detectaron NC, por lo que el sistema se encuentra listo para su utilización. Dentro de las principales NC que se detectaron se encuentran, los problemas en la presentación de los datos y en el resultado final principalmente en la búsqueda y los reportes que no devolvían la información completa que fueron detectados en la primera iteración. Se presenta en la Fig.15 un gráfico con el total de NC detectadas por iteración a cada uno de los CU que agrupan los requisitos funcionales.

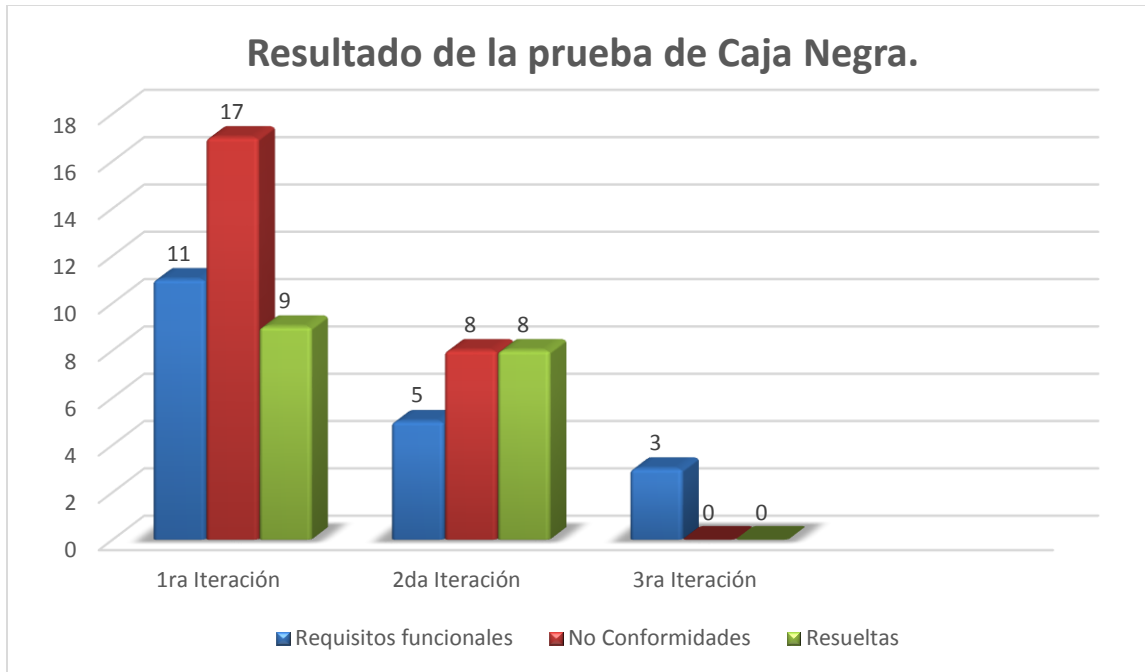


Fig. 15 Resultado de la Prueba de Caja Negra.

3.4.3 Prueba de Integración.

Se aplicaron pruebas de integración con el objetivo de probar si el Subsistema de Administración y el Subsistema de Transmisión de PRIMICIA v1.7 interactúan correctamente. Las pruebas de integración parten de los componentes individuales previamente probados y tienen como objetivo descubrir errores que se pueden producir en la interacción entre los módulos. Consiste en realizar pruebas para verificar que un conjunto de partes de un software funcionan juntos.

En el subsistema de administración para el canal Señal ACN se aplicaron las pruebas de integración incremental ascendente pues a medida en que se implementaba o modificaba algún componente se iban probando. Para la realización de estas pruebas se definieron dos niveles, los cuales se explican a continuación:

Primer nivel: Concluida la implementación del subsistema de administración se comprobó su funcionamiento y se determina si cumple con las funcionalidades correctamente. Se detectaron algunos errores los que fueron corregidos de forma satisfactoria.

Segundo nivel: Luego se realiza la integración del subsistema de administración con el subsistema de transmisión para comprobar que interactúan correctamente. En el proceso se detectaron varios errores como por ejemplo las medias que fueron catalogadas como temporal no se eliminaban una vez transmitida la noticia, las medias agregadas no salían por el canal. Estos errores fueron resueltos satisfactoriamente.

3.5 Conclusiones parciales.

Las tareas de la investigación asociadas a la fase de implementación y prueba permitieron la creación del modelo de implementación, que permitió representar la composición física de los componentes del subsistema de administración. Se identificaron los estándares y normas de codificación, que garantizan la legibilidad y facilitan la comprensión del código fuente de la aplicación. Durante la fase de pruebas de software se definió la aplicación de pruebas de Caja Negra guiadas por casos de prueba, realizando pruebas funcionales haciendo uso de la técnica partición de equivalencia. Las pruebas unitarias que realiza el propio Symfony y las pruebas de integración con el objetivo de probar si el Subsistema de Administración y el Subsistema de Transmisión de PRIMICIA v1.7 interactúan correctamente. Los resultados de las pruebas aplicadas demostraron que la solución cumple con calidad las especificaciones del cliente y que no contiene errores.

CONCLUSIONES GENERALES.

Los resultados obtenidos en el presente trabajo de diploma, dan cumplimiento satisfactorio a los objetivos generales de la investigación. Realizando de forma disciplinada cada una de las tareas propuestas para dar solución al problema planteado. De este modo se puede concluir que:

1. Con la utilización de las herramientas y metodología seleccionada se logró realizar y documentar todo el proceso de desarrollo del software; concluyendo que la utilización de las mismas fue la adecuada para lograr obtener un producto final que cumple con todas las especificaciones planteadas por el cliente.
2. La selección del marco de trabajo Symfony y el uso del patrón MVC permitió tener una correcta estructura para los elementos de software, simplificar la complejidad del mismo y tener una organización de los elementos.
3. La implementación del Subsistema de Administración para el canal Señal ACN cumple con las especificaciones trazadas, lo que se demostró a través de los resultados de las pruebas realizadas. Logrando así solucionar el problema que dio origen a la presente investigación y de este modo se cumplió con el objetivo general planteado.

De forma general se puede concluir que se cumplieron satisfactoriamente todos objetivos enunciados desde el inicio de la investigación práctica, por lo que se le dio solución al problema planteado.

RECOMENDACIONES.

Aún con el cumplimiento del objetivo y de los RF de la aplicación, se necesita incorporar un número de recomendaciones que se consideran fundamentales dentro de las que se encuentran:

- Realizar un estudio de posibles mejoras para facilitar la adaptabilidad del sistema a las necesidades de diferentes clientes.
- La implantación de la aplicación en diferentes instituciones nacionales o internacionales.
- Optimizar la codificación de la aplicación y actualizar la versión del framework empleado.

REFERENCIAS BIBLIOGRÁFICAS.

1. Subs1. [En línea] <http://www.alegsa.com.ar/Dic/subsistema.php>.
2. Rodríguez, Ernesto José Zúñigas. Diseño de interfaces de interacción externa para la Plataforma de Televisión Informativa PRIMICIA. Ciudad de la Habana : Universidad de las Ciencias Informáticas, 2010.
3. SPN. [En línea] <http://ocwus.us.es/periodismoproduccion-periodisticaasigppapartadosapartado6-10.html>.
4. Pérez, Kirenia Gómez. Subsistema de administración para PRIMICIA v2.0. Ciudad de la Habana : Universidad de las Ciencias Informáticas, 2013.
5. Ruber García Hernández, Yunior Hernandez Montaner. Plataforma de televisión Informativa. Ciudad de la Habana : Universidad de las Ciencias Informaticas, 2008.
6. VSN. [En línea] <http://www.vsn-tv.com/es/solucion/1/News-and-Live-production.html>.
7. Enewsroom. [En línea] <http://www.estructuretv.com/enewsroom.html>.
8. González, Carlos de Jesús Andrés. Diseño del Subsistema de Configuración de la Plataforma de Televisión Informativa, PRIMICIA. Ciudad de la Habana : Universidad de las Ciencias Informáticas, 2010.
9. Ivar Jacobson, Gardy Booch y James Rumbaugh. El Proceso Unificado de Desarrollo de Software. Madrid : s.n., 2000.
10. Delgado, LisandraCabrera. Diseño de las nuevas funcionalidades del Módulo de Redacción de la Plataforma de Televisión Informativa PRIMICIA. Ciudad de la Habana : Universidad de las Ciencias Informáticas, 2010.
11. Librosweb.es. LIBROSWEB. [En línea] [Citado el: 3 de Octubre de 2014.] <http://librosweb.es/javascript/>.
12. PHP. [En línea] <http://redgrafica.com/El-lenguaje-de-programacion-PHP>.
13. Netbeans. [En línea] <http://www.genbetadev.com/herramientas/netbeans-1>.
14. Modelado de Negocio. Software, Grupo de Ingenieria de. s.l. : Universidad de Sevilla, 2006.
15. Flujo de trabajo de requisitos. Ciudad de La Habana : Universidad de las Ciencias Informáticas, 2007.
16. Pressman, Roger S. Ingenieria de Software. Un enfoque práctico. Sexta edición. La Habana : Felix Varela, 2005.
17. Larman, Craig. UML y Patrones.Introduccion al análisis y diseño orientado a objetos. s.l. : Prentice Hal, 1998.
18. Symfony la guia definitiva. [En línea] http://www.librosweb.es/symfony_1_1.

19. Sommerville, Ian. Ingeniería de Software. Séptima Edición. 2005. 84-7829-074-5.
20. Rodríguez, Yaquelin Y. Morales y Remedio, Adisneisy C. Román. Propuesta de una Guía para estandarizar la codificación en la Universidad. Ciudad de la Habana : s.n., 2007.
21. Villar, Heikel Matos. Implementación de los subsistemas de Administración y Transmisión de la Plataforma de Televisión Informativa PRIMICIA. Ciudad de la Habana : Universidad de las Ciencias Informáticas, 2008.
22. Pérez, Daylin Grave de Peralta. Propuesta de Migración a Software Libre de Señal ACN. Ciudad de la Habana : Universidad de las Ciencias Informáticas, 2000.
23. AppWeb. [En línea] <http://www.alegsa.com.ar/Dic/aplicacion%20web.php>.
24. [En línea] <https://espanol.answers.yahoo.com/question/index?qid=20070611183202AA0B28L>.
25. Somerville, Ian. Ingeniería de Software. Séptima Edición. s.l. : 2005. 84-7829-074-5.
26. Salazar, Pedro Román. Observatorio Tecnológico. [En línea] 19 de Enero de 2011. [Citado el: 8 de Octubre de 2014.] <http://recursostic.educacion.es/observatorio/web/es/equipamiento-tecnologico/hardware/954-libros-electronicos-ebooks->.
27. DeConceptos.com. DeConceptos. [En línea] 4 de Noviembre de 2012. [Citado el: 8 de Octubre de 2014.] <http://deconceptos.com/tecnologia/manual-de-usuario>.
28. Symfony. Symfony. [En línea] 2013. [Citado el: 20 de Octubre de 2014.] <http://symfony.es/que-es-symfony..>
29. Estudioteca. Estudioteca. [En línea] 2012. [Citado el: 20 de Octubre de 2014.] [http://www.estudioteca.net/universidad/telecomunicaciones/gestor-base-datos/..](http://www.estudioteca.net/universidad/telecomunicaciones/gestor-base-datos/)
30. Programacinweb.net. Programacinweb.net. [En línea] 2009. [Citado el: 12 de Octubre de 2014.] <http://www.programacionweb.net/articulos/articulo/?num=686>.
31. Scribd. Scribd. [En línea] 2011. [Citado el: 20 de Octubre de 2014.] <http://www.scribd.com/doc/3062020/Capitulo-I-HERRAMIENTAS-CASE>.
32. Robohelp. Adobe Systems. [En línea] [Citado el: 20 de Octubre de 2014.] <http://www.adobe.com/es/products/robohelp>.
33. CKEditor. CKEditor. [En línea] 2013. [Citado el: 20 de Octubre de 2014.] <http://www.ckeditor.org>.
34. Patrones de Diseño. [En línea] [Citado el: 20 de Octubre de 2014.] <http://www.inf.utfsm.cl/~visconti/ili236/Documentos/08-Patrones.pdf..>
35. Ptronos. [En línea] [Citado el: 25 de Octubre de 2014.] <http://www.info-ab.uclm.es/asignaturas/42579/pdf/04-pitulo4a.pdf>.

36. Eclipse.org. OpenUP. [En línea] Eclipse, 2012. [Citado el: 3 de Octubre de 2014.] <http://epf.eclipse.org/wikis/openup/>.
37. Object Management Group. Object Management Group. [En línea] [Citado el: 3 de Octubre de 2014.] <http://www.uml.org/>.
38. NetBeans.org. NetBeans. [En línea] [Citado el: 3 de Octubre de 2014.] <https://netbeans.org/>.
39. PostgreSQL. PostgreSQL. [En línea] [Citado el: 3 de Octubre de 2014.] <http://www.postgresql.org/about/>.
40. Sencha. Sencha. [En línea] Sencha. [Citado el: 3 de Octubre de 2014.] <http://www.sencha.com/products/extjs/>.
41. —. Sencha Docs. [En línea] Sencha. [Citado el: 3 de Octubre de 2014.] docs.sencha.com/extjs/4.2.0/.
42. RENA. [En línea] 2008. [Citado el: 3 de Diciembre de 2014.] <http://www.rena.edu.ve/cuartaEtapa/Informatica/Tema13.html>.
43. Dr.Explain. Dr.Explain. [En línea] Indigo Byte Systems. [Citado el: 3 de Diciembre de 2014.] <http://www.drexplain.es/>.
44. Sigil. Sigil. [En línea] [Citado el: 3 de Diciembre de 2014.] <https://code.google.com/p/sigil/>.
45. Larman, Craig. UML y Patrones. México : Prentice Hall, 1999. ISBN: 970-17-0261-1.
46. Buschmann, Frank, y otros. Pattern Oriented Software Architecture. s.l. : Wiley, 1999. 0 471 95889 7.
47. Agile Modeling. [En línea] Ambysoft Inc. [Citado el: 6 de Marzo de 2015.] <http://www.agilemodeling.com/artifacts/useCaseDiagram.htm>.
48. Nielsen, Jakob. Usability Engineering. San Francisco : Morgan Kaufmann. ISBN 012-5184069.

GLOSARIO DE TÉRMINOS.

CSS:

Lenguaje de hojas de estilos creado para controlar la presentación de los documentos electrónicos definidos con HTML y XHTML. CSS es la mejor forma de separar los contenidos y su presentación y es imprescindible para la creación de páginas web complejas.

JavaScript:

JavaScript es un lenguaje de programación que se utiliza principalmente para crear páginas web dinámicas.

TIC:

Tecnologías de la Información y la Comunicación, son todos los servicios, software y hardware que interconectados contribuirán a mejorar las condiciones de vida de las personas.

Televisión:

Transmisión de imágenes a distancia mediante ondas.

IDE:

Entorno de Desarrollo Integrado. Es un programa compuesto por un conjunto de herramientas para facilitar la interacción del programador con el lenguaje en cuestión.

Aplicación web:

Estas son especificaciones de las aplicaciones informáticas. Tiene como característica especial que los usuarios pueden utilizarla accediendo a un servidor web a través de internet o una intranet mediante un navegador. Es una aplicación que se crea y codifica empleando lenguajes soportados e interpretados por los navegadores web.

Base Datos:

Una base de datos o banco de datos (en ocasiones abreviada con la sigla BD o con la abreviatura b. d) es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso. En la actualidad, y debido al desarrollo tecnológico de campos como la informática y la electrónica, la mayoría de las bases de datos están en formato digital (electrónico), que ofrece un amplio rango de soluciones al problema de almacenar datos.