

**Universidad de las Ciencias Informáticas
Facultad 6**



Título: Subsistema de administración para PRIMICIA v2.0

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

Autores:

Kirenia Gómez Pérez

Nivaldo Perdomo Delgado

Tutor:

Ing. Dorgis Montes de Oca Gómez

Junio de 2013



"Sólo existe un sentimiento mayor que el amor a la libertad: el odio al que te la quita".

"Ernesto Che Guevara".

Declaración de Autoría

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

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

Kirenia Gómez Pérez

Ing. Dorgis Montes de Oca Gómez

Firma del Autor

Firma del Tutor



Declaración de Autoría

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

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

Nivaldo Perdomo Delgado

Firma del Autor

Ing. Dorgis Montes de Oca Gómez

Firma del Tutor

Datos de Contacto

Tutor: Ing. Dorgis Montes de Oca Gómez

Centro de trabajo: Universidad de las Ciencias Informáticas

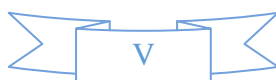
Título de la especialidad de graduado: Ingeniero en Ciencias Informáticas

Año de graduación: 2012

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

Correo electrónico: dorgis@uci.cu

Teléfono particular: 835 8900



Agradecimientos

Kirenia

Agradezco de todo corazón:

A mis padres Susana y Roberto por ser el principal motivo de inspiración en la realización de este trabajo, porque sin ellos no estaría hoy aquí. Por ofrecerme todo su apoyo y confiar plenamente en mí, por las frases de aliento en los momentos más duros y todos los años de sacrificios para que su niñita tuviera siempre lo mejor. Por las llamadas diarias que tanto bien me hicieron. Porque sé que han sufrido todo este tiempo mi ausencia en casa. Gracias a ellos he logrado todo en mi vida.

A mi Tata por ser mi segundo y más recto papa. Por exigirme buenos resultados y guiarme por el camino que debo transitar. Por cuidar de mis padres en mi ausencia. Por quererme tanto y ser mi apoyo y motivo para seguir adelante. Por entregar su vida y sacrificio diario en favor de mis padres y de mí. Tata; quiero que siempre estés orgulloso de tu hermana que tanto te quiere.

A Yordan, por llenar mi vida de tanto amor. Porque fue capaz de abandonar muchas cosas por mí. Por aguantar mi estrés y malacrianza durante tanto tiempo. Por brindarme un hombro donde llorar cuando algo me salía mal y donde reír en momentos de alegría. Por todas las noches que dedicó para ayudarme en los primeros y más difíciles años de la carrera. Por cuidarme tan bien como lo hace. Gracias por existir.

A mi suegra y cuñada por toda su preocupación y cariño.

A Raisa por su preocupación y apoyo.

A toda la familia que se ha preocupado por mí, en especial a mis tíos que se encuentran hoy acompañándome.

A mi compañero de tesis por su interés y responsabilidad en todo momento.

A mis amigos de estos cinco años.

A Anay por su amistad. Por estar a mi lado en cada momento. Por las veces que me ha brindado su apoyo incondicional. Por saber que puedo contar con ella sin importar lo difícil que sea la situación.

A mi tutor Dorgis por toda la ayuda y dedicación brindada. Por defendernos en todo momento y compartir cada minuto de tensión.

A Reynaldo por los días y noches dedicados a revisarme la tesis. Por toda su ayuda y apoyo.

A Yareisi por las ideas aportadas en la investigación.

A los miembros del tribunal y oponente por haber puesto todo su empeño y dedicación en perfeccionar esta investigación hasta convertirla en lo que es hoy.

A los amigos de toda la vida por apoyarme de una forma u otra en cada momento de la carrera en especial a Yisel, Heilen, Anisleisy, Javier y Yunita.

A todos los profesores que me acompañaron durante la carrera.

A todos los que me preguntaron algún día por la tesis y han contribuido en su desarrollo.

Nivaldo

Agradezco:

A mis padres y hermano por hacer posible este sueño, por conducirme siempre por el sendero correcto y más seguro, por darme fuerzas para continuar adelante en los momentos difíciles y por brindarme tanto amor y cariño.

A toda mi familia por el apoyo brindado, en especial para Isidra, Héctor y Mary.

A Yanela por su amor y cariño en todos estos años y por estar siempre a mi lado.

A Dorgis por toda su ayuda y dedicación.

A mi compañera de tesis por todo su apoyo y porque sin ella hubiese sido imposible cumplir este trabajo.

A los amigos que han contribuido de una forma u otra a sobrellevar estos 5 años de la carrera, en particular para aquellos que no pudieron terminar por diferentes causas.

Al oponente Félix por su ayuda y paciencia.

A los miembros del tribunal por perfeccionar esta investigación.

Dedicatoria

Kirenia

Dedico el resultado de esta investigación, con mucho amor:

A mis padres y hermano, que son el mayor amor de mi vida, por ser mi inspiración, porque todo lo que soy se lo debo a ustedes, por su ayuda y confianza en todo momento. Por permitirme estudiar lejos aun cuando me quieren cerca. Aquí está el fruto de su sacrificio y amor inmenso.

A mi novio Yordan, por llegar a mi vida y llenarla de amor y cariño, por estar a mi lado durante 5 años apoyándome en todas mis decisiones, porque te adoro mi pipi.

A todos los que estuvieron y están a mi lado.

Nivaldo

Dedico el resultado de la presente investigación:

A mi mamá Martha por ser la persona más importante en mi vida.

A mi papá Nelson por contar con su apoyo en todo momento.

A la memoria de mi abuela Meña por ser mi fuente de inspiración.

Resumen

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

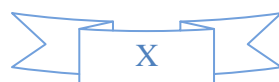
Actualmente PRIMICIA va camino a su segunda versión; el objetivo del presente trabajo es desarrollar un Subsistema de Administración con tecnologías de escritorio que permita la redacción y gestión de medias para PRIMICIA v2.0. El mismo se encuentra estructurado en tres capítulos que recogen las diferentes 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 PRIMICIA v2.0.

Palabras Claves:

Administración, PRIMICIA, subsistema.

Índice General

INTRODUCCIÓN	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.....	5
1.1 INTRODUCCIÓN	5
1.2 TÉRMINOS ASOCIADOS AL PROBLEMA	5
1.2.1 Sistema	5
1.2.2 Subsistema	5
1.2.3 Administración.....	6
1.2.4 Módulo	6
1.3 ANÁLISIS DEL OBJETO DE ESTUDIO	7
1.3.1 Descripción general.....	7
1.3.1.1 Procesos de administración en Enewsroom	8
1.3.1.2 Procesos de administración en Señal ACN	9
1.3.1.3 Procesos de administración en TV Energía	10
1.3.1.4 Procesos de administración en PRIMICIA v1.7.....	10
1.3.1.4.1 Proceso de redacción de noticia en PRIMICIA v1.7	12
1.3.1.4.2 Proceso de gestión de medias en PRIMICIA v1.7.....	12
1.4 SOLUCIONES SIMILARES A PRIMICIA	13
1.4.1 Enewsroom	13
1.4.2 Sonaps.....	14
1.4.3 Señal ACN	14
1.4.4 TV Energía	14
1.5 METODOLOGÍAS, HERRAMIENTAS Y TECNOLOGÍAS A UTILIZAR PARA EL DESARROLLO DE LA SOLUCIÓN	15
1.5.1 Metodología de desarrollo de software RUP	15
1.5.2 Lenguaje Unificado de Modelado (UML) como soporte de la modelación de la solución propuesta.....	16
1.5.3 Herramienta CASE Visual Paradigm	17
1.5.4 Lenguaje de programación.....	18
1.5.4.1 Lenguaje C++	18



1.5.5 Marco de trabajo	19
1.5.5.1 Qt	19
1.5.6 Entorno integrado de desarrollo	19
1.5.6.1 Qt Creator.....	20
1.5.7 Sistema gestor de base de datos (BD)	20
1.5.7.1 PostgreSQL.....	20
1.6 CONCLUSIONES	21
CAPÍTULO 2: PRESENTACIÓN DE LA SOLUCIÓN PROPUESTA.....	22
2.1 INTRODUCCIÓN	22
2.2 MODELO DEL DOMINIO	22
2.2.1 Eventos principales del entorno.....	23
2.2.2 Diagrama de Clases del Dominio	23
2.2.3 Glosario de Términos del Dominio.....	24
2.3 REQUISITOS.....	24
2.3.1 Requisitos Funcionales	25
2.3.2 Requisitos no Funcionales	26
2.3.2.1 Requisitos de Software.....	26
2.3.2.2 Requisitos de Hardware recomendados	26
2.3.2.3 Requisitos de restricciones en el diseño y la implementación.....	27
2.3.2.4 Requisitos de apariencia e interfaz externa	27
2.3.2.5 Requisitos de Seguridad.....	27
2.4 MODELADO DEL SISTEMA	27
2.4.1 Descripción de los actores del sistema.....	27
2.4.2 Diagrama de casos de uso del sistema	28
2.5 DESCRIPCIÓN TEXTUAL DE LOS CASOS DE USO DEL SISTEMA	28
2.6 CONCLUSIONES	41
CAPÍTULO 3: CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA.....	42
3.1 INTRODUCCIÓN	42
3.2 MODELO DE ANÁLISIS	42
3.3 DESCRIPCIÓN DE LA ARQUITECTURA DE SOFTWARE.....	42

3.3.1 Patrón arquitectónico.....	43
3.3.1.1 Modelo en Capas	43
3.3.2 Patrones de Diseño	44
3.3.2.1 Patrones Generales de Software para Asignar Responsabilidades	44
3.3.2.2 Pandilla de los Cuatro.....	45
3.4 MODELO DE DISEÑO.....	45
3.5 MODELO DE DESPLIEGUE	49
3.6 MODELO DE IMPLEMENTACIÓN	49
3.6.1 Estándar de codificación	51
3.7 VALIDACIÓN DEL SISTEMA	51
3.7.1 Pruebas de software.....	51
3.7.2 Niveles de Prueba	52
3.7.3 Diseño de Casos de Prueba.....	52
3.7.4 Pruebas de Rendimiento	60
3.8 CONCLUSIONES	62
CONCLUSIONES GENERALES.....	63
RECOMENDACIONES	64
BIBLIOGRAFÍA Y REFERENCIAS BIBLIOGRÁFICAS	65
GLOSARIO	67

Índice de Figuras y Tablas

Figura 1. Diagrama de Clases del Modelo de Dominio.	23
Figura 2. Diagrama de Casos de Uso del Sistema.	28
Figura 3. Diagrama de Clases del Diseño del Caso de Uso: Crear Noticia.	46
Figura 4. Diagrama de Clases del Diseño del Caso de Uso: Administrar Media.	47
Figura 5. Diagrama de Clases del Diseño del Caso de Uso: Administrar Pantalla.	48
Figura 6. Modelo de Despliegue.	49
Figura 8. Diagrama de Componentes.	50
Tabla 1. Descripción Textual del Caso de Uso: Administrar Medias.	29
Tabla 2. Descripción Textual del Caso de Uso: Crear Noticia.	34
Tabla 3. Descripción Textual del Caso de Uso: Administrar Pantalla.	36
Tabla 4. Casos de Prueba de Integración.	53
Tabla 5. Pruebas del Caso de Uso: Administar Medias.	54
Tabla 6. Pruebas del Caso de Uso: Administar Pantalla.	57
Tabla 7. Pruebas del Caso de Uso: Crear Noticia.	60
Tabla 8. Tiempo de Respuesta para el Subsistema de Administración Web y en el Subsistema de Administración de Escritorio.	61

Introducción

A lo largo de la historia se ha apreciado la necesidad progresiva de perfeccionar las formas de comunicación. El acontecer tecnológico, ha venido aparejado con innumerables avances, que han transformado la humanidad, principalmente el uso de las Tecnologías de la Información y las Comunicaciones (TIC). Desde el año 1950, la televisión se ha convertido en uno de los medios de comunicación de mayor importancia en el desarrollo del hombre, al tener la posibilidad de vincular a las informaciones el poder seductor de la imagen y del sonido. Hoy día varias instituciones como aeropuertos, terminales y hoteles necesitan que se muestre información a un gran número de personas utilizando para ello sus redes de televisión interna.

Para realizar esta tarea es necesario que se cuente con todo un sistema informático que permita la automatización de los procesos de confección y visualización de estas informaciones. Es por ello que en el 2005, comienza el trabajo por parte de las entidades nacionales del Ministerio de la Informática y las Telecomunicaciones para montar y poner en marcha la Plataforma de Televisión Digital Satelital Cubana. De esta forma se logra captar las señales de la televisión y la radio en todo el territorio nacional incluyendo las que se encuentran fuera del mismo **(1)**.

La existencia en Cuba de la primera plataforma para la transmisión de radio y televisión directa vía satélite permite eliminar las llamadas zonas de silencio¹. Mediante la instalación de receptores satelitales y antenas parabólicas en centros y entidades sociales, se logra la recepción de las señales a la población de esas regiones, donde viven hoy alrededor de 70 mil cubanos. Además se han instalado cientos de estos equipos en los centros médicos cubanos en el exterior, así como en las residencias de los cooperantes, lo cual posibilita la recepción de los canales Cuba Visión, Cuba Visión Internacional, Tele Rebelde, Canal Educativo, Canal Educativo 2, Canal Habana, Radio Rebelde, Radio Progreso, Radio Taíno, Radio Habana Cuba y Radio Reloj **(2)**.

La Universidad de las Ciencias Informáticas (UCI), ha contribuido con el desarrollo de plataformas informativas en el país, en la actualidad el desarrollo de estas plataformas se lleva a cabo en el Centro de

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

Geoinformática y Señales Digitales (GEYSED) de esta institución. Entre las soluciones obtenidas en la UCI están: Señal3, Señal ACN, TV Energía y PRIMICIA. Señal3 es una plataforma dirigida a mantener informada a toda la comunidad universitaria. Señal ACN fue creada para la Agencia Cubana de Noticias con el propósito de difundir la información con un mayor alcance; con esta se mantiene informados del acontecer nacional e internacional a los cooperantes en países como Venezuela, Haití y Bolivia; así como a las llamadas zonas de silencio del archipiélago cubano. Posteriormente surge TV Energía, desarrollada sobre un sistema operativo libre, GNU/Linux con el fin de automatizar los procesos que se realizan para visualizar la información en la red de televisión instalada en el Ministerio del Poder Popular para la Energía y Petróleo de Venezuela (MENPET).

Finalmente, en el departamento Señales Digitales del Centro GEYSED se desarrolló el producto PRIMICIA, el que se ha implementado con herramientas y tecnologías libres. Esta es una solución integral capaz de proveer un canal de televisión que puede transmitir información en varios formatos constantemente y puede ser ajustable a cualquier cliente dado sus necesidades e intereses. La plataforma se encuentra estructurada en dos subsistemas que se relacionan entre sí y actúan como un todo para brindar un resultado final a través de una red de televisión, estos son: el Subsistema de Administración y el Subsistema de Transmisión. El Subsistema de Administración es el responsable de realizar la administración del canal y toda la gestión de las noticias y recursos multimedia; y el Subsistema de Transmisión es el encargado de visualizar las noticias y materiales publicados.

El Subsistema de Administración se encuentra desarrollado con tecnologías web. El autor Pressman define en el capítulo dieciséis “Ingeniería Web” de su sexta edición; que en la mayoría de las Aplicaciones Web se encuentran los siguientes atributos, a continuación una cita textual de algunos de ellos **(3)**:

Concurrencia: un gran número de usuarios pueden tener acceso a la Web App ² al mismo tiempo. En muchos casos, los patrones de uso entre los usuarios finales variarán enormemente.

Carga Impredecible: el número de usuarios de la Web App puede variar en órdenes de magnitud cada día. El lunes pueden mostrarse 100 usuarios; el martes pueden usar el sistema 10 000.

Gobernada por los Datos: la función primordial de muchas Web App es usar hipermedia para presentar

² Web App: Hace referencia a una Aplicación Web. Desde una simple página web hasta un amplio sitio web.

contenido de texto, gráficos, audio y video al usuario final. Por lo general, las Web Apps se utilizan para tener acceso a información que existe en bases de datos.

Teniendo en cuenta los atributos mencionados se determina que el Subsistema de Administración que posee la plataforma no constituye una solución recomendable para escenarios donde el número de usuarios que interactúan con la aplicación es reducido y su magnitud no varía con frecuencia. Dicho subsistema requiere de una comunicación constante con el servidor que ejecuta la aplicación el que en ocasiones no presenta las prestaciones requeridas para ejecutarla de manera fluida. Además es importante contar con una buena conexión a internet lo que para muchos clientes no es posible.

Al mismo tiempo presenta un bajo rendimiento debido al manejo de grandes flujos de datos procedentes de una base de datos, como consecuencia se obtiene un tiempo de respuesta lento en la confección de la noticia y una menor velocidad de procesamiento. Tales condiciones traen como resultado que los procesos sean ineficientes y que no siempre se cuente con la inmediatez que se necesita; afectando de esta forma a clientes potenciales lo que ha permitido identificar la necesidad de alterar el Subsistema de Administración con la meta de perfeccionar el mismo.

Por las insuficiencias planteadas anteriormente en la **situación problemática** se formula el siguiente **problema a resolver**: ¿Cómo obtener un mayor rendimiento en el Subsistema de Administración de la Plataforma de Televisión Informativa PRIMICIA v2.0? Partiendo del problema planteado se define como **objeto de estudio** los procesos de administración para Plataformas de Televisión Informativas, en un **campo de acción** centrado a la informatización de los procesos de redacción y gestión de medias del Subsistema de Administración de PRIMICIA v2.0. Para solucionar dicho problema se tiene como **objetivo general**: desarrollar un Subsistema de Administración con tecnologías de escritorio que permita la redacción y gestión de medias para PRIMICIA v2.0.

Defendiendo la Idea de que desarrollándose un Subsistema de Administración con tecnologías de escritorio que permita la redacción y gestión de medias para PRIMICIA v2.0 se logrará un mayor rendimiento del sistema.

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

1. Caracterización de las soluciones similares en el ámbito nacional e internacional.
2. Caracterización del uso de los lenguajes de programación, herramientas y tecnologías a utilizar.
3. Realización del proceso de desarrollo del software teniendo en cuenta la metodología seleccionada.

4. Implementación del Subsistema de Administración.
5. Generación de los artefactos correspondientes de acuerdo a la metodología escogida.
6. Realización de pruebas para validar el sistema.

Con la finalidad de adquirir conocimientos que permitan el cumplimiento de los objetivos propuestos en el desarrollo de la investigación se utilizan los siguientes **métodos científicos**:

Métodos teóricos:

Análítico-Sintético: con el objetivo de valorar y definir las herramientas y las plataformas necesarias para el desarrollo del sistema.

Modelación: se utiliza en el proceso de desarrollo del software en el cual se crean un grupo de modelos que describen los procesos contribuyendo al desarrollo del sistema.

Métodos empíricos:

Observación: se utiliza la observación de tipo libre; 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 actualiza el estado del arte 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 a 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. Presentación de la Solución Propuesta: se seleccionan los Requisitos 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.

Capítulo 3. Construcción de la Solución Propuesta: se elabora la solución propuesta para el problema en cuestión. Se explica la arquitectura en Tres Capas y los patrones de diseño utilizados. Por último se valida el producto a través de las pruebas de caja negra, de integración y rendimiento.

Capítulo 1: Fundamentación Teórica

1.1 Introducción

En el presente capítulo se exponen los conceptos y elementos que constituyen la base teórica del trabajo de diploma, lo que facilita una mejor comprensión de la presente investigación; además se hace una caracterización de los procesos de administración para plataformas de televisión informativas. Se realiza un análisis de las principales plataformas de televisión informativas existentes en el mundo y sus procesos de administración haciendo un mayor énfasis en los procesos de redacción y gestión de medias, lo que permitió identificar características que tiene la solución que se propone. Por último se determina la metodología, herramientas, lenguajes y tecnologías a utilizar para el desarrollo de la solución propuesta.

1.2 Términos asociados al problema

1.2.1 Sistema

Según el Diccionario de la Real Academia de la Lengua Española (DRAE) el término sistema es un conjunto de reglas o principios sobre una materia racionalmente enlazados entre sí. Conjunto de cosas que relacionadas entre sí ordenadamente contribuyen a determinado objeto **(4)**.

Es posible entonces decir que un sistema informático es el conjunto de componentes de software y hardware interrelacionados que permiten la interacción con los usuarios para lograr un objetivo común. Durante la concepción de un sistema informático es importante tener en cuenta que las partes interconectadas funcionan como un todo, cualquier modificación de los componentes afecta el correcto funcionamiento del sistema propiamente dicho, por lo que también hay que tener claro la disposición de los nuevos elementos y sus funciones **(5)**.

Teniendo en cuenta lo planteado se puede definir un sistema como una serie de elementos relacionados entre sí, con el fin de lograr un propósito de valor común para todos. En el caso de los sistemas informáticos las partes que intervienen actúan como un todo por lo que cualquier cambio o modificación que se realice afecta al sistema en su totalidad.

1.2.2 Subsistema

El DRAE no recoge definiciones en su última edición para esta palabra; sin embargo es ya muy común en todos los ámbitos informáticos. Un subsistema es una forma de encapsulamiento, organización y

agrupación de elementos, ficheros y clases. Según la metodología de desarrollo de software *Rational Unified Process* (RUP), subsistema es un “Elemento de modelo que tiene la semántica de un paquete, ya que puede contener otros elementos de modelo, y una clase, ya que tiene comportamiento (...) Un subsistema ejecuta una o más interfaces, que defina el comportamiento que puede llevar a cabo” **(6)**.

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 relacionadas para cumplir con un propósito o fin determinado y cuyas partes deben reunir ciertas condiciones de tal manera que se complementen formando el sistema.

1.2.3 Administración

De acuerdo al DRAE administración es la acción y efecto de administrar. Empleo de administrador **(4)**.

La administración es un elemento trascendental en la concepción de cualquier sistema informático, se convierte en un componente imprescindible para su adecuado funcionamiento, pues simplifica el trabajo humano proporcionando mayor poder y calidad al producto. Un sistema informático debe proporcionar al encargado de la administración del mismo las siguientes funcionalidades administrativas **(7)**:

- Instalar y configurar el nuevo hardware y/o software.
- Agregar, borrar, modificar información de las cuentas de usuarios y establecer contraseñas.
- Garantizar la seguridad.
- Configuración del sistema.
- Solucionar problemas del sistema.

Teniendo en cuenta los elementos planteados se puede concluir que la administración consiste en planear, organizar, dirigir y controlar los recursos permitiendo alcanzar los objetivos propuestos, mediante la optimización eficiente y eficaz de los recursos humanos.

1.2.4 Módulo

Según el DRAE el término módulo, en cuestiones de programación se define como sigue:

“Un módulo es un software que agrupa un conjunto de subprogramas y estructuras de datos. Los módulos son unidades que pueden ser compiladas por separado y los hace reusables y permite que múltiples programadores trabajen en diferentes módulos en forma simultánea, produciendo ahorro en los tiempos

de desarrollo. Los módulos promueven la modularidad y el encapsulamiento, pudiendo generar programas complejos de fácil comprensión” (8).

Cada uno de los módulos de un programa, idealmente deberían cumplir las siguientes características (7):

Tamaño pequeño: permite aislar el impacto que pueda tener un cambio en el sistema, ya sea en el momento de corregir un error, o en el rediseño de una parte del código.

Independencia modular: implica comodidad para trabajar con ellos, para desarrollar un módulo no es necesario conocer detalles internos de otros.

La palabra módulo, es muy utilizada en diversas ramas de las ciencias informáticas. En programación un módulo es una parte de un programa, que realizará una o varias tareas del total que debe realizar el programa para cumplir con sus objetivos. Normalmente suelen estar organizados jerárquicamente en niveles, de forma que los módulos de nivel superior realizan las llamadas a los del nivel inferior. Cuando un módulo es llamado, recibe como entrada datos necesarios para realizar la o las tareas programadas; una vez concluidas estas, devuelve la salida al módulo que lo llamó. Un módulo puede llamar a uno o varios módulos de nivel inferior o del mismo nivel, pero nunca a uno de nivel superior.

1.3 Análisis del objeto de estudio

En el presente trabajo se ha definido como **objeto de estudio** los procesos de administración para plataformas de televisión informativas. Para lograr un mejor entendimiento de estos procesos se realiza una descripción de los mismos teniendo en cuenta diferentes plataformas desarrolladas a nivel nacional e internacional.

1.3.1 Descripción general

Todo sistema informático debe contar 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 (9). Los procesos de administración se encargan de gestionar las configuraciones del sistema y de los usuarios que acceden al mismo. Las plataformas de televisión informativas en particular realizan varias funciones para su administración teniendo en cuenta sus características y propósitos (10).

1.3.1.1 Procesos de administración en Enewsroom

Enewsroom cuenta con varios módulos de trabajo como:

Módulos de entrada (11):

- Previsiones: permite a los redactores y equipo técnico trabajar en conjunto para conformar la entrada de información al sistema.
- Captura: captura o digitaliza contenidos asociando los materiales audiovisuales con los metadatos correspondientes en el momento de su entrada en el sistema, clasificándolos para su posterior acceso por parte de los usuarios.

Módulos de organización (11):

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

Módulos de producción (11):

- Editor de noticias: permite realizar ediciones del material audiovisual y del texto de la noticia. Selecciona los materiales desde cualquier bruto³, edita en la línea de tiempo⁴ el video, audio y gráficos mientras se desarrolla el texto de la noticia, la sinopsis, o el minutado⁵. Permite la incorporación de voz en off⁶, diferentes modos de edición, detección de planos y escucha rápida.
- Documentación: documenta noticias y material audiovisual bruto y esta documentación posteriormente es enviada al módulo de archivo.

³ El término bruto se refiere a los materiales audiovisuales que pueden ser utilizados en la edición de informaciones.

⁴ La línea de tiempo permite ordenar una secuencia de eventos de tal forma que se visualice con claridad la relación temporal entre ellos.

⁵ Minutado es distribuir el tiempo correspondiente a una determinada actividad, como programas de televisión y actos públicos.

⁶ Se refiere a la técnica de producción donde se retransmite una voz no pronunciada visualmente delante de la cámara.

Módulos de salida (11):

- Emisión: gestiona la emisión a través de una escaleta dinámica controlando el video-servidor de emisión.
- Archivo: gestiona el archivo del sistema mediante volcados automáticos o manuales a soportes de seguridad (cintas, DVD), además de codificar indexando⁷ de nuevo el material audiovisual con la información contenida en el sistema.

1.3.1.2 Procesos de administración en Señal ACN

El Subsistema de Administración que posee Señal ACN permite la gestión de las noticias y recursos multimedia según los requerimientos de la Agencia Cubana de Noticias. Es la columna vertebral del sistema ya que 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 mediante este subsistema 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. Es una aplicación web desarrollada en Symfony v1.0.5 **(12)**.

El proceso de redacción en ACN se realiza de la siguiente forma **(12)**:

- 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 para la misma.
- Se llena el cuerpo de la noticia creando cada una de las pantallas que la conforman, las cuales pueden ser tipo texto, texto-imagen, imagen y video. Si la pantalla es de tipo imagen o video se le

⁷ Registrar ordenadamente datos e informaciones, para elaborar su índice.

asigna el recurso correspondiente. Cada una de las pantallas tiene una duración específica de acuerdo a la longitud del texto de la misma (10, 15 ó 25 segundos).

- La conformación de la noticia termina cuando el corrector revisa la misma y enmienda cualquier error cometido por el redactor, teniendo la posibilidad de modificar el título, la sección, el tema musical de fondo y las pantallas de la noticia.

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 redactar las mismas.

1.3.1.3 Procesos de administración en TV Energía

El Subsistema de Administración que posee TV Energía también desarrollado con tecnologías web está formado por seis paquetes **(13)**:

Redacción: permite la redacción de las noticias y la gestión de las que no estén publicadas.

Medias: permite la administración del archivo de recursos multimedia, en este paquete se suben al sistema las medias que van a ser utilizadas en la confección de las noticias, se modifican y eliminan de acuerdo a los intereses del MENPET.

Seguridad: controla el acceso de los usuarios al sistema y permite su gestión.

Editorial: permite la gestión de las secciones temáticas y las noticias que ya han sido publicadas, además permite que se administre el archivo de noticias del canal.

Gestión de Señal del Canal: permite toda la gestión de la señal del canal.

Reportes: da la posibilidad de generar reportes que ayuden a controlar la actividad de los redactores.

Cada usuario que interactúe con el sistema debe tener al menos un rol asignado, estos roles determinarán el grado de acceso que tendrán dentro del sistema. Después de analizar el capital humano con que cuenta el MENPET, se determinó que debían existir seis roles que determinarían el acceso a los recursos con los que pueden interactuar los usuarios después que entren al sistema, estos roles son los siguientes: Redactor, Editor, Administrador de Recursos Multimedia, Administrador de Usuario, Operador de Transmisión y Jefe del canal.

1.3.1.4 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 el funcionamiento del mismo 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 de la misma. El módulo de Redacción de Noticias de PRIMICIA v1.7 es el que contiene mayor responsabilidad en el proceso de redacción de la noticia 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 **(14)**:

- **Gestión de los usuarios del sistema:** permite adicionar y eliminar usuarios, así como establecer y modificar los permisos de acceso 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 reutilizar y eliminar las mismas.
- **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 infocinta:** 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 claves 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.

1.3.1.4.1 Proceso de redacción de noticia en PRIMICIA v1.7

Mediante el proceso de redacción de la noticia, es posible redactar y conformar los elementos que se quieren transmitir en determinada entidad que esté usando la plataforma. Cada institución o empresa tienen características particulares por lo que difieren en la forma de transmisión de noticias. Esto conduce a que el proceso de redacción sea diferente en cada entidad **(1)**.

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

1.3.1.4.2 Proceso de gestión de medias en PRIMICIA v1.7

La gestión de medias, es el proceso de almacenar organizadamente los archivos audiovisuales (imagen, video y audio) para controlar toda operación que se realice sobre ellos. Surge por la necesidad de organizar y administrar esta información con el fin de poder localizar cualquier archivo multimedia, conociendo aspectos específicos o generales de él. En el caso del almacenamiento, se debe usar un servidor de ficheros para almacenar las medias y una referencia en la base de datos para luego localizarlas.

Los archivos usados en la plataforma son música (audio), video e imagen, para los cuales se deben insertar varios datos propios de cada fichero, tanto a la hora de adicionar como de buscar un recurso en particular. Además, cuando se realiza una búsqueda entre los recursos, se le posibilita al usuario modificar, eliminar o visualizar el recurso al cual accedió. En caso de querer copiar el archivo se debe conocer el lugar donde se almacena o contactar con el encargado de la labor.

Actualmente PRIMICIA se encuentra en desarrollo de la segunda versión; en la misma el Subsistema de Administración que se propone está desarrollado con tecnologías de escritorio y posee las prestaciones siguientes:

- **Funcionalidades para la redacción de noticias.**
- **Gestión de recursos multimedia.**

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

1.4 Soluciones similares a PRIMICIA

Durante la búsqueda y el estudio de posibles aplicaciones informáticas, que sirvieran como un antecedente para el presente trabajo, se identificaron varios software similares. A continuación se exponen las características significativas de soluciones identificadas en el ámbito internacional y luego se describen las soluciones desarrolladas en Cuba.

1.4.1 Enewsroom

Enewsroom es un sistema de producción de noticias y contenidos audiovisuales, de la plataforma propietaria *Estructure Media Systems*, diseñado con el fin de facilitar el trabajo de periodistas y usuarios. Incorpora potentes motores documentales que agilizan la localización de contenidos en el sistema. Cuenta con Servidores multiformato que son capaces de digitalizar simultáneamente hasta tres calidades de vídeo (alta, baja y ultra baja). Integra las herramientas necesarias para la explotación de noticias y la introducción de flujos de trabajo flexibles que permiten a los usuarios realizar su trabajo con grandes facilidades **(11)**.

Sus características fundamentales son **(11)**:

- Sistema sencillo, intuitivo, robusto y fiable.
- 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.

1.4.2 Sonaps

El sistema Sonaps de Sony es un completo entorno en red que integra todos los aspectos del proceso de producción de informativos, desde la planificación, captación y edición, hasta la emisión, distribución y archivo. El sistema Sonaps transforma el proceso de producción de noticias integrando todos los aspectos del proceso de producción en un completo y sencillo flujo de trabajo, lleva a cabo considerables mejoras, comparadas con las operaciones actuales y permite una inversión controlada mientras maximiza los resultados y la calidad. Gracias al sistema Sonaps, periodistas, editores profesionales y productores pueden compartir contenido y metadatos en un colaborativo flujo de trabajo en red basado en ficheros **(15)**.

Luego del estudio realizado se determina que aunque las soluciones existentes en el ámbito internacional solventan de forma parcial el problema planteado, no constituyen una vía factible ya que se encuentran sujetas al pago de licencia y son desarrolladas con tecnologías web al igual que PRIMICIA. Por lo que es recomendable desarrollar las funcionalidades de la presente investigación sin realizar la integración de estas plataformas con PRIMICIA. Por otra parte, no se dispone de una alta inversión para realizar un producto de igual calidad pues necesitan de tecnologías especializadas con altos precios; sin embargo la plataforma que se propone realiza los procesos de administración sin llegar a generar altos costos por conceptos de equipamiento tecnológico.

1.4.3 Señal ACN

Señal ACN es uno de los principales precedentes de PRIMICIA por lo que se hace necesario su estudio y valoración. Creado por iniciativa del Comandante en Jefe Fidel Castro, para trasladar los contenidos de los periódicos cubanos a los internacionalistas en Venezuela, Belice, Haití, Honduras, Bolivia, Dominica y Jamaica, así como a las llamadas zonas de silencio del archipiélago cubano. El proyecto fue ideado en colaboración con la Universidad de las Ciencias Informáticas **(12)**.

1.4.4 TV Energía

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

sobre tecnologías libres para la transmisión. La División de Prensa del Ministerio transmite un canal de televisión interno con recepción en diferentes locales de su sede que comprende salas de espera de elevadores, comedores y oficinas; donde se colocan algunos materiales informativos y se encadena mayormente con la televisora del gobierno Venezolana de Televisión y un canal que se transmite en la Sede Central de la Compañía Petrolera de Venezuela (PDVSA).

Los sistemas encontrados en el ámbito nacional constituyen los principales antecedentes de PRIMICIA. Luego del análisis realizado se puede concluir que no resuelven el problema que dio origen a la presente investigación pues se encuentran desarrollados con tecnologías web al igual que dicha plataforma. Es importante reconocer que los procesos de negocio que implementan estas soluciones son sumamente significativos y de gran utilidad para resolver la problemática que se plantea.

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

Las formas de trabajo organizadas constituyen las bases de creación de un software, por lo que se debe contar con un proceso que ordene e integre las etapas de desarrollo del mismo. En el presente epígrafe se realiza un estudio para definir la metodología de desarrollo de software a aplicar, lo que constituye un marco de trabajo que es usado para estructurar, planear y controlar el proceso de desarrollo en sistemas de información. Se analizan las tecnologías, lenguajes de programación, Sistemas de Gestión de Base de Datos (SGBD) y *framework* de desarrollo apropiados teniendo en cuenta las características de la aplicación.

1.5.1 Metodología de desarrollo de software RUP⁸

En muchas ocasiones resulta complejo definir cómo se realizará el proceso de desarrollo de un software. Esto se debe a la necesidad de crear sistemas sofisticados y mejor adaptados a las nuevas tecnologías y a las necesidades de los usuarios. La elección de una metodología depende principalmente de las características particulares del proyecto y de la experiencia que tenga el equipo de desarrollo en el uso de ésta. RUP es una metodología flexible que puede ser adaptada a las necesidades de cada proyecto. Centra su atención en llevar la documentación del proyecto y cumplir con el plan definido para éste.

⁸ RUP: Rational Unified Process.

Constituye una guía que permite definir un conjunto de actividades indispensables para convertir los Requisitos Funcionales (qué debe hacer el sistema) del usuario en un conjunto de elementos necesarios en la construcción de productos **(16)**.

RUP se caracteriza esencialmente porque el proceso de desarrollo del software es: iterativo e incremental, dirigido por casos de uso y centrado en la arquitectura. Para mejor entendimiento de estas características es preciso mencionar que RUP se propone una estrategia de desarrollo donde se posibilite la realización de mini proyectos, los que se denominan iteraciones. Cada iteración proporciona un resultado observable y un incremento con relación a la iteración anterior. Los casos de uso capturan los requerimientos, se puede definir como una funcionalidad, parte de ella o incluso un conjunto de funcionalidades del sistema. Constituyen hilos conductores del proceso de desarrollo permitiendo la trazabilidad de los artefactos generados. La arquitectura es la definición de un conjunto de aspectos significativos del sistema que indican cómo debe ser construido, para esto se toman en cuenta los casos de usos significativos y otros elementos entre los que se pueden mencionar: el rendimiento, software y hardware requeridos para el correcto funcionamiento de la aplicación, así como la reutilización y flexibilidad **(5)**.

RUP divide el proceso en cuatro fases (Fase de Comienzo o Inicio, Fase de Elaboración, Fase de Construcción y Fase de Transición), dentro de las cuales se realizan varias iteraciones en número variable según el proyecto y en las que se hace un mayor o menor hincapié en las distintas actividades o flujos de trabajo como también se le llama **(16)**.

Como conclusión del análisis se determinó que RUP es la metodología que resulta factible utilizar, pues el equipo de desarrollo está compuesto principalmente por estudiantes, esto implica la posibilidad de que ocurran cambios de rol y la necesidad de incorporar nuevos miembros al proyecto. Es importante resaltar que el proyecto PRIMICIA la utilizó durante el desarrollo de la primera versión de la plataforma, por lo que ya se tiene experiencia en cuanto a su uso.

1.5.2 Lenguaje Unificado de Modelado (UML) como soporte de la modelación de la solución propuesta

UML es un lenguaje de modelado gráfico que permite modelar, construir y documentar los elementos que forman un sistema orientado a objetos. Este permite modelar artefactos conceptuales como son: procesos de negocio y funciones de sistema; además de artefactos concretos como son: escribir clases en un

lenguaje determinado, esquemas de bases de datos y componentes de software reusables. Es importante aclarar que UML es un lenguaje para especificar y no un método o un proceso. Aunque es flexible y permite aplicarse en una gran variedad de metodologías de desarrollo, no especifica en sí cual se debe de utilizar. Para modelar, UML utiliza diagramas que se pueden clasificar en tres tipos: diagramas de estructura estática que comprenden los diagramas de clases, de objetos, y de casos de uso; diagramas de comportamiento entre los que se encuentran los diagramas de actividades, de estados y de interacción (secuencia y colaboración) y por último los diagramas de implementación que lo integran los diagramas de componentes y de despliegue **(17)**.

En el desarrollo del Subsistema de Administración de PRIMICIA se utilizará UML v2.0 ya que permite la comprensión por parte de los desarrolladores de las funciones que debe realizar la aplicación, pues se encarga de modelar, construir y documentar los elementos que forman dicho subsistema. Es en un lenguaje totalmente independiente del lenguaje de programación que se use para implementar y se utiliza para modelar sistemas basados en la programación orientada a objetos.

1.5.3 Herramienta CASE ⁹ Visual Paradigm

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

Visual Paradigm v8.0 es la herramienta CASE seleccionada ya que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El software de modelado UML ayuda a una rápida construcción de aplicaciones de calidad y a un menor costo. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación **(18)**.

Visual Paradigm es una herramienta CASE multiplataforma que utiliza el lenguaje de modelado gráfico UML. El ambiente de Visual Paradigm proporciona medios intuitivos para realizar el análisis y el diseño orientados al objeto de un sistema, donde se pueden crear diagramas de UML. Esta herramienta brinda un entorno de desarrollo amigable para aquel que lo utilice, ya que permite organizar el trabajo a la

⁹ CASE: Ingeniería de Software Asistida por Computadora.

comodidad de cada cual. Permite la realización de ingeniería directa e inversa, el modelado de procesos de negocio, de requerimientos y de base de datos. Admite la integración con varios entornos de desarrollo (IDEs) además de ser una herramienta colaborativa pues soporta múltiples usuarios trabajando sobre el mismo proyecto y a la vez. Permite realizar el control de versiones, característica que tiene un impacto positivo en proyectos que cuentan con amplios equipos de desarrollo como PRIMICIA.

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

1.5.4 Lenguaje de programación

Se define como lenguaje de programación al elemento dentro de la informática que permite crear programas mediante un conjunto de instrucciones, operadores y reglas de sintaxis; que se pone a disposición del programador para que este pueda comunicarse con los dispositivos hardware y software existentes **(19)**.

1.5.4.1 Lenguaje C++

Es un lenguaje de programación basado en C que soporta directamente conceptos de la Programación Orientación a Objetos (POO). Desde el punto de vista de la POO ofrece una buena cantidad de recursos, como son las formas de encapsulamiento, la sobrecarga de operadores y funciones, la herencia y el polimorfismo. Los mayores beneficios se aprecian a largo plazo: alta calidad del software, mayor facilidad de adaptación, esta última apreciada sobre todo en grandes proyectos **(20)**. Es uno de los lenguajes de programación que se emplea en la actualidad. Puede emplearse mediante programación basada en eventos para crear programas que usen interfaz gráfica de usuario.

Las principales ventajas que presenta son **(20)**:

- **Difusión:** al ser uno de los lenguajes empleados en la actualidad, posee un gran número de usuarios y existe una gran cantidad de libros, cursos y páginas web dedicados a su estudio.
- **Versatilidad:** es un lenguaje de propósito general, por lo que se puede emplear para resolver cualquier tipo de problema.
- **Portabilidad:** está estandarizado y un mismo código fuente se puede compilar en diversas plataformas.

- Eficiencia: es uno de los lenguajes rápidos en cuanto a tiempo de ejecución.
- Herramientas: existe una gran cantidad de compiladores, depuradores y librerías para el trabajo con él.

1.5.5 Marco de trabajo

En el desarrollo de software, un marco de trabajo o *framework* (en inglés) es una estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros software para ayudar a desarrollar y unir los diferentes componentes de un proyecto **(21)**.

1.5.5.1 Qt

Para el desarrollo de la aplicación se seleccionó el marco de trabajo Qt v5.0.1. Es un marco de trabajo multiplataforma para el desarrollo de aplicaciones con interfaz gráfica de usuario o de consola. Presenta características entre las que se destacan las siguientes:

- Es portable desde sistemas de escritorio hasta sistemas operativos embebidos.
- Herramientas y entornos de desarrollo multiplataforma.
- Alto rendimiento en todas las plataformas.

Cuenta con abundante documentación, arquitectura que soporta el uso de complementos y hasta el momento ha sido liberado bajo dos licencias, la LGPL y la comercial **(21)**. Presenta varios componentes que facilitan el trabajo a los desarrolladores, los cuales se mencionan a continuación **(21)**:

- Las librerías Qt: clases escritas en C++ que facilitan el desarrollo.
- QtDesigner: para diseñar formularios visualmente.
- QtAssistant: acceso rápido a la documentación.
- QtLinguist: traducción rápida de programas.
- Qmake: simplifica el proceso de construcción de proyectos en las diferentes plataformas soportadas.

1.5.6 Entorno integrado de desarrollo

Un entorno integrado de desarrollo (en inglés *Integrated Development Environment* o *IDE*) es un programa compuesto por una serie de herramientas que utilizan los programadores para escribir códigos. Esta herramienta puede estar pensada para su utilización con un único lenguaje de programación o bien puede dar cabida a varios de estos. Las herramientas que normalmente componen un entorno integrado de desarrollo son las siguientes: un editor de texto, un compilador, un intérprete, unas herramientas para la automatización, un depurador, un sistema de ayuda para la construcción de interfaces gráficas de usuario y, opcionalmente, un sistema de control de versiones.

1.5.6.1 Qt Creator

En el desarrollo de la presente investigación se seleccionó como IDE Qt Creator v2.6.2. Es un entorno integrado de desarrollo multiplataforma, adaptado a las necesidades de los desarrolladores, permite crear aplicaciones de escritorio y plataformas de dispositivos móviles. Sus características relevantes se detallan a continuación **(21)**:

- Editor de código sofisticado: ofrece soporte para la edición de C++, ayuda sensible al contexto, completamiento de código y buena navegación.
- Diseño de interfaz de usuario integrado: proporciona dos editores visuales integrados: Qt Designer para la creación de interfaces de usuario y Qt Quick Designer para el desarrollo de interfaces de usuario animadas con el lenguaje QML.
- Proyecto y gestión de construcción: si se importa un proyecto existente o se crea uno desde cero, genera todos los archivos necesarios.

1.5.7 Sistema gestor de base de datos (BD)

El sistema gestor de la base de datos constituye la interfaz entre la BD, la aplicación que hace uso de esta y los usuarios. Tiene como objetivo facilitar al usuario las herramientas que le permitan administrar la información, sin tener en cuenta cómo se van guardando estas en el disco duro. Entre los gestores conocidos en la actualidad se encuentran MySQL, Oracle y PostgreSQL.

1.5.7.1 PostgreSQL

Para el desarrollo del presente trabajo se seleccionó como sistema gestor de base de datos PostgreSQL v9.1. PostgreSQL es un potente sistema de gestión de base de datos objeto-relacional multiplataforma; ha tenido una larga evolución, la cual se inicia en 1977 con el proyecto Ingres en la Universidad de Berkeley

bajo la licencia BSD (Berkeley Software Distribution) que permite el uso del código fuente en software no libre. Es adaptable a las necesidades del cliente y posee una variada documentación en diferentes idiomas debido fundamentalmente a sus numerosas comunidades de desarrollo. Presenta entre sus características significativas la atomicidad, consistencia, aislamiento y durabilidad de las operaciones tratadas.

1.6 Conclusiones

Durante este capítulo se desarrolló un análisis de los elementos teóricos que sirven como base en el presente trabajo de diploma. Para ello se realizó un estudio acerca de los principales conceptos asociados al tema como; sistema, subsistema, administración y módulo, además de las soluciones similares a PRIMICIA existentes en el mundo; llegando de esta forma a las siguientes conclusiones:

1. El análisis de las soluciones existentes tanto en el ámbito nacional e internacional permitió concluir que las características de dichos sistemas no cumplen con algunos de los requerimientos especificados para el subsistema de administración de PRIMICIA v2.0.
2. El uso de RUP como metodología de desarrollo permite la generación de los artefactos necesarios en el flujo de Implementación, este proceso es vital para un mayor entendimiento de la aplicación desarrollada.
3. Con el uso del lenguaje de modelado UML es posible crear una abstracción del sistema a implementar, permitiendo modelar las funcionalidades básicas de este.
4. El marco de trabajo Qt permite crear una aplicación de alta calidad.

Capítulo 2: Presentación de la Solución Propuesta

2.1 Introducción

En el presente capítulo se ofrece una descripción del diseño del Subsistema de Administración propuesto para dar solución al problema que dio origen a la investigación. Al no identificarse un negocio se expone la conceptualización del entorno mediante un modelo de dominio. Se enuncian los Requisitos no Funcionales (RNF) y Requisitos Funcionales (RF) del sistema y se realizan las descripciones de los casos de uso generados a partir de los RF identificados.

2.2 Modelo del dominio

Debido a que los procesos del negocio que tienen que ver con el objeto de estudio no se encuentran definidos, se emplea un modelo conceptual que permite mostrar al usuario los principales conceptos que se manejan en el entorno y de esta manera contribuir a la comprensión del contexto del sistema. En este epígrafe se presenta un glosario de términos para identificar todos los conceptos presentes.

Un modelo de dominio captura los tipos importantes de objetos en el contexto del sistema. Los objetos del dominio representan las "cosas" que existen o los eventos que suceden en el entorno en el que trabaja el sistema **(22)**.

Es un diccionario visual de términos o conceptos importantes en el dominio. Utiliza la notación UML de diagrama de estructura estática. Se enfoca en la semántica y proporciona un glosario de términos basado en sustantivos **(22)**.

El modelo de dominio es el punto de partida para lograr el diseño adecuado del sistema, siempre que los procesos de negocio no se puedan identificar fácilmente. Mediante este modelo se obtiene una descripción del problema, pues se visualizan las clases conceptuales y sus relaciones, desde el punto de vista del mundo real.

El diagrama que se presenta a continuación posibilita a los clientes, usuarios finales del sistema y desarrolladores un estándar para el entendimiento del problema. En él se expresan y vinculan los conceptos primarios del medio donde operará el Subsistema de Administración.

2.2.1 Eventos principales del entorno

El proceso de redacción en la Plataforma de Televisión Informativa PRIMICIA v2.0 se inicia cuando el **redactor** selecciona la opción Noticia de menú principal y luego presiona el botón Crear. A continuación el sistema mostrará el **área de trabajo** y el redactor le pondrá título a la noticia, elegirá un audio y seleccionará la **sección temática** a la que pertenecerá dicha noticia. Luego el redactor irá agregando **pantallas** y **componentes** a la noticia pudiendo definir un tiempo para la visualización de la pantalla y una **transición** de entrada y salida. También podrá ajustarle las **propiedades** a los componentes. Por último cuando el redactor considere que la noticia esté terminada deberá presionar la opción Guardar.

El proceso de gestión de medias en PRIMICIA v2.0 se inicia cuando el **Administrador de Media** selecciona la opción Medias del menú principal. A continuación el sistema muestra un formulario con los tipos de medias (imagen, video y audio) además del género y autor listados todos en una tabla. Luego el Administrador selecciona el tipo de media al que desee realizar la operación, (imagen, video o audio) y el sistema muestra las opciones para la gestión de medias (Insertar, Actualizar, Eliminar, y Reproducir, esta última en el caso de la imagen sería Visualizar). En la Figura 1 se muestra el Diagrama de Clases del Dominio del Sistema.

2.2.2 Diagrama de Clases del Dominio

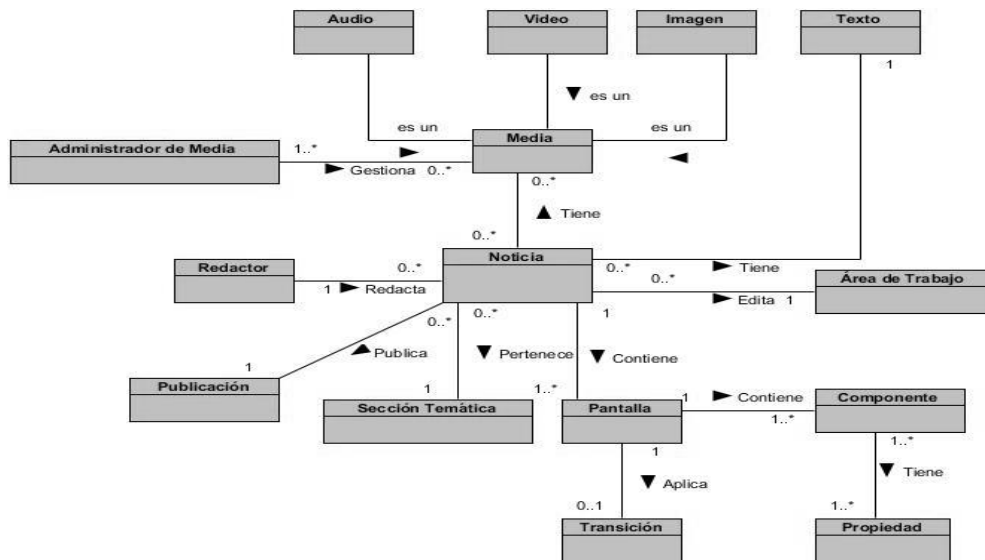


Figura 1. Diagrama de Clases del Modelo de Dominio.

2.2.3 Glosario de Términos del Dominio

Área de Trabajo: espacio donde se edita la noticia.

Noticia: conjunto de textos, imágenes, videos y audio que muestran la información que se desea hacer llegar a la teleaudiencia.

Redactor: persona que se encarga de redactar las noticias que se transmiten en la plataforma.

Transición: tiempo de entrada y salida de la noticia.

Pantalla: unidad más pequeña de la noticia. Es donde se crea la noticia.

Componente: son los componentes de la pantalla, de tipo: texto, imagen y video.

Propiedad: propiedad de los componentes que forman la pantalla tiene los atributos: identificador, pos X, pos Y, ancho, alto y opacidad.

Sección Temática: es la sección a la que pertenece la noticia. Pueden ser: deportivas, internacionales, culturales y ciencia.

Administrador de Medias: persona que gestiona los recursos multimedia que se utilizan en la redacción de las noticias.

Media: archivos multimedia (imagen, video o audio) que se almacenan y se utilizan para la confección de las informaciones que se transmiten por el canal.

Imagen: archivo con formato de imagen que se almacena y se utiliza para la confección de las noticias.

Video: archivo con formato de video que se almacena y se utiliza para la confección de las noticias.

Audio: archivo con formato de audio que se almacena y se utiliza para la confección de las noticias.

Texto: archivo con formato de texto que se almacena y se utiliza para la confección de las noticias.

Publicación: Conjunto de noticias que son publicadas para transmitir las por el canal.

2.3 Requisitos

La primera etapa de este proceso de negocio comienza por la tarea de la identificación de los requisitos con los que debe cumplir el software. Esto ayuda a que se logre un mejor diseño, una correcta implementación, lo que trae consigo que se alcance un resultado exitoso. Los Requisitos Funcionales son

capacidades o condiciones que el sistema debe ser capaz de realizar. A continuación se enumeran los RF del software que se propone.

2.3.1 Requisitos Funcionales

RF 1 Listar noticias. El sistema debe brindar un listado que contenga todas las noticias almacenadas en la base de datos.

RF 2 Visualizar noticia. El sistema debe permitir visualizar las noticias que se localicen en el listado de noticias.

RF 3 Actualizar noticia. El sistema debe admitir la actualización de las noticias que se encuentren en el listado de noticias.

RF 4 Eliminar noticia. El sistema debe admitir la eliminación de una o varias noticias que se encuentren en el listado de noticias.

RF 5 Crear noticia. El sistema debe permitir la creación de noticias haciendo uso de los componentes definidos.

RF 6 Editar Noticia. El sistema debe permitir la edición de las noticias creadas utilizando los elementos configurables.

RF 7 Crear pantalla. El sistema debe admitir adicionar una pantalla a la noticia.

RF 8 Editar pantalla. El sistema debe permitir editar una pantalla de la noticia.

RF 9 Editar texto. El sistema debe permitir editar un texto de alguna de las pantallas de la noticia del sistema.

RF 10 Eliminar pantalla. El sistema debe permitir eliminar una pantalla de la noticia.

RF 11 Insertar componente. El sistema debe permitir la inserción de los componentes de la noticia (texto, imagen y video).

RF 12 Eliminar componente. El sistema debe permitir eliminar un componente de la pantalla.

RF 13 Redimensionar texto o imagen. El sistema debe facilitar redimensionar cualquier componente texto o imagen de la noticia.

RF 14 Ajustar propiedades. El sistema debe admitir ajustar las propiedades de un componente de la noticia.

RF 15 Adicionar Media. El sistema debe brindar la posibilidad de adicionar recursos multimedia.

RF 16 Listar Media. El sistema debe brindar un listado que contenga todas las medias almacenadas en la base de datos que se muestran en tres categorías audio, video e imagen.

RF 17 Reproducir Medias. El sistema debe ser capaz de reproducir medias para lo cual tiene que poder gestionar el material y visualizarlo sin necesidad de descargar el fichero en la computadora donde se esté trabajando.

RF 18 Modificar Media. El sistema de acuerdo a un pedido del usuario deberá ofrecer la posibilidad de modificar los campos correspondientes a la media seleccionada.

RF 19 Eliminar Media. El sistema debe ser capaz de eliminar de la base de datos todos los campos correspondientes a la media seleccionada, así como la media en sí.

RF 20 Autenticar Usuario: El sistema debe brindar la posibilidad de autenticar a los usuarios, identificando el mismo y autorizando el acceso a la aplicación según su rol para realizar las funciones definidas para el mismo.

2.3.2 Requisitos no Funcionales

Los RNF son propiedades o cualidades que el producto debe tener, además son aspectos importantes que el producto debe cumplir para lograr que sea atractivo, usable, rápido o confiable **(23)**.

A continuación se exponen, separados por categorías, los diferentes RNF que el sistema debe tener.

2.3.2.1 Requisitos de Software

- Como Sistema Gestor de Base de Datos en el servidor se debe utilizar PostgreSQL 8.4 o superior.
- En el cliente se debe contar con la librería libqt5sql5-psql.
- En el cliente se debe contar con la librería G++.

2.3.2.2 Requisitos de Hardware recomendados

- Procesador Intel Core 2 Duo de 1.4 GHz o superior.
- Disco Duro de 80 GB o superior.

- Memoria RAM de 1 GB o superior.

2.3.2.3 Requisitos de restricciones en el diseño y la implementación

- Lenguaje de programación C++.
- QtCreator como IDE de desarrollo.
- Visual Paradigm como herramienta CASE.

2.3.2.4 Requisitos de apariencia e interfaz externa

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

2.3.2.5 Requisitos de Seguridad

El software solamente debe permitir a cada usuario realizar las acciones para las que tenga permisos; será necesaria la autenticación previa. Haciendo uso para ello del modelo de seguridad de autorización en la base de datos; expresada a través de los derechos de acceso donde se tiene:

Usuario: noticia y medias.

Rol: Redactor o Administrador de Medias.

Permiso: permisos de redacción y gestión de medias.

2.4 Modelado del sistema

En el presente epígrafe se definen los actores del sistema que deben interactuar con el módulo de Redacción de Noticias y el de Gestión de Medias del subsistema de administración de PRIMICIA. Por otra parte se confecciona el Diagrama de Casos de Uso del Sistema que muestra las asociaciones actor/caso de uso.

2.4.1 Descripción de los actores del sistema

Los actores representan a los seres humanos, a un software, hardware externo u otros sujetos que interactúen con el sistema que se esté especificando **(24)**.

Un actor del sistema no es parte del sistema en desarrollo, es un agente externo que intercambia información con el mismo en pos de obtener un resultado esperado. Durante el desarrollo de la presente investigación se definen como actores del sistema al **Usuario, Redactor y Administrador de Medias**.

Usuario: tiene los privilegios básicos para autenticarse en el sistema.

Redactor: interactúa de manera directa con el módulo Redacción de Noticias. Se encarga de redactar cada una de las noticias que serán transmitidas.

Administrador de medias: es el encargado de realizar todos los procesos básicos del módulo Gestión de Medias. Tiene acceso a los recursos de medias; las gestiona de acuerdo a criterios propios y entrega las medias necesarias.

2.4.2 Diagrama de casos de uso del sistema

Un diagrama de Casos de Uso del Sistema describe parte del modelo de casos de uso y muestra un conjunto de casos de uso y actores con una asociación entre cada par actor/caso de uso que interactúan (24). A continuación en la Figura 2 se muestra el Diagrama de Casos de Uso del Sistema.

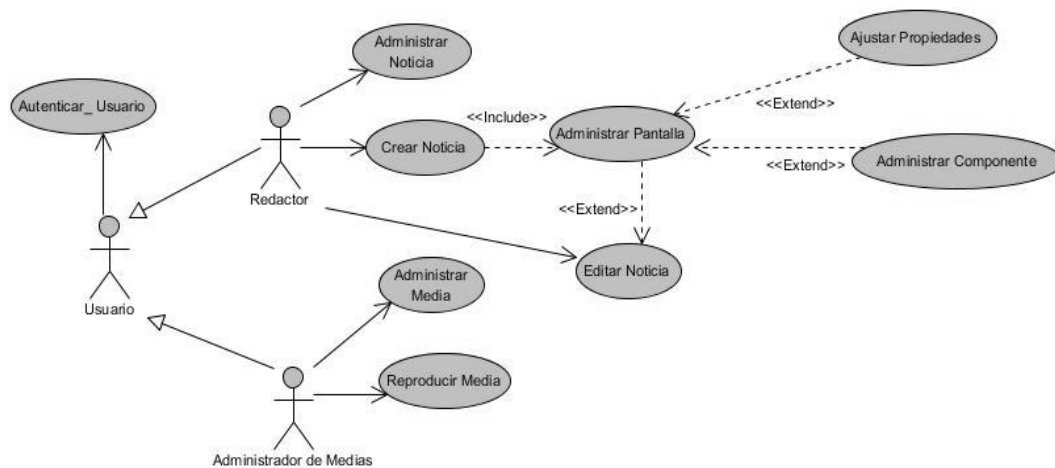


Figura 2. Diagrama de Casos de Uso del Sistema.

2.5 Descripción textual de los casos de uso del sistema

Un Caso de Uso (CU) es una unidad coherente de funcionalidad, externamente visible, proporcionada por una unidad del sistema y expresada por secuencias de mensajes intercambiados por la unidad del sistema y uno o más actores. El propósito de un CU es definir una pieza de comportamiento coherente, sin revelar la estructura interna del sistema. Desde el punto de vista de los usuarios, éstas pueden ser situaciones anormales. Desde el punto de vista de los sistemas, son las variaciones adicionales que deben ser descritas y manejadas (24).

Cada uno de los CU identificados presenta características particulares que para su mayor entendimiento se hace necesario describirlas textualmente. Estos describen qué hace el sistema, y no cómo lo hace. Los CU se obtienen de los servicios requeridos por los usuarios finales, lo que significa que representan un hilo conductor como guía en el desarrollo de software. En la especificación de los CU se describe, bajo la forma de acciones y reacciones, el comportamiento del sistema desde el punto de vista del usuario. A continuación se realiza una profunda y detallada descripción de los CU significativos del mismo.

Tabla 1. Descripción Textual del Caso de Uso: Administrar Medias.

Objetivo	Administrar las medias del sistema.	
Actores	Administrador de Medias.	
Resumen	Este caso de uso comienza cuando el Administrador de Medias desea listar, adicionar, modificar o eliminar algún fichero del repositorio de medias de la entidad. Para ello accede al menú principal y selecciona la opción Medias.	
Complejidad	Alta.	
Prioridad	Crítico.	
Precondiciones	El actor debe estar autenticado en la aplicación y tener los permisos de administración de medias.	
Postcondiciones	Se administran las medias del sistema.	
Flujo de eventos		
Flujo básico Administrar Media.		
	Actor	Sistema
1.	Selecciona la opción Medias del menú principal.	
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 opciones (Insertar, Actualizar, Eliminar, y Reproducir, esta última en el caso de la imagen sería Visualizar).
5.	Selecciona una opción.	
6.		Si Elige: a. Insertar (A), ver sección “Adicionar Medias”. b. Actualizar (B), ver sección “Actualizar Medias”. c. Eliminar (C), ver sección “Eliminar Medias”. d. Visualizar (D), ver sección “Visualizar Medias”. e. Reproducir (E), ir al Caso de Uso Reproducir Media .
Sección 1. “Adicionar Medias”		
Flujo Básico		
	Actor	Sistema
1.		Muestra los diferentes campos que debe llenar el usuario para adicionar la media.
2.	Llena los campos necesarios para adicionar el recurso, y selecciona la opción Guardar.	
3.		Realiza la acción, es decir, añade la URL de la media en la base de datos y muestra un mensaje de información “Se ha insertado correctamente”. Termina así el Caso de Uso.
Flujos Alternos		

Campos Vacíos		
	Actor	Sistema
1.	Deja vacío el campo “Nombre” o no selecciona la ubicación de la media y presiona el botón Guardar.	
2.		Muestra un mensaje de advertencia explicando que “Debe llenar todos los campos”.
Flujos Alternos		
Cancela la Creación		
	Actor	Sistema
1.	Presiona el botón Cancelar.	
2.		Cancela la creación de la media.
Sección 2. “Actualizar Medias”		
Flujo Básico		
	Actor	Sistema
1.	Selecciona en la tabla de medias (video, imagen o audio) la que desea actualizar.	
2.		Muestra los diferentes campos que conforman la media.
3.	Actualiza los campos necesarios y selecciona la opción Guardar.	
4.		Muestra un mensaje de información diciendo “Se ha actualizado correctamente” y se muestra la actualización en la tabla de Medias. Termina así el Caso de Uso.
Flujos Alternos		
Seleccionar Media		

	Actor	Sistema
1.	No selecciona una media del Listado de Medidas Disponibles y presiona el botón Actualizar.	
2.		Muestra un mensaje de advertencia diciendo “Debe seleccionar una media para actualizar”.
Campos Vacíos		
1.	Deja vacío el campo “Nombre” y presiona el botón Guardar.	
2.		Muestra un mensaje de advertencia diciendo “Debe llenar todos los campos”.
Cancela la Actualización		
1.	Presiona el botón Cancelar.	
		Cancela la actualización de la Media.
Sección 3. “Eliminar Media”		
Flujo Básico		
	Actor	Sistema
1.	Selecciona en la tabla de medias (video, imagen o audio) la que desea eliminar.	
2.		Muestra un mensaje de confirmación “¿Seguro que desea eliminar la media?”
3.	Selecciona la opción Sí.	
4.		Muestra un mensaje de información diciendo: “Se ha eliminado correctamente” y se elimina la media de la tabla de medias. Termina así el Caso de Uso.

Flujos Alternos		
Seleccionar Media		
	Actor	Sistema
1.	No selecciona en la tabla de medias disponibles la que desea eliminar.	
2.		Muestra un mensaje de advertencia diciendo "Debe seleccionar la media que desea eliminar".
Seleccionar Opción		
1.	Selecciona la opción No.	
2.		Cancela la eliminación de la Media.
Sección 4. "Visualizar Media"		
Flujo Básico		
	Actor	Sistema
1.	Selecciona en la tabla de imagen la que desea visualizar.	
2.		Muestra la imagen seleccionada. Termina así el Caso de Uso.
Flujo Alterno		
Seleccionar Imagen		
	Actor	Sistema
1.	No selecciona una imagen del listado de imágenes disponibles.	
2.		Muestra un mensaje de advertencia diciendo "Debe seleccionar la imagen que desea visualizar".
Relaciones	CU Incluidos	No existen.
	CU Extendidos	No existen.
Requisitos	RF 15, RF 16, RF 18 y RF 19.	

Funcionales	
Prototipo De Interfaz	

Tabla 2. Descripción Textual del Caso de Uso: Crear Noticia.

Objetivo	Crear las noticias que contendrá el sistema.	
Actores	Redactor.	
Resumen	El Caso de Uso se inicia cuando el redactor desea agregar una noticia al sistema. Para esto el redactor presiona el botón Crear.	
Complejidad	Alta.	
Prioridad	Crítico.	
Precondiciones	El actor debe estar autenticado en la aplicación y tener los permisos de redacción.	
Postcondiciones	Se adicionan noticias al sistema.	
Flujo de eventos		
Flujo básico Crear Noticia.		
	Actor	Sistema
1.	Selecciona la opción Noticia del menú principal.	
2.		Muestra un formulario donde aparece un listado con todas las noticias que contiene. Muestra las opciones (Crear, Editar, Visualizar y Eliminar).

3.	Presiona el botón Crear.	
4.		Muestra un formulario con los campos necesarios para crear la noticia.
5.	Ingresa el título de la noticia y selecciona el audio.	
6.		Muestra el título del audio seleccionado y brinda la opción de reproducirlo.
7.	Selecciona la sección temática a la que pertenece la noticia.	
8.		Muestra el nombre de la sección.
9.	Presiona el botón Adicionar para agregar una pantalla.	
10.		Ir al Caso de Uso Administrar pantalla.
11.	Presiona el botón guardar.	
12.		Guarda la noticia en la base de datos y en el XML y muestra el listado de noticias. Termina así el Caso de Uso.

Flujos Alternos

Campos Vacíos

	Actor	Sistema
1.	Deja vacío el campo "Título" y presiona el botón Guardar.	
2.		Muestra un mensaje de advertencia explicando que "Debe ingresar un título para la noticia".

Flujos Alternos

Cancela la Creación


	Actor	Sistema
1.	Presiona el botón Cancelar.	
2.		Cancela la creación de la noticia y muestra el listado de noticias.
Flujos Alternos		
Seleccionar Media		
	Actor	Sistema
1.	Presiona el botón guardar sin haber creado previamente la noticia	
2.		Muestra mensaje de información “No se han creado elementos para conformar la noticia.”
Relaciones	CU Incluidos	Administrar Pantalla.
	CU Extendidos	No existen.
Requisitos Funcionales	RF 5.	
Prototipo De Interfaz		

Tabla 3. Descripción Textual del Caso de Uso: Administrar Pantalla.


Objetivo	Administrar las pantallas de las noticias del sistema.
Actores	Caso de Uso Crear Noticia.
Resumen	El Caso de Uso se inicia cuando se necesita crear, editar y eliminar

	las pantallas de una determinada noticia del sistema.	
Complejidad	Alta.	
Prioridad	Crítico.	
Precondiciones	Debe estar creada la Noticia.	
Postcondiciones	Quedan actualizadas las pantallas de una noticia del sistema.	
Flujo de eventos		
Flujo básico Administrar Pantalla.		
	Actor	Sistema
1.		Muestra el formulario para introducir el tiempo de duración de la pantalla en segundos.
1.	Introduce el tiempo de duración de la pantalla.	
2.		Muestra el área de trabajo en blanco y se activan las opciones para insertar componentes (texto, imagen y video) y las opciones para editar los componentes así como los botones "Cancelar" y "Guardar".
3.	Hace clic en el botón del componente que desee insertar (texto, imagen o video).	
4.		Ir al Caso de Uso Administrar componente.
5.	Puede realizar las siguientes acciones a los componentes insertados:	
6.		Si el actor selecciona la opción: a. Eliminar componente, ir a la sección Eliminar componente del Caso de Uso Administrar

		<p>componente.</p> <p>b. Redimensionar, de los componentes texto e imagen, ir a la sección Redimensionar del Caso de Uso Administrar componente.</p> <p>c. Editar, del componente texto, ir a la sección Editar texto del Caso de Uso Administrar componente.</p> <p>d. Ajustar Propiedades, ir al Caso de Uso Ajustar Propiedades.</p>
7.	A las pantallas creadas le puede realizar diferentes acciones.	
8.		<p>Si el actor elige la opción:</p> <p>a. Adicionar (A), ir a la acción 1 del Flujo Básico de Eventos del Caso de Uso Administrar pantalla.</p> <p>b. Editar (B), ver sección Editar pantalla del Caso de Uso Administrar pantalla.</p> <p>c. Eliminar (C), ver sección Eliminar pantalla del Caso de Uso Administrar pantalla.</p>
Sección 1. "Eliminar Pantalla"		
Flujo Básico		
	Actor	Sistema
1.	Selecciona la pantalla que desea eliminar y presiona el botón Eliminar del formulario.	
2.		Muestra un mensaje de confirmación para eliminar la pantalla que diciendo "¿Desea eliminar la pantalla?".

3.	Confirma que desea eliminar la pantalla y presiona la opción Sí.	
4.		Elimina la pantalla de la noticia y actualiza la tabla del listado de pantallas. Termina así el Caso de Uso.
Flujos Alternos		
Cancela la Eliminación		
	Actor	Sistema
1.	Cancela la confirmación de que desea eliminar la pantalla y presiona la opción No.	
2.		Cancela la eliminación de la pantalla y muestra el área de trabajo con los componentes de la noticia que se está redactando.
Flujos Alternos		
Seleccionar Pantalla		
	Actor	Sistema
1.	No selecciona la pantalla y presiona el botón Eliminar.	
2.		Muestra un mensaje de advertencia diciendo "Debe seleccionar la pantalla que desea eliminar".
Sección 2. "Editar Pantalla"		
Flujo Básico		
	Actor	Sistema
1.	Selecciona la pantalla que desea editar y presiona el botón Editar.	

2.		Muestra en el área de trabajo la pantalla seleccionada con los componentes que contiene.
3.	Edita la pantalla realizando las modificaciones que sean necesarias y presiona la opción Guardar.	
4.		Muestra un mensaje de confirmación diciendo “¿Desea guardar los cambios?”. Termina así el Caso de Uso.
5.	Presiona la opción Sí.	
		Actualiza el estado de la pantalla así como el de los componentes de dicha pantalla. Termina así el Caso de Uso.
Flujos Alternos		
Seleccionar Pantalla		
	Actor	Sistema
1.	No selecciona una pantalla y presiona el botón Editar.	
2.		Muestra un mensaje de advertencia diciendo “Debe seleccionar una media para actualizar”.
Flujos Alternos		
Seleccionar Opción		
1.	Presiona la opción No.	
2.		El sistema no guarda los cambios.
Flujos Alternos		
Cancela la Edición		
1.	Selecciona la opción Cancelar.	
		Cancela la edición de la Pantalla.

Relaciones	CU Incluidos	No existen.
	CU Extendidos	Administrar Componente. Ajustar Propiedades.
Requisitos Funcionales	RF 7, RF 8 y RF 10.	
Prototipo De Interfaz		

2.6 Conclusiones

En el presente capítulo se establecieron las características principales de los componentes, así como el proceso para su funcionamiento. Se presentó el modelo del dominio, conjuntamente con la especificación de los RF y RNF del sistema. Además se elaboró el Modelo de Casos de Uso del Sistema y la descripción textual de los CU significativos. Arribando a las siguientes conclusiones:

1. La confección del Diagrama de Clases del Dominio brindó la interrelación existente entre cada una de las clases, esto permitió hacer comprensible el negocio para su posterior implementación.
2. La elaboración del Diagrama de Caso de Uso del Sistema permitió mostrar de manera explícita cómo funciona el software y la relación de cada uno de sus actores con el sistema. Permitted un mayor entendimiento al equipo de desarrollo del sistema a construir.
3. La captura de los requisitos y la posterior descripción de los CU sirvió de guía en el desarrollo del sistema y fue utilizado para la correcta implementación del mismo.

Capítulo 3: Construcción de la Solución Propuesta

3.1 Introducción

En el presente capítulo se abordarán los aspectos esenciales relacionados con la implementación de la solución que se propone para el problema planteado. En este se muestra el modelado de las clases que se han implementado para lograr una visualización de los elementos a realizar. Los diagramas de Clases del Diseño que se exponen, describen la estructura que tendrá el Subsistema de Administración al concluir la etapa de implementación. Se describe la arquitectura del sistema y se muestran desglosados todos los patrones que se han utilizado para lograr un mayor entendimiento de los componentes mostrados. Por último se presentan las pruebas de software realizadas para validar la aplicación.

3.2 Modelo de análisis

El Modelo de Análisis estructura los requisitos de un modo que facilita su comprensión, preparación, modificación y en general su mantenimiento. Puede considerarse como la primera aproximación al modelo de diseño **(25)**.

El Análisis es una disciplina de la metodología RUP, de la cual se puede prescindir de acuerdo a la manera en que se haya decidido implementar el sistema, pasando a analizar los requisitos como parte del diseño o como parte integrada de la captura de los requisitos, lo que demuestra que la manera de emplear el análisis es consecuente con la construcción del sistema **(25)**.

Teniendo en cuenta lo antes expuesto, se decide pasar del flujo de trabajo de requisitos directamente al diseño, sin la realización del análisis. A raíz de que los requisitos son bien conocidos y el lenguaje de programación y tecnologías sobre las que se desarrolla el sistema son elementos dominados, lo que permite prescindir del refinamiento de los requisitos e ir directo al diseño.

3.3 Descripción de la arquitectura de software

En un sistema informático la arquitectura es el conjunto de decisiones significativas sobre la organización de un software. Para describir y diferenciar una arquitectura se hace necesario emplear estilos arquitectónicos y de diseño que ayuden a definir una estructura para todos los componentes del sistema.

3.3.1 Patrón arquitectónico

Según Christopher Alexander en el libro *“The Timeless Way of Building”* define al patrón en la siguiente manera:

“Como elemento de un lenguaje, un patrón es una instrucción que muestra como puede ser usada esta configuración espacial una y otra vez para resolver el sistema de fuerzas, siempre que el contexto lo haga relevante.” “Cada patrón describe un problema que ocurre una y otra vez en el entorno, para describir después el núcleo de la solución a ese problema, de tal manera que esa solución pueda ser usada más de un millón de veces sin hacerlo siquiera dos veces de la misma forma **(26)**.” Existen diferentes tipos de patrones, entre ellos se pueden encontrar:

- Patrones arquitectónicos.
- Patrones de diseño.

Un patrón arquitectónico es un patrón de software que ofrece soluciones a problemas de arquitecturas comunes surgidos a la hora de construir un nuevo sistema. Su objetivo fundamental es facilitar la tarea del diseño de tal sistema. La utilización de patrones mejora la concepción y entendimiento de los sistemas, por estas razones se decide aplicarlos en el desarrollo del presente trabajo. Para describir y diferenciar una arquitectura se hace necesario emplear estilos arquitectónicos que ayuden a definir una estructura para todos los componentes del sistema.

Un estilo arquitectónico es una transformación impuesta al diseño de todo un sistema. Algunos de estos estilos arquitectónicos son: Arquitectura Centrada en Datos, Arquitectura de Flujo de Trabajo, Arquitectura de Llamada y Retorno, Arquitectura Orientada a Objetos **(26)**.

3.3.1.1 Modelo en Capas

En la presente investigación se hizo uso del estilo arquitectónico de Llamada y Retorno específicamente el patrón arquitectónico en capas. El Modelo en Capas es definido como una organización jerárquica, donde cada capa proporciona servicios a la capa inmediatamente superior y se sirve de las prestaciones que le brinda la inmediatamente inferior **(27)**. Las restricciones topológicas del patrón pueden incluir una limitación rigurosa que exige a cada capa operar sólo con capas adyacentes, y a los elementos de una

capa entenderse sólo con otros elementos de la misma; se supone que si esta exigencia se relaja, el patrón deja de ser puro y pierde algo de su capacidad heurística **(27)**.

Pueden implementarse sistemas con dos, tres, n capas, sin embargo en la presente investigación se hizo uso del modelo de tres capas:

Capa de presentación: reúne los aspectos del software relacionados con las interfaces y la interacción con los diferentes tipos de usuarios.

Capa de negocio: reúne los aspectos del software que se tienen que automatizar. Incluye las tareas, reglas y las restricciones que se aplican.

Capa de datos: constituye el puente entre las dos capas anteriores. Encapsula la lógica de acceso a datos.

3.3.2 Patrones de Diseño

Los patrones intentan codificar el conocimiento, las expresiones y los principios ya existentes. Estos brindan una serie de recomendaciones para organizar los distintos componentes del sistema y así tener un mejor entendimiento del problema al que se le quiere dar solución. Un patrón de diseño es **(27)**:

- Una solución estándar para un problema común de programación.
- Una técnica para flexibilizar el código haciéndolo satisfacer ciertos criterios.
- Un proyecto o estructura de implementación que logra una finalidad determinada.
- Una manera práctica de describir ciertos aspectos de la organización de un programa.

Los patrones de diseño se dividen en dos grupos: Patrones Generales de Software para Asignar Responsabilidades (GRASP) y Pandilla de los Cuatro (GOF). Los patrones GRASP son los encargados de guiar los principios fundamentales de la asignación de responsabilidades en el diseño orientado a objetos y los patrones GOF, se clasifican en dependencia del propósito con el cual van a ser utilizados en tres categorías: creacionales, estructurales y de comportamiento. Para la solución de esta aplicación se tuvo en cuenta diversos patrones de diseño, los cuales se presentan a continuación **(27)**:

3.3.2.1 Patrones Generales de Software para Asignar Responsabilidades

Experto: este patrón, está diseñado para que la responsabilidad de realizar una labor sea de la clase que tiene o puede tener los atributos involucrados. Está presente en todas las clases del sistema, cada responsabilidad está asignada a la clase que cuenta con la información necesaria para realizarla.

Creador: guía la asignación de responsabilidades relacionadas con la creación de objetos. La intención básica del patrón Creador es encontrar un creador que necesite conectarse al objeto creado en alguna situación.

Controlador: define quién deberá encargarse de atender un evento del sistema. Se debe diseñar de forma tal que no posean demasiada responsabilidad, delegando esta en otras clases mientras el controlador coordina la actividad.

Alta cohesión: propone asignar la responsabilidad de manera que la complejidad se mantenga dentro de límites manejables asumiendo solamente las responsabilidades que se deben manejar, evadiendo un trabajo excesivo. Este permitió una mayor claridad en el entendimiento del sistema y una mayor capacidad de reutilización.

Bajo Acoplamiento: debe haber pocas dependencias entre las clases. Si todas las clases dependen de todas no se puede extraer software de forma independiente y reutilizarlo.

3.3.2.2 Pandilla de los Cuatro

Observador: el patrón observador define una dependencia de uno-a-muchos entre objetos, de forma que cuando un objeto cambie de estado se notifique y actualicen automáticamente todos los objetos que dependen de él.

3.4 Modelo de Diseño

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

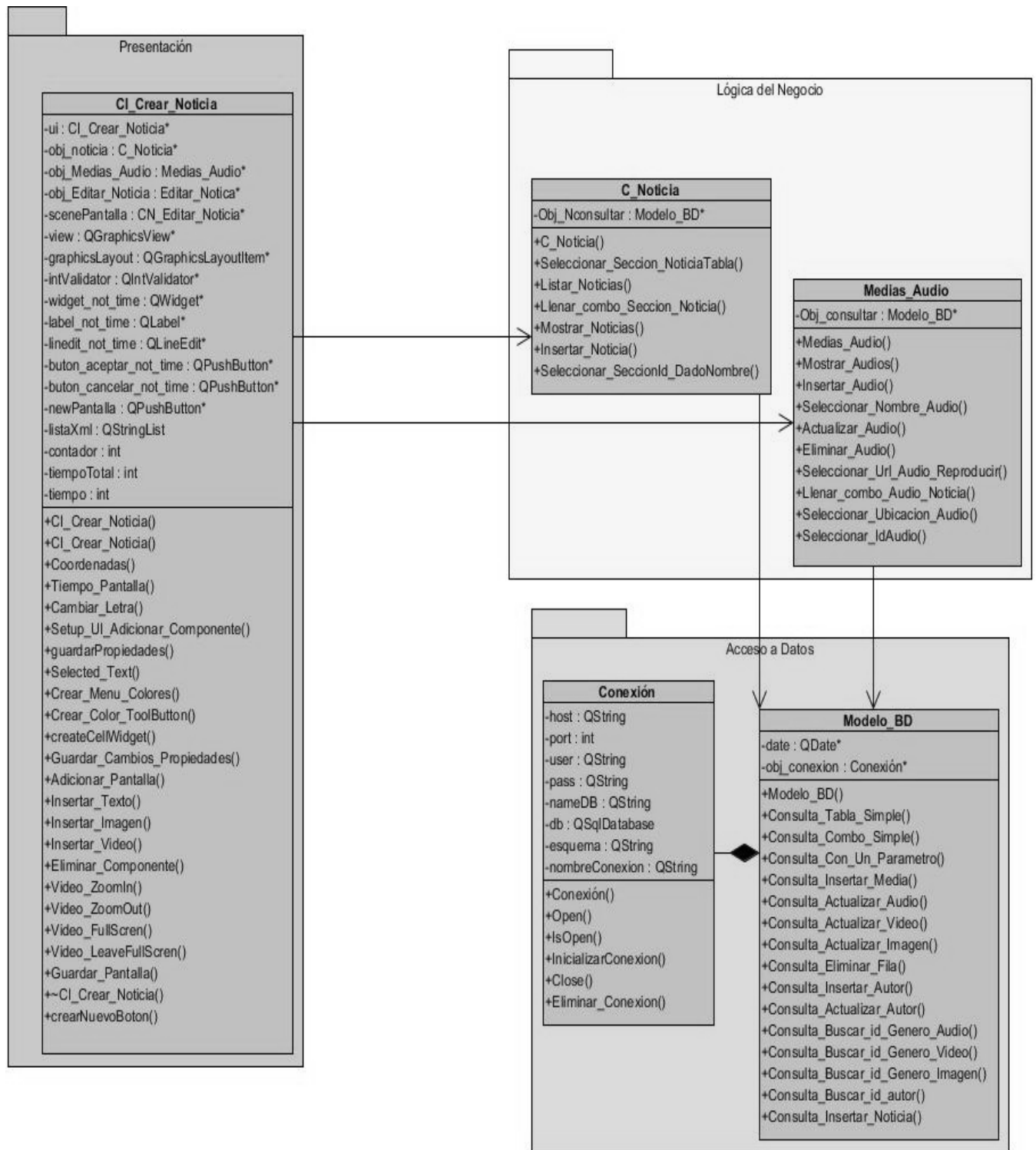


Figura 3. Diagrama de Clases del Diseño del Caso de Uso: Crear Noticia.



Figura 4. Diagrama de Clases del Diseño del Caso de Uso: Administrar Media.

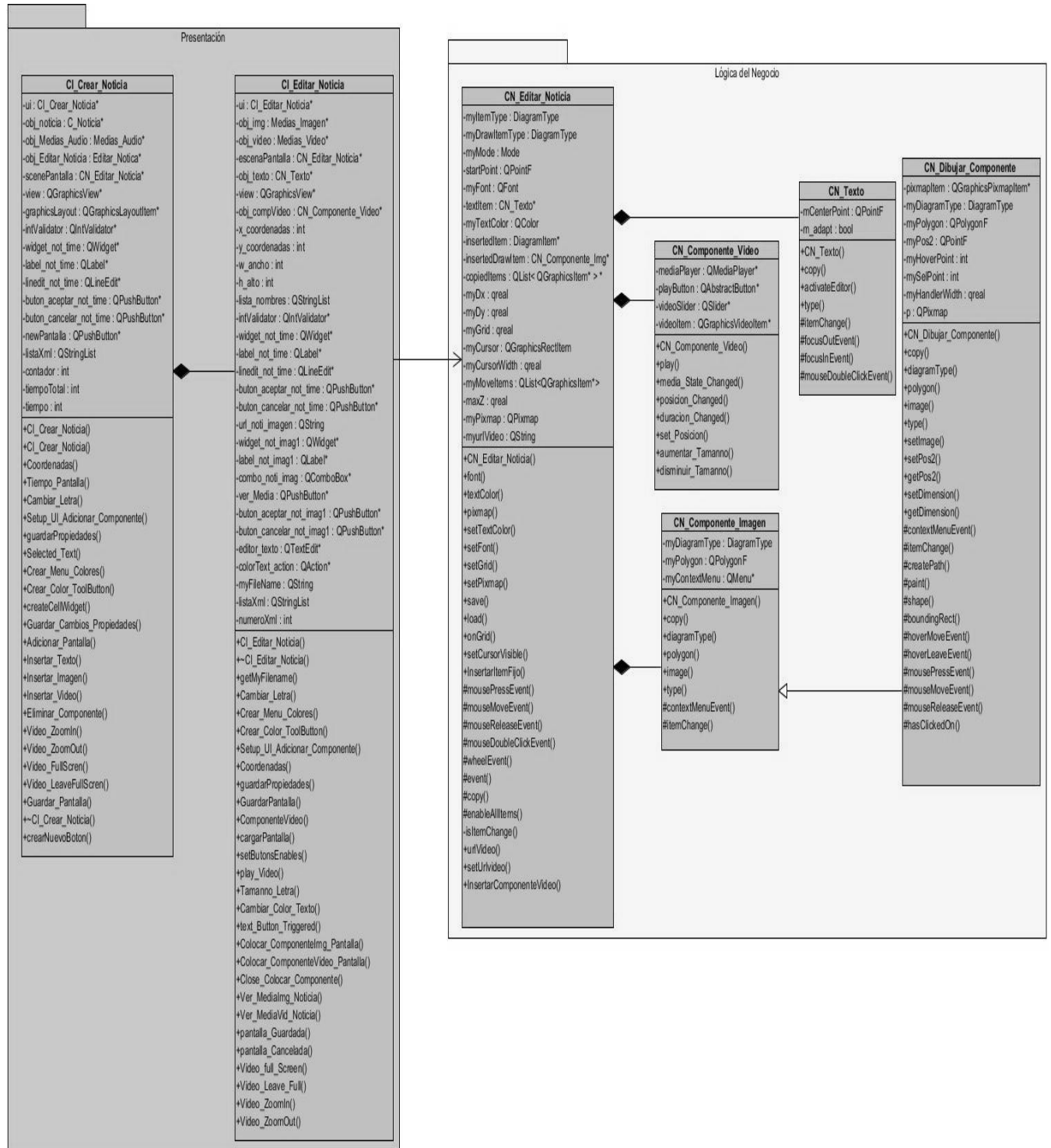


Figura 5. Diagrama de Clases del Diseño del Caso de Uso: Administrar Pantalla.

3.5 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 (22). A continuación se muestra el Modelo de Despliegue de la solución propuesta.

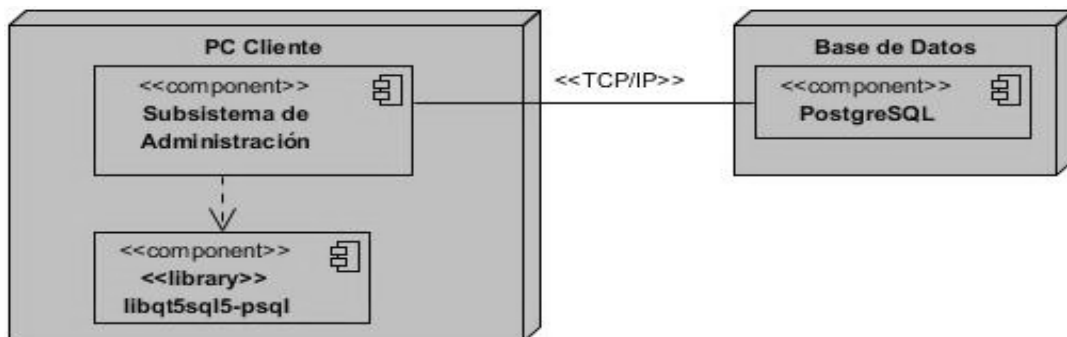


Figura 6. Modelo de Despliegue.

3.6 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 (22).

En el Diagrama de Componentes que se muestra a continuación en la Figura 8, 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.

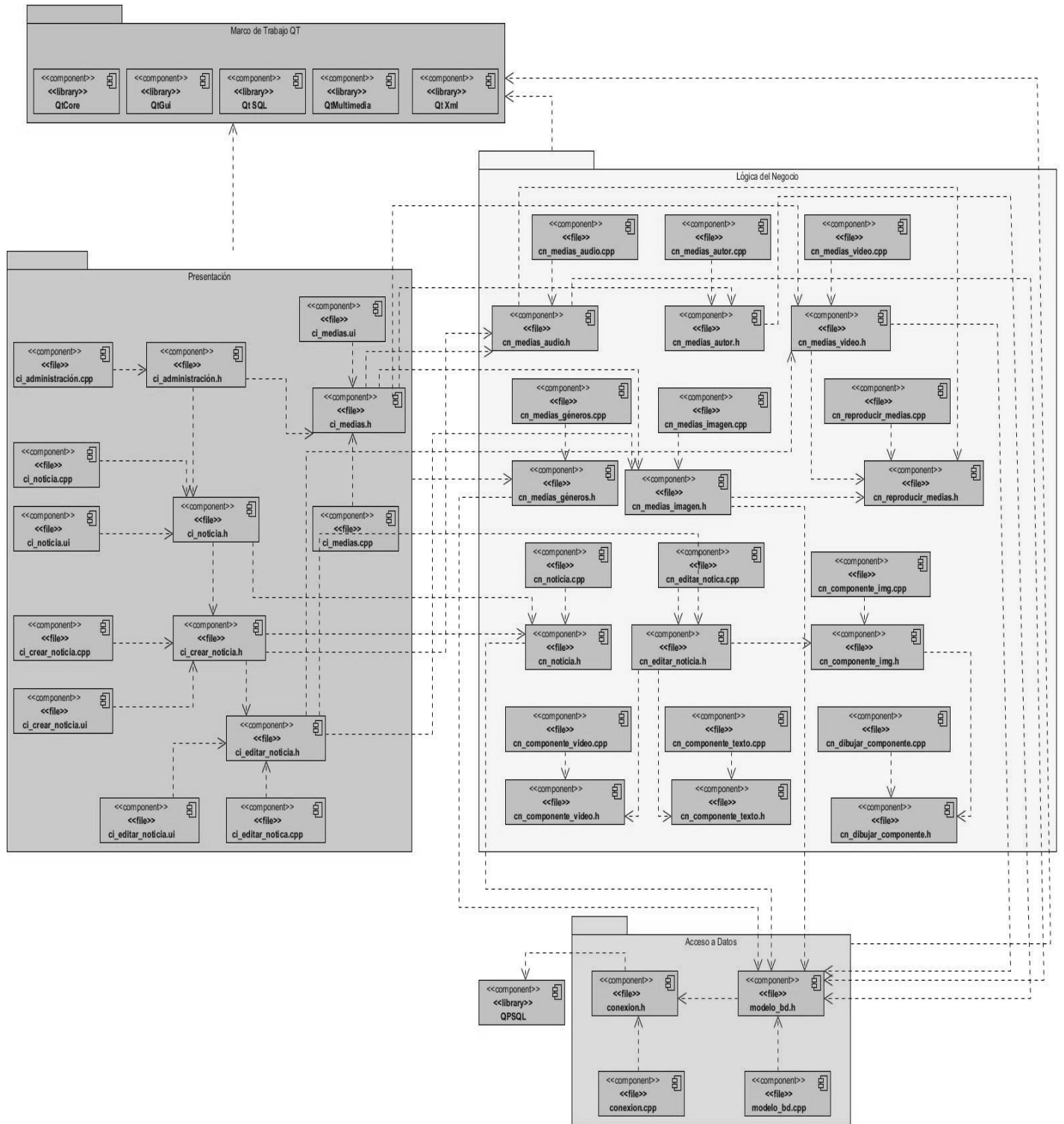


Figura 7. Diagrama de Componentes.

3.6.1 Estándar de codificación

Las convenciones o estándares de codificación son pautas de programación que no están enfocadas a la lógica del programa, sino a su estructura y apariencia física para facilitar la lectura, comprensión y mantenimiento del código **(28)**. A continuación se presentan las convenciones establecidas para el desarrollo de la presente investigación.

- Los nombres de las clases y métodos son sugerentes.
- Las variables se declaran en líneas separadas.
- Existen comentarios descriptivos junto a cada método.
- Los atributos y métodos comienzan con minúscula, cuando son dos palabras la segunda comienza con mayúscula.

3.7 Validación del sistema

El hecho de que el software sea desarrollado por personas condiciona la posibilidad de que el mismo tenga errores. Una buena práctica constituye planificar tareas que estén relacionadas con la garantía de la calidad en todas las fases del flujo de trabajo. En este epígrafe se validará la solución propuesta mediante la realización de pruebas con el objetivo de medir la calidad del subsistema desarrollado.

3.7.1 Pruebas de software

Las pruebas son una actividad en la cual un sistema o componente, son ejecutados bajo unas condiciones o requerimientos especificados, los resultados son observados y registrados, y una evaluación es hecha de algún aspecto del sistema. La prueba de software es un elemento crítico para la garantía de la calidad del software y representa una revisión final de las especificaciones del diseño y de la codificación **(29)**.

Las pruebas de software son el proceso de identificar las fallas o defectos en el sistema con el objetivo de asegurarse de que el mismo cumple con los requisitos del cliente antes de lanzarlo al mercado. Los siguientes son métodos de prueba comunes:

Pruebas de caja blanca: es un método de prueba que especifica cómo probar una realización de caso de uso o un escenario específico de la realización. Un método de prueba de este tipo puede incluir la verificación de la interacción entre los componentes que implementan dicho caso de uso. Los métodos de prueba basados en una realización de caso de uso típicamente especifican una prueba del sistema como caja blanca, es decir, una prueba de la interacción interna entre los componentes del sistema.

Pruebas de caja negra: con la realización de este método de prueba se pueden encontrar errores funcionales que presenta el software y no han sido detectados por los desarrolladores. Se efectúa sin tener acceso al código fuente de la aplicación; se trabaja con entradas y salidas. A continuación se abordan los aspectos significativos de estas pruebas.

Estas pruebas permiten encontrar:

- Funciones que estén incorrectas o ausentes.
- Errores de interfaz.
- Errores en estructuras de datos o en accesos a bases de datos externas.
- Errores de rendimiento.

3.7.2 Niveles de Prueba

Existen varios niveles de pruebas que son mencionados a continuación **(29)**:

Unitario: es el nivel que tienen como objetivo aislar cada parte del programa y mostrar que las partes individuales son correctas; detectando errores en los datos, lógica y algoritmos.

Integración: es el nivel donde los módulos individuales del software son combinados y probados como un grupo con el objetivo de detectar errores de interfaces y relaciones entre componentes.

Funcional: este nivel tiene como objetivo detectar errores en la implementación de los requisitos.

Sistema: el objetivo es verificar el sistema de software para comprobar si este cumple sus requisitos.

Aceptación: es el nivel que tiene como objetivo detectar fallas en la implementación del sistema.

3.7.3 Diseño de Casos de Prueba

Un Caso de Prueba especifica una forma de probar el sistema, incluyendo la entrada o resultado con lo que se ha de probar y las condiciones bajo las que ha de probarse. Un caso de prueba puede derivarse y por tanto puede seguir la traza de un caso de uso **(29)**. Existen varios Diseños de Casos de Pruebas dentro de los que se encuentran las clases de equivalencia; que ha sido utilizado para probar el Subsistema de Administración.

Se aplicaron pruebas de integración con el objetivo de probar si el Subsistema de Administración y el Subsistema de Transmisión de PRIMICIA v2.0 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 **(30)**. En teoría, la combinación de componentes válidos debería dar como resultado un software en el que no se detectan errores, pero en la práctica hay múltiples ocasiones en que se encuentran errores en el software integrado **(30)**. En la siguiente tabla se muestran los resultados de los casos de prueba luego de realizar las pruebas de integración no incremental donde se prueba cada módulo por separado y se prueba todo el producto.

Tabla 4. Casos de Prueba de Integración.

Nombre del Caso de Uso	Nombre de Sección	Elemento	Estado del Elemento	Salida del Subsistema de Administración	Entrada Esperada en el Subsistema de Transmisión
Crear Noticia.	Crear Noticia.	XML	Existe el XML con el formato correcto.	Se crea la noticia y luego se guarda en el XML con el formato correcto.	Se transmite la noticia.
			Existe pero es de formato inadecuado.	Se crea la noticia y luego se guarda en el XML con el formato incorrecto.	Falla de ejecución.
		BD	Base de datos activa.	Se guarda la dirección del XML que contiene la noticia en el servidor de BD y se encuentra disponible el servidor.	Se accede a la dirección del XML.
			No es accesible.	Se guarda la dirección del XML que contiene la noticia en el servidor de BD y no se encuentra disponible el servidor.	Falla de ejecución.

		RED	Lista.	Existe conexión para acceder a la BD.	Se accede a la DB.
			En problemas, desconectada.	Presenta problemas de conexión.	Falla de ejecución.

Para la validación de los procesos de redacción y gestión de medias del Subsistema de Administración de PRIMICIA v2.0 se seleccionó el método de pruebas de caja negra también denominadas pruebas de comportamiento haciendo uso de la técnica de partición equivalente; son pruebas funcionales y pertenecen al nivel de sistema. Se refiere a las pruebas que se llevan a cabo sobre la interfaz del software; se centran en verificar que el sistema cumple con los requisitos funcionales especificados en el capítulo anterior.

En el flujo de trabajo de prueba se realizaron dos iteraciones al subsistema, en la primera iteración se encontraron tres No Conformidades (NC) en el mismo. En los CU Crear Noticia, Administrar Pantalla y Administrar Media los requisitos funcionales RF5: Crear Noticia, RF8: Editar Pantalla y RF19: Eliminar Media no estaban funcionando correctamente. Al encontrar estas NC, se realizó posteriormente una segunda iteración de prueba para verificar que se habían eliminado los problemas de estos requisitos funcionales o si se encontraban nuevos problemas en el funcionamiento del sistema. Esta segunda iteración concluyó satisfactoriamente ya que no se encontró ningún problema en el Subsistema de Administración. A continuación se presentan los casos de prueba resultantes en la segunda iteración para los casos de uso críticos.

Tabla 5. Pruebas del Caso de Uso: Administrar Medias.

Nombre del Caso de Uso	Nombre de Sección	Escenario	Acción Realizada	Respuesta del Sistema	Resultado
Administrar Medias.	SC1: Adicionar Medias.	EC1: El Administrador de Medias adiciona las medias	El Administrador de Medias selecciona la opción Medias del menú principal.	El sistema muestra un formulario con los tipos de medias (imagen, video y audio) además	Satisfactorio.

		correctamente.	Llena los campos necesarios para adicionar el recurso y selecciona la opción guardar.	del género y autor listados todos en una tabla. El sistema adiciona la URL de las medias en la base de datos y muestra el mensaje “Se ha insertado correctamente”.	
		EC2: El Administrador de Medias deja campos vacíos.	El Administrador de Medias no completa los campos.	El sistema muestra un mensaje diciendo “Debe llenar todos los campos”.	Satisfactorio.
	SC2: Actualizar Medias.	EC1: El Administrador de Medias selecciona la media que desea modificar.	El Administrador de Medias debe seleccionar en la tabla de medias (video, imagen o audio) la que desea actualizar. Actualiza los campos necesarios y selecciona la opción Guardar.	El sistema muestra un mensaje diciendo “Se ha actualizado correctamente” y se muestra la actualización en la tabla de Medias.	Satisfactorio.
		EC2: El Administrador de Medias selecciona una	El actor no selecciona una media (audio, imagen o video) de	El sistema muestra un mensaje diciendo “Debe seleccionar una media para actualizar”.	Satisfactorio.

	media.	la tabla.		
	EC3: El Administrador de Medias deja los campos vacíos.	El Administrador de Medias no llena los campos.	El sistema muestra un mensaje diciendo "Debe llenar todos los campos".	Satisfactorio.
	EC4: En caso que el actor presione el botón Cancelar.	El actor presiona el botón Cancelar.	El sistema cancela la actualización de la media.	Satisfactorio.
SC3: Eliminar Medias.	EC1: El actor selecciona la media.	El actor debe seleccionar en la tabla de medias (video, imagen o audio) la que desea eliminar.	El sistema muestra un mensaje "¿Seguro que desea eliminar la media?"	Satisfactorio.
	EC2: El actor confirma la eliminación de la media.	El actor selecciona la opción Sí.	El sistema muestra un mensaje diciendo "Se ha eliminado correctamente" y se elimina la media de la tabla de medias.	Satisfactorio.
	EC3: El Administrador de Medias no selecciona la media.	El Administrador de Medias no selecciona la media que desea eliminar.	El sistema muestra un mensaje diciendo "Debe seleccionar la media que desea eliminar".	Satisfactorio.

		EC4: El actor selecciona la opción No.	El actor selecciona la opción No.	El sistema cancela la eliminación.	Satisfactorio.
	SC4: Visualizar Medias.	EC1: Visualizar las Medias.	El Administrador de Medias debe seleccionar en la tabla de imagen la que desea visualizar.	El sistema muestra la imagen.	Satisfactorio.
		EC2: No se visualiza la media.	El actor no selecciona la imagen que desea visualizar.	Muestra un mensaje diciendo "Debe seleccionar la imagen que desea visualizar".	Satisfactorio.

Tabla 6. Pruebas del Caso de Uso: Administrar Pantalla.

Nombre del Caso de Uso	Nombre de Sección	Escenario	Acción Realizada	Respuesta del Sistema	Resultado
Administar Pantalla	SC1: Crear Pantalla	EC1: El redactor adiciona pantallas a la noticia.	El redactor introduce el tiempo de duración de la pantalla. Hace clic en el botón del componente que desee insertar (texto, imagen o video) y presiona el botón guardar.	El sistema muestra el área de trabajo en blanco y se activan las opciones para insertar componentes (texto, imagen y video) y las opciones para editar los	Satisfactorio.

			El actor selecciona la sección temática de la noticia.	componentes así como los botones “Cancelar” y “Guardar”. El sistema crea la pantalla mostrándola en el formulario.	
		EC2: No se crea la pantalla.	El actor presiona la opción Cancelar.	El sistema cancela la creación de la pantalla.	Satisfactorio.
	SC2: Eliminar Pantalla.	EC1: El redactor elimina la pantalla.	El actor selecciona la pantalla que desea eliminar y presiona el botón Eliminar del formulario. El redactor confirma que desea eliminar la pantalla.	El sistema muestra un mensaje de confirmación para eliminar la pantalla. El sistema elimina la pantalla de la noticia y actualiza la tabla del listado de pantallas.	Satisfactorio.
		EC2: redactor confirma eliminación de la pantalla.	El redactor no presiona la opción Cancelar.	El sistema muestra el área de trabajo con los componentes	Satisfactorio.

			de la noticia que se está redactando.	
SC3: Editar Pantalla	EC1: Se edita la pantalla.	El redactor selecciona la pantalla que desea editar y presiona el botón Editar. Luego realiza las modificaciones necesarias.	El sistema muestra en el área de trabajo la pantalla seleccionada con los componentes que contiene. El sistema muestra un mensaje diciendo "Desea guardar los cambios". Luego guarda las modificaciones realizadas.	Satisfactorio.
	EC2: No se edita la pantalla.	El redactor selecciona la opción cancelar.	El sistema cancela la edición de la pantalla.	Satisfactorio.
	EC3: En caso que el actor presione el botón No.	El actor presiona el botón No.	El sistema no guarda los cambios.	Satisfactorio.

Tabla 7. Pruebas del Caso de Uso: Crear Noticia.

Nombre del Caso de Uso	Nombre de Sección	Escenario	Acción Realizada	Respuesta del Sistema	Resultado
Crear Noticia.	SC1: Crear Noticia.	EC1: Se crea la noticia.	El redactor presiona el botón Crear, ingresa el título de la noticia y selecciona el audio. Presiona el botón guardar.	El sistema guarda la noticia en la base de datos y en el XML y muestra el listado de noticias.	Satisfactorio.
		EC2: No se crea la noticia.	El actor deja el campo "Título" vacíos.	El sistema muestra un mensaje diciendo "Debe ingresar un título para la noticia".	Satisfactorio.
		EC3: No se crea la Noticia.	El actor presiona el botón Cancelar.	El sistema cancela la creación de la noticia.	Satisfactorio.

3.7.4 Pruebas de Rendimiento

Las pruebas de rendimiento son aquellas que son realizadas para determinar qué tan rápido un sistema realiza una tarea bajo ciertas condiciones pre-planificadas de trabajo. Estas pruebas también son utilizadas para validar y verificar diferentes aspectos de la calidad del software, como por ejemplo: escalabilidad, fiabilidad y el buen uso de los recursos. Las pruebas se esfuerzan en mejorar el rendimiento y sirven para diferentes propósitos. Pueden comparar dos sistemas para encontrar cuál de ellos funciona

mejor o pueden medir que partes del sistema o de carga de trabajo provocan que el conjunto rinda mal (31).

Se aplicaron pruebas de rendimiento con el objetivo de probar el rendimiento del Subsistema de Administración que se propone en relación con el Subsistema de Administración Web que posee PRIMICIA. Se realizaron específicamente las pruebas de carga, que es un tipo de prueba de rendimiento que se realiza generalmente para observar el comportamiento de una aplicación bajo una cantidad de peticiones esperadas. Esta prueba puede mostrar los tiempos de respuesta de todas las transacciones importantes de la aplicación.

Se iniciaron los procesos de redacción y gestión de medias correspondientes a ambos subsistemas y se realizó un monitoreo del tiempo de respuesta ante diferentes acciones realizadas. Los procesos se ejecutaron en una estación de trabajo con iguales características para ambos subsistemas, lo que permitió demostrar que el subsistema desarrollado con tecnologías de escritorio tiene mayor rendimiento que el subsistema web que posee la plataforma. A continuación se muestran las características de la estación, además en la tabla 8 se recoge el tiempo de respuesta de cada subsistema ante acciones realizadas. Lo cual permitió afirmar que el Subsistema de Administración desarrollado tiene mejor rendimiento que Subsistema de Administración Web.

Estación de Trabajo:

- ✓ Sistema Operativo:
 - Ubuntu 12.04 LTS.
- ✓ Hardware:
 - Memoria: 1GB.
 - Procesador: Intel Core 2 Duo a 2.2 GHz.

Tabla 8. Tiempo de Respuesta para el Subsistema de Administración Web y en el Subsistema de Administración de Escritorio.

Procesos Ejecutados	Acciones Realizadas	Tiempo de Respuesta (ms) en el Subsistema de Administración de Escritorio	Tiempo de Respuesta (ms) en el Subsistema de Administración Web
	Listar Noticia.	450	2150

Redacción de la Noticia.			
	Guardar Noticia.	515	2470
	Editar Noticia.	380	1650
Gestión de Medias.	Cargar Imagen (2 imágenes).	60	80
	Cargar Video (2 videos).	800	1690

3.8 Conclusiones

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

1. Durante la etapa de análisis y diseño del Subsistema de Administración para PRIMICIA v2.0 se realizó la confección de los diagramas de Clases del Diseño, los que han sido presentados en el actual capítulo. Los mismos representan la visión de todos los requisitos detectados que han sido materializados como las funcionalidades del problema planteado.
2. El Modelo de Implementación posibilitó completar la construcción del software y mostrar las relaciones entre los elementos de implementación. El Modelo de Despliegue permitió describir la distribución física del sistema.
3. Con la realización de las pruebas de caja negra y sus resultados satisfactorios en la última iteración de prueba se puede concluir que el software no presenta ningún error funcional. Por su parte con las pruebas de integración realizadas se puede concluir que el Subsistema de Administración y el Subsistema de Transmisión de PRIMICIA v2.0 interactúan correctamente a través de sus interfaces.
4. Las pruebas de rendimiento realizadas permiten concluir que el Subsistema de Administración que se propone presenta un mayor rendimiento que el Subsistema de Administración Web. Logrando así solucionar el problema que dio origen a la presente investigación.

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 la solución del problema a resolver. 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. El uso del marco de trabajo Qt permitió crear un software de calidad y con una mayor reutilización de código. Así como el uso de las bibliotecas que fueron una pieza fundamental en el desarrollo de la aplicación.
3. La selección del Modelo en Capas permitió tener una correcta estructura para los elementos de software, simplificar la complejidad del mismo y tener una organización de los elementos en capas.
4. La implementación del Subsistema de Administración para PRIMICIA v2.0 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.

La aplicación desarrollada otorga valor añadido al producto PRIMICIA, brindándole la posibilidad de adquirir nuevos clientes al ser una nueva versión del Subsistema de Administración; desarrollada con tecnologías de escritorio.

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 para la utilización de la misma debido al próspero, necesario y fundamental avance tecnológico, que engloban a la informática como ciencia, conjuntamente con su plano ingenieril:

- Aplicar nuevas funcionalidades al subsistema, entre ellas los efectos de entrada y salida a los componentes de la noticia.
- Ampliar la creación de módulos a la aplicación, entre ellos el módulo de infocinta y el de gestión de usuarios.

Bibliografía y Referencias Bibliográficas

1. **Gómez, Dorgis Montes De Oca.** Desarrollo del módulo de Redacción de Noticias para la Plataforma de Televisión Informativa PRIMICIA v2.0. 2012.
2. **Ruber Hernández Gracia, Yunior Montaner Hernández.** Sistema Automatizado de Teletexto para la Plataforma de Televisión Digital Satelital Cubana. Ciudad de La Habana : s.n., 2007.
3. **Pressman.** Ingeniería Web. Sexta Edición. Capítulo 16.
4. Real Academia Española. Real Academia Española. [En línea] 2009. [Citado el: 29 de enero de 2013.] <http://lema.rae.es/drae/?val=sistema>.
5. **Abel Díaz Berenguer. Alberto Ramón Román.** Sistema de Administración y Configuración de la solución de Captura y Catalogación de Medias. Habana : s.n., 2009.
6. **Rumbaugh, James, Jacobson, Ivar y Booch, Grady.** El Proceso Unificado de Desarrollo de Software. Madrid : s.n., 2010.
7. **González, Carlos de Jesús Andrés.** Diseño del Subsistema de Configuración de la Plataforma de Televisión Informativa, PRIMICIA. Ciudad de La Habana : s.n., 2010.
8. Real Academia Española. Real Academia Española. [En línea] 2012. [Citado el: 12 de enero de 2013.] http://buscon.rae.es/drael/SrvltConsulta?TIPO_BUS=3&LEMA=modulo.
9. **Chiavenato, Idalberto.** Introducción a la Teoría General de la Administración, Séptima Edición. Bogotá: McGraw-Hill Interamericana : s.n., 2004.
10. **Coulter, Stephen Robbins y Mary.** Administración, Octava Edición, Pearson Educación. 2005.
11. **Systems, Estructure Media.** Estructure. Estructure. [En línea] 2012. <http://www.estructure.es/enewsroom.htm>.
12. **Dayara Cobas Peláez, Dailyn Pérez Grave de Peralta.** Propuesta de Migración a Software Libre de Señal ACN. 2008.
13. **Jorge Daniel Olivares Tamayo, Bernardo Rey Almaguer.** Desarrollo del Canal Informativo del Ministerio del Poder Popular para la Energía y Petróleo de Venezuela: Subsistema de Administración. 2008.

14. **Cabrera, Lisandra Delgado.** Diseño de las nuevas funcionalidades del Módulo de Redacción de la Plataforma de Televisión Informativa PRIMICIA. Ciudad de La Habana : s.n., 2010.
15. Sonaps. Sonaps. [En línea] 2007.
<http://www.sony.es/pro/product/networkprod/sonaps/features#features>.
16. **Gil, Aros Celio.** RUP: Metodología en los sistemas y aplicaciones basadas en la web. 2008.
17. Una Introducción a los Perfiles UML. [En línea] Lidia Fuentes, A.V.
www.lcc.uma.es/~av/Publicaciones/04/UMLProfiles-Novatica04.pdf.
18. <http://www.visual-paradigm.com/Visual-paradigm>. **Visual-paradigm.** Available from: Visual paradigm. [En línea] www.visual-paradigm.com.
19. Definición.org. Definición.org. [En línea] www.definicion.org/lenguaje-de-programacion.
20. S.L. Mora, C++ paso a paso. [En línea] Available from: <http://gpsl.dlsi.ua.es/~slujan/materiales/cpp-muestra.pdf>.
21. **Katrib, M.** Programación Orientada a Objetos en CPP.
22. **C.Larman.** UML y PATRONES Introducción al análisis y diseño orientado a objeto. 1999.
23. **Sommerville, Ian.** Ingeniería del Software. Madrid: Pearson Educación : s.n., 2002.
24. Pressman. 2005.
25. **Torossi., Gustavo.** El Proceso Unificado de Desarrollo de Software. 2008.
26. **Torres, F.** Integración del PMBOK al RUP para proyectos de Desarrollo de Software. 1998.
27. **Shaw, David Garlan y Mary.** An Introduction to Software Architecture. New Jersey : s.n., 1994.
28. **Christopher, Alexander.** The Timeless Way of Building. Oxford University Press. : s.n. 0195024028.
29. Capacitación y guía para el desarrollo de software.It-Mentor.
30. Corporación Sybven-Integración Tecnológica. [En línea] 2013.
http://www.corporacionsybven.com/portal/index.php?option=com_content&view=article&id=246.
31. **Cosín, J.D.** Técnicas cuantitativas para la gestión en la ingeniería del software. 2007.

Glosario

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.

XML: proviene de eXtensible Markup Language (“Lenguaje de Marcas Extensible”). Se trata de un metalenguaje (un lenguaje que se utiliza para decir algo sobre otro lenguaje) extensible de etiquetas que fue desarrollado por el Word Wide Web Consortium (W3C), un consorcio internacional que elabora recomendaciones para la World Wide Web.

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.