

**UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS**

**Facultad 6**



**Título:** Visor de noticias de la plataforma PRIMICIA para dispositivos móviles.

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

**Autor:** Leonel Martínez Pulido

**Tutor(a):** Adrianet Ramos Velazco

La Habana, Junio 2015

“Año de 57 de la Revolución”

## **Declaración de autoría**

Declaro que soy el único autor del trabajo “**Visor de noticias de la plataforma PRIMICIA para dispositivos móviles**”, y autorizo a la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

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

---

**Autor:** Leonel Martínez Pulido

---

**Tutor:** Adrianet Ramos Velazco

## **Datos de contacto**

**Tutor:** Ing. Adrianet Ramos Velazco

**Formación académica:**

Ingeniero en Ciencias Informáticas, Universidad de Ciencias Informáticas, 2014.

**Centro Laboral:**

Centro de Desarrollo de Geoinformática y Señales Digitales. Facultad 6.

**Correo electrónico:** [arvelazco@uci.cu](mailto:arvelazco@uci.cu)

## **Agradecimientos**

*Muchas gracias a mi mamá, a mi papá, a mi hermana, a mi novia Yanetzi, a Yuned, a toda la gente del palenque y a todo el que de una forma u otra me apoyó.*

## **Dedicatoria**

*Se lo dedico en especial a mi mamá, a mi papa y a toda mi familia.*

*Leonel*

## Resumen

Las noticias son una forma de los medios de difusión de informar, comunicar a las personas acerca de cualquier suceso ocurrido a su alrededor. Con la evolución de las nuevas tecnologías a través de los medios de visualización se modificó la manera de acceder y leer las noticias de los medios tradicionales. Ahora se le pueden añadir audio, video o imágenes y publicarlas en la web a la cual tienen acceso millones de personas, convirtiéndose en un nuevo canal de comunicación para el intercambio de información. PRIMICIA es un sistema encargado de transmitir informaciones de manera inmediata y constante en distintos formatos a través de una red de televisión, adecuándose a las diferentes vías de transmisión existentes. Este sistema solo permite que las personas tengan acceso a las noticias cuando están situadas frente a un televisor. Este trabajo tiene como objetivo, a través del uso de las tecnologías móviles, desarrollar un Visor de Noticias de la plataforma PRIMICIA para dispositivos móviles que permita mostrar una lista de las noticias que se publiquen, en cualquier momento. Para el desarrollo del mismo se seleccionó Android como sistema operativo, para guiar el proceso de desarrollo de software se utilizó como metodología RUP, se utilizó el servicio web REST para el intercambio de información a través de diferentes aplicaciones, para la implementación se utilizó como lenguaje de programación JAVA en conjunto con el IDE de programación ECLIPSE y el SDK de Android. Se aplicaron pruebas de funcionalidad para probar el correcto funcionamiento del sistema.

**Palabras claves:** dispositivos móviles, Visor de Noticias, PRIMICIA, tecnologías móviles.

## **Abstract**

*The news is a form of the mass media to inform, inform people about any incident occurring around them. With the evolution of new technologies through the display means how to access and read the news from traditional media was changed. Now you can add audio, video or images, and publish them on the web which is accessible to millions of people, becoming a new communication channel for the exchange of information. PRIMICIA is a system responsible for transmitting information immediately and steadily in various formats via a television network, adapting to the different routes of existing transmission. This system only allows people to have access to news when placed in front of a television. This work aims, through the use of mobile technologies, develop a viewer News PRIMICIA platform for mobile devices that can display a list of news published at any time. For the development of the Android was selected as the operating system, to guide the process of software development was used as RUP, the REST web service for the exchange of information was used across different applications, to implement was used as a language Java programming together with the Eclipse IDE and the Android SDK programming. Functionality tests were used to test the operation of the system.*

**Keywords:** *mobile devices, Viewer News, PRIMICIA, mobile technologies.*

## Índice de contenido

Introducción .....	1
Capítulo I: Fundamentos teóricos .....	4
1.1 Medios de visualización de noticias.....	4
1.2 Tecnologías y dispositivos móviles.....	4
1.2.1 <i>Sistema operativo</i> .....	6
1.2.2 <i>Servicio Web</i> .....	9
1.3 PRIMICIA .....	12
1.4 Visores móviles .....	13
1.5 Metodología de desarrollo del software .....	15
1.6 Herramientas y tecnologías.....	16
1.6.1 <i>Framework</i> .....	17
1.6.2 <i>Lenguajes de programación</i> .....	18
1.6.3 <i>Entorno de desarrollo</i> .....	19
1.6.4 <i>Gestor de bases de datos</i> .....	21
1.6.5 <i>Entorno de desarrollo web</i> .....	22
1.6.6 <i>Lenguaje de Modelado</i> .....	23
1.6.7 <i>Herramienta CASE</i> .....	24
1.7 Conclusiones del capítulo.....	25
Capítulo II: Presentación de la propuesta de solución.....	26
2.1 Propuesta de solución .....	26
2.2 Modelo de Dominio .....	27

---

2.3	Requisitos del sistema.....	28
2.3.1	<i>Requisitos funcionales</i> .....	28
2.3.2	<i>Requisitos no funcionales</i> .....	30
2.4	Casos de usos del sistema.....	31
2.4.1	<i>Definición de los actores del sistema</i> .....	31
2.4.2	<i>Diagrama de casos de uso</i> .....	31
2.4.3	<i>Descripción de casos de uso</i> .....	32
2.5	Modelo de Diseño.....	34
2.5.1	<i>Diagrama de clases del diseño</i> .....	34
2.6	Patrones utilizados.....	37
2.6.1	<i>Patrones Arquitectónicos</i> .....	37
2.6.2	<i>Patrones de Diseño</i> .....	38
2.7	Diagrama de Secuencia.....	41
	Conclusiones del capítulo.....	42
	Capítulo III: Implementación y prueba del visor de noticias.....	43
3.1	Modelo de implementación.....	43
3.1.1	<i>Diagrama de Componentes</i> .....	43
3.2	Modelo de Despliegue.....	44
3.3	Código Fuente.....	45
3.3.1	<i>Estándar de codificación</i> .....	46
3.4	Pruebas de software.....	46
3.4.1	<i>Niveles de pruebas</i> .....	47
3.4.2	<i>Diseño de Casos de Prueba</i> .....	48
3.4.3	<i>Resultados de las pruebas</i> .....	50

3.5 Conclusiones del capítulo.....	51
Conclusiones generales.....	52
Recomendaciones .....	53
Bibliografía.....	54
Anexos.....	58
Glosario de Términos.....	72

## Índice de tablas

Tabla 1. Tabla comparativa entre Android OS, iOS, Windows Phone 7 y BlackBerry OS. ....	8
Tabla 2. Actores del sistema. ....	31
Tabla 3. Caso de uso: Mostrar noticias .....	32
Tabla 4. Descripción de las clases. ....	35
Tabla 5. Descripción de las variables asociadas al caso de uso Buscar Noticias.....	48
Tabla 6. Caso de prueba del caso de uso Buscar Noticia por Título .....	50
Tabla 7. Diseño del caso de prueba Buscar Noticia por fecha .....	63
Tabla 8. Descripción de las variables asociadas al caso de uso Visualizar Transmisión.....	64
Tabla 9. Diseño del caso de prueba Cambiar configuración .....	64
Tabla 10 Caso de Uso: Listar Noticias .....	66
Tabla 11 Caso de Uso: Buscar Noticias.....	67
Tabla 12 Descripción del CU: Visualizar Noticias.....	69

## Índice de figuras

Figura 1 Propuesta de Solución .....	27
Figura 2. Modelo de Domino de Visor de Noticias.....	28
Figura 3. Diagrama de Casos de uso.....	32
Figura 4. Diagrama de clases del diseño. ....	35
Figura 5. Representación del patrón Experto. ....	39
Figura 6. Representación del patrón Interpreter.....	40
Figura 7. Diagrama de Secuencia Mostrar Noticias. ....	42
Figura 8. Diagrama de Componentes Caso de Uso Mostrar Noticia. ....	44
Figura 9. Diagrama de Despliegue.....	45
Figura 10. Estándar de codificación. ....	46
Figura 11. Diagrama de Secuencia del Caso de Uso Listar Noticia. ....	58
Figura 12. Diagrama de Secuencia del Caso de Uso Visualizar Transmisión. ....	58
Figura 13. Diagrama de Secuencia del Caso de Uso Buscar Noticia. ....	59
Figura 14. Diagrama de Componentes del Caso de Uso Listar Noticia. ....	59
Figura 15. Diagrama de Componentes del Caso de Uso Buscar Noticia.....	60
Figura 16. Diagrama de Componentes de Caso de Uso Visualizar Transmisión.....	61
Figura 17. Diagrama de Clases CU Listar.....	62
Figura 18. Diagrama de Clases CU Buscar.....	62
Figura 19. Diagrama de Clases CU Visualizar Transmisión. ....	63

## Introducción

Las noticias son una forma que poseen los medios de difusión masivos para mantener informadas a las personas de forma simultánea e inmediata desde cualquier lugar. Con la evolución de las tecnologías de la información y las comunicaciones se modificó la manera de acceder y leer dichas noticias de los medios tradicionales. Ahora se le pueden añadir audio, video o imágenes y publicarlas en la web a la cual tienen acceso millones de personas, convirtiéndose en un nuevo canal de comunicación para el intercambio de información (Castagnola, 2009).

El propósito de la noticia es informar, comunicar a las personas acerca de acontecimientos actuales que han ocurrido en la localidad donde se vive, en la región, país o en el mundo, de manera veraz, clara y concisa. Producto a los constantes sucesos sociales, económicos, políticos que ocurren alrededor del hombre, es de gran importancia mantenerse informados por lo que la noticia es una parte fundamental en su vida diaria ya que puede influir, de manera positiva o negativa, en la toma de decisiones además de proveer conocimiento.

La UCI cuenta con varios centros de producción destinados al desarrollo de productos de software; uno de estos es el centro de desarrollo Geoinformática y Señales Digitales (GEYSED), el cual tiene como objetivo el desarrollo de sistemas informáticos de transmisión de señales digitales. La Plataforma de Televisión Informativa PRIMICIA, es uno de los productos que este centro desarrolla, el cual permite la transmisión automática y constante de noticias acompañadas de medias audiovisuales en un canal de televisión. Dicho producto se encuentra estructurado en dos subsistemas, el Subsistema Administración y Transmisión.

PRIMICIA limita a las personas interesadas a permanecer cerca de un televisor hasta que la noticia de su interés sea mostrada. Esto reduce el número de actividades humanas a realizar, provocando que una persona al no poder visualizar una noticia que se transmite por un canal televisivo determinado, pudiera quedar desinformado debido a que las noticias sólo se pueden observar en los televisores que estén mostrando el canal en ese instante.

A partir de lo anteriormente expuesto se plantea como **problema a resolver**: la plataforma informativa PRIMICIA solo permite que las personas tengan acceso a las noticias cuando están situadas frente a un televisor. Estableciéndose como **objeto de estudio**: los medios de visualización de noticias.

Con el propósito de dar solución al problema anteriormente planteado, se ha trazado como **objetivo general** de la investigación desarrollar un visor de noticias que posibilite la accesibilidad a las noticias de la Plataforma de Televisión Informativa PRIMICIA desde cualquier lugar con el empleo de dispositivos de acceso a la información. Enmarcado en el **campo de acción**: los medios de visualización de noticias a través de dispositivos Android.

Para darle solución al problema planteado se definen las siguientes **preguntas científicas**:

- ❖ ¿Cuáles dispositivos hacen posible la portabilidad de la información?
- ❖ ¿Cuáles son las dificultades que presentan las soluciones existentes?
- ❖ ¿Cómo acceder desde un dispositivo móvil con sistema operativo Android a la información contenida en la Plataforma de Televisión Informativa PRIMICIA?

Para lograr el **objetivo general** se tendrán en cuenta las siguientes **tareas de la investigación**:

- ❖ Análisis de los conceptos básicos y técnicos implicados en el desarrollo del visor de noticias.
- ❖ Caracterización de las soluciones existentes.
- ❖ Caracterización de la metodología y las herramientas a utilizar en el proceso de desarrollo del visor de noticias.
- ❖ Elaboración del modelo de casos de uso, de diseño y de implementación del visor de noticias.
- ❖ Desarrollo del visor de noticias para dispositivos móviles.
- ❖ Validación del aporte logrado a la visualización de las noticias de PRIMICIA con la introducción del visor para dispositivos móviles.

La presente investigación exigió la utilización de los **métodos científicos** empíricos y teóricos:

**Teóricos:**

- ❖ **Analítico-Sintético**: este método permitió estudiar la manera en que se visualizan las noticias en la plataforma PRIMICIA, facilitando la extracción de los elementos más importantes que se relacionan con el objeto de estudio, además facilita el estudio de las herramientas y tecnologías a utilizar.

- ❖ **Histórico-Lógico:** permite realizar un análisis profundo sobre la evolución que han tenido los dispositivos de acceso a información, principalmente los que utilizan Android como sistema operativo, mostrando sus características principales y su evolución tecnológica en los últimos cinco años.

**Empíricos:**

- ❖ **Entrevista:** se emplea para identificar las funcionalidades que debe cumplir el visor de noticias, así como los elementos relevantes a tener en cuenta para el desarrollo de la investigación. Esta entrevista se le realizó al líder del proyecto PRIMICIA.

El presente documento está constituido por tres capítulos fundamentales para dar solución a la problemática existente:

**Capítulo I: Fundamentos teóricos**

Constituye la base teórica de la investigación, en la cual son expuestos los principales conceptos que contribuyen al mejor entendimiento del problema en cuestión. Se describen los principales aspectos de las herramientas, tecnologías y metodología a emplear durante el desarrollo de la misma.

**Capítulo II: Presentación de la propuesta de solución**

En este capítulo se describe el análisis y diseño de las clases del sistema. Se hace un levantamiento de los requisitos funcionales y no funcionales necesarios para lograr que el sistema funcione correctamente. Se construyen los artefactos correspondientes al análisis y diseño, modelando la estructura del visor.

**Capítulo III: Implementación y prueba del visor de noticias**

En este capítulo se muestra el modelo de implementación como resultado del diseño anteriormente desarrollado, así como el diagrama de componentes del visor de noticias. Se describen las pruebas a realizar, con el objetivo de comprobar el correcto funcionamiento del visor.

## **Capítulo I: Fundamentos teóricos**

En el presente capítulo se realiza el estudio de los contenidos teóricos que sustentan la investigación. De forma general se tratan conceptos ligados al ámbito de la investigación y la selección de las diferentes tecnologías, herramientas y lenguajes a utilizar en todo el proceso de desarrollo.

### **1.1 Medios de visualización de noticias**

Según la Real Academia Española define como un medio: *“Modo o instrumento para conseguir algo”* además se define como visualización: *“Acción y efecto de visualizar”*. Por tanto se puede definir como medios de visualización de noticias al modo o instrumento que permite la visualización de una noticia determinada.

Existen diferentes medios de visualización de noticias dentro de los que se encuentran:

- ❖ Televisor
- ❖ Monitor
- ❖ *SmartPhones*
- ❖ *Tablets*
- ❖ *Laptop*

Estos tres últimos clasifican como dispositivos móviles los cuales actualmente hacen que el acceso a la información se pueda lograr desde cualquier lugar.

### **1.2 Tecnologías y dispositivos móviles**

El término “Tecnología Móvil” hace referencia a la posibilidad de trasladar una actividad determinada, que normalmente se inscribe en un espacio físico, de un sitio a otro. Es decir, se puede llevar a cabo una tarea determinada sin estar físicamente en la oficina, y efectuar un sin número de actividades potenciadas por el auge de nuevos dispositivos suficientemente pequeños (Pastrana, 2013).

Se puede decir que las tecnologías móviles son el conjunto de dispositivos y herramientas que dan la posibilidad de realizar cualquier actividad, que normalmente se realiza en un espacio físico, en cualquier lugar.

**Ventajas de la tecnología móvil (Pastrana, 2013):**

- ❖ Aumento de la productividad: la posibilidad de acceder a la información y realizar cualquier trabajo con la misma desde cualquier lugar, significa un ahorro de tiempo y presupuesto. Es posible minimizar el tiempo de inactividad ocasionado por viajes o interrupciones que llevan siempre las principales herramientas de productividad.
- ❖ Flexibilidad y rapidez: las soluciones móviles permiten llevar las aplicaciones informáticas y de gestión de la información a cualquier lugar, lo que significa que se podrá dar respuesta a las distintas expectativas con total flexibilidad.
- ❖ Portabilidad: es la capacidad de trasladar el número de teléfono móvil de un operando a otro, manteniendo el mismo número. Mientras que en sus servicios influye la adaptabilidad, escalabilidad, facilidad al instalar y conformidad.
- ❖ Usabilidad: la interacción entre el usuario y el dispositivo debe ser sencilla y fácil de usar.

Las tecnologías móviles han evolucionado con el paso del tiempo dando como resultado el surgimiento de nuevos dispositivos con capacidades computacionales, de reducido tamaño con posibilidad de conexión, conocidos actualmente como dispositivos móviles.

Los dispositivos móviles (también conocidos como computadora de mano) son aparatos de pequeño tamaño, con algunas capacidades de procesamiento, con conexión permanente o intermitente a una red. Tienen memoria limitada, diseñados específicamente para una función, pero que pueden llevar a cabo otras funciones más generales (Valencia, 2011).

Existen diferentes formas y criterios para clasificar los distintos dispositivos móviles que existen, una de estas formas de clasificación, es la de agruparlos según la función que desempeñan. La clasificación es la siguiente: dispositivos de comunicación, computación, reproductores multimedia, grabador multimedia consola portátil, consola portátil. Otra forma de definir los tipos de dispositivos móviles es la siguiente: dispositivo móvil de datos limitados, de datos básicos, de datos mejorados (Pérez, 2014).

Los dispositivos móviles están llegando a todos los niveles de nuestra sociedad, no sólo en temas relacionados con la comunicación entre personas también a nivel empresarial, estableciéndose como una nueva plataforma para el desarrollo de aplicaciones en diversos campos, como pueden ser las nuevas formas de comercio, la publicidad, las redes sociales y los medios de comunicación, convirtiéndose en herramientas útiles para facilitar el desarrollo tecnológico y la vida cotidiana del hombre.

### **1.2.1 Sistema operativo**

Existe una enorme y variada gama de sistemas informáticos para los que se diseñan sistemas operativos (Stallings, 1997). Tal es el caso de los teléfonos móviles que tratan de llevar más lejos el concepto de telefonía añadiendo a los terminales móviles funciones propias de los ordenadores. Los dispositivos móviles de la actualidad proporcionan una serie de servicios basados en el tráfico de datos a través de la red, por lo que necesitan un sistema operativo propio a este tipo de escenario.

### **Comparación entre Android y otros sistemas operativos para dispositivos móviles**

Los creadores de la tecnología Android han desarrollado un sistema operativo que cualquier fabricante puede licenciar e incluir en su dispositivo, sin unos requerimientos mínimos de hardware ni límites en la personalización de la interface de usuario. Su principal ventaja es la cantidad, gracias a este modelo el mercado puede ser inundado con cientos de modelos distintos y el usuario tiene libertad para elegir entre terminales de gama baja, media o alta (Ferrás, 2012).

### **Android vs Blackberry OS**

En el caso de Blackberry en la mayoría de los aspectos no supera a sus competidores. Existen poca cantidad de aplicaciones disponibles, no soporta múltiples cuentas de intercambio de mensajes, la captura y reproducción de videos realmente no es de buena calidad, como tampoco lo son la pantalla y su resolución para alta definición de imágenes, además que no permite video-llamadas, además la plataforma está un poco desorganizada y suele tardar un tiempo en ofrecer las actualizaciones para los equipos (Ferrás, 2012).

### **Android vs iOS**

La gran limitante en este caso es el férreo control sobre el hardware, siendo *Apple* la encargada de diseñar tanto el sistema iOS como el dispositivo donde se ejecuta. La principal ventaja de este modelo es la adaptación total del sistema operativo al dispositivo, puesto que ambos han sido concebidos como partes de un todo, por lo tanto ofrecen la misma experiencia de uso para todos los consumidores. Otras desventajas importantes con respecto a Android son el coste de adquisición y la nula variedad de dispositivos en el mercado, solo existe uno y no hay opciones de elección (Ferrás, 2012).

### **Android vs Windows Phone 7 OS**

Como fabricante del sistema, *Microsoft* requiere que todo teléfono que desee ejecutar *Windows Phone 7* disponga de unas características mínimas, para asegurar la consistencia de todos los usuarios del sistema, a partir de estas características los fabricantes de software son libres de ampliarlas en algunos casos y están obligados a cumplirlas con exactitud en otros. En este modelo se unen las ventajas del modelo *iPhone*, todos los usuarios obtienen la misma experiencia de uso y los desarrolladores saben que su aplicación funcionará de forma idéntica en todos los dispositivos *Windows Phone 7*, y las del modelo Android, no hay limitante a un solo hardware, existen distintos dispositivos de varios fabricantes, todos con unas características mínimas comunes pero con suficientes diferencias como para sentir que el cliente escoja el terminal que más se ajuste a su gusto (Ferrás, 2012).

El problema para los programadores que deseen utilizar aplicaciones creadas por ellos mismos, es que la única forma de instalar una aplicación en *Windows Phone 7* es mediante la tienda oficial de *Microsoft: Marketplace*; en la cual se deben registrar como desarrolladores para poder vender sus aplicaciones. Es decir si *Microsoft* no compra la aplicación de determinado desarrollador, este nunca podrá utilizarla, ni siquiera en su propio dispositivo (Ferrás, 2012).

A continuación se observa una tabla comparativa entre los sistemas operativos antes mencionados, donde la presencia del icono representativo de cada uno, indica su superioridad sobre los otros en determinado aspecto (Ferrás, 2012):

Tabla 1. Tabla comparativa entre Android OS, iOS, *Windows Phone 7* y BlackBerry OS.

	 Android	 Windows Phone	 IOS	 Blackberry
<b>Actualizaciones</b>				
<b>Aplicaciones</b>				
<b>Construcción de hardware</b>				
<b>Código abierto</b>				
<b>Navegación en internet</b>				
<b>Usabilidad y diseño</b>				

Teniendo en cuenta las comparaciones entre Android y otros sistemas operativos para dispositivos móviles se decide su uso para el desarrollo del Visor de Noticia. A continuación se justifica su uso.

### Android

Android es un sistema operativo orientado a terminales móviles basado en el núcleo de Linux. Permite controlar dispositivos por medio de bibliotecas desarrolladas o adaptadas por Google mediante el lenguaje de programación Java. Es una plataforma de código abierto, esto quiere decir que cualquier desarrollador puede crear aplicaciones y compilarlas a código nativo de ARM (API de Android). Aunque la mayoría de las aplicaciones están escritas en Java, no hay una máquina virtual Java en la plataforma. El *bytecode* no es ejecutado, sino que primero se compila en un ejecutable *Dalvik* que es una máquina virtual diseñada específicamente para este sistema operativo.

### Características de Android:

- ❖ Incluye un emulador de dispositivos, herramientas para depuración de memoria y análisis de rendimiento de software.

- ❖ El entorno de desarrollo que utiliza es *Eclipse*, usando el plugin de herramientas de desarrollo del propio Android.
- ❖ El *framework* de aplicaciones que utiliza permite el reemplazo y la reutilización de los componentes.
- ❖ Posee un navegador integrado basado en el motor *Open Source Webkit*.
- ❖ Utiliza una base de datos liviana llamada *SQLite*, la cual realiza un almacenamiento estructurado que se integra directamente con las aplicaciones.
- ❖ Tiene disponible el sistema multitarea real de aplicaciones, es decir, las aplicaciones que no están ejecutándose en primer plano reciben ciclos de reloj (Báez, y otros, 2011).

### Ventajas de Android

- ❖ **Open Source.** Lo que hace que Android brille sobre el resto de la multitud es su fuente abierta. Esto atrae a muchos programadores y desarrolladores de todos los ámbitos de la vida, ya que *Google* distribuye libremente su código fuente para el *kernel* de Linux y el código es de licencia pública para cualquier persona.
- ❖ **Aplicaciones.** Hay una aplicación para casi cualquier situación, cientos de miles de aplicaciones en el *Android Market* para atender cualquier necesidad.
- ❖ **Hardware.** Android tiene las opciones más numerosas de los *smartPhones* hasta la fecha. Android es lo suficientemente flexible como para funcionar en casi todos los teléfonos inteligentes, siempre y cuando coincida con los requisitos mínimos.
- ❖ **Software.** Android puede ejecutar múltiples aplicaciones al mismo tiempo, ya que tiene espacio de intercambio de gestión. También se ha construido en forma nativa *ActiveSync*, que le permite sincronizar fácilmente los contactos, correo electrónico e información personal una vez que el dispositivo está sincronizado con un ordenador de sobremesa (Báez, y otros, 2011).

#### 1.2.2 Servicio Web

Los servicios web surgen como consecuencia de la necesidad de crear un estándar para la comunicación entre las distintas plataformas y lenguajes de programación existentes.

Un servicio web es un componente al que se puede acceder mediante protocolos web estándar, utilizando XML para el intercambio de información. Los servicios web son componentes de aplicaciones distribuidas que están disponibles de forma externa. Se pueden utilizar para integrar aplicaciones escritas en diferentes lenguajes y que se ejecutan en plataformas diferentes. Los servicios Web son independientes de lenguaje y de la plataforma gracias a que los vendedores han admitido estándares comunes de servicios web (IA, 2013).

Se puede decir que un servicio web no es más que un conjunto de protocolos y estándares que se utilizan para el intercambio de información entre distintas aplicaciones desarrolladas en diferentes lenguajes y se ejecuten diferentes plataformas.

### **Servicio Web REST**

REST, forma abreviada para Transferencia de Estado Representacional, es un protocolo basado en cliente y servidor de comunicaciones de recursos que es cacheable y sin estado. Es una abstracción sobre la comunicación HTTP donde se da énfasis en la interfaz uniforme para hacer la comunicación estructurada y coherente (Manandhar, 2014).

Es un estilo arquitectónico, y un acercamiento a las comunicaciones que se utiliza a menudo en el desarrollo de servicios web. El uso de REST se prefiere a menudo sobre el estilo más pesado de SOAP (*Simple Object Access Protocol*) porque REST no aprovecha la mayor cantidad de ancho de banda, lo que hace que sea una mejor opción para su uso a través de Internet. El enfoque de SOAP requiere escribir o usando un programa de servidor proporcionado (para servir de datos) y un programa de cliente (para solicitar datos). REST consiste en leer una página Web designada que contiene un archivo XML. El archivo XML describe e incluye el contenido deseado. Una vez definido dinámicamente, los consumidores pueden acceder a la interfaz. (Rouse, 2014).

REST describe un conjunto de principios arquitectónicos por los cuales los datos pueden ser transmitidos a través de una interfaz estandarizada (como HTTP). REST no contiene una capa adicional de mensajería y se centra en las normas de diseño para la creación de servicios sin estados. Un cliente puede acceder al recurso mediante un único URI y una representación del recurso es devuelta. Si bien el acceso a los recursos REST con el protocolo HTTP, la dirección URL del recurso sirve como identificador de recursos y *GET*, *PUT*, *DELETE*, *POST* y *HEAD* son las operaciones HTTP estándar que se deben realizar en ese recurso.

**Pros de REST** (NordicAPIs, 2015):

- ❖ Sigue la filosofía del *Open Web*.
- ❖ Relativamente fácil de implementar y mantener.
- ❖ Separa claramente de cliente y servidor las implementaciones.
- ❖ La comunicación no está controlada por una sola entidad.
- ❖ La información puede ser almacenada por el cliente para evitar múltiples llamadas.
- ❖ Se puede devolver datos en múltiples formatos (JSON, XML, etc.).

**Contras de REST** (NordicAPIs, 2015):

- ❖ Sólo funciona en la parte superior del protocolo HTTP.
- ❖ Difícil de hacer cumplir la autorización y la seguridad en la parte superior de la misma.

Es importante saber cuándo usar o no este servicio por las características que tiene. Se puede usar cuando los clientes y servidores operan en un entorno Web o cuando la información de los objetos no tiene que ser comunicada al cliente y no se recomienda cuando se necesita hacer cumplir un contrato estricto entre cliente y servidor o al realizar transacciones de múltiples llamadas. Es mayormente utilizado por servicios de medios sociales, redes sociales, servicios de chat web y servicios móviles (NordicAPIs, 2015).

**Servicios Web SOAP (*Simple Object Access Protocol*)**

Es un protocolo de mensajería que permite a los programas que se ejecutan en diferentes sistemas operativos (como Windows y Linux) comunicarse utilizando el protocolo de transferencia de hipertexto (HTTP) y su lenguaje de marcado extensible (XML). SOAP especifica exactamente cómo codificar una cabecera HTTP y un archivo XML para que un programa en una computadora pueda llamar un programa en otro ordenador y transmitir información. SOAP también especifica cómo el programa llamado puede devolver una respuesta. A pesar de su emparejamiento frecuente con HTTP, SOAP soporta otros protocolos de transporte también (Rouse, 2014).

**Pros de SOAP** (NordicAPIs, 2015):

- ❖ Sigue un enfoque formal de la empresa.

- ❖ Trabaja en la parte superior de cualquier protocolo de comunicación, incluso de forma asíncrona.
- ❖ Información acerca de los objetos se comunica a los clientes.
- ❖ Seguridad y autorización son parte del protocolo.
- ❖ Puede ser descrito completamente usando WSDL.

**Contras de SOAP** (NordicAPIs, 2015):

- ❖ Pasa mucho ancho de banda de comunicación de metadatos.
- ❖ Es difícil de implementar e impopular entre los desarrolladores de dispositivos web y móviles.
- ❖ Utiliza sólo XML.

Es recomendable el uso de SOAP cuando los clientes necesitan tener acceso a los objetos disponibles en los servidores o cuando se quiere hacer cumplir un contrato formal entre el cliente y el servidor, pero cuando se desea que la mayoría de los desarrolladores utilicen fácilmente su API o cuando el ancho de banda es muy limitado no es muy recomendable (NordicAPIs, 2015).

Luego de analizar las características, los pros y los contras de estos servicios web, se seleccionó el servicio web REST ya que es el que más se adecua a las características del sistema a desarrollar.

### 1.3 PRIMICIA

Es un sistema multiplataforma, desarrollado sobre tecnologías libre, dicho producto se encuentra estructurado en dos subsistemas: Administración, encargado de la administración del canal y la gestión de noticias y recursos multimedia; Transmisión, responsable de transmitir las noticias y materiales publicados. Actualmente el proceso de transmisión se realiza de la siguiente forma: la aplicación primicia va a estar instalada en una pc la cual a través de una tarjeta de video exportadora que se encarga de convertir la señal de digital a analógica la envía hacia los televisores correspondientes.

Las principales funcionalidades que realiza este subsistema de Transmisión son (Sosa, 2012):

- ❖ Generar una cartelera, del ciclo de transmisión mostrando para cada noticia la sección temática y el titular, en el orden que se visualizarán.
- ❖ Visualizar noticias, compuestas por pantallas de tipo texto, texto-imagen, imagen y video.
- ❖ Reproducir fondo musical, mientras se muestran las noticias, excepto cuando se muestre un video.
- ❖ Mostrar cintillos informativos o infocintas, para promocionar eventos de última hora o acontecimientos de gran importancia.
- ❖ Mostrar información adicional a la noticia, la fecha, hora, tiempo restante de la pantalla y titular de la próxima noticia y sección temática.
- ❖ Transmitir televisión en vivo, proveniente de una señal externa.
- ❖ Mostrar patrón del canal, cuando éste se encuentre fuera de servicio.

## 1.4 Visores móviles

Los visores móviles son un tipo de aplicación utilizada para visualizar algún archivo o recurso como imágenes, noticias, multimedia y texto en los dispositivos móviles. Existen diferentes aplicaciones para mostrar noticias por ejemplo:

**Pulse:** Es la aplicación de noticias profesionales hecha a la medida. Esta aplicación galardonada con el premio *Apple Design* permite personalizar el acceso a las noticias que se quiere leer, explorar fácilmente contenido profesional de interés y compartir historias en las redes sociales favoritas (Pulse, 2015).

**Características** (Pulse, 2015).

- ❖ Lector de noticias original para Android.
- ❖ Permiten configurar las fuentes que más te interesan.
- ❖ Completa integración con *Facebook* y *Twitter*.
- ❖ Navega fácilmente entre la información del lector.
- ❖ Distribuye hasta 60 fuentes distintas en 5 páginas personalizables.
- ❖ Lee las noticias cuando no tengas conexión a Internet.

- ❖ Importa *feeds* RSS desde *Google Reader*.

**News Republic:** Un innovador agregador de noticias, posee una interfaz limpia y sensible de una gran cantidad de contenido de muchos canales diferentes, que abarcan Política, Estilo de Vida, Deportes, Tecnología, Cine, Música y más (Republic, 2015).

**Características de News Republic** (Republic, 2015):

- ❖ Servicio de noticias para Android.
- ❖ Contenidos personalizables.
- ❖ Más de 140 mil temas disponibles.
- ❖ Gran cantidad de fuentes.
- ❖ Modo sin conexión para recuperar automáticamente el contenido.

**Limitaciones de New Republic** (Republic, 2015):

- ❖ Requiere Android 1.5 o superior.

**Press (RSS Reader):** Press es un sencillo y elegante cliente RSS para los servicios de sincronización más populares. Es rápido y limpio, y da la libertad para disfrutar de las noticias favoritas (Press, 2014).

**Características de Press** (Press, 2014):

- ❖ No hay anuncios.
- ❖ Sincroniza con *Feedly*, *Feed Wrangler*, *Feedbin* y *Fever*.
- ❖ Notificaciones de fondo.
- ❖ Integración de legibilidad.
- ❖ Navegación vertical u horizontal para ver los artículos.
- ❖ Imagen con *zoom*.
- ❖ Cambio del estilo de fuente y tamaño de texto.
- ❖ Artículos con el texto alineado.
- ❖ Comparte los artículos que lees.

**Limitaciones de Press** (Press, 2014):

- ❖ Requiere Android 4.0 y versiones superiores.

Todas estas soluciones existentes tienen como característica común que consumen servicios RSS por lo que su utilización se ve limitada ya que PRIMICIA no brinda este servicio. Además debido a la poca información existente acerca de estas soluciones y al no tener acceso al código fuente de dichas aplicaciones para poder realizar los cambios necesarios y vincularlos al proyecto se decidió realizar la implementación del Visor Móvil de Noticias para la plataforma PRIMICIA.

El desarrollo de visores de noticias se ve reflejado en diversas funciones, tanto como para divulgar contenido como para visualizarlos. Para PRIMICIA, la creación del Visor de Noticias permitirá el acceso a las noticias que se estén publicando y en transmisión, en cualquier momento y desde cualquier lugar.

## 1.5 Metodología de desarrollo del software

Las metodologías de desarrollo de software surgen ante la necesidad de utilizar una serie de procedimientos, técnicas, herramientas y soporte documental a la hora de desarrollar un producto de software (Carrillo Pérez, 2008). Dichas metodologías representan un marco de trabajo que tiene entre sus principales funciones: guiar, planificar, estructurar, controlar, manipular y dirigir el proceso de desarrollo de sistemas de información.

### **RUP (*Rational Unified Process*)**

Aplica varias prácticas en el desarrollo de software, inicio, elaboración, construcción y transición. Suministra a todo el equipo con un fácil acceso a una base de conocimiento con guías, plantillas y herramientas para todas las actividades críticas del desarrollo de software. Esta metodología permite que todos los integrantes de un equipo de trabajo, conozcan y compartan el proceso de desarrollo, una base de conocimientos y los distintos modelos de cómo desarrollar el software utilizando un lenguaje de modelado común: UML (Carrillo Pérez, y otros, 2008).

Esta metodología define varias disciplinas dentro de las que se encuentran: Modelado de Negocios, Requerimientos, Análisis y Diseño, Implementación, Pruebas, Entrega, Configuración, Administración del Cambio, Administración de Proyectos y Ambiente (Jacobson, y otros, 2000).

RUP presenta 3 características que constituyen la esencia de todo el proceso de desarrollo (Booch, y otros, 2000):

- ❖ **Dirigido por los casos de uso:** los casos de uso reflejan lo que los usuarios futuros necesitan y desean, lo cual se capta cuando se modela el negocio y se representa a través de los requerimientos. A partir de aquí los casos de uso guían el proceso de desarrollo ya que los modelos que se obtienen, como resultado de los diferentes flujos de trabajo, representan la realización de los casos de uso.
- ❖ **Centrado en la arquitectura:** la arquitectura muestra la visión común del sistema completo en la que el equipo de proyecto y los usuarios deben estar de acuerdo, por lo que describe los elementos del modelo que son más importantes para su construcción, los cimientos del sistema que son necesarios como base para comprenderlo, desarrollarlo y producirlo económicamente. RUP se desarrolla mediante iteraciones, comenzando por los CU relevantes desde el punto de vista de la arquitectura. El modelo de arquitectura se representa a través de vistas en las que se incluyen los diagramas de UML.
- ❖ **Ciclo de vida iterativo e incremental:** una iteración involucra actividades de todos los flujos de trabajo, aunque desarrolla fundamentalmente algunos más que otros.

Además, RUP es la metodología empleada en el proyecto PRIMICIA, por lo que resulta natural ajustarse a ella para obtener una documentación uniforme y consistente con la generada en el proyecto.

## 1.6 Herramientas y tecnologías

Las herramientas son programas, aplicaciones o simplemente instrucciones que ofrecen la posibilidad de realizar varias funcionalidades con diferentes propósitos. Por otra parte, las tecnologías son el conjunto de conocimientos técnicos, ordenados científicamente, que permiten diseñar y crear bienes y servicios. La correcta selección de las herramientas y tecnologías informáticas a través de un estudio profundo de las características que las identifican, permiten agilizar y facilitar el desarrollo de los sistemas informáticos.

### 1.6.1 Framework

En el proceso de desarrollo de software un marco de trabajo, también definido como infraestructura digital o framework, es una estructura conceptual y tecnológica de software compuesta de componentes personalizables e intercambiables para el desarrollo de una aplicación. Puede ser considerado como una aplicación genérica incompleta y configurable a la que se le puede añadir las últimas piezas para construir una aplicación concreta. Puede incluir lenguaje interpretado, bibliotecas, soporte a programas y otras herramientas para ayudar a desarrollar y unir los diferentes componentes de un proyecto (Gutiérrez, 2012).

#### SDK de Android

SDK responde a las siglas *Software Development Kit*, lo que viene a ser un kit de desarrollo de software. Con él se puede desarrollar aplicaciones y ejecutar un emulador del sistema Android de la versión que sea (Valencia, 2012).

El *kit* de desarrollo de software de Android (Android SDK) contiene las herramientas necesarias para crear, compilar y empaquetar aplicaciones de Android. La mayoría de estas herramientas son de línea de comandos basada. La principal forma de desarrollar aplicaciones para Android se basa en el lenguaje de programación Java (Vogel, 2015).

#### Symfony2

*Symfony2* es un *framework* de PHP que permite centrarse en la lógica de tu negocio y te brinda numerosas herramientas para poner solución a los temas frecuentes de forma simple, ágil y robusta. Este *framework* de desarrollo web permite trabajar en diferentes formatos, usar componentes independientes, además de crear funcionalidades propias a través de los llamados *bundles* que son fácilmente reutilizables (Otero, 2014).

*Symfony2* es un *framework* de flexible acoplamiento que contiene componentes útiles estructurados en *bundles* (o paquetes); el *FOSRestBundle* proporciona una base sólida para empezar a construir los servicios REST en *Symfony2*. Puede proporcionar con enrutamiento automático para los recursos (probablemente entidad doctrine si utiliza *Doctrine*). Proporciona una buena base para un formato (JSON/HTML/XML) independiente de la capa de vista. Para el uso del *FOSRestBundle* se recomienda usar el *JMSSerializerBundle* que le permite serializar sus datos en un formato de salida solicitado como JSON, XML o YAML, y viceversa (Manandhar, 2014).

Resulta importante resaltar que el *framework* *Symfony2* es el utilizado por el proyecto PRIMICIA de ahí que haya sido seleccionado con el *framework* para el desarrollo de los servicios web.

### 1.6.2 Lenguajes de programación

Los lenguajes de programación son un conjunto de símbolos y reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos y expresiones, pueden usarse para crear programas que controlen el comportamiento físico y lógico de una máquina, permitiéndole a los programadores definir de manera precisa sobre qué datos debe operar una computadora, cómo estos deben ser almacenados o transmitidos. Este lenguaje intenta estar relativamente próximo al lenguaje humano o natural, representando en forma simbólica y en manera de un texto los códigos que podrán ser leídos y escritos por una persona que no necesariamente posea un conocimiento avanzado de la programación (Guevara, 2014).

#### Java

Java es un lenguaje de programación desarrollado por *Sun Microsystems*, fue presentado en la segunda mitad del año 1995 y desde entonces se ha convertido en un lenguaje de programación muy popular. Java es muy valorado porque los programas Javas se pueden ejecutar en diversas plataformas con sistemas operativos como *Windows*, *Mac OS*, *Linux* o *Solaris* (Guevara, 2014).

Las características principales que ofrece Java son:

**Lenguaje Simple:** se lo conoce como lenguaje simple porque viene de la misma estructura de C y C++; ya que C++ fue un referente para la creación de java por eso utiliza determinadas características de C++ y se han eliminado otras.

**Orientado a Objeto:** toda la programación en java en su mayoría está orientada a objeto, ya que al estar agrupados en estructuras encapsuladas es más fácil su manipulación.

**Distribuido:** permite abrir sockets, establecer y aceptar conexiones con los servidores o clientes remotos; facilita la creación de aplicaciones distribuidas ya que proporciona una colección de clases para aplicaciones en red.

**Robusto:** es altamente fiable en comparación con c, se han eliminado muchas características con la aritmética de punteros, proporciona numerosas comprobaciones en compilación y en tiempo de ejecución.

**Portable:** debido a que el código que se genera es un código independiente que es el que permite poder usarlo en las distintas plataformas, sin que cambie la esencia del software diseñado, es decir, que será el mismo programa en cualquier plataforma.

El principal motivo para la utilización de java como lenguaje de programación es que Android (descrito más adelante) tiene Java como lenguaje base, además de dar la posibilidad de crear o diseñar software y poder ser ejecutado en diferentes plataformas donde se requiere, sin la necesidad de estar modificándolo. Es una herramienta diseñada con muchos elementos parecidos a otros lenguajes pero en una versión mejorada, más sencilla de manejar, con menos complejidad y con barreras de seguridad en su lenguaje y ejecución.

## PHP

PHP (*PHP Hypertext Preprocessor*) es un lenguaje de programación interpretado de alto nivel y ejecutado en el lado del servidor. El propósito de este lenguaje es crear aplicaciones web con contenido dinámico, aplicaciones para líneas de comandos y aplicaciones web con interfaz gráfica. Una de las características de este lenguaje es que permite incrustar código HTML para realizar determinadas acciones y acceder a bases de datos para extraer información de él y mostrarla en la página (García, 2014).

### 1.6.3 Entorno de desarrollo

Un entorno de desarrollo informático IDE (por sus siglas en inglés *Integrated Development Environment*) es un programa informático compuesto por un conjunto de herramientas de programación. Consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica (GUI). Los IDE proveen un marco de trabajo amigable para la mayoría de los lenguajes de programación tales como C++, *Python*, Java, C#, *Visual Basic*, etc. (Maldonado, 2012).

## Eclipse

Eclipse es una plataforma de desarrollo, diseñada para ser extendida de forma indefinida a través de *plugins*. Fue concebida desde sus orígenes para convertirse en una plataforma de integración de herramientas de desarrollo. No tiene en mente un lenguaje específico, sino que es un IDE genérico, aunque goza de mucha popularidad entre la comunidad de desarrolladores del lenguaje

**Java** usando el *plugin* **JDT** que viene incluido en la distribución estándar del IDE. Eclipse es una plataforma ideal para el desarrollo de aplicaciones para dispositivos móviles (Eclipse, 2014).

Proporciona herramientas para la gestión de espacios de trabajo, escribir, desplegar, ejecutar y depurar aplicaciones.

**Principales características** (Eclipse, 2014):

- ❖ **Perspectivas, editores y vistas:** en Eclipse el concepto de trabajo está basado en las perspectivas, que no es otra cosa que una preconfiguración de ventanas y editores, relacionadas entre sí, y que permiten trabajar en un determinado entorno de trabajo de forma óptima.
- ❖ **Gestión de proyectos:** el desarrollo sobre Eclipse se basa en los proyectos, que son el conjunto de recursos relacionados entre sí, como puede ser el código fuente, documentación, ficheros configuración, árbol de directorios. El IDE proporciona asistencia y ayudas para la creación de proyectos. Por ejemplo, cuando se crea uno, se abre la perspectiva adecuada al tipo de proyecto que se está creando, con la colección de vistas, editores y ventanas pre-configurada por defecto.
- ❖ **Depurador de código:** se incluye un potente depurador, de uso fácil e intuitivo, y que visualmente ayuda a mejorar nuestro código. Para ello sólo se debe ejecutar el programa en modo depuración (con un simple botón). De nuevo, se tiene una perspectiva específica para la depuración de código, la perspectiva depuración, donde se muestra de forma ordenada toda la información necesaria para realizar dicha tarea.
- ❖ **Extensa colección de *plugins*:** están disponibles en una gran cantidad, unos publicados por Eclipse, otros por terceros. Al haber sido un estándar de facto durante tanto tiempo (no el único estándar, pero sí uno de ellos), la colección disponible es muy grande. Los hay gratuitos, de pago, bajo distintas licencias, pero casi para cualquier cosa que se imagine existe el *plugin* adecuado.

### **IDE Netbeans 8.0**

*Netbeans* es un entorno de desarrollo gratuito y de código abierto, permite el uso de un amplio rango de tecnologías de desarrollo tanto para escritorio, como aplicaciones Web, o para

dispositivos móviles. Su versión 8.0 proporciona nuevas mejoras como el soporte ampliado a *Maven* y Java EE con *PrimeFaces*; nuevas herramientas para HTML5, *AngularJS*, y mejoras en PHP y C/C++ (Gonzalez, 2014). Además viene con soporte para *Symfony* un gran framework MVC escrito en PHP.

#### 1.6.4 Gestor de bases de datos

En la actualidad los Sistemas de Gestión de Base de Datos (SGBD) sirven de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan, teniendo como propósito manejar de manera clara, sencilla y ordenada un conjunto de datos que posteriormente se convertirán en información relevante para una organización. A continuación se exponen las características del SGBD seleccionado así como los argumentos de su selección.

*PostgreSQL* es un servidor de base de datos, liberado bajo la licencia BSD<sup>1</sup>. Como muchos otros proyectos open source<sup>2</sup>, el desarrollo de *PostgreSQL* no es manejado por una sola compañía sino que es dirigido por una comunidad de desarrolladores y organizaciones comerciales las cuales trabajan en su desarrollo, dicha comunidad es denominada el PGDG (*PostgreSQL Global Development Group*). Fue el pionero en muchos de los conceptos existentes en el sistema objeto-relacional actual, incluido más tarde en otros sistemas de gestión comerciales. *PostgreSQL* es un sistema objeto-relacional, ya que incluye características de la orientación a objetos (*PostgreSQL*, 2012).

#### Características de PostgreSQL:

- ❖ Implementación del estándar SQL92/SQL99.
- ❖ Soporta distintos tipos de datos, además del soporte para los tipos base, también soporta datos de tipo fecha, monetarios, elementos gráficos, datos sobre redes y cadenas de *bits*. Por otra parte permite la creación de tipos propios.
- ❖ Incorpora una estructura de datos *array*<sup>3</sup>.

<sup>1</sup> Licencia de Distribución de Software *Berkeley*.

<sup>2</sup> Código abierto.

<sup>3</sup> Tipo de dato el cual almacena una secuencia de objetos definida por el desarrollador.

- ❖ Incorpora funciones de diversa índole: manejo de fechas, geométricas, orientadas a operaciones con redes, entre otras.
- ❖ Permite la declaración de funciones propias, así como la definición de disparadores.
- ❖ Soporta el uso de índices, reglas y vistas.
- ❖ Incluye herencia entre tablas, por lo que a este gestor de bases de datos se le incluye entre los gestores objeto-relacionales.
- ❖ Permite la gestión de diferentes usuarios, como también los permisos asignados a cada uno de ellos.

*PostgreSQL* es un gestor de bases de datos disponible en casi cualquier Unix (34 plataformas en la última versión estable), y una versión nativa de *Windows*, usando una estrategia de almacenamiento de filas llamada MVCC4 para conseguir una mejor respuesta en ambientes de grandes volúmenes, teniendo un mantenimiento y ajuste mucho menor que los productos de los proveedores comerciales. Conserva todas las características, estabilidad y rendimiento dando al traste con las necesidades que plantea el proyecto productivo, dado por las características potenciales del gestor y su gratitud.

### 1.6.5 Entorno de desarrollo web

El entorno web hace referencia a un ambiente de desarrollo y/o ejecución programas o servicios en el marco de la web en general. El entorno web es una forma de interfaz de usuario gráfico.

#### WampServer

Es un entorno de desarrollo web para *Windows* con el que podrás crear aplicaciones web con *Apache*, PHP y bases de datos *MySQL*. También incluye *PHPMysqlAdmin* y *SQLiteManager* para manejar tus bases de datos. Provee a los desarrolladores con los cuatro elementos necesarios para un servidor web: un Sistema Operativo (*Windows*), un manejador de base de datos (*MySQL*), un software para servidor web (*Apache*) y un software de programación script Web (PHP (generalmente), *Python* o PERL), debiendo su nombre a dichas herramientas. Lo mejor de todo es que WAMP5 es completamente gratuito. WAMP incluye, además de las últimas versiones de

---

<sup>4</sup>Multi-Versión para el control de concurrencia.

*Apache*, PHP y *MySQL*, versiones anteriores de las mismas, para el caso de que se quiera testear en un entorno de desarrollo particular.

### Herramientas para el diseño y desarrollo de páginas web

Del lado del servidor:

- ❖ *Windows*, como sistema operativo;
- ❖ *Apache*, como servidor web;
- ❖ *MySQL*, como gestor de bases de datos;
- ❖ PHP (generalmente), Perl, o *Python*, como lenguajes de programación.

Del lado del Cliente:

- ❖ Navegador de Internet

El uso de WAMP permite servir páginas HTML a Internet, además de poder gestionar datos en ellas, al mismo tiempo WAMP, proporciona lenguajes de programación para desarrollar aplicaciones Web (Ledesma, 2013).

#### 1.6.6 Lenguaje de Modelado

El Lenguaje Unificado de Modelado o UML (del inglés *Unified Modeling Language*) es un lenguaje gráfico para visualizar, especificar, construir y documentar los artefactos de un sistema con gran cantidad de software. UML 2.0 proporciona una forma estándar de escribir los planos de un sistema, cubriendo tanto las cosas conceptuales, tales como procesos del negocio y funciones del sistema, como las cosas concretas, tales como las clases escritas en un lenguaje de programación específico, esquemas de bases de datos y componentes software reutilizables (Orallo, 2011).

Las funciones de UML 2.0 son las siguientes:

- ❖ **Visualizar:** permite expresar de una forma gráfica un sistema de forma que otro lo puede entender.
- ❖ **Especificar:** permite especificar cuáles son las características de un sistema antes de su construcción.

- ❖ **Construir:** a partir de los modelos especificados se pueden construir los sistemas diseñados.
- ❖ **Documentar:** los propios elementos gráficos sirven como documentación del sistema desarrollado que pueden servir para su futura revisión.

### **1.6.7 Herramienta CASE**

Se puede definir a las herramientas CASE como un conjunto de programas y ayudas que dan asistencia a los analistas, ingenieros de software y desarrolladores, durante todos los pasos del ciclo de vida de desarrollo de un software. CASE es también definido como el conjunto de métodos, utilidades y técnicas que facilitan el mejoramiento del ciclo de vida del desarrollo de sistemas de información, completamente o en alguna de sus fases (CASE, 2010).

#### **Visual Paradigm 8.0**

Es una herramienta CASE: Ingeniería de Software Asistida por Computación. La misma propicia un conjunto de ayudas para el desarrollo de programas informáticos, desde la planificación, pasando por el análisis y el diseño, hasta la generación del código fuente de los programas y la documentación.

#### **Características:**

- ❖ Disponibilidad en múltiples plataformas (Windows, Linux).
- ❖ Diseño centrado en casos de uso y enfocado al negocio que generan un software de mayor calidad.
- ❖ Uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación.
- ❖ Capacidades de ingeniería directa e inversa.
- ❖ Modelo y código que permanece sincronizado en todo el ciclo de desarrollo.
- ❖ Disponibilidad de múltiples versiones, para cada necesidad.
- ❖ Licencia: gratuita y comercial.
- ❖ Fácil de instalar y actualizar.

- ❖ Diagramas de Procesos de Negocio - Proceso, Decisión, Actor de negocio, Documento.
- ❖ Diagramas de flujo de datos.
- ❖ Integración con Visio - Dibujo de diagramas UML con plantillas de *Microsoft Visio*.
- ❖ Editor de figuras.

## **1.7 Conclusiones del capítulo**

En el capítulo se analizaron los conceptos vinculados con el tema de la investigación como bases para el entendimiento del mismo, lo cual arrojó que el uso de los medios de visualización de noticias a través de las tecnologías y dispositivos móviles permite mantener informado al hombre sin importar el lugar donde se encuentre. La realización del estudio de los diferentes visores de noticias para dispositivos móviles permitió concluir que dada la insuficiente documentación de los visores existente se hace necesario el desarrollo de una aplicación para dispositivos móviles que permita la visualización de noticias para la plataforma informativa PRIMICIA. Luego de realizar un estudio de las metodologías, tecnologías y herramientas se definió como metodología de desarrollo RUP por ser la que se utiliza en el proyecto PRIMICIA. Se seleccionó el servicio web REST por ser el que más se ajusta a las características del sistema a desarrollar para el intercambio de datos entre aplicaciones diferentes, como lenguaje de programación se definió el uso de Java ya que el lenguaje base que utiliza Android que es sistema operativo que se va a utilizar en la solución.

## **Capítulo II: Presentación de la propuesta de solución**

En el presente se abordarán argumentos importantes para la elaboración del Visor de noticias para dispositivos móviles de PRIMICIA; además de efectuar la representación de los artefactos logrados a través del modelo de dominio y la especificación de los requisitos funcionales y no funcionales que debe presentar la aplicación una vez terminado. Se determinan los casos de uso del sistema y los actores que interactuarán con este así como también el modelo de diseño, los patrones utilizados en la solución y el diagrama de secuencia del sistema a desarrollar.

### **2.1 Propuesta de solución**

Se propone la elaboración de un Visor de Noticias que permita visualizar la información que puede ser de distintos formatos (imagen, texto, audio o video) de la plataforma informativa PRIMICIA a través de dispositivos móviles que utilizan Android como sistema operativo a partir de su versión 3.0. La aplicación será capaz de mostrar una lista con las noticias activas, mostrar el contenido de la noticia seleccionada en sus pantallas correspondientes, podrá también realizar una búsqueda de las noticias activas y archivadas con anterioridad según el criterio de búsqueda (fecha, temática o título) así como visualizar el flujo de trasmisión del canal televisivo de PRIMICIA. La arquitectura de la solución de basa en el patrón Modelo Vista Controlador, el intercambio de datos entre la base de datos y el Visor de Noticias se hará a través del servicio web REST. La aplicación funcionará como se muestra en la figura 1:

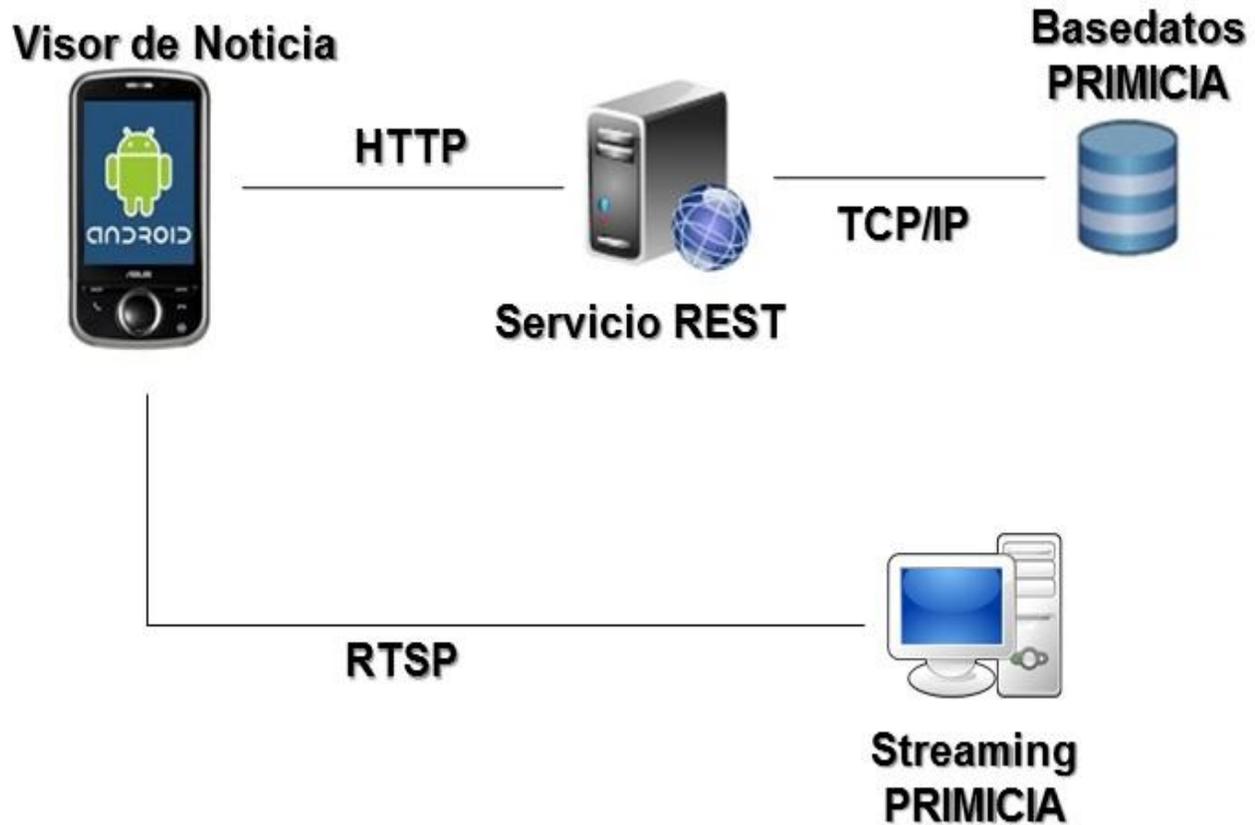


Figura 1 Propuesta de Solución

## 2.2 Modelo de Dominio

Un modelo del dominio es una representación visual de las clases conceptuales u objetos del mundo real en un dominio de interés. También se les denomina modelos conceptuales, ya que ayuda a comprender los conceptos clave de un negocio o un dominio de problema. A continuación se presenta el Modelo de Dominio del Visor de noticias para dispositivos móviles (Larman, 2013).

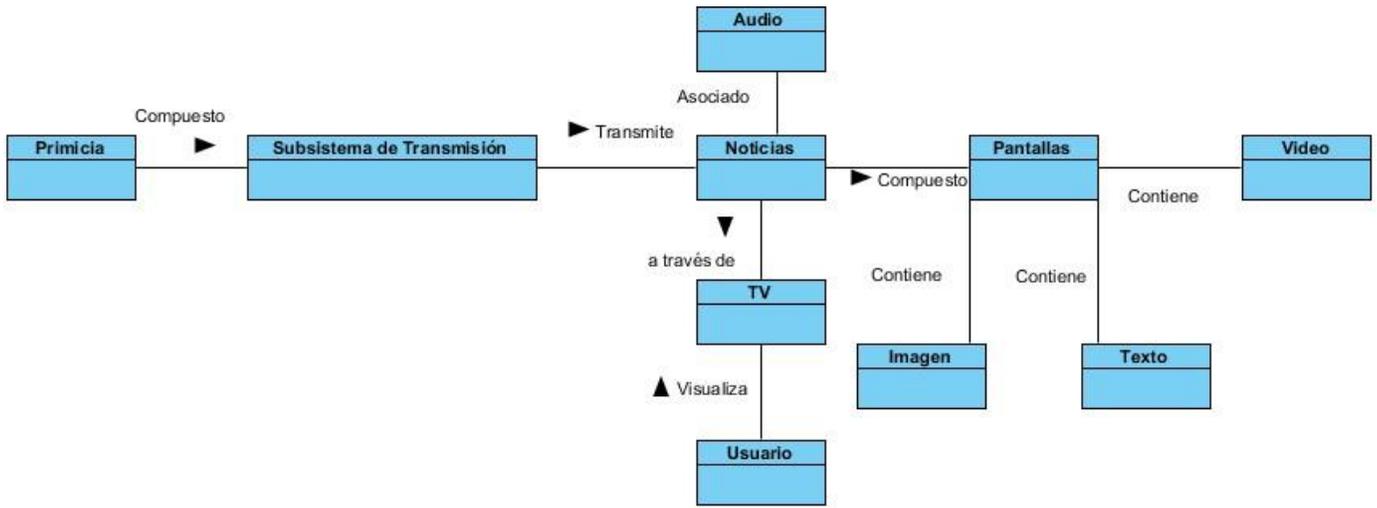


Figura 2. Modelo de Domino de Visor de Noticias.

Para brindar una mejor comprensión del diagrama del modelo de dominio a continuación se realiza una breve descripción de las clases del dominio.

**Usuario:** entidad encargada de interactuar con la aplicación.

**VisorNoticias:** entidad con la que el usuario interactúa.

**ListarNoticias:** entidad que va a permitir al usuario visualizar una lista de noticias.

**VisualizarTransmisión:** entidad encargada de mostrar el flujo de transmisión.

**Noticia:** entidad que se muestra, la cual puede ser del tipo texto, audio, imagen, texto-imagen o video.

## 2.3 Requisitos del sistema

Los requisitos se utilizan como datos de entrada en la etapa de diseño del producto y establecen qué debe hacer el sistema. Son una condición o capacidad que un usuario necesita para poder resolver un problema o lograr un objetivo (IAN Sommerville, 2012).

### 2.3.1 Requisitos funcionales

Los requisitos funcionales definen el comportamiento interno de un software, son condiciones que el sistema ha de cumplir. Estos muestran las funcionalidades que deben satisfacerse para cumplir con las especificaciones de software (IAN Sommerville, 2012).

**RF1: Listar Noticias activas.**

Descripción: permitirá al usuario ver una lista con las noticias activas en ese momento.

Salida: una lista con todas las noticias activas en ese momento.

**RF2: Mostrar pantallas con el contenido de la noticia seleccionada.**

Descripción: mostrará el contenido de una noticia que sea seleccionada por el usuario en sus pantallas correspondientes.

Salida: el contenido de una noticia.

**RF3: Buscar noticia por título.**

Descripción: permitirá al usuario realizar la búsqueda de una noticia de acuerdo a su título.

Entrada: el título de la noticia a buscar.

Salida: una lista con el resultado de la búsqueda.

**RF4: Buscar noticia por fecha.**

Descripción: permitirá al usuario realizar la búsqueda de una noticia de acuerdo a su fecha de publicación.

Entrada: la fecha de la noticia a buscar.

Salida: una lista con el resultado de la búsqueda.

**RF5: Buscar noticia por temática.**

Descripción: permitirá al usuario realizar la búsqueda de una noticia de acuerdo a la temática.

Entrada: la temática de la noticia a buscar.

Salida: una lista con el resultado de la búsqueda.

**RF6: Cargar el flujo de transmisión.**

Descripción: permitirá a la aplicación cargar el flujo de transmisión que será posteriormente mostrado.

**RF7: Mostrar el flujo de transmisión.**

Descripción: permitirá la visualización del flujo de transmisión.

### **2.3.2 Requisitos no funcionales**

Los requisitos no funcionales son propiedades emergentes de un sistema como la fiabilidad, el tiempo de respuesta y la capacidad de almacenamiento, pueden especificar el rendimiento del sistema, la protección, la disponibilidad, y otras propiedades emergentes. Además se conocen como un conjunto de características de calidad, que es necesario tener en cuenta al diseñar e implementar el software (IAN Sommerville, 2012).

#### **Usabilidad**

La herramienta utilizará el castellano como lenguaje base y debe ajustarse al entorno donde será aplicada. Debe ser capaz de aprovechar la infraestructura de red inalámbrica brindada por la institución. Debe permitir su utilización por cualquier usuario con conocimientos básicos en la tecnología móvil, brindándole la capacidad de trabajar de manera eficaz con la herramienta.

#### **Confiabilidad**

La veracidad de la información adquirida deberá corresponder sin ninguna alteración con los elementos obtenidos de la base de datos de PRIMICIA.

#### **Restricciones de Diseño e Implementación**

- ❖ El sistema deberá ser implementado en el lenguaje de programación Java.
- ❖ Se utilizará el marco de trabajo de desarrollo Android en su versión 23.0.2.
- ❖ Se empleará la herramienta de desarrollo *Eclipse* 4.2.1 y el sistema gestor de base de datos *PostgreSQL* 9.1.

#### **Software**

##### **Requisitos mínimos para el dispositivo móvil**

- ❖ Soporte para conexiones WIFI.
- ❖ Sistema operativo Android 3.0

#### **Hardware**

##### **Requisitos mínimos para el dispositivo móvil**

- ❖ Capacidad mínima de 10Mb en la memoria del dispositivo para la instalación y ejecución de la herramienta.

## 2.4 Casos de usos del sistema

Un caso de uso expresa todas las formas de usar un sistema para alcanzar una meta particular para un usuario. En conjunto, los casos de uso le proporcionan todos los caminos útiles de usar el sistema e ilustran el valor que este provee (Jacobson, 2013).

### 2.4.1 Definición de los actores del sistema

Según Roger Pressman “los casos de uso se definen desde el punto de vista de un actor. Un actor es un papel que desempeñan las personas (usuarios) o los dispositivos cuando interactúan con el software” (Pressman, 2013).

Como parte del sistema se identificó un actor para interactuar con el visor móvil de noticias:

Tabla 2. Actores del sistema.

Actor	Descripción
Usuario	Persona con acceso al sistema encargada de interactuar con las noticias que se muestran en el visor móvil de noticias.

### 2.4.2 Diagrama de casos de uso

Los diagramas de casos de uso representan o documentan gráficamente la interacción de los actores con las funcionalidades de un sistema.

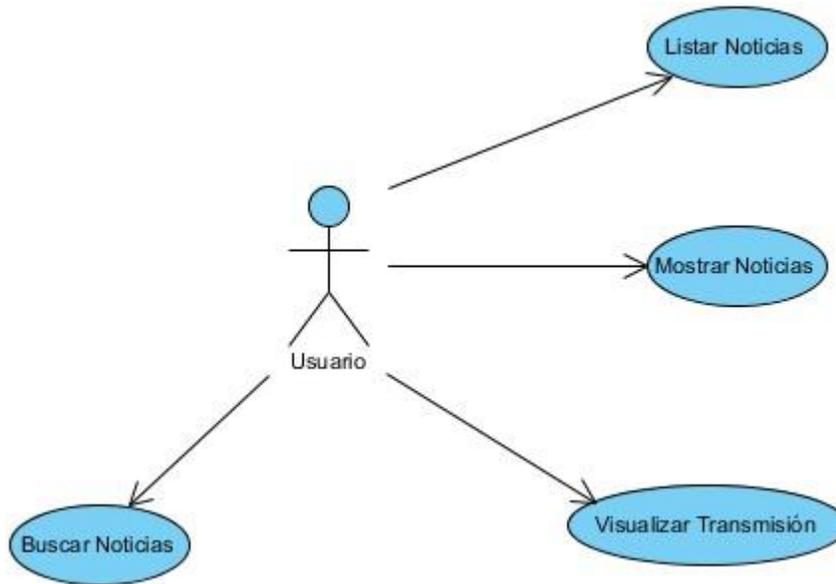


Figura 3. Diagrama de Casos de uso

### 2.4.3 Descripción de casos de uso

**Listar Noticias:** permite al usuario visualizar una lista de las noticias activas en el servidor de PRIMICIA.

**Mostrar Noticias:** permite al usuario visualizar las diferentes pantallas con el contenido de una noticia que haya sido previamente seleccionada.

**Buscar Noticia:** permite al usuario realizar una búsqueda de la noticia deseada siguiendo un criterio (título, fecha o temática).

**Visualizar Transmisión:** permite al usuario visualizar el flujo de transmisión emitido por PRIMICIA.

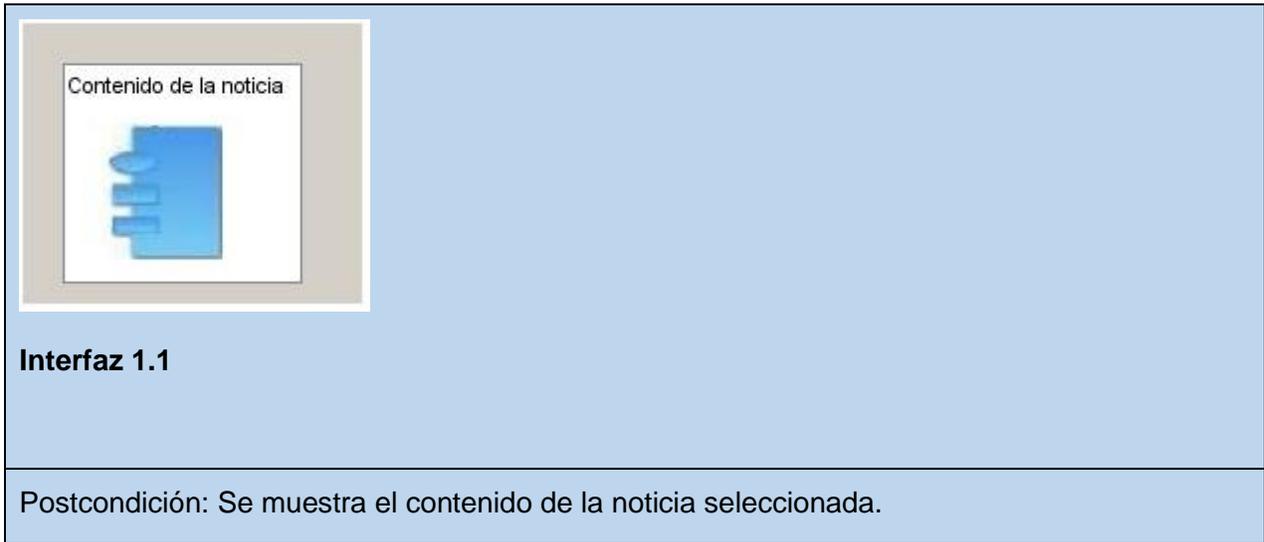
A continuación se muestra la descripción del caso de uso arquitectónicamente significativo del sistema.

#### Descripción del CU: Mostrar Noticias

Tabla 3. Caso de uso: Mostrar noticias

<b>Caso de Uso</b>	Mostrar Noticias
<b>Actores</b>	Usuario

<b>Propósito</b>	Este caso de uso se lleva a cabo con el objetivo de mostrarle al usuario una noticia que haya sido seleccionada.	
<b>Resumen</b>	El Caso de uso inicia cuando el actor selecciona una noticia de la lista. La aplicación muestra el contenido de la noticia seleccionada. El caso de uso termina cuando el usuario regresa a la lista de noticias.	
<b>Complejidad</b>	Media	
<b>Prioridad</b>	Crítico	
<b>Precondiciones</b>	Las noticias deben estar previamente listadas.	
<b>Referencias</b>	RF1, RF2.	
<b>Flujo Normal de Eventos</b>		
	<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>
1	Selecciona una noticia de la lista mostrada por el Visor de Noticias	
2		Muestra una pantalla con el contenido de la noticia seleccionada.
3	Pasa a la siguiente pantalla.	
4		Muestra una nueva pantalla con la otra parte del contenido de la noticia terminando así el caso de uso (Ver Interfaz 1.1).
<b>Prototipo de Interfaz</b>		



## 2.5 Modelo de Diseño

El modelo de diseño es un modelo de objetos que describe la relación física de los casos de uso centrándose en como los requisitos funcionales y no funcionales, junto con otras restricciones relacionadas con el entorno de implementación, tienen impacto en el sistema a considerar. Además, el modelo de diseño sirve de abstracción de la implementación del sistema y es, de ese modo, utilizada como una entrada fundamental de las actividades de implementación (Pressman, 2013).

### 2.5.1 Diagrama de clases del diseño

Los diagramas de clases del diseño son utilizados durante el proceso de análisis y diseño de los sistemas, donde se crea el diseño conceptual de la información que se manejará en el sistema, y los componentes que se encargarán del funcionamiento y la relación entre uno y otro. Describen gráficamente las especificaciones de las clases del software y de las interfaces en una aplicación. En su diseño contiene como información (clases, métodos, información sobre los tipos de los atributos, dependencias) (Larman, 2013).

En la figura 8 se muestra la representación del diagrama de clases del diseño Mostrar Noticias

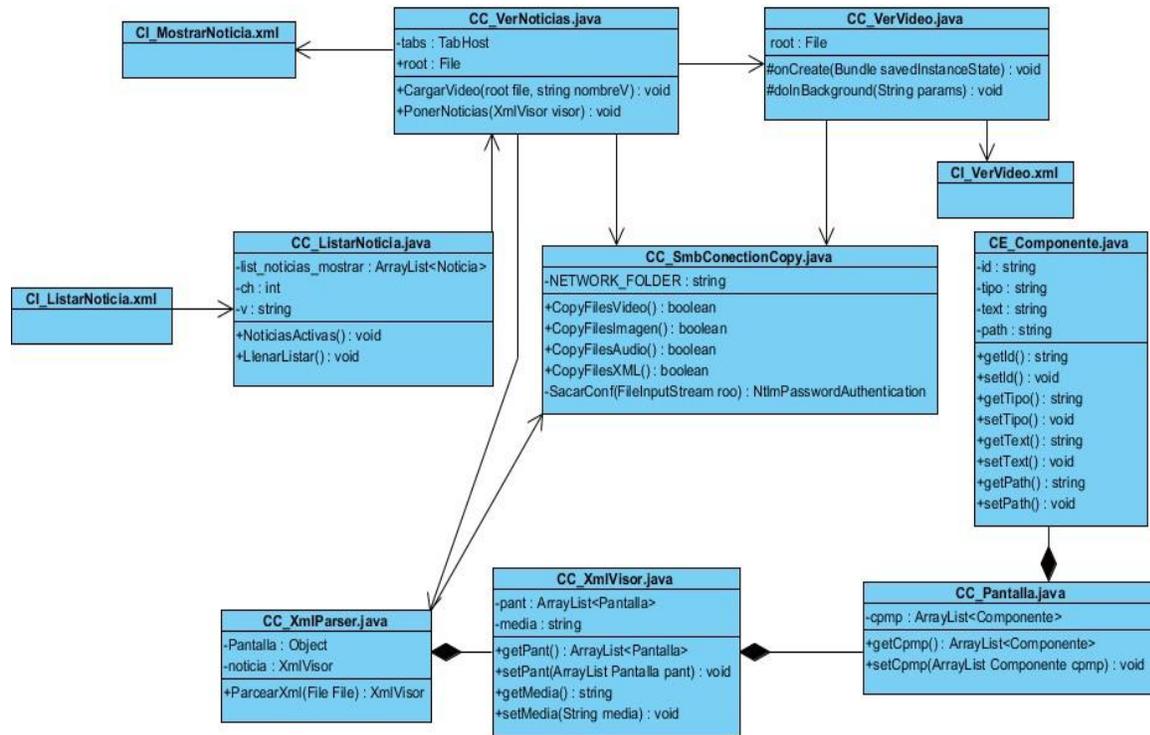


Figura 4. Diagrama de clases del diseño.

A continuación se realiza una breve descripción de las clases del diagrama de clases del diseño representado en al figura anterior.

Tabla 4. Descripción de las clases.

<b>Nombre: CI_MostrarNoticia</b>
<b>Descripción:</b> permite visualizar el contenido de una noticia que haya sido seleccionada.
<b>Nombre: CI_VerV</b>
<b>Descripción:</b> permite visualizar el video perteneciente al contenido de una noticia.
<b>Nombre: CC_ListarNoticia</b>
<b>Descripción:</b> permite la creación de la lista de noticias que se van a mostrar a través del método NoticiasActivas (). Además se encarga de enviar la información de que una noticia ha sido seleccionada a través del método StarActivity(). Permite también la creación de las listas de noticas que son creadas después de seleccionado un criterio de búsqueda a través de los métodos

FiltrarCateg(), BuscarXFecha() y BuscarNoticia().
<b>Nombre: CC_VerNoticia</b>
<b>Descripción:</b> es la clase controladora que posee los métodos necesarios para la creación de la vista CI_MostrarNoticia. Para ello se apoya en métodos como onCreate(Bundle savedInstanceState) el cual permite la conexión con la clase SmbConection para obtener los datos necesarios para la conformación de la noticia, además de enviar a parsear el archivo .xml correspondiente a la noticia.
<b>Nombre: CC_VerVideo</b>
<b>Descripción:</b> es la clase que posee los métodos necesarios para la creación de la vista CI_VerV.. Para ello se apoya en métodos como onCreate(Bundle savedInstanceState) el cual permite la conexión con la clase SmbConection, además de contar con el método PonerNoticia() que es el encargado de poner en las clases vistas los elementos que componen una noticia.
<b>Nombre: CC_SmbConectionCopy</b>
<b>Descripción:</b> es la clase que posee los métodos necesarios para la obtención de los elementos que conforman una noticia, como son imagen, audio y video, así como su .xml correspondiente mediante los métodos CopyFilesVideo(String nombre,File root), CopyFilesImagen(String nombre,File root), CopyFilesAudio(String nombre,File root)y CopyFilesXML(String nombre,File root).
<b>Nombre: CC_XmlParser</b>
<b>Descripción:</b> es la clase que se encarga de parsear el archivo .xml perteneciente a una noticia con el objetivo de encontrar todos los elementos por los que está compuesta la misma.
<b>Nombre: CC_XmlVisor</b>
<b>Descripción:</b> Es la entidad que se encarga de conformar el cuerpo de la noticia agrupando las pantallas correspondientes a la misma.
<b>Nombre: CC_Pantalla</b>
<b>Descripción:</b> es la clase que se encarga de conformar una pantalla con sus componentes correspondientes.
<b>Nombre: CE_Componente</b>

**Descripción:** es la clase que se encarga de conformar un componente.

## 2.6 Patrones utilizados

Los patrones son formas de describir las mejores prácticas, buenos diseños, y encapsulan la experiencia de forma tal que es posible para otros reutilizar dicha experiencia. Constituyen mecanismos cuyo objetivo es la solución de problemas que ocurren repetidamente dentro de un contexto muy bien definido. Las soluciones que son propuestas a través de patrones involucran algunas clases de estructuras que permiten contemplar los requisitos no funcionales

### 2.6.1 Patrones Arquitectónicos

La selección del patrón arquitectónico Modelo-Vista-Controlador (MVC) está basada en la existencia de clases modelos que son las encargadas del trabajo con los datos que van a ser mostrados en las vistas de la aplicación, teniendo las clases controladoras que son las encargadas de la interacción entre las Vista y el Modelo.

#### MVC

El patrón Modelo-Vista-Controlador (MVC) es un paradigma que divide las partes que conforman una aplicación en el Modelo, las Vistas y los Controladores, permitiendo la implementación por separado de cada elemento.

**Modelo:** es el objeto que representa los datos del programa. Maneja los datos y controla todas sus transformaciones. El Modelo no tiene conocimiento específico de los Controladores o de las Vistas, ni siquiera contiene referencias a ellos. Es el propio sistema el que tiene encomendada la responsabilidad de mantener enlaces entre el Modelo y sus Vistas, y notificar a las Vistas cuando cambia el Modelos.

**Vista:** es el objeto que maneja la representación visual de los datos representados por el Modelo. Genera una representación visual del Modelo y muestra los datos al usuario. Interactúa preferentemente con el Controlador, pero es posible que trate directamente con el Modelo a través de una referencia al propio Modelo.

**Controlador:** es el objeto que proporciona significado a las órdenes del usuario, actuando sobre los datos representados por el Modelo, centra toda la interacción entre la Vista y el Modelo.

Cuando se realiza algún cambio, entra en acción, bien sea por cambios en la información del Modelo o por alteraciones de la Vista. Interactúa con el Modelo a través de una referencia al propio Modelo (Patrón Modelo-Vista-Controlador, 2012).

### **2.6.2 Patrones de Diseño**

Los patrones de diseño de software son soluciones reutilizables de problemas recurrentes que aparecen durante el proceso de diseño de software orientado a objetos. Estos surgen por la necesidad de transmitir la experiencia a los demás desarrolladores.

Con el conocimiento de estos patrones, los programadores son capaces de identificar las situaciones en las que éstos tienen aplicación, y utilizarlos sin tener que detenerse para analizar el problema y vislumbrar diferentes estrategias de resolución. A continuación se realiza una descripción de algunos de los patrones de diseño utilizados en la solución.

#### **Patrones generales de software para asignar responsabilidades (GRASP)**

##### **Creador**

Los patrones de creación muestran la guía de cómo crear objetos cuando sus creaciones requieren tomar decisiones. Estas decisiones normalmente serán resueltas dinámicamente decidiendo que clases instanciar o sobre que objetos un objeto delegará responsabilidades. La valía de los patrones de creación dice como estructurar y encapsular estas decisiones (Larman, 2013).

Este patrón se evidencia en la clase `CC_Conexion.java` que es la encargada de crear instancias de la clase `CE_Noticia.java` para poder conformar las listas de noticias que se van a mostrar en las vistas correspondientes.

##### **Controlador**

El patrón controlador permite facilitar la centralización de actividades, delegar estas en otras clases con las que mantiene un modelo de alta cohesión (Larman, 2013).

Este patrón se pone de manifiesto en la clase controladora `CC_ListarNoticia.java` porque permiten facilitar la centralización de las actividades. Este patrón se utiliza para poder controlar el funcionamiento de la clase vista `CI_ListarNoticia.xml`.

##### **Experto**

Permite asignar una responsabilidad al experto en información: la clase que cuenta con la información necesaria para cumplir la responsabilidad (Larman, 2013).

Dicho patrón se evidencia en la clase CC\_VerNoticias.java en la cual recae la responsabilidad de la creación de la vista CI\_MostrarNoticia.xml ya que cuenta con toda la información necesaria para crearla. Dicha creación se realiza a través del método onCreate() de la clase CC\_VerNoticias.java y el contenido a mostrar en la vista CI\_MostrarNoticia.xml se le pasa a la misma a través del método PonerNoticias() que es en el que recae la responsabilidad del poner el contenido de la noticia en su vista correspondiente.

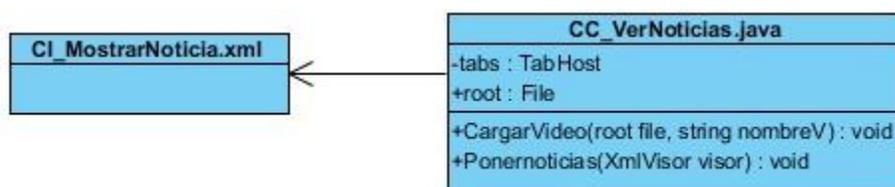


Figura 5. Representación del patrón Experto.

### Alta cohesión

Facilita la solución al problema ¿Cómo mantener manejable la complejidad? mediante la asignación de responsabilidades de manera que la información que almacena una clase sea coherente y esté relacionada con la clase (Larman, 2013).

El uso de dicho patrón queda evidenciado en la clase modelo CE\_Componente.java la cual posee la estructura necesaria para guardar la información de forma coherente y de acuerdo a su responsabilidad. Este patrón se utiliza en esta clase porque al guardar la información referente a un componente de forma coherente y de acuerdo a su responsabilidad se está facilitando la manejabilidad de la complejidad.

### Bajo acoplamiento

Tiene la idea de mantener las clases lo menos ligadas entre sí que se pueda. De tal forma que en caso de producirse una modificación en alguna de ellas, se tenga la mínima repercusión posible en el resto de clases, potenciando la reutilización y disminuyendo la dependencia entre las clases (Larman, 2013).

Se evidencia en la relación entre las clases CC\_ListarNoticias.java y la clase CC\_Conexion.java ya que estas clases pese a estar relacionadas se mantienen lo menos ligadas posible ya que al realizar cualquier cambio dentro de la clase CC\_Conexion.java los mismos tienen una mínima repercusión dentro de la clase CC\_ListarNoticias.java.

### Patrones Gang of Four (GOF)

Los patrones de diseño del grupo de GOF se clasifican en 3 grandes categorías basadas en su propósito: creacionales, estructurales y de comportamiento.

#### Interpreter

Patrón de comportamiento a nivel de clases, tiene como propósito dado un lenguaje definir la representación para su gramática junto con un intérprete que usa dicha representación para interpretar sentencias en ese lenguaje (Larman, 2013).

Este patrón se evidencia en la clase CC\_XmlParser.java que es la clase que se encarga de parsear el archivo .xml perteneciente a una noticia con el objetivo de encontrar todos los elementos por los que está compuesta la misma haciendo uso de una gramática definida, donde a través del intérprete se va a realizar una búsqueda de las palabras claves video, audio, texto e imagen dado el lenguaje descrito dentro del archivo .xml

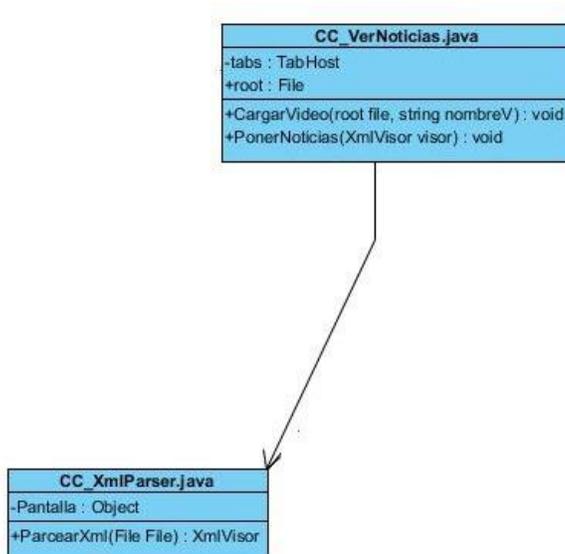


Figura 6. Representación del patrón *Interpreter*.

### ***Adapter***

Patrón de estructura tanto a nivel de clase como de objeto, tiene como propósito convertir la interfaz de una clase para lo que el cliente que la usa necesite, permitiendo así que trabajen juntas clases cuyas interfaces son incompatibles (Larman, 2013).

Este patrón se pone de manifiesto en la clase `NoticiaAdapter.java` la cual se encarga de cambiarle la estructura a las listas que se van a mostrar en la clase `Cl_ListarNoticia.xml`, este proceso se realiza a través del método `getView()` de la clase `NoticiaAdapter.java` que es el encargado de cambiarle el formato a la lista que se va a mostrar.

## **2.7 Diagrama de Secuencia**

Un diagrama de secuencia es la versión taquigráfica de un caso de uso, indica la forma en la que los eventos provocan transiciones de un objeto a otro; representa el comportamiento, describiendo la forma en la que las clases pasan de un estado a otro (Pressman, 2013).

En el diagrama de la figura 10 el actor usuario selecciona una noticia de la lista, luego la clase `CC_ListarNoticias` se encarga de mandar a ejecutar el método `StarActivity()` para enviar la información de que una noticia ha sido seleccionada hacia la clase `CC_VerNoticias`. Una vez que llega la información se ejecuta el método `onCreate()` el cual se encarga de realizar la petición del nombre y la dirección del archivo `.xml` correspondiente a la noticia seleccionada, la clase `SmbConection` mediante el método `CopyFilesXml` le devuelve a la clase `CC_VerNoticias` el nombre del archivo `.xml` y la dirección donde se encuentra. Después mediante la función `XmlParser.ParsearXml()` la clase `CC_VerNoticias` envía hacia la clase `CC_XmlParser` el archivo `.xml` a parsear y su dirección correspondiente. La clase `CC_XmlParser` se encarga de parsear el archivo `.xml` y haciendo uso del método `ParsearXml()` realiza las peticiones de los archivos de imagen y audio correspondientes a la clase `SmbConection()` la cual mediante los métodos `SmbConection.CopyFilesImagen()` y `SmbConection.CopyFilesAudio()` responde a la petición realizada.

Después de realizado este proceso la clase `CC_XmlParser` retorna un objeto noticia del tipo `XmlVisor` haciendo uso de la función `return noticia:XmlVisor`. Una vez obtenido dicho objeto la clase `CC_VerNoticias` a través del método `PonerNoticias()` envía a la clase `XmlVisor` los elementos a mostrar. La misma se encarga de conformar los elementos a mostrar, a través del método

getPant() se comunica con la clase CC\_Pantalla y obtiene la pantalla en la que se van a mostrar los elementos de la noticia y está a través del método getCmp() se comunica con la clase CE\_Componente y obtienen el componente correspondiente a la noticia. Luego de realizado este proceso la clase CC\_VerNoticias mediante la función tempAnadir.addview() va enviando los elementos a mostrar a la clase vista CI\_MostarNoticia. Si la pantalla a mostrar contiene un video el usuario selecciona ver video y la clase CC\_VerVideo ejecuta tempAnadir.addview() y añade el elemento video a la clase vista CI\_VerVideo.

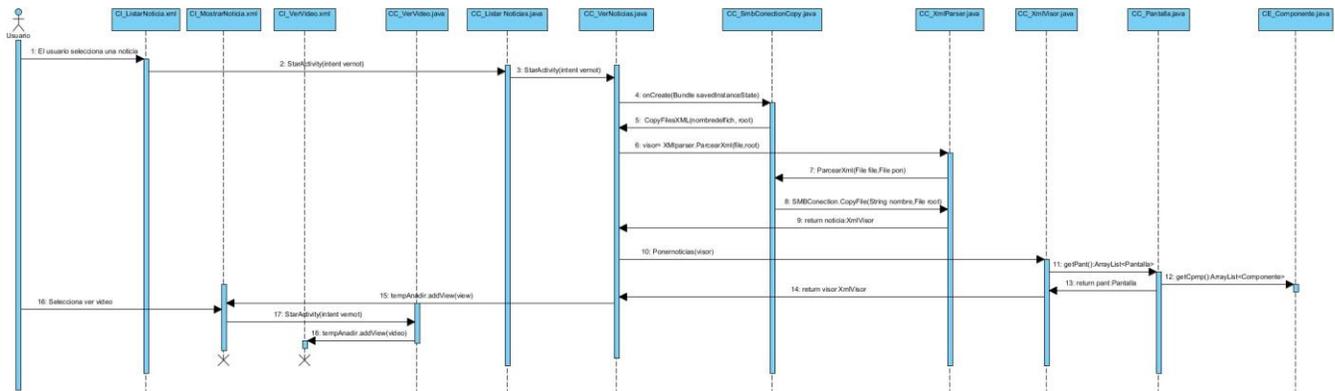


Figura 7. Diagrama de Secuencia Mostrar Noticias.

## Conclusiones del capítulo

Con la realización de este capítulo quedo claramente definida la propuesta de solución del Visor de Noticias, mediante el modelo de dominio se realizó una representación visual de las clases conceptuales para ayudar a comprender los conceptos claves del dominio del problema. Se identificaron las funcionalidades y los requisitos que el mismo debe cumplir. Se identificaron 4 casos de uso los cuales fueron detallados en la descripción de los casos de uso de sistema para lograr así un mayor entendimiento de los requisitos funcionales del sistema previamente establecidos. Además el uso de patrones en el modelo de diseño al permitió elaborar diagramas de clases de diseño fiables para la implementación de los diferentes casos de uso del visor web guiado por la arquitectura Modelo-Vista-Controlador. Se utilizaron los diagramas de secuencia para revisar los aspectos dinámicos del sistema.

## **Capítulo III: Implementación y prueba del visor de noticias**

En el presente capítulo se muestran los diagramas del modelo de implementación como resultado del diseño anteriormente elaborado. Además de la aplicación de pruebas para comprobar el correcto funcionamiento del Visor de Noticias.

### **3.1 Modelo de implementación**

El modelo de implementación representa la composición física de la implementación en términos de subsistemas de implementación, y elementos de implementación como son: directorios y archivos, incluyendo código fuente, datos y archivos ejecutables (Jacobson, y otros, 2000). Es de gran utilidad a la hora de implementar el sistema, pues facilita la organización del producto y lo hace más entendible a los desarrolladores.

#### **3.1.1 Diagrama de Componentes**

El diagrama de componentes describe la relación entre componentes de software, sus dependencias, comunicación y ubicación. Muestran las opciones de realización incluyendo código fuente, binario y ejecutable. Los componentes representan todos los tipos de elementos de software que entran en la fabricación de aplicaciones informáticas. Pueden ser simples archivos, paquetes y bibliotecas cargadas dinámicamente (Edward, 2012).

A continuación se describe el diagrama de componente del CU “Mostrar Noticia” el cual cuenta con 3 paquetes de implementación básicos. Estos paquetes son:

- ❖ El Paquete de Clases Vista, que agrupa los componentes que permiten la interacción directa con el usuario final del sistema, mostrando y recogiendo la información solicitada.
- ❖ El Paquete de Clases Controlador, que contendrá las clases que se encargaran de coordinar todos los demás elementos y transformar las peticiones del usuario en operaciones sobre el modelo y la vista
- ❖ El Paquete de Clases Modelo, que agrupa las clases que interactúan con la base de datos y representa la información con la que trabaja la aplicación y se encarga de acceder a los datos. (ver figura 12)

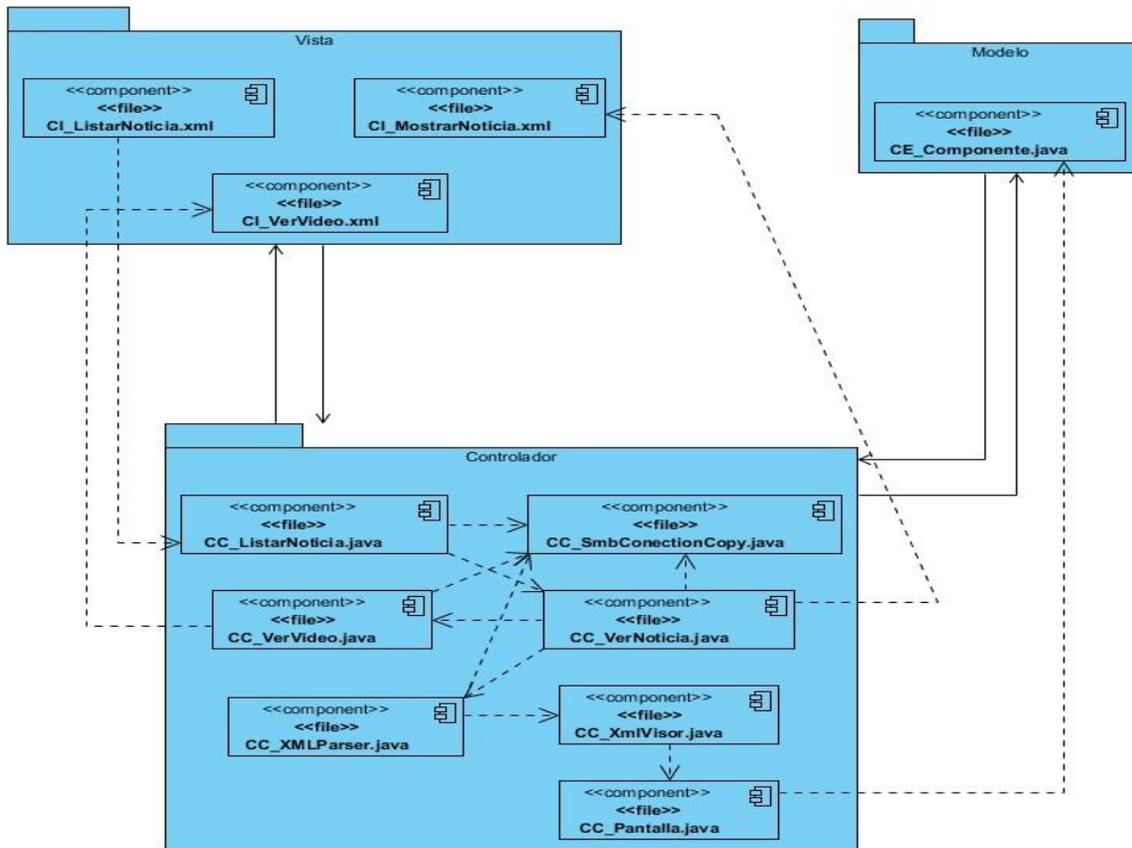


Figura 8. Diagrama de Componentes Caso de Uso Mostrar Noticia.

### 3.2 Modelo de Despliegue

El modelo de despliegue es un modelo de objetos que describe la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre los nodos de cómputo. Cada nodo representa un recurso de cómputo, normalmente un procesador o un dispositivo hardware similar. Puede describir diferentes configuraciones de red, incluidas las configuraciones para prueba y para simulación. (Jacobson, y otros, 2000). A continuación la figura 8 muestra el diagrama de despliegue del Visor Móvil de Noticias.

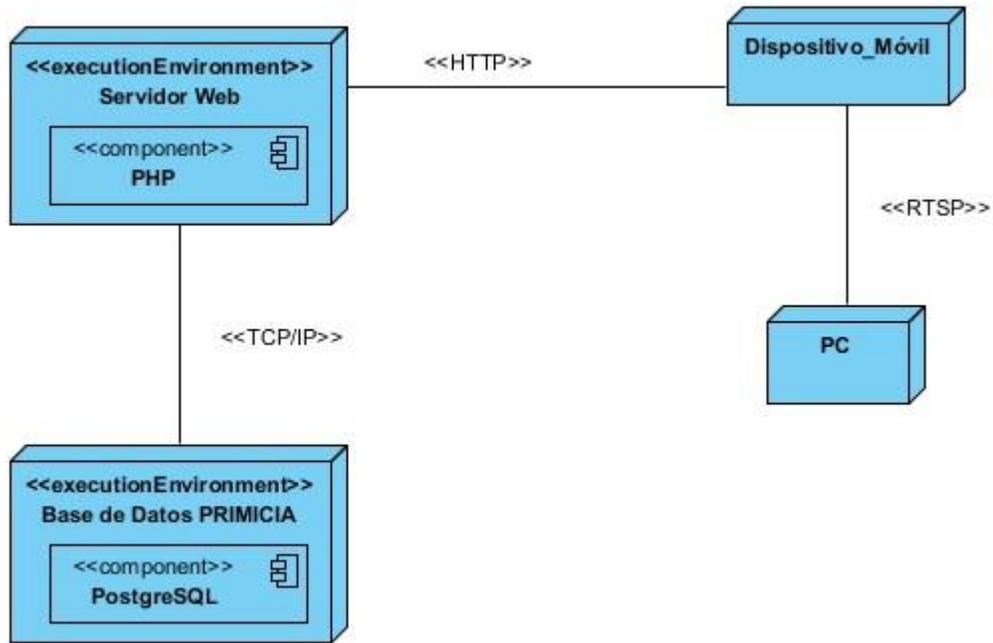


Figura 9. Diagrama de Despliegue.

En el diagrama realizado se representan 4 nodos principales. El nodo Dispositivo\_Móvil que requiere del el sistema operativo Android, el nodo Base de Datos Primicia el cual debe tener instalado el Sistema Gestor de Base de Datos PostgreSQL, el nodo Servidor Web en el cual debe estar implementados servicios web del tipo REST y el nodo PC desde donde se va a transmitir el streaming de PRIMICIA. El nodo Dispositivo\_Móvil estará conectado al nodo Servicios Web mediante el protocolo de transferencia HTTP, la conexión entre nodo Servicios Web y el nodo Base de Datos PRIMICIA se realizará a través del protocolo de comunicación TCP/IP y la comunicación entre el nodo Dispositivo\_Móvil y el nodo PC se realizará mediante el protocolo de transmisión RTSP.

### 3.3 Código Fuente

El código fuente de un sistema informático es un conjunto de líneas de texto que son las instrucciones que debe seguir la computadora para así poder ejecutar las funcionalidades requeridas por el software. El código fuente varía según el lenguaje de programación, y debe ser traducido al lenguaje máquina para que sea ejecutado.

### 3.3.1 Estándar de codificación

Un estándar de codificación, es un término que describe cuales deben ser los lineamientos y convenciones para escribir código en un lenguaje determinado. Dicho estándar varía de lenguaje en lenguaje ya que los lineamientos de una persona, no precisamente son los lineamientos de otra (Estandares, 2014).

#### Estilo de codificación utilizado:

- ❖ Los nombres de las variables estarán escritos con el estándar “*lowerCamelCase*”.
- ❖ Los nombres de los métodos estarán escritos con el estándar “*UpperCamelCase*”.
- ❖ El número de declaraciones por línea es de uno, es decir, se declarará cada variable en una línea distinta.
- ❖ Se debe usar una línea en blanco entre métodos, variables locales de un método y la primera sentencia.

```
OnDateSetListener inicio = new OnDateSetListener() {  
    @Override  
    public void onDateSet(DatePicker view, int year, int monthOfYear, int dayOfMonth) {  
  
        annoI = year;  
  
        mesI = monthOfYear;  
  
        diaI = dayOfMonth;  
  
        System.out.print(""+annoI+" "+mesI+" "+diaI);  
    }  
  
};
```

Figura 10. Estándar de codificación.

## 3.4 Pruebas de software

Las pruebas de software juegan un papel fundamental durante el proceso de desarrollo de software ya que son el elemento crítico para la garantía de la calidad del mismo. Aseguran que el software implementa correctamente las funcionalidades para el cual fue diseñado y que cumpla con las exigencias del cliente.

La IEEE (por sus siglas en inglés Institute of Electrical and Electronics Engineers, en español Instituto de Ingeniería Eléctrica y Electrónica) plantea que una prueba es: *“una actividad en la cual un sistema o uno de sus componentes se ejecuta en circunstancias previamente especificadas, los resultados se observan, se registran y se realiza una evaluación de algún aspecto”*.

La prueba es un conjunto de actividades que pueden planearse por adelantado y realizarse de manera sistemática. Por esta razón, durante el proceso de software, debe definirse una plantilla para la prueba del software: un conjunto de pasos que incluyen métodos de prueba y técnicas de diseño de casos de prueba específicos (Pressman, 2013).

### **3.4.1 Niveles de pruebas**

Existen varios niveles de pruebas posibles a realizar entre las que se encuentran: las unitarias, de integración, de sistema y de aceptación. Las pruebas unitarias son las que tienen como objetivo aislar cada parte del programa y mostrar que las partes individuales son correctas. En las pruebas de integración es donde los módulos individuales de software son combinados y probados como un grupo.

Por otra parte las pruebas del sistema tienen como objetivo verificar el sistema software para comprobar si este cumple sus requisitos, dentro de esta fase pueden desarrollarse distintos tipos de pruebas en función de los objetivos de las mismas como son: pruebas funcionales, pruebas de usabilidad, pruebas de rendimiento y pruebas de seguridad. También pueden realizarse las pruebas de aceptación las cuales tienen como función validar el desarrollo del componente de presentación de informaciones climáticas.

### **Tipos de Prueba**

Para comprobar las operaciones a realizar por la herramienta se aplicaron las pruebas de sistema donde se utilizó el tipo de pruebas funcionales.

- ❖ **Pruebas funcionales:** prueban una funcionalidad completa, donde pueden estar implicadas una o varias clases y la propia interfaz de usuario.

## Método de Prueba

### Prueba de caja negra

La cual se refiere a las pruebas que se llevan a cabo sobre la interfaz del software, por lo que pretenden demostrar que las funciones del software son operativas, que la entrada se acepta de forma adecuada, que se produce una salida correcta y que la integridad de la información externa se mantiene (Pressman, 2013).

Dentro del método de caja negra la técnica de la partición de equivalencia es una de las más efectivas pues permite examinar los valores válidos e inválidos de las entradas existentes en la herramienta, descubre de forma inmediata una clase de errores que, de otro modo, requerirían la ejecución de muchos casos antes de detectar el error genérico (Pressman, 2013).

#### 3.4.2 Diseño de Casos de Prueba

Los casos de pruebas se realizan con el objetivo de determinar que una funcionalidad ha sido implementada satisfaciendo las necesidades del cliente. Para cada caso de uso debe estar asociado un caso de prueba que recoja la especificación de ese caso de uso, dividido en secciones y escenarios, detallando las funcionalidades descritas en él y describiendo cada variable que recoge el caso de uso en cuestión (Pressman, 2013).

A continuación se detallan las variables que se encuentran asociadas al caso de uso Buscar Noticias en la siguiente tabla.

Tabla 5. Descripción de las variables asociadas al caso de uso Buscar Noticias

No	Nombre del Campo	Clasificación	Valor Nulo	Descripción
1	Título	Campo texto	No	Cadena de texto con longitud máxima de 65.
2	Fecha Inicio	Campo fecha	No	Campo de tipo fecha.

3	Fecha Fin	Campo fecha	No	Campo de tipo fecha.
4	Temática	Campo lista	No	Campo lista. Se selecciona una de las temáticas disponibles.

Esta descripción permitió que se realizara una matriz de datos, donde se evaluó y probó la validez de cada uno de los datos introducidos en el sistema, específicamente en la sección que se estuvo probando. Utilizando un juego de datos válidos e inválidos se identificó el empleo de la técnica de partición de equivalencia.

**Matriz de datos Escenario Buscar Noticias por Título**

Tabla 6. Caso de prueba del caso de uso Buscar Noticia por Título

Escenario	Variables (Enumeradas según la descripción anterior)				Descripción	Respuesta del sistema	Flujo Central
	1	2	3	4			
EC 1.1 Buscar por título	V	N/A	N/A	N/A	En este escenario se realiza la búsqueda de una noticia en el sistema correctamente.	El sistema muestra el listado de las noticias que coincidan con el título insertado.	1. Visor de Noticias/Buscar Noticias/Buscar por Título (La opción de búsqueda por título es la última de la lista de búsqueda). 2. Se llena el campo de título correctamente. 3. Se presiona el botón Buscar.
EC 1.2 Buscar por título con datos en blanco	I	N/A	N/A	N/A	En este escenario se realiza la búsqueda de una noticia en el sistema dejando el campo de título en blanco.	El sistema muestra una notificación de error en el título. Ej: "El campo título es obligatorio".	1. Visor de Noticias/Buscar Noticias/Buscar por Título (La opción de búsqueda por título es la última de la lista de búsqueda). 2. Se llena el campo de título incorrectamente. 3. Se presiona el botón Buscar.

**3.4.3 Resultados de las pruebas**

Luego de realizar las pruebas de caja negra a los casos de pruebas asociados a los diferentes casos de uso, se obtuvo como resultado el correcto funcionamiento de la herramienta así como la validación de los campos de entrada asociados a cada una de estas. Al finalizar fueron detectadas 4 no conformidades las cuales fueron resueltas a medida que fue avanzando el proceso de pruebas.

### **3.5 Conclusiones del capítulo.**

En el desarrollo del presente capítulo se realizó el diagrama de implementación del Visor de Noticias de la plataforma PRIMICIA para dispositivos móviles con el propósito de mostrar los componentes del sistema y sus relaciones, en la elaboración del diagrama de componentes se describió la composición física en que va a estar distribuido el sistema. A través del modelo de despliegue Se especificó el uso de los estándares de codificación para lograr un estilo claro y organizado del código durante la fase de implementación. Para comprobar el completo funcionamiento del Visor de Noticias se realizaron las pruebas funcionales a través la aplicación del método de caja negra a través de la técnica de prueba de particiones equivalentes permitiendo resolver todas las no conformidades.

## **Conclusiones generales**

Una vez culminado el estudio para lograr la creación del Visor de Noticias de la plataforma PRIMICIA para dispositivos móviles se concluyó que: los medios de difusión masiva a través del uso de las nuevas tecnología han encontrado una forma para mantener informada a las persona de manera simultánea e inmediata a través de los medios de visualización de noticias; las tecnologías y dispositivos móviles son una alternativas de las nuevas tecnologías en cuanto al acceso a la información ya que permiten acceder a cualquier información disponible sin importar el lugar que se encuentren, en el análisis de los visores existentes se definió que estos no responden a las necesidades del problema a resolver por las características que poseen, a través del uso de los servicios web se logró el intercambio de información entre el Visor de Noticias y la base de datos de PRIMICIA. El análisis y diseño de las clases de la aplicación permitió un mejor entendimiento de la propuesta solución, se realizó la implementación de un Visor de Noticias para dispositivos móviles con sistema operativo Android dándole respuesta a la problemática planteada. Teniendo en cuenta los resultados obtenidos, se concluye que los objetivos presentados al inicio de la investigación han sido cumplidos en su totalidad.

## **Recomendaciones**

- ❖ Se recomienda migrar el sistema para otros sistemas operativos diferentes de Android como IOS y Windows Mobile.

## Bibliografía

1. **Báez, Manuel, y otros. 2011.** Introduccion a Andriod. *UCM- Univerdad Complutense Madrid.* [En línea] 2011. <http://pendientedemigracion.ucm.es/info/tecnomovil/documentos/android.pdf>. 978-84-96285-39-5.
2. **Booch, Grady, Jacobson, Ivar y Rumbaugh, James. 2000.** El Proceso Unificado de Desarrollo de Software : s.n., 2000.
3. **Carrillo Pérez, Isaías, Pérez González, Rodrigo y Rodríguez Martín, David Aureliano. 2008.** *Metodología de desarrollo del software.* 2008.
4. **CASE. 2010.** ecured. [En línea] 2010. <http://www.ecured.cu/index.php/CASE>.
5. **Castagnola, Graciela. 2009.** LAS TICS Y LOS MEDIOS DE COMUNICACION. *Tecnologías de la Información y Comunicación - USAL Rosario.* [En línea] 12 de Noviembre de 2009. <http://ticusal.blogspot.com/2009/11/las-tics-y-los-medios-de-comunicacion.html>.
6. **Cueto, Juan José Flores y Bertolotti Zuñiga, Carmen .** DIAGRAMA DE CLASES EN UML. *DIAGRAMA DE CLASES EN UML.* [En línea]
7. **Eclipse, IDE. 2014.** [En línea] 10 de enero de 2014. <http://www.genbetadev.com/herramientas/eclipse-ide>.
8. **Estandares. 2014.** [En línea] 9 de Noviembre de 2014. [codigofeliz.com/estándares.html](http://codigofeliz.com/estándares.html).
9. **Ferrás, Claudia Beatriz Larramendi. 2012.** *Propuesta de arquitectura para desarrollo de aplicaciones compuestas para dispositivos móviles con Android.* La Habana : s.n., 2012.
10. **García, Alan. 2014.** Introducción al lenguaje PHP (Parte 1). [En línea] 29 de Agosto de 2014. <http://www.altabulador.com/introduccion-al-lenguaje-php-parte1/>.
11. **Gonzalez, Juan Jose Jimenez. 2014.** Netbeans IDE 8.0: ya disponible en versión beta. *Netbeans IDE 8.0: ya disponible en versión beta.* [En línea] 7 de Febrero de 2014. <http://www.genbetadev.com/herramientas/netbeans-ide-8-0-ya-disponible-en-version-beta>.
12. **Guevara, Jorge Martínez Ladrón de. 2014.** [En línea] 2014. <http://pendientedemigracion.ucm.es/info/tecnomovil/documentos/fjava.pdf>.

13. **Gutiérrez, Javier J. 2012.** ¿Qué es un framework web? [En línea] 2012.  
<http://www.cssblog.es/guias/Framework.pdf>.
14. **IA, Dept. Ciencia de la Computación e. 2013.** Introducción a los Servicios Web. Invocación de servicios web SOAP. [En línea] 2013.  
<http://www.jtech.ua.es/j2ee/publico/servc-web-2012-13/sesion01-apuntes.html>.
15. **IAN Sommerville. 2012.** Ingeniería del Software. *Ingeniería del Software*. [En línea] 2012.  
<http://www.filecrop.com/ian-sommerville-ingenieria-de-software-septima-edicion.pdf.html>.
16. **Insunza, Mario y Villanueva, Miguel. 2011.** *The Eclipse Foundation*. [En línea] 2011.  
[http://epf.eclipse.org/wikis/openupsp/openup\\_basic/guidances/roadmaps/openup\\_basic\\_roadmap,\\_vEruwNrEdqiM\\_](http://epf.eclipse.org/wikis/openupsp/openup_basic/guidances/roadmaps/openup_basic_roadmap,_vEruwNrEdqiM_).
17. **IVAR JACOBSON. 2013.** CASO DE USO 2.0. [En línea] 2013.  
[www.ivarjacobson.com/download.ashx?id=2018](http://www.ivarjacobson.com/download.ashx?id=2018).
18. **Jacobson, Ivar. 2013.** [En línea] 2013. [www.ivarjacobson.com/download.ashx?id=2018](http://www.ivarjacobson.com/download.ashx?id=2018).
19. **Jacobson, Ivar, Booch, Grady y Rumbaugh, James. 2000.** *El Proceso Unificado de Desarrollo de Software*. 2000.
20. **Larman, Craig. 2013.** UML y Patrones. *UML y Patrones*. [En línea] 2013.  
<http://www.ceneinnova.com/eddyesanchez/archivos/ads/UML%20y%20Patrones%20%202da%20Edicion.pdf>.
21. **Ledesma, Anibal. 2013.** ¿ Que es WampServer ? [En línea] 29 de Noviembre de 2013.  
[https://prezi.com/h0j-\\_95vhkj2/que-es-wampserver/](https://prezi.com/h0j-_95vhkj2/que-es-wampserver/).
22. **Maldonado, Daniel. 2012.** El CoDiGo K. Qué son los IDE de programación. [En línea] 2012.
23. **Manandhar, Geshan. 2014.** Geshan's Blog. *3 Bundles to get started with REST in Symfony 2 and some tips*. [En línea] 18 de Febrero de 2014. [geshan.blogspot.com/2014/07/3-bundles-to-get-started-with-rest-in.html](http://geshan.blogspot.com/2014/07/3-bundles-to-get-started-with-rest-in.html).
24. **NordicAPIs. 2015.** Soap vs. Rest A NordicAPIs infographic. [En línea] 20 de enero de 2015.  
<http://nordicapis.com/rest-vs-soap-nordic-apis-infographic-comparison/>.

25. **Orallo, Enrique Hernández. 2011.** El Lenguaje Unificado de Modelado (UML). [En línea] 2011. [www.disca.upv.es/enheror/pdf/ActaUML.PDF](http://www.disca.upv.es/enheror/pdf/ActaUML.PDF)..
26. **Otero, Andrés Molina. 2014.** Webinar: "Symfony2: un framework para maximizar tu productividad en desarrollo web". *Webinar: "Symfony2: un framework para maximizar tu productividad en desarrollo web"*. [En línea] 2014. <http://www.iebschool.com/webinars/webinar-symfony2-framework-maximizar-productividad-desarrollo-web/>.
27. **Pastrana, Lourdes Velázquez. 2013.** [En línea] 2013. <http://www.enterate.unam.mx/Articulos/2013/agosto/telefcel.html>.
28. **Pérez, Jorge Luis Uribe. 2014.** *PROGRAMACIÓN PARA DISPOSITIVOS MÓVILES*. Tijuana, Baja California : s.n., 2014.
29. **PostgreSQL-es. 2012.** [En línea] 2012. <http://www.postgresql.org.es/>.
30. **Press, (RSS). 2014.** Aplicaciones para Andriod en Google play. [En línea] 2014 de enero de 2014. [https://play.google.com/store/apps/details?id=com.twentyfivesquares.press&hl=es\\_419](https://play.google.com/store/apps/details?id=com.twentyfivesquares.press&hl=es_419).
31. **Pressman . 2011.** Ingeniería del Software 6ta edición. [En línea] 2011. <http://ebookbrowse.com/roger-s-pressman-ingenieria-del-software-v-ed-pdf-d47183684>.
32. **Pressman, Roger. 2013.** [En línea] 2013. <http://es.slideshare.net/jes4791/ingenieria-del-software-un-enfoque-practico>.
33. **Pulse, LinkedIn. 2015.** Aplicaciones para Andriod en Google play. [En línea] 6 de abril de 2015. [https://play.google.com/store/apps/details?id=com.alphonso.pulse&hl=es\\_419](https://play.google.com/store/apps/details?id=com.alphonso.pulse&hl=es_419).
34. **Republic, News. 2015.** Aplicaciones para Andriod en Google play. [En línea] 9 de abril de 2015. [https://play.google.com/store/apps/details?id=com.mobilesrepublic.appy&hl=es\\_419](https://play.google.com/store/apps/details?id=com.mobilesrepublic.appy&hl=es_419).
35. **Rouse, Margaret. 2014.** REST (representational state transfer). [En línea] 2014. <http://searchsoa.techtarget.com/definition/REST>.
36. **—. 2014.** Searchsoa Techtarge. *SOAP (Simple Object Access Protocol) definition*. [En línea] Diciembre de 2014. <http://searchsoa.techtarget.com/definition/SOAP>.

37. **Silberschatz, Abraham, Galvin, Peter Baer y Gagne, Greg. 2006.** *Fundamentos de Sistemas Operativos, 7ma Edición.* Madrid : McGraw-Hill, 2006. 84-481-4641-7.
38. **Sosa, Yanary Hernández. 2012.** *Desarrollo del Componente de presentación de informaciones climáticas para la Plataforma de Televisión Informativa, PRIMICIA.* La Habana : s.n., 2012.
39. **Stallings, William. 1997.** *Sistemas Operativos, 2ed.* Madrid : PRENTICEHAL L, 1997. 84-89660-22-0.
40. **Valencia, Arturo Patricio Quiroz. 2011.** *Inclusión de las TICs como eje transversal en el Bachillerato en Ciencias.* Quito, Ecuador : s.n., 2011.
41. **Valencia, Isabel. 2012.** [En línea] 2012. <http://www.androidpit.es/sdk-android>.
42. **Vogel, Lars. 2015.** *Android Development with Android Studio or Eclipse ADT - Tutorial.* *Android Development with Android Studio or Eclipse ADT - Tutorial.* [En línea] 12 de 01 de 2015. <http://www.vogella.com/tutorials/Android/article.html>.

## Anexos

### Anexo 1. Diagrama de Secuencia del Caso de Uso Listar Noticia.

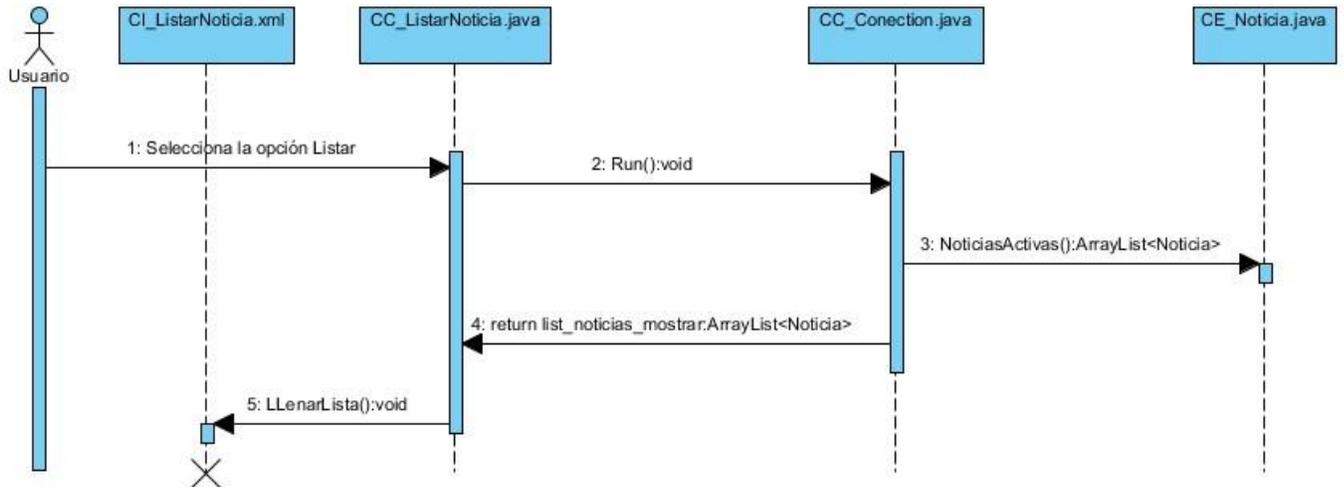


Figura 11. Diagrama de Secuencia del Caso de Uso Listar Noticia.

### Anexo 2. Diagrama de Secuencia del Caso de Uso Visualizar Transmisión.

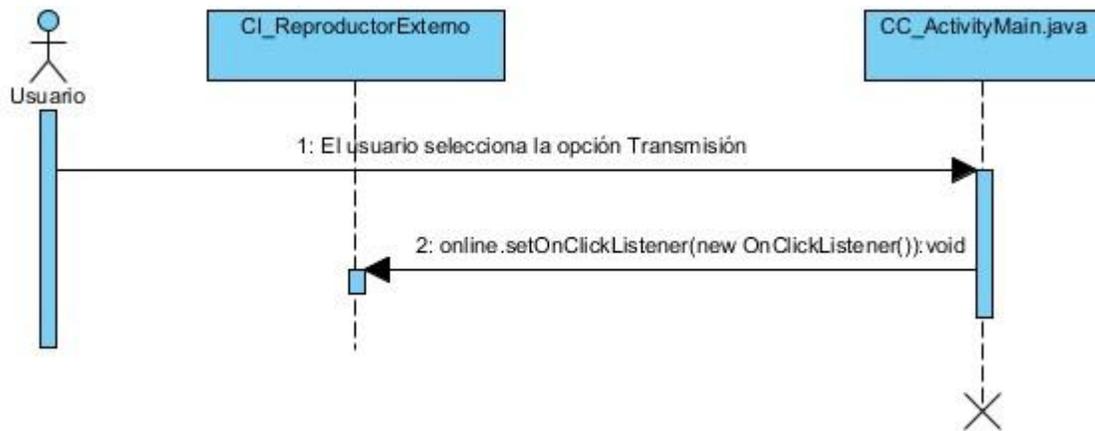


Figura 12. Diagrama de Secuencia del Caso de Uso Visualizar Transmisión.

**Anexo 3. Diagrama de Secuencia del Caso de Uso Buscar Noticia.**

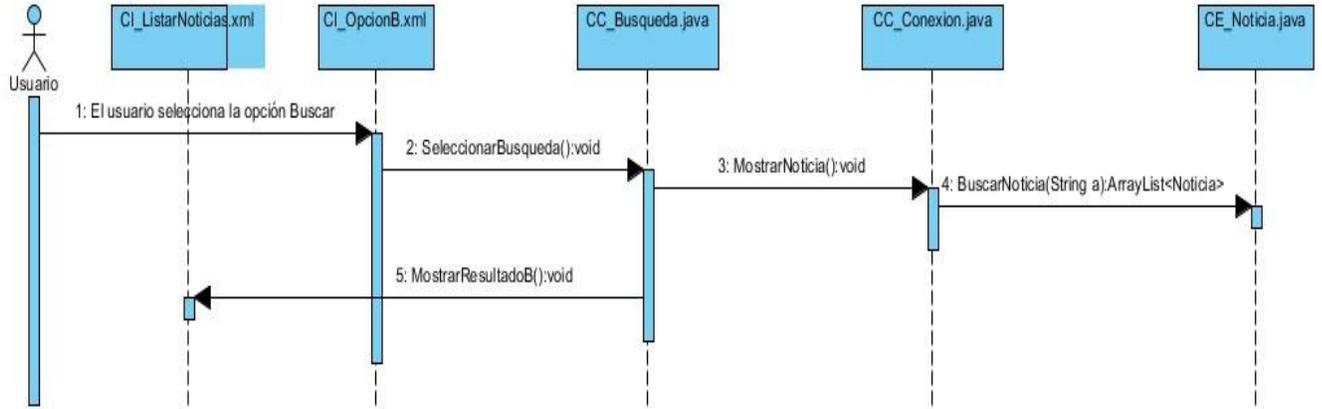


Figura 13. Diagrama de Secuencia del Caso de Uso Buscar Noticia.

**Anexo 4. Diagrama de Componentes del Caso de Uso Listar Noticia.**

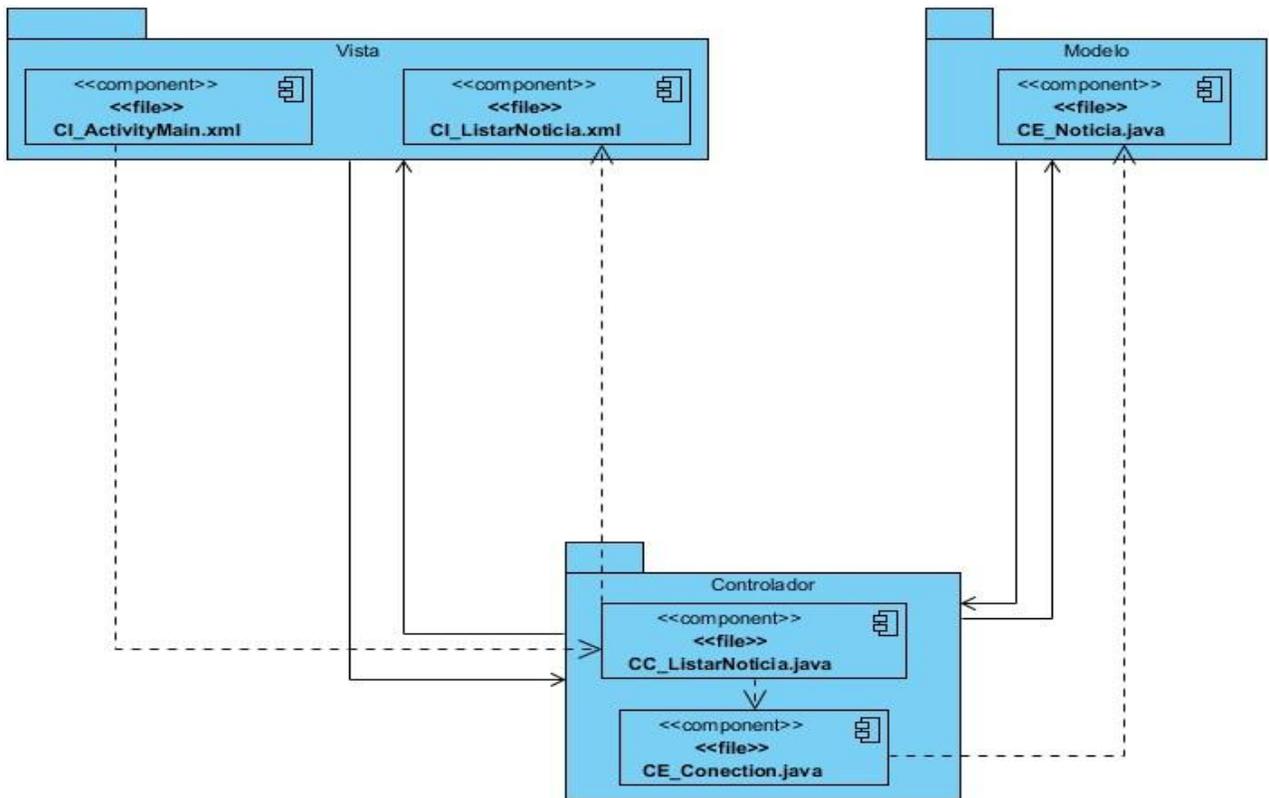


Figura 14. Diagrama de Componentes del Caso de Uso Listar Noticia.

**Anexo 5. Diagrama de Componentes del Caso de Uso Buscar Noticia.**

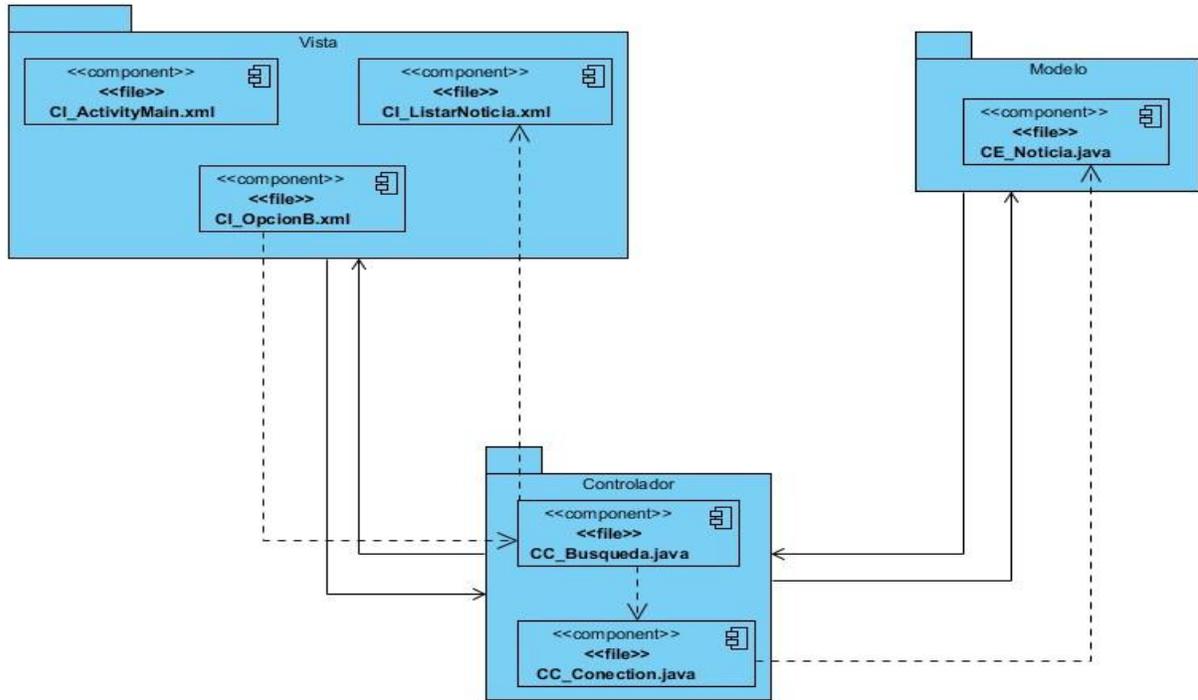


Figura 15. Diagrama de Componentes del Caso de Uso Buscar Noticia.

**Anexo 6. Diagrama de Componentes Caso de Uso Visualizar Transmisión.**

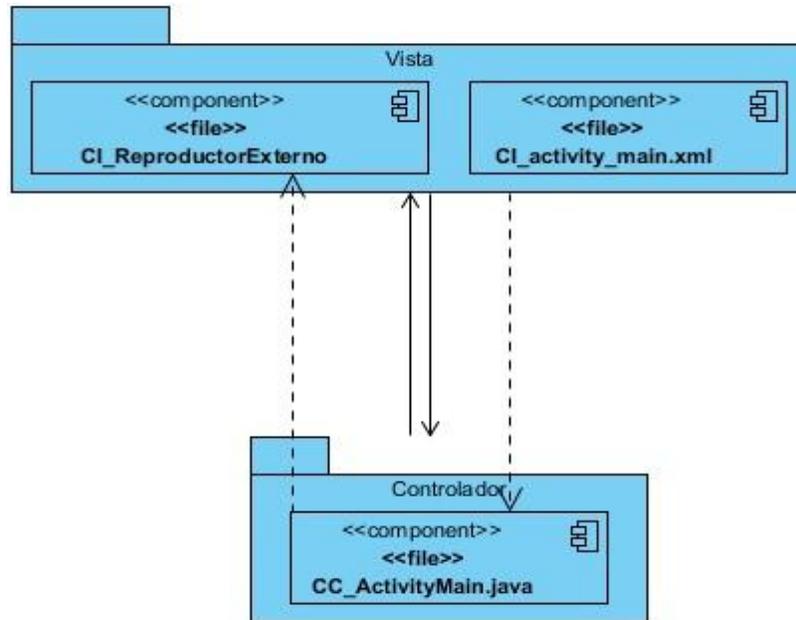


Figura 16. Diagrama de Componentes de Caso de Uso Visualizar Transmisión.

Anexo 7. Diagrama de Clases CU: Listar.

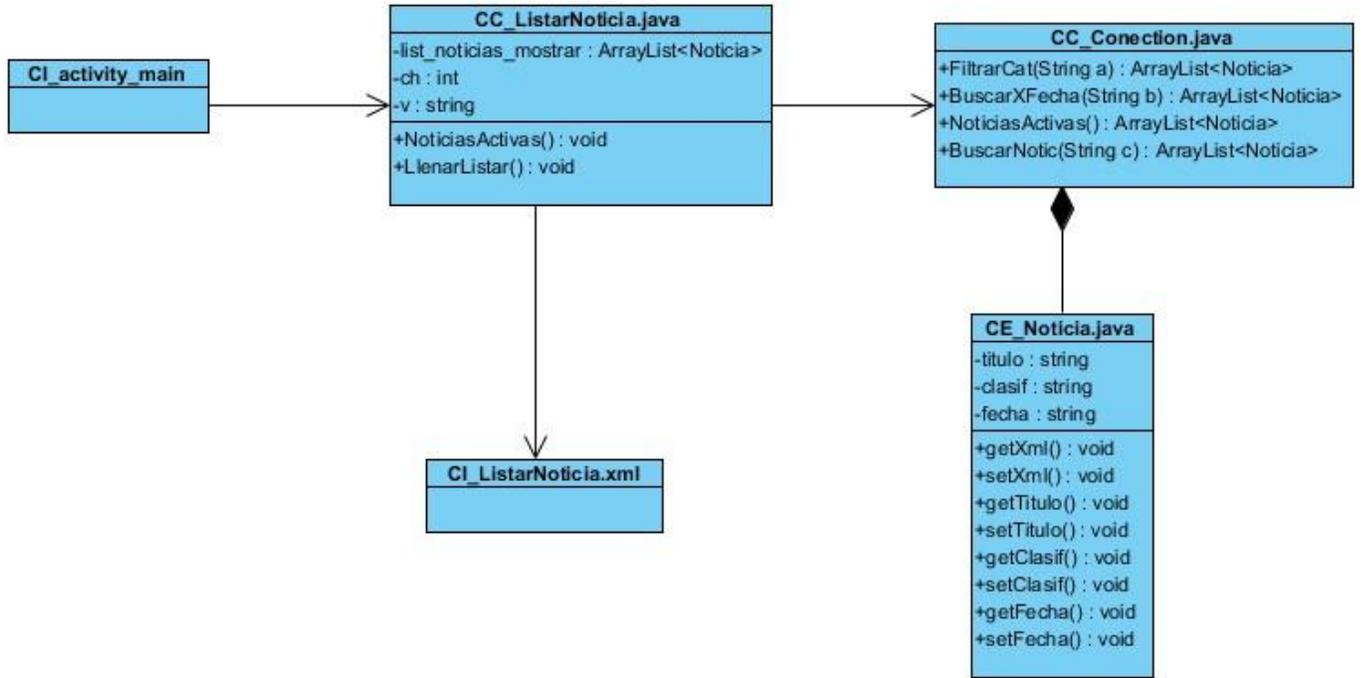


Figura 17. Diagrama de Clases CU Listar.

Anexo 8. Diagrama de Clases CU: Buscar.

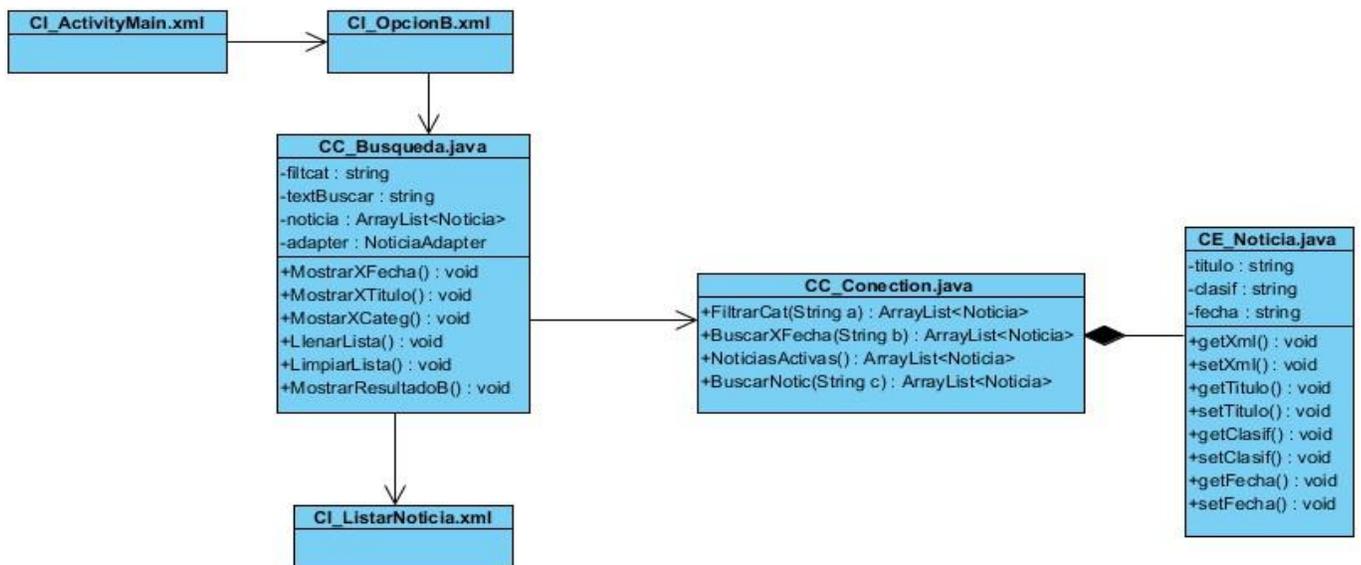


Figura 18. Diagrama de Clases CU Buscar.

**Anexo 9. Diagrama de Clases CU Visualizar Transmisión.**

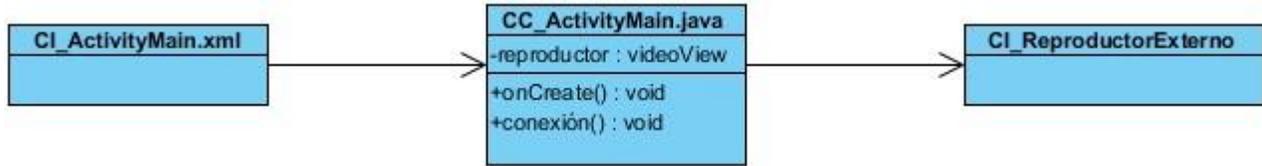


Figura 19. Diagrama de Clases CU Visualizar Transmisión.

**Anexo 10. Diseño del caso de prueba Buscar Noticia por fecha.**

Tabla 7. Diseño del caso de prueba Buscar Noticia por fecha

Escenario	Variables (Enumeradas según descripción anterior)				Descripción	Respuesta del sistema	Flujo Central
	1	2	3	4			
EC 2.1 Buscar por Fecha	N/A	V	V	N/A	En este escenario se realiza la búsqueda de una noticia en el sistema correctamente.	El sistema muestra el listado de las noticias que estén dentro del rango de fecha seleccionado.	1. Visor de Noticias/Buscar Noticias/Buscar por Fecha (La opción de búsqueda por fecha es la segunda de la lista de búsqueda). 2. Se introducen las fechas correctas. 3. Se presiona el botón Buscar.
EC 2.2 Buscar por Fecha fuera de rango.	N/A	V	I	N/A	En este escenario se realiza la búsqueda de una noticia en el	El sistema muestra una notificación de error en el título. Ej: "La	1. Visor de Noticias/Buscar Noticias/Buscar por Fecha (La opción de búsqueda por título es la última de la lista de búsqueda).

					sistema introduciendo una fecha fuera de rango.	fecha introducida se refiere a una ubicación en el futuro”.	2. Se introducen las fechas correctas. 3. Se presiona el botón Buscar.
--	--	--	--	--	---	---	---

**Anexo 11. Descripción de las variables asociadas al caso de uso Visualizar Transmisión.**

Tabla 8. Descripción de las variables asociadas al caso de uso Visualizar Transmisión.

No	Nombre del Campo	Clasificación	Valor Nulo	Descripción
1	Número IP	Campo texto	No	Cadena de números con longitud máxima 15 caracteres con el formato xxx.xxx.xxx.xxx.
2	Nombre del Streaming.	Campo texto	No	Cadena de caracteres.

**Anexo 12. Diseño del caso de prueba Visualizar Transmisión sección Cambiar configuración.**

Tabla 9. Diseño del caso de prueba Cambiar configuración

Escenario	Variables (Enumeradas según la descripción anterior)		Descripción	Respuesta del sistema	Flujo Central
	1	2			

EC 1.1 Cambiar configuración	V	V	En este escenario se cambia la configuración del streaming recibido.	El sistema muestra una notificación de datos guardados correctamente. Ej: "Datos guardados correctamente".	<ol style="list-style-type: none"> <li>1. Visor de Noticias/Buscar Noticias/Buscar por Fecha (La opción de búsqueda por fecha es la segunda de la lista de búsqueda).</li> <li>2. Se introducen las fechas correctas.</li> <li>3. Se presiona el botón Buscar.</li> </ol>
EC 1.2 Cambiar configuración con ip incorrecto.	I	V	En este escenario se cambia la configuración del streaming recibido con el ip incorrecto.	El sistema muestra una notificación de datos guardados correctamente. Ej: "Datos guardados correctamente".	<ol style="list-style-type: none"> <li>1. Visor de Noticias/Buscar Noticias/Buscar por Fecha (La opción de búsqueda por título es la última de la lista de búsqueda).</li> <li>2. Se introducen las fechas correctas.</li> <li>3. Se presiona el botón Buscar.</li> </ol>

### Anexo 13. Entrevista

Fecha: Diciembre 2014.

Entrevistador: Leonel Martínez Pulido

Entrevistados: Ing. Félix Iván Romero Rodríguez (Líder del Proyecto PRIMICIA), Ing. Mairena Arzuaga Limonta.

#### Preguntas:

1- ¿Qué beneficio aportaría para la plataforma PRIMICIA un visor de noticias que les permitiera a los usuarios acceder a las noticias publicadas desde cualquier lugar?

2- ¿Cuáles son los requisitos que debería cumplir el visor de noticias para convertirse en una herramienta de fácil uso para los usuarios?

**Respuestas:**

Pregunta 1: Después de analizadas las respuestas obtenidas se pudo apreciar que ambos entrevistados coincidieron en que el mayor beneficio que el visor aportaría a la plataforma PRIMICIA es que ampliaría su forma de accesibilidad logrando que las persona pueda ver las noticias transmitidas por PRIMICIA desde cualquier lugar.

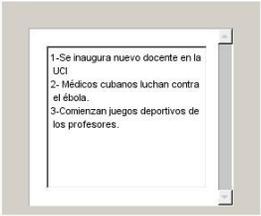
Pregunta 2: Ambos entrevistados mencionaron varios requisitos que a su consideración debería cumplir la aplicación pero después de analizados los mismos se puedo apreciar que ambos coincidían en algunos requisitos que consideraban indispensables para la aplicación los mismo son:

- ❖ Listar noticias activas.
- ❖ Mostrar pantallas con el contenido de la noticia seleccionada.
- ❖ Buscar noticias por título.
- ❖ Buscar noticias por fecha.
- ❖ Buscar noticias por temática.
- ❖ Cargar el flujo de transmisión.
- ❖ Visualizar el flujo de transmisión.

**Anexo 14. Descripción del CU: Listar Noticias.**

Tabla 10 Caso de Uso: Listar Noticias

<b>Caso de Uso</b>	Listar Noticias
<b>Actores</b>	Usuario
<b>Propósito</b>	Este caso de uso se lleva a cabo con el objetivo de mostrarle al usuario una lista de noticias.
<b>Resumen</b>	El Caso de uso inicia cuando el actor selecciona la opción Lista Noticias. La aplicación muestra una lista de noticias activas. El caso de uso termina cuando se muestra la lista de noticias activas.

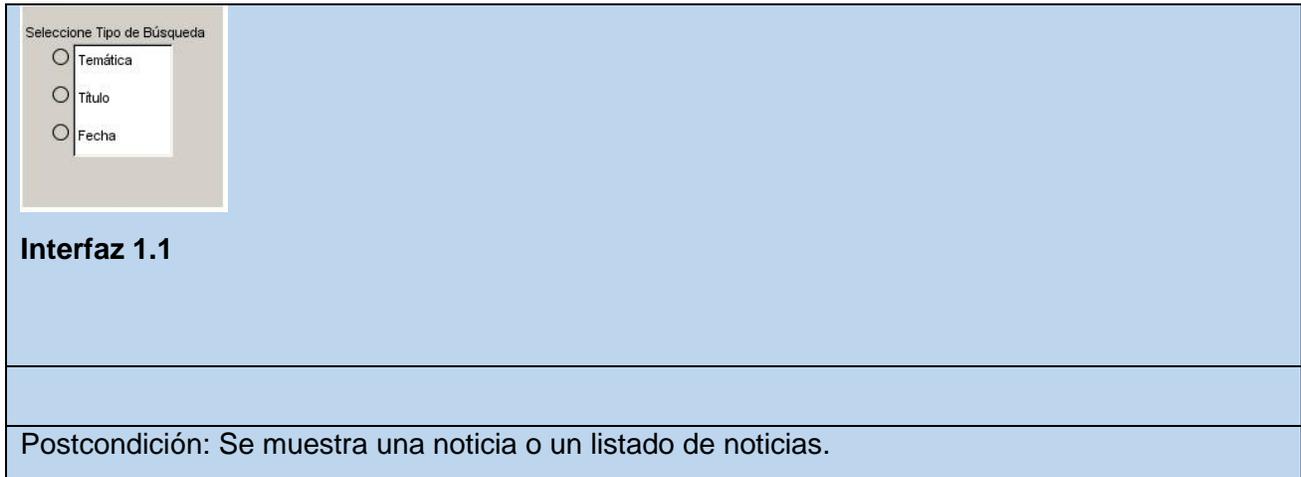
<b>Complejidad</b>	Media
<b>Prioridad</b>	Crítico
<b>Precondiciones</b>	El usuario debe seleccionar la opción Listar Noticias.
<b>Referencias</b>	RF1
<b>Flujo Normal de Eventos</b>	
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>
1. El caso de uso se inicia cuando el actor selecciona la opción Listar Noticias del Visor de Noticias	2. El sistema muestra una lista con las noticias activas terminando así el caso de uso.
<b>Prototipo de Interfaz</b>	
 <p>1-Se inaugura nuevo docente en la UCI 2- Médicos cubanos luchan contra el ébola. 3-Comienzan juegos deportivos de los profesores.</p>	
<b>Interfaz 1.1</b>	
Postcondición: Se muestra el contenido de la noticia seleccionada.	

**Anexo 15. Descripción del CU: Buscar Noticias.**

Tabla 11 Caso de Uso: Buscar Noticias

<b>Caso de Uso</b>	Buscar Noticia
<b>Actores</b>	Usuario
<b>Propósito</b>	Este caso de uso se lleva a cabo con el objetivo de mostrarle al usuario una noticia o un listado de noticia según el criterio de

	búsqueda seleccionado.
<b>Resumen</b>	El Caso de uso inicia cuando el actor selecciona la opción Buscar Noticia. La aplicación muestra los criterios de búsqueda (título, fecha, temática). El usuario selecciona el criterio de búsqueda deseado. El caso de uso termina cuando el sistema muestra una noticia o una lista de noticias según el criterio de búsqueda seleccionado.
<b>Complejidad</b>	Media
<b>Prioridad</b>	Normal
<b>Precondiciones</b>	El usuario debe haber seleccionado la opción Buscar Noticia.
<b>Referencias</b>	RF3, RF4, RF5
<b>Flujo Normal de Eventos</b>	
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>
1. El caso de uso se inicia cuando el actor selecciona la opción Buscar Noticia en el Visor de Noticias.	2. El sistema muestra la opción Buscar Noticia con los criterios de búsqueda: <ul style="list-style-type: none"> <li>➤ Título</li> <li>➤ Fecha</li> <li>➤ Temática</li> </ul>
3. El usuario selecciona el criterio de búsqueda deseado.	4. El sistema muestra una noticia o un listado de noticias según el criterio de búsqueda seleccionado por el usuario. En caso de no encontrar alguna noticia el sistema muestra un mensaje “No se encontró ninguna noticia según el criterio seleccionado” terminando el caso de uso.
<b>Prototipo de Interfaz</b>	



**Anexo 14. Descripción del CU: Visualizar Transmisión.**

Tabla 12 Descripción del CU Visualizar Noticias

<b>Caso de Uso</b>	Visualizar Transmisión
<b>Actores</b>	Usuario
<b>Propósito</b>	Este caso de uso se lleva a cabo con el objetivo de mostrarle al usuario el flujo de trasmisión de primicia.
<b>Resumen</b>	El Caso de uso inicia cuando el actor selecciona la opción Mostrar Transmisión. La aplicación muestra el flujo de transmisión del canal de trasmisión primicia. El caso de uso termina cuando el usuario detiene la reproducción del flujo de transmisión.
<b>Complejidad</b>	Media
<b>Prioridad</b>	Normal
<b>Precondiciones</b>	El usuario debe haber seleccionado la opción Mostrar Transmisión.
<b>Referencias</b>	RF6, RF7
<b>Flujo Normal de Eventos</b>	
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>
1. El caso de uso se inicia cuando el actor selecciona la opción Mostrar Transmisión en el Visor	2. El sistema muestra en un reproductor el flujo de transmisión de primicia.

de Noticias	Terminando así el caso de uso.
<b>Prototipo de Interfaz</b>	
	
<b>Interfaz 1.1</b>	
<b>Sección 2. Cambiar Configuración Transmisión</b>	
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>
<p>1. El caso de uso se inicia cuando el actor selecciona la opción cambiar configuración en el Visor de Noticias.</p>	<p>2. El sistema muestra la opción cambiar configuración</p>
<p>3. El usuario introduce los nuevos parámetros número ip y nombre del streaming.</p>	<p>4. El sistema guarda la nueva configuración muestra un mensaje "Guardado correctamente" terminando así el caso de uso.</p>
<b>Prototipo de Interfaz</b>	



Dir IP

Nombre del Streaming

**Interfaz 2.1**

Postcondición: Se muestra el flujo de transmisión de primicia.

## Glosario de Términos

**Bundles:** paquetes.

**JSON:** acrónimo de Java Script Object Notation, es un formato ligero para el intercambio de datos. JSON es un subconjunto de la notación literal de objetos de Java Script que no requiere el uso de XML.

**HTML:** Lenguaje de marcas de hipertexto (HyperText Markup Language). Es usado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes. HTML se escribe en forma de “etiquetas”, rodeadas por corchetes angulares (<,>). También puede describir, hasta un cierto punto, la apariencia de un documento, y puede incluir un script (por ejemplo Javascript), el cual puede afectar el comportamiento de navegadores Web y otros procesadores de HTML.

**HTTP:** (Hypertext Transfer Protocol) Protocolo de Transferencia de Hipertexto utilizado en la WWW para transmitir las páginas de información entre el programa navegador y el servidor. Se destaca que el HTTP seguro es un protocolo HTTP mejorado con funciones de seguridad con clave simétrica.

**XML:** Metalenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium (W3C)

**YAML:** formato de serialización de datos legible por humanos inspirado en lenguajes como XML, C, Python, Perl, así como el formato para correos electrónicos especificado por el RFC 2822.