



Universidad de las Ciencias Informáticas

Facultad 6

APLICACIÓN MÓVIL PARA LA GESTIÓN DE NOTICIAS Y MEDIAS DE LA PLATAFORMA PRIMICIA

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

Autores: Yosiel Medero Cuello

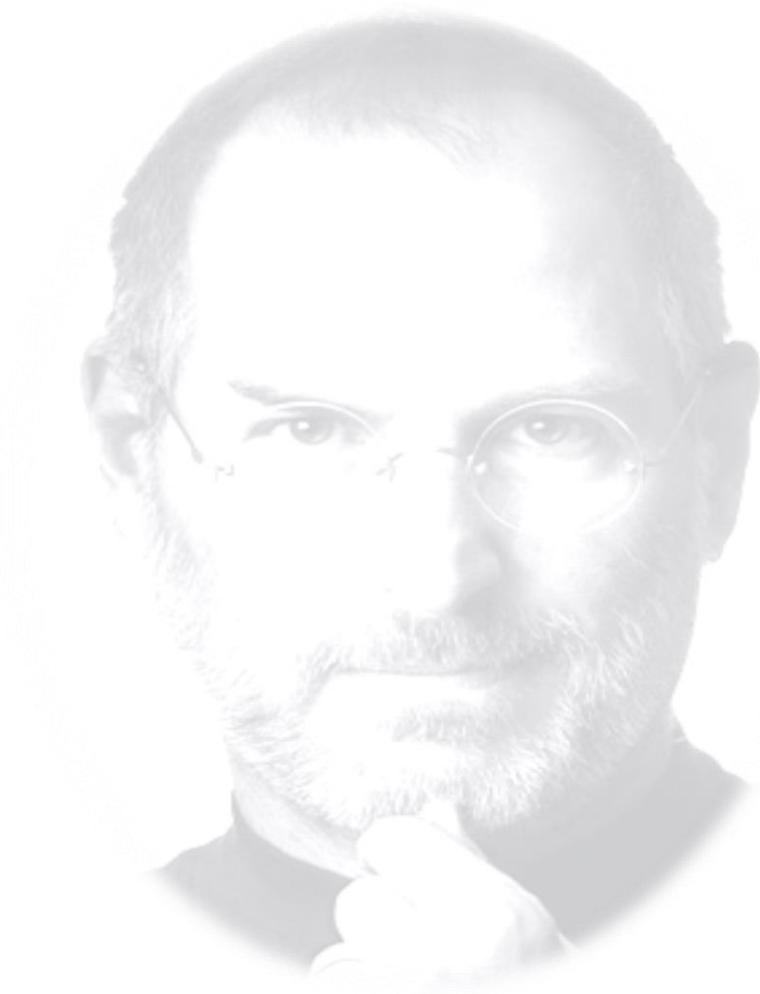
George Alexander Paneque Suárez

Tutores: Ing. Yordany Piñeiro Gómez

Ing. Aramis Romero Carballea

Ing. Felix Ivan Romero Rodríguez

La Habana, junio de 2015
“Año 57 de la Revolución”



"Sé un punto de referencia de calidad. Algunas personas no están acostumbradas a un ambiente donde la excelencia es aceptada."

Steve Jobs

DECLARACIÓN DE AUTORÍA

Declaramos ser autores de este trabajo y autorizamos a la Universidad de las Ciencias Informáticas a que haga uso de la misma en su beneficio, con carácter exclusivo.

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

Yosiel Medero Cuello

Firma del Autor

George Alexander Paneque Suárez

Firma del Autor

Ing. Yordanys Piñeiro Gómez

Firma del Tutor

Ing. Aramis Romero Carballea

Firma del Tutor

Ing. Felix Ivan Romero Rodríguez

Firma del Tutor

DATOS DE CONTACTO

Tutor: Ing. Yordanys Piñeiro Gómez

Graduado de Ingeniería en Ciencias Informáticas en la Universidad de las Ciencias Informáticas en julio de 2007.

Categoría docente: Asistente

Profesor de las asignaturas Ingeniería de Software I y II, Gestión de Software y Preparación para la Prueba de Nivel de IGSW.

Centro de Trabajo: Universidad de las Ciencias Informáticas (UCI). Centro de Innovación y Calidad de la Educación (CICE).

Cargo: Profesora. Vicerrectoría de Formación.

Dirección: Carretera a San Antonio, Km 2 ½, Reparto Torrens, La Lisa, La Habana. Universidad de las Ciencias Informáticas (UCI), Edificio: 27, Apto: 201.

Teléfono Oficina: +53 – 7 – 837 – 2522.

Correo electrónico: ypineirog@uci.cu

Tutor: Ing. Aramis Romero Carballea

Graduado de Ingeniería en Ciencias Informáticas en la Universidad de las Ciencias Informáticas en julio de 2013.

Dirección: Carretera a San Antonio, Km 2 ½, Reparto Torrens, La Lisa, La Habana. Universidad de las Ciencias Informáticas (UCI), Edificio: 75, Apto: 201.

Correo electrónico: aramis@uci.cu

Tutor: Ing. Felix Ivan Romero

Graduado de Ingeniería en Ciencias Informáticas en la Universidad de las Ciencias Informáticas en julio de 2011.

Dirección: Carretera a San Antonio, Km 2 ½, Reparto Torrens, La Lisa, La Habana. Universidad de las Ciencias Informáticas (UCI), Edificio: 150, Apto: 105.

Correo electrónico: firomero@uci.cu

El avance de las tecnologías ha provocado que muchos países incursionen en la creación de plataformas, que facilitan el proceso de creación y publicación de las noticias que se transmiten por diversos medios. Cuba no está exenta de los retos trazados, específicamente en el centro GEySED de la Universidad de las Ciencias Informáticas, se desarrolló la Plataforma de Televisión Informativa PRIMICIA. En esta se llevan a cabo los procesos de redacción de noticias y gestión de medias, para lo cual se debe trabajar desde las estaciones de trabajo con una completa dependencia al servidor de aplicaciones. Esto provoca una mayor inversión de tiempo desde que se concibe la noticia por el periodista, en un lugar donde se encuentre trabajando sin conexión, hasta que se estructura y se publica la misma. Para eliminar estas dependencias se creará una aplicación que permita redactar las noticias en dispositivos móviles. Para guiar el desarrollo del sistema se utilizó como metodología de desarrollo OpenUP¹, Visual Paradigm como herramienta CASE y UML como lenguaje de modelado. Como marco de trabajo Eclipse con el SDK² de Android y SQLite como gestor de base de datos para el almacenamiento de datos en el dispositivo. Esta aplicación móvil favorecerá la creación de noticias sin tener conexión al servidor de aplicaciones de PRIMICIA, posibilitando accesibilidad y portabilidad de la información.

Palabras claves: administración, móvil, noticias, medias

¹Proceso Unificado Abierto.

² Kit de desarrollo de software.

The development of technology has caused that many countries venture into platforms creation that facilitate the process of creating and publishing news that are transmitted by several means. Cuba is not excluded from challenges; Informative PRIMICIA platform was developed specifically in the GEySED center of the University of Informatics Sciences. In this platform are carried out news drafting and medias management processes, this work must be done from workstations with complete dependence to the application server. This causes a greater investment of time since news is conceived by a journalist, in a place where this person is working offline, until it is structured and published. To eradicate these dependencies, an application will be created that can write news on mobile devices. To guide the development of the system OpenUP was used as a development methodology, Visual Paradigm as CASE tool and UML as modeling language. Eclipse as a framework for working with the Android SDK and SQLite as database manager for storing data in the device. This mobile application will support the creation of news without server connection PRIMICIA applications, enabling information accessibility and portability.

Keywords: management, mobile, news, media

Introducción	1
Capítulo 1. Redacción de noticias y gestión de medias. Las tecnologías para su automatización	6
Introducción.....	6
1.1 Principales conceptos asociados al tema de investigación	6
1.2 Módulos de redacción de noticias y gestión de medias del subsistema de administración de PRIMICIA	7
1.3 Análisis de las posibles plataformas a utilizar para el desarrollo de aplicaciones en dispositivos móviles.....	9
1.3.1 <i>Windows Phone</i>	9
1.3.2 <i>iOS</i>	10
1.3.3 <i>Android</i>	10
1.4 Sistema operativo Android.....	11
1.4.1 <i>Características de Android</i>	12
1.4.2 <i>Componentes de una aplicación Android</i>	12
1.4.3 <i>Ventajas y desventajas de Android</i>	14
1.5 Sistemas de producción de noticias para dispositivos móviles.....	14
1.6 Tendencias y tecnologías actuales	16
1.6.1 <i>Metodología de Desarrollo OpenUP</i>	16
1.6.2 <i>Lenguaje Unificado de Modelado (UML 2.0)</i>	17
1.6.3 <i>Herramienta CASE Visual Paradigm for UML 8.0</i>	17
1.6.4 <i>Lenguaje de Programación Java</i>	18
1.6.5 <i>Entorno de Desarrollo Integrado Eclipse</i>	18
1.6.6 <i>Paquete de Desarrollo de Software (Software Development Kit, SDK) de Android</i>	19
1.6.7 <i>Herramienta de Desarrollo de Android (Android Development Tools, ADT)</i>	19
1.6.8 <i>Gestor de Base de Datos SQLite</i>	19
Conclusiones parciales.....	20
Capítulo 2. Análisis del sistema	22
Introducción.....	22
2.1 Modelo Conceptual.....	22

2.2	Requisitos de Software	24
2.2.1	<i>Especificación de Requisitos Funcionales</i>	24
2.2.2	<i>Especificación de Requisitos no Funcionales</i>	26
2.3	Modelación del Sistema	27
2.3.1	<i>Actores del sistema</i>	27
2.3.1	<i>Casos de uso del sistema</i>	28
2.3.2	<i>Descripción textual de los casos de uso del sistema</i>	30
	Conclusiones parciales.....	37
Capítulo 3.	Diseño del Sistema	38
	Introducción.....	38
3.1.	Arquitectura de Software	38
3.1.1.	<i>Estilo de Arquitectura</i>	38
3.1.2.	<i>Patrón Arquitectónico Modelo – Vista – Controlador</i>	39
3.1.3.	<i>Vista Lógica</i>	39
3.2.	Modelo de Diseño	40
3.2.1.	<i>Diagrama de Clases del Diseño</i>	40
3.2.2.	<i>Patrones de Diseño</i>	41
3.3.	Diseño de la Base de Datos.....	43
3.3.1.	<i>Diagrama Entidad - Relación</i>	44
3.4.	Diagrama de Despliegue	44
Capítulo 4.	Implementación y Pruebas.....	46
	Introducción.....	46
4.1	Modelo de Implementación	46
4.1.1	<i>Diagrama de Componentes</i>	46
4.2	Estándar de Codificación	47
4.3	Pruebas del Sistema.....	48
4.3.1	<i>Pruebas de Caja Negra</i>	48
4.3.2	<i>Resultado de las pruebas</i>	56
4.3.3	<i>Pruebas de Aceptación</i>	57

4.4 Validación de la contribución lograda.....	57
Referencias Bibliográficas	62
Anexos.....	65

Índice de tablas y figuras

FIGURAS

Figura 1 Porcentajes de teléfonos inteligentes vendidos según su SO (Tomás, 2013)	11
Figura 2 Modelo conceptual	23
Figura 3 Diagrama de casos de uso del sistema	29
Figura 4 Vista lógica	40
Figura 5 Diagrama de clases del diseño del caso de uso Gestionar Noticias	41
Figura 6 Diagrama entidad-relación	44
Figura 7 Diagrama de despliegue	45
Figura 8 Diagrama de componentes del caso de uso Gestionar Noticias	47

TABLAS

Tabla 1 Descripción de los conceptos del Modelo Conceptual	23
Tabla 2 Actores del sistema	27
Tabla 3 Casos de uso del sistema	28
Tabla 4 Caso de prueba del caso de uso “Gestionar Noticias”	50
Tabla 5 Matriz de datos SC 1: “Adicionar noticias”	53
Tabla 6 Matriz de datos SC 2: “Eliminar noticias”	54
Tabla 7 Matriz de datos SC 3: “Modificar noticias”	55
Tabla 8 Matriz de datos SC 4: “Mostrar datos de la noticia”	56

Introducción

La búsqueda constante del hombre por satisfacer sus necesidades de comunicación ha sido el impulso que ha logrado el establecimiento en el mundo de medios que son cada vez más poderosos y veloces en el proceso comunicativo. En la actualidad son numerosos los medios de comunicación con que cuenta la humanidad, formando parte de estos: la radio, la prensa escrita y digital, el internet, la televisión y la telefonía móvil. Éstos poseen características en común, asociadas a los diversos formatos en los que transmiten la información, como son: video, imagen, audios, texto, tablas y gráficos. Presentan el contenido de una forma directa, sencilla y mantienen informado a gran cantidad de público que los usan a diario.

La televisión presenta importancia en la humanidad por su fácil acceso. Permite difundir imágenes, textos, videos y sonidos, que se expresan en acontecimientos en tiempo real o almacenado de forma previa. Permanece, aún con el desarrollo del internet, como un medio muy requerido a nivel mundial y su importancia radica en el hecho de que permite conocer noticias, ofrece entretenimiento y utiliza lenguajes de comunicación para miles de personas en cualquier parte del mundo. Su alto índice de implantación la convierte en el instrumento perfecto para canalizar el mercado de la información, en torno a la posibilidad que brinda la tecnología digital de una manipulación interactiva de la información (García, 2005).

Con el creciente desarrollo de las tecnologías y la necesidad de los usuarios de mantenerse conectados, se ha integrado la televisión con otros medios para lograr mayor disponibilidad de la información; siendo uno de ellos la telefonía móvil. Éstos ostentan múltiples facilidades, dentro de las que se destaca la de mantener informados a los usuarios de los sucesos que ocurren diariamente de manera inmediata. Debido a la usabilidad y portabilidad que brindan, se ha demostrado que es más factible el proceso de comunicación entre los usuarios a través de estos.

El centro de desarrollo Geoinformática y Señales Digitales (GEySED), perteneciente a la Facultad 6 de la Universidad de las Ciencias Informáticas (UCI), desarrolló la Plataforma de Televisión Informativa PRIMICIA. Esta es una solución integral que provee un canal de televisión para la transmisión automática y constante de información en diferentes formatos. Está estructurada por dos subsistemas: Administración y Transmisión, que se relacionan entre sí y actúan como un todo para brindar un resultado final a través de una red de televisión. Es de interés para la presente investigación el primero de ellos, a través del cual se realiza toda la gestión de la información y recursos multimedia que se transmiten.

Introducción

El subsistema de administración de PRIMICIA es una solución web que puede ser utilizada desde las estaciones de trabajo y dispositivos móviles, teniendo una conexión directa con su servidor de aplicaciones. En la actualidad, la mayor parte de los periodistas que se encargan de elaborar las pre-noticias se encuentran dispersos espacialmente, en constante movilidad y sin conexión a internet. Esto provoca que el mecanismo que se sigue para crear la misma se torne complejo y aumente el tiempo de manera considerable. La complejidad está dada por el proceso que se lleva a cabo desde la recepción de la noticia en el centro, el orden de prioridad que se le da a la misma, la asignación a un miembro del equipo de editores, el formato que se le debe dar, hasta su concepción final.

Teniendo en cuenta la descripción anterior se deriva que los módulos de redacción de noticias y gestión de medias del subsistema de administración de PRIMICIA presentan las siguientes limitaciones:

- La accesibilidad de los módulos está comprometida por no tener variantes que favorezcan que un número mayor de usuarios que intervienen en el proceso de creación de una noticia (periodistas) puedan ir estructurando la misma, independientemente del lugar, el momento en el que se encuentren y de las condiciones técnicas que posean (sin conexión al servidor).
- La portabilidad de los módulos está afectada ya que hoy no es posible transportar la aplicación, sin embargo, es una necesidad en el negocio la movilidad constante de la persona (periodista) que se encarga de crear las noticias.

A partir de la situación problemática descrita con anterioridad se identifica el siguiente **problema de investigación**: la deficiente portabilidad y accesibilidad de los módulos de redacción de noticias y gestión de medias del subsistema de administración de PRIMICIA, dificulta la inmediatez en el proceso de creación de las noticias.

Para enmarcar los límites del presente trabajo se define como **objeto de estudio**: los módulos de redacción de noticias y gestión de medias del subsistema de administración de PRIMICIA.

Definiéndose como **campo de acción**: aplicaciones móviles que garantizan la portabilidad y accesibilidad de funcionalidades asociadas a la creación de noticias.

Para dar solución al problema planteado se define como **objetivo general**: desarrollar una aplicación para dispositivos móviles, que garantice la portabilidad y accesibilidad de los módulos de redacción de noticias y gestión de medias del subsistema de administración de PRIMICIA.

Introducción

Con el propósito de guiar la investigación se formulan las siguientes **preguntas científicas**:

1. ¿Qué referentes teóricos sustentan el desarrollo de aplicaciones móviles que garantizan la creación de noticias?
2. ¿Qué características tienen los módulos de redacción de noticias y gestión de medias del subsistema de administración de PRIMICIA?
3. ¿Cómo estructurar el proceso de desarrollo de la aplicación para dispositivos móviles que garantice la gestión de las noticias y medias de la plataforma de televisión informativa PRIMICIA?
4. ¿La aplicación desarrollada garantiza la portabilidad y accesibilidad de los módulos de redacción de noticias y gestión de medias del subsistema de administración de PRIMICIA a través de dispositivos móviles?

Para darle solución al objetivo general se definen las siguientes **tareas de investigación**:

1. Establecimiento de los fundamentos teóricos que sustentan el desarrollo de aplicaciones móviles que garantizan la creación de noticias.
2. Caracterización de los módulos de redacción de noticias y gestión de medias del subsistema de administración de PRIMICIA.
3. Desarrollo de la aplicación para dispositivos móviles que garantice la gestión de noticias y medias que se publican a través de la plataforma de televisión informativa PRIMICIA.
4. Validar la contribución lograda a través de la introducción de la aplicación informática, en los grados de portabilidad y accesibilidad de los módulos de redacción de noticias y gestión de medias del subsistema de administración de PRIMICIA, empleando dispositivos móviles.

Con el objetivo de dar cumplimiento a las tareas propuestas, se utilizaron como **métodos científicos de investigación**:

Método Analítico – Sintético: Se aplica con el objetivo de determinar las generalidades y especificidades en el objeto de estudio y el campo de acción; así como en la fundamentación teórica y elaboración de la solución propuesta. Se sintetizan los aspectos más importantes de la investigación para extraer los conceptos fundamentales asociados al problema y las bases teóricas para cumplir con el objetivo propuesto.

Método de Modelación: Se representan los diagramas para una correcta estructura del sistema, lo cual permite un mejor entendimiento de las funcionalidades que debe tener la aplicación.

Histórico - lógico: Este método sirve de apoyo para el estudio crítico de los trabajos anteriores que constituyen referentes teórico-prácticos en el tratamiento del problema planteado y tomarlos como base de comparación con los resultados alcanzados.

Análisis documental: Este método se utiliza para la selección y recolección de la documentación asociada al tema abordado, con el objetivo de extraer la información necesaria para lograr los resultados esperados.

Como técnica de recopilación de la información se utilizó la **entrevista no estructurada** al personal del proyecto que se encarga de forma directa de la redacción de noticias y gestión de las medias, para obtener información asociada a la solución, sus características y cualidades.

El presente documento se estructura de la siguiente manera:

Capítulo 1: Fundamentación teórica: En este capítulo se abordan los conceptos y temas asociados a los procesos de redacción de noticias y gestión de medias del subsistema de administración de PRIMICIA, así como las soluciones existentes en este campo. También se analizan las tendencias, las tecnologías y las metodologías que actualmente son empleadas en el desarrollo de sistemas informáticos que garanticen la creación de noticias a través de dispositivos móviles.

Capítulo 2: Análisis del sistema: En este capítulo se describe detalladamente las características del sistema a desarrollar. Se realiza un modelo conceptual que describe de forma sintetizada los procesos de redacción de noticias y gestión de medias del subsistema de administración de PRIMICIA. Se exponen los requisitos funcionales y no funcionales de la aplicación y se presenta el modelo de casos de uso del sistema y las descripciones textuales de los casos de uso que forman parte del mismo.

Capítulo 3: Diseño del sistema: Se describe la arquitectura. Se exponen los diagramas asociados a la etapa de diseño referentes a: diagramas de clases de diseño y diagramas de arquitectura de datos, (diagrama de clases persistentes y el diagrama entidad – relación). Se explican los patrones de diseño empleados y concluye el capítulo con el diagrama de despliegue.

Capítulo 4: Implementación y pruebas del sistema: Se muestran los diagramas de componentes y se describen los estándares de codificación. Se detallan las pruebas realizadas para la comprobación del correcto funcionamiento del sistema construido. Se muestra la valoración de la contribución lograda con la introducción de la aplicación informática, en los grados de portabilidad y usabilidad de los módulos de redacción de noticias y gestión de medias del subsistema de administración de PRIMICIA, empleando dispositivos móviles.

Introducción

Para finalizar se presentan las conclusiones generales, recomendaciones, referencias bibliográficas y un conjunto de anexos para un mejor entendimiento de lo expuesto en la investigación.

Capítulo 1. Redacción de noticias y gestión de medias. Las tecnologías para su automatización

Introducción

En el presente capítulo se abordan una serie de conceptos asociados al dominio del problema para facilitar un mejor entendimiento del mismo y se describen los módulos de redacción de noticias y gestión de medias de la plataforma PRIMICIA. También se realiza un análisis de las plataformas que se utilizan para la redacción de noticias en dispositivos móviles. Por último se describe la metodología seleccionada, así como las principales tecnologías empleadas para el desarrollo de la aplicación.

1.1 Principales conceptos asociados al tema de investigación

Multimedia

Se utiliza para referirse a cualquier objeto o sistema que utiliza múltiples medios de expresión físicos o digitales para presentar o comunicar información. Los medios pueden ser variados, desde texto e imágenes, hasta animación, sonido, video, entre otros (ABC, 2007).

Componentes

Los componentes son recursos que se pueden incorporar en la redacción de las noticias. Tienen diversas propiedades que se pueden cambiar, configurando la presentación de dichos componentes en la noticia. Estos componentes se encuentran presentes en la noticia a través de texto, imagen y video (Montes de Oca, 2012).

Dispositivos móviles

Un dispositivo móvil se puede definir como un aparato de pequeño tamaño, con algunas capacidades de procesamiento, con conexión permanente o intermitente a una red, con memoria limitada, que ha sido diseñado específicamente para una función, pero que puede llevar a cabo otras funciones más generales.

Capítulo 1. Fundamentación Teórica

Aplicación móvil

Una aplicación es simplemente un programa informático creado para llevar a cabo o facilitar una tarea en un dispositivo informático. Es aquel software que puede ejecutarse en una plataforma móvil o una aplicación web que se diseñó para una plataforma móvil (MasterMagazine, 2010).

Red inalámbrica

Una red inalámbrica es una conexión establecida entre nodos sin necesidad de una conexión física (cables), sino estableciendo la comunicación por ondas electromagnéticas. La transmisión y recepción de los datos se realiza a través de puertos destinados a establecer la conexión entre las estaciones móviles (Bordallo, 2013).

1.2 Módulos de redacción de noticias y gestión de medias del subsistema de administración de PRIMICIA

El subsistema de administración de PRIMICIA, está compuesto por diferentes módulos que en su conjunto, posibilitan su correcto funcionamiento. A continuación se presenta una descripción de los módulos que son de interés para esta investigación: redacción de noticias y gestión de medias, que posibilita un mejor entendimiento de los procesos que forman parte de los mismos.

Módulo de Redacción de Noticias

El módulo de redacción de noticias está orientado a la redacción y edición de las informaciones que conjuntamente con los recursos multimedia conforman la noticia. Este módulo da la posibilidad de guardar un archivo persistente así como las configuraciones de la noticia. El archivo persistente se entiende como el xml que se crea con todos los datos alusivos a la noticia. Por su parte, las configuraciones de la noticia, se refieren al tamaño, posicionamiento y efectos de entrada y salida de los textos, imágenes y videos según sea el caso. Este módulo crea una manera factible para su edición y visualización conjuntamente con las funciones del subsistema de Transmisión, a través del archivo persistente anteriormente mencionado.

Las noticias están compuestas por pantallas y estas a su vez pueden tener cualquier combinación entre imagen, texto y video. Esto permite que el redactor cree la noticia de una forma más intuitiva según su criterio. De forma similar, se le pueden adicionar efectos de transición a cada pantalla, así como la duración exacta que deben tener cada una de ellas de forma independiente.

Capítulo 1. Fundamentación Teórica

Según (Montes de Oca, 2012), en este módulo se prioriza la flexibilidad de las propiedades y la estructura de los componentes de la noticia, lo cual posibilita un despliegue en entornos con disímiles características. Con esto se obtiene un software eficiente y adaptable a las necesidades del cliente.

Este módulo tiene una completa dependencia del que se encarga de gestionar las medias. Para redactar y publicar una noticia se hace necesario que al menos exista un audio almacenado y si se desea que la noticia sea más completa debe existir de igual forma imágenes y videos guardados.

Módulo de Gestión de Medias

El módulo de gestión de medias tiene como objetivo la manipulación de las medias (imagen, video y audio), que se utilizarán en la redacción de las noticias. En este módulo se almacena de forma organizada los archivos audiovisuales y se controla toda operación que se realice con estos. Para el almacenamiento de las medias se utiliza un servidor de ficheros y una referencia en la base de datos con criterios de búsquedas, así como varios datos pertenecientes a cada fichero que permite posteriormente localizarlas.

Para realizar cualquier operación con los recursos multimedia, se debe especificar previamente si se quiere adicionar un nuevo recurso o buscar uno específico de los almacenados en la base de datos. Los archivos con los que trabaja la aplicación son audio, imagen y video, los cuales tienen asociados varios datos, que se tienen en cuenta para adicionar o buscar un recurso en particular. Además, cuando se realiza una búsqueda entre las multimedia, se le posibilita al usuario modificar, eliminar o visualizar el recurso al cual accedió. En caso de querer copiar el archivo se debe conocer el lugar donde se almacena o contactar con el encargado de trabajar con los audiovisuales.

Estos módulos están limitados a ser utilizados desde los dispositivos que tengan conexión con la plataforma, ya que si no existe o se interrumpe la conexión al servidor de aplicaciones de PRIMICIA, se entorpece el proceso de redacción y gestión de las medias.

Se desea implementar una solución que permita pre-elaborar las noticias sin que exista una completa dependencia al servidor de aplicaciones. Una vez que se establezca la conexión se puedan subir estas noticias para su edición en la plataforma.

Capítulo 1. Fundamentación Teórica

1.3 Análisis de las posibles plataformas a utilizar para el desarrollo de aplicaciones en dispositivos móviles

En la actualidad son varios los sistemas operativos (SO) que han surgido en el mercado de los dispositivos móviles. Los que más se destacan son iOS de Apple, Windows Phone de Microsoft y Android de Google³. La salida del iPhone con el sistema operativo iOS de Apple ha marcado el desarrollo y evolución de los dispositivos móviles. Por su parte Android SO utilizado por Google para teléfonos inteligentes de última generación, tiene varias peculiaridades que lo diferencian de sus principales competidores. Luego Microsoft decidió crear un estilo propio de su SO móvil con su innovador Windows Phone y la interfaz Metro, la cual es sencilla de usar y muy intuitiva.

Para evitar errores y cumplir correctamente con las exigencias del cliente, se hace necesario llevar a cabo un estudio de los distintos sistemas operativos que se mencionaron con anterioridad. Las siguientes comparaciones permitieron encaminar la investigación y determinar cuál SO es el más recomendado para el desarrollo de la aplicación en los dispositivos móviles.

1.3.1 *Windows Phone*

Windows Phone es la apuesta planteada por Microsoft en el mundo de la tecnología móvil comenzada en el año 2010. Permite una correcta integración con equipos de su misma arquitectura, dígame Windows XP, Windows 7 y Windows 8. Posee una buena autonomía que lleva a cabo un plan de energía, permitiendo entrar en un modo de menos consumo de energía cuando la batería se está agotando.

Una de las principales desventajas se encuentra en el número de aplicaciones disponibles y la frecuencia con las que estas se actualizan. Las personalizaciones del sistema se encuentran restringidas a la forma en que ha sido desarrollada por su fabricante, haciendo posible que solo se puedan hacer modificaciones en algunas de ellas y suelen requerir un hardware bien potente. El código fuente de este SO es privativo por lo que no está disponible para el desarrollo y mejora del mismo.

³ Google Inc. es una empresa multinacional estadounidense especializada en productos y servicios relacionados con Internet, software, dispositivos electrónicos y otras tecnologías.

Capítulo 1. Fundamentación Teórica

1.3.2 iOS

iOS es el sistema operativo usado por Apple para sus móviles, creado en el año 2007. Este SO mantiene un estricto control y mantenimiento sobre su software y hardware. El uso de las aplicaciones en estos dispositivos es sobresaliente, ya que solo los dispositivos de Apple pueden contar con este sistema operativo. Esto permite de que Apple tenga una correcta cohesión en sus productos, ya que siempre sabe sobre qué hardware va a correr su SO. Entre las principales desventajas se encuentra que es un sistema “cerrado”, esto significa que al igual que Windows Phone su código no es modificable para los desarrolladores de aplicaciones móviles. Estas aplicaciones solo pueden ser obtenidas desde su tienda, para las cuales se tiene un extremo y detallado control. Debido a la política que mantiene Apple el intercambio de estos dispositivos con otros fabricantes es casi imposible.

1.3.3 Android

Android es el sistema operativo de Google, siendo el más utilizado por las personas en el mundo según el estudio que llevó la empresa Gartner Group (Tomás, 2013). Una de sus ventajas es la reutilización de su código, es abierto para cualquier persona, con conocimiento pertinente, que desee realizar cambios en el software de su dispositivo móvil. La comunidad de desarrolladores es extensa y continuamente crea soluciones para Android.

Un estudio llevado por la empresa Gartner Group, muestra la evolución del mercado de los sistemas operativos para móviles según el número de terminales vendidos. Cabe enfatizar el destacado ascenso de la plataforma Android, que en cinco años ha alcanzado una cuota de mercado en torno al 85% (Tomás, 2013). Ver Figura 1.

Capítulo 1. Fundamentación Teórica

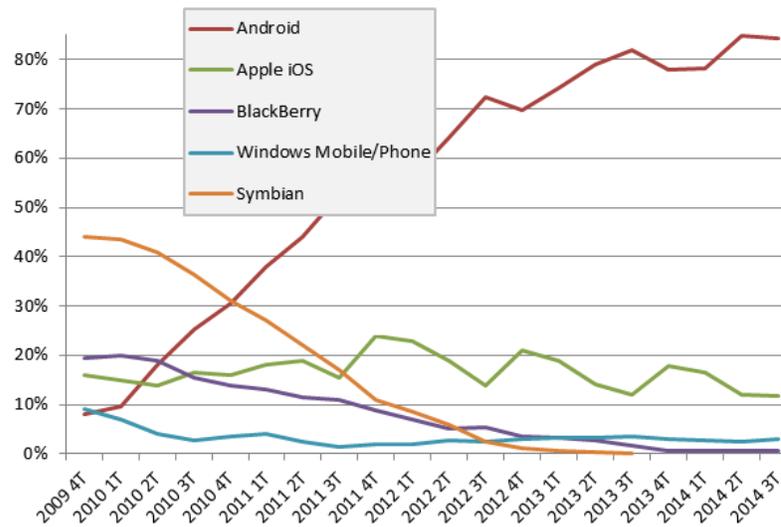


Figura 1 Porcentajes de teléfonos inteligentes vendidos según su SO (Tomás, 2013)

Luego de un análisis de los distintos sistemas a emplear se decidió escoger Android por las siguientes características:

1. Es un sistema operativo de gran demanda por lo que los desarrolladores son conscientes de la gran cantidad de usuarios que crean aplicaciones. Esto propicia que el sistema tenga una gran fortaleza y sea estable.
2. Una de las características de especial relevancia es su presencia en los dispositivos de gama media – baja, mientras que los dispositivos iOS y Windows Phone por sus características del sistema deben pertenecer a la gama alta.
3. Su sistema está basado en el kernel de Linux por lo que lo hace un software libre, permitiendo que los desarrolladores realicen aplicaciones y puedan probarlos en ellos. Esto ayuda a que muchos usuarios lo elijan para su uso en sus dispositivos móviles.

1.4 Sistema operativo Android

Android es un sistema operativo para dispositivos móviles basado en un kernel Linux (versión 2.6). El proyecto Android está integrado por Google y un conglomerado de otras empresas tecnológicas agrupadas bajo el nombre de *Open Handset Alliance*. El objetivo principal de esta alianza empresarial (que incluye a fabricantes de dispositivos y operadores, con firmas tan relevantes como Samsung, LG, Intel, Texas Instruments, entre otras) es el desarrollo de estándares abiertos para la telefonía móvil como medida para

Capítulo 1. Fundamentación Teórica

incentivar su desarrollo y para mejorar la experiencia del usuario. Con Android se busca reunir en una misma plataforma todos los elementos necesarios que permitan al desarrollador controlar y aprovechar al máximo cualquier funcionalidad ofrecida por un dispositivo móvil (llamadas, mensajes de texto, cámara, agenda de contactos, conexión Wi-Fi, Bluetooth, aplicaciones ofimáticas, videojuegos, etc.), así como poder crear aplicaciones que sean verdaderamente portables, reutilizables y de rápido desarrollo (Developers, 2014).

1.4.1 Características de Android

Android es una plataforma neutral en cuanto al desarrollo de aplicaciones, el programador tiene las mismas posibilidades para crear una aplicación que Google o que cualquier empresa perteneciente al *Open Handset Alliance*. A continuación se destacan las características más significativas de Android:

- Amplia variedad de diseños (VGA, librerías de gráficos 2D y 3D).
- Almacenamiento de datos en Base de datos SQLite.
- Conectividad (GSM/EDGE, CDMA, EV-DO, UMTS, Bluetooth y Wi-Fi).
- Mensajería (SMS y MMS).
- Máquina virtual de Java.
- Las aplicaciones escritas en Java pueden ser compiladas y ejecutadas en la máquina virtual de Dalvik, la cual es una especializada máquina virtual diseñada para uso en dispositivos móviles.
- Soporte de formatos (MPEG-4, H.264, MP3, AAC, OGG, AMR, JPEG, PNG, GIF).
- Soporte para hardware adicional (cámaras de vídeo, sensores, pantallas táctiles, GPS, acelerómetros).
- Entorno de desarrollo (emulador, herramientas de depuración, perfiles de memoria y funcionamiento, plugin para Eclipse IDE y Android Studio) (Developers, 2014).

1.4.2 Componentes de una aplicación Android

Todas las aplicaciones en Android pueden descomponerse en cuatro tipos de bloques o componentes principales. Cada aplicación será una combinación de uno o más de estos componentes, que deberán ser declarados de forma explícita en un fichero con formato XML denominado “AndroidManifest.xml”, junto a otros datos asociados como valores globales, clases que implementa, datos que puede manejar, permisos, entre otros. Este fichero es básico en cualquier aplicación en Android y permite al sistema desplegar y ejecutar correctamente la aplicación.

Capítulo 1. Fundamentación Teórica

A continuación se exponen los cuatros tipos de componentes de una aplicación Android (Aranaz, 2009):

1. Activity

Un componente Activity refleja una determinada actividad llevada a cabo por una aplicación y que lleva asociada típicamente una ventana o interfaz de usuario. Es importante señalar que no contempla únicamente el aspecto gráfico, sino que éste forma parte del componente Activity a través de vistas representadas por clases como View y sus derivadas. Este componente se implementa mediante la clase de mismo nombre Activity. Muy vinculado a este componente se encuentran los Intent, una interesante novedad introducida por Android. Un Intent consiste básicamente en la voluntad de realizar alguna acción, generalmente asociada a unos datos. Lanzando un Intent, una aplicación puede delegar el trabajo en otra, de forma que el sistema se encarga de buscar qué aplicación entre las instaladas es la que puede llevar a cabo la acción solicitada. Por ejemplo, abrir una URL en algún navegador web, o escribir un correo electrónico desde algún cliente de correo.

2. Broadcast Intent Receiver

Un componente Broadcast Intent Receiver se utiliza para lanzar alguna ejecución dentro de la aplicación actual cuando un determinado evento se produzca (generalmente, abrir un componente Activity). Por ejemplo, una llamada entrante o un SMS recibido. No tiene interfaz de usuario asociada, pero puede utilizar la API de Notification Manager, para avisar al usuario del evento producido a través de la barra de estado del dispositivo móvil. Este componente se implementa a través de una clase de nombre BroadcastReceiver.

3. Service

Un componente Service representa una aplicación ejecutada sin interfaz de usuario, y que generalmente tiene lugar en segundo plano mientras otras aplicaciones (éstas con interfaz) son las que están activas en la pantalla del dispositivo.

4. Content Provider

Con el componente Content Provider, cualquier aplicación en Android puede almacenar datos en un fichero, en una base de datos SQLite o en cualquier otro formato que considere. Además, estos datos pueden ser compartidos entre distintas aplicaciones. Una clase que implemente el componente Content Provider contendrá una serie de métodos que permite almacenar, recuperar, actualizar y compartir los datos de una aplicación. Existe una colección de clases para distintos tipos de gestión de datos en el paquete

Capítulo 1. Fundamentación Teórica

android.provider. Además, cualquier formato adicional que se quiera implementar deberá pertenecer a este paquete y seguir sus estándares de funcionamiento.

1.4.3 Ventajas y desventajas de Android

Ventajas

1. Puede instalarse prácticamente en todo tipo de dispositivos, sean móviles, portátiles e incluso en equipos electrodomésticos.
2. Es libre con licencia Apache y código abierto para que un desarrollador no solo pueda modificar su código sino también mejorarlo.
3. Puede adaptarse a la perfección a todo tipo de necesidades.
4. Garantiza que, en caso de haber un error, sea detectado y reparado con mayor rapidez sin depender de nadie para pedir autorización a su cambio. (Bello, 2011).

Desventajas

1. Consumo de la batería: debido a la posibilidad de tener varias aplicaciones corriendo a la vez en el sistema esto implica un gasto incremental en el consumo de la batería.
2. Fragmentación Total: posee algunos problemas de compatibilidad con aplicaciones no creadas por su compañía, teniendo que adaptarlos a cada teléfono móvil y dispositivo que utilice esta tecnología (Bello, 2011).

1.5 Sistemas de producción de noticias para dispositivos móviles

AP ENPS móvil

Es una solución que permite la redacción, visualización y edición de noticias, supervisar tareas, ver escaletas y las historias de edición, buscar contactos, cargar medias (imagen, video) y la comunicación entre usuarios a través de mensajes. Este sistema permite la producción de noticias multiplataforma en cualquier lugar a través de teléfonos inteligentes, tabletas, ordenadores portátiles o de sobremesa (AP ENPS, 2012).

Dalet On-The-Go

Esta aplicación está diseñada para los periodistas y permite establecer una conexión con su sala de redacción, donde se ejecuta el Dalet Noticias Suite. Esta solución ofrece 3 funciones principales:

Capítulo 1. Fundamentación Teórica

1. Read-In: Permite a los usuarios navegar y buscar con palabras clave y obtener actualizaciones sobre las últimas noticias registradas, incluyendo artículos incorporados dentro de estas. Los usuarios sólo pueden hacer clic para acceder a audio o video disponible.
2. My Assignments: Permite a los periodistas comprobar sus asignaciones individuales en un día determinado, así como detalles relacionados con los metadatos y medios de comunicación asociados a esas tareas.
3. Contribution: Permite a los usuarios subir historias, vídeos, imágenes y audios directamente desde su dispositivo a su sitio de origen Dalet.

iNews

El sistema informático iNews es un centro dinámico de creación y distribución de noticias; permite montar y emitir noticias de última hora, unificar las operaciones de redacción y atraer audiencia a través de una amplia gama de canales: televisión, web, dispositivos móviles y medios sociales (INEWS, 2013).

Este sistema cuenta con varias funcionalidades que se presentan a continuación:

1. Flujo de trabajo de noticias integrado: Controla todos los aspectos de la creación y distribución de noticias a través de herramientas integradas para la edición e información desde dispositivos móviles, gráficos para noticias, administración de pautas, control de emisión y distribución multicanal.
2. Producción de noticias basadas en Internet: Crea, visualiza y edita piezas y pautas desde cualquier lugar a través de la interfaz de usuario basada en el navegador. Simplifica la administración del sistema a nivel del servidor con arquitectura de cliente ligero.
3. Colaboración: Comparte contenido y colabora en las piezas desde cualquier lugar buscando, visualizando y editando contenido en cualquier sistema iNews dentro de un grupo o red de una estación a través de la interfaz basada en el navegador o en una aplicación cliente local.
4. Administración simple: Con la integración de LDAP, se puede vincular iNEWS con la seguridad de red informática para un proceso de inicio de sesión consolidado. La posibilidad de obtener acceso a la base de datos en línea, garantiza las mejores prácticas de mantenimiento y la virtualización reduce el tamaño de hardware de iNEWS.
5. Flujos de trabajo móviles:
 - Crea, explora y edita guiones de iNews desde cualquier lugar usando un dispositivo móvil.

Capítulo 1. Fundamentación Teórica

- Revisa y aprueba contenido de noticias.
- Obtiene acceso a contenido de noticias y pautas en tiempo real en cualquier sistema iNEWS dentro de una red o grupo de estaciones.

Conclusiones de los sistemas estudiados

El estudio de los sistemas anteriormente analizados permitió identificar un conjunto de atributos y funcionalidades básicas que tienen estrecha relación con el proceso general de almacenamiento y sincronización de archivos, sin embargo no se ajustan al negocio en particular que se desea resolver. La mayoría de estos sistemas son propietarios, por lo que la adquisición de los mismos representaría un alto valor monetario. Cuba se encuentra en una migración a Software Libre y especialmente la UCI aumenta sus esfuerzos en lograr una soberanía tecnológica con el uso de tecnologías libres y este tipo de sistemas no garantiza el cumplimiento de esta política. No obstante, las soluciones que existen permiten obtener una visión general asociada a las diversas formas que dan solución a problemáticas que son del interés de la presente investigación, como es el caso de los protocolos que utilizan para la transferencia de datos, la manera de almacenar la información, las tecnologías que utilizan para integrar los textos, las imágenes y el audio.

1.6 Tendencias y tecnologías actuales

En este apartado se especifican las tecnologías que permitirán estructurar y desarrollar el sistema que se desea construir. Se describe la metodología, lenguaje de modelado, herramientas CASE, lenguajes de programación y sistema gestor de base datos, teniendo en cuenta sus características, funcionalidades y ventajas.

1.6.1 Metodología de Desarrollo OpenUP⁴

Es una metodología que aplica un enfoque iterativo e incremental dentro de un ciclo de vida estructurado. Utiliza una filosofía ágil que se orienta a la naturaleza de colaboración para el desarrollo de software, basada en RUP (Rational Unified Process por sus siglas en inglés). Contiene el conjunto mínimo de prácticas que ayudan a un equipo de desarrollo de software a realizar un producto de alta calidad, de una forma eficiente.

⁴ OpenUP: Proceso Unificado Abierto (por sus siglas en inglés Open Unified Process).

Capítulo 1. Fundamentación Teórica

Este tipo de metodología es un proceso mínimo y suficiente, lo que significa que sólo el contenido fundamental y necesario es incluido. Por lo tanto no provee lineamientos para todos los elementos que se manejan en un proyecto pero tiene los componentes básicos que pueden servir de base a procesos específicos. La mayoría de los elementos de esta metodología están declarados para fomentar el intercambio de información entre los equipos de desarrollo y mantener un entendimiento compartido del proyecto, sus objetivos, alcance y avances.

Esta metodología se encuentra dirigida por cuatros principios básicos que se muestran a continuación:

- Colaboración para unificar intereses y compartir conocimientos.
- Equilibrio de prioridades competentes a maximizar el valor de los involucrados con el resultado del proyecto.
- Enfoque en la articulación de la arquitectura.
- Desarrollo continuo para obtener retroalimentación y realizar las mejoras respectivas.

Open UP es un proceso completo, flexible y corto, fomenta el uso de técnicas ágiles y principios, mientras que tiene un ciclo de vida estructurado y probado que hace referencia en la continua entrega de software de calidad. Es apropiada para proyectos pequeños y de bajos recursos. Tiene un enfoque centrado al cliente con iteraciones cortas. Evita la elaboración de documentación, diagramas e iteraciones innecesarias.

1.6.2 Lenguaje Unificado de Modelado (UML 2.0)

El Lenguaje Unificado de Modelado (UML), es un lenguaje de modelado visual que se usa para especificar, visualizar, construir y documentar artefactos de un sistema de software. Captura decisiones y conocimiento sobre los sistemas que se deben construir. Está pensado para usarse con todos los métodos de desarrollo, etapas del ciclo de vida, dominios de aplicación y medios. La especificación de UML no define un proceso estándar pero está pensado para ser útil en un proceso de desarrollo iterativo. Pretende dar apoyo a la mayoría de los procesos de desarrollo orientados a objetos (Fowler, 2007).

1.6.3 Herramienta CASE Visual Paradigm for UML 8.0

Visual Paradigm es una herramienta profesional para el modelado con UML. Soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientado a objetos, construcción, pruebas y despliegue. Proporciona un diseño centrado en casos de uso y enfocado al negocio que genera un software

Capítulo 1. Fundamentación Teórica

de mayor calidad. Es capaz de generar código e ingeniería inversa para varios lenguajes de programación. Visual Paradigm for UML ha sido desarrollada para todos los sistemas operativos compatibles con Java, incluyendo Windows, Linux y Mac OS X (Visual Paradigm Group, 2006).

Es una herramienta considerada muy completa, fácil de usar, con soporte multiplataforma y de probada utilidad para el analista, proporciona excelentes facilidades de interoperabilidad con otras aplicaciones. Posee un conjunto de ayudas para el desarrollo de programas informáticos, desde la planificación, análisis y diseño, hasta la generación del código fuente y la documentación. Tiene licencia dual, gratuita y comercial. Soporta las últimas versiones de UML y la Notación y Modelado de Procesos de Negocios, a través de la utilización de un enfoque Orientado a Objetos con abundante documentación. En adición al soporte de Modelado UML, Visual Paradigm provee un generador de mapeo de objetos-relacionales para los lenguajes de programación como Java, .NET y PHP. Una de sus principales ventajas es que incorpora el soporte para trabajo en equipo, lo que permite que varios desarrolladores trabajen a la vez en el mismo diagrama y vean en tiempo real los cambios hechos por sus compañeros (Visual Paradigm Group, 2006).

1.6.4 Lenguaje de Programación Java

Java es un lenguaje de programación orientado a objetos. Este lenguaje toma mucha de su sintaxis de lenguaje de programación C y C++, aunque presenta un modelo de objetos más simple y elimina herramientas de bajo nivel, que generalmente producen muchos errores, como la manipulación directa de punteros o memoria. Google utiliza Java por defecto para Android, se pueden utilizar otros lenguajes como C, C# o C++, pero el oficial y el soportado de forma nativa por Android es Java. Una de las características fundamentales es que programas escritos en el lenguaje Java pueden ejecutarse igualmente en cualquier tipo de hardware. También resuelve en gran medida el problema de las fugas de memoria debido a que posee un recolector de basura, el cual se encarga de eliminar los objetos cuyas instancias ya no existan, liberando así el espacio en memoria ocupado por estos.

1.6.5 Entorno de Desarrollo Integrado Eclipse

Eclipse, fue creado en origen por la compañía International Business Machine (IBM), aunque posteriormente fue liberado por Eclipse Foundation en el año 2006. Es soportado por sistemas operativos como GNU/Linux, Mac OS X, Windows XP, Vista, Windows 7, Windows 8, por lo que es multiplataforma y software libre (Foundation, 2014).

Capítulo 1. Fundamentación Teórica

Es un entorno de desarrollo integrado, de código abierto y multiplataforma. Es una potente y completa plataforma de programación, desarrollo y compilación de elementos tan variados como sitios web, programas en C++ o aplicaciones Java. En él se pueden encontrar todas las herramientas y funciones necesarias para el trabajo, recogidas además en una atractiva interfaz que lo hace fácil y agradable de usar. Eclipse es ampliamente conocido por las características que ofrece a los desarrolladores. Eclipse cuenta con el plugin de Herramientas de Desarrollo de Android el SDK que permite desarrollar libremente aplicaciones para el sistema operativo Android. Es un IDE muy competente, debido en parte, a su tiempo de uso y la preferencia de los desarrolladores por esta plataforma, y en adición, las versiones que se liberan son cada vez más completas.

1.6.6 Paquete de Desarrollo de Software (Software Development Kit, SDK) de Android

El SDK es el paquete de desarrollo necesario para programar e implementar todo tipo de aplicaciones para Android. Este paquete o kit de desarrollo incluye las APIs y herramientas necesarias para desarrollar las aplicaciones, y probar el código, utilizando Java como lenguaje de programación.

1.6.7 Herramienta de Desarrollo de Android (Android Development Tools, ADT)

El ADT es un plug-in para el IDE Eclipse que está diseñado para darle un poderoso entorno integrado en el cual construir aplicaciones para Android. ADT amplía las capacidades de Eclipse para permitirle configurar rápidamente nuevos proyectos para Android, crear una interfaz de usuario de la aplicación, agregar paquetes basados en la API de Android, y exportar en .apk con el fin de distribuir la aplicación. (Developers, 2014)

1.6.8 Gestor de Base de Datos SQLite

Sistema completo de bases de datos que soporta múltiples usabilidades, con la característica de ser ágil pero robusto. Es un sistema gestor de bases de datos relacional compatible con Atomicidad, Consistencia, Aislamiento y Durabilidad ACID (por sus siglas en inglés Atomicity, Consistency, Isolation and Durability), comprendida en una biblioteca relativamente pequeña. Es una librería escrita en C que implementa un motor de bases de datos para SQL92.

Este gestor es un sistema que soporta tablas, índices y vistas que no necesita de un servidor para su utilización. Es capaz de escribir y leer directamente sobre ficheros que se encuentran en el disco duro. Es multiplataforma e imparcialmente se puede utilizar archivos en sistemas de 32 y 64 bits.

Capítulo 1. Fundamentación Teórica

La base de datos se almacena en un único fichero a diferencia de otros SGBD que hacen uso de varios archivos. SQLite emplea registros de tamaño variable de forma tal que se utiliza el espacio en disco que es realmente necesario en cada momento.

Algunas de las ventajas que este gestor presenta:

1. Tamaño: Posee una pequeña memoria y una única biblioteca necesaria para acceder a bases de datos, lo que lo hace ideal para aplicaciones de bases de datos incorporadas.
2. Rendimiento de base de datos: SQLite realiza operaciones de manera eficiente y es más rápido que MySQL y PostgreSQL .
3. Portabilidad: Se ejecuta en muchas plataformas y sus bases de datos pueden ser fácilmente portadas sin ninguna configuración o administración.
4. SQL: Implementa un gran subconjunto de la ANSI⁵ – 92 SQL estándar, incluyendo sub-consultas, generación de usuarios y vistas.
5. Costo: SQLite es de dominio público y por tanto, es libre de utilizar para cualquier propósito sin costo y se puede redistribuir libremente.

Conclusiones parciales

A partir de los aspectos descritos en el presente capítulo se concluye que:

- La descripción de los módulos de redacción de noticias y gestión de medias del subsistema de administración de PRIMICIA permitió tener una visión detallada de las funcionalidades que tiene, sin embargo su estructura visual no se ajusta a los estándares a seguir para el desarrollo de una aplicación móvil, por lo que se hizo necesario reevaluar cómo presentar la información.
- Las soluciones estudiadas en el campo que se investiga no se ajustan a las necesidades expuestas en la investigación ya que las funcionalidades que incluyen responden al negocio para el que está diseñado cada uno de los sistemas, además de ser todas ellas software privativo y la dificultad de modificar el código fuente no permite que éstos se puedan integrar con PRIMICIA.
- Para crear la aplicación que permite la redacción de noticias y gestión de las medias se identificó como plataforma de desarrollo Android ya que presenta amplia documentación, está presente en una amplia gama de dispositivos y tiene una demanda alta entre los usuarios.

⁵ ANSI: Instituto Nacional Estadounidense de Estándares (por sus siglas en inglés: American National Standards Institute)

Capítulo 1. Fundamentación Teórica

- Las tecnologías escogidas para el desarrollo de la aplicación, fueron determinadas por las características de éstas. Fueron escogidos el marco de trabajo Eclipse con el SDK de Android, gestor de base de datos SQLite, Visual Paradigm como herramienta para la generación de artefactos y lenguaje de programación Java, lo cual posibilitará la obtención de una solución bien documentada, ofreciendo facilidad para el desarrollo de futuras versiones de la aplicación.

Capítulo 2. Análisis del sistema

Introducción

En el presente capítulo se realiza un estudio del sistema que se desea desarrollar, se representa un modelo conceptual para describir brevemente cómo se lleva a cabo el proceso de redacción de noticias en el subsistema de administración de PRIMICIA. Se precisan las características fundamentales y cualidades del sistema a desarrollar a través de los requisitos funcionales y no funcionales. Por último se realiza el diagrama de casos de uso del sistema así como la descripción de sus actores y casos de uso.

2.1 Modelo Conceptual

El modelo conceptual es una herramienta de comunicación fundamental. Permite que los desarrolladores piensen formalmente en el problema y validen su comprensión del mismo. Mediante este modelo se establece además un vocabulario propio del problema; y junto a los requerimientos constituye la entrada más importante para el diseño. Es importante aclarar que por lo general se demandan varias iteraciones para obtener un buen modelo.

El modelo conceptual, como el modelo de dominio es una representación visual estática que se realiza con el objetivo de lograr mejor comprensión del contexto para la obtención de requisitos del sistema. Básicamente, de manera visual no es más que un diagrama de clase UML que modela los conceptos básicos asociados al dominio del problema, sus propiedades más importantes y las relaciones que resultan imprescindibles para contextualizar dichos conceptos, a fin de poder brindar elementos que permitan la identificación de los requisitos del sistema. Además, es válido para cualquier metodología de desarrollo de software. (Larman, 1999). En la Figura 2 se muestra el modelo conceptual.

Capítulo 3. Diseño del Sistema

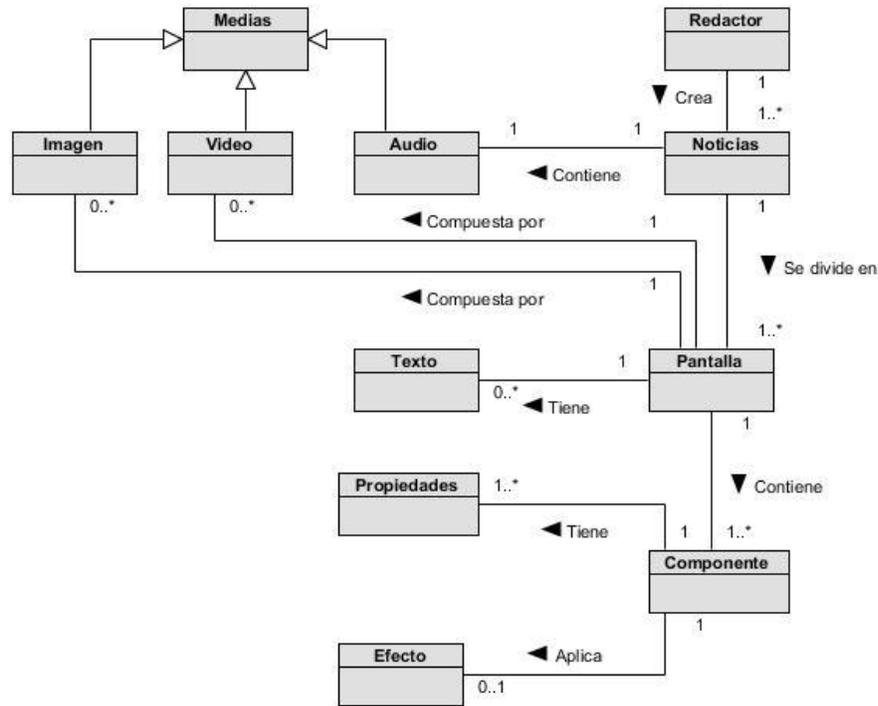


Figura 2 Modelo conceptual

Tabla 1 Descripción de los conceptos del Modelo Conceptual

Conceptos	Descripción
Redactor	Se encarga de redactar cada una de las noticias que son transmitidas.
Pantalla	Elemento que está asociado a una noticia y que a su vez puede estar compuesto por varios componentes.
Componente	Elemento que se compone por imagen, video, texto y se asocia a una pantalla en la noticia.
Efecto	Después de creado un componente se puede hacer uso de este elemento, el cual permite asociarle al mismo, efectos visuales de entrada y salida.
Propiedades	Proceso asociado a un componente, encargado de redimensionar, eliminar y mover el mismo.

Texto	Contenido en forma de párrafo asociado a una pantalla.
Medias	Archivo multimedia que puede ser de tres tipos: imagen, video y audio.

2.2 Requisitos de Software

La ingeniería de requisitos ayuda a los ingenieros a entender el problema en cuya solución trabajar. Incluye el conjunto de tareas que conducen a comprender cuál será el impacto del software sobre el negocio, qué es lo que el cliente quiere y cómo interactuarán los usuarios finales sobre el sistema (Pressman, 2008).

2.2.1 Especificación de Requisitos Funcionales

Los requisitos funcionales son condiciones o capacidades que el sistema debe cumplir, suficientemente buenas como para llegar a un acuerdo entre los clientes (incluyendo usuarios) sobre qué debe y qué no debe hacer el sistema. (Jacobson, y otros, 2000). A continuación se listan los requisitos definidos para la aplicación.

- RF1.** Autenticar usuario.
- RF2.** Adicionar noticia.
- RF3.** Modificar noticia.
- RF4.** Buscar noticia por nombre.
- RF5.** Eliminar noticia.
- RF6.** Listar noticias.
- RF7.** Visualizar noticia.
- RF8.** Adicionar pantalla a la noticia.
- RF9.** Eliminar pantalla de noticia.
- RF10.** Adicionar tiempo de duración de la pantalla.
- RF11.** Adicionar efecto de transición de la pantalla.
- RF12.** Guardar pantalla.
- RF13.** Adicionar imagen.
- RF14.** Modificar imagen.
- RF15.** Eliminar imagen.

- RF16.** Buscar imagen por nombre.
- RF17.** Visualizar imagen.
- RF18.** Adicionar video.
- RF19.** Modificar video.
- RF20.** Eliminar video.
- RF21.** Buscar video por nombre
- RF22.** Visualizar video
- RF23.** Asociar video a la pantalla
- RF24.** Asociar imagen a la pantalla.
- RF25.** Adicionar texto.
- RF26.** Modificar texto.
- RF27.** Eliminar texto.
- RF28.** Adicionar audio.
- RF29.** Modificar datos de audio.
- RF30.** Eliminar audio.
- RF31.** Reproducir audio.
- RF32.** Buscar audio por nombre.
- RF33.** Asociar audio a la noticia.
- RF34.** Sincronizar noticia.
- RF35.** Adicionar efecto de transición a la imagen.
- RF36.** Adicionar efecto de transición al texto.
- RF37.** Reproducir video de la pantalla.
- RF38.** Mover texto de la pantalla.
- RF39.** Redimensionar texto de la pantalla.
- RF40.** Eliminar texto de la pantalla.
- RF41.** Modificar texto de la pantalla.
- RF42.** Mover imagen de la pantalla.
- RF43.** Redimensionar imagen de la pantalla.
- RF44.** Eliminar imagen de la pantalla.
- RF45.** Mover video de la pantalla.

RF46. Redimensionar video de la pantalla.

RF47. Eliminar video de la pantalla.

2.2.2 Especificación de Requisitos no Funcionales

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable. En muchos casos los requerimientos no funcionales son fundamentales en el éxito del producto. Están vinculados a requerimientos funcionales, es decir, una vez que se conozca lo que el sistema debe hacer se puede determinar cómo ha de comportarse, qué cualidades debe tener o cuán rápido o grande debe ser. Los requerimientos no funcionales también añaden funcionalidad al producto, pues hacen que un producto sea fácil de usar, seguro, o demanda cierta cantidad de procesamiento (Pressman, 2008).

RNF 1. Apariencia o interfaz externa: La interfaz incluirá mecanismos para recuperarse ante errores, utilizando para ello mensajes de comprobación para la ejecución de determinadas funcionalidades. Se utilizarán términos obtenidos de la experiencia de las personas que más utilizarán el sistema y las interfaces se diseñarán de manera uniforme, para lograr un rápido aprendizaje de los usuarios. Los colores serán convenientemente utilizados dada la funcionalidad y objeto del sistema (predomina el azul), logrando una vista agradable a los usuarios y correspondiéndose con el logotipo de PRIMICIA.

RNF 2. Usabilidad: La aplicación debe tener características tales, que las personas que interactúen con ellas no tengan que ser expertos en informática ni poseer conocimientos avanzados en el tema para su explotación. La interfaz debe permitir la utilización de la aplicación por cualquier usuario con conocimientos básicos de trabajo en dispositivos móviles. El sistema será implementado siguiendo la regla de los tres (3) clic, que garantiza acceder a la información pertinente sin superar esa cifra al pulsar sobre un anclaje.

RNF 3. Confiabilidad: La utilización de la información en el dispositivo ya sea creándola o editándola en la base de datos local, deben corresponderse sin ninguna alteración con los elementos existentes en la base de datos de PRIMICIA.

Capítulo 3. Diseño del Sistema

RNF 4. Requisitos del Software:

- Se debe disponer en el dispositivo móvil de una versión superior a 2.2 del Sistema Operativo Android.

RNF 5. Requisitos del Hardware:

- El dispositivo debe soportar tecnología Wi-Fi 802.11 b/g/n.
- Se requiere un microprocesador con una velocidad de 650 MHz o superior.
- Se requiere un mínimo de 512 Mb de memoria RAM.
- Dispositivo con una resolución igual o superior a 320 x 480 píxeles, para una correcta visualización de la aplicación.

2.3 Modelación del Sistema

El diagrama de casos de uso del sistema contiene actores, casos de uso del sistema y sus relaciones; describe lo que el sistema debería hacer por sus usuarios y bajo qué restricciones. Permite que los desarrolladores y clientes lleguen a un acuerdo sobre los requerimientos, y proporciona la entrada fundamental para el análisis, diseño y las pruebas. Documenta el comportamiento de un software desde el punto de vista del usuario. Por tanto los casos de uso determinan los requisitos funcionales del sistema, los cuales representan las funcionalidades que un sistema puede ejecutar (Wesley, 2004).

2.3.1 Actores del sistema

Un actor no es más que un usuario o elemento externo que interactúa con la herramienta del sistema. No necesariamente debe ser una persona, también puede ser un sistema informático. Los actores utilizan los casos de usos para desarrollar partes del trabajo que poseen valor para el negocio.

Se identificó el siguiente actor:

Tabla 2 Actores del sistema

Actores	Descripción
Redactor	Usuario del teléfono móvil que posee los privilegios de almacenar en su dispositivo las noticias que determine.

Capítulo 3. Diseño del Sistema

2.3.1 Casos de uso del sistema

Los casos de uso son fragmentos de funcionalidad que el sistema ofrece para aportar un resultado de valor para los actores. De manera más precisa, un caso de uso especifica una secuencia de acciones que el sistema debe llevar a cabo interactuando con sus actores, incluyendo alternativas dentro de una secuencia. A continuación se identifican los casos de uso que hacen referencia a los requisitos funcionales que requiere el sistema a construir.

Tabla 3 Casos de uso del sistema

Referencia a requisitos	Nombre del Caso de Uso	Prioridad	Descripción
RF1	Autenticar usuario	Crítico	Constituye un elemento de seguridad dentro del sistema, esta funcionalidad permite a los usuarios autenticarse para poder subir las noticias al sistema.
RF2 RF3 RF4 RF5 RF6 RF7	Gestionar noticia	Crítico	Incluye las operaciones de insertar, eliminar, buscar, modificar y listar noticias.
RF8 RF9	Administrar pantalla	Crítico	Permite realizar las operaciones de insertar y eliminar las pantallas en la aplicación.
RF10 RF11 RF12	Adicionar propiedades a la pantalla	Secundario	Permite adicionar una duración (tiempo en segundos) a la pantalla, ponerle un efecto de entrada y de salida y permite guardar la pantalla.
RF13 RF14 RF15 RF16 RF17	Gestionar imagen	Crítico	Permite realizar las operaciones de insertar, modificar, buscar, visualizar y eliminar las imágenes en la aplicación.
RF18 RF19 RF20 RF21 RF22	Gestionar video	Crítico	Permite realizar las operaciones de insertar, modificar, buscar, visualizar y eliminar los videos en la aplicación.
RF25 RF26 RF27	Gestionar texto	Crítico	Permite realizar las operaciones de insertar, modificar y eliminar los textos en la aplicación.
RF28 RF29 RF30 RF31 RF32	Gestionar audio	Crítico	Permite realizar las operaciones de insertar, modificar y eliminar los audios en la aplicación.

Capítulo 3. Diseño del Sistema

RF33		Sincronizar noticia	Crítico	Permite conectar la aplicación crear un xml con los datos de la noticia y subirla al sistema.
RF35	RF 36	Adicionar efecto a las multimedia	Secundario	Permite realizar la adición de los efectos y características de las noticias.
RF37		Reproducir video	Secundario	Permite reproducir un video que esté asociado a una pantalla.
RF23 RF33 RF39 RF41 RF43 RF45 RF47	RF24 RF38 RF40 RF42 RF44 RF46	Manipular multimedia	Crítico	Permite realizar la adición de imagen y video en la pantalla, además de mover, redimensionar y eliminar las multimedia de la pantalla.

A continuación se presenta en la Figura 3 el Diagrama de casos de uso del sistema.

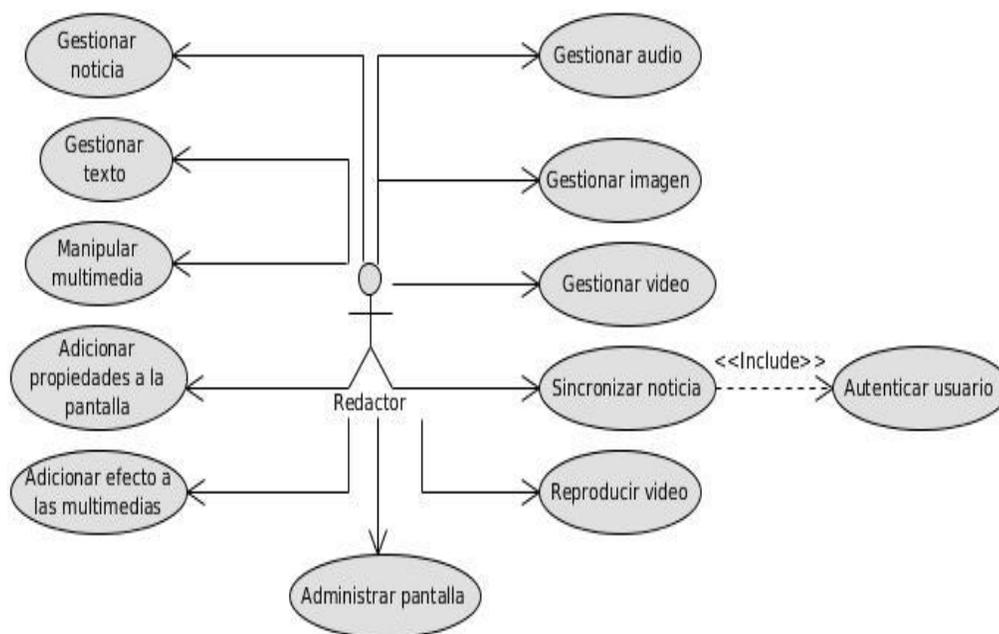


Figura 3 Diagrama de casos de uso del sistema

Capítulo 3. Diseño del Sistema

2.3.2 Descripción textual de los casos de uso del sistema

Dado que la descripción textual de los casos de uso es muy extensa sólo se muestra la descripción del caso de uso Gestionar Noticias, el resto de las descripciones se encuentran en el Expediente de Proyecto.

Descripción extendida del caso de uso Gestionar Noticias:

Caso de Uso	Gestionar Noticias
Actores	Redactor de noticias
Propósito	Este caso de uso se lleva a cabo con el objetivo de redactar las noticias.
Resumen	El caso de uso inicia cuando el actor desea realizar algunas de las siguientes operaciones: Insertar, Modificar, Eliminar, Listar o Buscar las noticias en el dispositivo móvil y termina cuando han sido completadas las acciones seleccionadas.
Complejidad	Alta
Prioridad	Crítico
Precondiciones	Debe existir al menos un archivo de audio almacenado.
Referencias	RF2, RF3, RF4, RF5, RF6, RF7
Flujo Normal de Eventos	
Acciones del Actor	Respuesta del Sistema
1. El caso de uso se inicia cuando se selecciona la opción <i>Gestionar Noticias</i> en la Página Principal. (Ver interfaz 1.1)	2. El sistema muestra una interfaz que contiene el listado de noticias registradas. <ul style="list-style-type: none">• Si el actor desea adicionar una noticia, ir a la sección “Adicionar Noticia”.• Si el actor desea eliminar una noticia, ir a la sección “Eliminar Noticia”.• Si el actor desea modificar una noticia, ir a la sección “Modificar Noticia”.

Capítulo 3. Diseño del Sistema

	<ul style="list-style-type: none"> • Si el actor desea visualizar una noticia, ir a la sección “Visualizar Noticia”. • Si el actor desea buscar una noticia, ir a la sección “Buscar Noticia”.
--	--

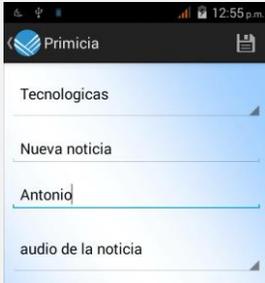
Prototipo de Interfaz 1.1



Sección “Adicionar Noticia”

Acciones del Actor	Respuesta del Sistema
<p>1. El actor selecciona la opción <i>Adicionar Noticia</i> que se encuentra contenida en el actionbar.</p>	<p>2. El sistema muestra una interfaz para adicionar una noticia que contiene:</p> <ul style="list-style-type: none"> - Un Spinner para seleccionar la sección a la cual pertenece esa noticia. - Un EditText para introducir el título de la noticia. - Un Spinner para seleccionar el audio asociado a la noticia. - Un botón que representa la acción de Guardar.

Capítulo 3. Diseño del Sistema

<p>3. El actor introduce todos los datos asociados a la noticia y da clic en el botón <i>Guardar</i>. (Ver interfaz 1.1)</p>	<p>4. El caso de uso termina cuando el sistema adiciona la nueva noticia, y muestra la interfaz que contiene las pantallas asociada a esa noticia y el mensaje de confirmación de que la noticia fue adicionada satisfactoriamente.</p>
Flujo Alternativo	
<p>3.1 El actor no introduce algunos de los datos asociados a la noticia y da clic en el botón <i>Guardar</i>.</p>	<p>3.2 El sistema muestra un mensaje de error indicando que debe llenar los campos para que se pueda ejecutar la acción. Se realiza el paso 3 de la Sección “Adicionar Noticia”. (Ver interfaz 1.2)</p>
<p>3.1 El actor da clic en el botón <i>Atrás</i> del teléfono.</p>	<p>3.2 El sistema cierra la interfaz de Adicionar Noticia sin realizar cambios.</p>
Prototipo de Interfaz 1.1	
	
Prototipo de Interfaz 1.2	

Capítulo 3. Diseño del Sistema



Sección “Eliminar Noticia”

Acciones del Actor	Respuesta del Sistema
1. El actor mantiene presionada la noticia que desea eliminar del listado que se muestra en la interfaz Gestionar Noticias.	2. El sistema muestra un cuadro de diálogo que contiene la opción Eliminar. (Ver interfaz 1.1)
3. El actor selecciona la opción <i>Eliminar</i> .	4. El sistema muestra un mensaje indicando que debe confirmar si está seguro que desea eliminar la noticia seleccionada. (Ver interfaz 1.2)
5. El actor da clic en el botón <i>Sí</i> .	6. El caso de uso termina cuando el sistema elimina la noticia seleccionada, muestra la interfaz que contiene el listado de las noticias, actualizada y el mensaje de confirmación de que la noticia fue eliminada satisfactoriamente.

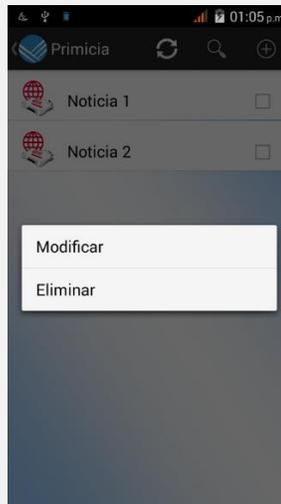
Flujos Alternos

Capítulo 3. Diseño del Sistema

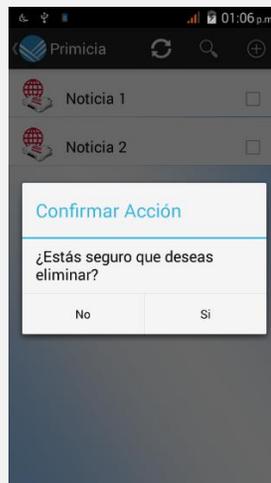
5.1 El actor da clic en el botón *No*.

5.2 El sistema cierra la ventana sin realizar cambios y muestra la interfaz que contiene el listado de las noticias.

Prototipo de Interfaz 1.1



Prototipo de Interfaz 1.2



Sección "Modificar Noticias"

Acciones del Actor

Respuesta del Sistema

Capítulo 3. Diseño del Sistema

<p>1. El actor mantiene presionada la noticia que desea modificar, de la interfaz que muestra el listado de las noticias.</p>	<p>2. El sistema muestra un cuadro de diálogo que contiene la opción Modificar. (Ver interfaz 1.1 de la sección Eliminar Noticia)</p>
<p>3. El actor selecciona la opción <i>Modificar</i>.</p>	<p>4. El sistema muestra un cuadro de diálogo con los datos de la noticia seleccionada.</p> <p>- Dos botones: Actualizar y Cancelar. (Ver interfaz 1.1)</p>
<p>5. El actor introduce los nuevos datos y da clic en el botón <i>Actualizar</i>.</p>	<p>6. El caso de uso termina cuando el sistema actualiza los datos de la noticia seleccionada, muestra la interfaz que contiene el listado de las noticias, actualizada y el mensaje de confirmación de que la noticia fue modificada satisfactoriamente. (Ver interfaz 1.2)</p>

Flujos Alternos

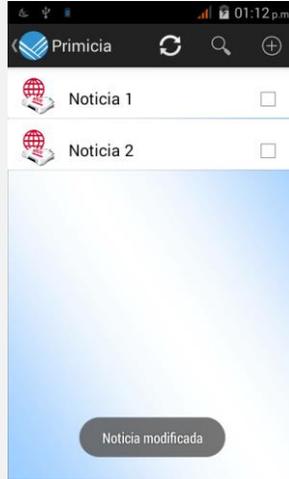
<p>5.1 El actor introduce los nuevos datos y da clic en el botón <i>Cancelar</i>.</p>	<p>5.2 El sistema no realiza cambios, cierra el cuadro de diálogo y muestra la interfaz que contiene el listado de las noticias.</p>
---	--

Prototipo de Interfaz 1.1



Capítulo 3. Diseño del Sistema

Prototipo de Interfaz 1.1



Sección “Buscar Noticias”

Acciones del Actor	Respuesta del Sistema
1. El actor selecciona la opción <i>Buscar</i> contenida en el actionbar que se muestra en la pantalla que contiene el listado de las noticias.	2. El sistema muestra un cuadro de diálogo que permite realizar la búsqueda de las noticias. (Ver interfaz 1.1)
3. El actor introduce palabras asociadas al título de la noticia.	4. El caso de usa termina cuando el sistema captura los caracteres insertados y muestra una nueva lista con las noticias relacionadas.

Flujos Alternos

3.1 Introduce palabras asociadas al título de una noticia que no está creada.	3.2 El sistema muestra el listado de noticias vacío.
---	--

Prototipo de Interfaz 1.1

Capítulo 3. Diseño del Sistema



Sección “Visualizar Noticias”

Acciones del Actor	Respuesta del Sistema
1. El actor selecciona la noticia que desea visualizar del listado que se muestra en la interfaz que contiene el listado de las noticias.	2. El sistema muestra las pantallas asociadas a la noticia seleccionada.

Conclusiones parciales

A partir del desarrollo del presente capítulo se arribaron a las siguientes conclusiones:

- Se logró un mejor entendimiento del entorno real del subsistema de administración de PRIMICIA a través de la realización de un modelo conceptual. Este modelo sirvió de base para la identificación de los requisitos funcionales de la aplicación.
- Se determinaron las diferentes funcionalidades y cualidades que el sistema debe cumplir mediante la especificación de requisitos. Se identificaron cuarenta y siete (47) requisitos funcionales que fueron agrupados en trece (13) casos de uso, de los cuales nueve (9) son críticos y cuatro (4) secundarios.
- Se realizó la descripción textual de cada caso de uso, lo cual permitió tener un mejor entendimiento de las funcionalidades que la aplicación debe tener, el comportamiento del mismo y las secuencias de actividades que debe realizar el usuario para lograr su objetivo.

Capítulo 3. Diseño del Sistema

Introducción

Cuando se inicia la construcción de un sistema de software, luego de conocer cuáles son las necesidades básicas del usuario final, el ambiente donde será desarrollado el mismo, las condiciones para ello, las tecnologías y recursos necesarios para lograr un ambiente de trabajo adecuado, entonces se sientan las bases del esqueleto arquitectónico del sistema (Pressman, 2008).

En el presente capítulo se realiza una descripción de la arquitectura definida y los patrones de arquitectura y de diseño que se utilizan. Además, se muestran los diagramas propuestos por la metodología apoyándose en modelos que permiten la comprensión del diseño propuesto, así como el modelo de datos empleado.

3.1. Arquitectura de Software

La arquitectura de software es un conjunto de patrones que proporcionan un marco de referencia necesario para guiar la construcción de un software, permitiendo a los programadores, analistas y todo el conjunto de desarrolladores del software compartir una misma línea de trabajo y cubrir todos los objetivos y restricciones de la aplicación. Es considerada el nivel más alto en el diseño de la arquitectura de un sistema, puesto que establecen la estructura, funcionamiento e interacción entre las partes del software.

De acuerdo con Robert Pressman la arquitectura de software no es otra cosa que una descripción de los subsistemas y los componentes de un sistema informático y las relaciones entre ellos. De igual manera, la arquitectura de software Modelo – Vista – Controlador, incluye todos estos aspectos y además, brinda mejores opciones para proyectos informáticos de gran alcance y complejidad.

3.1.1. *Estilo de Arquitectura*

Un estilo arquitectónico o variante arquitectónica define a una familia de sistemas informáticos en términos de su organización estructural. Un estilo arquitectónico describe componentes y las relaciones entre ellos con las restricciones de su aplicación, la composición asociada y el diseño para su construcción. El estilo utilizado para el desarrollo de la aplicación es el de Llamada y Retorno.

Capítulo 3. Diseño del Sistema

3.1.2. Patrón Arquitectónico Modelo – Vista – Controlador

Para el desarrollo de aplicaciones en dispositivos móviles sobre la plataforma Android se propone el uso del patrón arquitectónico Modelo – Vista – Controlador. Este patrón permite separar los datos de la aplicación, la interfaz de usuario y la lógica del negocio en tres componentes distintos que se relacionan entre sí para al final tener como resultado un producto de software (Androideity.com, 2014). El modelo contiene el núcleo de la funcionalidad (dominio) de la aplicación, encapsula el estado de la aplicación y es independiente del Controlador y la Vista. La vista es la presentación del Modelo y puede acceder al Modelo pero nunca cambiar su estado y el controlador reacciona a la petición del cliente, ejecutando la acción adecuada y creando el modelo pertinente.

En esta parte del modelo también juega la decisión de comunicarse mediante servicio web. La vista es la interfaz con la que el usuario interactúa que son construidas en XML. El controlador son todas las clases que le proporcionan información a las interfaces. Estos controladores se programan en lenguaje Java y son el núcleo de la aplicación.

Vista: La vista no es más que la interfaz con la que va a interactuar el usuario. Incluye las interfaces de usuario y aquellos artefactos de Android sin interfaz gráfica. En android la vista está conformada por los ficheros XML. Tiene todos los elementos necesarios para mostrarle al usuario lo que desea observar.

Controlador: Está compuesto por las clases del negocio, que permiten desplegar y consumir información de la interfaz de usuario. Hace de puente entre la Vista y el Modelo, incluyendo clases con métodos que alberguen la lógica de la herramienta. Siempre los artefactos de la capa Vista invocarán a las clases albergadas en esta capa, nunca accederán al Modelo directamente. Las clases controladoras contenidas en esta capa de negocio son las que albergan el control y gestión sobre todo el contenido referente a cada caso de uso.

Modelo: Es la representación de la información con la cual el sistema opera, por lo tanto gestiona todos los accesos a dicha información, tanto consultas como actualizaciones, implementando también los privilegios de acceso que se hayan descrito en las especificaciones de la aplicación.

3.1.3. Vista Lógica

La vista lógica del sistema ilustra las realizaciones de casos de uso claves, subsistemas, paquetes y clases que abarcan el comportamiento significativo arquitectónicamente. Muestra un subconjunto del modelo de

Capítulo 3. Diseño del Sistema

diseño, es decir, un subconjunto de las clases, subsistemas y paquetes, y realizaciones de casos de uso (Pressman, 2008). En la Figura 4 se representa la vista lógica de la aplicación.

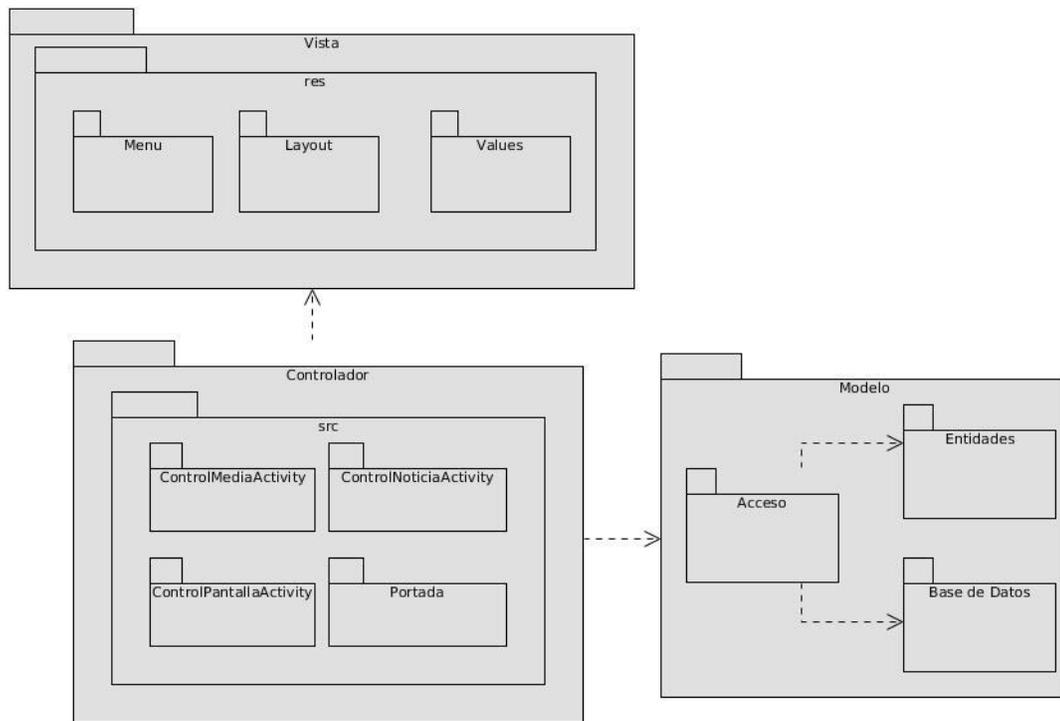


Figura 4 Vista lógica

3.2. Modelo de Diseño

El modelo de diseño es una abstracción de la implementación del sistema. Se utiliza para concebir y documentar el diseño del sistema de software. Es un producto de trabajo integral y compuesto que contiene los artefactos: clases de diseño, interfaz, paquetes de diseño, subsistemas de diseño, ejecución de casos de uso, realizaciones de casos de uso, entre otros (IBM, 2006). En el presente acápite se mostrará el diagrama de clases del diseño y los patrones que se utilizaron para su estructuración.

3.2.1. Diagrama de Clases del Diseño

Un diagrama de clases sirve para visualizar las relaciones entre las clases que involucran el sistema, los cuales responden al cumplimiento de los requisitos funcionales. Los diagramas de clases del diseño son subproductos del modelo de diseño proporcionando una perspectiva estática que representa el diseño estructural del sistema, mostrando un conjunto de clases y sus atributos (Larman, 2003). La realización de

Capítulo 3. Diseño del Sistema

los mismos en el desarrollo del trabajo en curso sirve de guía y entrada a los desarrolladores para el flujo de trabajo de implementación.

A continuación se muestra en la Figura 5 el diagrama de clases del diseño del caso de uso Gestionar Noticias, para el cual se tuvieron en cuenta los patrones de diseño anteriormente descritos.

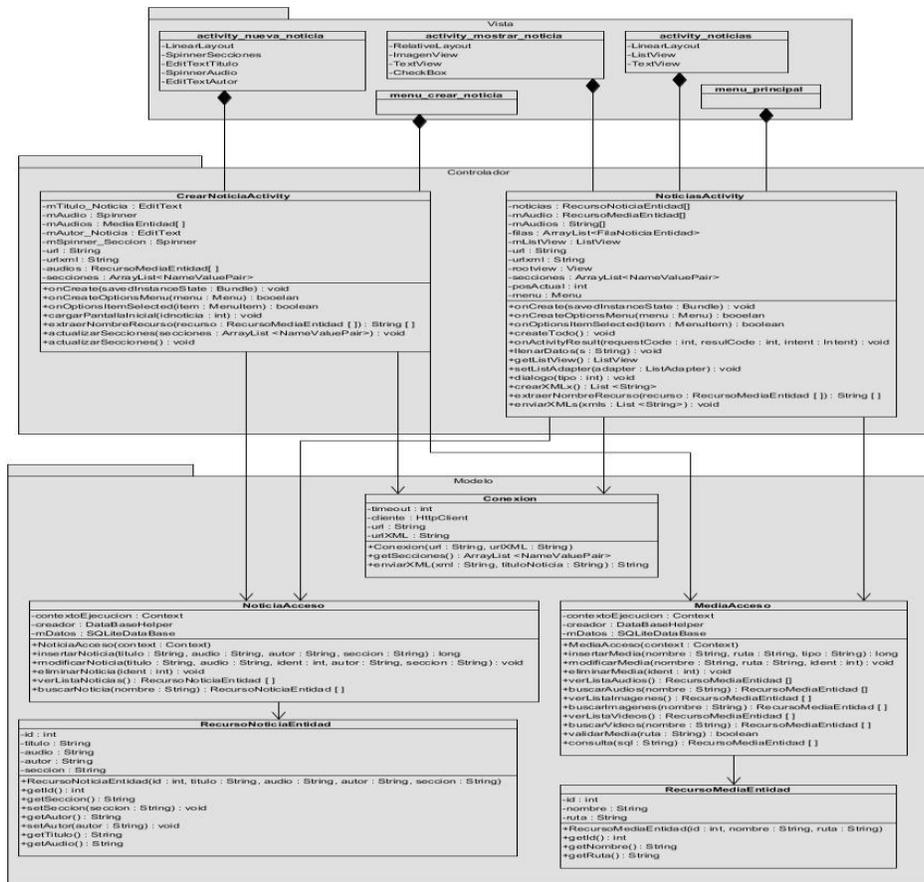


Figura 5 Diagrama de clases del diseño del caso de uso Gestionar Noticias

3.2.2. Patrones de Diseño

Para el desarrollo de software un elemento fundamental y de buena práctica, lo constituye el uso de patrones. Representan una excelente manera de trabajar en base a la calidad. Además, un patrón de diseño es una buena práctica documentada o solución, que se ha aplicado con éxito en múltiples ambientes para erradicar problemas comunes de diseño de software, con una probada efectividad y con características de reutilización (Fowler, 2007).

Capítulo 3. Diseño del Sistema

Patrones GRASP

Durante el desarrollo del sistema se utilizaron patrones básicos de asignación de responsabilidades como Patrones Generales de Software para Asignar Responsabilidades (General Responsibility Assignment Software Patterns por sus siglas en inglés) GRASP.

Experto: Se encarga de asignar la responsabilidad a la clase que cuenta con la información necesaria para efectuar la tarea que tiene encomendada. En la clase *AudioActivity* se evidencia el uso de este patrón ya que dicha clase es la única que contiene la información para gestionar todo lo referente a los audios que se almacenan en la aplicación.

Controlador: Es utilizado por todas las clases que forman parte de la capa Controlador. Estas clases poseen la responsabilidad de controlar el flujo de eventos mediante las actividades correspondientes. Para la construcción de la aplicación se definieron siete clases controladoras (*NoticiasActivity*, *CrearNoticiaActivity*, *AudioActivity*, *VideoActivity*, *ImagenActivity*, *PantallasActivity*, *InicioActivity*) encargadas de controlar la lógica del negocio en lo que respecta a su caso de uso correspondiente. La clase *NoticiasActivity* tiene la función de controlar las operaciones realizadas sobre las noticias en la aplicación.

Creador: Tiene en cuenta para la asignación de responsabilidades a las clases relacionadas con la creación de objetos, de forma tal que una instancia de un objeto sólo pueda ser creada por el objeto que contiene la información necesaria. El uso de este patrón permite crear las dependencias mínimas necesarias entre las clases, lo cual favorece al mantenimiento del sistema. Este patrón se evidencia en la clase *NoticiasActivity* perteneciente a la capa Controlador, crea un objeto de la clase *Conexión*, incluida en la capa Modelo, para poder comunicarse.

Bajo acoplamiento: Este patrón es utilizado en la creación de clases independientes para la lógica de los diferentes tipos de clases, actividades y entidades. Esto trae como ventaja que sólo se realicen acciones sobre el tipo de entidad que se solicite o formulario correspondiente y no sobre todo el conjunto. Estas clases se encargan de la representación de los elementos reales de cada uno respectivamente y las actividades, de manejar los componentes visuales de cada funcionalidad dentro de los diferentes casos de uso.

Alta cohesión: Propone asignar la responsabilidad de manera que la complejidad se mantenga dentro de límites manejables, para evadir un trabajo excesivo. Su utilización mejora la claridad y facilidad con que se

Capítulo 3. Diseño del Sistema

entiende el diseño, simplifica el mantenimiento y las mejoras de funcionalidad, generan un bajo acoplamiento, soporta mayor capacidad de reutilización.

Patrones GoF

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

A continuación se describe el patrón GoF aplicado en el diseño del sistema.

Adapter: Convierte la interfaz de una clase en otra distinta que es la que esperan los clientes. Permite que cooperen clases que de otra manera no podrían, por tener interfaces incompatibles. Se utiliza para transformar una interfaz en otra, de tal modo que una clase que no pudiera utilizar la primera, haga uso de ella a través de la segunda. Un ejemplo de este patrón se observa en la clase *MediasPagerAdapter* donde se crea el visual de las clases *ImagenActivity*, *AudioActivity* y *VideoActivity*.

3.3. Diseño de la Base de Datos

Las bases de datos son un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso. Es una colección de información organizada de forma que un programa de ordenador pueda seleccionar rápidamente los fragmentos de datos que necesite. Una base de datos correctamente diseñada permite obtener acceso a la información exacta y actualizada (Hernández, 2010). A continuación se representa en la Figura 6 el diagrama entidad – relación de la aplicación.

3.3.1. Diagrama Entidad - Relación

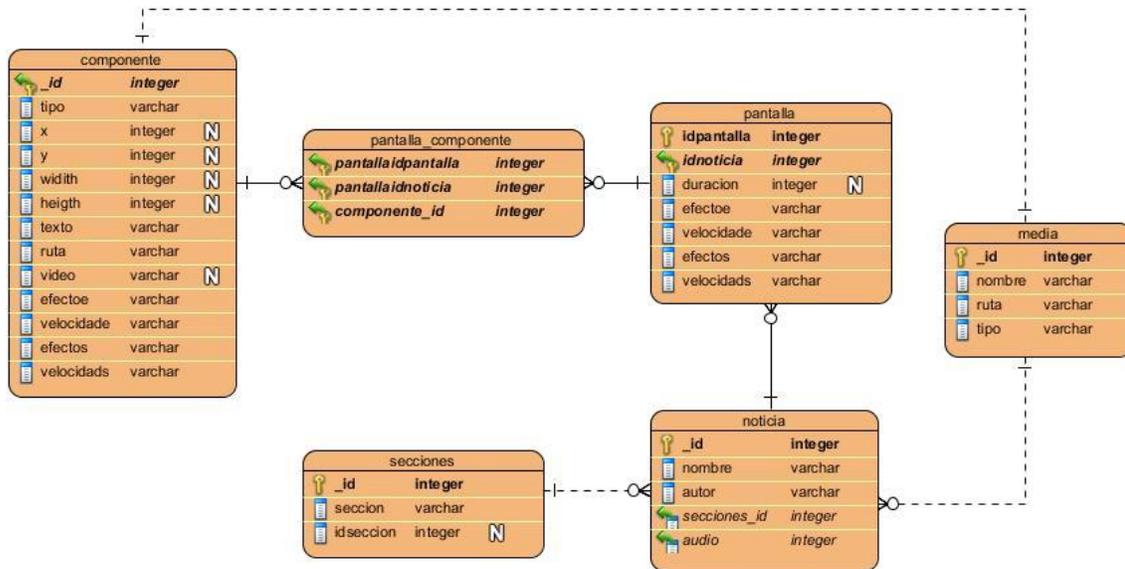


Figura 6 Diagrama entidad-relación

3.4. Diagrama de Despliegue

Describe la distribución física de los distintos nodos que componen el sistema y la relación que se establece entre ellos. Estos nodos representan los elementos de software que entran en la realización del sistema y se relacionan mediante soportes bidireccionales. Su objetivo es mostrar cómo y dónde se desplegará el sistema. En la Figura 7 se muestra el diagrama de despliegue perteneciente al sistema propuesto.

Descripción de los nodos físicos del sistema:

<<device>> Dispositivo Móvil: En el dispositivo móvil es donde se va a encontrar la aplicación para la redacción de las noticias.

<<executionEnvironment>> Servidor de Aplicaciones Plataforma PRIMICIA: Sistema con el cual la aplicación se conecta a través de un servicio web para sincronizar las noticias que se gestionan en el teléfono.

<<executionEnvironment>> Servidor de Base de Datos de PRIMICIA: Es el nodo del cual se extraen los datos que el sistema va a entregarle a la aplicación móvil por el servicio implementado.

Capítulo 3. Diseño del Sistema

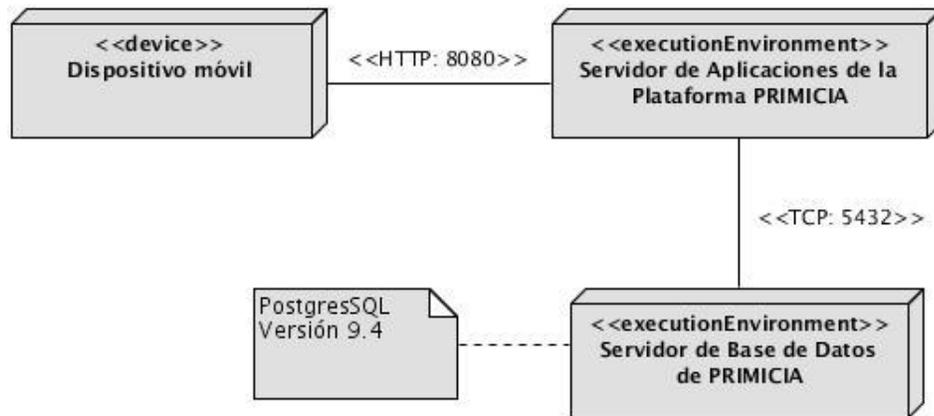


Figura 7 Diagrama de despliegue

Conclusiones parciales

A partir del desarrollo del presente capítulo se arribaron a las siguientes conclusiones:

- La definición del estilo Llamada y Retorno en conjunto con el patrón arquitectónico Modelo – Vista – Controlador y los patrones de diseño GRASP y GoF empleados, aseguró una adecuada estructuración de la aplicación, haciendo más fácil la integración de los componentes del sistema, su actualización, corrección y mejora. De igual forma garantizó una mayor simplicidad en el desarrollo así como la escalabilidad.
- El modelo físico constituye la base persistente final para el almacenamiento eficiente de la información.
- La realización del modelo de despliegue brinda una distribución completa del acople de los distintos componentes por lo que está compuesto el sistema. Se tendrá en cuenta que la aplicación se ejecutará en un dispositivo móvil para favorecer la movilidad del redactor (periodista) y cuando se tengan las condiciones técnicas o lo desee se podrá comunicar con el servidor de aplicaciones de PRIMICIA, para enviar las noticias.

Capítulo 4. Implementación y Pruebas

Introducción

La prueba es un elemento crítico para la garantía de la calidad del software, es el proceso que permite verificar y revelar la calidad de un producto de software. Son utilizadas para identificar posibles fallos de implementación, calidad o usabilidad de un programa. Básicamente es una de las fases del desarrollo de software consistente en probar las aplicaciones construidas. Las pruebas de software se integran dentro de las diferentes fases del ciclo del software dentro de la Ingeniería de Software (Jacobson, y otros, 2000).

En el presente capítulo se presentan los artefactos que forman parte de los flujos de trabajo de implementación y prueba de la solución propuesta.

4.1 Modelo de Implementación

El modelo de implementación describe como los elementos del modelo de diseño, se implementan en términos de componentes. Muestra cómo se organizan los componentes de acuerdo a los mecanismos de estructuración disponibles en el entorno de implementación y en el lenguaje o lenguajes de programación utilizados, y como dependen los componentes unos de otros (Jacobson, y otros, 2000).

4.1.1 Diagrama de Componentes

Un diagrama de componentes representa las dependencias entre componentes de software, incluyendo componentes de código fuente, componentes del código binario, y componentes ejecutables. Se utilizan para modelar la vista estática de un sistema. Muestra la organización y las dependencias entre un conjunto de componentes. No es necesario que un diagrama incluya todos los componentes del sistema, normalmente se realizan por partes. Cada diagrama describe un apartado del sistema. Uno de los usos principales es que puede servir para ver qué componentes pueden compartirse entre sistemas o entre diferentes partes de un sistema (Rivera, 2014).

En la Figura 8 se muestra el diagrama de componentes del caso de uso Gestionar Noticias.

Capítulo 4. Implementación y Pruebas

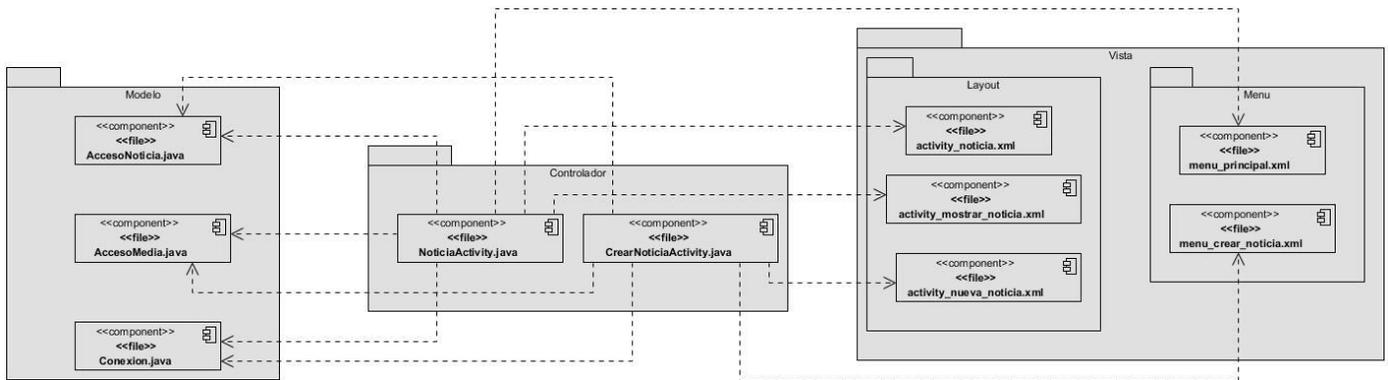


Figura 8 Diagrama de componentes del caso de uso Gestionar Noticias

4.2 Estándar de Codificación

Las convenciones o estándares de codificación son pautas de programación que no están enfocadas a la lógica del programa, sino a su estructura y apariencia física para facilitar la lectura, comprensión y mantenimiento del código. Mantener el código es la facilidad con que el sistema de software puede modificarse para añadirle nuevas características, modificar las ya existentes, depurar errores, o mejorar el rendimiento (Becerra, 2011).

Para el desarrollo de la herramienta se definieron varios estándares de codificación los cuales se describen a continuación:

Número de declaración por línea

Se debe declarar cada variable en una línea distinta, de esta forma cada variable se puede comentar por separado.

Ejemplo:

```
private RecursoNoticia[] noticias;  
private Recurso[] audios;  
ArrayList<Fila> filas;  
private ListView mListview;  
private View rootView;  
private int posActual;
```

Capítulo 4. Implementación y Pruebas

Asignación de nombres

Cada tipo de elemento debe nombrarse con una serie de reglas determinadas.

Clases e Interfaces: La letra inicial debe ser mayúscula, ya sea simple o compuesta.

Ejemplo:

```
public class Noticias extends ActionBarActivity
public class CrearPantalla extends ActionBarActivity implements
    View.OnTouchListener {
```

Métodos: Deben ser escritos en forma de verbos. La primera letra de la palabra comienza con minúscula, las demás palabras comienzan con mayúscula.

Ejemplo:

```
private void cargarPantalla()
```

Variables: Se debe comenzar con minúscula. Y no usar en ningún caso caracteres extraños como es “_”.

Ejemplo:

```
private EditText mNombreNoticia;
private EditText mDescripcionNoticia;
private Spinner mCategory;
private Spinner mAudio;
```

4.3 Pruebas del Sistema

Las pruebas son básicamente un conjunto de actividades dentro del desarrollo de software y dependiendo del tipo de pruebas, estas actividades podrán ser implementadas en cualquier momento del proceso de desarrollo. También son un proceso que se enfoca sobre la lógica interna del software y las funciones externas, es un proceso de ejecución de un programa con la intención de descubrir un error.

4.3.1 Pruebas de Caja Negra

Las pruebas de caja negra se llevan a cabo sobre la interfaz del software. Se centran principalmente en los requisitos funcionales del software. Estas pruebas permiten obtener un conjunto de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un programa. En ellas se ignora la estructura de control, concentrándose en los requisitos funcionales del sistema y ejercitándolos.

Capítulo 4. Implementación y Pruebas

Estas pruebas intentan encontrar las funciones incorrectas o ausentes, errores de rendimiento, errores de inicialización y terminación y errores en la estructura de datos o acceso a la base de datos externa.

Dentro de los métodos de caja negra se utilizó la **técnica de partición de equivalencia**. Esta se encarga de dividir el dominio de entrada de un programa en clases de datos de los que se pueden derivar casos de prueba. El diseño de estos casos de prueba para la partición de equivalencia se basa en la evaluación de las clases de equivalencia.

Caso de Prueba del Caso de Uso Gestionar Noticias

Un caso de prueba es un conjunto de condiciones o variables bajo las cuáles un analista determinará si una aplicación o un sistema software, es parcial o completamente satisfactoria. Se pueden derivar de los casos de uso del sistema permitiendo así la validación de los requisitos funcionales del sistema.

Casos de Prueba del Caso de Uso Gestionar Noticias

- 1. Descripción general:** El caso de uso inicia cuando el usuario desea realizar algunas de las siguientes operaciones: Insertar, Modificar, Eliminar o Visualizar los datos asociados a una noticia y termina cuando han sido completadas las acciones seleccionadas.
- 2. Condiciones de ejecución:** Debe tener algún audio almacenado en la aplicación.
- 3. Secciones a probar en el Caso de Uso Gestionar Noticias.**

Capítulo 4. Implementación y Pruebas

Tabla 4 Caso de prueba del caso de uso “Gestionar Noticias”

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo central
SC 1: “Adicionar noticias”.	EC 1.1: “Adicionar noticia satisfactoriamente”.	El usuario entra los datos de la noticia, y da clic en el actionbar en el botón “Guardar”. El sistema adiciona la nueva noticia y muestra la interfaz Gestionar Pantalla y un mensaje indicando, que la noticia fue adicionada satisfactoriamente.	Inicio/Gestionar Noticias/Añadir noticia/Clic en el botón Guardar.
	EC 1.2: “Adicionar noticias dejando campos vacíos que son obligatorios”.	El usuario no llena algún campo obligatorio de la noticia y da clic en el botón “Guardar”. El sistema indica al actor que debe llenar los campos faltantes, mostrándole un mensaje de campos vacíos.	Inicio/Gestionar Noticias/Añadir noticia/Clic en el botón Guardar.
	EC 1.3: “Cancelar la adición de la noticia”.	El usuario da clic en el botón “Cancelar”. El sistema cierra la ventana Adicionar Noticias.	Inicio/Gestionar Noticias/Añadir noticia/Clic en el botón Cancelar.

Capítulo 4. Implementación y Pruebas

SC 2: “Eliminar noticia”	EC 2.1: “Eliminar noticias satisfactoriamente”.	El usuario mantiene presionada la pantalla mostrándosele un menú en la noticia que desea eliminar y da clic en el botón Eliminar. El sistema elimina la noticia y muestra la interfaz Gestionar Noticia actualizada.	Inicio/Gestionar Noticias/Seleccionar una noticia del listado/Mantiene presionada la pantalla/Clic en el botón Eliminar.
	EC 2.2: “Cancelar la eliminación de la noticia”.	El usuario mantiene presionada la pantalla mostrándosele un cuadro de Diálogo de la noticia que desea eliminar, el usuario da atrás y sale del cuadro de diálogo. El sistema no elimina la noticia y mantiene visible la ventana Gestionar Noticias.	Inicio/Gestionar Noticias/Seleccionar una noticia del listado/Mantiene presionada la pantalla/Clic en el botón Eliminar/Clic en la opción Cancelar
SC3: Modificar noticias.	EC 3.1: “Modificar noticia satisfactoriamente”.	El usuario selecciona la noticia que desea actualizar y da clic en el botón Modificar. Introduce los nuevos datos y da clic en el botón Actualizar. El sistema actualiza los datos de la noticia seleccionada, y muestra la interfaz Gestionar Noticias actualizada.	Inicio/Gestionar Noticia/ Seleccionar la noticia del listado/Clic en el botón Editar/Clic en el botón Actualizar.

Capítulo 4. Implementación y Pruebas

	EC 3.2: “Modificar noticias dejando campos vacíos”.	El usuario selecciona la noticia que desea actualizar y da clic en el botón Modificar. El usuario deja alguno de los campos obligatorios vacío y da clic en el botón Actualizar. El sistema indica al actor que debe llenar los campos faltantes, indicándole que tiene campos vacíos.	Inicio/Gestionar Noticia/ Seleccionar la noticia del listado/Clic en el botón Editar/Clic en el botón Actualizar.
	EC 3.3: “Cancelar modificación de una noticia”.	El usuario selecciona la noticia y da clic en el botón Modificar. Introduce los nuevos datos y da clic en el botón Cancelar. El sistema no realiza cambios, cierra la ventana y muestra la interfaz Gestionar Noticias.	Inicio/Gestionar Noticia / Seleccionar la noticia del listado/Clic en el botón Editar/Clic en el botón Cancelar.
SC4: Mostrar datos de la noticia.	EC 4.1: “Mostrar datos de la noticia satisfactoriamente”.	El usuario selecciona la opción de mostrar noticias. El sistema muestra las noticias.	Inicio/Gestionar Noticias /Clic en el botón Mostrar.

Capítulo 4. Implementación y Pruebas

4. Matriz de datos:

SC 1: “Adicionar Noticias”

Tabla 5 Matriz de datos SC 1: “Adicionar noticias”

ID del escenario	Escenario	Secciones	Título	Autor	Audio	Respuesta del sistema	Resultado de la prueba
EC 1.1	Adicionar noticias satisfactoriamente.	V	V	V	V	El sistema adiciona la nueva noticia y muestra la interfaz donde se listan	Satisfactorio
EC 1.2	Adicionar noticia dejando campos vacíos que son obligatorios.	I	V	V	V	El sistema indica que el actor debe llenar los campos obligatorios.	Satisfactorio
		V	I	V	V	El sistema indica que el actor debe llenar los campos obligatorios.	Satisfactorio
		V	V	I	V	El sistema indica que el actor debe llenar los campos obligatorios.	Satisfactorio
		V	V	V	I	El sistema indica que el actor debe llenar los campos obligatorios.	Satisfactorio

Capítulo 4. Implementación y Pruebas

EC 1.3	Cancelar la adición de la noticia.	NA	NA	NA	NA	El sistema cierra la ventana Adicionar Noticias.	Satisfactorio
--------	------------------------------------	----	----	----	----	--	---------------

SC 2: “Eliminar noticias”.

Tabla 6 Matriz de datos SC 2: “Eliminar noticias”

ID del escenario	Escenario	Noticia Registrada	Respuesta del sistema	Resultado de la prueba
EC 2.1	Eliminar noticia satisfactoria mente.	V	El sistema elimina la noticia y muestra la interfaz Gestionar noticias actualizada.	Satisfactorio
EC 2.2	Cancelar la eliminación de una noticia.	NA	El sistema no elimina la noticia y mantiene visible la ventana Gestionar Noticias.	Satisfactorio

Capítulo 4. Implementación y Pruebas

SC 3: “Modificar noticias”

Tabla 7 Matriz de datos SC 3: “Modificar noticias”

ID del escenario	Escenario	Secciones	Título	Autor	Audio	Respuesta del sistema	Resultado de la prueba
EC 3.1	Modificar noticias satisfactoriamente.	V	V	V	V	El sistema actualiza los datos de la noticia seleccionada, y muestra la interfaz Gestionar Noticias actualizada.	Satisfactorio
EC 3.2	Modificar noticias dejando campos vacíos.	I	V	V	V	El sistema indica al actor que debe llenar los campos vacíos.	Satisfactorio
		V	I	V	V	El sistema indica al actor que debe llenar los campos vacíos.	Satisfactorio
		V	V	I	V	El sistema indica al actor que debe llenar los campos vacíos.	Satisfactorio
		V	V	V	I	El sistema indica al actor que debe llenar los campos vacíos.	Satisfactorio
EC 3.3	Cancelar modificaci	NA	NA	NA	NA	El sistema no realiza cambios, cierra la	Satisfactorio

Capítulo 4. Implementación y Pruebas

	ón de una noticia.					ventana y muestra la interfaz Gestionar Noticias.	
--	--------------------	--	--	--	--	---	--

SC 4: “Mostrar datos de la noticia”

Tabla 8 Matriz de datos SC 4: “Mostrar datos de la noticia”

ID del escenario	Escenario	Servicio seleccionado	Respuesta del sistema	Resultado de la prueba
EC 4.1	Mostrar datos de la noticia satisfactoria mente.	V	El sistema muestra los datos de la noticia.	Satisfactorio

4.3.2 Resultado de las pruebas

Una vez ejecutadas las pruebas de caja negra haciendo uso para ello de los casos de pruebas por cada caso de uso, se verificó el correcto funcionamiento de la aplicación. Para lograr estos resultados se realizaron tres iteraciones, en dos de ellas se detectaron ocho (8) no conformidades de la aplicación y tres (3) no conformidades en el documento. En la tercera iteración todas las no conformidades fueron resueltas.

Capítulo 4. Implementación y Pruebas

Primera Iteración:

- Las medias que se adicionaban no se mostraban en el listado de medias. (Aplicación) (Significativo)
- Permitía eliminar un audio, imagen o video almacenado en la aplicación, que estaba asociado a una noticia. (Aplicación) (Significativo)
- No guardaba las configuraciones o cambios realizados sobre las pantallas. (Aplicación) (Significativo)
- La interfaz construida no se adaptaba a las diferentes dimensiones de las pantallas de los dispositivos móviles. (Aplicación) (Significativo)
- Errores de validación cuando se deseaba insertar una media. (Aplicación) (Significativo)
- Errores en relaciones plasmadas en el modelo conceptual. (Documento) (No Significativo)

Segunda Iteración:

- No se cerraba el cuadro de diálogo de progreso cuando se sincroniza una noticia. (Aplicación) (Significativo)
- Permitía agregar una noticia dejando campos que debían ser obligatorios vacíos. (Aplicación) (Significativo)
- Errores de ortografía en la aplicación. (Aplicación) (No Significativo)
- Errores en relaciones plasmadas en el diagrama de casos de uso del sistema. (Documento) (No Significativo)
- Errores en la definición de la arquitectura a seguir. (Documento) (Significativo)

4.3.3 Pruebas de Aceptación

Luego de la culminación de las pruebas de caja negra se procedió a realizar las pruebas de aceptación; bajo un ambiente controlado para determinar si el software realizaba lo que había solicitado el cliente. Terminadas estas pruebas y corregidas las no conformidades, el cliente avaló la solución con la entrega del Acta de Aceptación del Producto.

4.4 Validación de la contribución lograda

Tomando en cuenta la caracterización de los módulos de redacción de noticias y la gestión de medias, realizada en la investigación y las pruebas de funcionalidad (método de caja negra) realizadas a la

Capítulo 4. Implementación y Pruebas

aplicación; los autores de esta investigación concluyen que la utilización de la aplicación móvil asociada a los módulos de redacción de noticias y gestión de medias del subsistema de administración de PRIMICIA, permite valorar como positiva la contribución a los grados de portabilidad y accesibilidad de la información que se gestiona a través de los módulos antes mencionados.

Se arriba a dicha conclusión, teniendo en cuenta que:

- La relevancia en un proceso de redacción de noticias radica en la precisión y rapidez que puedan tener los redactores de noticias (periodistas) para recopilar información de diferentes fuentes, ubicadas en distintos lugares, que puedan ser estructuradas (creadas) por éstos y enviadas en el menor tiempo posible.
- Los redactores sean capaces no sólo de enviar a través de los diferentes canales la información, las imágenes y los videos asociados a la noticia, sino que puedan acceder a diversos elementos visuales que les permita crear su noticia, utilizando para ello las mismas potencialidades que les brinda un sistema al cual acceden desde una computadora conectada al servidor de aplicaciones que lo gestiona.
- El proceso que se siga para crear una noticia sea flexible en tanto se potencie la creación y configuración de noticias sin depender de la conectividad al servidor de aplicaciones.

La aplicación desarrollada cumple con los principios anteriormente mencionados, directamente asociados a las variables portabilidad y accesibilidad, las cuales estaban limitadas sustancialmente. La propuesta es una variante similar al sistema existente, que sirve de complemento para lograr una solución integral que contribuya a alcanzar los objetivos propuestos.

Capítulo 4. Implementación y Pruebas

Conclusiones parciales

A partir del desarrollo del presente capítulo se arribaron a las siguientes conclusiones:

- La confección del diagrama de componentes permitió representar una vista estática de la aplicación, mostrando la organización y dependencia que existe entre los componentes físicos que se necesitan para ejecutar la misma.
- Los estándares de codificación utilizados fueron descritos para hacer más fácil el entendimiento del código por el desarrollador así como un mejor mantenimiento de la aplicación en el futuro.
- Se realizaron las pruebas al sistema donde se emplea el método de caja negra y se aplica la técnica de partición de equivalencia. Esto permitió verificar los requisitos funcionales del sistema, demostrar que las funciones de la aplicación son operativas, producen un resultado satisfactorio y el sistema cumple con los objetivos trazados.

Conclusiones

Luego de la investigación e implementación desarrolladas, se logró dar cumplimiento a las tareas de la investigación, así como a los objetivos planteados inicialmente, obteniendo como resultado una aplicación que cumple con todos los requisitos y la calidad requerida, lo cual permite arribar a las siguientes conclusiones:

- En la actualidad los sistemas informáticos que se encargan de crear y distribuir noticias están diseñados para negocios y características específicas. De ahí la necesidad de haber creado una aplicación móvil para la gestión de las noticias.
- La arquitectura y diseño del sistema permitieron que se obtuviera una solución que fuera capaz de satisfacer las necesidades del cliente.
- Para comprobar la calidad y correcto funcionamiento del sistema, se diseñaron y ejecutaron casos de prueba, que arrojaron resultados satisfactorios, demostrando el cumplimiento de los requisitos funcionales establecidos en la fase inicial del proceso de desarrollo del sistema.
- Las pruebas realizadas demostraron que a partir de la utilización de la aplicación desarrollada, se garantizan altos grados de accesibilidad y portabilidad de los módulos de redacción de noticias y gestión de medias del subsistema de administración de PRIMICIA.

Recomendaciones

Al concluir la presente investigación asociada al desarrollo de la aplicación móvil para la gestión de noticias y medias de la plataforma PRIMICIA, se propone la siguiente recomendación:

- Desarrollar la aplicación de gestión de noticias y medias para otras plataformas móviles.

Referencias Bibliográficas

ABC, Definicion. 2007. Definición ABC. [En línea] 2007.
<http://www.definicionabc.com/tecnologia/multimedia.php>.

Android. 2013. Android. [En línea] 2013. [Citado el: 28 de octubre de 2014.]
<https://play.google.com/store/apps/details?id=org.thialfihar.android.apg&hl=es>.

Androideity.com. 2014. Androideity. [En línea] 10 de mayo de 2014. [Citado el: 20 de febrero de 2015.]
<http://androideity.com/2012/05/10/la-importancia-del-mvc-en-android/>.

AP ENPS, Móvil. 2012. AP ENPS Mobile. [En línea] 2012. [Citado el: 10 de junio de 2015.]
http://www.enps.com/pages/features_and_specifications/features/78/ap_enps_mobile.

Aranaz, Tudela, Jaime. 2009. *DESARROLLO DE APLICACIONES PARA DISPOSITIVOS MÓVILES SOBRE LA PLATAFORMA ANDROID DE GOOGLE*. Madrid : s.n., 2009.

Arquitectura, Software. 2010. Definición de Arquitectura de Software. *Definición de Arquitectura de Software*. [En línea] 2010. [Citado el: 2 de febrero de 2015.]
<http://www.mastermagazine.info/termino/3916.php>.

Autores, Colectivo de. 2007. ABC. [En línea] 2007. [Citado el: 12 de enero de 2015.]
<http://definicionabc.com/comunicacion/noticia.php>.

Becerra, Tristán. 2011. Revisiones de código y estándares de codificación. *Revisiones de código y estándares de codificación*. [En línea] 2011. [Citado el: 6 de mayo de 2015.] <https://msdn.microsoft.com/es-es/library/aa291591%28v=vs.71%29.aspx>.

Bello, A. 2011. Notadiario. *Notadiario*. [En línea] 14 de diciembre de 2011. [Citado el: 20 de abril de 2015.]
<http://www.notadiario.com/sci-tech/ventajas-y-desventajas-de-un-android/>.

Bordallo, Jose Carlos. 2013. Telefonía móvil, historia, ventajas y desventajas. *Telefonía móvil, historia, ventajas y desventajas*. [En línea] 24 de noviembre de 2013. [Citado el: 2 de abril de 2015.]
<http://es.slideshare.net/TheBestGamer1/telefona-mvil-historia-ventajas-y-desventajas>.

Referencias bibliográficas

- Brooks, Frederick Phillips. 2011.** Prácticas de Software. *Prácticas de Software*. [En línea] 2011. [Citado el: 11 de febrero de 2013.] www.practicadesoftware.com.ar.
- Developers, Android. 2014.** Android developers. [En línea] 2014. [Citado el: 10 de mayo de 2015.] <http://developer.android.com..>
- Fernández Luna, Juan Manuel. 2006.** Tipos de dispositivos móviles. *Tipos de dispositivos móviles*. [En línea] septiembre de 2006. [Citado el: 20 de abril de 2015.] http://leo.ugr.es/J2ME/INTRO/intro_4.htm.
- Foundation, Eclipse. 2014.** Eclipse Foundation. [En línea] 2014. www.eclipse.org.
- Fowler. 2007.** *UML distilled, A brief guide to standard object modeling lenguaje*. Tercera. s.l. : Pearson Education, 2007. ISBN: 978-8131-715-65-9.
- García, Alberto. 2005.** *La implantación técnica de la televisión digital terrestre en España. Modelos Europeos*. Madrid : s.n., 2005.
- García, David Luis. 2011.** *Implementación de Módulo de Gestión de Medias de la Plataforma de Televisión Informativa PRIMICIA v2.0*. Universidad de las Ciencias Informáticas. La Habana : s.n., 2011.
- Graiana. 2013.** Concepto y características de la noticia. [En línea] 14 de julio de 2013. [Citado el: 12 de enero de 2015.] <http://es.slideshare.net/graiana12/concepto-y-caractersticas-de-la-noticia>.
- Gutiérrez, Demián. 2009.** UML Diagrama de Paquetes. [En línea] 2009. [Citado el: 14 de Abril de 2014.] http://www.codecompiling.net/files/slides/UML_clase_05_UML_paquetes.pdf.
- INEWS. 2013.** *INEWS*. [En línea] 2013. [Citado el: 26 de abril de 2015.] <https://www.avid.com/es/products/inews>.
- Jacobson, Ivar, Booch, G. y Rumbaugh, J. 2000.** *El proceso unificado de desarrollo de software*. Madrid: Pearson Educación, 2000. ISBN: 84-7829-036-2.
- Larman, Craig. 1999.** *UML y Patrones. Introducción al análisis y diseño orientado a objeto*. México: PRENTICE HALL, 1999. ISBN 0-13-748880-7.

Referencias bibliográficas

- MasterMagazine. 2010.** Definición de Aplicación. [En línea] 2010. <http://www.mastermagazine.info/termino/3874.php>.
- Microsoft. 2013.** Diagramas de componentes de UML. [En línea] 2013. [Citado el: 20 de abril de 2015.] <https://msdn.microsoft.com/es-es/library/dd409390.aspx>.
- Montes de Oca, Dorgis. 2012.** *Desarrollo del módulo de Redacción de Noticias para la Plataforma de Televisión Informativa PRIMICIA v2.0*. La Habana: s.n., 2012.
- Pressman, Roger S. 2008.** *Ingeniería de Software: un enfoque práctico*. s.l.: Mc Graw Hill, 2008. ISBN: 970-10-5473-3.
- Rivera, Eduardo. 2014.** Arquitectura de Software II - Diagrama de Componentes y Despliegue. [En línea] 2014. [Citado el: 30 de Abril de 2014.] <http://http://es.scribd.com/doc/7884665/Arquitectura-de-Software-II-Diagrama-de-Componentes-y-Despliegue>.
- Romero, Felix Ivan. 2011.** *Implementación del Módulo de Redacción de la Plataforma de Televisión Informativa PRIMICIA*. Universidad de las Ciencias Informáticas. La Habana : s.n., 2011.
- Tomás, Jesús. 2013.** Diploma de Especialización en desarrollo de aplicaciones para Android. [En línea] Copyright, 2013. [Citado el: 151 de marzo de 2015.] <http://www.androidcurso.com/index.php/tutoriales-android-fundamentos/31-unidad-1-vision-general-y-entorno-de-desarrollo/98-comparativa-con-otras-plataformas>.
- Velásquez, Pastrana, Lourdes. 2013.** Generaciones en la Telefonía Celular. [En línea] 2013. [Citado el: 25 de octubre de 2014.] <http://www.enterate.unam.mx/Articulos/2013/agosto/telefcel.htm>.
- Visual Parading Group. 2006.** *Reasons to Choose Visual Paradigm*. 2006.
- Wilder, D F. 2009.** Diagrama de Componentes. [En línea] 19 de mayo de 2009. [Citado el: 5 de abril de 2015.] http://diagramacomponente.blogspot.com/2009/05/definicion_18.html.

Anexos

Entrevista realizada al especialista y jefe de proyecto Ing. Felix Ivan Romero Rodríguez, sobre características y cualidades del subsistema de Administración web PRIMICIA.

Desarrollo de la entrevista:

- Saludos y comunicación del objetivo que persigue el entrevistador.
- Presentaciones individuales de los entrevistados.
- Aspectos a abordar en la entrevista:
 - Pregunta #1: ¿La aplicación tendrá la misma estructura que el subsistema actual?
 - Pregunta #2: ¿Las funcionalidades a implementar se corresponden en su totalidad con las especificadas en los módulos de redacción de noticias y gestión de medias de PRIMICIA?
 - Pregunta #3: ¿Cómo debe ser la gestión de las noticias?
 - Pregunta #4: ¿Cuáles son los datos que se deben visualizar al listar las noticias?
 - Pregunta #5: ¿Cuáles son las opciones que deben brindar las pantallas?
 - Pregunta #6: ¿Cuáles son las opciones de configuración para los componentes de las noticias?
 - Pregunta #7: ¿Cómo debe ser la gestión de medias?
 - Pregunta #8: ¿Cuáles son los tipos de medias que serán utilizados?
 - Pregunta #9: ¿Cómo será el envío de los datos desde la aplicación hasta guardar los datos en el servidor de PRIMICIA?