



Universidad de las Ciencias Informáticas

Facultad 1

Lector de noticias para dispositivos móviles con Sistema Operativo NovaDroid.

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

Autor: Elioenai Reyes González

Tutores: Mtr. Yoandy Pérez Villazón
Ing. Ivelisse Montero Jimenez

La Habana, junio de 2017
"Año 59 de la Revolución"

DECLARACIÓN DE AUTORÍA

Declaro por este medio que yo Elioenai Reyes González, con carné de identidad 93060105623 soy el autor principal del trabajo titulado ‘Lector de noticias para dispositivos móviles con Sistema Operativo NovaDroid’ y autorizo a la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio, así como los derechos patrimoniales con carácter exclusivo.

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

Elioenai Reyes González
Firma del autor

Mtr. Yoandy Pérez Villazón
Firma del tutor

Ing. Ivelisse Montero Jimenez
Firma del cotutor

DATOS DE CONTACTO

Tutores: Mtr. Yoandy Pérez Villazón e Ing. Ivelisse Montero Jimenez

Formación académica: Ingeniero en Ciencias Informáticas, Universidad de Ciencias Informáticas, 2017

Correo electrónico: ergonzalez@estudiantes.uci.cu

DEDICATORIA

Dedicada a Juan Carlos Reyes Guzmán (papá), Elaine González Jardines (mamá) y Joel Reyes Guzmán (Tío).

AGRADECIMIENTOS

Gracias a Dios y las personas que estuvieron presente en cada paso, ellas saben quiénes son.

RESUMEN

Las noticias, independientemente del tema, del día o del lugar en que se vive, tienen una gran importancia para la vida de las personas; se dan por el conjunto de medios de comunicación que informan sobre los sucesos más destacados. A partir de la evolución de las nuevas tecnologías se modificó la manera de acceder y leer las noticias con respecto a medios tradicionales, en la actualidad las noticias pueden presentarse en diversas maneras, dígase noticias textuales, imágenes, videos, entre otras. Las noticias al ser publicadas en sitios web posibilitan que las personas puedan acceder desde cualquier lugar de manera inmediata. Esta investigación tiene como objetivo, a través del uso de las tecnologías móviles, desarrollar un lector de noticias para dispositivos móviles que permita mostrar las noticias que se publiquen en cualquier momento. Para guiar el proceso de desarrollo de software se utilizó como metodología AUP en su variación para la Universidad de las Ciencias Informáticas, para la implementación se utilizó como lenguaje de programación Java en conjunto con el entorno de desarrollo integrado Android Studio y se utilizó la librería Jsoup para la captura de las publicaciones. La aplicación de las pruebas de funcionalidad, compatibilidad y de unidad demostraron el correcto funcionamiento del sistema en correspondencia con los requerimientos del cliente.

Palabras clave:

lector, noticias, sitio web, tecnologías móviles, metodología.

TABLA DE CONTENIDOS

| | |
|--|-----------|
| INTRODUCCIÓN..... | 1 |
| CAPÍTULO 1. PRINCIPALES REFERENTES TEÓRICOS QUE SUSTENTAN LA ELABORACIÓN DE LECTOR DE NOTICIAS PARA DISPOSITIVOS CON SISTEMA OPERATIVO NOVADROID..... | 7 |
| 1.1. TIC E INTERNET SOBRE EL ENTORNO CUBANO | 7 |
| 1.2. PRINCIPALES CONCEPTOS Y DEFINICIONES ASOCIADOS AL TEMA | 10 |
| 1.3. ESTUDIO DE SISTEMAS HOMÓLOGOS | 14 |
| 1.4. HERRAMIENTAS Y TECNOLOGÍAS..... | 16 |
| CAPÍTULO 2. ANÁLISIS Y DISEÑO DE LA PROPUESTA DE SOLUCIÓN..... | 22 |
| 2.1. ANÁLISIS | 22 |
| 2.2. PROPUESTA DE SOLUCIÓN..... | 22 |
| 2.3. DIAGRAMA DE DOMINIO..... | 22 |
| 2.4. REQUISITOS FUNCIONALES DEL SISTEMA | 24 |
| 2.5. REQUISITOS NO FUNCIONALES DEL SISTEMA..... | 27 |
| 2.6. HISTORIAS DE USUARIO (HU) | 28 |
| 2.7. ARQUITECTURA DE SOFTWARE | 31 |
| 2.8. PATRONES DE DISEÑO..... | 31 |
| 2.9. DIAGRAMAS DE CLASE DEL DISEÑO | 35 |
| 2.10. MODELO DE DATOS | 37 |
| 2.11. MODELO DE DESPLIEGUE | 37 |
| CAPÍTULO 3. IMPLEMENTACIÓN Y VALIDACIÓN DEL LECTOR DE NOTICIAS PARA DISPOSITIVOS MÓVILES CON SISTEMA OPERATIVO NOVADROID..... | 38 |
| 3.1. DIAGRAMA DE PAQUETES | 42 |
| 3.2. ESTÁNDARES DE CODIFICACIÓN | 38 |
| 3.3. PRUEBAS DE SOFTWARE..... | 41 |
| CONCLUSIONES..... | 49 |
| RECOMENDACIONES..... | 50 |
| REFERENCIAS BIBLIOGRÁFICAS..... | 51 |

| | |
|-----------------------------------|-----------|
| ANEXOS | 56 |
| ANEXO 1 HISTORIAS DE USUARIO..... | 56 |
| ANEXO 2 ENTREVISTA | 66 |
| GLOSARIO DE TÉRMINO | 68 |

ÍNDICE DE TABLAS

| | |
|--|----|
| Tabla 1: indicadores físicos de las TIC en Cuba (ONE, 2015)..... | 8 |
| Tabla 2: ventajas obtenidas por los consumidores de Internet (El empleo de Internet como nuevo canal de distribución: un análisis de sus principales ventajas e inconvenientes, 2003). | 9 |
| Tabla 3: comparación entre aplicaciones homólogas. | 15 |
| Tabla 4: diferencias entre las fases de AUP y AUP-UCI (Informáticas, 2015). | 17 |
| Tabla 5: requisitos funcionales. | 24 |
| Tabla 6: historia de Usuario # 4. | 28 |
| Tabla 7: historias de usuario # 7. | 29 |
| Tabla 8: variables del caso de prueba para el escenario “adicionar sitio”..... | 46 |
| Tabla 9: casos de prueba Introducir “Adicionar sitio”: Fuente de elaboración propia..... | 47 |

INTRODUCCIÓN

En los últimos tiempos se ha evidenciado un desarrollo de las Tecnologías de la Información y las Comunicaciones (TIC) (ITU, 2014). Con la evolución de las TIC, internet pasó de ser un instrumento especializado de la comunidad científica y militar a ser una red de fácil uso que modificó las pautas de interacción social. Internet, o red de redes, consiste en una red informática mundial, descentralizada, formada por la conexión directa entre computadoras mediante un protocolo especial de comunicación (Real Academia Española, 2017). Pero sería un error considerar internet únicamente como una red de computadoras. Se puede considerar las computadoras simplemente como el medio que transporta la información. En este caso internet sería una gran fuente de información práctica (Elena Barragans, 2006).

Con internet, los sistemas de medios de comunicación masivos han sufrido cambios debido al desarrollo y difusión de nuevas tecnologías de la información. Partiendo de la necesidad de las personas de mantenerse informados respecto a disímiles temas de interés, puede afirmarse entonces que internet ha ocupado un papel decisivo en este marco (Simón Peña Fernández et al., 2014), garantizando y facilitando la rápida actualización y acceso a la información. Para ello, se han desarrollado sitios web con el objetivo de difundir información de interés para el público receptor, ya sea a través de noticias textuales, imágenes, videos, entre otros.

Como parte de la evolución de la tecnología, ha tenido lugar un auge de los dispositivos móviles (Fernando Rivero, 2016). 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. De acuerdo con esta definición existen multitud de dispositivos móviles, desde los reproductores de audio portátiles hasta los navegadores GPS¹, pasando por los teléfonos móviles, los PDA² o los Tablet PC³ (Arturo Baz Alonso et al. 2006). Según el IDC⁴, empresa de análisis de mercado, los SO⁵ más populares en esta esfera son Android, Windows Phone e iOS, destacando Android que

¹ Siglas en inglés para: Global Positioning System.

² Siglas en inglés para: Personal Digital Assistant.

³ Siglas en inglés para: Personal Computer.

⁴ Siglas en inglés para: International Data Corporation.

⁵ Siglas en español para: Sistema Operativo.

domina el mercado con un **63.66%** de dispositivos activos que ejecutan este SO según Operating System Market Share (Operating system market share, 2017).

En la medida que aumenta el uso de las tecnologías móviles en torno al desarrollo de aplicaciones, las mismas se han ido adaptando cada vez más a las diversas necesidades o requerimientos para miles de usuarios. De ahí que surjan aplicaciones enfocadas a extraer las más recientes publicaciones de los sitios web de interés para los usuarios.

Una de las principales formas de actualizar las recientes publicaciones es la utilización de *Web Feed* o Fuente Web. Esta última se puede definir como un archivo generado por algunos sitios webs (portales, *blogs*, diarios de noticias, entre otros) que contiene una versión específica de la información publicada de estos (Elías Fernández Martín, 2006). *Web Feed* se utiliza para suministrar información actualizada frecuentemente a sus suscriptores. Cuando una página web "redifunde"⁶ su contenido mediante una Fuente Web, los internautas pueden estar informados de sus novedades haciendo uso de un programa "agregador"⁷ para acceder a sus fuentes suscritas desde un mismo lugar sin necesidad de tener que entrar a la página web. *Web Feed* es muy utilizado en los *blogs* o bitácoras de internet, así como en prensa electrónica. Cada día hay más medios que utilizan este sistema de redifusión web. La gran difusión de este tipo de servicio web ha estimulado el interés en otros sectores que comienzan a ver en las fuentes web una forma efectiva de difundir y compartir información (Microsoft, 2017).

A nivel mundial existen diversas aplicaciones que hacen uso del *Web Feed*. En su gran mayoría, son aplicaciones que tienen un coste monetario o las versiones gratis poseen "publicidad no deseada"⁸ y funcionalidades limitadas. Otra desventaja de estas soluciones es que no están pensadas para las particularidades del entorno cubano, sino para entornos con acceso constante a internet. Cuba es un país con limitaciones de conectividad a internet. Solo en determinadas entidades del país está disponible el servicio de acceso a internet; y aunque fueron concebidos los denominados "parques wifi"⁹, no garantizan que las personas puedan estar todo el tiempo conectadas a internet.

⁶ Reenvío o reemisión de contenidos desde una fuente original.

⁷ Tipo de aplicación que permite suscribirse a fuentes web para recibir información actualizada de un sitio web.

⁸ Publicidad que aparece en aplicaciones no deseadas por el usuario y que en algunos casos puede llegar a enviar información privada del usuario al exterior, es decir, viola claramente la privacidad del usuario.

⁹ Zonas con señal Wifi para garantizar el acceso masivo a Internet.

La Universidad de las Ciencias Informáticas (UCI) es un centro docente-productivo líder en el proceso de informatización que lleva a cabo el país; proceso que aboga por los principios de soberanía tecnológica y la socio-adaptabilidad. La UCI desde el año 2010 posee una alianza estratégica con la empresa GEDEME¹⁰ para el desarrollo de soluciones conjuntas de apoyo a la informatización de la sociedad. Como parte del trabajo conjunto se desarrolla y establece en las tabletas que comercializa GEDEME el Sistema Operativo NovaDroid siendo este una variante del Sistema Operativo Android.

NovaDroid se caracteriza por adaptarse a las condiciones tecnológicas existentes en Cuba, uno de los elementos en los que se ha trabajado es lograr un máximo aprovechamiento de la tableta añadiéndole aplicaciones que no requieran conectividad para su funcionamiento. Ejemplo de este tipo de aplicaciones es Osmand para la geolocalización de lugares, así como la inclusión de un cliente de correo que permite revisar y leer correos previamente descargados, entre otros.

La gran mayoría de la población cubana tiende a mantenerse informado a través de los medios de prensa, ejemplo de ello es Cubadebate¹¹, que en 2016 recibió 30.240.466 visitas (Cubadebate, 2016). El avance en la informatización, a partir de la introducción de dispositivos móviles, ofrece una vía para que los usuarios puedan contar con acceso a las fuentes de noticias con mayor inmediatez. Al tener como referente la existencia de soluciones que permiten la captura en línea y visualización de noticias, los usuarios de NovaDroid han notificado al proveedor GEDEME su interés en que este Sistema Operativo incluya un mecanismo que les posibilite obtener las noticias de diversas fuentes para su posterior lectura, aun cuando los dispositivos no se encuentren conectados a internet.

Teniendo en cuenta lo expresado anteriormente, se plantea como **problema de la investigación**: ¿Cómo garantizar la disponibilidad de acceso a los medios de noticias desde NovaDroid en entornos de baja conectividad a internet para la mejora de la consulta de noticias? Por tanto, se define como **objeto de estudio**: la captura de noticias desde dispositivos móviles.

Para dar solución al problema de investigación planteado se define como **objetivo general**: desarrollar un lector de noticias para dispositivos móviles con Sistema Operativo NovaDroid que permita obtener, consultar y leer las noticias en entornos de baja conectividad a internet.

¹⁰ Empresa Industrial para la Informática, las Comunicaciones y la Electrónica.

¹¹ Sitio cubano, medio de información alternativa que alerta sobre campañas de difamación contra Cuba. Publica en exclusiva las Reflexiones de Fidel, también noticias en general.

El **campo de acción** está enmarcado en la captura de noticias desde dispositivos móviles con Sistema Operativo NovaDroid.

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

1. ¿Cuáles son los referentes teóricos fundamentales que sustentan la investigación relacionados con el desarrollo de herramientas para la implementación de un lector de noticias para dispositivos móviles con Sistema Operativo NovaDroid?
2. ¿Qué fundamentos presenta el análisis y diseño del lector de noticias para dispositivos con Sistema Operativo NovaDroid?
3. ¿Qué características debe tener la implementación del lector de noticias para dispositivos con Sistema Operativo NovaDroid?
4. ¿Cuáles son las pruebas de software que se deben realizar al lector de noticias para dispositivos con Sistema Operativo NovaDroid?

Objetivos específicos:

1. Caracterizar los referentes teóricos fundamentales que sustentan la investigación relacionados con el desarrollo de herramientas para la implementación de un lector de noticias para dispositivos con Sistema Operativo NovaDroid.
2. Diseñar las funcionalidades del lector de noticias para dispositivos con Sistema Operativo NovaDroid.
3. Implementar las funcionalidades del lector de noticias para dispositivos con Sistema Operativo NovaDroid.
4. Realizar pruebas de software a las funcionalidades del lector de noticias para dispositivos con Sistema Operativo NovaDroid.

Para alcanzar dichos objetivos se plantea desarrollar las siguientes **tareas de investigación**:

1. Realización del estudio del arte para documentar los diferentes sistemas que utilizan *Web Feed* para la actualización de publicaciones.
2. Selección de las tecnologías, herramientas y estándares que se necesitan para implementar el lector de noticias mediante una aplicación para dispositivos con Sistema Operativo NovaDroid.

3. Selección de la metodología de desarrollo de software para guiar el desarrollo del sistema de lector de noticias para consultar noticias de Cuba y el mundo mediante una aplicación para dispositivos con Sistema Operativo NovaDroid.
4. Definición y especificación de los requisitos del software para adecuar el desarrollo del sistema a las exigencias del cliente.
5. Descripción de la arquitectura para el desarrollo más adecuado del lector de noticias para consultar noticias de Cuba y el mundo mediante una aplicación para dispositivos con Sistema Operativo NovaDroid.
6. Implementación de los requisitos especificados en la programación del lector de noticias para consultar noticias de Cuba y el mundo mediante una aplicación para dispositivos con Sistema Operativo NovaDroid.
7. Definición de las pruebas a realizar al lector de noticias para dispositivos con Sistema Operativo NovaDroid para validar su funcionamiento.
8. Aplicación de las pruebas de software a la solución implementada mediante pruebas de funcionalidad y verificación de los resultados.

Al concluir con la presente investigación se espera obtener como **posibles resultados**:

1. Marco teórico referencial sobre la implementación de un sistema de lector de noticias para consultar noticias de Cuba y del mundo mediante una aplicación para dispositivos con Sistema Operativo NovaDroid.
2. Una aplicación para dispositivos con Sistema Operativo NovaDroid que permita a los usuarios consultar publicaciones.

Para el desarrollo de la presente investigación se emplearon los siguientes métodos científicos.

Métodos a nivel teórico:

- **Analítico – sintético:** permite seleccionar y resumir elementos relevantes que se relacionan con lectores de noticias con el uso de *Web Feed*.
- **Análisis Histórico-Lógico:** se utiliza para el estudio del estado del arte de los sistemas lectores de *Web Feed*, destacando las ventajas y desventajas de las mismas.

Métodos empíricos:

- **Entrevista:** a través de una serie de preguntas al cliente, permite obtener la información necesaria para determinar características, cualidades y requisitos que debe tener el lector de noticias sobre Cuba y el mundo para dispositivos móviles con Sistema Operativo NovaDroid.
- **Revisión documental:** permite analizar documentación acerca de otros lectores de noticias utilizando *Web Feed* para obtener experiencia y determinar ciertas características que deben poseer este tipo de sistemas.

El presente documento de trabajo de diploma está compuesto por introducción, desarrollo dividido en tres (3) capítulos, conclusiones generales, recomendaciones y referencias bibliográficas de las citas utilizadas a lo largo de la investigación. Para complementar la investigación, finalmente se presentan una serie de anexos.

El **capítulo 1** titulado “Principales referentes teóricos que sustentan la elaboración de un lector de noticias para consultar noticias de Cuba y el mundo para dispositivos con Sistema Operativo NovaDroid” aborda los principales aspectos teóricos que sustentan la investigación, se definen conceptos fundamentales, se realiza un estudio del arte acerca de sistemas similares y finalmente se selecciona la metodología, herramientas y tecnologías a utilizar a lo largo de toda la investigación.

El **capítulo 2** titulado “Análisis y Diseño de la propuesta de solución” presenta todo lo referente al análisis y diseño del lector de noticias para dispositivos con Sistema Operativo NovaDroid. En este capítulo se presentan una serie de diagramas, se definen las funcionalidades que debe poseer, arquitectura del sistema, entre otros elementos de análisis y diseño que permite una mejor comprensión del sistema a implementar.

El **capítulo 3** titulado “Implementación y validación del lector de noticias para dispositivos móviles con Sistema Operativo NovaDroid” expone los estándares de codificación, algoritmos complejos utilizados, y otros elementos de implementación. Además, se realizan una serie de pruebas al sistema para comprobar que el objetivo planteado para la presente investigación fue cumplido.

CAPÍTULO 1. PRINCIPALES REFERENTES TEÓRICOS QUE SUSTENTAN LA ELABORACIÓN DE LECTOR DE NOTICIAS PARA DISPOSITIVOS CON SISTEMA OPERATIVO NOVADROID

En el presente capítulo se relacionan los elementos teóricos fundamentales que sustentan a la investigación. Primeramente, se realiza un análisis acerca de lo que a TIC e internet se refiere, para luego exponer elementos importantes acerca de la actualización de noticias por internet haciendo uso de los canales *Web Feed*, específicamente enfocado hacia dispositivos con Sistema Operativo NovaDroid.

1.1. TIC e internet sobre el entorno cubano

Se entiende por TIC al “conjunto de herramientas aplicadas principalmente a la gestión de la información, que abarcan un abanico muy amplio de soluciones” (Corrales, 2009).

A través de la utilización de diferentes medios tecnológicos en distintos formatos (audio, texto, imagen, video, entre otros), las TIC posibilitan a las personas la adquisición, producción, almacenamiento, tratamiento, comunicación, registro y presentación de información. Con este término se hace referencia a programas informáticos, multimedia, videojuegos, entre otros, pero sobre todo a los grandes avances alcanzados por la poderosa red de redes.

En las últimas décadas las TIC han evolucionado rápidamente, por lo que ha llegado a alcanzar un elevado auge en el desarrollo de la sociedad, a continuación se relacionan sus principales ventajas (Huidobro, 2006):

- Brindar grandes beneficios y adelantos a la salud y la educación.
- Potenciar a las personas y actores sociales de redes de apoyo e intercambio y listas de discusión.
- Apoyar a personas y empresas a presentar y vender sus productos a través de la internet.
- Impartir nuevos conocimientos para la empleabilidad que requieren muchas competencias (integración, trabajo en equipo, motivación, disciplina, entre otros).
- Ofrecer nuevas formas de trabajo, como teletrabajo.
- Brindar facilidades, exactitud, menores riesgos y costos.
- Dar acceso al flujo de conocimientos e información para empoderar y mejorar las vidas de las personas.

- Permitir el aprendizaje interactivo y la educación a distancia.

Como puede constatarse, las ventajas anteriores están de una forma u otra estrechamente vinculadas al uso de la internet. Cuba es un país que presenta una gran obsolescencia tecnológica, donde además el acceso a internet es limitado, pero aun así las TIC desempeñan un gran papel en la informatización de la sociedad. La siguiente tabla muestra los indicadores físicos en lo que a TIC se refiere en Cuba en los últimos años.

Tabla 1: indicadores físicos de las TIC en Cuba (ONE, 2015).

| Concepto | UM | 2009 | 2010 | 2011 | 2012 | 2013 | 2014 |
|--|----|---------|---------|---------|---------|---------|---------|
| Cantidad de computadoras existentes. | M | 700,0 | 724,0 | 783,0 | 834,0 | 1 014,4 | 1 067,4 |
| De ellas en red. | M | 455,0 | 434,4 | 469,8 | 500,4 | 515,4 | 533,9 |
| Cantidad de usuarios con servicios de internet. | M | 1 600,0 | 1 790,0 | 2 610,0 | 2 871,0 | 2 923,0 | 3 048,5 |
| Computadoras personales por 1000 habitantes. | U | 62 | 64 | 70 | 74 | 90 | 95 |
| Usuarios de internet por 1000 habitantes. | U | 142 | 159 | 232 | 257 | 261 | 271 |
| Dominios registrados bajo (.cu) | U | 142 | 159 | 232 | 257 | 261 | 271 |
| Total de abonados del sistema celular. | M | 752,8 | 1 127,9 | 1 431,5 | 1 792,3 | 2 104,6 | 2 636,7 |
| Abonados móviles del sistema celular. | M | 621,2 | 1 003,0 | 1 315,1 | 1 681,6 | 1 995,7 | 2 530,8 |
| Abonados fijos del sistema celular. | M | 131,6 | 124,9 | 116,4 | 110,7 | 108,9 | 105,9 |
| Cobertura de la población de celular móvil. | % | 79,3 | 82,4 | 83,7 | 85,3 | 85,3 | 85,3 |

Como puede observarse en la tabla anterior el uso de internet en Cuba es limitado, lo cual dificulta la informatización de la sociedad cubana debido a que la internet es un medio que interactúa con el conjunto de la sociedad, es mucho más que una tecnología, pues se trata de un medio de comunicación, de interacción y de organización social (Internet y la sociedad red., 2011). Para un mejor entendimiento acerca de las ventajas que trae consigo la consulta de información en internet se puede observar la siguiente tabla:

Tabla 2: ventajas obtenidas por los consumidores de internet (El empleo de Internet como nuevo canal de distribución: un análisis de sus principales ventajas e inconvenientes, 2003).

Fuente: elaboración propia.

| Ventajas | Principales argumentaciones |
|--|---|
| Disponibilidad de gran cantidad de información. | La información disponible está alcanzando cotas inimaginables en los mercados tradicionales. Relaciona esta ventaja con la rapidez de obtención de la información, así como la inmediatez y disponibilidad completa del medio digital. |
| Mayor facilidad para realizar comparaciones. | Otorgan especial relevancia a las comparaciones. |
| Eliminación de asimetrías en la información. | En los mercados electrónicos, a diferencia de lo tradicionalmente ocurrido en los mercados físicos, las diferencias informativas entre vendedores y compradores son mínimas porque el acceso a la información acerca del mercado y los competidores es ahora global y accesible para todas las partes de cualquier negocio. |
| Reducción de costes. | Decrementos en los niveles de costes en los procesos de búsqueda de información. |

Como pudo observarse en la tabla anterior, con el uso de la internet se propicia la disponibilidad de información, esto permite tener acceso en cualquier momento a cualquier información rompiendo las barreras geográficas que puedan existir, siendo este un aspecto fundamental para que a través de los distintos medios se propicie la distribución de noticias. Los cambios en el flujo informativo se concretan en cuatro aspectos esenciales: la aceleración en la difusión de noticias, el aumento de la flexibilidad en su

distribución a través de múltiples plataformas, la abundancia comunicativa con la consiguiente fragmentación de la audiencia y la expansión global de la información (Feenstra, y otros, 2012).

Internet ha incrementado extraordinariamente la rapidez de circulación de las noticias, que se diseminan instantáneamente y de forma viral a través las redes sociales, como *Twitter*. En este contexto, la cobertura informativa se produce en tiempo real y los medios digitales actúan con una lógica de actualización constante de las noticias instaurando un ciclo de noticias ininterrumpido. Por otra parte, también se registra una mayor flexibilidad en la difusión de noticias. La información deja de estar circunscrita a un único soporte y circula por múltiples vías a partir de una distribución multiplataforma. Como consecuencia, se configuran diversos y variados mercados de noticias más allá de los medios convencionales como los *blogs*, las redes sociales, los medios alternativos o el *microblogging*¹² (Feenstra, y otros, 2012).

1.2. Principales conceptos y definiciones asociados al tema

1.2.1. *Sindicación de contenidos* (Sánchez Tarragó, 2007)

Una de las tecnologías actuales, capaz de personalizar y agregar valor a los servicios en internet, es la llamada sindicación de contenidos por medio de Fuente Web.

A continuación, se presentará un grupo de aspectos generales sobre la sindicación de contenidos y Fuente Web, y se ejemplificará su utilización actual en diversos entornos, entre ellos, las instituciones de información. Finalmente, se expondrán algunas tendencias globales.

La palabra syndicar, o asociar contenidos, procede del inglés *To syndicate* en inglés significa: "publicar artículos simultáneamente en diferentes medios mediante una fuente". Es el proceso mediante el cual un productor, o un distribuidor de contenidos, proporcionan información en formato digital a un suscriptor o a una red de suscriptores, generalmente con el ánimo de que estos los integren en sus propios sitios web.

La sindicación de contenidos permite a un sitio web, principalmente a aquellos que tienen actualizaciones frecuentes, informar a los interesados sobre las novedades del sitio, típicamente por medio de la distribución de titulares de noticias y contenidos breves. Cuando el usuario observa un elemento de interés en su canal, puede seguir el enlace para obtener el contenido completo.

¹² Servicio que permite a sus usuarios enviar y publicar mensajes breves, generalmente solo de texto.

Por tanto, la sindicación de contenidos constituye una opción para evitar navegar de una página en otra en busca de información de interés. De esta forma se convierte en un medio de comunicación a la carta, integrado por suscripciones de todo tipo de fuentes y ordenado según el criterio de cada usuario. Es una forma de personalizar los contenidos que ofrecen las publicaciones electrónicas, portales y sitios web. Un servicio de valor añadido con respecto a los mecanismos más tradicionales de entrega de contenidos electrónicos. La sindicación de contenidos constituye una nueva manera de llegar a una audiencia cada vez más segmentada (Eva Domínguez, 2005).

1.2.2. *Web Feed*

Una Fuente Web, o *Web Feed*, es un documento generalmente basado en *XML*¹³ el cual contiene referencias electrónicas a versiones extendidas. La prensa electrónica y *blogs* o bitácoras recurren comúnmente a ellos, aunque las fuentes web también son empleadas para entregar información estructurada con respecto al tiempo meteorológico (Web Feed, 2017).

Los dos principales formatos son *RSS*¹⁴ y *Atom*. A menudo el término *RSS* (*Really Simple Syndication*) se usa erróneamente para referirse a Fuente Web, independientemente de que el formato de dicha Fuente Web sea *RSS* o *Atom* (Microsoft, 2017).

Las fuentes web han sido diseñadas para ser entendidas fácilmente por máquinas más que entendidas fácilmente por humanos, lo cual llega a crear confusión la primera vez que una persona se enfrenta a ellos. Esto significa que una Fuente Web puede ser empleada para transferir información automáticamente desde una página web a otra, sin ninguna intervención humana. Son utilizadas en los blogs, o bitácoras, de internet, así como en prensa electrónica. Cada día hay más medios que utilizan este sistema de redifusión web. La gran difusión de este tipo de servicio web ha estimulado el interés en otros sectores que comienzan a ver en las fuentes web una forma efectiva de difundir y compartir información empresarial (Microsoft, 2017), (Elías Fernández Martín, 2006).

¹³ Siglas en inglés para: eXtensible Markup Language.

¹⁴ Siglas en inglés para: Really Simple Syndication.

Ventajas del uso del *Web Feed*:

- Mantenerse al día con respecto a información deseada puede llegar a ser difícil. Mediante las fuentes web se pueden obtener las últimas noticias al respecto en cuanto ésta sea actualizada, sin la necesidad de ir buscando de sitio en sitio.
- Gran ahorro de tiempo para los visitantes ya que es posible acceder rápidamente a todos los contenidos nuevos publicados en varios sitios, sin tener que visitarlos uno por uno.
- Búsqueda de titulares de distintos sitios en un solo lugar.
- A diferencia de las notificaciones vía correo electrónico, mediante las fuentes web no existen direcciones electrónicas involucradas; así se evita publicidad, Spam, Virus, entre otros.
- Se puede cancelar la suscripción a una Fuente Web sin necesidad de aviso.

1.2.3. RSS

Los denominados RSS hacen referencia a ficheros XML que permiten compartir contenidos en la web. RSS es un acrónimo que puede representar varias entidades: Rich Site Summary, RDF Site Summary o Really *Simple Syndication*. Por este motivo, muchos autores prefieren llamarlo simplemente RSS. Una definición sencilla de RSS la ofrece D'Souza: es un formato de XML (*Extensible Markup Language*), que utiliza el modelo de metadatos RDF (*Resource Description Framework*) para representar información sobre los recursos en el web (Sánchez Tarragó, 2007).

Según algunos autores, RSS se ha convertido en el formato XML más popular actualmente. Sus aplicaciones son múltiples y miles de sitios web utilizan hoy el RSS como un mecanismo para atraer usuarios.

Puede observarse un incremento sostenido en el número de canales de RSS desde septiembre de 2001 hasta solamente mayo de 2006 como establece Syndic8, uno de los directorios de RSS más grandes en internet (Ilustración 1).

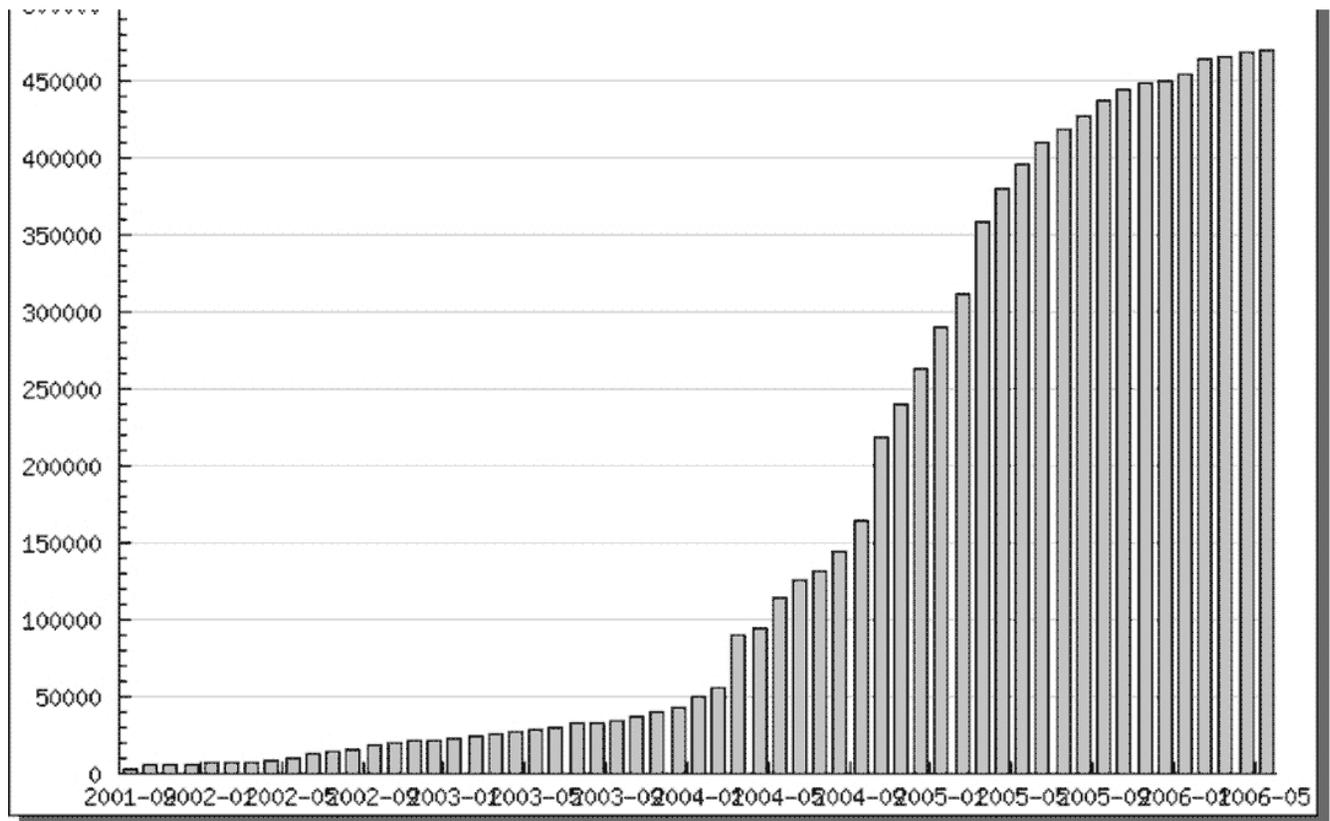


Ilustración 1: crecimiento del número de canales RSS, 2001- mayo 2006

Fuente: Syndic8. Statistics. 2006.

1.2.4. Atom

Atom es un formato de documento basado en XML que describe listas de información relacionada conocidas como "Web Feed". El principal caso de uso que *Atom* aborda es la sindicación de contenido web.

Los formatos RSS y *Atom* cubren sobre el **99%** de los *Web Feeds*. Alrededor del **75%** son RSS y el **25%** son *Atom* (Chris Wellons, 2013).

1.3. Estudio de sistemas homólogos

Aplicaciones (Joe Hindy, 2016), (Google Play, 2017)

- **Feedly:** es un lector de RSS que permite organizar y acceder rápidamente desde un navegador web o de sus aplicaciones para teléfonos inteligentes a todas las noticias y actualizaciones de *blogs* y demás páginas que el sistema soporta. Entre otras características permite ordenar todos los contenidos de manera que facilita al usuario ahorrar tiempo por no tener que revisar una a una todas las fuentes de noticias. Es necesario de poseer una previa cuenta Feedly o en Google para el inicio de sección. Actualiza demasiado por lo que genera cargas innecesarias en el dispositivo. Es gratuita.
- **Flipboard:** es una aplicación para tabletas y *Smartphone* que sirve como una revista digital que se alimenta del contenido elegido por el usuario. Primero se deben elegir temas generales que gusten, se arma el Flipboard y se puede comenzar a leer el contenido. También provee una "edición diaria" llena de noticias. Es mejor utilizar en tabletas, pero se ve muy bien en los teléfonos también. Es gratuita.
- **gReader:** es un buen lector de RSS antiguo. Tiene un interfaz moderno con la integración de *Feedly*. Viene con sincronización bidireccional, lectura sin conexión, soporte para *podcasts*, una variedad de temas, modo nocturno y varias otras personalizaciones. Presenta muchos *bugs*¹⁵. Hay una versión gratuita y otra pagada.(Nion, 2017)
- **GrazeRSS:** es un lector de RSS relativamente nuevo que se basa en la aplicación de código abierto NewsRob. Este es un lector de RSS muy ligero y provee sólo unas pocas características incluyendo la lectura fuera de línea y a descarga por *Web Feed*. La interfaz es simple. Es gratuita y de código abierto.
- **Inoreader:** es un lector de RSS que permite buscar noticias basadas en categorías que se adapten a sus intereses. También ofrece la posibilidad de guardar noticias para su posterior lectura, el archivo de sus suscripciones y solo le mostrará elementos no leídos incluso cuando inicie sesión en otros dispositivos. Para ver las noticias en fuera de línea exige un coste monetario. Es gratuita para usar sin compras en la aplicación (Inoreader, 2017).

¹⁵ Error de software o simplemente fallo.

- **News360:** es un lector de RSS. Contiene más de 100.000 fuentes en una variedad de categorías para que pueda seguir los *blogs*, sitios web y fuentes de noticias. También obtiene intereses y recomienda a otros sitios web y artículos. Es gratuita (News360, 2017).
- **News Republic:** lector RSS autónomo. Agrupa noticias basadas en un tema. También cuenta con una compleja información de tendencias y sistema de navegación de etiquetas que le permite seguir todo tipo de temas y luego otros temas basados en esos temas. Admite la creación de cuentas con Facebook, Google+ y correo electrónico. Presenta muchos *bugs*. Es gratuita con compras en la aplicación (News Republic, 2017).
- **Palabre:** es en realidad un cliente Feedly. La aplicación ofrece lectura sin conexión, puede importar *Web Feeds* RSS de OPML¹⁶. También tiene soporte para *Twitter*, *The Old Reader* y *Flickr* usando extensiones. Es gratis con compras en la aplicación.
- **Cuba Noticias:** lector RSS que permite organizar noticias por categoría, descargar noticias breves automáticamente, lectura fuera de línea, importar más fuentes de noticias a través de OPML. Compartir noticias vía SMS, correo electrónico, *Facebook*, *Twitter*. Es gratuita.

A continuación, se realiza una comparación entre las aplicaciones anteriormente abordadas, la cual se puede apreciar en la Tabla 3; destacando las principales desventajas que estas presentan.

Tabla 3: comparación entre aplicaciones homólogas.

Fuente: elaboración propia

| Sistemas Analizados | Código abierto | Lectura fuera de línea | Coste monetario | Actualización sistemática |
|---------------------|----------------|------------------------|-----------------|---------------------------|
| Feedly | No | No | No | No |
| Flipboard | No | No | No | Si |
| gReader | No | Si | Si | Si |
| GrazeRSS | Parcialmente | Si | No | No |

¹⁶ Siglas en inglés para: Outline Processor Markup Language

| | | | | |
|----------------------|----|----|--------------|--------------|
| Inoreader | No | Si | Parcialmente | Si |
| News360 | No | Si | No | Si |
| News Republic | No | No | Parcialmente | No |
| Palabre | No | Si | Parcialmente | Si |
| Cuba Noticias | No | Si | No | Parcialmente |

Después de realizar un análisis comparativo entre las principales características de los sistemas homólogos estudiados, se puede apreciar que no es posible aprovechar ninguna de esas soluciones debido a que no satisfacen las necesidades de la situación problemática. A partir de lo anteriormente planteado se evidencia la necesidad de implementar una nueva solución para la lectura de publicaciones en dispositivos móviles con Sistema Operativo NovaDroid que se encuentran en el entorno cubano.

1.4. Herramientas y tecnologías

1.4.1. Metodología de desarrollo de software

Agile Unified Process UCI (AUP-UCI) (Informáticas, 2015).

Es una versión simplificada de la metodología tradicional *RUP*¹⁷. Describe de una manera simple y fácil de entender la forma de desarrollar aplicaciones de software de negocio usando técnicas ágiles y conceptos que se mantienen válidos en *RUP*.

Características:

- Desarrollo Dirigido por Pruebas.
- Modelado ágil.
- Gestión de Cambios ágil.
- Refactorización de Base de Datos para mejorar la productividad.

Al no existir una metodología de software universal, y ya que toda metodología debe ser adaptada a las características de cada proyecto (equipo de desarrollo, recursos, entre otros), se exige que el proceso sea

¹⁷ Siglas en inglés para: Rational Unified Process.

configurable. Se decidió hacer una variación de la metodología *AUP*, de forma tal que se adapte al ciclo de vida definido para la actividad productiva de la UCI.

Se decide utilizar esta metodología para guiar el proceso de desarrollo de software, ya que es una metodología ágil fácilmente adaptable a los proyectos de la UCI. Es una metodología ágil por lo que la prioridad es satisfacer al cliente mediante tempranas y continuas entregas del software. Es factibles para equipos pequeños, con pocos roles y artefactos. A continuación, se muestran las variaciones de AUP-UCI con respecto a AUP en cuanto a sus fases:

Tabla 4: diferencias entre las fases de AUP y AUP-UCI (Informáticas, 2015).

| Fases AUP | Fases variación AUP-UCI | Objetivos de las fases (AUP-UCI) |
|---------------------|--------------------------------|---|
| Inicio | Inicio | Durante el inicio del proyecto se llevan a cabo las actividades relacionadas con la planeación del proyecto. En esta fase se realiza un estudio inicial de la organización cliente que permite obtener información fundamental acerca del alcance del proyecto, realizar estimaciones de tiempo, esfuerzo y costo y decidir si se ejecuta o no el proyecto. |
| Elaboración | Ejecución | En esta fase se ejecutan las actividades requeridas para desarrollar el software, incluyendo el ajuste de los planes del proyecto considerando los requisitos y la arquitectura. Durante el desarrollo se modela el negocio, obtienen los requisitos, se elaboran la arquitectura y el diseño, se implementa y se libera el producto |
| Construcción | | |
| Transición | | |
| | Cierre | En esta fase se analizan tanto los resultados del proyecto como su ejecución y se realizan las actividades formales de cierre del proyecto. |

1.4.2. Lenguajes de programación.

Java (Openjdk 8)

Es un lenguaje multiplataforma debido a que posee una máquina virtual de java o JVM por sus siglas en inglés, y cuenta con la característica de ser uno de los lenguajes de programación más utilizados en el mundo (Oracle, 2017). Presenta una sintaxis sencilla, orientada a objetos que permite optimizar el tiempo y ciclo de desarrollo (compilación y ejecución). Además, es independiente de las arquitecturas debido a que el compilador no produce un código específico para un tipo de arquitectura (GROUSSARD, 2012). Se utiliza para la implementación de las aplicaciones en el Sistema Operativo Android.

Se selecciona este lenguaje de programación ya que es el lenguaje nativo de Android. La tecnología JAVA es totalmente gratis, y puede ser empleada en plataformas como Mac, Linux y finalmente Windows.

XML¹⁸

XML es uno de los formatos más utilizados para intercambiar información entre aplicaciones de diferentes plataformas. Son ficheros de texto donde los campos o elementos de información que contienen se delimitan mediante pares de etiquetas (Tim Bray, 2006).

1.4.3. Herramienta CASE.

Las herramientas CASE son un conjunto de programas y ayudas que dan asistencia a los analistas, ingenieros de software y desarrolladores, durante todos los pasos del ciclo de vida para el desarrollo de un software.

Visual Paradigm es una herramienta de diseño que facilita el desarrollo de software. Ofrece un paquete completo útil para la captura de requisitos, la planificación del software, la planificación de pruebas, el modelado de clases y el modelado de datos.

Las principales características de la herramienta son:

- Soporta las últimas versiones del UML.
- Posee un poderoso generador de documentación y reportes en formato PDF, HTTP y JPG.
- Proporciona soporte para varios lenguajes en la generación de código e ingeniería inversa como: java, C++, CORBA IDL, PHP, Ada y Python.

¹⁸ Siglas en inglés para: *eXtensible Markup Language*

- Disponibilidad en múltiples plataformas (Windows, Linux) Capacidades de ingeniería directa e inversa.

Se selecciona Visual Paradigm for UML en su versión 8.1 como herramienta para el modelado UML, pues permite trabajar de forma colaborativa, hacer un trabajo organizado y ágil. Posibilita la realización de los diagramas necesarios para el desarrollo y mejor entendimiento de la aplicación. Permite realizar ingeniería inversa a partir del código fuente. Al ser seleccionado el lenguaje de modelado UML, es conveniente tener en cuenta su vinculación con Visual Paradigm, resaltando que este último presenta abundante documentación y demostraciones interactivas (Software Design Tools for Agile Teams, 2017).

1.4.4. Entorno de desarrollo.

Android Studio 1.3.2

Es el entorno de desarrollo integrado oficial para el desarrollo de aplicaciones Android, basado en IntelliJ IDEA, desarrollado por Google y fue anunciado en mayo de 2013. Es un IDE¹⁹ distribuido de forma libre bajo la licencia de Apache *License* 2.0. Tiene integrado una herramienta llamada *Gradle* para la construcción y compilación de la aplicación. Contiene plantillas para ayudar a construir aplicaciones con características comunes. Presenta la ventaja de que permite crear dispositivos virtuales para emular las aplicaciones, debido a que cuenta con un paquete de desarrollo de software, SDK²⁰ por sus siglas en inglés (Google, 2017).

Eclipse

Es una plataforma de desarrollo de código abierto, diseñado para ser extendido de forma indefinida a través de *plugins*. Es un entorno de desarrollo multiplataforma para crear, integrar e implementar herramientas de desarrollo de aplicaciones para su uso en una amplia gama de tecnología informática. No tiene en mente un lenguaje específico, sino que es un IDE genérico, aunque goza de mucha popularidad entre la comunidad de desarrolladores del lenguaje Java. No es más que un entorno de desarrollo en el que localizarían todas las herramientas y funciones necesarias para el trabajo, recogidas además en una atractiva interfaz que lo hace fácil y agradable de usar (M. Milinkovich Nov, 2017).

Selección del entorno de desarrollo integrado

¹⁹ Siglas en inglés para: Integrated Development Environment

²⁰ Siglas en inglés para: Software Development Kit

Se selecciona Android Studio en su versión 1.3.2 para el desarrollo de la aplicación móvil, pues es el entorno de programación oficial para desarrolladores de Android y está basado en IntelliJ IDEA, uno de los IDE para java. Ofrece comodidad para los desarrolladores, permitiendo invocar, durante el desarrollo de aplicaciones, las herramientas necesarias como una forma más ágil de trabajo. En este entorno de desarrollo a través del análisis de código, se destacan los errores de forma inmediata, brindándole soluciones. Su forma de construir los APK es más serio y lo más parecido a un proyecto en java.

Jsoup

Es una biblioteca de Java para trabajar con HTML. Proporciona una API muy conveniente para extraer y manipular datos, utilizando lo mejor de los métodos DOM²¹, CSS²² y jQuery²³. Jsoup está diseñado para tratar con todas las variedades de HTML; desde primitivas y validadas, hasta las de etiquetas inválidas; Jsoup creará un árbol de análisis sintáctico razonable (Jonathan Hedley, 2016).

Android SDK

El paquete de desarrollo de software incluye un conjunto de herramientas para ayudar al programador, tales como bibliotecas, un manejador de emuladores, documentación, códigos ejemplos y tutoriales. Es un paquete multiplataforma. Permite la integración con Eclipse usando el Android Development Tools (ADT) y también con Android Studio (Google, 2017). Se usa en el desarrollo de la solución debido a que es imprescindible para el entorno de desarrollo integrado seleccionado.

SQLite

Es una herramienta de software libre que permite almacenar información en dispositivos de una forma sencilla, eficaz, potente, rápida y en equipos con pocas capacidades de hardware. Es utilizada para el desarrollo de la aplicación. Esta se puede usar tanto en dispositivos móviles como en sistemas de escritorio, sin necesidad de realizar procesos complejos de importación y exportación de datos, por su compatibilidad entre las diversas plataformas disponibles. No necesita un proceso separado funcionando como servidor ya que lee y escribe directamente sobre archivos que se encuentran en el disco duro. Su base de datos se almacena en un único fichero a diferencia de otros SGBD que hacen uso de varios archivos (Vogel, 2010).

²¹ Siglas en inglés para: Document Object Model

²² Siglas en inglés para: Cascading Style Sheets

²³ Es una biblioteca multiplataforma de JavaScript (lenguaje de programación interpretado)

Para el desarrollo de la aplicación en NovaDroid se decidió utilizar SQLite como herramienta para el almacenamiento, debido a su popularidad y utilidad en este Sistema Operativo por su pequeño tamaño en memoria y la realización de las operaciones de manera eficiente y rápida; además de precisar poca configuración y es de código libre.

CAPÍTULO 2. ANÁLISIS Y DISEÑO DE LA PROPUESTA DE SOLUCIÓN

En el presente capítulo serán presentados los resultados obtenidos después de haber cumplido con las fases de análisis y diseño que propone la metodología seleccionada. Para ello se realiza un análisis de los elementos necesarios para confeccionar un lector de noticias mediante *Web Feed* para dispositivos móviles. Por otra parte, se exponen los requisitos funcionales y no funcionales, se especifican las historias de usuario para describir más al detalle cada uno de los requisitos funcionales y se presentan cada uno de los artefactos que propone la metodología para el desarrollo del software. También se explican los diagramas y artefactos referentes al comportamiento de la aplicación, así como la arquitectura facilitando de esta manera una mayor comprensión del sistema a desarrollar.

2.1. Análisis

En esta fase se realiza un estudio para determinar los elementos indispensables para el desarrollo de un lector de noticias para dispositivos móviles mediante *Web Feed*.

2.2. Propuesta de Solución

El presente trabajo propone como solución una aplicación cliente para dispositivos móviles con Sistema Operativo NovaDroid. Dicha aplicación consumirá desde un sitio web, del cual extraerá primeramente su nombre y dirección URL, luego extraerá las publicaciones del sitio mediante los *Web Feed* y finalmente a partir de esta información extraída mostrará al usuario la publicación en su totalidad. La aplicación permitirá al usuario organizar sus sitios de preferencia por temática, guardar las publicaciones para lectura fuera de línea, así como mantener actualizadas automáticamente las publicaciones según la preferencia.

2.3. Diagrama de dominio.

Un modelo de dominio permite comunicar los términos importantes y las relaciones entre ellos, además puede ser tomado como el punto de partida para el diseño del sistema. El objetivo del modelado del dominio es comprender y describir las clases más importantes dentro del contexto del sistema. Estos modelos son construidos con las reglas de UML durante la fase de concepción, en la tarea de construcción del modelo de dominio, presentado como uno o más diagramas de clases y que contiene, no conceptos propios de un sistema de software sino de la propia realidad física (Larman, 2017).

En el diagrama de dominio se muestra las relaciones existentes entre los principales conceptos asociados al sistema de publicación de contenidos, para de esta manera unificar el vocabulario entre los usuarios y los desarrolladores. Como se muestra en la siguiente ilustración.

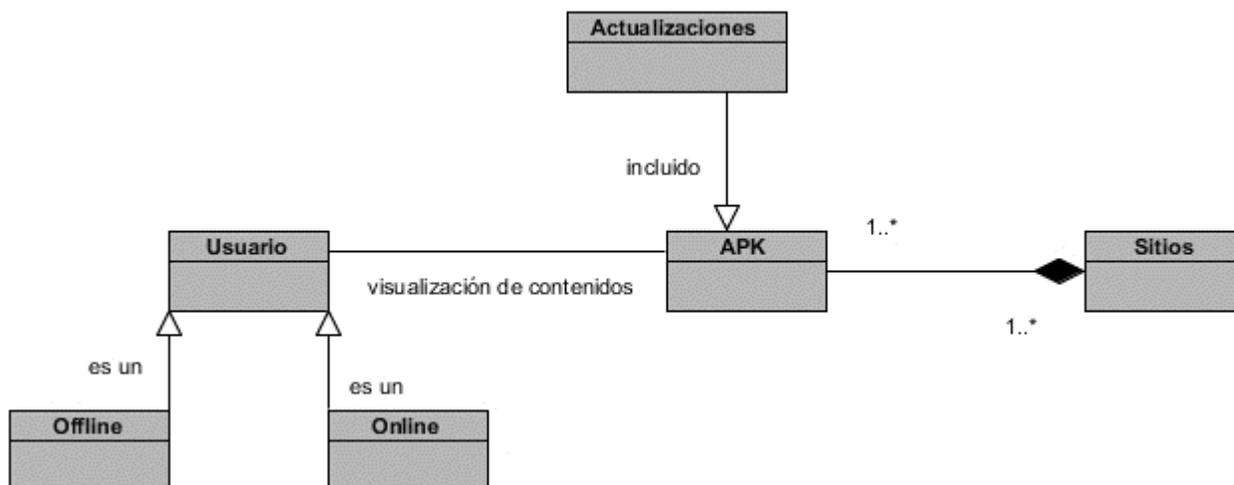


Ilustración 2: diagrama de dominio

2.3.1. Descripción de las clases del diagrama de dominio.

Los usuarios *offline* luego de haber cargado el sitio web y las publicaciones recientes de forma *online* podrán visualizar posteriormente la últimas noticias del sitio web en cuestión de forma *offline*, en caso de haber descargado determinada publicación podrá visualizarla también de forma *offline* ya que la publicación será cargada desde una ubicación local en el dispositivo. En caso del que el usuario seleccione la preferencia de actualización automática la aplicación actualizará las publicaciones de forma automática cuando el dispositivo se encuentre conectado a la red, o sea, en modo *online*. Los usuarios *online* visualizarán las publicaciones en tiempo real ya que este proceso será de forma *online*.

Sitio: conjunto de componentes que contiene módulos que le ofrecen diferentes tipos de contenidos y servicios para los usuarios.

Actualizaciones: conjunto de funcionalidades que permiten actualizar el contenido publicado permitiendo su posterior lectura *offline* para los usuarios.

Apk²⁴: combinación de servicios, que le brindan al usuario la facilidad de gestionar el contenido publicado en determinado sitio web.

Offline: usuario que interactúa con la Aplicación Móvil sin tener conexión para gestionar el contenido previamente extraído del sitio y guardado localmente.

Online: usuario que interactúa con la Aplicación Móvil teniendo conexión para extraer las publicaciones del sitio.

2.4. Requisitos Funcionales del Sistema

Los requisitos funcionales son enunciados acerca de servicios que el sistema debe proveer, de cómo debería reaccionar el sistema a entradas particulares y de cómo debería comportarse el sistema en situaciones específicas. En algunos casos, los requerimientos funcionales también explican lo que no debe hacer el sistema (Ian Sommerville, 2011). En aras del buen funcionamiento de la solución propuesta, se definen los siguientes requisitos funcionales:

Tabla 5: requisitos funcionales.

Fuente: elaboración propia

| No | Requisito funcional (RF) | Descripción | Complejidad | Prioridad para el cliente |
|----|--------------------------|--|-------------|---------------------------|
| 1 | Mostrar ayuda | La aplicación debe mostrar al usuario una ayuda de navegación en la misma | Baja | Media |
| 2 | Listar sitios recientes | La aplicación debe mostrar al usuario el listado de los sitios accedidos recientemente teniendo cuenta los | Baja | Alta |

²⁴ Siglas en inglés para: Android Application Package.

| | | | | |
|---|-----------------------------------|---|-------|------|
| | | siguientes datos: <ul style="list-style-type: none"> • Nombre del sitio • Dirección URL | | |
| 3 | Adicionar sitio | La aplicación debe permitir al usuario añadir nuevo atendiendo a los siguientes parámetros: <ul style="list-style-type: none"> • Dirección de sitio • Temática • Actualizable o no | Media | Alta |
| 4 | Eliminar sitio | La aplicación debe permitir al usuario eliminar sitio(s). | Baja | Alta |
| 5 | Mostrar sitio | La aplicación debe mostrar al usuario la información del sitio web cuenta los siguientes datos: <ul style="list-style-type: none"> • Nombre del sitio • Dirección URL | Baja | Alta |
| 6 | Guardar sitio | La aplicación debe permitir guardar el sitio que el usuario prefiera de forma fuera de línea. | Media | Alta |
| 7 | Mostrar <i>Web Feed</i> del sitio | La aplicación debe mostrar la información del <i>Web Feed</i> del sitio previamente | Media | Alta |

| | | | | |
|-----------|--|--|-------|-------|
| | | seleccionado por el usuario. | | |
| 8 | Mostrar publicación | La aplicación debe permitir al usuario ver la publicación en su totalidad. | Media | Alta |
| 9 | Guardar publicación | La aplicación debe permitir al usuario guardar publicación de su preferencia para posterior lectura fuera de línea. | Media | Alta |
| 10 | Eliminar publicación | La aplicación debe permitir al usuario eliminar publicaciones. | Baja | Alta |
| 11 | Organizar sitios | La aplicación debe permitir organizar los sitios guardados según la temática o preferencia elegida por el usuario. | Baja | Media |
| 12 | Actualizar publicaciones de los sitios | La aplicación debe permitir actualizar sistemáticamente si es preferencia del usuario las publicaciones de los sitios previamente guardados por usuario. | Alta | Media |

2.5. Requisitos No Funcionales del Sistema

Los requisitos son limitaciones sobre servicios o funciones que ofrece el sistema. Incluyen restricciones tanto de temporización y del proceso de desarrollo, como impuestas por los estándares. Los requerimientos no funcionales se suelen aplicar al sistema como un todo, más que a características o a servicios individuales del sistema (Ian Sommerville, 2011). Para lograr la satisfacción del cliente, y una buena calidad en el sistema, se definieron los siguientes requerimientos no funcionales, basados en lo establecido por las normas ISO 25000 Calidad del Producto de Software, específicamente la ISO/IEC 25010, la cual define las características de calidad que se tienen en cuenta al evaluar las propiedades de un producto de software:

Fiabilidad

RNF1. Mientras se captura la información del RSS del sitio la aplicación mostrará al usuario que dicha acción se está realizando.

RNF2. Mientras se extrae la información del sitio seleccionado por el usuario se mostrará que dicha acción se está realizando.

RNF3. Si ocurre algún error en conexión la aplicación notificará al usuario dicho error, mostrará posibles causas con posibles soluciones.

RNF4. Luego de que el usuario realice acciones como eliminar, guardar ya sea un sitio o publicación la aplicación notificará el resultado de dicha instrucción.

Portabilidad

RNF5. Sistema Operativo Android con versión desde 4.0 hasta Marshmallow 6.0–6.0.1.

RNF6. Memoria RAM del dispositivo con capacidad mínima de 256 Mb.

RNF7. Almacenamiento interno o tarjeta micro SD con capacidad mínima de 100Mb.

RNF8. Soporte para conexiones WIFI.

Usabilidad

RNF9. Soporte para los idiomas: inglés y español.

RNF10. Interfaz de usuario basada en *Material Design*²⁵.

2.6. Historias de Usuario (HU)

Las Historias de Usuario son un enfoque de requerimientos ágil que se focaliza en establecer conversaciones acerca de las necesidades de los clientes. Son descripciones cortas y simples de las funcionalidades del sistema, narradas desde la perspectiva de la persona que desea dicha funcionalidad, usualmente un usuario (COHN, 2008). A continuación, se muestra la HU referente a la propuesta de solución:

Tabla 6: historia de Usuario # 4.

Fuente: elaboración propia

| Numero: 5 | Nombre del requisito: Mostrar sitio |
|--|--|
| Programador: Elioenai Reyes González | Iteración Asignada: 5 |
| Prioridad: media | Estimado: 2 días |
| Riesgo en Desarrollo: <ul style="list-style-type: none"> Fallo de conexión. Fallo del sitio. Interrupción del fluido eléctrico. | Tiempo Real: 1 día |
| Descripción: La aplicación debe mostrar al usuario la información del sitio cuenta los siguientes datos: <ul style="list-style-type: none"> Nombre del sitio Dirección URL | |
| Observaciones: Inicialmente se visualizará el sitio con sus respectivos datos, pero en el caso de no existir conexión con el sitio, se notificará al usuario. | |

²⁵ Material Design es una normativa de diseño enfocado en la visualización del sistema operativo Android

Prototipo de interfaz:

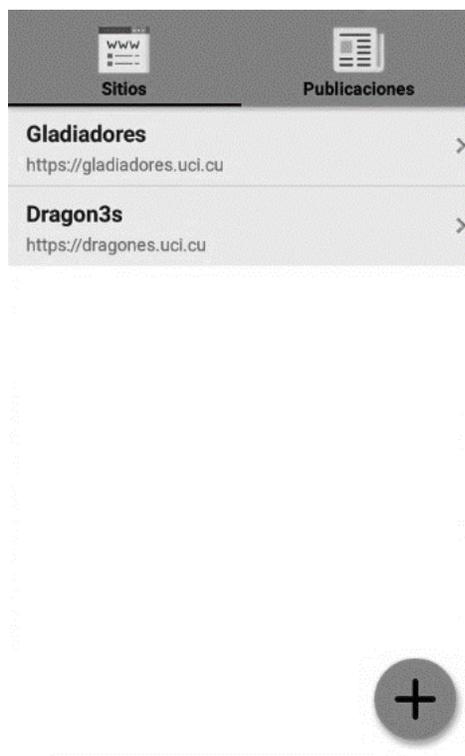


Tabla 7: historias de usuario # 7.

Fuente: elaboración propia

| | |
|---|--|
| Numero: 7 | Nombre del requisito: mostrar Web Feed de sitio |
| Programador: Elioenai Reyes González | Iteración Asignada: 7 |
| Prioridad: media | Estimado: 2 días |
| Riesgo en Desarrollo: <ul style="list-style-type: none"> Fallo de conexión. Fallo del sitio. | Tiempo Real: 1 día |

- Interrupción del fluido eléctrico.

Descripción:

La aplicación debe mostrar la información de la Fuente Web teniendo en cuenta los siguientes elementos:

- Título de publicación
- Fecha de publicación
- Descripción de publicación

Observaciones:

Inicialmente se visualizará este listado de las últimas publicaciones del sitio, pero en el caso de no existir conexión con el mismo, se notificará al usuario.

Prototipo de interfaz:

Nuevo servidor de gestión de dependencias de PHP en la UCI

Fri, 17 Mar 2017 14:11:26 +0000

Hola, esta vez les traigo una nueva herramienta que ha puesto a disposición de la Comunidad y de ..

Seguda fase de la Arquitectura de Referencia para PHP

Thu, 09 Mar 2017 14:15:24 +0000

Hola, les dejo un link a un post del portal de Dragones en el que es introducen algunos detalles ..

FELIZ AÑO 2017

Sat, 17 Dec 2016 17:09:36 +0000

Nuestra comunidad les desea un feliz fin de año y un próspero año 2017, en el próximo año estarem..

Problemas con dependencias del Composer UCI

Wed, 26 Oct 2016 13:35:34 +0000

Hola a todos, en los últimos días se han estado presentando problemas con la

2.7. Arquitectura de Software

Según Roger Pressman: “*En su forma más simple, la arquitectura del software es la estructura u organización de los componentes del programa, la manera en que estos interactúan y la estructura de datos que utilizan*” (PRESSMAN, 2010). O sea, la arquitectura de software es una forma de representar sistemas mediante el uso de la abstracción, de forma que aporte el más alto nivel de comprensión de los mismos. Esta representación incluye los componentes fundamentales del software, su comportamiento y formas de interacción para satisfacer los requisitos del sistema. En el presente trabajo de diploma se propone una arquitectura Modelo-Vista-Controlador (MVC), ya que la aplicación estará basada en la interacción de un cliente con un servidor a través de la web y esta arquitectura es ampliamente usada para este propósito, además de que es empleada cada vez más en la implementación de aplicaciones Android.

- **Modelo:** maneja todo lo referente a la persistencia de datos de la aplicación, las clases entidades, el acceso a la red y los elementos necesarios para manejar dichos elementos (paquete model).
- **Vista:** representa la interfaz gráfica para la interacción con el usuario. Dentro se ubican todos los componentes que intervienen en la visualización del resultado de la comunicación con el Controlador (paquete view).
- **Controlador:** aquí se tendrán las clases que interactúan con el componente Vista, recibiendo las solicitudes de eventos de los usuarios y con el Modelo registrando los cambios realizados por el mismo (paquete controller).

2.8. Patrones de Diseño

Un patrón de diseño es una descripción de la comunicación entre objetos y clases, personalizada para resolver un problema de diseño general en un contexto particular. Identifica clases, instancias, roles, colaboraciones y la distribución de responsabilidades (Erich Gamma, 1994). Se presentan como pares de problema-solución con nombre, sugiriendo aspectos relacionados con la asignación de responsabilidades. Los patrones de diseño se caracterizan por:

- Representar soluciones técnicas a problemas concretos.
- Propiciar la reutilización.
- Representar problemas frecuentes.

2.8.1. Patrones GRASP

Los patrones GRASP²⁶ describen los principios fundamentales de la asignación de responsabilidades a objetos (LARMAN, 2004). El nombre se eligió para indicar la importancia de captar estos principios, si se quiere diseñar un software de manera eficaz.

- **Experto:** se usa más que cualquier otro al asignar responsabilidades, es un principio básico que suele utilizarse en el diseño orientado a objetos (LARMAN, 2004). Consiste en la asignación de una responsabilidad a la clase que cuenta con la información necesaria para llevarla a cabo. El uso de este patrón da pie a un bajo acoplamiento y una alta cohesión, lo que favorece al hecho de tener sistemas más robustos y de fácil mantenimiento. El cumplimiento de una responsabilidad requiere a menudo información distribuida en varias clases de objetos. En la siguiente ilustración se muestra el uso de este patrón con el que se pretende que los objetos realicen las acciones relacionadas con la información que posee.

```
String url = args[0];
rssFeed = rssParser.getRSSFeed(url);
Log.d("rssFeed", " " + rssFeed);
if (rssFeed != null) {
    Log.e("RSS URL",
        rssFeed.getTitle() + "" + rssFeed.getLink() + ""
        + rssFeed.getDescription() + ""
        + rssFeed.getLanguage());
    RSSDatabaseHelper rssDb = new RSSDatabaseHelper(context.getApplicationContext());
    WebSite site = new WebSite(rssFeed.getTitle(), rssFeed.getLink(), rssFeed.getRSSLink(),
        rssFeed.getDescription(), thematic.getNameIconThematic());
    rssDb.addSite(site);
}
```

Ilustración 3: ejemplo del patrón experto

- **Bajo Acoplamiento:** el acoplamiento es una medida de la fuerza con que una clase está conectada a otras clases, con que las conoce y con que recurre a ellas. Una clase con bajo (o

²⁶ Siglas en inglés para: General Responsibility Assignment Software Patterns

débil) acoplamiento no depende de muchas otras (LARMAN, 2004). El bajo acoplamiento soporta el diseño de clases más independientes y reutilizables, lo cual reduce el impacto de los cambios y acrecienta la oportunidad de una mayor productividad. Ejemplo del uso de este patrón en la solución propuesta se muestra en las clases Global.java y Constants.java, donde se minimizan las relaciones de estas con el resto de las clases.

- **Alta Cohesión:** en la perspectiva del diseño orientado a objetos, la cohesión (o más exactamente, la cohesión funcional) es una medida de cuán relacionadas y enfocadas están las responsabilidades de una clase. Una alta cohesión caracteriza a las clases con responsabilidades estrechamente relacionadas que no realicen un trabajo enorme, clases con responsabilidades moderadas en un área funcional que colaboran con las otras para llevar a cabo las tareas (LARMAN, 2004). Se evidencia este patrón en la clase RSSDatabaseHandler.java y RSSParser.java, cuyas funcionalidades son utilizadas por otras clases para gestionar la base de datos y extraer información del sitio respectivamente.
- **Controlador:** un controlador es un objeto de interfaz no destinada al usuario que se encarga de manejar un evento del sistema (LARMAN, 2004). En la siguiente ilustración se muestra el uso de este patrón.

```
ListView lv = getListView();

// Lanzar una nueva pantalla al seleccionar un elemento de lista única
lv.setOnItemClickListener(new OnItemClickListener() {

    public void onItemClick(AdapterView<?> parent, View view,
        int position, long id) {
        Intent in = new Intent(getApplicationContext(), ActivityDisplayWebPage.class);

        // Obtener datos de la página
        String page_url = ((TextView) view.findViewById(R.id.page_url)).getText().toString();
        String title = ((TextView) view.findViewById(R.id.title)).getText().toString();
        String description = ((TextView) view.findViewById(R.id.link_website)).getText().toString();
        String pub_date = ((TextView) view.findViewById(R.id.pub_date)).getText().toString();
        in.putExtra(TAG_LINK, page_url);
        in.putExtra(TAG_TITLE, title);
        in.putExtra(TAG_PUB_DATE, pub_date);
        in.putExtra(TAG_DESCRIPTION, description);
        startActivity(in);
    }
});
```

Ilustración 4: ejemplo del patrón controlador

2.8.2. Patrones GoF

Los patrones GoF²⁷ son alternativas de solución a problemas conocidos, pero son mucho más específicas las situaciones en las que se aplican. Se clasifican en creacionales, estructurales y de comportamiento (Erich Gamma, 1994). A continuación, se enuncian brevemente los utilizados en el sistema.

- **Singleton:** está diseñado para restringir la creación de objetos pertenecientes a una clase o el valor de un tipo a un único objeto. Su intención consiste en garantizar que una clase solo tenga una instancia y proporcionar un punto de acceso global a ella (Erich Gamma, 1994). Se empleó en la creación de la clase `ControlerConexion.java` para garantizar que no exista duplicación de los objetos que controlan el acceso a la red y a la base de datos, y así no incurrir en problemas de concurrencia.
- **Observador:** permite a los objetos captar dinámicamente las dependencias entre objetos, de tal forma que un objeto notificará a los objetos dependientes de él cuando cambia su estado, siendo actualizados automáticamente. En la solución propuesta, este patrón es utilizado al crear clases que extiendan de la clase nativa de Android. En la siguiente ilustración se muestra el uso de este patrón.

```
btnDownload = (FloatingActionButton) findViewById(R.id.btnDownload);  
btnDownload.setOnClickListener(this);  
btnDownload.setOnLongClickListener(this);
```

Ilustración 5: Ejemplo del patrón observador

- **Adaptador:** se emplea para generar los elementos de los componentes visuales de la lista de contenidos, que requiere de un adaptador para crear los contenidos que se muestran a los usuarios. Se evidencia en las clases `ViewPagerAdapter.java` y `CustomAdapterForGrid.java`. En las siguientes ilustraciones se muestra el uso de este patrón.

²⁷ Siglas en inglés para: Gang of Four

```
private void setupViewPager(ViewPager viewPager) {  
    ViewPagerAdapter adapter = new ViewPagerAdapter(getSupportFragmentManager());  
    adapter.addFrag(new SitesFragment(), getString(R.string.tap_title_sites));  
    adapter.addFrag(new PubsFragment(), getString(R.string.tab_title_pubs));  
    viewPager.setAdapter(adapter);  
}
```

Ilustración 6: ejemplo del patrón adaptador

```
GridView gv = (GridView) findViewById(R.id.gridView1);  
gv.setAdapter(new CustomAdapterForGrid(this, nameIcons, Constants.Id_res_icons));
```

Ilustración 7: ejemplo del patrón adaptador

2.9. Diagramas de clase del diseño

Un DCD²⁸ representa las especificaciones de las clases e interfaces software en una aplicación. Las clases de diseño de los DCD muestran las definiciones de las clases software en lugar de los conceptos del mundo real (Larman, 2004).

²⁸ Siglas en español para: diagrama de clases de diseño.

Lector de noticias para dispositivos móviles con **Capítulo 2** Sistema Operativo NovaDroid.



Ilustración 8: diagrama de clases

2.10. Modelo de Datos

Para el desarrollo del lector de noticias para dispositivos móviles con Sistema Operativo NovaDroid se definió el siguiente modelo de datos apoyado en el diagrama Entidad-Relación, el cual define todos los datos que se introducen, se almacenan, se editan y se generan dentro del sistema.

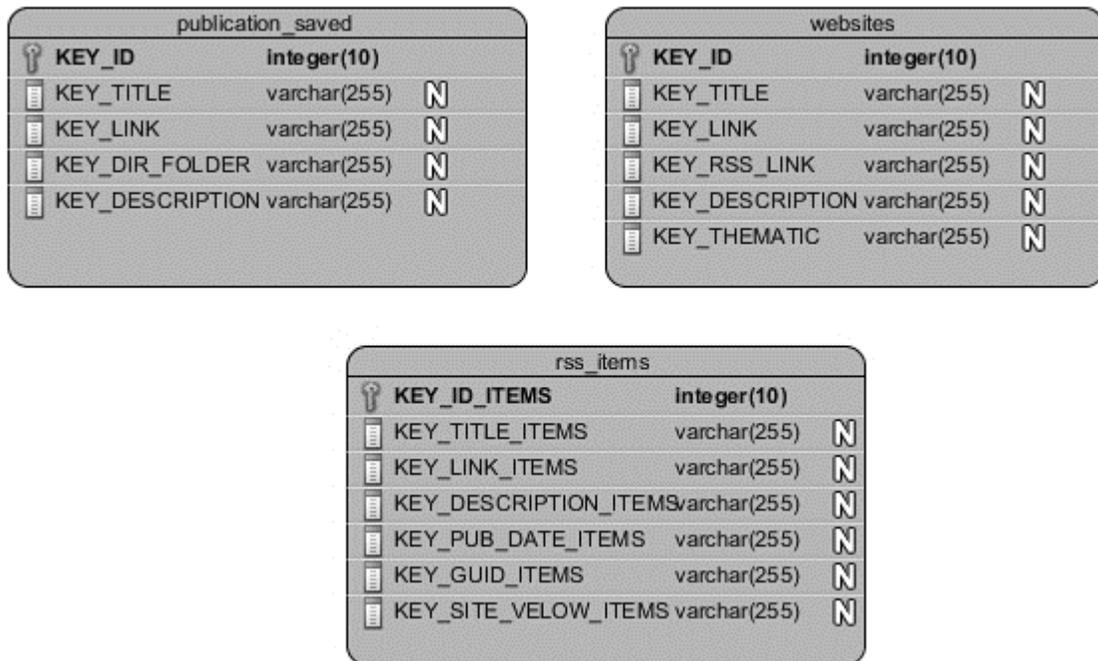


Ilustración 9: modelo de datos

2.11. Modelo de despliegue

Un diagrama de despliegue detalla las capacidades de red, las especificaciones del servidor, los requisitos de hardware y la información relacionada con la forma en la que los componentes se comunicarán a lo largo de la infraestructura del sistema. También se utiliza para visualizar la distribución de los componentes de software en los nodos físicos. El mismo está compuesto por: nodos, dispositivos y conectores (Roger S. Pressman, 2010).



Ilustración 10: diagrama de despliegue

CAPÍTULO 3. IMPLEMENTACIÓN Y VALIDACIÓN DEL LECTOR DE NOTICIAS PARA DISPOSITIVOS MÓVILES CON SISTEMA OPERATIVO NOVADROID

En este capítulo se describen los elementos definidos en el proceso de desarrollo de la aplicación Lector de noticias para dispositivos móviles con Sistema Operativo NovaDroid, para obtener un producto acorde a cada uno de los atributos de calidad. Se describe los elementos físicos del sistema y sus relaciones a través del diagrama de componentes. Se exponen los estándares de codificación utilizados para el desarrollo de la aplicación y finalmente se ejecutan las pruebas de software para verificar el correcto funcionamiento de la aplicación.

3.1. Estándares de codificación

Los estándares de codificación son un conjunto de reglas o patrones de codificación a seguir por los desarrolladores con el objetivo de establecer un orden y un formato común en el código fuente del sistema en desarrollo. Estos estándares deben cumplir con dos principios: legibilidad y mantenibilidad, lo cual repercute en lo bien que el programador entienda el código y la facilidad con que el sistema puede modificarse (Microsoft, 2016). Con el propósito de estandarizar las nomenclaturas en la implementación de los componentes y obtener un buen producto, se utilizan los siguientes estándares de codificación:

- **Añadir un único espacio a ambos lados de operadores**

```
Cursor cursor = db.query(TABLE_RSS, new String[] { KEY_ID, KEY_TITLE,
        KEY_LINK, KEY_RSS_LINK, KEY_DESCRIPTION, KEY_THEMATIC }, KEY_THEMATIC + "=?",
        new String[] {thematic}, null, null, null, null);

if (cursor != null) {
    cursor.moveToFirst();
}
```

Ilustración 11 ejemplo de estándar de codificación espacio entre operadores

- **Sentencias**

El carácter inicio de bloque "{" debe situarse al final de la línea que inicia el bloque y "}" debe situarse en una nueva línea tras la última línea del bloque y alineada con respecto al primer carácter de dicho bloque. Todas las sentencias de un bloque deben encerrarse entre llaves "{...}", aunque el bloque

contenga solo una sentencia. Esta práctica permite añadir código sin cometer errores accidentalmente al olvidar añadir las llaves. En la siguiente ilustración se muestra el uso de este estándar de codificación.

```
/**
 *set metodos
 */
public void setId(Integer id){
    this.id = id;
}

public void setTitle(String title){
    this.title = title;
}

public void setLink(String link){
    this.link = link;
}

public void setRss_link(String rss_link){
    this.rss_link = rss_link;
}

public void setDescription(String description){
    this.description = description;
}

public void setThematic_name(String thematic_name) {
    this.thematic_name = thematic_name;
}
```

Ilustración 12: ejemplo de estándar de codificación sentencias

- **Asignación de nombres:** cada tipo de elemento debe nombrarse con una serie de reglas determinadas.

Métodos

Se nombrarán en minúscula y en caso de nombres con dos palabras, deberán estar unidas y la primera letra de cada palabra deberá estar en mayúscula. En la siguiente ilustración se muestra el uso de este estándar de codificación.

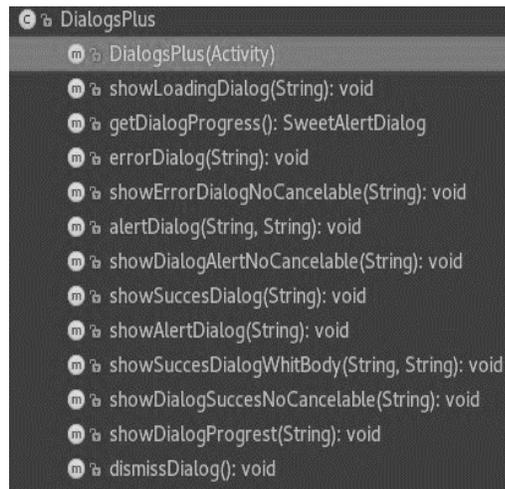


Ilustración 13: ejemplo de estándar de codificación asignación de nombres a métodos

Clases

Todo nombre de una clase debe iniciar en mayúscula. En la siguiente se muestra el uso de este estándar de codificación.



Ilustración 14: ejemplo de estándar de codificación asignación de nombres a clases

Variables

Los nombres de las variables deben comenzar con minúscula. No se permitirá el empleo de caracteres especiales, así como nombre sin ningún tipo de significado funcional. Se trabajarán con las variables i, j, k para el empleo en los bucles for. En la siguiente ilustración se muestra el uso de este estándar de codificación.

```
Integer id;  
String title;  
String link;  
String rss_link;  
String description;  
String thematic_name
```

Ilustración 15: ejemplo de estándar de codificación asignación de nombres a variables

Constantes

Los nombres de constantes deberían escribirse todo en mayúsculas con las palabras separadas por subrayados ("_"). Todas serán declaradas como public static final o private static final. En la siguiente Ilustración se muestra el uso de este estándar de codificación.

```
//fuentes  
public static final String STREATWEAR_FONT = "fonts/streatwear.otf";  
public static final String DITI_SWEET_FONT = "fonts/diti_sweet.ttf";  
public static final String GERMANIA_ONE_FONT = "fonts/germania_one.ttf";  
public static final String SLIMJIM_FONT = "fonts/slimjim.ttf";  
public static final String VOLATIRE_FONT = "fonts/volatire.ttf";
```

Ilustración 16: ejemplo de estándar de codificación asignación de nombres a variables constantes

3.2. Pruebas de software

Las pruebas de software comprenden una fase del proceso de desarrollo que se centra en asegurar la calidad, fiabilidad y robustez de un software, dentro de un contexto o escenario donde está previsto que este sea utilizado. Se encuentra encaminado a medir el cumplimiento de las funcionalidades establecidas por el cliente, reduciendo de esta manera el número de errores no detectados. Esta fase es una de las más costosas del ciclo de vida del software. En sentido estricto, deben realizarse pruebas a todos los artefactos generados durante la construcción de un producto, lo que incluye especificaciones de requisitos, casos de uso, diagramas de diversos tipos y, por supuesto, el código fuente y el resto de productos que forman parte de la aplicación (Roger S. Pressman, 2010).

3.1. Diagrama de paquetes

Un diagrama de paquetes en el Lenguaje Unificado de Modelado representa las dependencias entre los paquetes que componen un modelo. Es decir, muestra cómo un sistema está dividido en agrupaciones lógicas y las dependencias entre esas agrupaciones. Dado que normalmente un paquete está pensado como un directorio, los diagramas de paquetes suministran una descomposición de la jerarquía lógica de un sistema (Damián Gutierrez, 2009).

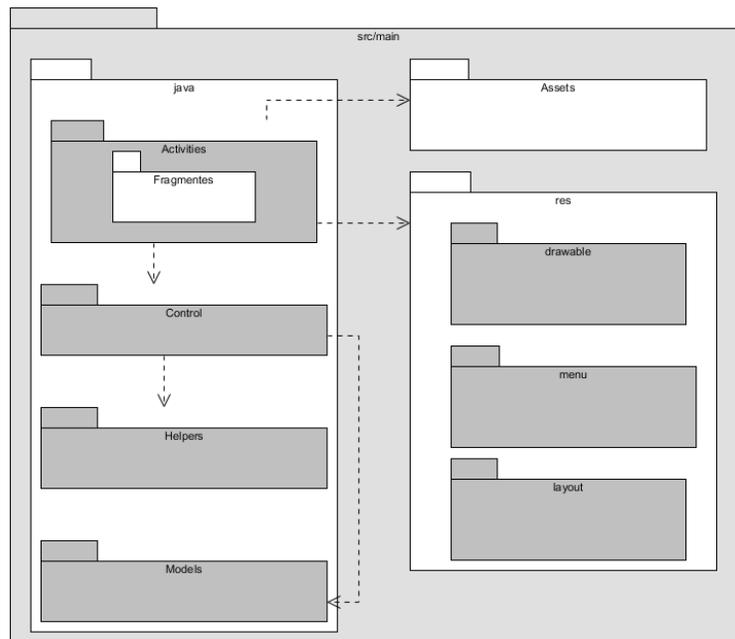


Ilustración 17: diagrama de paquetes

3.1.1. Pruebas unitarias

Una prueba unitaria es una forma de comprobar el correcto funcionamiento de un módulo de código. Esto sirve para asegurar que cada uno de los módulos funcione correctamente por separado (Prueba unitaria, 2014), (Barrientos, 2014). Generalmente son realizadas por los mismos programadores puesto que, al conocer con mayor detalle el código, se les simplifica la tarea de elaborar conjuntos de datos de prueba para testarlo. Por último, es importante que las funcionalidades de cada componente unitario sean cubiertas, por al menos, dos casos de prueba, los cuales deben centrarse en probar al menos una funcionalidad positiva y una negativa (Glenford J. Myers, 2011).

Las pruebas de unidad siempre están orientadas a caja blanca y le permiten al programador conocer si determinada funcionalidad se puede agregar al sistema sin afectar su funcionamiento. Facilitan que el programador cambie el código para mejorar su estructura (refactorización), puesto que permiten hacer pruebas sobre los cambios y asegurarse de que no han introducido errores.

Para la realización de las pruebas unitarias es utilizada la extensión de Android JUnit, que no es un marco de pruebas unitarias, es el elemento central de la práctica de pruebas de XP²⁹. Es una parte integral del SDK de Android. Permitted probar los componentes específicos mediante clases de casos de pruebas. A cada una de las clases de la aplicación se le realizaron varias pruebas con el objetivo de obtener y corregir los errores existentes antes de realizar la entrega al cliente.

Resultado de las pruebas unitarias

Insuficiencias identificadas en las iteraciones:

- No se actualizaban los contenidos después de la lectura del *Web Feed*.
- No se validaba correctamente la dirección URL del sitio.
- No se identificaba cuando el dispositivo estaba *offline*.

Se les dio solución a las insuficiencias, obteniendo finalmente resultados con el éxito esperado en las siguientes etapas de iteración de pruebas. A continuación, se muestran algunas imágenes donde se evidencian algunos de los resultados obtenidos.

²⁹ Siglas en inglés para: Extreme Programming

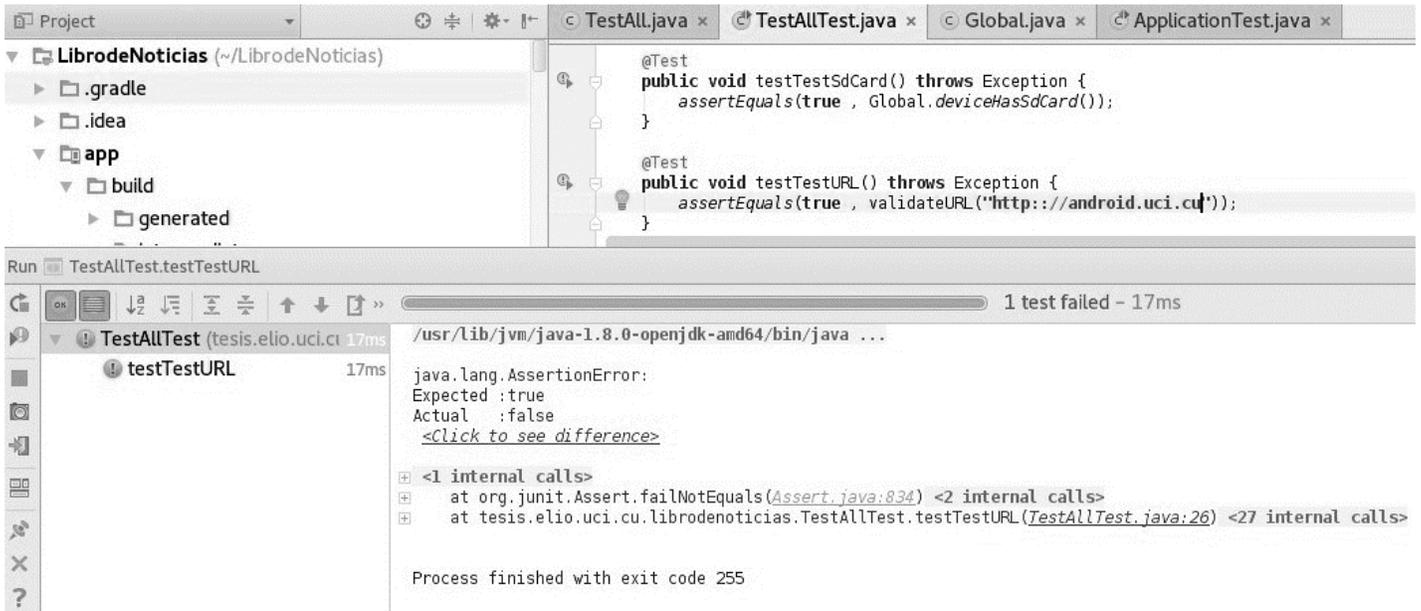


Ilustración 18: resultados de las pruebas unitarias validando la URL del sitio.



Ilustración 19: resultados de las pruebas unitarias validando la URL del sitio.

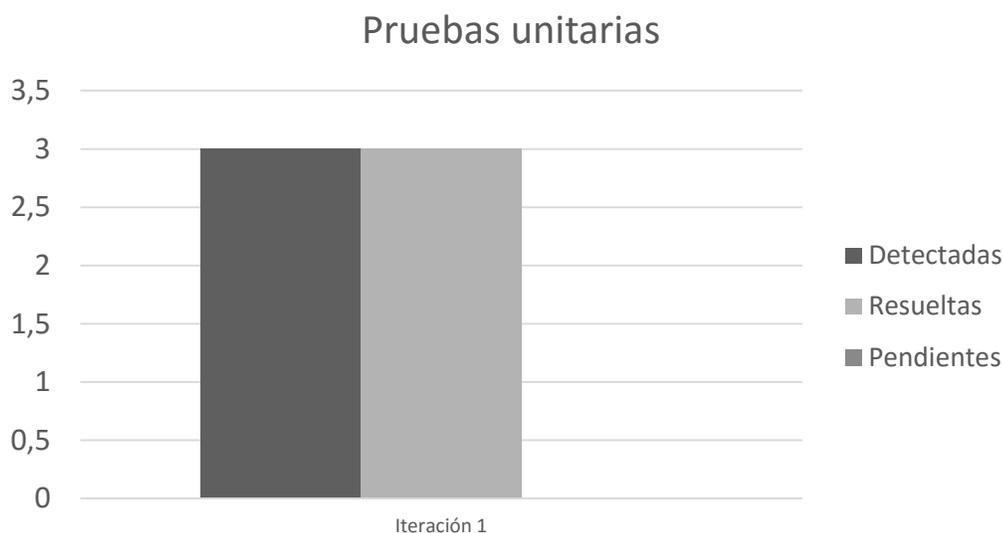


Ilustración 20: representación gráfica de los resultados de las pruebas unitarias

3.1.2. Pruebas de funcionalidad

Las pruebas funcionales son aquellas que se llevan a cabo sobre la interfaz del software sin prestar atención al código, por lo que los casos de prueba son creados con el objetivo de demostrar que la entrada es aceptada de forma adecuada y que se produce una salida correcta. El diseño de estas pruebas se realiza con la intención de detectar funciones incorrectas o ausentes, errores en accesos a bases de datos externas, errores de interfaz, errores de rendimiento y errores de inicialización y de terminación. Dentro de la prueba se incluyen la técnica de partición de equivalencia que será la empleada en la validación. Esta técnica divide el campo de entrada de un programa en clases de datos de los que se pueden derivar casos de prueba (Roger S. Pressman, 2010).

Se realizaron los casos de prueba para validar el funcionamiento de la propuesta de solución implementada para comprobar que el sistema se conecte con el sitio correctamente. A continuación, en las Tablas 8 y 9, se muestra la especificación del caso de prueba para el requisito "Adicionar Sitio".

Tabla 8: variables del caso de prueba para el escenario “adicionar sitio”.

Fuente: elaboración propia

| No. | Nombre del campo | Clasificación | Valor Nulo | Descripción |
|-----|------------------|---------------|------------|--|
| 1 | Insertar URL | Área de texto | No | Campo obligatorio con formato de dirección sitio web |

Tabla 9: casos de prueba Introducir “Adicionar sitio”: Fuente de elaboración propia

| Escenario | Descripción | 1 | 2 | Respuesta | Flujo central |
|--|---|---|------------------------------|---|--|
| <i>Escenario 1 Adicionar Sitio Datos correctos</i> | Se adiciona sitio. | Se inserta la dirección URL del sitio válidamente | Se selecciona la temática | Muestra el mensaje “Sitio adicionado con éxito” | El usuario introduce los datos de forma válidos, selecciona la temática y da clic en el botón “Cargar”. |
| <i>Escenario 2 Adicionar Sitio URL del sitio incorrecta</i> | No se inserta la dirección URL del sitio válidamente. | Se introduce dirección URL del sitio inválida | Se selecciona la temática | Muestra el mensaje “ <i>Dirección URL invalidad)</i> ” | El usuario introduce la dirección URL del sitio invalida, selecciona la temática y da clic en el botón “Cargar” |
| <i>Escenario 3 Adicionar Sitio No selección de la temática</i> | No se selecciona la temática del sitio. | Se inserta la dirección URL del sitio válidamente | No se selecciona la temática | Muestra el mensaje “ <i>Seleccione la tematicice por favor)</i> ” | El usuario introduce la dirección URL del sitio válida pero no selecciona la temática y da clic en el botón “Cargar” |

Resultados de pruebas funcionales

Las pruebas funcionales se realizaron en tres iteraciones a medida que avanzaba el sistema, las cuales guiaron la calidad del mismo y determinaron en cada momento si estaba o no en condiciones de continuar avanzando con el ciclo de desarrollo. Estas pruebas arrojaron en un principio un total de 3 no conformidades en la lógica del funcionamiento y validaciones incorrectas; permitiendo posteriormente darle solución. En la siguiente ilustración se representa gráficamente el resultado que arrojaron estas pruebas.

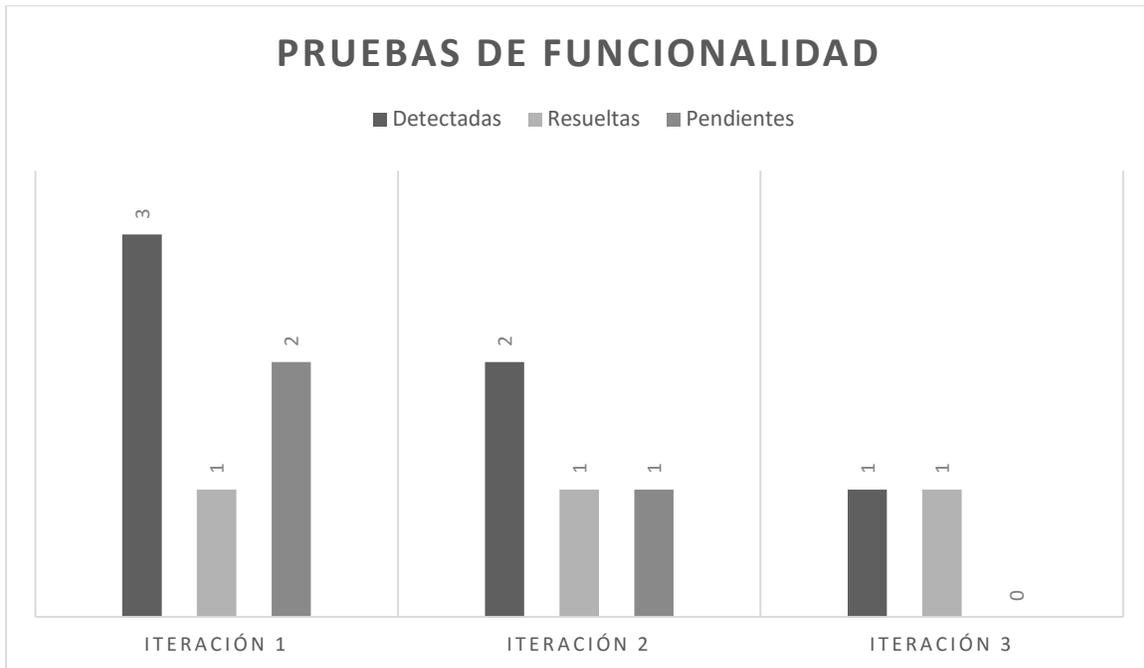


Ilustración 21: representación gráfica de los resultados de las pruebas funcionales

3.1.3. Prueba de compatibilidad

Las pruebas de compatibilidad son muy significativas para mostrar una calidad adecuada del sistema en diferentes entornos. Se realizarán a software o aplicación, para comprobar que son compatibles con todos los navegadores de internet o todos los sistemas operativos. Estas pruebas son realmente importantes para que el producto llegue a todos los usuarios y que todos puedan utilizarlo, permitiendo descubrir problemas del sistema antes de que se encuentre en línea (Roger S. Pressman, 2010).

Para el desarrollo de estas pruebas se utilizó la herramienta de Android Studio, AVD³⁰, para crear diferentes prototipos de teléfonos móviles, con diferentes versiones y resoluciones de pantalla. Se comprobó la compatibilidad de la aplicación para cada nivel de API superior al valor especificado *minSdkVersion*³¹.

³⁰ Siglas en inglés para: Android Virtual Device.

³¹ Valor entero que designa el nivel de API mínimo que se requiere para ejecutar la aplicación.

CONCLUSIONES

Una vez concluido el desarrollo de la investigación se arriba a las siguientes conclusiones:

- La caracterización de los elementos teóricos de la investigación permitió abordar los conceptos asociados al dominio de trabajo, al concluir el estudio de homólogos se extrajeron funcionalidades útiles para el desarrollo del Lector de noticias para dispositivos móviles con Sistema Operativo NovaDroid.
- El empleo de la arquitectura MVC de conjunto con el lenguaje de programación Java y las tecnologías Android Studio, SQLite y Jsoup, guiado por la metodología AUP en su versión para la UCI posibilitó llevar a buen término el desarrollo de la solución.
- La ejecución de pruebas de software al Lector de noticias para dispositivos móviles con Sistema Operativo NovaDroid permitió la verificación de todos los requisitos identificados y la aceptación por parte del cliente.

RECOMENDACIONES

Implementar en la aplicación las funcionalidades necesarias para la descarga de publicaciones de forma planificada y automática sin la necesidad de la visualización de las mismas por parte del usuario.

REFERENCIAS BIBLIOGRÁFICAS

10 best RSS reader apps for Android. [En línea] [Citado el: 05 de diciembre de 2016.] <http://www.androidauthority.com/best-rss-reader-apps-for-android-2-174770/>.

Argentina, Ministerio de Educación. 2009. RSS. Sindicación de contenidos. 2009.

COHN, M. 2008. Agile Estimating Planning. Agile Development Practices. [En línea] 2008. <https://www.mountaingoatsoftware.com/uploads/presentations/Agile-EstimatingPlanning-Agile-Development-Practices-2008.pdf>.

Corrales, Antonio Rafael. 2009. La integración de las Tecnologías de la Información y Comunicación (TIC) en el área de la Educación Física. *Hekademos*. Diciembre de 2009.

El empleo de internet como nuevo canal de distribución: un análisis de sus principales ventajas e inconvenientes. **Blanco, Carlos Flavián y Sarasa, Raquel Gurrea. 2003.** 72, 2003.

Feenstra, Ramón Andrés y Casero-Ripollés, Andreu. 2012. Revista Comunicación y hombre. *Nuevas formas de producción de noticias en el entorno digital y cambios en el periodismo.* 2012.

GAMMA, E., HELM, R., JOHNSON, R. y VLISSIDES, J. Design Patterns: elements of Reusable Object-Oriented Software. [En línea] <http://www.uml.org.cn/c++/pdf/DesignPatterns.pdf>.

García Perellada, Lilia Rosa. 2014. *Propuesta de arquitectura para una Nube Privada con soporte para Infraestructura como Servicio.* La Habana: s.n., 2014.

GARLAN, D. y SHAW, M. 1993. An introduction to software architecture. Advances in software engineering and knowledge engineering. [En línea] 1993. <https://books.google.com/books?hl=en&lr=&id=k8bsCgAAQBAJ&oi=fnd&pg=PA1&dq=%22beyond+the+algorithms+and+data+structures+of+the+computation:%22+%22and+frameworks+for+systems+that+serve+the+needs+of+specific%22+%22as+industry+and+scientific%22+%22choices+a>.

Google. 2017. Android Developers. [En línea] Creative Commons Attribution, 2017. [Citado el: 14 de enero de 2016.] <http://developer.android.com/intl/es/tools/studio/studio-features.html>.

GROUSSARD, Thierry. 2012. *JAVA 7: los fundamentos del lenguaje Java.* s.l.: ediciones Eni, 2012.

Huidobro, José. 2006. Tecnologías de información y comunicación. Universidad Politécnica de Madrid : s.n., 2006.

Información, Oficina Nacional de estadística e. 2015. Anuario estadístico de Cuba. Tecnologías de la Información y las Comunicaciones. 2015.

Informáticas, Universidad de las Ciencias. 2015. Programa de mejora. *Metodología de desarrollo para la Actividad productiva de la UCI.* 2015.

Internet y la sociedad red. **Castells, Manuel. 2011.** s.l.: la factoría, 2011, Vol. 14.

Jonathan Hedley. 2016. Jsoup Java HTML. [En línea] 2016. <https://jsoup.org/>.

LARMAN, C. 2004. Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development, Third Edition. [En línea] 2004. <https://aanimesh.files.wordpress.com/2013/09/applying-umland-patterns-3rd.pdf>.

Leite, Ricardo. 2008. Sindicación de Contenidos web - RSS Web Feeds. 2008.

Oracle. 2017. Java, cree el futuro. [En línea] Oracle, 2017. [Citado el: 14 de Enero de 2016.] <http://www.oracle.com/es/technologies/java/features/index.html>.

PRESSMAN, R.S. 2010. *Software Engineering. A practitioner's Approach. 7th.* New York: mcGraw Hill. 2010.

RSS. RSS explicada. [En línea] [Citado el: 26 de noviembre de 2016.] <http://www.rss.nom.es/>.

RSS explicada. Ejemplos. [En línea] [Citado el: 05 de diciembre de 2016.] <http://www.rss.nom.es/ejemplos-contenido-rss/>.

2005. RSS revoluciona la distribución de contenidos online. [En línea] 2005. http://www.iese.edu/es/files/art_computing_tonijuan_rss_may05_tcm5-5572.pdf.

«Aplicaciones de Android en Google Play». 2017. Accedido mayo 15. <https://play.google.com/store?hl=es>.

Arturo Baz Alonso, Irene Ferreira Artime, María Álvarez Rodríguez, y Rosana García Baniello. 2006. «Dispositivos móviles». http://isa.uniovi.es/docencia/SIGC/pdf/telefonía_movil.pdf.

- Barrientos, Pablo Andrés. 2014.** «Enfoque para pruebas de unidad basado en la generación aleatoria de objetos». Tesis, Facultad de Informática. <http://hdl.handle.net/10915/34969>.
- Chris Wellons. 2013.** «Atom vs. RSS « null program». <http://nullprogram.com/blog/2013/09/23/>.
- Cubadebate. 2016.** «Cubadebate: Lo más leído y comentado en 2016». *Cubadebate*. <http://www.cubadebate.cu/noticias/2017/01/03/cubadebate-lo-mas-leido-y-comentado-en-2016/>.
- Damián Gutierrez. 2009.** «UML Diagramas de Paquetes». septiembre. http://www.codecompiling.net/files/slides/UML_clase_05_UML_paquetes.pdf.
- Elena Barragans. 2006.** «INTERNET COMO MEDIO DE COMUNICACIÓN». https://www.google.com.cu/url?sa=t&rct=j&q=&esrc=s&source=web&cd=10&sqj=2&ved=0ahUKEwiz7pPs5ZXUAhUI7IMKHRXnCxAQFghdMAk&url=http%3A%2F%2Fcaguleta.webs.uvigo.es%2FTRABAJOS%2520PARA%2520LA%2520WEB%2FINTERNET%2520caract.%2520y%2520estructura.doc&usg=AFQjCNEFnx8CCbrCNOBr9uLTR4kJp7_rhA&cad=rja.
- Elías Fernández Martín. 2006.** «Web 2.0: sindicación de contenidos [RSS] - Dipòsit Digital de Documents de la UAB». <https://ddd.uab.cat/record/20010>.
- Erich Gamma. 1994.** «Design Patterns Elements of Reusable Object-Oriented Software». [https://sophia.javeriana.edu.co/~cbustaca/docencia/DSBP-2016-03/recursos/Erich%20Gamma,%20Richard%20Helm,%20Ralph%20Johnson,%20John%20M.%20Vlissides-Design%20Patterns_%20Elements%20of%20Reusable%20Object-Oriented%20Software%20%20-Addison-Wesley%20Professional%20\(1994\).pdf](https://sophia.javeriana.edu.co/~cbustaca/docencia/DSBP-2016-03/recursos/Erich%20Gamma,%20Richard%20Helm,%20Ralph%20Johnson,%20John%20M.%20Vlissides-Design%20Patterns_%20Elements%20of%20Reusable%20Object-Oriented%20Software%20%20-Addison-Wesley%20Professional%20(1994).pdf).
- Eva Domínguez. 2005.** «RSS: Una Rápida Solución para Segmentar - El Cuarto BitEl Cuarto Bit». *La Vanguardia*. <http://blogs.lavanguardia.com/elcuartobit/rss-una-rapida-solucion-para-segmentar>.
- «Feed - EcuRed». 2017.** Accedido mayo 5. <https://www.ecured.cu/Feed>.
- Fernando Rivero. 2016.** «Informe Mobile en España y en el Mundo 2016». Ejecutivo 1. España: ditrendia. pdf. http://www.amic.media/media/files/file_352_1050.pdf.
- Glenford J. Myers. 2011.** *The Art of Software Testing*. Second Edition. John Wiley & Sons, Inc. http://www.51testing.com/N_download/lib/TestingTechDL/ArtofSoftwareTesting.pdf.

- Ian Sommerville. 2011.** «Ingeniería de Software-Ian Sommerville-9 edición-español[1]». http://www.academia.edu/15366832/Ingenieria-de-Software-Ian-Somerville-9-edicion-esp%C3%B1ol_1_.
- «Inoreader - The content reader for power users who want to save time. » 2017. *Inoreader*. Accedido mayo 15. <https://www.inoreader.com>.
- ITU. 2014.** «Evolución de las TIC a escala mundial». <http://itunews.itu.int/es/5089-Evolucion-de-las-TIC-a-escala-mundial-.note.aspx>.
- Joe Hindy. 2016.** «10 best RSS reader apps for Android». *Android Authority*. de abril de. <http://www.androidauthority.com/best-rss-reader-apps-for-android-2-174770/>.
- M. Milinkovich Nov. 2017.** «About the Eclipse Foundation». <http://www.eclipse.org/org/>.
- Microsoft. 2017. «Inténtelo: Crear un lector de noticias RSS». <https://msdn.microsoft.com/es-es/library/cc294852.aspx>.
- «**News Republic**». 2017. Accedido mayo 15. <http://www.news-republic.com/>.
- «News360: Your personalized news reader app». 2017. Accedido mayo 15. <https://news360.com/>.
- Nion, Noin. 2017.** «gReader - Best RSS reader app for Android for Feedly/TOR». *noinnion*. Accedido mayo 15. <http://noinnion.com/greader/>.
- ONE. 2017.** «Oficina Nacional de Estadísticas. Cuba». Accedido mayo 29. <http://www.one.cu/aec2015.htm>.
- Operating system market share. 2017. «Operating system market share». <https://www.netmarketshare.com/operating-system-market-share.aspx?qprid=8&qpcustomd=1>.
- «**Prueba unitaria - Prueba-unitaria.pdf**». 2017. Accedido mayo 17. <http://umh2818.edu.umh.es/wp-content/uploads/sites/884/2016/02/Prueba-unitaria.pdf>.
- Real Academia Española. 2017.** «DLE: internet - Diccionario de la lengua española - Edición del Tricentenario». <http://dle.rae.es/?id=LvskgUG>.
- Roger S. Pressman. 2010.** *Ingeniería del Software. Un Enfoque Práctico*. SÉPTIMA EDICIÓN. México: The McGraw-Hill Companies, Inc. <http://cotana.informatica.edu.bo/downloads/Id-Ingenieria.de.software.enfoque.practico.7ed.Pressman.PDF>.

Sánchez Tarragó, Nancy. 2007. «Sindicación de contenidos con canales RSS: aplicaciones actuales y tendencias». *ACIMED* 15 (3): 0-0.

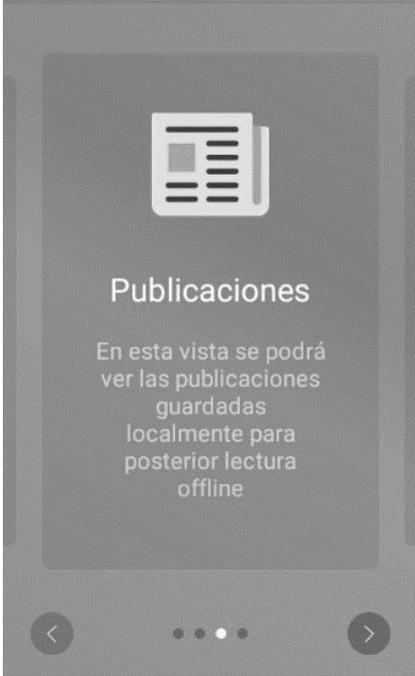
Simón Peña Fernández, Iñaki Lazkano Arrillaga, y Antxoka Agirre Maiora. 2014. «El periodismo y la necesidad de estar bien informado. Una propuesta formativa - 42407». febrero. <https://revistas.ucm.es/index.php/HICS/article/viewFile/45036/42407>.

«Software Design Tools for Agile Teams, with UML, BPMN and More». 2017. <https://www.visual-paradigm.com/>.

Tim Bray. 2006. «Extensible Markup Language (XML)». W3C® (MIT, INRIA, Keio). <http://www.w3pdf.com/W3cSpec/XML/2/REC-xml11-20060816.pdf>.

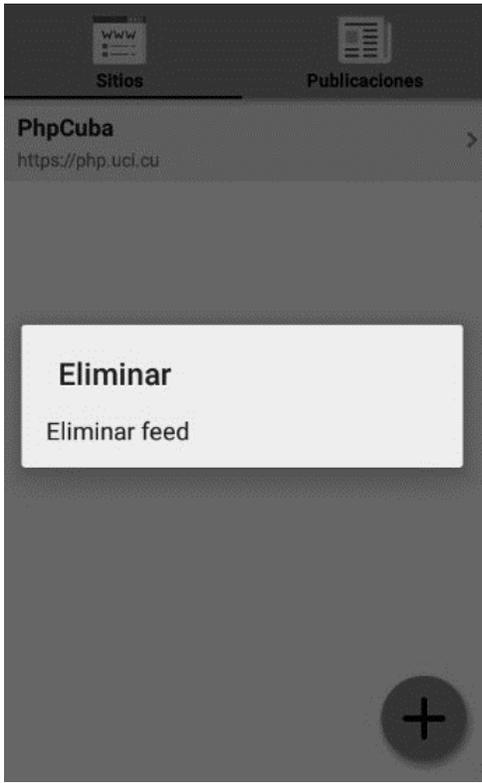
ANEXOS

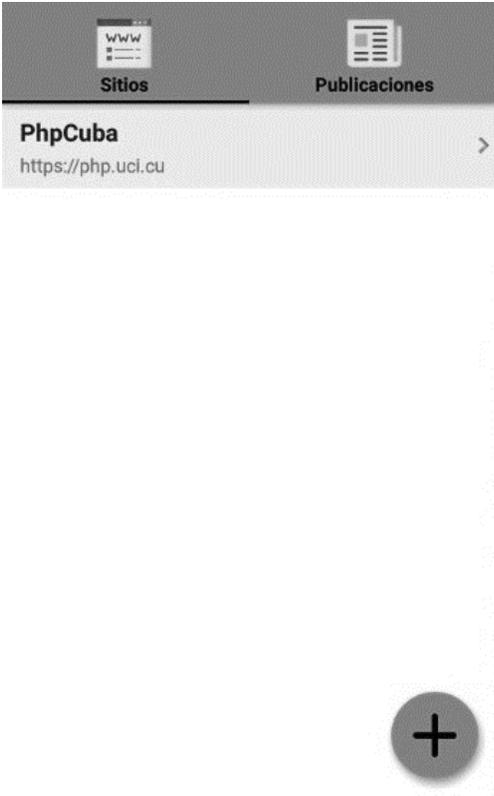
Anexo 1 Historias de Usuario

| | | | |
|---|--|--|--|
| Numero: 1 | | Nombre del requisito: mostrar ayuda | |
| Programador: Elioenai Reyes González | | Iteración Asignada: 1 | |
| Prioridad: media | | Estimado: 1 días | |
| Riesgo en Desarrollo: <ul style="list-style-type: none"> Interrupción del fluido eléctrico. | | Tiempo Real: 1 día | |
| Descripción: La aplicación debe mostrar al usuario una ayuda de navegación en la misma | | | |
| Observaciones: Se visualizarán varias funcionalidades de la aplicación y breve explicación de su uso. | | | |
| Prototipo de interfaz: | | | |
|  | |  | |

| | | | |
|--|--|--|--|
| Numero: 2 | | Nombre del requisito: listar sitios recientes | |
| Programador: Elioenai Reyes González | | Iteración Asignada: 2 | |
| Prioridad: media | | Estimado: 3 días | |
| Riesgo en Desarrollo: <ul style="list-style-type: none">• Fallo de conexión.• Fallo del sitio.• Interrupción del fluido eléctrico. | | Tiempo Real: 1 día | |
| Descripción: <p>La aplicación debe mostrar al usuario el listado de los sitios accedidos recientemente teniendo cuenta los siguientes datos:</p> <ul style="list-style-type: none">• Nombre del sitio• Dirección URL | | | |
| Observaciones: | | | |
| Prototipo de interfaz:  | | | |

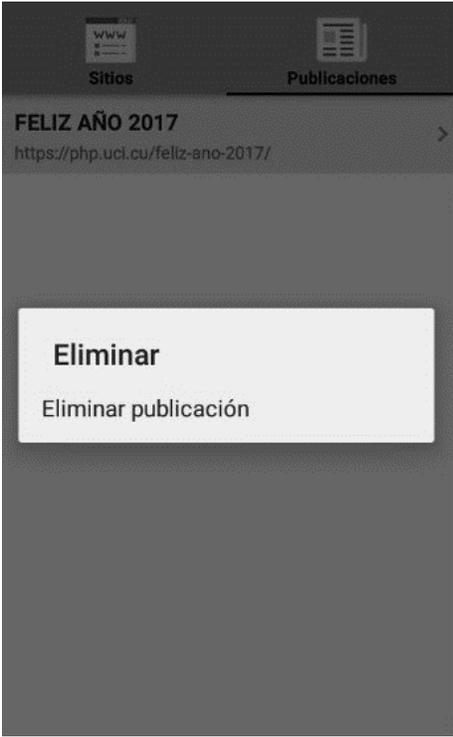
| | | | |
|--|--|--|--|
| Numero: 3 | | Nombre del requisito: adicionar sitio | |
| Programador: Elioenai Reyes González | | Iteración Asignada: 3 | |
| Prioridad: media | | Estimado: 6 días | |
| Riesgo en Desarrollo: <ul style="list-style-type: none">• Fallo de conexión.• Fallo del sitio.• Interrupción del fluido eléctrico. | | Tiempo Real: 3 día | |
| Descripción: <p>La aplicación debe permitir al usuario añadir nuevo atendiendo a los siguientes parámetros:</p> <ul style="list-style-type: none">• Dirección de sitio• Temática• Actualizable o no | | | |
| Observaciones: <p>En caso de que no exista conexión o que el sitio no tenga Web Feed se notificara al usuario.</p> | | | |
| Prototipo de interfaz:  | | | |

| | | | |
|--|--|---|--|
| Numero: 4 | | Nombre del requisito: eliminar sitio | |
| Programador: Elioenai Reyes González | | Iteración Asignada: 3 | |
| Prioridad: media | | Estimado: 1 hora | |
| Riesgo en Desarrollo: <ul style="list-style-type: none">• Interrupción del fluido eléctrico. | | Tiempo Real: 1 hora | |
| Descripción: <p>La aplicación debe permitir al usuario eliminar sitio(s).</p> | | | |
| Observaciones: | | | |
| Prototipo de interfaz: | | | |
|  <p>The image shows a mobile application interface. At the top, there are two tabs: 'Sitios' and 'Publicaciones'. Below the tabs, a card displays 'PhpCuba' with the URL 'https://php.ucl.cu'. A dialog box is overlaid on the screen with the title 'Eliminar' and the text 'Eliminar feed'. At the bottom right, there is a circular button with a plus sign.</p> | | | |

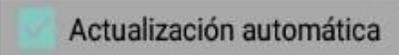
| | | | |
|--|--|--|--|
| Numero: 6 | | Nombre del requisito: guardar sitio | |
| Programador: Elioenai Reyes González | | Iteración Asignada: 3 | |
| Prioridad: media | | Estimado: 2 horas | |
| Riesgo en Desarrollo: <ul style="list-style-type: none">• Interrupción del fluido eléctrico. | | Tiempo Real: 4 horas | |
| Descripción: <p>La aplicación debe permitir guardar el sitio que el usuario prefiera de forma fuera de línea.</p> | | | |
| Observaciones: | | | |
| Prototipo de interfaz:  <p>The image shows a mobile application interface. At the top, there are two tabs: 'Sitios' (selected) and 'Publicaciones'. Below the 'Sitios' tab, there is a list item for 'PhpCuba' with the URL 'https://php.uci.cu' and a right-pointing arrow. At the bottom right of the screen, there is a circular button with a plus sign (+).</p> | | | |

| | |
|---|------------------------------|
| Numero: 8 Nombre del requisito: mostrar publicación | |
| Programador: Elioenai Reyes González | Iteración Asignada: 4 |
| Prioridad: media | Estimado: 2 días |
| Riesgo en Desarrollo: <ul style="list-style-type: none">• Fallo de conexión.• Fallo del sitio.• Interrupción del fluido eléctrico. | Tiempo Real: 1 día |
| Descripción: <p>La aplicación debe permitir al usuario ver la publicación en su totalidad.</p> | |
| Observaciones: <p>Si no existe el directorio de salva no será posible cargar la publicación para lectura fuera de línea.</p> | |
| Prototipo de interfaz:  | |

| | |
|---|------------------------------|
| Numero: 9 Nombre del requisito: guardar publicación | |
| Programador: Elioenai Reyes González | Iteración Asignada: 4 |
| Prioridad: media | Estimado: 2 días |
| Riesgo en Desarrollo: <ul style="list-style-type: none">• Fallo de conexión.• Fallo del sitio.• Interrupción del fluido eléctrico. | Tiempo Real: 1 día |
| Descripción: <p>La aplicación debe permitir al usuario guardar publicación de su preferencia para posterior lectura fuera de línea.</p> | |
| Observaciones: <p>La aplicación creará un directorio donde realizará las salvas de las publicaciones de ser borrado este directorio no será posible cargar la publicación para lectura fuera de línea.</p> | |
| Prototipo de interfaz:  <p>The screenshot shows a mobile application interface for 'PhpCuba' (Comunidad Cubana de PHP). At the top, there is a header with the text 'PhpCuba' and 'Comunidad Cubana de PHP'. Below the header is a dark navigation bar with a hamburger menu icon and the text 'Menú'. The main content area displays a large text message: 'FELIZ AÑO 2017'. Below this message, it says 'Osay / 17 diciembre, 2016'. There is a small image showing '2016' with a 'Salva realizada!!' (Save successful!!) notification overlay. At the bottom, there is a partial view of another message: 'Nuestra comunidad les desea un feliz fin de año y un próspero año'. A download icon is visible in the bottom right corner of the screenshot.</p> | |

| Numero: 10 | | Nombre del requisito: eliminar publicación | |
|---|--|---|--|
| Programador: Elioenai Reyes González | | Iteración Asignada: 4 | |
| Prioridad: media | | Estimado: 3 horas | |
| Riesgo en Desarrollo: <ul style="list-style-type: none">• Interrupción del fluido eléctrico. | | Tiempo Real: 4 horas | |
| Descripción: <p>La aplicación debe permitir al usuario eliminar publicaciones.</p> | | | |
| Observaciones: | | | |
| Prototipo de interfaz: | | | |
|  | | | |

| | |
|---|---|
| Numero: 11 | Nombre del requisito: organizar sitios |
| Programador: Elioenai Reyes González | Iteración Asignada: 5 |
| Prioridad: media | Estimado: 1 días |
| Riesgo en Desarrollo: <ul style="list-style-type: none"> Interrupción del fluido eléctrico. | Tiempo Real: 1 día |
| Descripción: | |
| La aplicación debe permitir organizar los sitios guardados según la temática o preferencia elegida por el usuario | |
| Observaciones: | |
| Si una temática no posee sitios clasificados en la misma se eliminará automáticamente. | |
| Prototipo de interfaz: | |
|  | |

| | | | |
|--|--|---|--|
| Numero: 12 | | Nombre del requisito: actualizar publicaciones de los sitios | |
| Programador: Elioenai Reyes González | | Iteración Asignada: 5 | |
| Prioridad: media | | Estimado: 3 días | |
| Riesgo en Desarrollo: <ul style="list-style-type: none"> • Fallo de conexión. • Fallo del sitio. • Interrupción del fluido eléctrico. | | Tiempo Real: 5 día | |
| Descripción: La aplicación debe permitir actualizar sistemáticamente si es preferencia del usuario las publicaciones de los sitios previamente guardados por usuario | | | |
| Observaciones: | | | |
| Prototipo de interfaz: | | | |
|  | | | |

Anexo 2 Entrevista

Con el objetivo de recoger información para el desarrollo del Lector de noticias para dispositivos móviles con Sistema Operativo NovaDroid se aplica el cuestionario de la entrevista.

Preguntas realizadas:

- ¿Cómo surge la propuesta de un sistema para la lectura de publicaciones en sitios web a través de dispositivos móviles? ¿Porque?
- ¿Qué objetivo se persigue con el desarrollo de este sistema para la lectura de publicaciones en sitios web a través de dispositivos móviles?
- ¿Cuáles son los requisitos funcionales para el sistema?
- ¿Cuáles son los requisitos no funcionales para el sistema?

Anexo 3 Acta de Aceptación

 **Acta de aceptación de productos de trabajo**

ACTA DE ACEPTACIÓN DE PRODUCTOS DE TRABAJO

En cumplimiento del **Convenio de colaboración** establecido entre el **Centro de Software Libre (CESOL)** y el estudiante **Elioenai Reyes González** de la Facultad 1 de la Universidad de las Ciencias Informáticas y en función de la ejecución del proyecto: **Lector de noticias para dispositivos móviles con Sistema Operativo NovaDroid**, se hace entrega de los productos que se relacionan a continuación:

- Aplicación para dispositivos móviles con Sistema Operativo NovaDroid.

La parte Cliente, luego de haber revisado los productos de trabajos relacionados anteriormente procede a firmar la aceptación de los mismos en total conformidad.

| Entrega | Recibe |
|---|--|
| Nombre y apellidos: Elioenai Reyes González | Nombre y apellidos: Yoandy Pérez Villazón |
| Cargo: Estudiante Facultad 1 | Cargo: Director del Centro de Soluciones Libres |
| Firma:  | Firma:  |

Fecha: 31/05/2017

GLOSARIO DE TÉRMINO

IntelliJ IDEA: es un ambiente de desarrollo integrado para el desarrollo de programas informáticos. Es desarrollado por *JetBrains*³².

Gradle: es un sistema de compilación que reúne en uno solo las mejores prestaciones de otros sistemas de compilación.

Web Feed: es un archivo generado por algunos sitios *web* que contiene una versión específica de la información publicada en esa *web*

RSS: un formato XML para syndicar o compartir contenido en la web.

Plugins: es una aplicación o programa informático que se relaciona con otra para agregarle una función nueva y generalmente muy específica.

³² JetBrains, creador IntelliJ IDEA, es un proveedor de software líder en tecnología que se especializa en la creación de herramientas de desarrollo inteligentes.