

“Universidad de las Ciencias Informáticas”



“Lector de Canales RSS para Dispositivos Móviles.”

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

AUTORES:

Yaima Ricardo Portelles

Javier Rodríguez Rivas

TUTOR:

Ing. Lex Karel Zayas Hernández.

Año 52 de la Revolución

Ciudad Habana, Junio 2010

Declaración de la Autoría

Declaro ser autor de la presente tesis, reconociendo a la Universidad de las Ciencias Informáticas los derechos patrimoniales de esta de manera exclusiva.

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

Yaima Ricardo Portelles.

Firma del Autor

Javier Rodríguez Rivas.

Firma del Autor

Ing. Lex Karel Zayas Hernández.

Firma del Tutor

Dedicatoria

Dedicatoria Yaima

Dedicatoria Javier

Opinión del Tutor del Trabajo de Diploma

Agradecimientos

A nuestros padres por permitirnos la vida y guiarnos hasta aquí; por hacernos hoy, hombres de bien.

A nuestros maestros y profesores que con tanto sacrificio nos han dado un poco más que conocimientos.

A la UCI por ser a lo largo de cinco años más que una escuela. Por dotarnos de tantos conocimientos y experiencias inolvidables.

A todos lo que han tenido que ver con nuestra educación y formación para ser cada día mejores, vaya nuestros más sinceros agradecimientos;

Yaima y Javier.

Frase

Resumen

Las tecnologías de servicio de mensajería inalámbrica tienen un amplio alcance en el mercado mundial. Con su utilización se hace posible la creación de considerables números de servicios que ofrecen grandes ganancias a las compañías que se desempeñan en este sector; tal es el caso de la Unidad Móvil de Negocio de ETECSA(Cubacel), la cual presenta entre sus líneas actuales, el desarrollo de aplicaciones encaminadas a la telefonía móvil, la misma cuenta con algunos de los servicios de valor agregado complementarios, así como algunos de los servicios básicos que pudiese ofrecer, aumentando así las ganancias para la ganancias obtenidas por esta empresa. Teniendo en cuenta lo anteriormente expuesto, se propone el desarrollo de un sistema que haga posible la lectura, visualización y almacenamiento de las diferentes fuentes de lectura de canales RSS para dispositivos móvil, con el fin de aumentar el número de servicios que posee la empresa y aumentando paralelamente las prestaciones recibidas por parte de clientes nacionales y extranjeros, el cual fue desarrollado satisfactoriamente utilizando técnicas y herramientas afines con tal propósito, suplantando las necesidades promedio presentadas por proveedores y clientes.

Tabla de Contenido

Declaración de la Autoría..... 2

Dedicatoria 3

Dedicatoria Yaima..... 4

Dedicatoria Javier..... 5

Opinión del Tutor del Trabajo de Diploma..... 6

Agradecimientos..... 7

Frase 8

Resumen..... 9

Introducción..... 1

Capítulo1: Fundamentación Teórica..... 4

 1.1 Estudio del Estado del Arte: 4

 1.1.1 Servicio de Lector de Canales RSS YomoMedia. 4

 1.1.2 Servicio de Lector de Canal RSS Viigo..... 4

 1.1.3 Servicio de Lector de Canal RSS TouchTicker..... 5

 1.1.4 Servicio de Lector de Canal RSS Blackberry. 5

 1.2 Introducción a la Telefonía Móvil 5

 1.3 Redifusión o Sindicación de contenidos. 8

 1.3.2 Formato RSS 9

 1.3.3 Formato Atom 10

 1.4.1 Metodologías de Desarrollo de Software 11

1.5 Plataforma de Desarrollo. 13

 1.5.1 Java..... 13

 1.6 Tecnologías a Utilizar..... 14

 1.6.1 J2ME (*Java 2 Micro Edition*)..... 14

 1.6.2 J2ME Wireless Toolkit 15

 1.6.3 Interfaz de programación de aplicaciones (API) 15

 1.7 Herramientas de Desarrollo. 19

 1.7.1 Eclipse..... 19

 1.8 Control de Versiones 19

 1.8.1 SVN..... 20

 1.9 Herramienta de Modelado..... 20

 1.9.1 Visual Paradigm 21

 1.10 Conclusiones 21

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA.....	22
2.1 Objetivos estratégicos de la organización	22
2.2 Objeto de Automatización	22
2.3 Propuesta de Sistema.....	22
2.3.1 Flujo de procesos del sistema propuesto.....	22
2.3.2 Arquitectura Propuesta	23
2.5 Análisis comparativo de otras soluciones existentes con la propuesta	26
2.5 Conclusiones	27
CAPÍTULO 3: EXPLORACIÓN, PLANIFICACIÓN y DISEÑO.....	28
3.1 Fase de Exploración.....	28
3.1.1 Historias de Usuario	28
3.2 Fase de Planificación	31
3.2.1 Estimación de esfuerzo por historias de usuario	32
3.2.2 Plan de Iteraciones	32
3.2.3 Plan de duración de las Iteraciones	33
3.2.4 Plan de Entregas	33
3.3 Diseño.....	34
3.3.1 Tarjeta CRC	34
3.4 Conclusiones	34
CAPÍTULO 4: IMPLEMENTACION Y PRUEBA.....	35
4.1 Iteración 1.....	35
4.2 Iteración 2.....	37
4.3 Iteración 3.....	39
4.4 Módulos y Librerías Utilizadas en la Implementación.....	40
4.4.1 Módulo KXML v1.0	40
4.4.2 Módulo Sony Ericsson J2ME SDK 1.2.....	41
4.4.3 J2ME Wireless Toolkit 1.0.4	41
4.5 Fase de Prueba	42
4.5.1 Pruebas de Aceptación.....	42
4.6 Conclusiones	43
Capitulo 5. Estudio de Factibilidad	44
5.1 Modelo matemático COCOMO II	44
5.2 Características del Proyecto	44
5.2.1 Entradas externas	45

5.2.2 Salidas Externas	45
5.2.3 Consultas Externas.....	46
5.2.4 Archivos Lógicos Internos	46
5.2.4 Archivos de Interfaz Externos.....	47
5.2.5 Puntos de Función Desajustados.....	47
5.3 Cálculo de instrucciones fuentes, esfuerzo, tiempo de desarrollo, cantidad de hombres y costo.	48
5.3.1 Cálculo de instrucciones fuentes.....	48
5.4 Cálculo de Esfuerzo Nominal.....	48
5.4.1 Ajuste del Esfuerzo Nominal	49
5.5 Cálculo de Esfuerzo Nominal	49
5.5.2 Calculo del costo total del proyecto.....	52
5.6 Resultados.....	52
5.7 Beneficios tangibles e intangibles	53
5.8 Análisis de costo	53
5.9 Conclusiones	53
Conclusiones Generales	54
Recomendaciones	55
Referencias Bibliográficas	56
Glosario de Términos	58
Anexos.....	61
Anexo #1: Pruebas de Aceptación.....	61

Introducción

El surgimiento de la telefonía móvil acompañada del avance tecnológico figura como uno de los pasos de avance de mayor impacto en el campo de las telecomunicaciones. El alto grado de aceptación por parte de los clientes ha motivado a gran número de empresas a respaldar su desarrollo mientras que otras han surgido específicamente con tal propósito.

Actualmente la telefonía móvil ha llegado a convertirse en una prestación básica para los usuarios lo cual se debe a su eficiencia en términos de conectividad, movilidad, disminución de costos, rapidez, aumento de los servicios y calidad de los mismos. La misma continua incrementando sus potencialidades, llegando incluso a diferir un tanto del objetivo principal de su creación: la transmisión de voz, pero manteniendo el concepto de servir como medio de comunicación.

Entre sus beneficios encontramos la posibilidad de brindar servicios de amplio acceso a redes de computadoras, incluyendo la red de internet, lo cual implica una gran cadena de productores de contenidos, proveedores de servicios y fabricantes de equipos, así como una nueva percepción de la rentabilidad apoyada en la gestión e inversión de la infraestructura, obligando a seguir una serie de estándares de calidad y seguridad que derivan en una mejor relación productor-consumidor.

El crecimiento de las redes de operadoras, la implementación de nuevos servicios y disminución de los costos de las comunicaciones aumenta la competitividad entre compañías proveedoras en un gran número de países y ha dado lugar a la apertura del mercado de las telecomunicaciones a esferas sociales donde años atrás no era posible costear la implantación de la infraestructura ni la explotación de las prestaciones actuales. Todos estos factores han fomentado el uso y auge de la telefonía móvil, la cual poco a poco ha ido suplantando la línea fija tradicional, por otra parte los productores de terminales móviles han incorporado tecnologías que permiten la transmisión de datos, dando a su vez a los proveedores, la posibilidad del uso de datos móviles que dan soporte a nuevos servicios.

Entre estos se encuentran los conocidos como servicios de mensajería que son la base de los servicios de valor agregado como chat, envío de postales, consultas de noticias y partes climatológicos, acceso a correo electrónico, verificación del estado de cuentas bancarias, realización de transmisiones de compra-venta, gestión de los principales canales RSS, transmisión de datos y de voz.

La gestión de canales RSS a través de la telefonía móvil es uno de los servicios que en la actualidad goza de gran aceptación debido a las bondades que ofrece a los clientes. El mismo se utiliza para difundir información de una página web con las últimas actualizaciones que se han hecho en ella. Este tipo de lector es muy útil para cualquier sitio de noticias, foros, weblogs o paginas que emitan contenidos de forma continuada. De esta forma, ya no es necesario visitar la web original cada cierto tiempo en busca de actualizaciones.

Existen varias empresas de dispositivos móviles, en el contorno internacional, que cuentan con este servicio. Tal es el caso de Phone Feeds (<http://www.phonefeeds.com>), My Yahoo! Mobile RSS (<http://mobile.yahoo.com>) y Lite Feeds (<http://www.litefeeds.com>). También se emplea un servicio que envía alertas de SMS al teléfono o al beeper cuando los canales suscritos al lector RSS se actualizan (<http://feedbeep.com>). En el caso específico de nuestro país hasta finales del año 2003 se contaba con dos operadoras que ofrecían servicios de telefonía móvil, C-COM y Cubacel, las mismas operaban con tecnología GSM y TDMA respectivamente, estas compañías se fusionan y pasan a formar parte de la Unidad de Negocios Móviles de ETECSA(Cubacel).

La Universidad de las Ciencias Informáticas junto a la Unidad de Negocios Móviles de ETECSA trabajan en dirección al desarrollo y mejoramiento de los servicios de la telefonía móvil. La mayoría de las herramientas que se utilizan a nivel mundial para la lectura de canales RSS mediante telefonía móvil no son aplicables a nuestro escenario puesto que son herramientas patentadas como software propietario, a lo que se le suma un bloqueo económico impuesto de manera inhumana por el gobierno norteamericano.

Partiendo de la **situación problemática** anterior, se identifica el siguiente **problema de la investigación**: ¿Como brindar un servicio de lectura de canales RSS a dispositivos móviles?

A partir de lo anterior se define como **objeto de estudio** del presente trabajo: La gestión de Canales RSS mediante dispositivos móviles.

Se identifica además como **objetivo general** de la investigación: Implementar un sistema capaz de leer, almacenar y visualizar diferentes fuentes de los canales RSS disponibles en los principales medios de prensa del país.

Se plantea como **campo de acción**: La infraestructura tecnológica y de servicios de La Unidad de Negocios Móviles de ETECSA (Cubacel).

Las **tareas de investigación** para cumplimentar los objetivos propuestos son:

- ✓ Investigación de los principales medios de prensa y de noticias del país que tienen implementado el servicio de canales RSS.
- ✓ Estudio de la documentación existente sobre API's en función del reconocimiento y manipulación de documentos XML.
- ✓ Investigación de las técnicas utilizadas para el envío y recepción de mensajes de notificación. (Push Messages).
- ✓ Estudio y selección de los conceptos y herramientas necesarias para el desarrollo de la investigación.
- ✓ Implementación de la aplicación haciendo uso de J2ME.

Para un mejor entendimiento de los temas a tratar en la investigación se define a continuación la estructura de la misma.

En el **Capítulo 1** se abordan los principales conceptos y términos tratados en la investigación, centrándonos en el marco teórico y el estudio del estado del arte. En el **Capítulo 2** se identifican y se describen los procesos a automatizar, llegando así a la propuesta del sistema a desarrollar enfatizando en la arquitectura y el flujo de procesos. En el **Capítulo 3** se describen las fases de exploración y planificación propias de la metodología de desarrollo utilizada para la implementación del sistema propuesto. En el **Capítulo 4** se especifica y se describe las fases de diseño, implementación y prueba propias de la metodología de desarrollo utilizada. En el **Capítulo 5** se describe el proceso de estudio de factibilidad para la ejecución del sistema propuesto.

Capítulo1: Fundamentación Teórica

En este capítulo se realiza un análisis de todos los aspectos y conceptos teóricos que sustentan la investigación y se consideran necesarios para entender la misma. Se describen las principales características de la metodología, tecnología y herramientas que se proponen para el desarrollo del software.

1.1 Estudio del Estado del Arte:

En el mundo existen numerosas empresas que se dedican a prestar servicios de valor agregado para la telefonía móvil. Muchas facilitan un servicio de Lector de Canal RSS, utilizando diferentes medios de acceso a los mismos. En nuestro país se ha dado los primeros pasos para concretar prestaciones de este tipo, sin embargo producto de la escasa competitividad entre proveedores dichos servicios no han sido totalmente explotados, lo cual imposibilita realizar un estudio profundo de los mismos a nivel nacional.

A continuación, se detallan servicios de Lector de Canal RSS para dispositivos móviles de diferentes proveedores y nacionalidades:

1.1.1 Servicio de Lector de Canales RSS YomoMedia.

YomoMedia: Requiere que el usuario se cree una cuenta en YomoMedia. Una vez registrado puede disfrutar de los cientos de canales de vídeo, de audio o de noticias disponibles, así como suscribirse a nuevos canales RSS y marcarlos para que éstos sincronicen con el móvil. Una vez sincronizado, tendrá noticias, vídeos y podcasts de audio en su móvil organizados por canales y categorías.

1.1.2 Servicio de Lector de Canal RSS Viigo.

Viigo: Este servicio ya viene bien surtido de canales de todo tipo, repletos de noticias para todos los gustos e inquietudes. Permite que accedas al contenido más popular de la red. Evidentemente, también puedes suscribirte a canales RSS de su elección, no presenta limitaciones al contenido ofrecido por Viigo. El acceso al contenido es inmejorable, accediendo a mucho contenido desde una sola y única interfaz.

1.1.3 Servicio de Lector de Canal RSS TouchTicker.

TouchTicker: Ofrece una barra en la que se va mostrando los titulares de las últimas noticias provenientes de tus canales RSS, pero moviendo de derecha a izquierda. Si te interesa una noticia, puedes abrir TouchTicker. Una vez abierto tendrás acceso a la noticia completa, con texto y fotos.

1.1.4 Servicio de Lector de Canal RSS Blackberry.

Blackberry: El usuario lo único que tiene que hacer es ir a la web y abrirse una cuenta de manera gratuita, y activar sólo las que le interesan ver desde su dispositivo móvil. A continuación instalas un programa en tu dispositivo móvil que sincroniza con littlefeeds y el cual verás de una manera fácil y sencilla ya que la información está adaptada para que pueda ser visualizada en pantalla sin problemas.

1.2 Introducción a la Telefonía Móvil

La telefonía móvil ha revolucionado el área de las comunicaciones, redefiniendo cómo percibimos las comunicaciones de voz. Tradicionalmente, los teléfonos celulares se mantuvieron fuera del alcance de la mayoría de los consumidores debido a los altos costos involucrados.

Como resultado, las compañías proveedoras de servicios invirtieron tiempo y recursos en encontrar nuevos sistemas de mayor capacidad, y por ende, menor costo. Los sistemas celulares se están beneficiando de estas investigaciones y han comenzado a desarrollarse como productos de consumo masivo.

La telefonía celular es un sistema de comunicación telefónica totalmente inalámbrica. Se verá, como los sonidos se convierten en señales electromagnéticas, que viajan a través del aire, siendo recibidas y transformadas nuevamente en mensajes.

Los teléfonos móviles se han convertido en todo un fenómeno social. Antes eran caprichos de algunos afortunados, que hicieron de su uso un símbolo de exclusividad económica. Ahora son objetos casi imprescindibles en la vida de millones de personas. Con ellos se comunican en cualquier parte del planeta, trabajan y pasan un tiempo de ocio. En algunos casos, su utilidad roza incluso la adicción. Y es que sus nuevos modelos, colores atractivos, melodías rompedoras y servicios revolucionarios cautivan, cada día, a muchos consumidores, sobre todo jóvenes, que desean estar siempre a la vanguardia de las últimas tendencias. De hecho, hoy lo que se lleva traspasa incluso las barreras de la imaginación. Celulares que son completos ordenadores en miniatura. Captan fotografías de alta resolución, disponen de correo electrónico. El teléfono móvil es un dispositivo

inalámbrico electrónico que permite tener acceso a la red de telefonía celular o móvil. Se denomina celular debido a las antenas repetidoras que conforman la red, cada una de las cuales es una célula, si bien existen redes telefónicas móviles satelitales. Su principal característica es su portabilidad, que permite comunicarse desde casi cualquier lugar. Aunque su principal función es la comunicación de voz, como el teléfono convencional, su rápido desarrollo ha incorporado otras funciones como son cámara fotográfica, agenda, acceso a Internet, reproducción de vídeo e incluso GPS y reproductor mp3.

La empresa CUBACEL ha concebido e iniciado el desarrollo de un programa dirigido a masificar el uso de estas tecnologías, que hasta la fecha se ha extendido solo en una pequeña porción del sector empresarial y del turismo, fundamentalmente.

En la actualidad se incluyen servicios de valor añadido de mensajes cortos (SMS), “Call Centers” y acceso móvil a Internet, y recientemente comenzó a operar una nueva empresa, CARITEL, con el sistema GSM, que facilita a los visitantes europeos el uso de sus celulares en el país.

Servicios de valor Agregado

El mundo de los servicios de valor agregado es basto y diverso, englobando distintas necesidades y amplias porciones del mercado de telefonía móvil. La gama de servicios es grande y brinda potencialidades que pueden ser explotadas en áreas de prioridad baja tales como el entretenimiento o en temas más relevantes para la actividad laboral y cotidiana. El servicio de valor agregado basado en MMS, es la instrumentación que se hace del servicio básico MMS para llevar a cabo un modelo de negocio habilitado por las capacidades de esta tecnología.

Un simple mensaje SMS que se enviaba desde un teléfono celular a otro se convirtió en toda una industria de aplicaciones con diversos servicios que incluían noticias, votaciones, chat, partes climatológicos, envío de postales entre otros. A su vez, con el surgimiento de los MMS, los usuarios se beneficiaron de nuevas aplicaciones que incluyen una combinación de imágenes, sonidos, videos y texto, experimentando una experiencia más agradable.

Existen cuatro tipos de servicios de valor agregado, entre los cuales se encuentran:

1. Servicios basados en el contenido. No poseen ningún tipo de procesamiento, encapsulado o tratamiento.
2. Servicios básicos. Pueden ser de dos tipos:

- Aplicaciones de entretenimiento. Evolución natural de los servicios actuales de descarga de los elementos de entretenimiento para personalizar los terminales de los usuarios (logos, melodías, refrescadores de pantalla).
 - Aplicaciones de almacenamiento de imágenes fijas. Existen dos vertientes: la publicación de fotografías u otros elementos en álbumes Web, o el almacenamiento en un espacio privado del usuario para evitar el consumo de recursos en el terminal.
3. Servicios complementarios. Permiten la creación de contenidos por el usuario como el envío de postales o felicitaciones, y los compositores de mensajes completos, ofreciendo al usuario repositorios de contenidos elementales como imágenes y sonidos.
4. Servicios de información multimedia. Son similares a los servicios de alertas SMS, pero explotan las capacidades ofrecidas por la tecnología MMS. Entre ellos se encuentran los servicios de suscripción, los servicios de noticias, los servicios financieros, partes climatológicos.

Short Message Service o SMS

Un Mensaje Corto (Short Message) es un mensaje de texto breve enviado desde un dispositivo móvil MO (Mobile Originated), hacia un dispositivo móvil MT (Mobile Terminated), a través de un SMS. Un SM estándar consiste en 160 caracteres alfanuméricos máximo, aunque se pueden enviar mensajes con mayor cantidad de caracteres utilizando compresión de datos.

Los SMS no son enviados punto a punto, de manera que se hace necesaria una entidad en la red inalámbrica encargada del proceso de envío. Esta entidad es conocida como Centro de Servicio de Mensajes Cortos (Short Message Service Center o SMSC) y es responsable de la transmisión, almacenamiento y reenvío de SM entre Entidades de Mensajería Corta (*Short Message Entity o SME*).

Global System for Mobile Communications o GSM

El Sistema Global para Comunicaciones Móviles (*Global System for Mobile Communications o GSM*). Este estándar soporta múltiples servicios, destacando entre ellos el Servicio de Mensajería Corta (*Short Message System o SMS*) el cual es un servicio de telefonía móvil que cuenta con aceptación universal, posibilitando la transmisión de mensajes cortos (SM) alfanuméricos entre usuarios móviles y sistemas externos como correo electrónico, paging4 y sistemas de correos de voz (IEC, 2007).

Como parte del proceso de evolución del estándar GSM, la versión 97 del mismo, desarrollada por 3GPP (*3rd Generation Partnership Project*) hace posible el surgimiento del Servicio General de Radio-Paquetes (*General Packet Radio Service* o *GPRS*), en el cual fueron adicionadas capacidades de empaquetamiento de datos. Con el surgimiento de GPRS también surgen nuevos servicios, pudiéndose mencionar entre los más conocidos el Servicio de Mensajes Multimedia (*Multimedia Message Service* o *MMS*), diseñado específicamente para GPRS (IEC, 2007).

General Packet Radio Service o GPRS

General Packet Radio Service (GPRS) o Servicio General de Paquetes vía Radio es una extensión del Sistema Global para Comunicaciones Móviles (*Global System for Mobile Communications* o *GSM*) para la transmisión de datos no conmutada (o por paquetes).

GPRS da mejor rendimiento a la conmutación de paquetes de servicios, en contraposición a la conmutación de circuitos, donde una cierta calidad de servicio (QoS) está garantizada durante la conexión. Por este motivo, se considera más adecuada la conexión conmutada para servicios como la voz que requieren un ancho de banda constante durante la transmisión, mientras que los servicios de paquetes como GPRS se orientan al tráfico de datos.

1.3 Redifusión o Sindicación de contenidos.

La sindicación de contenido es un proceso mediante el cual un productor o un distribuidor de contenidos en Internet, proporciona información digital a un suscriptor o una red de suscriptores.

Este proceso se apoya en una clase de programas denominados comúnmente agregadores o lectores de feeds, que permiten leer las fuentes de sindicación fuente preseleccionada para la selección de contenidos determinar la información que corresponde a cada canal y colocar a disposición del usuario, en correspondencia con su perfil, la información completa a la que previamente se suscribió.

La redifusión de contenidos consiste en disponer parte del contenido de una página web a otros sitios o suscriptores individuales. El estándar de sindicación web más extendido es RSS, seguido por Atom. Los programas informáticos compatibles con alguno de estos estándares consultan periódicamente una página con titulares que enlazan con los artículos completos en el sitio web original. A diferencia de otros medios de comunicación, los derechos de redifusión de contenidos web suelen ser gratuitos, y no suele mediar un contrato entre las partes sino una licencia de normas de uso.

Las Fuentes o canales RSS son uno de los avances más importantes en la distribución y difusión de la información a través de internet. RSS permite a cualquier persona estar permanentemente informada con un mínimo esfuerzo.

RSS (*Rich Site Summary*) brinda a los usuarios la posibilidad de recibir directamente en su lector de canal RSS información actualizada de las fuentes a las cuales está suscrito, sin necesidad de visitar periódicamente la web original en busca de actualizaciones.

Ventajas de los Lectores de canales RSS.

- ✓ Permite la recepción de las actualizaciones de las páginas web de noticias y blog disminuyendo el esfuerzo por parte de los usuarios en encontrar dichas actualizaciones en las fuentes originales.
- ✓ Permite al usuario tomar la decisión sobre las fuentes de las cuales desea mantenerse actualizado.
- ✓ Los lectores de canales RSS están libre de SPAM, pues no es necesario que el usuario informe su correo electrónico para avisos de actualizaciones.
- ✓ Los lectores de canales RSS permiten cancelar la suscripción de la fuente web de manera rápida y sencilla.
- ✓ Tanto el servicio de lectura de canal RSS como las herramientas para llevar a cabo dicha tarea son libre de costo.

Desventajas de los Lectores de canales RSS.

- ✓ Algunas de las herramientas para la lectura de canales RSS están disponibles solo en idioma inglés.
- ✓ No todos los lectores de canales RSS muestran las fotos.

1.3.1 Formato RSS

RSS son las tres letras que se utilizan como acrónimo para denominar diferentes estándares de la familia XML, desarrollados con vista a lograr la sindicación de contenidos.

Estándares RSS:

- ✓ RDF Site Summary (RSS 0.9 y 1.0) que se basa en la especificación RDF (*Resource Description Framework*) de metadatos, con el objetivo de nutrirse de titulares obtenidos de Webs de terceros.

- ✓ *Rich Site Summary* (RSS 0.91) es un formato de texto, estándar y público, basado en el modelo de metadatos RDF, que permite distribuir titulares de noticias y contenidos en Internet de forma automatizada.
- ✓ *Really Simple Syndication* (RSS 2.0) es el último de los formatos de RSS, no utiliza RDF y en consecuencia tiene una estructura mucho más simple.

Estas versiones presentan diferencias, fundamentalmente porque cada una de ellas se desarrolló por empresas diferentes: diferente grado de complejidad, variabilidad en el nombre, en las fechas de creación, en su estabilidad y uso.

Aún cuando existen varios estándares, puede decirse que para el uso son más recomendables aquellos vigentes y estables: RSS 1.0, la versión más fácil de usar, para aplicaciones basadas en RDF; y RSS 2.0, la versión más completa, para un uso general.

1.3.2 Formato Atom

Se reconoce Atom como un formato mucho más flexible que los referidos anteriormente, capaz de trabajar con datos mucho más complejos en forma más simple, además brinda la posibilidad de crear software para organizar y filtrar los contenidos. Presenta la información a texto completo.

Este formato nació para resolver la confusión creada por la existencia de estándares similares para la distribución de contenidos. Pero más que eliminar el problema de la multiplicidad de estándares, ha creado un formato nuevo, condenado a convivir con los que pretendía sustituir.

Atom, como formato, ha desarrollado dos versiones: la 0.3 y la 1.0. La primera de las versiones nunca se completó. Aún cuando esta versión se lanzó como una prueba, se difundió ampliamente y tuvo una gran aceptación en el mercado.

Sindicación de contenidos web

La sindicación de contenidos es un proceso dirigido a organizar y difundir la información disponible en la Web, parte del principio de suscripción y se auxilia de un conjunto de programas a los que se les denomina agregadores.

Estas herramientas permiten agilizar la navegación en la Web y reducir el tiempo utilizado en recorrer cada una de las fuentes Web que consulta un usuario, acción en la que se emplea bastante tiempo y que en ocasiones no arroja resultado alguno.

Luego de un estudio de las características de dos de los formatos de Sindicación de contenidos existentes a nivel mundial, RSS y Atom, se escoge el formato RSS como soporte y base de la investigación debido a sus ventajas sobre el formato Atom y ser por consecuencia el formato a utilizar a petición del cliente.

1.4 Metodologías de Desarrollo.

1.4.1 Metodologías de Desarrollo de Software

Cualquier proyecto de desarrollo de software, para alcanzar el éxito, debe regirse por una metodología de desarrollo la cual controle el ciclo de vida del software. Esta debe ser capaz de establecer un entorno que mejore la productividad de los desarrolladores y genere la creación de mejores productos de software, aumentando la calidad de los mismos y disminuyendo posibles retrasos en el tiempo de entrega, la detección tardía de errores y costos innecesarios, evitando así la insatisfacción del cliente.

Actualmente existen varias metodologías de desarrollo de software y diferentes aspectos a tener en cuenta a la hora de elegir la que mejor se adapta a un proyecto específico, como son el tamaño del mismo, la complejidad y su tiempo de duración. Las metodologías de desarrollo de software pueden tener varias clasificaciones. Se agrupan en: estructuradas, orientadas a objeto y para sistemas de tiempo real; o también en metodologías ligeras o ágiles y pesadas o robustas.

- ✓ Las Metodologías Orientadas a Objetos tienen como esencia la identificación y organización de conceptos del dominio de la aplicación, a fin de obtener aquellos que se requieren en el dominio de la solución, y no tanto su representación final en un lenguaje de programación. Se caracterizan por la fácil división del sistema en subsistemas independientes, la reutilización de componentes y ser iterativas e incrementales. Algunos exponentes de estas metodologías, basadas en UML además, son RUP, OPEN y METRICA.
- ✓ Las Metodologías para Sistemas de Tiempo Real están más orientadas al control que a los datos. Se caracterizan por la priorización y gestión de procesos concurrentes, comunicación y sincronización entre tareas, respuesta ante eventos externos, manejo de interrupciones y tolerancia a fallos. Un ejemplo de esta metodología es MAST.
- ✓ Las Metodologías Pesadas o Robustas están diseñadas para proyectos complejos, con un gran número de personas involucradas. Se centran en el control del proceso, estableciendo las actividades involucradas, los artefactos a producir y las herramientas y notaciones que se utilizarán. Entre estas se encuentran RUP y MSF.

✓ Las Metodologías Ligeras o Ágiles dan gran peso a la interacción con el cliente y al desarrollo incremental y acelerado del software en iteraciones muy cortas, contando generalmente con un equipo pequeño de desarrolladores. Tienen como base de sus resultados la comunicación con los usuarios involucrados en el proceso. Algunos de sus exponentes más destacados son XP y RAD.

Metodología XP

XP Consiste en crear reglas y prácticas simples, ingenuas o extrañas al principio, fácilmente adoptadas luego, apoyadas unas en otras, con reducción de actividades improductivas.

Metodológicamente cuestiona la forma actual de producir software, con realimentación tardía, prueba al final, descubrimiento de errores en producción. Reexamina las prácticas corrientes de desarrollo. Es una de las varias "metodologías livianas" surgidas para bajar costos de desarrollo.

Características de XP

- ✓ Acepta cambios de requerimientos aún tardíos en el ciclo de desarrollo.
- ✓ Integra gerentes, clientes y desarrolladores en la búsqueda de calidad en el software.
- ✓ Mejora el proyecto en comunicación, simplicidad, realimentación y emprendimiento.
- ✓ Mantiene el diseño simple y claro.
- ✓ Ensayo el software desde el primer día.
- ✓ Entrega temprano.
- ✓ Implementa los cambios al ir siendo sugeridos.
- ✓ Introduce un cambio en la forma de programar.
- ✓ No ahorra en hardware, construye programas entendibles y extensibles.
- ✓ Automatiza las pruebas: escribe código de prueba antes, durante y después de la programación: antes, al definir la funcionalidad; durante, al descubrir errores; después, en la integración.
- ✓ No tropieza dos veces con la misma pulga (bug).
- ✓ Enfatiza la satisfacción del cliente y el trabajo en equipo.

1.5 Plataforma de Desarrollo.

1.5.1 Java

Java es un lenguaje de programación con el que podemos realizar cualquier tipo de programa. En la actualidad es un lenguaje muy extendido y cada vez cobra más importancia tanto en el ámbito de Internet como en la informática en general. Está desarrollado por la compañía Sun Microsystems con gran dedicación y siempre enfocado a cubrir las necesidades tecnológicas más punteras.

Una de las principales características por las que Java se ha hecho muy famoso es que es un lenguaje independiente de la plataforma. Es una ventaja significativa para los desarrolladores de software. Esto lo consigue porque se ha creado una Máquina de Java para cada sistema que hace de puente entre el sistema operativo y el programa de Java y posibilita que este último se entienda perfectamente.

La independencia de plataforma es una de las razones por las que Java es interesante para Internet, Java está desarrollándose incluso para distintos tipos de dispositivos además de computadoras como móviles, agendas y en general para cualquier cosa que se le ocurra a la industria.

La versatilidad y eficiencia de la tecnología Java, la portabilidad de su plataforma y la seguridad que aporta, la han convertido en la tecnología ideal para su aplicación a redes.

Java le permite jugar en línea, participar en sesiones de chat con personas de todo el mundo, calcular los intereses de hipoteca y ver imágenes en tres dimensiones, entre otras muchas aplicaciones. Es también esencial para las aplicaciones de intranet y otras soluciones de comercio electrónico que constituyen la base informática de las empresas.

1.6 Tecnologías a Utilizar.

1.6.1 J2ME (Java 2 Micro Edition)

J2ME es la plataforma basada en el lenguaje Java que Sun Microsystems ha creado para la programación de dispositivos inalámbricos pequeños como teléfonos celulares, paginadores y PDA.



Los principales componentes de la Plataforma Java 2, Edición micro (*plataforma J2ME*) son *Connected Device Configurations*, *Connected Limited Device Configurations* y *Mobile Information Device Profiles*, así como otras muchas herramientas y tecnologías que llevan las soluciones Java a los mercados de consumo y dispositivos integrados.

Las tecnologías J2ME contienen un JRE altamente optimizado, especialmente desarrollado para el mercado de gran consumo, abarca una amplia gama de aparatos de tamaño muy reducido y permite ejecutar programas de seguridad, conectividad y utilidades en tarjetas inteligentes, buscapersonas, sintonizadores de TV y otros pequeños electrodomésticos. Las tecnologías J2ME representan únicamente una parte de la gama de productos de software de Java. Las plataformas Java relacionadas son la Plataforma Java 2, Edición estándar (*plataforma J2SE*) y la Plataforma Java 2, Edición empresa (*plataforma J2EE*). La tecnología Java ofrece, así mismo, métodos de creación de servicios Web, transferencia de información XML, numerosos protocolos de red, kits de herramientas y la aplicación Java Web Start.

La máquina virtual es la base de la plataforma, es el interprete del lenguaje y sobre la cual se han de ejecutar las aplicaciones, también sobre esta máquina virtual corren las configuraciones (CDC y CLDC), las cuales incorporan API's básicas para la creación de aplicaciones y sirven de soporte a los perfiles. Los perfiles incluyen la mayor parte de las clases y API's que se van a utilizar en la programación, como pueden ser instrucciones de entrada y salida o de inicio y terminación de la aplicación.

1.6.2 J2ME Wireless Toolkit

El Java Wireless Toolkit (anteriormente conocido como *Java 2 Platform Micro Edition* es un estado de la caja de herramientas del arte para el desarrollo de aplicaciones inalámbricas que se basan en Conexión de J2ME *Limited Device Configuration* (CLDC) y *Mobile Information Device Profile* (MIDP), y diseñado para funcionar en teléfonos celulares, asistentes personales digitales principales, y otros pequeños dispositivos móviles. El kit de herramientas incluye el entorno de emulación, la optimización del rendimiento y las características de ajuste, la documentación y ejemplos que necesitan los desarrolladores para llevar las aplicaciones inalámbricas eficientes y exitosas en el mercado rápidamente.

Java Wireless Toolkit se puede utilizar como un entorno de desarrollo independiente o con un IDE, como el NetBeans Mobility Pack. Vendedores IDE encontrará que la interfaz unificada emulador simplifica la tarea de incorporar el Java Wireless Toolkit en sus entornos de desarrollo.

El emulador de juego de herramientas cumple plenamente con los equipos pertinentes de la tecnología de compatibilidad de la API, asegurando que todas las API están presentes y reaccionan de manera coherente con las implementaciones compatibles. En el modo independiente, los usuarios pueden configurar las preferencias individuales, construir aplicaciones, crear *Java Archive* (JAR) y *Java Application Descriptor* (JAD), archivos y más, utilizando la interfaz del juego de herramientas KToolbar amistoso, o de su línea de comandos. Cuando se integra con el IDE, utilidades de la guía y las preferencias aparecen en las selecciones de menú del IDE, y también puede ser controlado desde el símbolo del IDE interfaz de la línea. Cuando se utiliza con un IDE, el conjunto de herramientas apoya la depuración a nivel fuente.

1.6.3 Interfaz de programación de aplicaciones (API)

Una interfaz de programación de aplicaciones o API (*Application Programming Interface*) es el conjunto de funciones y procedimientos (o métodos, en la programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

Una interfaz de programación representa una interfaz de comunicación entre componentes de software. Se trata del conjunto de llamadas a ciertas bibliotecas que ofrecen acceso a ciertos servicios desde los procesos y representa un método para conseguir abstracción en la programación, generalmente entre los niveles o capas inferiores y los superiores del software. Uno de los principales propósitos de una API consiste en proporcionar un conjunto de funciones de uso general, por ejemplo, para dibujar ventanas o iconos en la pantalla. De esta forma, los programadores se benefician de las ventajas de la API haciendo uso de su funcionalidad, evitándose el trabajo de programar todo desde el principio. Las API's asimismo son abstractas: el software que proporciona una cierta API generalmente es llamado la implementación de esa API.

1.6.3.1 DOM (*Document Object Model*)

El DOM es una jerarquía de objetos predefinidos que describen los elementos de la página web que está mostrando el navegador, así como otras características del proceso de navegación (como son el historial, el tamaño de la ventana de navegación o el contenido de la barra de estado del navegador). Si no se está familiarizado con la programación orientada a objetos, el concepto de objeto puede resultar algo difuso. Un objeto es, en el fondo, un conjunto de variables y funciones que actúa sobre dichas variables, encapsuladas en un mismo paquete. El acceso a las funciones y a las variables se realiza mediante una interfaz bien definida que aísla al programador de la necesidad de conocer cómo están implementadas internamente dichas funciones. De este modo, la programación orientada a objetos resulta muy intuitiva, y más próxima al conocimiento humano.

A pesar de sus orígenes, DOM se ha convertido en una utilidad disponible para la mayoría de lenguajes de programación (*Java, PHP, JavaScript*) y cuyas únicas diferencias se encuentran en la forma de implementarlo.

DOM transforma todos los documentos XHTML en un conjunto de elementos llamados nodos, que están interconectados y que representan los contenidos de las páginas web y las relaciones entre ellos. Por su aspecto, la unión de todos los nodos se llama árbol de nodos.

Para poder utilizar las utilidades de DOM, es necesario “transformar” la página original. Una página HTML normal no es más que una sucesión de caracteres, por lo que es un formato muy difícil de manipular. Por ello, los navegadores web transforman automáticamente todas las páginas web en una estructura más eficiente de manipular.

1.6.3.2 SAX (*Simple API for XML*)

SAX fue originalmente definido como un API de Java, pero ya ha sido portado a otros lenguajes orientados a objetos. La idea de SAX es sencilla, el analizador lee el documento XML desde el inicio y avisa a la aplicación qué es lo que encontró. Existen muchos parsers XML que implementan el API SAX, incluyendo Xerces, Crimson, Oracle XML Parser y Ælfred.

Proporciona mecanismos para leer desde un documento XML. Es un API orientada a eventos, actualmente en su versión 2.0. No conlleva a la generación de estructuras internas, esto es, no crea un árbol del documento. Deben registrar manejadores de eventos que utilizarán los parsers. Es una alternativa popular al Document Object Model (DOM)

Ventajas de SAX:

- ✓ Se puede comenzar a procesar tan pronto como se recibe.
- ✓ No se tiene que esperar a leer todo el documento antes de actuar sobre los datos.
- ✓ El documento entero no tiene que residir en memoria.
- ✓ Es conveniente para documentos largos.

Funcionamiento:

- ✓ La mayoría de los programas SAX son más específicos y sólo trabajan con ciertos XML.
- ✓ Buscan por elementos o atributos particulares en lugares específicos y responden a ellos de manera particular.
- ✓ La parte complicada de la mayoría de los programas SAX es la estructura de datos para almacenar la información obtenida por el Parser.

Características de DOM.

- ✓ DOM genera el árbol jerárquico en memoria y es posible regresar a modificar nodos.
- ✓ No es rápido ni sencillo.
- ✓ DOM para procesar la información crea un árbol jerárquico.

Características de SAX

- ✓ SAX procesa el documento o información en XML.
- ✓ SAX procesa la información por eventos.
- ✓ SAX procesa la información en XML conforme esta sea presentada (evento por evento), efectivamente manipulando cada elemento a un determinado tiempo, sin incurrir en uso excesivo de memoria.
- ✓ SAX es un "Parser" ideal para manipular archivos de gran tamaño, ya que no ocupa generar un árbol en memoria.
- ✓ Es rápido y sencillo.
- ✓ SAX funciona por eventos no es posible manipular información una vez procesada.

Es mejor DOM cuando:

- ✓ Se requiere modificar la estructura del XML.
- ✓ Se comparte el documento en memoria con otras aplicaciones.
- ✓ El tamaño del documento no es muy grande.

Es mejor SAX cuando:

- ✓ La tarea a realizar requiere mucha memoria y alto rendimiento.
- ✓ No es necesario recorrer la estructura completa del documento.
- ✓ Se requiere ir procesando los elementos del archivo a medida que van llegando.

Debido a las características de SAX y DOM, para el desarrollo de esta investigación se elige SAX, debido a las características que presenta el mismo, las ventajas que presenta con respecto a DOM y el funcionamiento del mismo.

1.7 Herramientas de Desarrollo.

1.7.1 Eclipse

Un IDE (*Integrated Development Environment*) es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica, que provee un marco de trabajo amigable para la mayoría de los lenguajes de programación.

Eclipse es una poderosa plataforma de código abierto que permite integrar diferentes aplicaciones para construir un entorno integrado de desarrollo. Su principal aplicación es JDT (*Java Development Tool* o Herramienta de Desarrollo para Java), además de una serie de *plug-ins* que se encuentran acompañando a la plataforma, tales como: Ant, Compare, Core, CVS, Debug, Help, Jface, Releng, Scripting, Search, SWT, Text, UI, Update, Team y WebDAV (IBM, 2007).

El usuario se comunica con Eclipse por intermedio del *framework* que se inicia al ejecutar Eclipse (*workbench*). El *workbench* es la interfaz de usuario de la plataforma que está compuesto de un conjunto de vistas, editores y perspectivas. Las herramientas integradas a Eclipse operan en archivos del espacio de trabajo (*workspace*) del usuario. El *workspace* consta de uno o más proyectos donde cada uno se mapea a un directorio especificado por el usuario en el sistema de ficheros (IBM, 2007). Eclipse se considera como uno de los IDE más poderosos para el desarrollo de aplicaciones en Java, provee soporte mediante *plug-ins* para trabajar con entornos J2SE (*Java 2 Platform, Standard Edition*) y J2EE (*Java 2 Platform, Enterprise Edition*) utilizados en el desarrollo de este trabajo. Con la adición de estos *plug-ins* se puede facilitar la automatización de diferentes tareas para los desarrolladores, tales son los casos del *plug-in* JBoss IDE y de un *plug-in* que permite la administrar el control de versiones de un proyecto. Seguidamente abordaremos más sobre el mismo.

1.8 Control de Versiones

Se llama control de versiones a la gestión de los diversos cambios que se realizan sobre los elementos de algún producto o una configuración del mismo. Los sistemas de control de versiones facilitan la administración de las distintas versiones de cada producto desarrollado, así como las posibles especializaciones realizadas. Estos, pueden ser clasificados de acuerdo a la manera en que se realiza el almacenamiento del código de las siguientes formas:

Centralizados: cuando existe un repositorio centralizado de todo el código, del cual es responsable un único usuario (o conjunto de ellos). Se facilitan las tareas administrativas a

cambio de reducir la potencia y flexibilidad, pues todas las decisiones fuertes (como crear una nueva rama) necesitan la aprobación del responsable.

Distribuidos: si se aumenta la capacidad de decisión distribuida. Esto da más flexibilidad pero puede dificultar bastante la sincronización (Wikipedia, 2007).

1.8.1 SVN

Subversion es un sistema de control de versiones de código abierto. Subversion maneja ficheros y directorios, así como los cambios hechos sobre ellos a través del tiempo, esta capacidad permite la recuperación de versiones anteriores de la información que se trata o examinar el historial de cambios de dicha información. En este aspecto puede pensarse en subversion como una especie de “máquina del tiempo” para ficheros. Subversion puede operar a través de la red, lo cual permite su uso desde diferentes puestos de trabajo en un equipo de desarrollo, incluso en algunos niveles permite manejar y modificar los mismos datos desde diferentes ubicaciones, lo cual sin dudas incrementa la colaboración laboral, teniendo en cuenta que las modificaciones progresivas no son hechas a través de un único conducto y que no se pone en riesgo la integridad de la información, la cual siempre puede ser recuperada en caso de cambios errados.

Algunos sistemas de control de versiones son además sistemas de control o administración de configuraciones (**Source Code Management o SCM**), estos sistemas son modificados específicamente para manejar grandes cantidades de código fuente y cuentan con diversas funcionalidades específicas del desarrollo de software, así por ejemplo, comprensión de lenguajes de programación, o suministro de herramientas de desarrollo de software. Subversion no forma parte de estos sistemas, en su lugar, es un sistema general que puede ser usado para manejar cualquier colección de ficheros, desde código fuente, archivos de audio, video, imágenes, documentos u hojas de cálculo (Pilato, y otros, 2004).

1.9 Herramienta de Modelado

Las herramientas de modelado, son herramientas que se emplean para la creación de modelos de sistemas que ya existen o que se desarrollarán. Estas permiten crear una representación del sistema, a bajo costo y riesgo mínimo. A bajo costo porque es un conjunto de gráficos y textos que representan el sistema, pero no son el sistema físico real, el cual por supuesto es más costoso. Además minimizan los riesgos, porque los cambios que se deban realizar, por errores o cambios en los requerimientos, se pueden realizar más fácil y rápidamente sobre el modelo que sobre el sistema ya implementado.

1.9.1 Visual Paradigm

Las herramientas de modelado, son herramientas que se emplean para la creación de modelos de sistemas que ya existen o que se desarrollarán. Estas permiten crear una representación del sistema, a bajo costo y riesgo mínimo. A bajo costo porque es un conjunto de gráficos y textos que representan el sistema, pero no son el sistema físico real, el cual por supuesto es más costoso. Además minimizan los riesgos, porque los cambios que se deban realizar, por errores o cambios en los requerimientos, se pueden realizar más fácil y rápidamente sobre el modelo que sobre el sistema ya implementado.

Visual Paradigm para UML (**VP-UML**) es una herramienta UML (*Unified Modeling Language*) profesional, que soporta todo el ciclo de vida de desarrollo de software. Utilizar UML, ayuda a una rápida construcción de aplicaciones de calidad, mejores y con menor coste. Visual Paradigm para UML está diseñado para una amplia gama de usuarios, incluyendo herramientas de modelado y documentación que cubren desde los procesos del negocio hasta las especificaciones detalladas de un sistema, soportando más de 20 tipos de diagramas, minimizando los tiempos invertidos y aumentando la eficacia de modelar. Visual Paradigm puede ejecutarse de diferentes formas, el mismo puede ser instalado, o portable, es multiplataforma y posee variados ambientes, es capaz de generar código para Java y .NET, de manera que pueden desarrollarse fácilmente proyectos escritos en estos lenguajes o hacer ingeniería inversa a partir de su código fuente para generar diagramas. Además, para la selección de esta herramienta, se tomó en cuenta la experiencia del grupo de desarrollo y sus conocimientos sobre la misma.

1.10 Conclusiones

En este capítulo se trataron los principales conceptos emprendidos en la investigación, se realizó un análisis del estado de los servicios de valor agregado, se hizo un estudio sobre la situación actual de Cubacel, y fue detallada la metodología, plataforma, tecnologías y herramientas de desarrollo utilizadas en la construcción del presente software.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

En el presente capítulo se explican los procesos que serán objetos de automatización, así como la propuesta de sistema para el Lector de canales RSS a través de dispositivos móviles, haciendo énfasis en el flujo de procesos del mismo y su arquitectura. También se hace un análisis comparativo de las soluciones existentes con la nueva propuesta planteada en el presente trabajo.

2.1 Objetivos estratégicos de la organización

La Unidad de Negocio Móvil de ETECSA provee de soluciones y servicios a sus clientes dentro del mundo de la telefonía móvil, prestando especial atención al empleo de las nuevas tecnologías de acceso a contenidos, nuevos modelos de telefonía móvil, así como nuevos formatos multimedia. Constantemente actualiza su infraestructura tecnológica garantizando capacidad inmediata de respuesta a los distintos modelos de negocios que puedan presentar operadores, clientes y proveedores de contenidos.

2.2 Objeto de Automatización

Serán objeto de automatización los procesos de lectura, almacenamiento y visualización las diferentes fuentes de canales RSS mencionados anteriormente. Dicho proceso quedaría descrito de la siguiente forma:

Conformación del canal RSS, proceso mediante el cual se crea la información que será consultada por los usuarios luego de realizar su solicitud.

Publicación de Contenido: proceso encargado de la publicación en una interfaz *J2ME* del contenido a consultar por el cliente.

Acceso al Contenido: proceso responsable de permitir el acceso por parte de los clientes a los contenidos específicos que le corresponden.

2.3 Propuesta de Sistema

Se propone el desarrollo y puesta en marcha de un sistema capaz de brindar a los usuarios de la telefonía móvil el consumo de las fuentes actualizadas de los Canales RSS de los principales medios de prensa del país.

2.3.1 Flujo de procesos del sistema propuesto

El usuario accede al servicio de lectura de canales RSS desde su dispositivo móvil. Se le muestra una interfaz con los principales medios de prensa del país que exportan las noticias por un canal RSS tienen. Luego el usuario escoge el medio de prensa del cual desea recibir este servicio, Se le muestra otra interfaz mostrándole el tipo de noticia que

desea leer de ese medio de prensa. El usuario escoge el tipo de noticia que desea recibir en su móvil. El sistema se encargara de buscar el medio de prensa y el tipo de noticia seleccionada por el usuario y con la ayuda de un Parser de código XML se encargara de mostrar al usuario la información deseada mediante una aplicación J2ME.

2.3.2 Arquitectura Propuesta

Arquitectura de 2 capas (Cliente - Servidor):

Esta arquitectura consiste básicamente en un cliente que realiza peticiones a otro programa (el servidor) que le da respuesta. Aunque esta idea se puede aplicar a programas que se ejecutan sobre una sola computadora es más ventajosa en un sistema operativo multiusuario distribuido a través de una red de computadoras.

En esta arquitectura la capacidad de proceso está repartida entre los clientes y los servidores, aunque son más importantes las ventajas de tipo organizativo debidas a la centralización de la gestión de la información y la separación de responsabilidades, lo que facilita y clarifica el diseño del sistema.

La separación entre cliente y servidor es una separación de tipo lógico, donde el servidor no se ejecuta necesariamente sobre una sola máquina ni es necesariamente un sólo programa. Los tipos específicos de servidores incluyen los servidores web, los servidores de archivo, los servidores del correo, etc. Mientras que sus propósitos varían de unos servicios a otros, la arquitectura básica seguirá siendo la misma.

Una disposición muy común son los sistemas multicapa en los que el servidor se descompone en diferentes programas que pueden ser ejecutados por diferentes computadoras aumentando así el grado de distribución del sistema.

Ventajas:

- ✓ Centralización del control: los accesos, recursos y la integridad de los datos son controlados por el servidor de forma que un programa cliente defectuoso o no autorizado no pueda dañar el sistema. Esta centralización también facilita la tarea de poner al día datos u otros recursos (mejor que en las redes P2P).
- ✓ Escalabilidad: se puede aumentar la capacidad de clientes y servidores por separado. Cualquier elemento puede ser aumentado (o mejorado) en cualquier momento, o se pueden añadir nuevos nodos a la red (clientes y/o servidores).
- ✓ Fácil mantenimiento: al estar distribuidas las funciones y responsabilidades entre varios ordenadores independientes, es posible reemplazar, reparar, actualizar, o incluso

trasladar un servidor, mientras que sus clientes no se verán afectados por ese cambio (o se afectarán mínimamente). Esta independencia de los cambios también se conoce como encapsulación.

- ✓ Existen tecnologías, suficientemente desarrolladas, diseñadas para el paradigma de C/S que aseguran la seguridad en las transacciones, la amigabilidad del interfaz, y la facilidad de empleo.

Desventajas:

- La congestión del tráfico ha sido siempre un problema en el paradigma de C/S. Cuando una gran cantidad de clientes envían peticiones simultáneas al mismo servidor, puede ser que cause muchos problemas para éste (a mayor número de clientes, más problemas para el servidor). Al contrario, en las redes P2P como cada nodo en la red hace también de servidor, cuantos más nodos haya, mejor es el ancho de banda que se tiene.
- El paradigma de C/S clásico no tiene la robustez de una red P2P. Cuando un servidor está caído, las peticiones de los clientes no pueden ser satisfechas. En la mayor parte de redes P2P, los recursos están generalmente distribuidos en varios nodos de la red. Aunque algunos salgan o abandonen la descarga; otros pueden todavía acabar de descargar consiguiendo datos del resto de los nodos en la red.
- El software y el hardware de un servidor son generalmente muy determinantes. Un hardware regular de un ordenador personal puede no poder servir a cierta cantidad de clientes. Normalmente se necesita software y hardware específico, sobre todo en el lado del servidor, para satisfacer el trabajo. Por supuesto, esto aumentará el coste.
- El cliente no dispone de los recursos que puedan existir en el servidor. Por ejemplo, si la aplicación es una Web, no podemos escribir en el disco duro del cliente o imprimir directamente sobre las impresoras sin sacar antes la ventana previa de impresión de los navegadores.

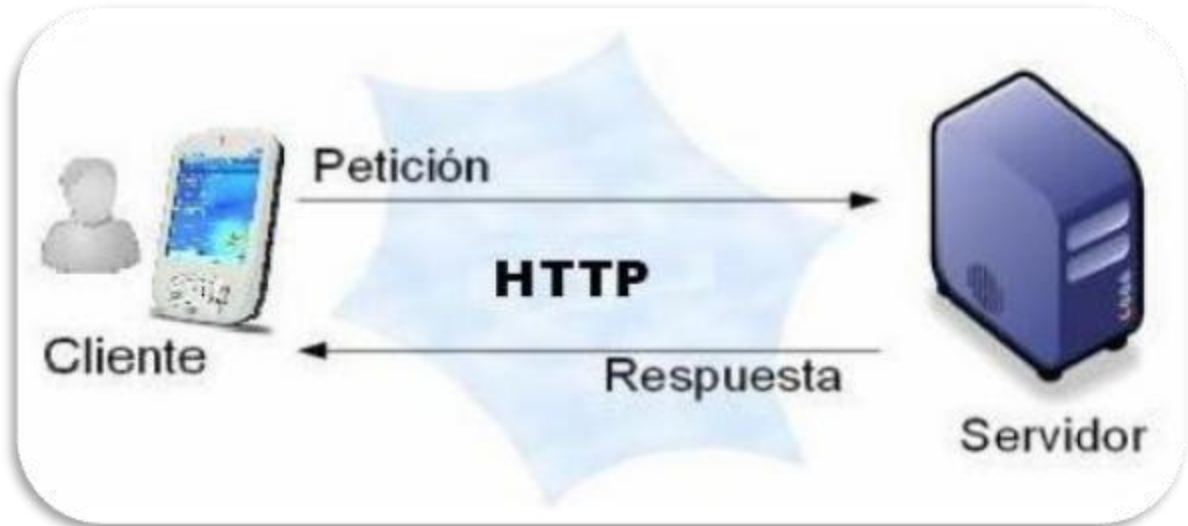


Fig. 1: Ejemplo de Arquitectura de 2 capas (Cliente-Servidor).

2.4 Patrones de Diseño aplicados:

Un patrón de diseño es una buena práctica documentada, o solución, que se ha aplicado con éxito en ambientes múltiples para solucionar un problema que se repite en los sistemas de dichos ambientes y en situaciones específicas.

2.4.1 Patrón Solitario (Singleton):

Este patrón se encarga de garantizar que una clase sólo tiene una única instancia, proporcionando un punto de acceso global a la misma.

2.4.2 Patrón Fachada (Facade):

Con este patrón se proporciona una interfaz unificada de alto nivel que, representando a todo un subsistema, facilita su uso. La “fachada” satisface a la mayoría de los clientes, sin ocultar las funciones de menor nivel a aquellos que necesiten acceder a ellas. El patrón fachada es usado en diferentes situaciones, entre ellas cuando se pretende proporcionar una interfaz simple para un subsistema complejo, o cuando es necesario separar al cliente de los componentes de un subsistema, reduciendo así el número de objetos con los que el cliente trata, y facilitando el uso del subsistema.

2.4.3 Patrón Proxy:

En este patrón un objeto en un contexto es representado por otro (el proxy) en un contexto diferente. El proxy sabe cómo redirigir las invocaciones a métodos entre los distintos objetos participantes.

De manera general su lógica establece que un “apoderado remoto” proporcione un “representante local” para un objeto en un espacio de direcciones diferente, estos representantes también son conocidos como “embajadores”. Mediante este mecanismo se introduce un nivel de “indirección” en el acceso al objeto, que permite al apoderado remoto ocultar el hecho de que el objeto reside en un espacio de direcciones distinto, siendo esta precisamente la fórmula utilizada por J2ME para su implementación.

2.5 Análisis comparativo de otras soluciones existentes con la propuesta

En la fase inicial del desarrollo del software se analizaron algunos servicios de lector de canal RSS. Luego de realizar un estudio profundo sobre estos, se eligieron las características más afines, con el objetivo de incorporarlo a la solución del presente trabajo. Del servicio de Viigo, se selecciono la características el cliente pueda ver primero una vista previa de la información antes de leerla completa. Del servicio de TouchTicker se selecciono la características de imágenes, ya que el cliente puede ver la imagen de la información que está consultando en el momento específico, he interactúa con la interfaz más amigable del tema específico. Del servicio de Viigo, se selecciono las características de hacer varias interfaces amigables para el cliente ya que el mismo puede moverse con su móvil e interactuar con la interfaz de forma segura y más organizada

Se definieron nuevas características que no fueron encontradas en las soluciones estudiadas y que eran de interés para la nueva propuesta. Estas características se exponen a continuación:

- ✓ La información del Lector será publicada en una interfaz *J2ME*, brindando al usuario la posibilidad de consultar el medio de prensa en el momento que lo desee.
- ✓ Los usuarios pueden seleccionar el medio de prensa y el tipo de noticia de ese medio de prensa del cual desee leer la información.
- ✓ Se enviará la dirección donde está publicada la información a través de una notificación *Push-Message*.

Se considera que las características tomadas de otros sistemas así como las añadidas en la nueva propuesta, son las que más se ajustan al entorno en que nuestro sistema se va a desenvolver. No es aplicable a nuestras condiciones teniendo en cuenta que uno de los objetivos de la nueva propuesta era que el usuario consultara su información a través de una interfaz J2ME en el momento que lo desee.

2.5 Conclusiones

En este capítulo fueron explicados los diferentes objetivos estratégicos de Cubacel, se expuso la propuesta de sistema de lectura, almacenamiento y visualización de canal RSS para dispositivos móviles, se definió la arquitectura a seguir y el flujo de procesos del mismo.

CAPÍTULO 3: EXPLORACIÓN, PLANIFICACIÓN Y DISEÑO.

En el presente capítulo se describen las fases de exploración, planificación y diseño pertenecientes a la metodología de desarrollo XP (*Extreme Programming*) utilizada para la implementación del sistema que se propone, mostrando los artefactos generados durante el transcurso de las mismas.

3.1 Fase de Exploración

Durante esta fase se realiza el proceso de identificación de las funcionalidades del sistema que el cliente desea, también conocidas como historias de usuario. En la misma los clientes plantean a grandes rasgos las características del sistema de interés para la primera entrega del producto.

3.1.1 Historias de Usuario

Las historias de usuario son la forma en que se especifican en *Extreme Programming* los requisitos del sistema. Estas se escriben desde la perspectiva del cliente aunque los desarrolladores pueden brindar ayuda en su identificación. El contenido de las mismas debe ser concreto y sencillo. El tiempo de desarrollo ideal para una historia de usuario es entre una y tres semanas (Beck, 1999).

Las historias de usuario también conducen el proceso de los test de aceptación, que son empleados para verificar que las historias han sido implementadas correctamente. Existen diferencias entre las historias de usuario y la tradicional especificación de requisitos, siendo su principal diferencia el nivel de detalle. Las historias de usuario solamente proporcionan los detalles sobre la estimación de riesgo y cuánto tiempo conllevará la implementación de dicha historia (Fowler, 2004).

Escala Nominal de Prioridad en el Negocio:

Alta: Se le otorga a las Historias de Usuario que resultan funcionalidades fundamentales en el desarrollo del software.

Media: Se le otorga a las Historias de Usuario que resultan para el cliente como funcionalidades a tener en cuenta.

Baja: Se le otorga a las Historias de Usuario que constituyen funcionalidades que sirven de ayuda al control de elementos asociados al equipo de desarrollo, a la estructura y no tienen nada que ver con el software en desarrollo.

Escala Nominal de Riesgo en Desarrollo:

Alta: Cuando para la implementación de la UH se considera la posible existencia de errores que lleven a inoperatividad del código.

Media: Cuando pueden aparecer errores en la implementación de la UH que puedan retrasar la entrega de la versión.

Baja: Cuando pueden aparecer errores que serán tratados con relativa facilidad sin que traigan perjuicios para el desarrollo del proyecto.

A continuación se muestran las UH que serán utilizadas en el desarrollo de la aplicación propuesta.

Tabla 1. Historia de Usuario **Acceder al Servicio de Lectura de Canales RSS.**

Historias de Usuario	
Número: 1	Nombre: Acceder al Servicio de Lectura
Usuario: Cliente	
Prioridad en el Negocio: Bajo	Riesgo en Desarrollo: Bajo
Puntos Estimados: 1	Iteración Asignada: 3
Descripción:	
Observaciones: El Cliente accede desde su dispositivo móvil al servicio que se está ofertando.	

Tabla 2. Historia de Usuario **Aceptar Servicio.**

Historias de Usuario	
Número: 2	Nombre: Aceptar Servicio
Usuario: Cliente	
Prioridad en el Negocio: Alto	Riesgo en Desarrollo: Bajo
Puntos Estimados: 2	Iteración Asignada: 1
Descripción: El cliente después de acceder al servicio de lecturas de canales RSS, se le muestra una interfaz con los medios de prensa del país que tienen implementado este tipo de servicio.	
Observaciones:	

Tabla 3. Historia de Usuario **Acceder Tipo de Noticia del Medio de Prensa.**

Historias de Usuario	
Número: 3	Nombre: Acceder Tipo de Noticia del Medio de Prensa.
Usuario: Cliente	
Prioridad en el Negocio: Alto	Riesgo en Desarrollo: Medio
Puntos Estimados: 2	Iteración Asignada: 2
Descripción: El cliente cuando escoge el medio de prensa se le muestra otra interfaz con los tipos de noticias (Culturales, Deportivas, Nacionales, Internacionales, Ciencia y Técnica entre otros) seleccionando uno de los mismos.	
Observaciones:	

Tabla 4 Historia de Usuario **Recibir Resumen de Noticias.**

Historias de Usuario	
Número: 4	Nombre: Recibir Resumen de Noticias
Usuario: Cliente	
Prioridad en el Negocio: Medio	Riesgo en Desarrollo: Alto
Puntos Estimados: 2	Iteración Asignada: 2
Descripción: El cliente accede a un resumen del contenido relacionado con el tipo de noticia escogido.	
Observaciones:	

Tabla 5 Historia de usuario **Ver Detalles de Noticia.**

Historias de Usuario	
Número: 5	Nombre: Ver Detalles de Noticia.
Usuario: Cliente	
Prioridad en el Negocio: Medio	Riesgo en Desarrollo: Alto
Puntos Estimados: 2	Iteración Asignada: 2
Descripción: El Cliente si desea leer la noticia completa del tipo de medio de prensa que escogió tiene la opción Ver Detalles, a través de su dispositivo móvil.	
Observaciones:	

3.2 Fase de Planificación

Durante la fase de planificación se establece por parte del cliente la prioridad de las diferentes historias de usuario, y en correspondencia los programadores realizan una estimación del esfuerzo necesario de cada una de ellas. Se toman acuerdos sobre el contenido de la primera entrega. Las estimaciones de esfuerzo asociado a la

implementación de las historias son establecidas por los programadores utilizando como medida la cantidad de puntos. Un punto equivale a una semana ideal de programación.

3.2.1 Estimación de esfuerzo por historias de usuario

Para el desarrollo del sistema propuesto se realizó una estimación del esfuerzo para cada una de las historias de usuario identificadas, los resultados de este estudio se muestra en la siguiente tabla.

Tabla 3.2.1 Estimación de Esfuerzo por UH.

No. UH	Historia de Usuario (UH)	Puntos Estimados
1	Acceder al Servicio de Lectura de Canales RSS	1
2	Aceptar Servicio.	2
3	Acceder Tipo de Noticia del Medio de Prensa.	2
4	Recibir Resumen de Noticias	3
5	Ver Detalles de Noticia	3

3.2.2 Plan de Iteraciones

Una vez identificadas las Historias de Usuario del sistema y estimado el esfuerzo dedicado a la realización de cada una de estas, se procede a la planificación de la etapa de implementación del Software. Para dar cumplimiento a dicha tarea se decidió realizar la misma en tres iteraciones, las cuales son explicadas a continuación.

3.2.2.1 Iteración 1

En esta iteración se implementarán las historias de usuario de nivel de prioridad Alta, creando en ella la línea base de la arquitectura del sistema. Además se obtendrá la versión 0.1 en la que el cliente podrá probar algunas de las funcionalidades solicitadas descritas en las UH 2,3.

3.2.2.2 Iteración 2

En la segunda iteración se llevará a cabo la implementación de la historia de usuario dos, de prioridad media, en la que se dispondrá como resultado final una funcionalidad que permita al cliente acceder al contenido del tipo de noticia según el medio de prensa seleccionado desde su dispositivo móvil. Al concluir esta iteración se le dará al cliente la versión de prueba 0.2 de la propuesta.

3.2.2.3 Iteración 3

Esta iteración es la dedicada a la implementación de las historias de usuario de nivel de prioridad Baja. Se desarrollará en el transcurso de la misma la versión 1.0 de RSSMobile.cu, adicionando las funcionalidades descrita en la UH 1.

3.2.3 Plan de duración de las Iteraciones

Con la utilización de la metodología de desarrollo XP se hace necesario la creación de un plan de duración de iteraciones como parte del ciclo de vida del proyecto. En dicho plan de duración se contará con cada una de las iteraciones a realizar, las historias de usuarios que se implementarán en la iteración y la duración en semanas.

Tabla 3.2.3 Plan de Duración de las Iteraciones.

Iteración	Historias de Usuario a Implementar	Duración en Semanas
Iteración 1	1. Aceptar Servicio. 2. Acceder Tipo de Noticia del Medio de Prensa.	4 Semanas.
Iteración 2	1. Recibir Resumen de Noticias 2. Ver Detalles de Noticia	6 Semanas.
Iteración 3	1. Acceder al Servicio de Lectura de Canales RSS	1 Semana

3.2.4 Plan de Entregas

A continuación se muestra una tabla con el plan de entrega propuesto para cada una de las versiones de RSSMobile.cu.

Tabla 3.2.4 Plan de Entregas.

Sistema	Fin Iteración 1 3ra Semana Abril	Fin Iteración 2 1ra Semana Mayo	Fin Iteración 3 4ta Semana Mayo
RSSMobile.cu	0.1	0.2	1.0

3.3 Diseño

Durante el diseño de la solución, y de cada historia de usuario que lo requiera, la máxima simplicidad posible es la clave para el éxito de XP. Se debe tener en cuenta que un diseño complejo siempre tarda más en desarrollarse que uno simple, y que siempre es más fácil añadir complejidad a un diseño simple que quitarla de uno complejo, siempre se deben hacer las cosas lo más sencillas posible, evitando añadir una funcionalidad no contemplada a una iteración.

3.3.1 Tarjeta CRC

Para el diseño del sistema XP (*Xtreme Programming*) sugiere llevar a cabo reuniones entre todos los desarrolladores implicados, donde se toman las decisiones mediante el uso de tarjetas CRC (*class, responsibilities and collaboration*, en inglés). Cada tarjeta representa un objeto del sistema, en la que su nombre aparece escrito en la parte superior, sus responsabilidades en la parte izquierda y los objetos con los que colabora en la parte derecha.

A continuación se muestra el diseño correspondiente a las clases utilizadas en la implementación del sistema.

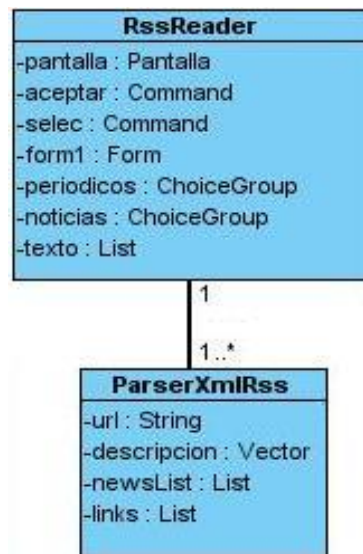


Figura 3.3.1 Diagrama de Clases Persistentes.

3.4 Conclusiones

En el presente capítulo fueron abordados los contenidos de las fases de exploración, planificación y diseño de la metodología XP utilizada para el desarrollo del software, siendo generados los artefactos de las mismas, se explicaron las historias de usuario y se definió el trabajo a realizar en cada una de las iteraciones del proyecto, así como el plan de entrega de cada uno de los módulos.

CAPÍTULO 4: IMPLEMENTACION Y PRUEBA.

En el presente capítulo se detallan las iteraciones que serán llevadas a cabo durante la implementación del sistema, se describen las tareas generadas por cada una de las Historias de Usuario y las pruebas de aceptación efectuadas.

A lo largo del desarrollo de las iteraciones se realizará la implementación de las historias de usuario que componen cada una de estas. Se llevará a cabo una revisión del plan de iteraciones y se modificará este de ser necesario. De acorde con el plan de desarrollo se descomponen las UH en tareas asignándolas posteriormente a un equipo (o una persona) responsable de su implementación. Estas tareas serán usadas por los programadores, pueden ser descritas utilizando un lenguaje técnico y no necesariamente deben ser entendibles para el cliente.

Se llevarán a cabo tres iteraciones de desarrollo sobre el sistema, obteniéndose al finalizar un producto listo para su aplicación. A continuación se muestran cada una de las iteraciones.

4.1 Iteración 1.

En esta iteración se abordarán las UH de nivel de prioridad ALTA y se construirá la base de la arquitectura del sistema. Se obtendrá la primera versión con la que el cliente podrá probar que las funcionalidades que este considera principales en la aplicación.

Tabla 4.1 UH pertenecientes a la primera iteración.

Historias de Usuario	Estimación	Real
Aceptar Servicio.	2	2
Acceder Tipo de Noticia del Medio de Prensa.	2	2
Total	4	4

Mostrar Servicio

Tabla 4.1.1 Tarea #1 de la UH #2

Tarea	
Número de Tarea: 1	Número de Historia: 2
Nombre de Tarea: Creación de Interfaz de Medios de Prensa.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 2
Fecha Inicio:	Fecha de Fin:
Responsable: Yaima Ricardo Portelles – Javier Rodríguez Rivas	
Descripción: Se crea una interfaz que muestra los principales medios de prensa del país que tienen implementado un servicio de lectura de canales RSS.	

Tabla 4.1.2. Tarea #2 de la UH #2

Tarea	
Número de Tarea: 2	Número de Historia: 2
Nombre de Tarea: Selección de Medio de Prensa.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 2
Fecha Inicio:	Fecha de Fin:
Responsable: Yaima Ricardo Portelles – Javier Rodríguez Rivas	
Descripción: Luego de mostrar los medios de prensa se crea una conexión a la página web seleccionada por el usuario.	

Tabla 4.1.4. Tarea #1 de la UH #3

Tarea	
Número de Tarea: 1	Número de Historia: 3
Nombre de Tarea: Creación de Interfaz de Tipos de Noticias.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 2
Fecha Inicio:	Fecha de Fin:
Responsable: Yaima Ricardo Portelles – Javier Rodríguez Rivas	
Descripción: Se crea una interfaz que muestra los tipos de noticias a leer por parte del usuario del medio de prensa seleccionado.	

Tabla 4.1.5. Tarea #2 de la UH #3

Tarea	
Número de Tarea: 2	Número de Historia: 3
Nombre de Tarea: Selección del Tipo de Noticia.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 2
Fecha Inicio:	Fecha de Fin:
Responsable: Yaima Ricardo Portelles – Javier Rodríguez Rivas	
Descripción: Luego de mostrar los tipos de noticias se crea un vínculo de acceso a la información que presenta la página web seleccionada del tipo de noticia especificado.	

4.2 Iteración 2.

Tabla 4.2 UH pertenecientes a la segunda iteración.

Historias de Usuario	Estimación	Real
Recibir Resumen de Noticias	3	3
Ver Detalles de Noticias.	3	3

Total	6	6
--------------	----------	----------

Recibir Resumen de Noticias

Tabla 4.2.1. Tarea #1 de la UH #4

Tarea	
Número de Tarea: 1	Número de Historia: 4
Nombre de Tarea: Implementación de la Clase ParserXML.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 2
Fecha Inicio:	Fecha de Fin:
Responsable: Yaima Ricardo Portelles – Javier Rodríguez Rivas	
Descripción: Se implementa la clase ParserXML para parsear los archivos XML contenidos en las páginas web de los medios de prensa.	

Tabla 4.2.2. Tarea #2 de la UH #4

Tarea	
Número de Tarea: 2	Número de Historia: 4
Nombre de Tarea: Crear Resumen de Noticias.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 2
Fecha Inicio:	Fecha de Fin:
Responsable: Yaima Ricardo Portelles – Javier Rodríguez Rivas	
Descripción: Se crea una interfaz para mostrar al usuario un resumen de las noticias del tipo escogido por dicho usuario que se encuentran en el medio de prensa.	

Ver Detalles de Noticia.

Tabla 4.2.3. Tarea #1 de la UH #5

Tarea	
Número de Tarea: 2	Número de Historia: 4
Nombre de Tarea: Crear Interfaz con los detalles de la noticia.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 2
Fecha Inicio:	Fecha de Fin:
Responsable: Yaima Ricardo Portelles – Javier Rodríguez Rivas	
Descripción: Se crea una interfaz para mostrar al usuario la noticia completa.	

4.3 Iteración 3.

Tabla 4.3 UH pertenecientes a la tercera iteración.

Historias de Usuario	Estimación	Real
Acceder al Servicio de Lectura de Canales RSS	1	1
Total	1	1

Acceder al Servicio de Lectura de Canales RSS

Tabla 4.3.1. Tarea #1 de la UH #1

Tarea	
Número de Tarea: 1	Número de Historia: 1
Nombre de Tarea: Implementación de la clases RSSMobile .cu.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 2
Fecha Inicio:	Fecha de Fin:
Responsable: Yaima Ricardo Portelles – Javier Rodríguez Rivas	

Descripción: Se implementará la clase RSSMobile.cu con funcionalidades que permitirán la creación de nuevas interfaces, insertar nuevos medios de prensa y tipo de noticia a leer por parte de los usuarios.

Tabla 4.3.2. Tarea #2 de la UH #1

Tarea	
Número de Tarea: 2	Número de Historia: 1
Nombre de Tarea: Crear vínculo de aceptación del servicio.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 2
Fecha Inicio:	Fecha de Fin:
Responsable: Yaima Ricardo Portelles – Javier Rodríguez Rivas	
Descripción: Se creará una funcionalidad (vínculo) para el acceso al servicio de lecturas del canales RSS desde un dispositivo móvil.	

4.4 Módulos y Librerías Utilizadas en la Implementación.

4.4.1 Módulo KXML v1.0

KXML es un pequeño analizador XML de tipo pull, especialmente diseñado para entornos con limitaciones tales como applets, Java MIDP personales o dispositivos.

El tipo pull de análisis basado en XML combina algunas de las ventajas de SAX y DOM:

A diferencia de impulsar programas de análisis (SAX), este tipo de programas de análisis como kXML hacen posible modelar las rutinas de procesamiento de XML después de la estructura del documento XML procesado. Procesamiento de eventos es similar a un InputStream. Si una parte del flujo requiere de atención especial, el analizador sólo puede ser delegado a un método especializado mediante la entrega del analizador.

A pesar de lo anterior también es posible con una explícita DOM, DOM lo general requiere que la estructura del documento se halle presente en la memoria principal.

A diferencia de los análisis basados en DOM, XML son los acontecimientos inmediatamente accesible, cuando estén disponibles, no es necesario esperar a que todo el árbol para construir.

4.4.2 Módulo Sony Ericsson J2ME SDK 1.2

Java es una parte integral del pensamiento de Sony Ericsson. Sony Ericsson introdujo una máquina virtual multi-tarea en 2006 y el primer teléfono totalmente compatible con MSA en 2008. El Sony Ericsson Plataforma Java JSR tiene soporte completo, incluyendo Java 3D y proporciona acceso a funciones de hardware como el acelerómetro incorporado. Sony Ericsson es un miembro del Comité Ejecutivo de Java ME y apoya el programa y Java Verified JATAF (*Java Application Terminal Alineación Marco*), que aborda la fragmentación de las implementaciones de Java ME.

Las herramientas de desarrollo incluyen el Sony Ericsson SDK para Java ME con emulación completa en 3D y la depuración en el dispositivo, así como las directrices de los desarrolladores y plug-ins.

El Sony Ericsson SDK para Java ME™ Platform es una manera fácil y eficiente para crear MIDP 1.0, MIDP 2.0 y Java 3D. Depurar el código fuente en tiempo real con los muy aclamados en el dispositivo las funciones de depuración y gestión de MIDlets en el auricular mediante el Device Explorer. Apoya el desarrollo de aplicaciones para el Sony Ericsson Java Platform, que es la implementación de Java en la plataforma propiedad de Sony Ericsson.

4.4.3 J2ME Wireless Toolkit 1.0.4

El Sun J2ME Wireless Toolkit 1.0.4 apoya el desarrollo de aplicaciones Java que se ejecutan en dispositivos compatibles con la CLDC 1.0 y MIDP 1.0 especificaciones tales como teléfonos celulares, paginadores de dos vías, y palmtops.

Este kit de herramientas no incluye un entorno de desarrollo integrado.

Esta herramienta ofrece varias y nuevas características convincentes para ayudar a crear MIDlets que son compactos, eficientes y fáciles de usar:

- * Apoyo a la ofuscación en el ciclo de la construcción
- * Método de perfiles
- * El seguimiento de la memoria
- * Red de vigilancia
- * Velocidad del dispositivo de emulación

El J2ME Wireless Toolkit apoya la creación de aplicaciones MIDP con las siguientes características principales:

- Construcción y embalaje: Usted escribe el código fuente y herramientas de la que cuida del resto. Con sólo pulsar un botón, la caja de herramientas compila el código fuente, preverifies los archivos de clase, y los paquetes de un MIDlet suite.
- Ejecución y seguimiento: Se puede ejecutar un MIDlet suite directamente en el emulador o instalarlo mediante un proceso que se asemeja a la instalación de aplicaciones en un dispositivo real. Un monitor de memoria, monitor de red, y un perfilador de método se proporcionan para analizar el funcionamiento de su MIDlets.
- Firma MIDlet suite: El conjunto de herramientas contiene herramientas para firmar criptográficamente paquetes de MIDlets. Esto es útil para comprobar el funcionamiento de MIDlets en ámbitos de protección diferentes.

4.5 Fase de Prueba

Uno de los pilares de XP es el proceso de pruebas (Beck, 1999). XP anima a probar tanto como sea posible. Esto permite aumentar la calidad de los sistemas reduciendo el número de errores no detectados y disminuyendo el tiempo transcurrido entre la aparición de un error y su detección. También permite aumentar la seguridad de evitar efectos colaterales no deseados a la hora de realizar modificaciones.

XP divide las pruebas del sistema en dos grupos: pruebas unitarias, encargadas de verificar el código y diseñada por los programadores, y pruebas de aceptación o pruebas funcionales destinadas a evaluar si al final de una iteración se consiguieron las funcionalidades requeridas diseñadas por el cliente final (Crispin, y otros, 2002).

4.5.1 Pruebas de Aceptación

Las pruebas de aceptación son pruebas de caja negra que se crean a partir de las historias de usuario (Crispin, y otros, 2002). Durante las iteraciones las historias de usuario seleccionadas serán traducidas a pruebas de aceptación. En ellas se especifican, desde la perspectiva del cliente, los escenarios para probar que una historia de usuario ha sido implementada correctamente. Una historia de usuario puede tener todas las pruebas de aceptación que necesite para asegurar su correcto funcionamiento. Las pruebas de aceptación significan la satisfacción del cliente con el producto desarrollado, el final de una iteración y el comienzo de la siguiente (Miller, 2001). El objetivo final de estas es garantizar que los requerimientos han sido cumplidos y que el sistema es aceptable (Wells, 2006).

Las pruebas de aceptación se muestran en el [anexo 1](#).

4.6 Conclusiones

Las fases de Implementación y Prueba sin lugar a dudas son las que cierran el desarrollo de una aplicación siguiendo la metodología XP, los artefactos generados son de vital importancia para comprobar si ha sido satisfactorio el trabajo realizado a lo largo de la implementación.

Capítulo 5. Estudio de Factibilidad

La rentabilidad, estimación de esfuerzo y costo de un proyecto de software son sin lugar a dudas una de las tareas de mayor importancia para la producción de software. Debido a que cada vez es mayor la necesidad de obtener datos objetivos que permitan evaluar, predecir y mejorar la calidad del software así como el tiempo y coste de desarrollo del mismo.

5.1 Modelo matemático COCOMO II

COCOMO, propuesto y desarrollado por Barry Boehm, es uno de los modelos de estimación de costos mejor documentados y utilizados. El modelo permite determinar el esfuerzo y tiempo que se requiere en un proyecto de software a partir de una medida del tamaño del mismo expresada en el número de líneas de código que se estimen generar para la creación del producto software.

Consiste básicamente en la aplicación de ecuaciones matemáticas sobre los Puntos de Función sin ajustar o la cantidad de líneas de código (SLOC, *Source Lines of Code*) estimados para un proyecto. Estas ecuaciones se encuentran ponderadas por ciertos factores de costo (*cost drivers*) que influyen en el esfuerzo requerido para el desarrollo del software.

Está compuesto por tres modelos que se adaptan tanto a las necesidades de los diferentes sectores descritos, como al tipo y cantidad de información disponible en cada etapa del ciclo de vida de desarrollo, lo que se conoce por granularidad de la información.

El estudio de factibilidad de este sistema se realizó mediante el modelo de diseño temprano que se utiliza en las primeras etapas del desarrollo en las cuales se evalúan las alternativas de hardware y software de un proyecto. En estas etapas se tiene poca información, se conoce muy poco del tamaño del producto a ser desarrollado, de la naturaleza de la plataforma, del personal a ser incorporado al proyecto o aspectos específicos del proceso a utilizar. Este nivel de detalle en este modelo es consistente con el nivel general de información disponible y con el nivel general de estimación detallada que es necesaria en esta etapa, lo que concuerda con el uso de Puntos de Función, para estimar tamaño usa Puntos de Función No Ajustados como métrica de medida y el uso de un número reducido de factores de costo.

5.2 Características del Proyecto

El primer paso a llevar a cabo para la estimación del proyecto consiste en la obtención de los Puntos de Función desajustados, los cuales están dados por la suma de cada una de

las entradas, las salidas y las consultas externas del sistema, así como los archivos lógicos internos y de interfaz externos. A continuación se muestran cada una de estas características aplicadas al software en cuestión.

5.2.1 Entradas externas

Se definen como un proceso elemental mediante el cual ciertos datos cruzan la frontera del sistema desde afuera hacia adentro. En el caso particular de la aplicación propuesta se cuenta con siete entradas externas, especificadas en la siguiente tabla:

Tabla 5.2.1 Entradas Externas.

Nombre de la Entrada Externa	Cantidad de ficheros	Cantidad de elementos	Clasificación (simple, media y compleja)
Acceder al Servicio de Lectura de Canales RSS.	1	2	Simple
Aceptar Servicio.	1	3	Simple
Total	2	5	

5.2.2 Salidas Externas

Se definen como un proceso elemental con componentes de entrada y de salida mediante el cual datos simples y datos derivados (esto es, datos que se calculan a partir de otros datos) cruzan la frontera del sistema desde adentro hacia afuera). Las salidas externas vinculadas al proyecto se describen a continuación.

Tabla 5.2.2 Salidas Externas

Nombre de la Salida Externa	Cantidad de ficheros	Cantidad de elementos	Clasificación (simple, media y compleja)
Acceder Tipo de Noticia del Medio de Prensa.	1	5	Simple

Recibir Resumen de Noticias	1	5	Simple
Ver Detalles de Noticia	1	3	Simple
Total	3	13	

5.2.3 Consultas Externas

Se definen como un proceso elemental con componentes de entrada y de salida donde un Actor del sistema rescata datos de uno o más Archivos Lógicos Internos o Archivos de Interfaz Externos. Los datos de entrada no actualizan ni mantienen ningún archivo (lógico interno o de interfaz externo) y los datos de salida no contienen datos derivados (es decir, los datos de salida son básicamente los mismos que se obtienen de los archivos).

Tabla 5.2.3 Consultas Externas

Nombre de la Petición	Cantidad de Ficheros	Cantidad de elementos	Clasificación (simple, media y compleja)
Total		0	

5.2.4 Archivos Lógicos Internos

Constituyen un grupo de datos relacionados lógicamente e identificables por el usuario, que residen enteramente dentro de los límites del sistema y se mantienen a través de entradas externas. En este caso no se cuentan con archivos lógicos internos, como se muestra en la siguiente tabla.

Tabla 5.2.4 Archivos Lógicos Internos

Nombre del fichero interno	Cantidad de records	Cantidad de elementos de datos	Clasificación (simple, media y compleja)
Total		0	

5.2.4 Archivos de Interfaz Externos

Son un grupo de datos relacionados lógicamente e identificables por el usuario, que se utilizan solamente para fines de referencia. Los datos residen enteramente fuera de los límites del sistema y se mantienen por las Entradas Externas de otras aplicaciones, es decir, cada Archivo de Interfaz Externo es un Archivo Lógico Interno de otra aplicación. En este caso fueron identificados dos Archivos de interfaz externa, como se detalla en la siguiente tabla:

Tabla 5.2.4 Archivos de Interfaz Externos

Nombre de la entrada externa	Cantidad de records	Cantidad de elementos de datos	Clasificación (simple, media y compleja)
Total		0	

5.2.5 Puntos de Función Desajustados

La siguiente tabla está basada en las características del sistema anteriormente expuestas, el producto de la cantidad existente de cada una de ellas y el peso correspondiente a las mismas dan como resultado final, los puntos de función desajustados pertenecientes al proyecto.

Tabla 5.2.5 Puntos de Función Desajustados

Elementos	Simple		Medio		Complejo		Subtotal
	No.	Peso	No.	Peso	No.	Peso	
Entrada Externa	5	3	0	4	0	6	17
Salida Externa	13	4	0	5	0	7	56
Consultas Externas	0	3	0	4	0	6	17
Fichero Lógico Interno	6	7	0	10	0	15	42
Fichero interfaz Externa	0	5	0	7	0	10	0

Total (UFP)							132
--------------------	--	--	--	--	--	--	------------

5.3 Cálculo de instrucciones fuentes, esfuerzo, tiempo de desarrollo, cantidad de hombres y costo.

5.3.1 Cálculo de instrucciones fuentes.

El modelo COCOMO II usa Puntos Función (FP) y/o Líneas de Código Fuente (SLOC) como base para medir tamaño en los modelos de estimación de Diseño Temprano y Post-Arquitectura. Los Puntos de Función procuran cuantificar la funcionalidad de un sistema de software. La meta es obtener un número que caracterice completamente al sistema. Son útiles estimadores ya que están basados en información que está disponible en las etapas tempranas del ciclo de vida del desarrollo de software.

Las características y valores obtenidos anteriormente han sido plasmados en la siguiente tabla:

Tabla 5.3.1 Características del Sistema

Características	Valor
Puntos de función desajustados	103
Lenguaje (Java)	53
Instrucciones fuentes por puntos de función	5459
Instrucciones fuentes	5.45KSLOC

5.4 Cálculo de Esfuerzo Nominal

Posteriormente se procede al cálculo del esfuerzo nominal, ecuación que se toma como base tanto en el método de diseño preliminar, al cual se hace alusión anteriormente, como en el modelo Post arquitectura, ambos definidos por COCOMO II.

Tabla 5.3.2 Factores de Escala

Nombre	Valor	Justificación
Precedentes	1.24	Existen varios proyectos similares a nivel internacional, pero no se encontró ninguno a nivel nacional.
Flexibilidad	1.01	Cuenta con alta flexibilidad en cuanto a los

de desarrollo		requerimientos establecidos inicialmente.
Cohesión del equipo	1.10	El equipo de desarrollo presenta una alta cohesión.
Solución de riesgos	2.83	Se identificaron riesgos críticos en un porcentaje elevado.
Madurez del proceso	1.56	Se cuenta con la experiencia necesaria como para que el software cumpla con las funcionalidades requeridas
Total (SF)	7.74	

5.4.1 Ajuste del Esfuerzo Nominal

El esfuerzo calculado anteriormente es un valor nominal y debe ser ajustado en base a las características del proyecto para lo cual se tiene un conjunto de Multiplicadores de esfuerzo (ME_i) que representan las características del proyecto y expresan su impacto en el desarrollo total del producto de software.

5.5 Cálculo de Esfuerzo Nominal

Posteriormente se procede al cálculo del esfuerzo nominal, ecuación que se toma como base tanto en el método de diseño preliminar, al cual se hace alusión anteriormente, como en el modelo Post arquitectura, ambos definidos por COCOMO II.

$$PM_{nominal} = A \times (Size)^B$$

$$PM_{nominal} = 2.94 \times 5.45^{0.98}$$

$$PM_{nominal} = 15.5 \text{ meses/hombre}$$

Donde

$PM_{nominal}$: Esfuerzo nominal requerido en meses – hombre

Size: Tamaño estimado del software, en Puntos de Función sin ajustar (KSLOC)

A: constante que se utiliza para capturar los efectos multiplicativos en el esfuerzo requerido de acuerdo al crecimiento del tamaño del software. (A= 2.94).

B: Constante denominada Factor escalar y su valor esta dado por la resultante de los aspectos positivos sobre los negativos que presenta el proyecto.

$$B = 0.91 + 0.01 \times \sum(W_i)$$

$$B = 0.91 + 0.01 \times 7.74$$

$$B = 0.98$$

Donde

W_i : Variables escalares que indican las características que el proyecto presenta en lo que a su complejidad y entorno de desarrollo se refiere. La siguiente tabla muestra los valores asignados a cada una de estas variables:

Tabla 5.3.2 Factores de Escala

Nombre	Valor	Justificación
PREC	1.24	Existen varios proyectos similares a nivel internacional, pero no se encontró ninguno a nivel nacional.
FLEX	1.01	Cuenta con alta flexibilidad en cuanto a los requerimientos establecidos inicialmente.
TEAM	1.10	El equipo de desarrollo presenta una alta cohesión.
RESL	2.83	Se identificaron riesgos críticos en un porcentaje elevado.
PMAT	1.56	Se cuenta con la experiencia necesaria como para que el software cumpla con las funcionalidades requeridas
Total (SF)	7.74	

5.5.1 Ajuste del Esfuerzo Nominal

El esfuerzo calculado anteriormente es un valor nominal y debe ser ajustado en base a las características del proyecto para lo cual se tiene un conjunto de Multiplicadores de esfuerzo (ME_i) que representan las características del proyecto y expresan su impacto en el desarrollo total del producto de software (2).

Tabla 5.3.3 Multiplicadores de Esfuerzo

Nombre	Valor	Justificación
RCPX	1.74	La complejidad del sistema es alta

RUSE	1.00	El nivel de reutilización del producto es alto.
PDIF	1.00	Uso de memoria y almacenamiento normal, plataforma estable
PREX	1.12	Baja experiencia del personal en la utilización de lenguaje, así como de la plataforma
PERS	1.00	La capacidad del personal es alta
FCIL	1.00	Se utilizaron entornos de desarrollo integrados, herramientas de modelación y automatización que facilitan el trabajo.
SCED	1.00	El sistema se desarrolló en el tiempo establecido
Total (EM)	1.95	

$$PM_{ajustado} = PM_{nominal} \times \prod (ME_i)$$

$$PM_{ajustado} = 15.5 \text{ meses/hombre} \times 1.95$$

$$PM_{ajustado} = 30.23 \text{ hombres/mes}$$

5.5.1.1 Calculo del tiempo de desarrollo del software

El tiempo requerido para el desarrollo del proyecto está dado por la siguiente ecuación:

$$TDEV = C \times (PM_{ajustado})^F$$

$$TDEV = 3.64 \times 30.23^{0.25}$$

$$TDEV = 8.5$$

Donde

C: constante con valor 3.64

$$F = D + 0.2 \times 0.01 \times \sum SF$$

$$F = 0.24 + 0.2 \times 0.01 \times 7.74$$

$$F = 0.25$$

Donde

D: constante cuyo valor es 0.24

SF: valor de los factores de escala

5.5.2 Cálculo del costo total del proyecto

Para el cálculo del costo total correspondiente al proyecto en cuestión, COCOMO II define la siguiente ecuación:

$$C = CHM \times PM$$

$$C = 200 \times 15.5$$

$$C = 3100$$

Donde

C: costo total

CHM: costo teniendo en cuenta salario de todos los obreros, el cual es calculado por la ecuación

$$CHM = CH \times sal$$

$$CHM = 2 \times 100$$

$$CHM = 200$$

Donde

Sal: Salario medio por cada trabajador

CH: Cantidad de personas destinadas al proyecto, lo que es calculado a través de la ecuación:

$$CH = PM / TDEV$$

$$CH = 15.5 / 8.5$$

$$CH \approx 2$$

5.6 Resultados

La siguiente tabla muestra los resultados obtenidos luego de haber realizado todo el análisis correspondiente y haber efectuado todos los cálculos para determinar el costo y esfuerzo requeridos por el proyecto.

Cálculo de:	Valor
Esfuerzo	30.23 hombres/mes
Tiempo de desarrollo	6 meses
Cantidad de hombres	2 hombres
Salario medio	100 pesos
Costo	3100 pesos

5.7 Beneficios tangibles e intangibles

El Sistema de Servicio de Lector de Canal RSS es un producto con fines comerciales. El beneficio fundamental está dado por la posibilidad de poder contar con un sistema que brinde a los clientes la prestación de recibir su lector según la frecuencia que estos decidan, y con ello aumentar los ingresos de Cubacel por este concepto. Como beneficio intangible aportado por el sistema desarrollado se puede mencionar que servirá de base para otros servicios de valor agregado que serán desarrollados por la entidad antes mencionada. Vale destacar que haciendo uso de la presente solución, pueden brindarse otros servicios similares, como recibir diariamente, por ejemplo, la primera noticia de un determinado periódico.

5.8 Análisis de costo

El desarrollo de un producto siempre presenta un costo de producción, el cual es justificado en base a los beneficios reportados por el mismo. Los productos informáticos no están exentos de ello. El sistema presentado en este trabajo no precisa grandes gastos, debido en gran medida a la utilización de plataformas, tecnologías y herramientas libres que no requieren del pago de licencia por su uso o comercialización, su costo está fundamentalmente determinado por el salario devengado por los desarrolladores que toman parte en su desarrollo.

5.9 Conclusiones

Todo producto conlleva en si un costo, este solo es justificado cuando al final de la etapa de producción cuando el fruto del producto supera con creces lo que haya costado producirlo. En el caso de la aplicación propuesta se puede concluir que es factible ya que solo aparece con gasto significativo el salario de los desarrolladores.

Conclusiones Generales

Con el desarrollo de este trabajo se profundizó en el conocimiento de la telefonía móvil y los servicios de valor agregado. Se utilizó la metodología XP para ordenar el proceso de desarrollo de software, se obtuvo la implementación de un servicio Lector de Canal RSS para Dispositivos Móvil con las funcionalidades previstas para el primer ciclo de desarrollo del software. El diseño y la implementación se rigieron por los estándares de la plataforma J2ME. Se definieron los parámetros de configuración e instalación con la documentación correspondiente. Se puede concluir que se ha cumplido satisfactoriamente el objetivo trazado para esta investigación, enfatizando en lo siguiente: fue implementada una aplicación que permite la lectura de canales RSS mediante dispositivos móviles.

Se desarrolló una interfaz J2ME que permite consultar el servicio de Lector de Canal RSS por parte de los usuarios del sistema.

Recomendaciones

A continuación se plantean las recomendaciones que en opinión del autor deben ser tenidas en cuenta con el objetivo de realizar un seguimiento de este trabajo:

- ✓ Integrar el sistema a la Plataforma de Servicios de Cubacel.
- ✓ Utilizar este sistema como base para la implementación de otros servicios de valor agregado en la plataforma de servicios de Cubacel.
- ✓ Realizar pruebas de este sistema en condiciones reales.

Referencias Bibliográficas

- Alexander, Christopher, Ishikawa, S. y Silverstein, Murray. 1997.** *A Pattern Language: Towns, Buildings, Construction*. Oxford, England: Oxford University Press, 1997.
- Beck, Kent. 1999.** *Extreme Programming Explained*. s.l.: Addison-Wesley, 1999. 0201616416.
- Beck, Kent y Cunningham, Ward. 1989.** *A Laboratory For Teaching Object-Oriented Thinking*. New Orleans: SIGPLAN Notices, 1989.
- Begin, Clinton, Goodin, Brandon and Meadors, Larry. 2007.** *JBatis in Action*. s.l.: Manning Publications, 2007.
- Crispin, Lisa y House, Tip. 2002.** *Testing Extreme Programming*. s.l.: Addison-Wesley, 2002. 0321113551.
- . **1999.** *Planning Extreme Programming*. s.l.: Addison-Wesley, 1999. 0201710919.
- IBM. 2007.** Eclipse Platform Technical Overview. *Eclipse*. [En línea] 2007. [Citado el: 12 de Abril de 2009.] <http://www.eclipse.org/articles/Whitepaper-Platform-3.1/eclipse-platform-whitepaper.pdf>.
- IEC. 2007.** International Engineering Consortium. *Global System for Mobile Communication (GSM)*. IEC. [En línea] 2007. [Citado el: 26 de Febrero de 2009.] <http://www.iec.org/online/tutorials/gsm/index.html...>
- . **2005.** Short Message. *Search Tech Target*. [En línea] 2 de Mayo de 2005. [Citado el: 15 de Febrero de 2009.] http://searchmobilecomputing.techtarget.com/sDefinition/0,290660,sid40_gci773769,00.html.
- Wikipedia. 2007.** Control de versiones. *Wikipedia enciclopedia libre*. [Online] 2007. [Cited: Abril 11, 2009.] http://es.wikipedia.org/wiki/Control_de_versiones.
- . **2008.** Entorno de desarrollo integrado. *Wikipedia enciclopedia libre*. [En línea] 2008. [Citado el: 19 de Marzo de 2009.] http://es.wikipedia.org/wiki/Ambiente_integrado_de_desarrollo.
- . **2009.** Framework. *Wikipedia enciclopedia libre*. [En línea] 2009. [Citado el: 6 de Marzo de 2009.] <http://es.wikipedia.org/wiki/Framework>.

—. **2009**. Interfaz de programación de aplicaciones. *Wikipedia enciclopedia libre*. [En línea] 2009. [Citado el: 14 de Marzo de 2009.]

http://es.wikipedia.org/wiki/Application_Programming_Interface.

—. **2009**. Plain Old Java Object. *Wikipedia enciclopedia libre*. [En línea] 2009. [Citado el: 9 de Marzo de 2009.] http://es.wikipedia.org/wiki/Plain_Old_Java_Object.

—. **2008**. Plataforma de Desarrollo. *Wikipedia enciclopedia libre*. [En línea] 2008. [Citado el: 3 de Marzo de 2009.] http://es.wikipedia.org/wiki/Plataforma_de_desarrollo.

El Lenguaje Java, 2010. Desarrollo de aplicaciones para dispositivos inalámbricos (J2ME) [En línea]. <http://www.mailxmail.com/curso-desarrollo-aplicaciones-dispositivos-inalambricos-j2me>.

Cursos J2ME, 2009 [en línea] <http://www.javamovil.info/J2ME/index.php>

Glosario de Términos

API: (*Application Programming Interface*) Interfaz de Programación de Aplicaciones.

COCOMO: (*COnstructive COst MOdel*) Modelo Constructivo de Costo: es un modelo matemático de base empírica utilizado para estimación de costes de software.

CRC: (*Class, Responsibility and Collaboration*) son una metodología para el diseño de software orientado por objetos creada por Kent Beck y Ward Cunningham.

DOM: (*Document Object Model*) Modelo de Objetos del Documento: es esencialmente una interfaz de programación de aplicaciones que proporciona un conjunto estándar de objetos para representar documentos.

Framework: estructura de soporte definida, en la cual otros proyectos de software pueden ser organizados y desarrollados.

GPRS: (*General Packet Radio Service*) Servicio General de Paquetes de Radio. Es un estándar de comunicación para teléfonos móviles que transmite la información por grupos significativos o paquetes.

GSM: (*Global System for Mobile Communications*) Sistema Global para comunicaciones Móviles. Sistema de telefonía celular digital para comunicaciones móviles de segunda generación.

HTML: (*Hyper Text Markup Language*) Lenguaje de Etiquetas de Hipertexto. Es un lenguaje de etiquetas diseñado para estructurar textos y presentarlos en forma de hipertexto, es el formato estándar de las páginas web.

HTTP: (*Hypertext Transfer Protocol*) Protocolo de transferencia de hipertexto.

IDE: (*Integrated Development Environment*) Entorno Integrado de Desarrollo.

J2EE: (Java 2 Platform Enterprise Edition) es una plataforma de programación para desarrollar y ejecutar software de aplicaciones en Lenguaje de programación Java con arquitectura de N niveles distribuida.

J2ME: (Java 2 Micro Edition) es una especificación de un subconjunto de la plataforma Java orientada a proveer una colección certificada de API's de desarrollo de software para dispositivos con recursos restringidos.

J2SE: (Java 2 Standard Edition) es una colección de API's del lenguaje de programación Java útiles para muchos programas de la Plataforma Java.

JAD: (*Java Decompiler*) es un decompilador para el lenguaje de programación Java. Proporciona una Interfaz de línea de comandos para extraer el código fuente de los archivos de clase.

JAR: (*Java ARchive*): es un tipo de archivo que permite ejecutar aplicaciones escritas en lenguaje Java.

JAVA: Lenguaje de programación multiplataforma desarrollado por Sun Microsystem.

Javascript: es un lenguaje de scripting basado en objetos no tipado y liviano, utilizado para acceder a objetos en aplicaciones.

JATAF: (*Java Application Terminal Alignment Framework*) es un nuevo proyecto de colaboración que involucra a empresas y desarrolladores que trabajan juntos para hacer de Java ME una verdadera plataforma ubicua para la implementación de aplicaciones en dispositivos móviles.

JDT: (*Java Development Tool*) Herramienta de Desarrollo para Java.

kXML: es un analizador sintáctico de XML ligero basado en Java diseñado para ser ejecutado en sistemas integrados limitados como dispositivos móviles personales.

MIDP: (*Mobile Information Device profile*) es una versión de J2ME integrada en el hardware de celulares relativamente modernos que permite el uso de programas java denominados MIDlets, tales como juegos, aplicaciones o todo tipo de software.

MMS: (*Multimedia Messaging System*) Servicio de Mensajería Multimedia. Es un estándar de mensajería que le permite a los teléfonos móviles enviar y recibir contenidos multimedia.

P2P: (*Peer-to-Peer*) es una red de computadoras en la que todos o algunos aspectos de ésta funcionan sin clientes ni servidores fijos, sino una serie de nodos que se comportan como iguales entre sí.

PDA: (*Personal Digital Assistant*) Asistente Digital Personal: es una computadora de mano originalmente diseñado como agenda electrónica (calendario, lista de contactos, bloc de notas y recordatorios) con un sistema de reconocimiento de escritura.

PHP: (*PHP: Hypertext Preprocessor*) es un lenguaje interprete, de programación de servidores web de código abierto, de alto nivel, cuyas órdenes se incrustan entre las etiquetas de las páginas web que han sido escritas en HTML.

Plugin: Es un programa adicional que puede ser añadido a una aplicación para aumentar la funcionalidad de esta.

SLOC: (*Source Lines of Code*) Líneas de Código Fuente: es un software de medida utilizada para medir el tamaño de un programa de software, contando el número de líneas en el texto de la fuente del programa de código.

SMS: (*Short Message Service*) Servicio de Mensajes Cortos. Es un servicio disponible en los teléfonos móviles que permite el envío de mensajes cortos.

URL: (*Uniform Resource Locator*) Localizador Uniforme de Recurso. Es una secuencia de caracteres, de acuerdo a un formato estándar, que se usa para nombrar recursos, como documentos e imágenes en Internet, por su localización.

VAS: (*Value Added Services*) Servicios de Valor Agregado.

XHTML: (*extensible Hypertext Markup Language*) Lenguaje Extensible de Mercado de Hipertexto. XHTML es la versión XML de HTML.

XML: (*extensible Markup Language*) Lenguaje de Marcas Extensible. Es un metalenguaje extensible de etiquetas.

XP: (*Extreme Programming*) Programación Extrema. Es una metodología ágil de desarrollo de software.

Anexos

Anexo #1: Pruebas de Aceptación.

Pruebas de Aceptación para la Historia de Usuario **Acceder al Servicio de Lectura.**

Prueba de Aceptación	
Código: HU1_P1	UH: 1. Acceder al Servicio de Lectura.
Nombre: Acceder al Servicio de Lectura.	
Descripción: Prueba para verificar la funcionalidad de acceso al servicio de lectura de canales RSS.	
Condiciones de Ejecución: La plataforma de J2ME se ejecute correctamente en el móvil. El servicio debe estar iniciado y ejecutándose.	
Entrada / Pasos de Ejecución: Se intenta recibir una solicitud de acceso al servicio.	
Resultado Esperado: La solicitud es atendida correctamente. Se muestra una interfaz con los medios de prensa válidos.	
Evaluación de la Prueba: Prueba satisfactoria.	

Pruebas de Aceptación para la Historia de Usuario **Aceptar Servicio.**

Prueba de Aceptación	
Código: HU2_P1	UH: 2. Aceptar Servicio.
Nombre: Acceder al medio de prensa.	
Descripción: Prueba para verificar la funcionalidad de selección de un medio de prensa.	
Condiciones de Ejecución: El servicio debe estar iniciado y ejecutándose. Se ha seleccionado un medio de prensa.	
Entrada / Pasos de Ejecución: Se intenta crear un link con el medio de prensa seleccionado.	
Resultado Esperado: El link es creado correctamente.	
Evaluación de la Prueba: Prueba satisfactoria.	

Pruebas de Aceptación para la Historia de Usuario **Acceder Tipo de Noticia del Medio de Prensa.**

Prueba de Aceptación	
Código: HU3_P1	UH: 3. Acceder Tipo de Noticia del Medio de Prensa.
Nombre: Acceder al tipo de noticia.	
Descripción: Prueba para verificar la funcionalidad de selección de un tipo de noticia.	
Condiciones de Ejecución: El servicio debe estar iniciado y ejecutándose. Se ha seleccionado un tipo de noticia.	
Entrada / Pasos de Ejecución: Se intenta crear un link con el tipo de noticia seleccionado.	
Resultado Esperado: El link es creado correctamente.	
Evaluación de la Prueba: Prueba satisfactoria.	

Pruebas de Aceptación para la Historia de Usuario **Recibir Resumen de Noticia.**

Prueba de Aceptación	
Código: HU4_P1	UH: 4. Recibir Resumen de Noticia.
Nombre: Recibir Resumen de Noticia.	
Descripción: Prueba para verificar la funcionalidad de crear un resumen de noticias.	
Condiciones de Ejecución: El servicio debe estar iniciado y ejecutándose. Se creará una lista con un resumen de la noticia seleccionada.	
Entrada / Pasos de Ejecución: Se intenta crear una lista con un resumen de noticias.	
Resultado Esperado: La lista es creada y mostrada satisfactoriamente.	
Evaluación de la Prueba: Prueba satisfactoria.	

Pruebas de Aceptación para la Historia de Usuario **Ver Detalles de Noticia.**

Prueba de Aceptación	
Código: HU5_P1	UH: 5. Ver Detalles de Noticia.
Nombre: Ver Detalles de Noticia.	
Descripción: Prueba para verificar la funcionalidad de leer un detalle de una noticia.	
Condiciones de Ejecución: El servicio debe estar iniciado y ejecutándose. Se creará un texto con el detalle de una noticia seleccionada.	
Entrada / Pasos de Ejecución: Se intenta crear un texto con los detalles específicos de una noticia.	
Resultado Esperado: El texto es creado y mostrado satisfactoriamente.	
Evaluación de la Prueba: Prueba satisfactoria.	