

**Universidad de las Ciencias Informáticas.**

**Facultad 2 “Telecomunicaciones y Seguridad**



**Título: Portal WAP para la plataforma de entrega de contenido.**

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

Autor(es): Eduardo Gómez Dorado.

Luis Darlyn Paradela López.

Tutor(es): Ing. Damián Ilizastegui Arriba.

Co-Tutor: Ing. Miguel Ángel Torres Pérez.

Ciudad de la Habana

\_ del 2012

# Pensamiento

## DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor de este trabajo y autorizo al <nombre área> de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

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

"[Insertar nombre(s) de autor(es)]"

\_\_\_\_\_

"[Insertar nombre(s) de tutor(es)]"

\_\_\_\_\_

## DATOS DE CONTACTO

"[insertar breve curriculum e información de contacto del tutor]"

"[insertar breve curriculum e información de contacto del asesor]"

"[insertar breve curriculum e información de contacto del consultante]"

## AGRADECIMIENTOS

"[Insertar agradecimientos (opcional)]"

## DEDICATORIA

"[Insertar dedicatoria (opcional)]"

## RESUMEN

El presente trabajo aborda las formas de entrega de contenidos a dispositivos móviles y como posibilitar su venta a través de Internet. Cuba ha experimentado un crecimiento en la utilización de la tecnología celular, aumentando las entidades desarrolladoras de aplicaciones para dispositivos móviles, lo que ha contrastado con la carencia de portales que permita la comercialización de este tipo de aplicaciones. El propósito trazado consiste en el desarrollo de un portal WAP para la población nacional que permita la descarga de contenidos desde dispositivos móviles. Para ello fue necesario estudiar y analizar los distintos métodos para realizar la entrega de contenido, la adaptación del mismo al dispositivo en cuestión, asignación de restricciones de uso sobre el contenido, y cómo lograr que sean visualizados en los celulares. Además se generaron los artefactos que posibilitan la creación de la aplicación, realizándose pruebas a la misma para comprobar su correcto funcionamiento. Obteniéndose como resultado una aplicación que es visualizada mediante Internet por dispositivos celulares posibilitando a los usuarios de la telefonía móvil de Cuba acceder a los contenidos de los diferentes proveedores.

**Palabras Claves:** portal WAP, contenido, adaptación de contenido.

## ÍNDICE

<b>RESUMEN .....</b>	<b>7</b>
<b>Introducción .....</b>	<b>11</b>
<b>Capítulo 1 “Fundamentación teórica” .....</b>	<b>15</b>
1.1    Introducción .....	15
1.2    Antecedentes.....	15
1.3    Arquitectura de la tecnología WAP. ....	17
1.4    Métodos para la descarga de contenido .....	23
1.5    Administración de Derechos Digitales OMA (OMA DRM) .....	25
1.6    Los ficheros CDR.....	27
1.7    Métodos, Herramientas y Tecnologías a utilizar.....	28
1.8    Conclusiones .....	36
<b>Capítulo 2 “Características del sistema” .....</b>	<b>38</b>
2.1    Introducción .....	38
2.2    Objeto de automatización .....	38
2.3    Propuesta del sistema .....	38
2.4    Modelo de dominio .....	39
2.5    Especificación de los requisitos del sistema .....	44
2.6    Modelo de casos de uso del sistema .....	48
2.7    Conclusiones .....	50
<b>Capítulo 3: Diseño del sistema.....</b>	<b>51</b>
3.1    Introducción .....	51
3.2    Descripción de la arquitectura.....	51
3.3    Modelo de diseño .....	53
3.4    Estándares de nomenclatura y codificación utilizados .....	58
3.5    Descripción de las clases.....	59
3.6    Conclusiones .....	60
<b>Capítulo 4: Implementación y prueba .....</b>	<b>62</b>



4.1	Introducción .....	62
4.2	Modelo de implementación .....	62
4.3	Diagrama de despliegue .....	62
4.4	Diagrama de componentes .....	64
4.5	Pruebas al software .....	66
4.6	Conclusiones .....	74
<b>Conclusiones generales .....</b>		<b>76</b>
<b>Recomendaciones.....</b>		<b>78</b>

## Índice de Figuras

Figure 1: Arquitectura WAP.....	18
Figura 2: Componentes de la arquitectura WAP.....	19
Figura 3: Métodos para los derechos digitales .....	26
Figura 4: Ejemplo de arquitectura para la entrega combinada.....	27
Figura 5: Propuesta del sistema.....	39
Figure 6: Modelo de dominio .....	40
Figure 7: Modelo de dominio Contenido .....	41
Figure 8: Modelo de dominio Valor Agregado.....	42
Figura 9: Diagrama de casos de uso del sistema .....	49
Figura 10: Ejemplo de utilización del patrón creador .....	55
Figura 11: Ejemplo de patrón controlador.....	57
Figura 12: Ejemplo de patrón inyección de dependencias.....	58
Figura 13: Ejemplo de nombre de clase .....	59
Figura 14: Diagrama de despliegue.....	63
Figura 15: Diagrama de componentes .....	64
Figura 16: Diagrama de componentes del paquete manager .....	65
Figura 17: Pruebas de caja blanca .....	66

Figura 18: Función a la que se aplica camino básico .....	68
Figura 19: Grafo de flujo.....	68
Figura 20: Prueba a la funcionalidad lastContentsByDevice.....	72
Figura 21: Resultado de la prueba a la funcionalidad lastContentsByDevice .....	73
Figura 22: Pruebas de caja negra (44) .....	73
Figura 23: Gráfico de resultados de las pruebas .....	74

## **Índice de Tablas**

Tabla 1: Actores del sistema .....	48
Tabla 4: Descripción de los nodos de despliegue.....	63

## **Introducción**

Con el arribo de la telefonía móvil digital se ha abierto un abanico de posibilidades para trasladar los contenidos y servicios de Internet hacia los teléfonos móviles. Surge entonces el protocolo WAP (Wireless Application Protocol o Protocolo de Aplicaciones Inalámbricas) que permitió visualizar en los micros navegadores páginas especialmente diseñadas para móviles. La principal ventaja de este protocolo es permitir la movilidad al tiempo que se accede a los datos de interés.

En la actualidad la tecnología móvil está teniendo un crecimiento a un ritmo vertiginoso, que amenaza con desplazar a las computadoras personales (PCs) para convertirse en el dispositivo mayoritario en el acceso a Internet. Esto no significa acceso pleno a Internet, no se debe olvidar las limitaciones que tienen estos dispositivos, las pantallas son notablemente más pequeñas por lo que no es posible presentar más que algunas pocas líneas de texto. Se deberá tener en cuenta a la hora de entrar datos que esto requerirá tiempo adicional del usuario debido al teclado que presentan. También la capacidad de memoria y de procesador es mucho menor. Sus redes inalámbricas tienen menos ancho de banda y es más lenta por lo que los datos a mostrar deben ser pequeños, como íconos y limitadas imágenes.

Este crecimiento en el acceso a Internet es debido a que muchas de las empresas que ofertan servicios mediante la red de redes han adaptados sus páginas para que sean accedidas desde un dispositivo móvil. En Cuba desde hace solo unos años ha habido un incremento en el número de usuarios de la telefonía celular, aumentando gradualmente los servicios que se prestan a través de ella. Como parte de los esfuerzos realizados por la Empresa de Telecomunicaciones de Cuba (ETECSA) en esta rama, se pretende brindar a la población la posibilidad de acceder a Internet desde los teléfonos celulares. Sin embargo para ello no cuenta con una aplicación que sea visualizada en los dispositivos móviles y que permita brindar servicios de valor agregado así como la descarga de contenidos de interés.

Por tal motivo ETECSA solicitó a la Universidad de las Ciencias Informáticas el desarrollo de un sistema informático que brinde las prestaciones necesarias para la

# Capítulo 1 “Fundamentación Teórica”

## Portal WAP para la plataforma de entrega de contenido

Universidad de las Ciencias Informáticas

---

descarga de contenidos tales como imágenes, videos, tonos, aplicaciones y juegos a través de los teléfonos celulares y ofrezca servicios de valor agregado como promociones, estado del tiempo, noticias y directorio telefónico.

Las descargas además deben ser compatibles con el dispositivo móvil que realice la petición, debe por adición garantizar los derechos digitales de los proveedores, así como generar reportes en un formato específico de facturación para ETECSA. Debido a la necesidad existente surge el siguiente **problema a resolver**: ¿Cómo posibilitar la venta y entrega de contenidos y servicios de valor agregado a los usuarios de telefonía móvil en Cuba? En la búsqueda de una solución se toma como **objeto de estudio** las aplicaciones para la venta y entrega de contenidos. Constituye entonces como **objetivo general** de la investigación: Desarrollar un portal WAP para la venta y entrega de contenidos a móviles. En la investigación a desarrollar se define como **campo de acción**: Portales WAP para la venta y entrega de contenidos. Para obtener un enfoque más preciso y detallado de la investigación que se desea realizar, se plantean los siguientes **objetivos específicos**:

- Realizar el marco teórico de las aplicaciones de entrega de contenidos para dispositivos móviles y de las herramientas, metodologías y tecnologías para el proceso de desarrollo de la solución.
- Desarrollar un módulo que permita la entrega de contenidos multimedia de diferentes proveedores y que estos sean adaptables a los dispositivos móviles que los soliciten.
- Desarrollar un módulo que permita proteger el contenido multimedia mediante técnicas de DRM<sup>1</sup> ya implementadas.
- Desarrollar un subsistema de usuarios que permita a estos registrarse y autenticarse para descargar contenidos.
- Desarrollar un módulo que almacene un registro de todas las descargas realizadas por los usuarios.
- Desarrollar los siguientes servicios de valor agregado:

---

<sup>1</sup> Administración de Derechos Digitales por su traducción al español

# Capítulo 1 “Fundamentación Teórica”

Portal WAP para la plataforma de entrega de contenido

Universidad de las Ciencias Informáticas

---

- Sistema de publicidad
- Sistema de noticias.
- Estado del tiempo.
- Búsqueda en el directorio telefónico y páginas amarillas.

Las siguientes **preguntas científicas** guiarán la investigación a realizar:

- ¿Cómo permitir la visualización del sitio en los dispositivos móviles?
- ¿Cómo lograr la correcta entrega de contenidos?
- ¿Cómo garantizar las restricciones de uso impuestas por los proveedores a los contenidos?
- ¿Cómo generar el reporte de las descargas para la facturación del contenido?
- ¿Cómo mostrar los servicios de valor agregado?
- ¿Cómo garantizar la seguridad de la aplicación?

Para la correcta planificación y evaluación de la calidad, así como el seguimiento del avance de la investigación, se han desarrollado las **siguientes tareas de investigación**:

- Elaborar un estudio del arte para la venta de contenidos de dispositivos móviles.
- Definir la arquitectura de la aplicación.
- Evaluar las herramientas a utilizar para el desarrollo de la aplicación.
- Realizar el levantamiento de requisitos.
- Realizar el diseño de la aplicación.
- Implementar las funcionalidades de la aplicación.
- Diseñar e implementar los casos de prueba.
- Realizar pruebas al sistema.

# *Capítulo 1 “Fundamentación Teórica”*

*Portal WAP para la plataforma de entrega de contenido*

*Universidad de las Ciencias Informáticas*

---

Durante el transcurso de la investigación se usarán distintos tipos de métodos de investigación específicamente los métodos teóricos:

**Analítico – Sintético:** Este método permitirá extraer los elementos más relevantes relacionados con el objeto de estudio, posibilitando así una mejor comprensión de la documentación existente. El cual se ve evidenciado en el capítulo uno durante la investigación de los sistemas y tecnologías a utilizar.

**Histórico – Lógico:** Este método ayuda a obtener la evolución y desarrollo histórico del tema de investigación y de otros temas relacionados. La utilización del método presenta la evolución y tendencias de la tecnología WAP.

Durante la realización de la investigación se utiliza el método empírico empleado para la acumulación de información sobre la investigación en curso es utilizado más específicamente la:

**Observación:** Mediante este método es posible la obtención de la información directa de sistemas similares al deseado, obteniéndose el conocimiento necesario de las características presentes en ellos.

La siguiente investigación se encuentra estructura en cuatro capítulos donde:

El capítulo uno llamado “Fundamentación Teórica” abarca el estudio del estado del arte, analizando la tecnología WAP y su utilización en la entrega de contenidos a teléfonos móviles. Se exponen conceptos fundamentales y se presenta una breve descripción de las herramientas y metodologías a utilizar en el desarrollo de la investigación.

En el capítulo dos titulado “Características del Sistema” se exponen los requisitos funcionales y no funcionales. También se realiza una descripción detallada de los casos de usos y se refleja el resultado del negocio de la aplicación.

En el capítulo tres titulado “Diseño del Sistema” donde se exponen el diseño de clases e iteración de la aplicación, así como el flujo de datos existente entre las clases. Se describe la arquitectura de la aplicación y los patrones de diseños utilizados.

# *Capítulo 1 “Fundamentación Teórica”*

*Portal WAP para la plataforma de entrega de contenido*

*Universidad de las Ciencias Informáticas*

---

En el capítulo cuatro llamado “Implementación y Prueba” se exponen los principales pasos para la implementación de la aplicación, así como la realización de pruebas para reducir los posibles errores.

## **Capítulo 1 “Fundamentación teórica”**

### **1.1 Introducción**

El presente capítulo aborda el estado del arte de aplicaciones existentes a nivel mundial que proveen servicios similares a los solicitados por el cliente. También las principales características de los métodos de descarga, exponiendo las ventajas y desventajas que poseen, y se hace alusión a la evolución de las tecnologías que posibilitan el acceso a internet mediante los dispositivos móviles. Además se abordan conceptos que facilitan la comprensión del trabajo y se describen las herramientas, metodologías y lenguajes a utilizar durante el desarrollo de la aplicación.

### **1.2 Antecedentes**

El teléfono móvil es un dispositivo inalámbrico electrónico que permite tener acceso a la red de telefonía celular o móvil. Se denomina celular en la mayoría de países latinoamericanos debido a que el servicio funciona mediante una red de celdas, donde cada antena repetidora de señal es una célula, si bien también existen redes telefónicas móviles satelitales. (1) Su principal característica es su portabilidad, que permite comunicarse desde casi cualquier lugar. La principal función es la comunicación de voz, como el teléfono convencional.

Estos dispositivos además se han convertido en herramientas de entretenimiento y trabajo. Se pueden usar a modo de cámaras fotográficas, como vía para pasar el rato gracias a los juegos, y con el desarrollo de las redes inalámbricas fue posible leer emails y hasta navegar por Internet. (2)

La cantidad de usuarios que utilizan estos dispositivos para navegar por Internet se incrementa cada día. Por ello muchos proveedores de servicios han creado portales para ser visualizado desde estos dispositivos. Entre los que se encuentra m.gmail.com para acceder a la cuenta Gmail desde el móvil, m.youtube.com sitio

# *Capítulo 1 “Fundamentación Teórica”*

## *Portal WAP para la plataforma de entrega de contenido*

*Universidad de las Ciencias Informáticas*

---

para ver, compartir y subir videos. Existen muchos otros que permiten obtener las noticias del día, la descarga de determinados programas, otros creados para conocer amigos como el sitio m.facebook.com. El portal que brinda la empresa NOKIA permite la descarga de contenidos, además de presentar servicios como tienda, mensajería, entre otros.

Mobango es otro de los portales muy visitados que se especializa en la descarga de contenidos, permite publicarlos y adaptarlos al móvil que realiza la descarga. Estas aplicaciones tienen gran similitud con algunas de las características solicitadas, pero solo abarcan un conjunto de ellas siendo descartadas para el empleo en la solución deseada.

Empresas proveedoras de servicios para móviles como TELCEL y MOVISTAR cuentan con portales diseñados para estos dispositivos, por el cual brindan sus servicios y productos.

Al acceder al portal móvil de TELCEL se obtiene servicios como la compra de contenidos, la actualización de las redes sociales como Facebook, Twitter y WordPress, navegar en sitios de información del Clima, Sección Amarilla entre otros.

(3)

El portal móvil de MOVISTAR oferta servicios como descarga de contenidos, acceso al estado del tiempo, actualizar los perfiles de las redes sociales como Facebook y Twitter. (4) Estos sitios presentan muchos de los servicios que son solicitados por ETECSA, pero la utilización de uno de ellos presenta algunos inconvenientes. Esto se debe a que dicho portal pertenece a una empresa con la cual debe obtenerse un acuerdo para el derecho de uso y luego realizarse las modificaciones necesarias para cumplir con los servicios solicitados.

En nuestro país la Empresa Cubana Nacional de Software (Desoft) ha desarrollado aplicaciones como la plataforma “Bluye Eye” que permite a los usuarios de la telefonía celular de Cuba acceder a diferentes servicios como, información de eventos culturales, deportivos, estado del tiempo y números telefónicos de lugares de interés, mediante mensajes de textos SMS (en español Servicios de Mensajes Cortos). Esta plataforma no muestra en los navegadores móviles los servicios requeridos, además no permite la descarga de contenidos.



# *Capítulo 1 “Fundamentación Teórica”*

## *Portal WAP para la plataforma de entrega de contenido*

*Universidad de las Ciencias Informáticas*

---

Teniendo en cuenta que no existe un producto que se adapte a las características solicitadas y que aplicaciones similares no pueden ser utilizadas por motivos de licencias de software, además de modificaciones que se les debe realizar, se decide el desarrollo de una aplicación para dispositivos móviles que permita a los usuarios de la telefonía móvil en Cuba acceder a los servicios requeridos.

Dicho portal será soportado por el protocolo WAP, donde un gran número de dispositivos tiene soporte para él, posibilitando crear aplicaciones que serán visualizadas en el micro-navegador de los teléfonos celulares y trabajar con diferentes redes de comunicación inalámbrica. (5) Estas aplicaciones son ajenas a las implementaciones de las capas inferiores del protocolo, convirtiéndolas en portales web que cumplen con las exigencias y características de estos dispositivos.

A continuación se muestra la arquitectura del protocolo WAP, así como las principales especificaciones para la creación de las páginas con el objetivo de que sean soportadas por él.

### **1.3 Arquitectura de la tecnología WAP.**

La tecnología WAP está posicionada en el punto de convergencia de dos tecnologías de red que evolucionan de forma muy rápida: las comunicaciones inalámbricas y la Internet. (5) Además incluye las especificaciones a nivel de datos transmitidos por la red de comunicaciones y las especificaciones a nivel de Internet como son: XML, URLs y los formatos de datos con los que se va a trabajar. (5) Este ha sido el factor fundamental para el acceso a la información a través de los dispositivos inalámbricos. En las soluciones implementadas se debe tener en cuenta que estos poseen limitantes en cuanto a potencia de procesamiento, menos memoria RAM, pantallas muy pequeñas, menor ancho de banda y teclados limitados.

La comunicación que lleva a cabo WAP 1.x se realiza mediante un WAP Gateway que se encarga de convertir la información enviada por el teléfono hacia el servidor web, y este responde al Gateway realizándose el proceso contrario. Esta tecnología WAP se basa en 3 elementos:

# Capítulo 1 “Fundamentación Teórica”

## Portal WAP para la plataforma de entrega de contenido

Universidad de las Ciencias Informáticas

- Cliente: el cliente será el micro-navegador WML de un terminal móvil. Este será el equivalente al navegador de un PC en el Internet "fijo" que todos conocemos.
- Pasarela: la pasarela constituye la interfaz entre la red inalámbrica y la red física. Es lo que se conoce como el WAP Gateway.
- Servidor web: es el encargado de procesar las peticiones del cliente y enviar las páginas solicitadas.

La figura 1 muestra el flujo de eventos. (6)

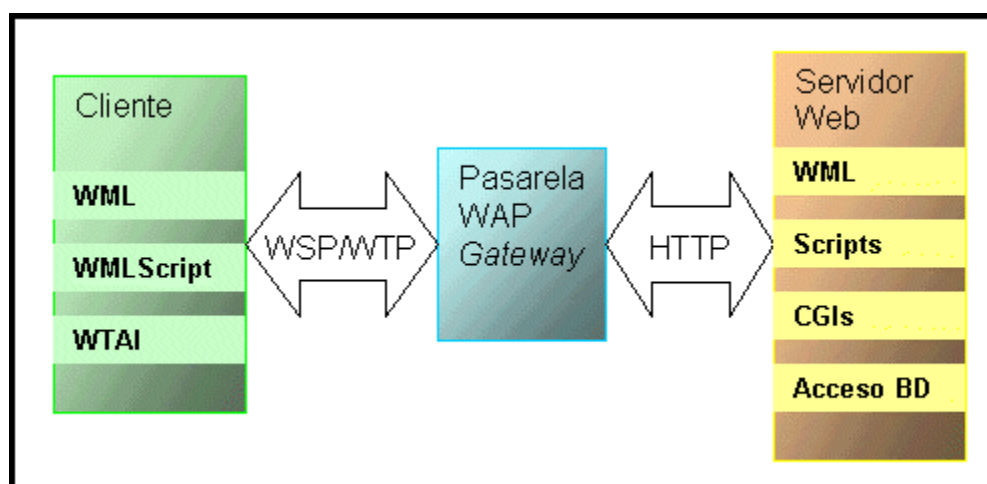


Figure 1: Arquitectura WAP

WAP funciona de la siguiente forma: el cliente tiene un teléfono móvil que soporta la tecnología WAP, desde su teléfono realiza una petición codificada que envía a través de las redes móviles al servidor, que ejerce la función de pasarela y que tiene codificadores y decodificadores.

El GATEWAY WAP o pasarela WAP tiene la función de ser un traductor puro o compilador de lenguajes, (5) tanto de entrada como de salida. Este traduce los paquetes que vengan con formato WML y WMLScript con encabezado del protocolo WAP a WML y WMLScript con encabezado de protocolo web HTTP (en español Protocolo de transferencia de hipertexto) respectivamente, tanto del teléfono que soporta el protocolo WAP al servidor web y viceversa, en respuesta para el teléfono.

Los componentes que forman parte pila de los protocolos de WAP son los mostrados en la figura 2. (5)

# Capítulo 1 “Fundamentación Teórica”

Portal WAP para la plataforma de entrega de contenido

Universidad de las Ciencias Informáticas

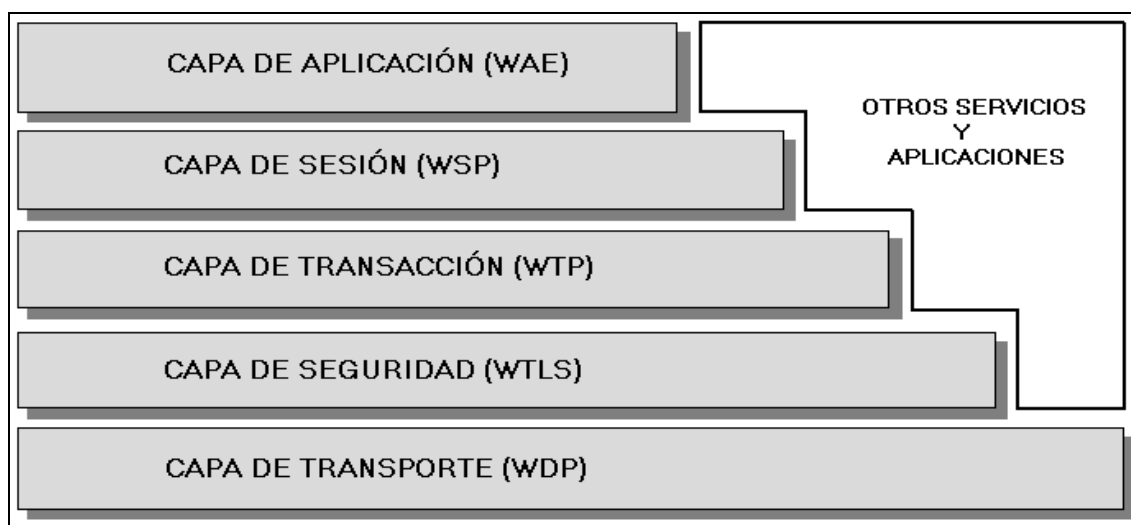


Figura 2: Componentes de la arquitectura WAP

La **Capa de aplicación** (WAE en inglés Wireless Application Environment) es una capa de propósito general que aúna la tecnología de Internet y la tecnología de la telefonía móvil digital. Su principal objetivo es proporcionar la interoperabilidad necesaria tanto a los fabricantes de telefonía como a los operadores que proporcionan servicios de Internet. (5) Está enfocado principalmente sobre los aspectos del cliente de la arquitectura del sistema de WAP que incluye un micro navegador que soporta WML y WMLScript.

El **Protocolo inalámbrico de sesión** (WSP) constituye la capa que se sitúa por debajo de la capa de aplicación, proporcionando la capacidad necesaria para:

- Establecer una conexión fiable entre el cliente y el servidor, y liberar esta conexión de una forma ordenada.
- Intercambiar contenido entre el cliente y el servidor utilizando codificación compacta.
- Suspender y recuperar la sesión.

El **Protocolo inalámbrico de transacción** (WTP) se establece para proporcionar los servicios necesarios que soporten aplicaciones de “navegación” (del tipo petición/respuesta).

# Capítulo 1 “Fundamentación Teórica”

## Portal WAP para la plataforma de entrega de contenido

Universidad de las Ciencias Informáticas

---

La **Capa inalámbrica de seguridad de transporte** (WTLS) es un protocolo basado en el estándar SSL, utilizado en el entorno Web para la proporción de seguridad en la realización de transferencias de datos. (7)

La **Capa de transporte** (WDP en inglés Wireless Datagram Protocol) es la capa de transporte en la arquitectura WAP. Opera sobre las portadoras de datos que se emplean en las comunicaciones. Ofrece un servicio general de transporte a las capas superiores de dicha arquitectura. (5)

Con el avance de la tecnología móvil y la aparición de redes inalámbricas como GPRS<sup>2</sup> que permite circular datos a mayor velocidad, se crea la versión 2.0 de WAP, en la cual se incluyen protocolos de Internet.

### 1.3.1 Versión WAP 2.0

WAP 2.0, agrega soporte para el protocolo estándar de comunicación de Internet tales como HTTP en la capa de aplicación, TLS (en español Seguridad en la Capa de Transporte) y TCP (en español Protocolo de Control de Transmisión) en la capa de transporte y optimiza el uso de ancho de banda. Este protocolo utiliza XHTML-MP (XHTML Mobile Profile), un subconjunto de XHTML que incluye el XHTML básico, y WCSS (WAP CSS), un subconjunto de CSS2 más ciertas extensiones específicas para móviles, como lenguajes para la presentación de las interfaces de usuario mejorando por ejemplo el soporte de los gráficos (incluye color). De esta forma se consigue que el diseño de contenidos con WAP 2.0 sea muy similar al diseño de contenidos de la WWW (en inglés **World Wide Web**). La pasarela WAP no es estrictamente necesaria en esta especificación, pero su presencia puede tener funciones útiles como caché web y para dar soporte a las opciones de TCP y HTTP antes mencionadas. (8)

Actualmente los dispositivos móviles son capaces de visualizar portales que son diseñados para la WWW, dejando de lado el concepto de crear aplicaciones específicamente para dispositivos móviles. Sin embargo se determina la utilización

---

<sup>2</sup> General Packet Radio Service, o tecnología de transmisión de datos por conmutación de paquetes.

# Capítulo 1 “Fundamentación Teórica”

## Portal WAP para la plataforma de entrega de contenido

Universidad de las Ciencias Informáticas

---

de la especificación de WAP 2.0 para la creación de los portales de la aplicación, debido a que es soportado por una gran variedad de dispositivos, y permite que se acceda desde redes inalámbricas de bajas velocidades. La creación de un portal WAP, que son aquellos creados utilizando la especificación del protocolo WAP, posibilita posicionar la aplicación en un punto de convergencia entre los celulares que solamente soportan este protocolo, y los que incluyen tecnologías propias de internet, permitiendo que un mayor número de usuarios pueda acceder a la aplicación.

### 1.3.2 Lenguajes de etiquetado

Es una forma de codificar un documento incorporando etiquetas o marcas que contienen información adicional acerca de la estructura del texto o de su presentación. El lenguaje de marcas más extendido y que es fundamento de la Web es el HTML. Sin embargo, este concepto no debe confundirse con los lenguajes de programación ya que éstos poseen funciones aritméticas o variables, a diferencia de los lenguajes de marcado. (9)

#### **WML Wireless Markup Lenguaje.**

El Wireless Markup Language, es un lenguaje de etiquetado del protocolo WAP 1.1 cuyo origen es el XML (eXtensible Markup Language). Este lenguaje se utiliza para construir las páginas que aparecen en las pantallas de los teléfonos móviles y los asistentes personales digitales (PDA) dotados de tecnología WAP. Es una versión reducida del lenguaje HTML (Hypertext Markup Language, en español Lenguaje de Marcado de Hipertexto) que facilita la conexión a Internet de dichos dispositivos y que además permite la visualización de páginas web en dispositivos inalámbricos que incluyan la tecnología WAP. La visualización de la página dependerá del dispositivo que se use y de la forma en que este interprete el código, ya que varían entre sí. WML es un metalenguaje, lo que implica que además de usar etiquetas predefinidas se pueden crear componentes propios y tiene ciertas similitudes con otro lenguaje de etiquetas bastante conocido, el HTML, utilizado para la creación de páginas web convencionales. (10)

# Capítulo 1 “Fundamentación Teórica”

Portal WAP para la plataforma de entrega de contenido

Universidad de las Ciencias Informáticas

---

## XHTML Mobile Profile (XHTML MP)

XHTML MP (del inglés: Extensible HyperText Markup Language Mobile Profile) es el lenguaje de etiqueta definido en la especificación de WAP 2.0, determinado en la especificación de servicios móviles creado por la WAP Forum ahora la Open Mobile Alliance (OMA).

XHTML MP es un subconjunto del XHTML que es la versión más estricta del HTML. XHTML Mobile Profile es HTML básico y también un subconjunto del XHTML sumándole algunos de los elementos y atributos de la versión completa del XHTML. El objetivo de XHTML MP es unir la tecnología de navegación a través de un móvil con las de los ordenadores. (11)

## API<sup>3</sup> Wall Next Generation (WNG)

WALL (Wireless Abstraction Library, en español biblioteca de abstracción móvil) es una biblioteca de etiquetas JSP que permite al desarrollador escribir páginas móviles parecidas al HTML plano, mientras que entrega WML o XHTML - MP al dispositivo el cual origina peticiones HTTP, dependiendo de las características del dispositivo en sí mismo. Estas son consultadas dinámicamente utilizando la API de WURFL (**W**ireless **U**niversal **R**esource **F**ile en español Archivo de Recursos Móviles Universal). (12)

De los lenguajes mencionados anteriormente, se decide utilizar el XHTML MP, de la especificación de WAP 2.0, debido a que los teléfonos celulares tienen cada vez más soporte sobre este lenguaje, posibilitando explotar sus características y crear aplicaciones atractivas, además de ser el básico recomendado por la W3C. La condición de Cuba de país trae como consecuencia que existan dispositivos que solamente soporten el protocolo WAP 1.1. Puesto que la aplicación está dirigida fundamentalmente a los usuarios cubanos, es una prioridad que la mayor cantidad de la población pueda acceder a los datos mostrados en ella. Para mitigar esta

---

<sup>3</sup> Interfaz de programación de aplicaciones. 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.

amenaza, se realiza una vista básica del portal utilizando WNG, mostrando en los micros navegadores el portal que sea soportado por el dispositivo.

## 1.4 Métodos para la descarga de contenido

La descarga de contenidos es el proceso de copiar un archivo en línea hacia el dispositivo propio. La aplicación debe permitir la descarga de contenidos por el cliente. Debe entregarlo además de forma segura, confiable, evitando el malware y garantizando la integridad de la entrega. Se debe también proteger los derechos de autor sobre el contenido. Existen diferentes tipos de métodos para la entrega de contenidos entre los que se encuentran:

### 1.4.1 Descarga Directa

Este método es posible utilizarlo en un gran número de móviles. Es el comúnmente usado en internet, donde la URL para la descarga termina con el nombre del archivo y su extensión. Este método tiene como inconveniente que no garantiza la entrega correcta del contenido, no existe la forma de garantizar la protección del contenido. El método no posibilita validar la entrega realizada para evitar facturaciones indebidas. La descarga directa no cumple con las condiciones necesarias para permitir que la generación de los reportes de las descargas y su posterior facturación sea un proceso confiable como se debe tener durante la entrega del contenido. (13)

### 1.4.2 Descarga de Midlets

Este método permite descargar aplicaciones J2ME e incluye una nueva característica y es el envío de un fichero o descriptor JAD (del inglés Java Application Descriptor). (14) El descriptor posee varios parámetros donde uno fundamental es la URL de notificación donde será enviado el estado de la descarga cuando esta haya terminado.

Este método permite la entrega segura de aplicaciones java, permitiendo conocer el estado final de la descarga. Con la utilización de este método se resuelve el problema de la generación de los reportes de descarga indebida por entrega de un contenido incompleto. Sin embargo la descarga de Midlets solo permite descargar aplicaciones J2ME lo cual no es suficiente para la cantidad de contenidos que pueden ser utilizados por los dispositivos móviles.

#### 1.4.3 OMA<sup>4</sup> Download (Descarga OMA)

Descarga OMA se basa en dos servicios de descarga ya existentes y exitosos: Básico Descarga HTTP y descarga de MIDlets.

El objetivo de la Descarga OMA es proporcionar un servicio similar a Descarga MIDlet. La diferencia entre Descarga MIDlet y Descarga OMA es que este último no está diseñado específicamente para la descarga de aplicaciones J2ME, o cualquier otro tipo de media específica, Descargar OMA es un marco general de descarga. Las similitudes entre los dos servicios de descarga permiten la reutilización de la infraestructura de servidor.

Descarga OMA puede descargar cualquier tipo de objeto multimedia. El típico destino de los objetos multimedia descargados por Descarga OMA, es el almacenamiento de manera persistente en el dispositivo, con el fin de personalizar el dispositivo o mejorar su funcionalidad.

Descarga OMA, extiende de HTML Descarga Básico con dos pasos adicionales:

1. Antes de descargar, una descripción del objeto multimedia de comunicación es descargada. La descripción es un pequeño archivo, el descriptor de descarga, que contiene metadatos sobre el objeto multimedia y las instrucciones para el agente de la descarga en el dispositivo, como descargar el objeto multimedia.
2. Después de la descarga, un informe de situación se registra en el sitio Web, lo que indica el resultado de la descarga. (15)

La entrega de este descriptor de descarga se puede realizar de dos formas diferentes, mediante la Entrega Separada y la Entrega Combinada.

La Entrega Separada realiza la descarga de contenido en dos pasos fundamentales. Primero se envía el descriptor especificando los datos del contenido así como la URL para la descarga y la de notificación, luego el dispositivo verifica si soporta el contenido y luego si es soportado realiza la petición del contenido.

---

<sup>4</sup> Open Mobile Alliance. Organización de estándares que desarrolla estándares abiertos para la industria de telefonía móvil.



# *Capítulo 1 “Fundamentación Teórica”*

## *Portal WAP para la plataforma de entrega de contenido*

*Universidad de las Ciencias Informáticas*

---

La Entrega Combinada es similar a la anteriormente mencionada, solo que la entrega del descriptor y el contenido se realiza en un solo flujo de petición respuesta.

La utilización de este estándar se debe en gran medida a que entrega una confirmación de la descarga, creando un reporte de estado que puede ser posteoado en la web luego de terminada la descarga. Esto posibilita la monitorización de la calidad del servicio y sirve de base para la facturación. Con esto también se tiene la posibilidad de considerar los derechos digitales de los contenidos.

Se debe permitir la descarga de contenidos desde la mayor cantidad posible de celulares, por tal motivo se decide implementar los tres métodos de descarga. Esto brinda la posibilidad de que si un dispositivo no soporta la Descarga OMA y pretende descargar una aplicación java esta se realiza mediante el método de Descarga Midlets que posibilita una entrega confiable. De no ser soportado ninguno de los dos métodos anteriores se procede a realizar la Descarga Directa que es soportada por un número mayor de celulares.

### **1.5 Administración de Derechos Digitales OMA (OMA DRM)**

Hay una necesidad de los proveedores de contenido y operadores para controlar el uso de descargas de objetos multimedia. Descargar es el medio por el cual el objeto multimedia es entregado al dispositivo, mientras que la Administración de los Derechos Digitales (DRM por sus siglas en inglés) es el medio para controlar la el uso del objeto multimedia una vez que se ha descargado.

DRM permite a los proveedores de contenido definir los derechos de los objetos multimedia. Es posible asociar diferentes derechos con un solo objeto multimedia. Diferentes derechos pueden tener diferentes precios. DRM hace posible vender los derechos para usar los objetos multimedia, en lugar de venderlos a ellos.

Esto son algunos ejemplos de derechos soportados por OMA DRM:

1. El derecho de mostrar, ejecutar e imprimir el objeto multimedia.
2. El derecho de mostrar, ejecutar e imprimir el objeto multimedia un número limitado de veces.

# Capítulo 1 “Fundamentación Teórica”

## Portal WAP para la plataforma de entrega de contenido

Universidad de las Ciencias Informáticas

3. Derecho de mostrar, ejecutar e imprimir el contenido en un intervalo de tiempo. (15)

Con este estándar se protegerán los derechos digitales de los proveedores de contenidos, posibilitando un mayor control sobre los mismos.

Los métodos DRM se encuentran representados en la figura 3, tomado de la especificación de OMA DRM. (16)

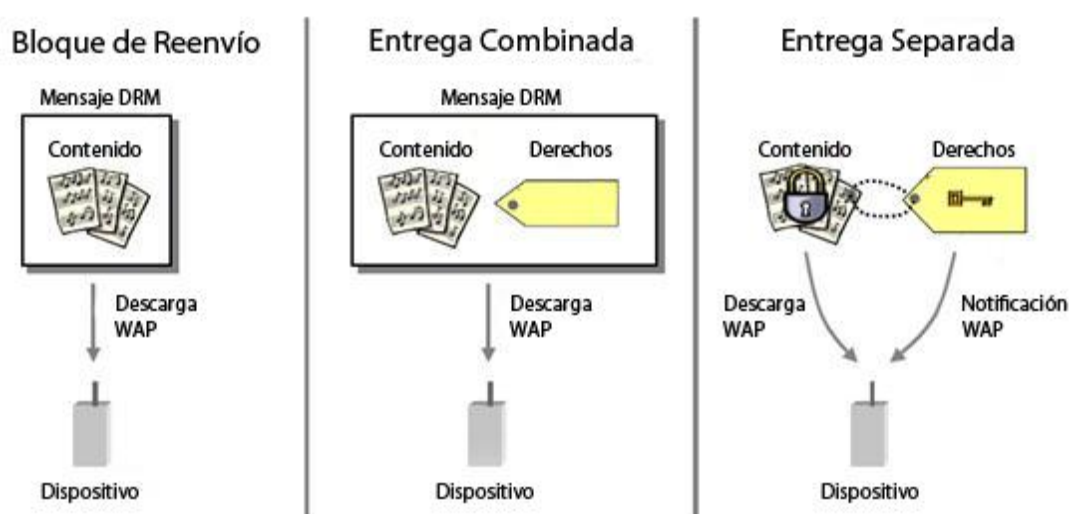


Figura 3: Métodos para los derechos digitales

El método de bloqueo de reenvío posibilita que el usuario no pueda compartir el contenido a través de Bluetooth. El método tiene una desventaja debida a que el contenido puede ser transferido mediante un cable USB hacia una PC y este luego se puede copiar hacia otros dispositivos.

Existen otros métodos para la protección del contenido que utilizan un descriptor que define los permisos play, display, execute y print que se le asignen a cada contenido. La entrega combinada utiliza este descriptor enviándose junto con el contenido en un solo flujo de petición respuesta. El archivo de derechos se entrega en texto plano provocando la posibilidad de que sean violados los derechos del contenido.

La figura 4 muestra un ejemplo de la arquitectura para el método de entrega combinada. (16)

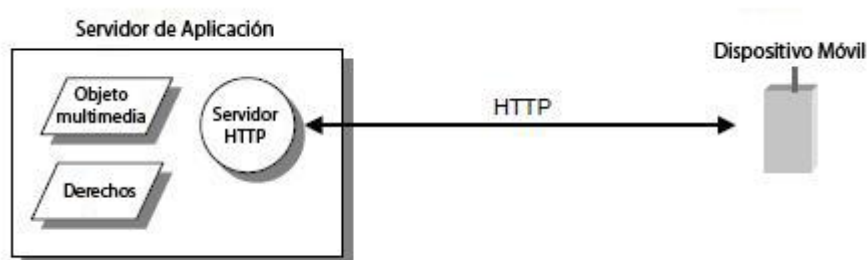


Figura 4: Ejemplo de arquitectura para la entrega combinada

Mediante el método de entrega separada el descriptor se entrega en una notificación (WAP Push) aparte del contenido. Además el proveedor del contenido necesita convertir el texto plano en DCF (del inglés DRM Content Format), esto incluye la encriptación del contenido mediante AES generando una llave de 128 bits que impide que los usuarios que no tengan acceso a esta no puedan acceder al contenido. Este método parecería el más seguro ya que encripta el contenido pero su debilidad radica en que la llave es entregada por un canal no seguro como el WAP Push. (16)

Es empleado en la aplicación los métodos propuestos por el estándar, permitiendo aplicar las restricciones de uso que imponen los proveedores sobre sus contenidos, evitando el uso indebido de los mismos.

## 1.6 Los ficheros CDR<sup>5</sup>

Los CDR (en inglés Call Details Record) son el archivo que contiene la información del uso del sistema, como identidad de la fuente, del destino, el monto a pagar entre otros datos que se requieran. (17) Luego de realizada la entrega y se notifique que fue satisfactoria se procede a generar este fichero, el cual contendrá todos los datos referente al cliente y al contenido descargado. Posteriormente con dicho fichero se realizará el cobro del contenido a través del sistema de facturación de ETECSA.

La utilización de estos ficheros permite almacenar los datos referentes a las descargas, para luego ser entregados a ETECSA y que esta realice el cobro del contenido descargado.

---

<sup>5</sup> Call Details Record

## 1.7 Métodos, Herramientas y Tecnologías a utilizar

Para la construcción de la aplicación se ha definido la utilización de los siguientes métodos, herramientas y técnicas: RUP (Proceso Unificado de Desarrollo por su significado en español) como metodología de desarrollo, JAVA como lenguaje de programación, utilizando como plataforma de desarrollo Java 2 Enterprise Edition (J2EE), con el IDE Eclipse, utilizando el Framework Spring y como servidor se utiliza Apache Tomcat 6.0.

### 1.7.1 Metodología de desarrollo de software

Una metodología de desarrollo de software es un conjunto de procedimientos, técnicas, herramientas y un soporte documental para el desarrollo de productos software, puede seguir uno o varios modelos de ciclo de vida, el ciclo de vida indica qué es lo que hay que obtener a lo largo del desarrollo del proyecto pero no cómo hacerlo, mientras que la metodología indica cómo hay que obtener los distintos productos parciales y finales.

#### 1.7.1.1 Proceso Unificado de Desarrollo

**Proceso Unificado de Desarrollo (RUP):** es una metodología de desarrollo de software que está basada en componentes e interfaces bien definidas, y junto con el **Lenguaje Unificado de Modelado (UML)**, constituye una de las metodologías utilizadas para el análisis, implementación y documentación de sistemas orientados a objetos.

Algunos aspectos que diferencian a RUP de las demás metodologías y lo que lo hace único, es que en RUP, los casos de uso no son sólo una herramienta para especificar los requisitos del sistema, sino que también guían su diseño, implementación y prueba. Los casos de uso constituyen un elemento integrador y una guía del trabajo. (18)

Además de utilizar los casos de uso para guiar el proceso; se presta especial atención al establecimiento temprano de arquitectura robusta que no se vea fuertemente impactada ante cambios posteriores durante la construcción y el mantenimiento. También este propone que cada fase se desarrolle en iteraciones.

Se decide utilizar esta metodología porque es un proceso que puede especializarse para una gran variedad de sistemas de software, en diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de aptitud y diferentes tamaños de proyecto. Además porque la gran cantidad de documentación permite que todo el equipo de desarrollo sea guiado hacia el avance del desarrollo del software, conociendo cada cual lo que debe hacer en qué momento. También es utilizada por la forma disciplinada de asignar tareas y responsabilidades (quién hace qué, cuándo y cómo), por el desarrollo iterativo de software, administración de requisitos, uso de arquitectura y modelado visual del software.

## 1.7.2 Herramienta CASE<sup>6</sup>

Las herramientas CASE (Ingeniería de Software Asistida por Computadora) son aplicaciones informáticas que posibilitan aumentar la productividad en el desarrollo de software, permite mejorar el tiempo de desarrollo, mantenimiento de los sistemas informáticos y mejorar la productividad. Estas herramientas nos pueden ayudar en todos los aspectos del ciclo de vida de desarrollo del software en tareas como el proceso de realizar un diseño del proyecto, implementación de parte del código automáticamente con el diseño dado. Estas herramientas tienen como objetivos mejorar la planificación de un proyecto, automatizar el desarrollo del software, la documentación, la generación de código, las pruebas de errores y la gestión del proyecto, ayuda a la reutilización del software, portabilidad y estandarización de la documentación. (19)

### 1.7.2.1 Visual paradigm para UML 8.0

Visual Paradigm para UML es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El software de modelado UML ayuda a una rápida construcción de aplicaciones con calidad y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. (20)

---

<sup>6</sup> *Computer Aided Software Engineering*, Ingeniería de Software Asistida por Computadora

Se utiliza esta herramienta CASE para la realización del modelado de la aplicación, fundamentalmente por su característica multiplataforma, y además por su generación de código.

### 1.7.3 Lenguaje Unificado de Modelado (UML)

**Lenguaje Unificado de Modelado (UML**, por sus siglas en inglés, Unified Modeling Language) es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. Es un sistema de notación elegante, expresivo y flexible, que entre otras cosas incluye el significado de sus notaciones. UML no guía al desarrollador en la forma de realizar el análisis y diseño orientados a objetos, ni le indica cual proceso de desarrollo adoptar. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocio, funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes reutilizables. (21)

Es importante resaltar que UML es un "lenguaje de modelado" para especificar o para describir métodos o procesos. Se utiliza para definir un sistema, para detallar los artefactos en el sistema y para documentar y construir pero no dice cómo implementar dicho sistema. En otras palabras, es el lenguaje en el que está descrito el modelo.

Con este tipo de lenguaje de modelado el equipo de desarrollo cuenta con cierta experiencia adquirida en el transcurso de los años de estudio. Además es utilizado por la herramienta CASE que será utilizada en la investigación para modelar los diagramas.

### 1.7.4 Java 2 Enterprise Edition (J2EE) plataforma de desarrollo

El Java 2 Platform, Enterprise Edition (J2EE) define el estándar para el desarrollo de aplicaciones empresariales de varios niveles. La plataforma J2EE simplifica las aplicaciones empresariales basándolas en componentes estandarizados y modulares, proporcionando un conjunto completo de servicios a esos componentes, y manejando muchos detalles de comportamiento de las aplicaciones de forma automática, sin necesidad de programación compleja. (22)

# Capítulo 1 “Fundamentación Teórica”

## Portal WAP para la plataforma de entrega de contenido

Universidad de las Ciencias Informáticas

---

Se utiliza esta plataforma porque añade soporte completo para Enterprise JavaBean, Java Servlet API, Java Server Pages y la tecnología XML, además de proporcionar APIs para servicios Web, permitiendo así el desarrollo de aplicaciones web.

### 1.7.4.1 Lenguaje java

Java es un lenguaje orientado a objetos. Los objetos agrupan en estructuras encapsuladas tanto sus datos como los métodos (o funciones) que manipulan esos datos, es un lenguaje compilado e interpretado. Es un lenguaje robusto pues fue diseñado para crear software altamente fiable. Para ello proporciona numerosas comprobaciones en compilación y en tiempo de ejecución. Es utilizado para crear aplicaciones multiplataforma.

Java es utilizado por ser un lenguaje libre, multiplataforma, además el equipo de desarrollo posee experiencia en su empleo. Además brinda la posibilidad de escribir una vez el programa y poder ejecutarlo en cualquier tipo de plataforma que posea la máquina virtual de java sin tener que recompilarlo.

### 1.7.5 Entorno de desarrollo

Un entorno de desarrollo (en inglés Integrated Development Environment **IDE**) es un programa informático compuesto por un conjunto de herramientas de programación. Puede ser utilizado para un solo lenguaje o para otros lenguajes ya existentes. (23)

#### 1.7.5.1 Eclipse v3.5

Entorno de desarrollo integrado de código abierto, multiplataforma, con arquitectura abierta que permite la instalación de módulos la cual posibilita extenderse a otros lenguajes, no precisamente java. Eclipse Galileo dispone de un editor de texto con resaltado de sintaxis, la compilación es en tiempo real. Mediante este IDE se pueden realizar pruebas unitarias con JUnit, posee asistentes para creación de proyectos, clases, tests, entre otros. Mediante "Plugins" libremente disponibles es posible añadir control de versiones con Subversión. (24)

Se escogió Eclipse v3.5 para el desarrollo de la aplicación ya que este IDE posee gran cantidad de plugins los cuales agilizan el trabajo, presenta un navegador de repositorios el cual es integración con el control de versiones(svn) Subversión, además de la experiencia del equipo de trabajo en el mismo.

#### 1.7.6 Frameworks o marco de trabajo

Estos incluyen bibliotecas, librerías, entre otras herramientas, para así ayudar a desarrollar y unir los diferentes componentes de un proyecto. Estos permiten la reutilización de componentes y un rápido desarrollo del software. (25)

La utilización de un Framework en la aplicación reduciría el tiempo de desarrollo pues se dedicaría menos tiempo a escribir código de bajo nivel para alcanzar las configuraciones necesarias. También permite la reutilización de componentes, guiándose a la aplicación hacia una política de diseño bien definida.

Uno de los Frameworks utilizados para el desarrollo en java es Spring Framework, debido a su fácil integración con J2EE, por lo que se decide utilizarlo para la realización de la aplicación que será producto de la actual investigación.

##### 1.7.6.1 EL marco de trabajo Spring (Spring Framework 2.5)

El marco de trabajo (Framework en inglés) Spring (también conocido simplemente como Spring) es un Framework de código abierto que puede ser utilizado para el desarrollo de aplicaciones Java con gran calidad y rapidez.

Spring Framework se ha popularizado en la comunidad de programadores en Java al considerársele una alternativa, y sustituto del modelo de Enterprise JavaBean. Posee una amplia documentación de sus soluciones, proporcionando una programación coherente y configuraciones bien entendidas. Mientras que las características fundamentales de este Framework pueden emplearse en cualquier aplicación hecha en Java, existen muchas extensiones y mejoras para construir aplicaciones basadas en Web por encima de la plataforma empresarial de Java (Java Enterprise Platform). (26)

Se utilizan para el desarrollo de la aplicación los módulos de Spring de Seguridad, Servicios Web y Modelo Vista Controlador.

##### 1.7.6.2 Seguridad de Spring (Spring Security).

Este es un poderoso y muy personalizable Framework en temas de autenticación y control de acceso, es el estándar para crear aplicaciones seguras basadas en Spring. Es fácil de aprender, provee un gran número de directrices para operaciones



# Capítulo 1 “Fundamentación Teórica”

## Portal WAP para la plataforma de entrega de contenido

Universidad de las Ciencias Informáticas

---

comunes lo que permite añadirle seguridad a las aplicaciones en tan solo unas pocas líneas de XML, proporcionando una solución de seguridad integral, gestionando la autenticación y autorización. Su seguridad es basada en roles y lista de control de acceso. Gestiona la seguridad a varios niveles, como URLs que se solicitan al servidor, acceso a métodos, a clases javas y otros. (27)

Su utilización permite aplicarle seguridad a la aplicación al crear un sistema de control de acceso basado en roles, permite añadirle seguridad a las URLs que se desean acceder entre otros métodos.

### 1.7.6.3 Servicios Web de Spring (Spring Web Services)

Spring Web Service es un producto de Spring Framework para facilitar la creación de servicios web basados en el intercambio de documentos. Se basa en servicios en los cuales primero se establece un contrato y luego se implementan, evitando atar al contrato como sucede en los casos en los cuales se genera el mismo a partir de las clases java.

Las principales características de Spring-WS:

- Facilita aplicar las mejores prácticas para la creación de servicios web
- Facilidad para distribuir los pedidos XML a través de diferentes tipos de mapeos
- Integración con Spring Framework. (28)

La utilización de esta herramienta permite obtener los servicios de valor agregado con los que debe contar la aplicación como son las promociones, las noticias, el estado del tiempo y la consulta a las páginas amarillas, servicios que se brindan a través de servicios web publicados en el servidor.

### 1.7.6.4 Modelo Vista Controlador de Spring (Spring MVC)

Spring MVC es uno de los módulos del Framework de Spring, y como su propio nombre indica, implementa una arquitectura Modelo – Vista – Controlador que se utilizará como base para desarrollar la aplicación. (29) Su utilización permite separar todo el modelo de las vistas con solo una pequeña configuración XML posibilitando la creación de diferentes portales empleando el mismo negocio. De esta forma se

facilita la creación de varias vistas que serán mostradas en los dispositivos de acuerdo a sus características.

#### **1.7.7 Herramienta de control de versiones**

Una versión, revisión o edición de un producto, es el estado en el que se encuentra dicho producto en un momento dado de su desarrollo o modificación. 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 (por ejemplo, para algún cliente específico).

El control de versiones se realiza principalmente en la industria informática para controlar las distintas versiones del código fuente. Sin embargo, los mismos conceptos son aplicables a otros ámbitos como documentos, imágenes, sitios web, etcétera.

Aunque un sistema de control de versiones puede realizarse de forma manual, es muy aconsejable disponer de herramientas que faciliten esta gestión. Durante toda la etapa de desarrollo de la aplicación se utilizará Subversión como sistema de control de versiones.

##### **1.7.7.1 SVN (Subversión)**

Subversión es un sistema de control de versiones diseñado específicamente para reemplazar al popular CVS. Es software libre y se le conoce también como SVN por ser el nombre de la herramienta utilizada en la línea de órdenes.

Este puede acceder al repositorio a través de redes, lo que le permite ser usado por personas que se encuentran en distintas computadoras. A cierto nivel, la posibilidad de que varias personas puedan modificar y administrar el mismo conjunto de datos desde sus respectivas ubicaciones fomenta la colaboración. Se puede progresar más rápidamente sin un único conducto por el cual deban pasar todas las modificaciones. Y puesto que el trabajo se encuentra bajo el control de versiones, no hay razón para temer por que la calidad del mismo vaya a verse afectada, si se ha hecho un cambio incorrecto a los datos, simplemente deshace dicho cambio. (30)

#### 1.7.7.2 Cliente SVN utilizado v 1.5.3

TortoiseSVN es un cliente intuitivo y fácil de usar, que proporciona un acceso rápido y visual a todas las funciones de Subversión. Puede ser usado sin un entorno de desarrollo y pequeñas imágenes muestran qué archivos o directorios necesitan ser enviados al repositorio. Es utilizado para la sincronización de la aplicación con el sistema de control de versiones. (31)

#### 1.7.8 Servidor WEB

Un servidor Web puede definirse como un programa que escucha las peticiones de los clientes (usuarios o navegantes), realizadas a través de un navegador web y las atiende o satisface, proporcionando los recursos que soliciten usando el protocolo HTTP o el protocolo HTTPS (la versión cifrada y autenticada).

##### 1.7.8.1 Servidor Apache Tomcat 6.0

Tomcat es el servidor Web más utilizado a la hora de trabajar con Java en entornos web; Tomcat es una implementación completamente funcional de los estándares de JSP y Servlets. Tomcat está integrado en la implementación de referencia Java 2 Enterprise Edition (J2EE) de Sun Microsystems. Funciona como un contenedor de servlets desarrollado en la Apache Software Foundation (Fundación de Software Apache). Tomcat implementa las especificaciones de los Servlets y de Java Server Pages (JSP). (32)

Es utilizado este servidor por ser de software libre, tener abundante documentación, agregando que permite compilar las páginas JSP para ser mostradas en la web, además posibilita la depuración de las aplicaciones java.

#### 1.7.9 ALEMBIK

El Servidor de Transcodificación de Multimedia ALEMBIK es una aplicación java que provee servicios de transcodificación para diferentes tipos de multimedia (video, imágenes, audio, etc.). Esta consiste en tomar un contenido existente en un formato y transformarlo en otro atendiendo a las características que se desean obtener. Para realizar la transformación se debe especificar la dirección URL del recurso y un conjunto de parámetros al que el archivo nombrado debe ser transformado.

Una vez que el servidor obtiene y valida todas las peticiones comienza la transcodificación. Al finalizar esta, retorna la dirección URL que apunta al archivo transcodificado o un error especificando las causas del mismo. (33)

Se utiliza este servidor porque brinda la posibilidad de convertir un contenido de un formato a otro, posibilitando entregar un contenido que sea soportado por el dispositivo que realice la descarga.

## 1.8 Conclusiones

Se puede concluir que:

- En el estudio realizado de algunas de las aplicaciones que presentan muchos de los servicios solicitados por ETECSA, estos no pueden ser utilizados porque pertenecen a empresas con las que debe obtenerse un acuerdo para el derecho de uso y luego realizarse las modificaciones necesarias para cumplir con los servicios solicitados, debido a que estos no responden totalmente a las especificidades del sistema.
- La metodología de desarrollo utilizada es RUP por tener características perfectamente ajustables al proyecto y el conocimiento por parte del equipo de desarrollo es mucho mayor que con otras descritas.
- La herramienta CASE utilizada para el modelado del sistema es Visual Paradigm para UML 8.0 que permite crear los modelos gráficos de aspecto profesional.
- El lenguaje de modelado utilizado es el estándar UML para la realización de los diagramas de casos de uso, componentes, secuencia, entre otros.
- El lenguaje de programación utilizado es Java para la realización de la implementación de la solución propuesta.
- El IDE de desarrollo utilizado es Eclipse 3.5 para la implementación de la aplicación porque posibilita la compilación del lenguaje java, y a través de módulos instalados es posible sincronizarse con el control de versiones.
- El marco de trabajo utilizado es Spring, el cual brinda una fuerte implementación del patrón arquitectónico **Modelo-Vista-Controlador**, lo que posibilita la separación de las diferentes capas de los componentes que se pretenden construir, permitiendo una mayor organización en el resultado final. Además, el empleo de este marco de trabajo facilita la interoperabilidad

## *Capítulo 1 “Fundamentación Teórica”*

### *Portal WAP para la plataforma de entrega de contenido*

*Universidad de las Ciencias Informáticas*

---

entre componentes, debido a que posibilita el consumo de servicios entre ellos. El empleo del módulo seguridad de Spring que posibilita crear aplicaciones seguras, aplicando reglas a las URLs que se desean acceder entre otras restricciones, además te permite crear aplicaciones con control de acceso basados en roles.

- El servidor Web utilizado es Apache Tomcat 6.0 que permite compilar las páginas JSP y además posibilita la depuración de código.
- Las herramientas para el control de versiones utilizadas son SVN del lado del servidor y el cliente TortoiseSVN por su fácil sincronización con el sistema de control de versiones y facilidad de uso.
- Los tres métodos para la descarga de contenidos mencionados en el epígrafe correspondiente, serán utilizados porque posibilitan que la mayor cantidad de dispositivos móviles puedan descargar contenidos.

## **Capítulo 2 “Características del sistema”**

### **2.1 Introducción**

En el presente capítulo se muestra el modelo de dominio obtenido y se especifican los requisitos del software a construir. Se presenta además el diagrama de casos de uso del sistema (DCUS) y las descripciones de dos de ellos. Para el desarrollo de todo lo antes mencionado se tiene como guía la metodología de desarrollo de software RUP.

### **2.2 Objeto de automatización**

El desarrollo de un portal para dispositivos móviles posibilita el acceso a los servicios de valor agregado y contenidos a los usuarios de la telefonía móvil en Cuba. Para ello se pretende informatizar la entrega de contenidos multimedia, adaptándolos al dispositivo que realiza la petición y garantizando la aplicación de los derechos digitales. Además se debe permitir la generación de un archivo para la facturación y se mantendrá la actualización de las noticias, el estado del tiempo y las promociones publicitarias de disímiles proveedores.

### **2.3 Propuesta del sistema**

Se propone una aplicación que entre sus funcionalidades permita brindar servicios de valor agregado tales como promociones, estado del tiempo, noticias, directorio telefónico. La aplicación además permitirá la descarga de contenidos multimedia desde cualquier dispositivo móvil. Brindará la interfaz que cada dispositivo móvil soporte según las propiedades físicas del mismo y brindará los servicios de manera amigable al usuario, para que su navegación sea sencilla y agradable.

El usuario se debe registrar en la aplicación para poder realizar la descarga de los contenidos como videos, música, fondos, aplicaciones, timbres y juegos ofertados por diversos proveedores a través del portal los cuales se encuentran almacenados en una base de datos. Los servicios de valor agregado se obtendrán a través de servicios web, los cuales son brindados por proveedores de forma remota. De esta manera se mantendrán actualizados las noticias, el estado del tiempo y las

## Capítulo 2: Características del Sistema

### Portal WAP para la plataforma de entrega de contenido

Universidad de Las Ciencias Informáticas

promociones mostradas en el portal. El portal contará con dos interfaces posibilitando que pueda ser visualizado en cualquier dispositivo que cuente con el protocolo WAP. Será utilizado el Framework de Spring para unir todas las partes que se crearán, y permitir la abstracción de todas las configuraciones de bajo nivel.

Para una mejor comprensión se ilustra la propuesta del sistema en la figura 5.

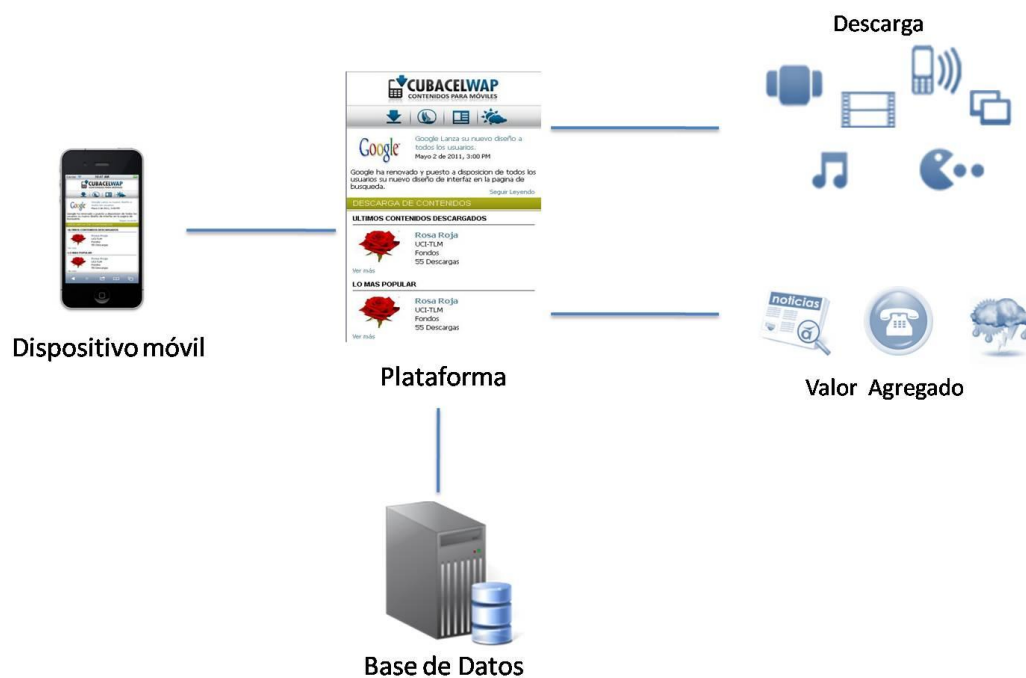


Figura 5: Propuesta del sistema

## 2.4 Modelo de dominio

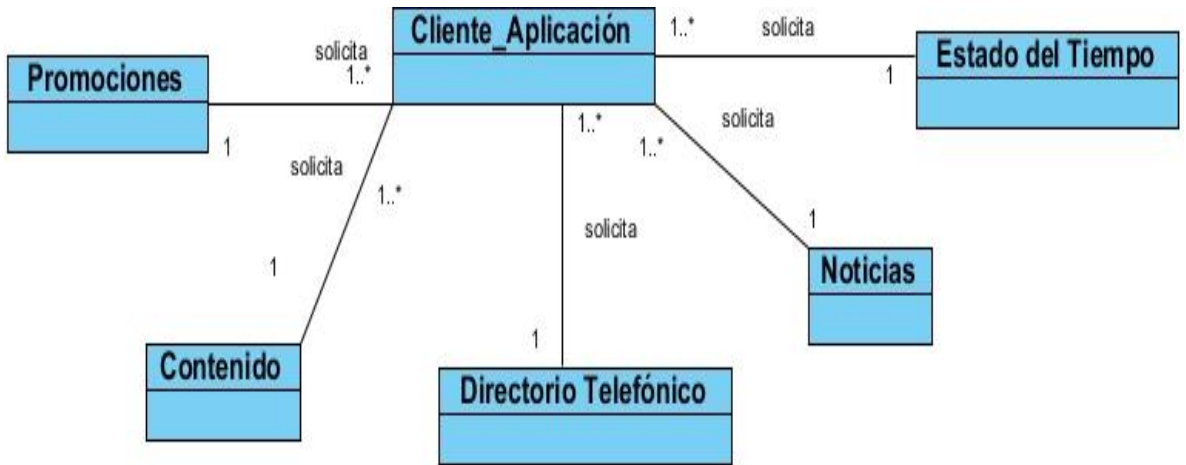
El modelo de dominio ayuda a comprender los conceptos que utilizan los usuarios, los conceptos con los que trabajan y con los que deberá trabajar la aplicación. Es un artefacto de la disciplina de análisis, construido con las reglas de UML durante la fase de concepción, en la tarea construcción del modelo de dominio, presentado como uno o más diagramas de clases y que contiene, no conceptos propios de un sistema de software sino de la propia realidad física (34). Es un diagrama con los objetos reales existentes relacionados con el Sistema que se desea desarrollar y las relaciones que existen entre ellos.

## Capítulo 2: Características del Sistema

### Portal WAP para la plataforma de entrega de contenido

Universidad de Las Ciencias Informáticas

#### 2.4.1 Diagrama de clases del dominio

Se procedió a seccionar el modelo de dominio para una mayor claridad y comprensión en el documento, dichas secciones estarán representadas por un elemento que mostrará las principales clases del modelo de dominio, luego se mostrarán los modelos de dominio de Contenido y Agregado representados en las figuras 6, 7 y 8 respectivamente. El modelo de dominio en su generalidad se muestra en el . ¡Error! No se encuentra el origen de la referencia..

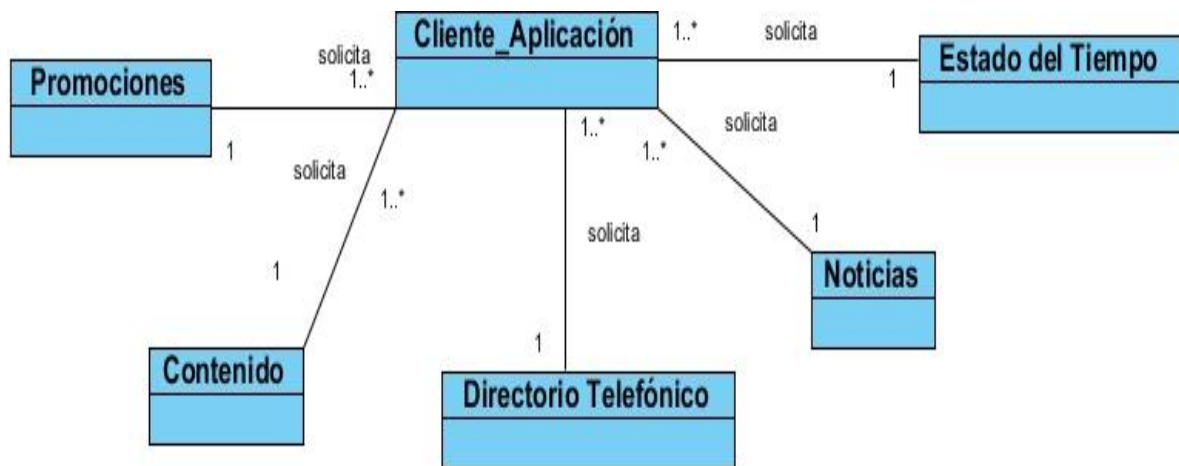


Figure 6: Modelo de dominio



## Capítulo 2: Características del Sistema

### Portal WAP para la plataforma de entrega de contenido

Universidad de Las Ciencias Informáticas

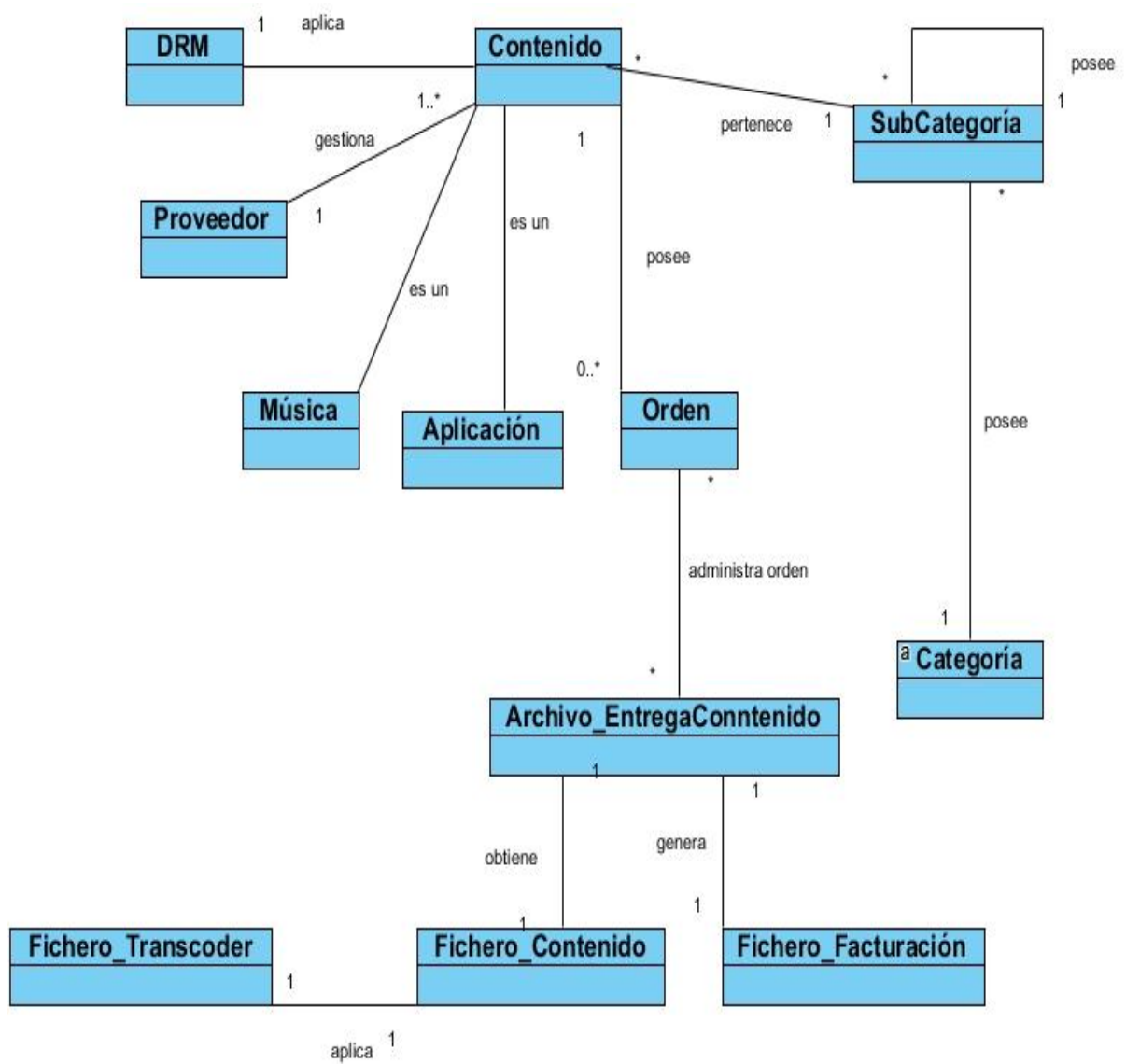


Figure 7: Modelo de dominio Contenido

## Capítulo 2: Características del Sistema

Portal WAP para la plataforma de entrega de contenido

Universidad de Las Ciencias Informáticas

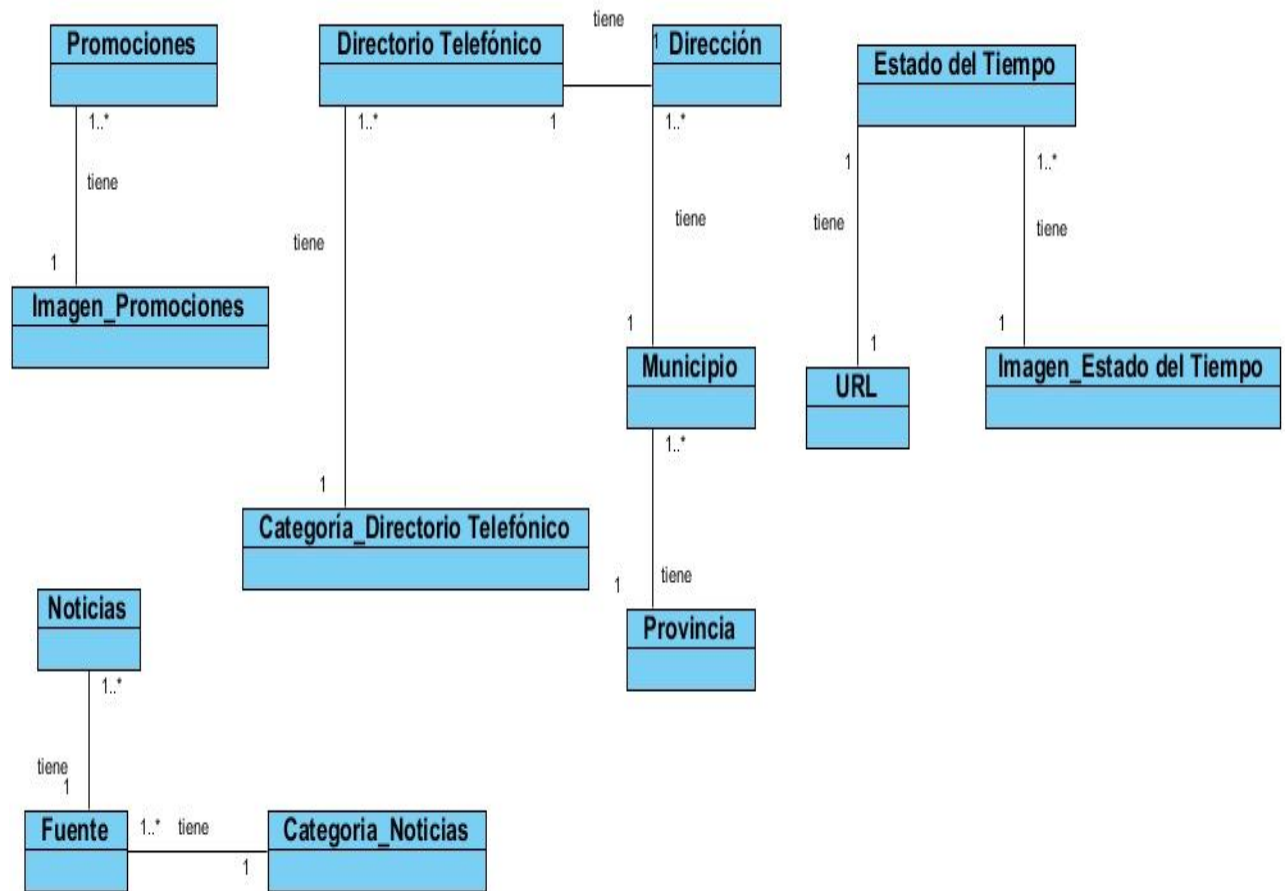


Figure 8: Modelo de dominio Valor Agregado

### 2.4.2 Conceptos del modelo de dominio

- ✓ Cliente\_Aplicación: Solicita los servicios que ofrece la aplicación.
- ✓ Promociones: Representa los datos referentes a las Promociones.
- ✓ Estado del Tiempo: Representa a los datos referentes al Estado del Tiempo.
- ✓ Noticias: Representa a los datos referentes a las Noticias.
- ✓ Directorio Telefónico: Representa los datos referentes al Directorio Telefónico de Páginas Amarillas.
- ✓ Contenido: Representa los Contenidos que se gestionan.
- ✓ Imagen\_Promociones: Representa la imagen relacionada con una Promoción determinada.
- ✓ Imagen\_Estado del Tiempo: Representa la imagen relacionada con un Estado del Tiempo determinado.

## *Capítulo 2: Características del Sistema*

### *Portal WAP para la plataforma de entrega de contenido*

*Universidad de Las Ciencias Informáticas*

---

- ✓ URL: Representa la URL Fuente que proporciona el Estado del Tiempo de un Canal RSS.
- ✓ Fuente: Representa los datos de una Fuente de Noticias de un Canal RSS.
- ✓ Categoría\_Noticias: Representa la Categoría de una Noticia.
- ✓ Categoría\_Directorio Telefónico: Representa la Categoría de una Entrada Telefónica.
- ✓ Dirección: Representa la Dirección de una Entrada Telefónica determinada.
- ✓ Municipio: Representa el Municipio de la Dirección de una Entrada Telefónica determinada.
- ✓ Provincia: Representa la Provincia de la Dirección de una Entrada Telefónica determinada.
- ✓ DRM: Seguridad que se aplica a los contenidos para conservar el derecho de autor.
- ✓ Proveedor: Representa la entidad que se encarga de proveer los contenidos.
- ✓ Música: Representa los contenidos de tipo música.
- ✓ Aplicación: Representa los contenidos de tipo aplicaciones.
- ✓ Orden: Representa una petición para adquirir un contenido.
- ✓ Archivo\_EntregaConntenido: Encargado de gestionar el proceso de entregar el contenido al cliente.
- ✓ Fichero\_Contenido: Módulo que contiene la base de datos de los contenidos que el cliente desea descargar como tonos, logos y melodías.
- ✓ Fichero\_Transcoder: Representa el subsistema de modificación de formato del contenido.
- ✓ Fichero\_Facturación: Módulo que efectúa el cobro por la descarga de los contenidos.
- ✓ Categoría: Representa la categoría referente a una Sub-Categoría.
- ✓ SubCategoría: Representa las diferentes Sub-Categorías de un contenido determinado.

## Capítulo 2: Características del Sistema

### Portal WAP para la plataforma de entrega de contenido

Universidad de Las Ciencias Informáticas

## 2.5 Especificación de los requisitos del sistema

### 2.5.1 Requisitos funcionales

Los requisitos funcionales establecen los comportamientos de la aplicación así como los servicios que debe proporcionar el mismo.

Cantidad de requisitos funcionales	Prioridades		
	Alta	Media	Baja
19	4 RF9, RF10, RF11, RF12	4 RF5,RF1,RF6 ,RF16,	11 RF4, RF2,RF7, RF8, RF3,RF13,RF14 ,RF15,RF17,RF 18,RF19

#### RF1. Registrar usuario

Se permite a un visitante insertar su información de registro:

- Nombre
- Apellidos
- Usuario
- Contraseña
- Confirmación de la contraseña

#### RF2. Des-autenticar usuario

Se permite a un usuario autenticado en el portal des autenticarse.

#### RF3. Descargar contenido

Se permite a través del portal WAP realizar la descarga del contenido seleccionado.

## *Capítulo 2: Características del Sistema*

### *Portal WAP para la plataforma de entrega de contenido*

*Universidad de Las Ciencias Informáticas*

---

#### **RF4. Autenticar usuario**

Se permite a los usuarios registrados del portal autenticarse insertando:

- Usuario
- Contraseña

#### **RF5. Modificar perfil de usuario**

Se permite a un usuario modificar cualquier información insertada durante su registro y otros datos.

- Nombre(s)
- Apellido
- Correo electrónico
- Cambiar contraseña
- Repetir contraseña
- Número de fax

#### **RF6. Listar contenido más descargado por categoría.**

Se muestra el listado de contenidos correspondiente a la categoría seleccionada.

#### **RF7. Listar los últimos contenidos incorporados por categoría.**

Se muestra el listado de los últimos contenidos incorporados correspondientes a la categoría seleccionada.

#### **RF8. Listar contenidos disponibles por subcategoría.**

Se muestra el listado de contenidos disponibles correspondiente a la sub – categoría seleccionada.

#### **RF9. Entregar contenido al cliente.**

Se permite al cliente obtener el contenido.

#### **RF10. Proteger contenido solicitado.**

Se permite proteger el contenido solicitado mediante el uso del algoritmo Forward Lock.

## *Capítulo 2: Características del Sistema*

### *Portal WAP para la plataforma de entrega de contenido*

*Universidad de Las Ciencias Informáticas*

---

#### **RF11. Transformar el contenido en correspondencia con el móvil que está realizando la petición.**

Se permite realizar el proceso de transformación de contenido para que pueda ser accedido por cualquier tipo de dispositivo móvil.

#### **RF12. Generar el archivo CDR.**

Se permite generar el archivo CDR para el posterior cobro de la descarga luego de terminada.

#### **RF13. Mostrar estado del tiempo.**

Se permite mostrar un resumen del estado de tiempo.

#### **RF14. Listar noticias.**

Se permite mostrar un listado con los títulos de las noticias más recientes disponibles en la plataforma.

#### **RF15. Listar promociones.**

Se permite mostrar el listado de promociones disponibles.

#### **RF16. Buscar directorio telefónico.**

Se permite buscar en el directorio telefónico por:

- Entidad
- Categoría
- Provincia
- Municipio

#### **RF17. Mostrar detalles del estado del tiempo.**

Se permite mostrar con detalles la información referente al estado de tiempo.

#### **RF18. Mostrar resumen de la noticia por ID.**

Se permite mostrar información resumida de una noticia según el identificador de la misma.

#### **RF19. Mostrar fuente de la noticia.**

## *Capítulo 2: Características del Sistema*

### *Portal WAP para la plataforma de entrega de contenido*

*Universidad de Las Ciencias Informáticas*

---

Se permite ver la fuente original donde está publicada la noticia dada su URL.

#### **2.5.2 Requisitos no funcionales**

Estos son propiedades o cualidades que la aplicación debe cumplir. A continuación se describen los requisitos no funcionales teniendo en cuenta su categoría.

##### **2.5.2.1 Soporte**

La UCI debe brindar soporte tanto a administradores como a clientes que necesiten un breve entrenamiento para operar la aplicación, en dependencia de las demandas del comprador, puede ser mantenimiento, entrenamiento de personal, integración con otro software u operar directamente la aplicación.

##### **2.5.2.2 Usabilidad**

###### **Formato de fecha y hora utilizadas en la aplicación**

En la interfaz visual se utilizará un formato de fecha y de hora, para efectos de almacenamiento se utilizará el formato estándar según el SGBD. El formato disponible para la interfaz visual es:

MMM Dd de yyyy, hh:mm am/pm Ej.: Noviembre 3 de 2011, 04:30 pm.

###### **Mensajes de textos en la interfaz**

Tanto la interfaz, como los mensajes para interactuar con los usuarios, así como los mensajes de error, deberán ser en idioma castellano y tener una apariencia estándar. Los mensajes de error deberán ser lo suficientemente informativos para dar a conocer la severidad del error.

##### **2.5.2.3 Software**

Para el funcionamiento de la aplicación se debe utilizar la máquina virtual de Java versión 1.6.0\_20 o superior.

Contenedor Web que soporte la especificación JSP, Apache Tomcat 6.0.

Distribución Debian del sistema operativo Linux para el servidor de aplicaciones web.

## Capítulo 2: Características del Sistema

### Portal WAP para la plataforma de entrega de contenido

Universidad de Las Ciencias Informáticas

---

#### 2.5.2.4 Hardware

La aplicación se desplegará en un servidor con 4gb de memoria RAM como requerimiento mínimo, procesador Core 2 Duo a 2.6 ghz o superior y debe contar además con un almacenamiento en disco de 500gb.

#### 2.5.2.5 Apariencia o Interfaz interna

La implementación de las funcionalidades de la aplicación debe seguir con las restricciones de codificación adoptados por la dirección del proyecto. Con esto se garantiza una fácil comprensión del código por parte de los desarrolladores para de futuras versiones.

#### 2.5.2.6 Portabilidad

La aplicación deberá permitir la descarga de contenido a los dispositivos móviles que soporten los métodos de descarga como OMA, Midlets y Descarga Directa. Existen algunos dispositivos que no soportan estos métodos de entrega de contenido, ejemplo de ello es el Iphone el cual por restricciones de su compañía solamente permite la descarga de aplicaciones del App Store.

## 2.6 Modelo de casos de uso del sistema

A continuación se muestran los casos de uso del sistema con el objetivo de describir el comportamiento de la aplicación desde la perspectiva del usuario.

### 2.6.1 Actores del sistema

Actor	Descripción
Cliente	Representa aquella persona que accede al sistema y ejecuta todas las funcionalidades de ella
Usuario	Representa aquella persona que accede al sistema y ejecuta todas las funcionalidades exceptuando la descarga de contenido y la modificación del perfil de usuario.

Tabla 1: Actores del sistema



## Capítulo 2: Características del Sistema

### Portal WAP para la plataforma de entrega de contenido

Universidad de Las Ciencias Informáticas

#### 2.6.2 Diagrama de casos de uso del sistema

Los diagramas de casos de uso muestran la funcionalidad del sistema desde la perspectiva que tienen los usuarios y lo que el sistema debe de hacer para satisfacer los requisitos propuestos. Es decir, describen un uso del sistema y cómo este interactúa con el usuario. Luego de tener un conocimiento del dominio del sistema y de los requerimientos del mismo se obtuvo el siguiente diagrama de casos de usos del sistema mostrado en la figura 9.

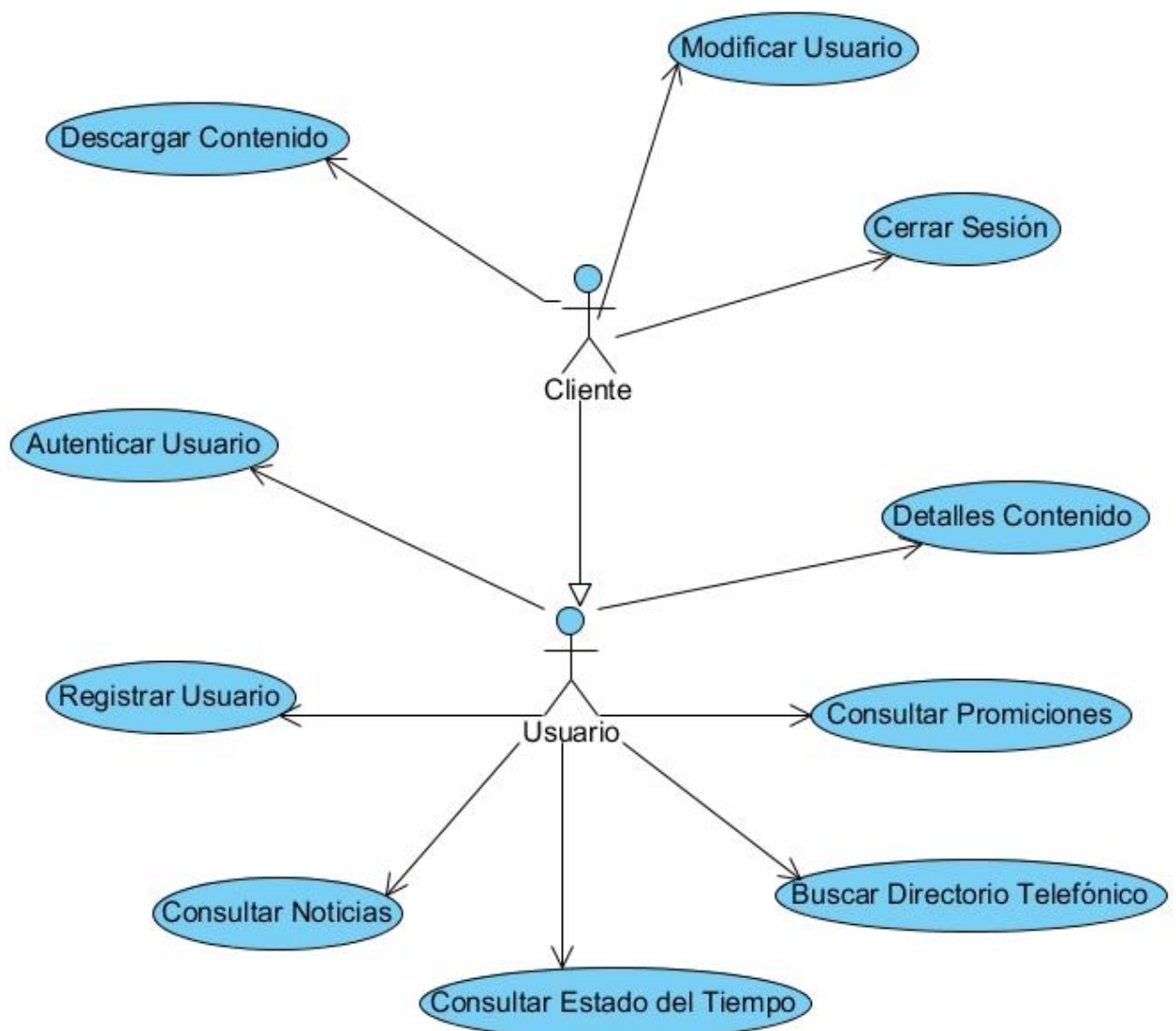


Figura 9: Diagrama de casos de uso del sistema

## *Capítulo 2: Características del Sistema*

### *Portal WAP para la plataforma de entrega de contenido*

*Universidad de Las Ciencias Informáticas*

---

#### **2.6.3 Descripción de los casos de uso**

El diagrama de casos de uso es importante por su forma gráfica de mostrar la interacción del usuario con las funcionalidades del sistema, siendo la descripción de casos de usos la que muestra de forma detallada la acción del usuario y la respuesta que debe brindar la aplicación. A continuación se describen los casos de uso arquitectónicamente más significativos como “Descargar Contenido” y “Autenticar Usuario”.

#### **2.7 Conclusiones**

Se puede concluir que:

- Utilizando RUP y aprovechando sus principales características, se generaron los artefactos necesarios, permitiéndose una mayor comprensión de lo que el sistema debe hacer.
- La identificación de las funcionalidades y características del producto, reflejadas en el modelado del sistema, donde se especifican los actores y las acciones que estos realizan, permite que se tenga una idea más clara de cómo debe comportarse el sistema desde el punto de vista del usuario.

## **Capítulo 3: Diseño del sistema**

### **3.1 Introducción**

En el presente capítulo se tratan los elementos relacionados con el diseño de la aplicación. Se especifican los diagramas de clases del diseño y secuencia. Se muestra la concepción de la arquitectura y algunos rasgos esenciales en el funcionamiento del marco de trabajo, con el objetivo de brindar una mayor comprensión de los componentes implementados. Se realiza una descripción del estándar de codificación, los patrones empleados durante el desarrollo y una muestra de las clases con sus operaciones.

### **3.2 Descripción de la arquitectura**

La arquitectura Cliente – Servidor está compuesta por dos partes fundamentales, la primera es el servidor y la segunda el conjunto de clientes. El servidor es un programa que recibe una solicitud y devuelve una respuesta, generalmente un servidor puede atender múltiples peticiones al mismo tiempo. El cliente son las estaciones de trabajo que solicita dichos servicios. Ambas deben estar conectadas entre sí mediante una red. (35) En este modelo las aplicaciones se dividen de forma que en el servidor se encuentra la parte que debe ser compartida por varios usuarios, y en el cliente se visualizara solo lo particular de cada usuario. (36)

Para la construcción de un programa relativamente fácil de modificar y ajustar a escala en el servidor la arquitectura de software posee un estilo arquitectónico llamado Sistemas de Llamada y Retorno. El mismo contiene un patrón llamado arquitectura en capas el cual posee una organización jerárquica, donde cada capa proporciona servicios a la capa superior y actúa como cliente de la capa inferior. La utilización de este patrón significa una fácil migración debido a que el acoplamiento con el entorno se ubica en las capas inferiores siendo estas las únicas a modificar en caso de existir un cambio. Proporciona un mejor mantenimiento debido que los cambios apenas afectan las capas inferiores o superiores. Para implementar los

## *Capítulo 3: Diseño del Sistema*

### *Portal WAP para la plataforma de entrega de contenido*

*Universidad de Las Ciencias Informáticas*

---

niveles superiores no se requiere conocer el entorno subyacente, basta con las interfaces que proporcionan los niveles inferiores.

Teniendo en cuenta lo anteriormente planteado se define para la realización de la aplicación utilizar una arquitectura en capas, lo que permite fragmentar la codificación. Utilizando específicamente el patrón arquitectónico **Modelo Vista Controlador** que incluye el marco de trabajo de Spring el cual propone la utilización de 3 capas con las cuales se separan los datos de la aplicación, la interfaz de usuario y la lógica de negocio. Esto proporciona un mejor entendimiento y reusabilidad del código que será generado.

La seguridad se tiene en cuenta en todos los procesos de la aplicación. Esta es considerada el talón de Aquiles de los sistemas informáticos por tanto la aplicación pretende minimizar las amenazas de seguridad para lo cual utiliza el Framework Spring Security. Este Framework posibilita crear aplicaciones de control de acceso basado en roles (RBAC<sup>7</sup> por sus siglas en inglés).

También existe un manejo de excepciones que permite controlar los posibles errores que puedan ser lanzados durante la ejecución de la aplicación. Estos errores pueden provocar un mal funcionamiento en una operación determinada causando molestias en los usuarios. Los posibles errores serán guardados en los LOG para su posterior revisión. La figura 10 muestra de forma gráfica la arquitectura planteada.

---

<sup>7</sup> Role Based Access Control

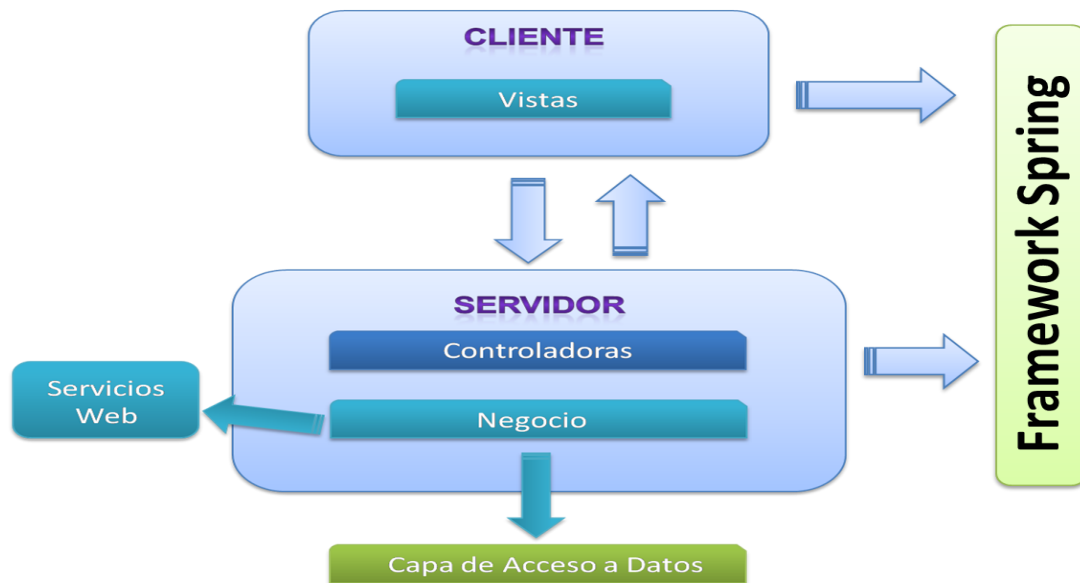


Figura 10: Arquitectura de la aplicación

### 3.3 Modelo de diseño

El modelo de diseño es una abstracción de la implementación del sistema. Se utiliza para concebir y para documentar el diseño del sistema de software. Es un producto de trabajo integral y compuesto que abarca todas las clases de diseño, subsistemas, paquetes, colaboraciones y las relaciones entre ellos. (37)

#### 3.3.1 Diagramas de clases del diseño

El diagrama de clases es el diagrama principal de diseño y análisis para un sistema. En él, la estructura de clases del sistema se especifica, con relaciones entre clases y estructuras de herencia. Durante el análisis del sistema, el diagrama se desarrolla buscando una solución ideal. Durante el diseño, se usa el mismo diagrama, y se modifica para satisfacer los detalles de las implementaciones. (38)

#### 3.3.2 Diagramas de interacción

Existen dos tipos de diagramas de interacción, ambos se basan en la misma información y muestra un patrón de interacción entre objetos, pero cada uno de ellos hace énfasis en un aspecto en particular. Estos diagramas son: el diagrama de colaboración y el diagrama de secuencia.

### 3.3.2.1 Diagramas de secuencia

El diagrama de secuencia es un tipo de diagrama usado para modelar interacción entre objetos en un sistema según UML. Un diagrama de secuencia muestra la interacción de un conjunto de objetos en una aplicación a través del tiempo y se modela para cada caso de uso.

### 3.3.3 Patrones utilizados

Patrón es una pareja de problema/solución con un nombre y que es aplicable a otros contextos, con una sugerencia sobre la manera de usarlo en situaciones nuevas.

(37)

#### 3.3.3.1 Patrones de diseño

Los patrones de diseño son el esqueleto de las soluciones a problemas comunes en el desarrollo de software. Estos brindan una solución ya probada y documentada a problemas de desarrollo de software. Se debe tener presente los siguientes elementos de un patrón: su nombre, el problema (cuando aplicar un patrón), la solución (descripción abstracta del problema) y las consecuencias (costos y beneficios). (39)

- **Patrones GRASP<sup>8</sup>**

Los patrones GRASP describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones.

**Experto:** Consiste en asignar la responsabilidad a la clase que cuenta con la información necesaria para realizar determinada tarea, es decir el experto en información. Así se garantiza el que la aplicación sea más fácil de entender. Con la utilización de este patrón los objetos se valen de su propia información por lo que se conserva el encapsulamiento. Soporta un bajo acoplamiento permitiendo tener sistemas robustos y de fácil mantenimiento. Este patrón se ve reflejado en las clases del negocio que cada una tiene su función específica.

---

<sup>8</sup> Patrones Generales de Software para Asignar Responsabilidades.

**Creador:** Consiste en asignarle la responsabilidad de crear una instancia de una clase a aquella que agrega o contiene los objetos de la que se desea crear, registra las instancias de la misma, utiliza los objetos de ella, tiene los objetos de inicialización de ella cuando este objeto sea creado. Este patrón guía la asignación de responsabilidades relacionadas con la creación de objetos. Se brinda soporte a un bajo acoplamiento lo cual supone menos dependencias respecto al mantenimiento y mejores oportunidades de reutilización. Este patrón se evidencia en la clase WapFacadeImpl desde la cual se utiliza las implementaciones de las clases manager. La figura 13 se observa la utilización del patrón en cuestión.

```
/**  
package cu.uci.comcel.portal.wap.advance.facade.impl;  
  
import java.io.IOException;  
  
public class WapFacadeImpl implements IWapFacade {  
  
    private INewsManager newsManager;  
    private IPromotionManager promotionManager;  
    private IPhoneDirectoryManager phoneDirectoryManager;  
    private IWeatherManager weatherManager;  
    private IDownloadManager downloadsManager;  
  
    public void setNewsAdmin(NewsManagerImpl newsAdmin) {  
        this.newsManager = newsAdmin;  
    }  
  
    public void setPromotionAdmin(PromotionManagerImpl promotionAdmin) {  
        this.promotionManager = promotionAdmin;  
    }  
  
    public void setPhoneDirectoryAdmin(PhoneDirectoryManagerImpl phoneDirectoryAdmin) {  
        this.phoneDirectoryManager = phoneDirectoryAdmin;  
    }  
}
```

Figura 10: Ejemplo de utilización del patrón creador

**Bajo acoplamiento:** El acoplamiento es una medida de la fuerza con que una clase está conectada a otras, con que las conoce y con que recurre a ellas. Una clase con alto (o fuerte) acoplamiento recurre a muchas otras. Esto provoca que sean más difíciles de entender cuando están aisladas y son más difíciles de reutilizar porque se requiere la presencia de otras clases de las que dependen. Spring plantea el uso de interfaces para evitar un alto acoplamiento, por lo que en aplicación este patrón

## *Capítulo 3: Diseño del Sistema*

### *Portal WAP para la plataforma de entrega de contenido*

*Universidad de Las Ciencias Informáticas*

---

se ve claramente reflejado, ya que tanto la capa de negocio como la capa de acceso a datos tienen interfaces que son implementadas por otras clases.

**Alta cohesión:** La cohesión es una medida de cuán relacionadas y enfocadas están las responsabilidades de una clase. Una alta cohesión caracteriza a las clases con responsabilidades estrechamente relacionadas que no realicen un trabajo enorme. Una clase con baja cohesión las hace difíciles de comprender, conservar y reutilizar, además de que las afectan constantemente los cambios. Es un principio que debe estar presente durante el diseño, es un patrón evaluativo que se aplica al juzgar decisiones de diseño. Anteriormente se representó el uso del patrón experto, el cual distribuye el comportamiento entre las clases que cuentan con la información, alentando con ello definiciones de clases sencillas y más cohesivas que son más fáciles de comprender y mantener. Así se brinda soporte a la Alta Cohesión.

**Controlador:** Un controlador es un objeto de interfaz no destinada al usuario que se encarga de manejar un evento del sistema. Define además el método de su operación. El hecho de delegar a un controlador la responsabilidad de la operación de un sistema entre las clases del dominio soporta la reutilización de la lógica para manejar los procesos afines del negocio en aplicaciones futuras. Este patrón es utilizado mediante el framework Spring por lo que cada una de las vistas de la aplicación tendrá un controlador para manejar sus eventos. En la figura 14 se muestra un ejemplo del patrón controlador.



```
package cu.uci.comcel.portal.wap.advance.controller;

import java.util.ArrayList;

public class DownloadsControllerAdvance extends AbstractController {
    private static final Logger logger = Logger.getLogger(DownloadsControllerAdvance.class);

    @Override
    public ModelAndView handleRequestMethod(HttpServletRequest req,
        HttpServletResponse reps) throws Exception {
```

Figura 11: Ejemplo de patrón controlador

- **Patrones GOF<sup>9</sup>**

**Fachada:** Proporciona una interfaz unificada de alto nivel que agrupa las funcionalidades de un subsistema facilitando su uso. Simplifica el acceso a un conjunto de clases debido que el cliente se comunica con ellas a través de una sola interfaz. La fachada sabe que clase es responsable de cada petición, pero es completamente transparente para ellas. La utilización de este patrón se puede apreciar en el paquete Facade donde se encuentra contenida la fachada relacionada con la aplicación. Esto garantiza independencia y portabilidad además de permitir la estructuración en capas de la aplicación.

**Inversión de control (IoC)/ Inyección de dependencias:** La inversión de control es un principio que permite asignar la responsabilidad del control en la creación de objetos a un agente externo como un contenedor, en lugar de hacerlo las clases de la aplicación. Spring proporciona un contenedor que maneja el ciclo de vida de los objetos y las dependencias entre ellos. Mientras que la inversión de control es un principio de diseño general, la inyección de dependencias es un patrón de diseño concreto que rige este principio. El patrón inyección de dependencias consiste en

---

<sup>9</sup> Gang of Four por sus siglas en inglés (Pandilla de Cuatro).

inyectar objetos a una clase en vez de hacerlo la misma clase. Este patrón se implementa mediante un contenedor que inyecta a cada objeto los objetos necesarios según las relaciones plasmadas en un fichero de configuración y típicamente es implementado por un Framework externo a la aplicación como Spring. Se utiliza este patrón en la aplicación mediante el Framework Spring para inyectar las dependencias a las clases managers y facade. En la figura 15 se muestra un ejemplo de la utilización de este patrón.

```
<bean name="baseController2"
      class="cu.uci.comcel.portal.wap.advance.controller.AbstractController"
      abstract="true">
  <property name="greeting">

</property>
  </property>
  <property name="transcoding" ref="transcodingBean"></property>
  <property name="managerFacade" ref="wapManagerFacade2"></property>
  <property name="basePath" value="${base.path}"></property>
  <property name="backAdmin" ref="backAdmin"></property>
  <property name="wurflManager" ref="wurflManager"></property>
  <property name="fileUtil" ref="fileUtil"></property>
  <property name="wapUserFacade" ref="userFacade"></property>
</bean>

<bean name="wapManagerFacade2"
      class="cu.uci.comcel.portal.wap.advance.facade.impl.WapFacadeImpl">
  <property name="newsAdmin" ref="newsAdmin2" />
  <property name="promotionAdmin" ref="promotionAdmin2" />
  <property name="phoneDirectoryAdmin" ref="phoneDirectoryAdmin2" />
  <property name="weatherAdmin" ref="weatherAdmin2" />
  <property name="downloadsAdmin" ref="downloadsAdmin2" />
</bean>
```

Figura 12: Ejemplo de patrón inyección de dependencias

### 3.4 Estándares de nomenclatura y codificación utilizados

Los estándares de codificación son un conjunto de directrices, normas y reglamentos enfocados a la especificación de cómo debe escribirse el código fuente de la aplicación. Estos incluyen pautas sobre la nomenclatura de las variables, clases y paquetes; la correcta indentación del código, cómo escribir estructuras de control, entre otros aspectos.

La correcta elaboración y utilización de los estándares de codificación permiten que todos los desarrolladores implementen siguiendo las mismas pautas y así puedan entender el código del resto como si estuvieran mirando el de ellos, así la aplicación será más legible, uniforme y fácil de mantener.

### Nomenclatura general

- El nombre de las clases, métodos y variables deben ser estar escritos en inglés.
- En todo momento se utilizarán nombres que sean claros, concretos y libres de ambigüedades. Ejemplo: "lastContent" y no solamente "content",
- El nombre de todas las variables y métodos comenzarán con letra minúscula y si este está compuesto por varias palabras se utilizará el estilo de escritura lowerCamelCase, que dicta que para un nombre compuesto por varias palabras comenzará con minúscula pero todas las palabras internas que lo componen comienzan con mayúscula.

### Identación

- En el contenido siempre se indentará este con tabulaciones, nunca utilizando espacios en blanco.

### Clases

- El nombre de las clases controladoras estará en correspondencia con la funcionalidad que represente y seguido de la palabra "Controller".
- El nombre de las clases comenzará con mayúsculas y cada salto de palabras debe iniciar con mayúsculas como se muestra en la figura 19

```
public class PhoneDirectoryManagerImpl implements IPhoneDirectoryManager {
```

Figura 13: Ejemplo de nombre de clase

### Rutinas y métodos

- Un buen nombre para una función o método es aquel que describe todo lo que la rutina hace.

### Nombre de variables

- El nombre de las variables debe empezar con letra minúsculas y de existir un cambio de palabra comenzaría con mayúscula.
- El nombre de las variables debe ser sugerente al valor que se refiere.

## 3.5 Descripción de las clases

---

Clases	Descripción
--------	-------------

---

<b>IWapFacade</b>	Contiene los métodos necesarios para obtener los datos solicitados por el usuario.
<b>DownloadsController</b>	Es la encargada de solicitar los datos específicos para la creación de la vista downloads la cual mostrará los datos existentes.
<b>ContentController</b>	Es la clase encargada de buscar el contenido seleccionado por el usuario y mostrar los detalles del mismo en la vista content.
<b>InitDownloadController</b>	Es la clase encargada de seleccionar el método de descarga y en dependencia del método seleccionado muestra la vista correspondiente a este método.

---

### 3.6 Conclusiones

Se puede concluir que:

- El diseño propuesto posibilitó la implementación satisfactoria de la aplicación, ya que tuvo en cuenta las características del marco de trabajo empleado, que permite a través del módulo MVC una mayor organización, mantenimiento y facilita la interoperabilidad entre los componentes de la aplicación.
- Los patrones de diseño permitieron dar solución a problemas comunes en la implementación, algunos de los patrones utilizados fueron el Controlador, Fachada, Inyección de Dependencias, entre otros.
- La definición de los estándares de codificación que se asumieron en la implementación de la aplicación permiten obtener un código más legible y entendible, ayudando a la comunicación entre desarrolladores.

## *Capítulo 3: Diseño del Sistema*

*Portal WAP para la plataforma de entrega de contenido*

*Universidad de Las Ciencias Informáticas*

---

- La descripción de las clases y sus operaciones como una de las tareas a tener en cuenta durante la documentación del código ayuda a conocer que se hace y por qué.

## **Capítulo 4: Implementación y prueba**

### **4.1 Introducción**

En el presente capítulo se abordan temas como la implementación de la aplicación en términos de componentes previamente identificados en el diseño. Se muestra el diagrama de despliegue de la aplicación donde se especifican los nodos necesarios para el despliegue. Se especifica la distribución de las librerías, clases e interfaces mediante el diagrama de componentes. Por último se realizan las pruebas necesarias a la aplicación para detectar la mayor cantidad de errores y deficiencias posibles, para ello se llevan a cabo las pruebas de caja blanca, caja negra y unidad.

### **4.2 Modelo de implementación**

Es una colección de componentes y los subsistemas de aplicación que los contienen. El mismo describe la organización de los elementos estáticos del software (código, datos y otros artefactos de acompañamiento) del entorno de desarrollo, en términos de paquetes, capas, y la gestión de la configuración (propiedades, la estrategia de liberación, entre otros).

### **4.3 Diagrama de despliegue**

Un diagrama de despliegue muestra cómo y dónde se desplegará el sistema. Describen la topología del sistema, la estructura de los elementos de hardware y el software que ejecuta cada uno de ellos. El diagrama de despliegue representa a los nodos y sus relaciones. Los nodos son conectados por asociaciones de comunicación tales como enlaces de red, conexiones TCP/IP. (40)

La figura 16 muestra el diagrama de despliegue para la aplicación donde se evidencia la distribución de los nodos y los protocolos para la comunicación entre ellos.

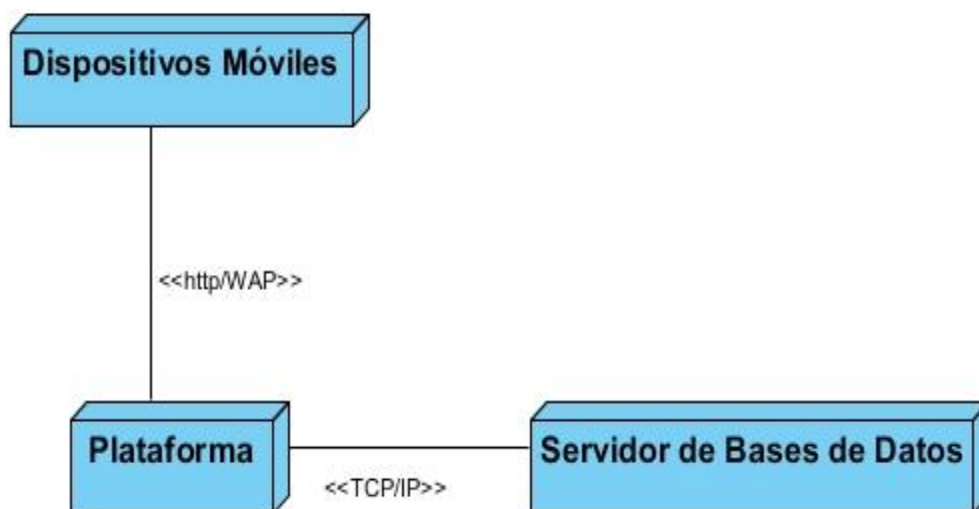


Figura 14: Diagrama de despliegue

Nodos	Descripción
Dispositivos Móviles	Representa el dispositivo mediante el cual el usuario se conecta utilizando un navegador a la plataforma de Gestión de Contenidos.
Plataforma	Representa el servidor donde se encuentra desplegada la plataforma de Gestión de Contenidos para dispositivos móviles.
Servidor de Bases de Datos	Representa el servidor donde se encuentra alojada la base de datos de la plataforma y que se comunica mediante el protocolo TCP/IP con la esta.

Tabla 2: Descripción de los nodos de despliegue

### 4.3.1 Protocolos utilizados

**HTTP:** Hypertext Transfer Protocol (en español protocolo de transferencia de hipertexto) es el protocolo usado en cada transacción en la Web. Es el protocolo utilizado para realizar la comunicación entre los dispositivos móviles que lo soporten y la plataforma.

**WAP:** El protocolo de acceso inalámbrico es el utilizado para la comunicación entre los dispositivos móviles de memorias pequeñas y bajo ancho de banda, y la plataforma.

**TCP/IP:** Este protocolo sirve para enlazar computadoras con diferentes sistemas operativos y que no pertenecen a la misma red. Este protocolo garantiza la entrega

de los datos que son enviados. Este es el protocolo utilizado para realizar la comunicación entre el servidor de bases de datos y la plataforma.

#### 4.4 Diagrama de componentes

Un diagrama de componentes ilustra los fragmentos de software, controladores embebidos, etc. que conformarán un sistema. Un diagrama de componentes tiene un nivel de abstracción más elevado que un diagrama de clase - usualmente un componente se implementa por una o más clases (u objetos) en tiempo de ejecución. Estos son bloques de construcción, como así eventualmente un componente puede comprender una gran porción de un sistema. (41)

La figura 17 muestra el diagrama de paquetes de la aplicación.

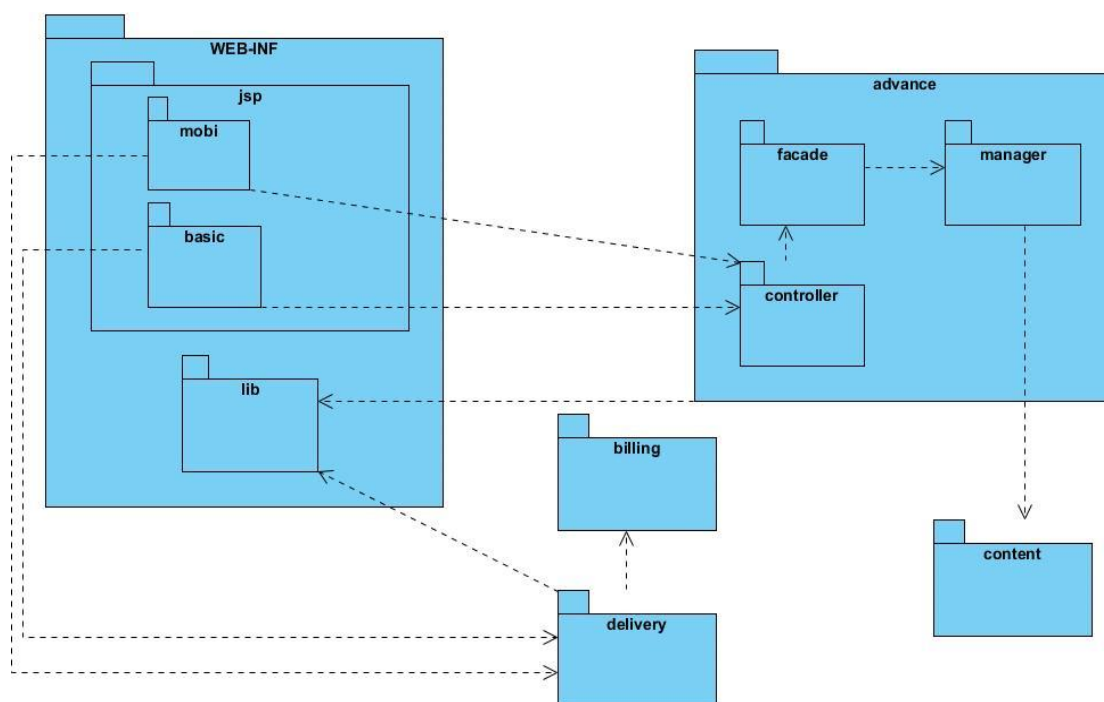


Figura 15: Diagrama de componentes

En el diagrama de paquetes se ilustran los paquetes utilizados en la aplicación, se procederá a continuación a explicar algunos de los más importantes:

**advance:** Es el contenedor de los paquetes facade, manager y controller. De ellos el paquete facade es el encargado de poseer la interfaz utilizada en la aplicación. El paquete manager contiene las clases que se encargan de la lógica de negocio. El



## Capítulo 4: Implementación y Prueba

### Portal WAP para la plataforma de entrega de contenido

Universidad de Las Ciencias Informáticas

controller posee las clases que se encargan de controlar la creación de las vistas con las que interactúa el usuario.

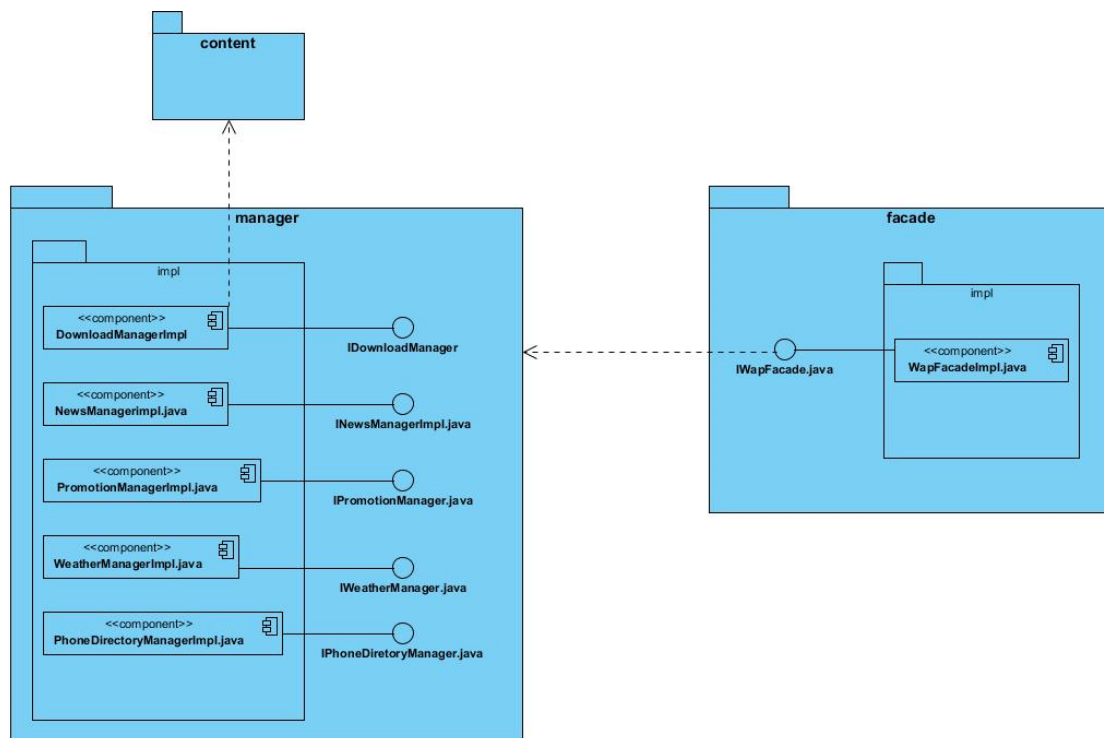


Figura 16: Diagrama de componentes del paquete manager

**jsp:** Es el contenedor de los paquetes mobi y basic. En el mobi se encuentran las vistas que serán visualizadas en los dispositivos móviles una resolución de más de 200 pixeles de ancho. Por su parte el paquete basic contendrá las vistas que serán expuestas en los dispositivos móviles que poseen poca capacidad de memoria y procesador.

**lib:** Es el contenedor de todas las librerías utilizadas para la realización y el buen funcionamiento de la aplicación.

**delivery:** Es el contenedor de las clases que intervienen en la descarga de contenidos de la aplicación.

**billing:** Es el contenedor de las clases necesarias para la generación del archivo CDR luego de haberse realizado la descarga correspondiente.

### 4.5 Pruebas al software

Según la IEEE una prueba es: "Una actividad en la cual un sistema o uno de sus componentes se ejecuta en circunstancias previamente especificadas, los resultados se observan, registran y se realiza una evaluación de algún aspecto". El objetivo fundamental de las mismas es medir el grado en que el software cumple con los requisitos.

Las pruebas son el proceso de ejecución de un programa con la intención de descubrir un error. Un buen caso de prueba es aquel que tiene altas probabilidades de detectar un error no descubierto hasta entonces. Una prueba tiene éxito si logra descubrir algún error que no ha sido detectado hasta entonces. (42)

#### 4.5.1 Pruebas de caja blanca

Al desarrollar un nuevo software o sistema de información, la primera etapa de pruebas a considerar es la etapa de pruebas unitarias o también llamada pruebas de caja blanca (White Box), estas pruebas también son llamadas pruebas modulares ya que nos permiten determinar si un módulo del programa está listo y correctamente terminado, estas pruebas no se deben confundir con las pruebas informales que realiza el programador mientras está desarrollando el módulo. En la figura 20 se muestra el flujo de la prueba de caja blanca. (43)

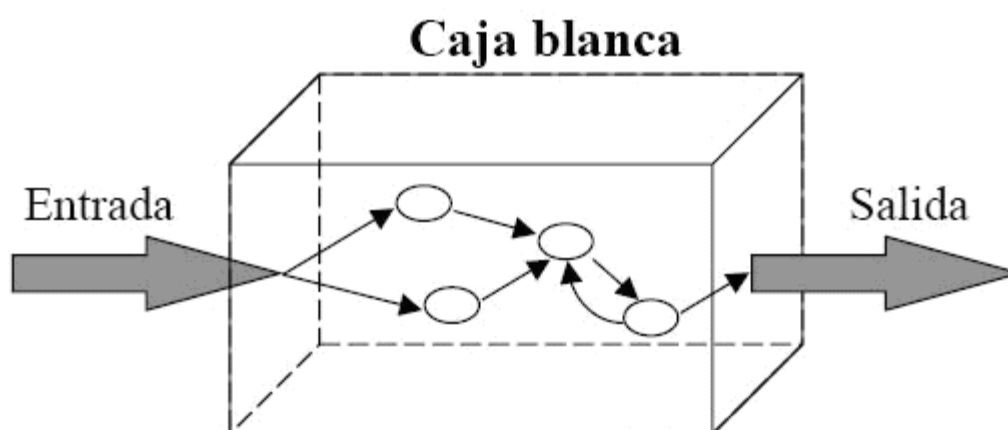


Figura 17: Pruebas de caja blanca

En la figura 24 se muestra lo que se considera una representación clásica de pruebas de caja blanca, en este tipo de pruebas el cubo representaría un sistema en

## Capítulo 4: Implementación y Prueba

Portal WAP para la plataforma de entrega de contenido

Universidad de Las Ciencias Informáticas

---

donde se pueden observar los diversos componentes que forman parte del mismo.  
(43)

Mediante los métodos de prueba de caja blanca se puede obtener casos de prueba que:

- Garanticen que se ejercita por lo menos una vez todos los caminos independientes de cada módulo.
- Ejerciten todas las decisiones lógicas en sus vertientes verdadera y falsa.
- Ejecuten todos los bucles en sus límites y con sus límites operacionales.
- Ejerciten las estructuras internas de datos para asegurar su validez.

Algunas técnicas de prueba de caja blanca son:

- Prueba de condición: ejercita las condiciones lógicas contenidas en el módulo de un programa.
- Prueba del flujo de datos: selecciona caminos de prueba de un programa de acuerdo con la ubicación de las definiciones y los usos de las variables del programa.
- Prueba de bucles: se centra exclusivamente en la validez de las construcciones de bucles.
- La prueba del camino básico: permite obtener una medida de la complejidad lógica de un diseño procedimental y usar esa medida como guía para la definición de un conjunto básico de caminos de ejecución. La idea es realizar un grupo de casos de prueba que garanticen que se pueda probar por lo menos una vez cada una de las sentencias del programa. (42)

Se realizaron pruebas de camino básico a las funcionalidades de la aplicación, esta permite definir los diferentes caminos independientes de la función y probar su funcionamiento al menos una vez. A continuación se muestra el proceso desarrollado al método "getLastContentByDevice" de la clase "DownloadsControllerAdvance".

Para ello inicialmente se analiza el código y enumeran las instrucciones como se muestra en la figura 21. Seguidamente es necesario representar el grafo de flujo (o grafo del programa), que va a representar el flujo de control lógico del código anterior, a través de nodos, aristas y regiones. Quedando como muestra la figura 22.

## Capítulo 4: Implementación y Prueba

### Portal WAP para la plataforma de entrega de contenido

Universidad de Las Ciencias Informáticas

---

```
private List<ContentEntity> getLastContentByDevice(long idDevice,
    long idCategory, String action) {

    List<ContentEntity> contents = null; //1
    if (action == "last")// 2

        contents = managerFacade.lastContentByDevice(0, 5, idDevice,idCategory);//3

    else if(action == "topDownload")//4

        contents = managerFacade.lastUploadContentByDevice(idDevice, idCategory);//5

    for (ContentEntity contentEntity : contents) {//6

        String contentUrl = "../download/init/" +
            fileUtil.buildFileName (contentEntity.getName(),contentEntity.getMimeType())+
            "?code=" + contentEntity.getCode();// 7
        contentEntity.setContentUrl(contentUrl);// 7
    }//8

    return contents;//9
}
```

Figura 18: Función a la que se aplica camino básico

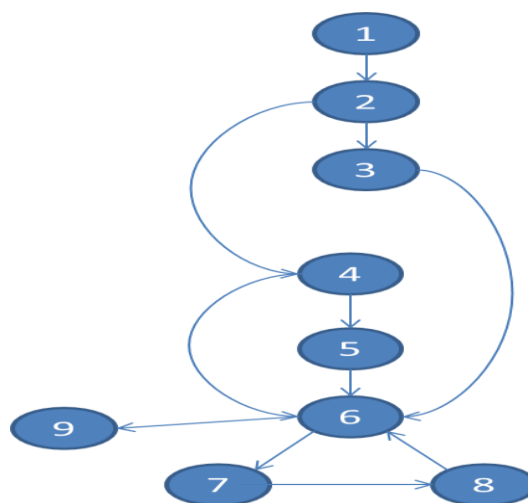


Figura 19: Grafo de flujo

## Capítulo 4: Implementación y Prueba

Portal WAP para la plataforma de entrega de contenido

Universidad de Las Ciencias Informáticas

---

Luego de realizado el grafo es necesario conocer la cantidad de caminos independientes que se deben buscar para probar; y para esto se calcula la complejidad ciclomática, que se puede calcular de tres formas:

1. El número de regiones (cantidad de regiones cerradas) del grafo de flujo más una unidad coincide con la complejidad ciclomática. Por lo que  
$$V(G) = 3 + 1 = 4$$
2.  $V(G) = (A - N) + 2$  donde “A” es el número de aristas del grafo de flujo y “N” es el número de nodos del mismo.  
$$V(G) = 11 - 9 + 2 = 4$$
3.  $V(G) = P + 1$  donde “P” es el número de nodos predicado (nodos con más de una arista saliente) contenidos en el grafo. (44)  
Los nodos 2, 4 y 6 son nodos predicados.  $V(G) = 3 + 1 = 4$

Como se muestra en cualquiera de las tres formas anteriores la complejidad ciclomática es 4, por lo que la cantidad de caminos básicos que puede tomar el algoritmo durante su ejecución es 4 y quedan definidos así:

Camino básico # 1: 1-2-3-6-7-8-9

Camino básico # 2: 1-2-4-5-6-7-8-9

Camino básico # 3: 1-2-4-6-7-8-9

Camino básico # 4: 1-2-4-6-9

Seguidamente se preparan los casos de pruebas que forzarán la ejecución de cada camino del conjunto básico, quedando de la siguiente forma:

**Descripción:** Se debe retornar la lista de los últimos contenidos devueltos por la función “getLastContentByDevice”, los parámetros “idDevice” e “idCategory” toman valores de -1 o superiores.

Esta descripción es común para todos los casos de prueba que se realizarán.

**Caso de prueba para el camino básico # 1:**

**Condición de ejecución:** Para esta prueba es necesario que la variable “action” posea el valor de “last”.

**Entrada:** La variable “action” con valor “last”.

## *Capítulo 4: Implementación y Prueba*

### *Portal WAP para la plataforma de entrega de contenido*

*Universidad de Las Ciencias Informáticas*

---

**Resultados esperados:** Teniendo en cuenta la condición de ejecución se espera que devuelva los últimos contenidos que han sido subidos, como resultado se retorna la lista de los últimos contenidos subidos.

El resultado obtenido fue correcto.

#### **Caso de prueba para el camino básico # 2:**

**Condición de ejecución:** Para esta prueba es necesario que la variable “action” posea el valor de “topDownload”.

**Entrada:** La variable “action” con valor “topDownload”.

**Resultados esperados:** Teniendo en cuenta la condición de ejecución se espera que devuelva los contenidos más descargados, como resultado se retorna la lista de los contenidos más descargados.

El resultado obtenido fue correcto.

#### **Caso de prueba para el camino básico # 3:**

**Condición de ejecución:** Para esta prueba es necesario que la variable “action” posea un valor distinto a “topDownload” y “last”.

**Entrada:** La variable “action” con valor distinto a “topDownload” y “last”.

**Resultados esperados:** Teniendo en cuenta la condición de ejecución se espera que la lista vacía, como resultado se retorna la lista de los contenidos vacía.

El resultado obtenido fue correcto.

#### **Caso de prueba para el camino básico # 4:**

**Condición de ejecución:** Para esta prueba es necesario que la variable “action” posea un valor distinto a “topDownload” y “last”.

**Entrada:** La variable “action” con valor distinto a “topDownload” y “last”.

**Resultados esperados:** Teniendo en cuenta la condición de ejecución se espera que la lista vacía, como resultado se retorna la lista de los contenidos vacía.

El resultado obtenido fue correcto.

Luego de aplicados los casos de prueba los resultados obtenidos fueron los esperados concluyéndose que las funcionalidades están implementadas correctamente.

#### **4.5.2 Pruebas unitarias**

Otro de los métodos de prueba aplicado a la solución fue el de las pruebas unitarias, el cual centra el proceso de verificación en la menor unidad del diseño del software: el componente software o módulo.

Dichas pruebas fueron realizadas a las clases del negocio ubicadas en el paquete manager de la aplicación. Se le realizaron las pruebas a la clase “NewsManagerImpl” comprobando la correcta obtención de las noticias, las cuales poseen un título, una URL y un identificador. En la clase “PromotionManagerImpl” se prosigue a realizar las pruebas a la obtención de la lista de promociones existentes en la base de datos, las cuales, son ofertadas por diferentes proveedores y poseen un identificador, una imagen, un título y un texto. En la manager “PhoneDirectoryManagerImpl” se le realizaron las pruebas unitarias a sus métodos. Esta clase posee categorías compuestas por un identificador y un nombre estos datos no pueden ser vacíos, al igual que las lista que contiene las categorías existentes. Se le realizaron las pruebas de caja blanca a la clase manager “WeatherManagerImpl” la cual mantendrá actualizado a los usuarios sobre el Estado Meteorológico actual, la misma posee un objeto de “WeatherClient” que tiene una descripción, sumario, título, fecha. Estos campos no pueden ser nulos o vacíos en la aplicación para garantizar un buen funcionamiento de sus métodos. Y por último se realizó la prueba a la clase “DownloadsManagerImpl” en la que se obtiene la lista de entidades de los contenidos, los cuales poseen una URL, una descripción y un MimeType del contenido.

La figura 23 muestra un ejemplo de las pruebas unitarias realizadas con JUnit a la funcionalidad “lastContentsByDevice”, que permite obtener los últimos contenidos dado un dispositivo.

## Capítulo 4: Implementación y Prueba

### Portal WAP para la plataforma de entrega de contenido

Universidad de Las Ciencias Informáticas

```
protected void runTest() {
    ApplicationContext appContext = new ClassPathXmlApplicationContext(
        "cu/uci/comcel/portal/wap/advance/test/prueba.xml");
    DownloadsManagerImpl objDownload = (DownloadsManagerImpl) appContext
        .getBean("downloadsAdmin2");
    try {
        List<ContentEntity> downloads = objDownload.lastContentsByDevice();
        for (int j = 0; j < downloads.size(); j++) {
            String url = downloads.get(j).getContentUrl();
            String description = downloads.get(j).getDescription();
            String mimeType = downloads.get(j).getMimeType();

            if (downloads.get(j).getContentUrl() == null
                || downloads.get(j).getDescription() != null
                || downloads.get(j).getMimeType() != null) {
                assertTrue("Lista Vacía", downloads != null);
                assertTrue("La Descripción esta Vacía", description != null);
                assertTrue("La URL es Nula", url != null);
                assertTrue("MimeType esta Nulo", mimeType != null);
            }
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

Figura 20: Prueba a la funcionalidad lastContentsByDevice

La realización de las pruebas a los módulos de la aplicación arrojó resultados satisfactorios, al no encontrarse ningún error. También no ocurrió ninguna falla en el transcurso de las pruebas y los tiempos de respuestas de los métodos testeados se encuentran entre 2.203 y 4.594 segundos, dependiendo estos tiempos de la cantidad de datos que sean extraídos de la base de datos.

En la figura 24 se muestra un ejemplo del resultado de la prueba unitaria realizada a la funcionalidad "lastContentsByDevice".



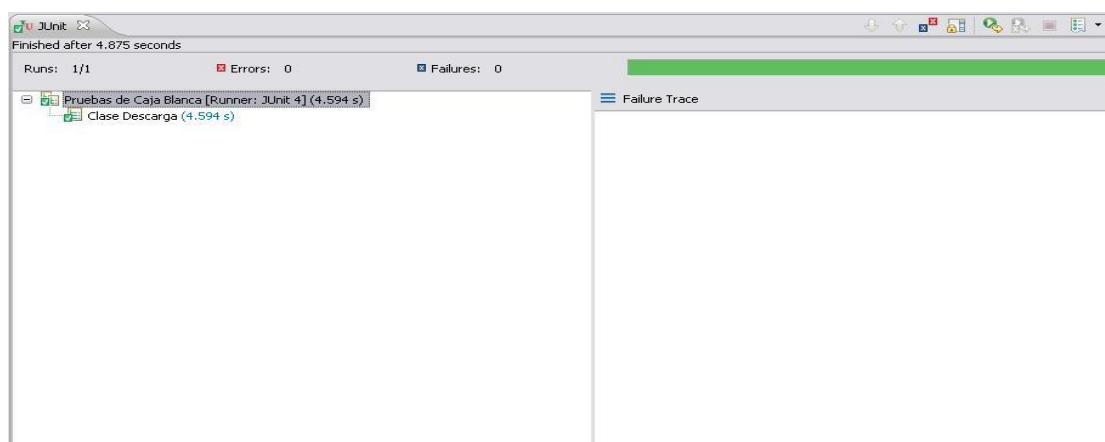


Figura 21: Resultado de la prueba a la funcionalidad lastContentsByDevice

El resto de las pruebas ejercidas a las clases se encuentran en el **¡Error! No se encuentra el origen de la referencia..**

### 4.5.3 Pruebas de caja negra

Se le aplicaron las pruebas de caja negra a las interfaces del software, el cual demostró que las funcionalidades de la aplicación son óptimas. Los casos de prueba demostraron que todas las entradas fueron aceptadas adecuadamente y las cuales dan una salida correcta, así como que la integridad de la información externa se mantiene. En la figura 25 se muestra el flujo de las pruebas de caja negra. (43)

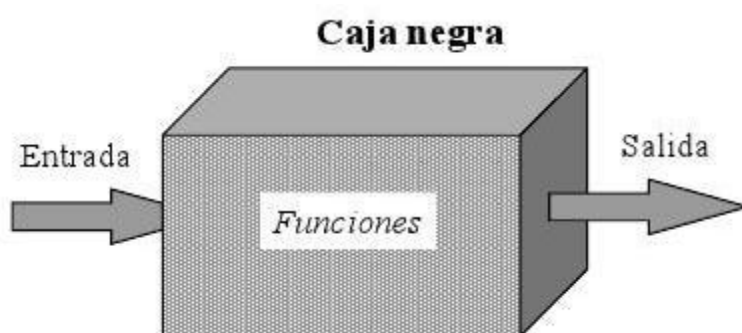


Figura 22: Pruebas de caja negra (44)

Las descripciones de los casos de pruebas se encuentran en el **¡Error! No se encuentra el origen de la referencia..**

### Resultado de las pruebas

## Capítulo 4: Implementación y Prueba

Portal WAP para la plataforma de entrega de contenido

Universidad de Las Ciencias Informáticas

Una vez aplicadas las pruebas de caja negra a las funcionalidades de la aplicación se obtuvieron las no conformidades siguientes:

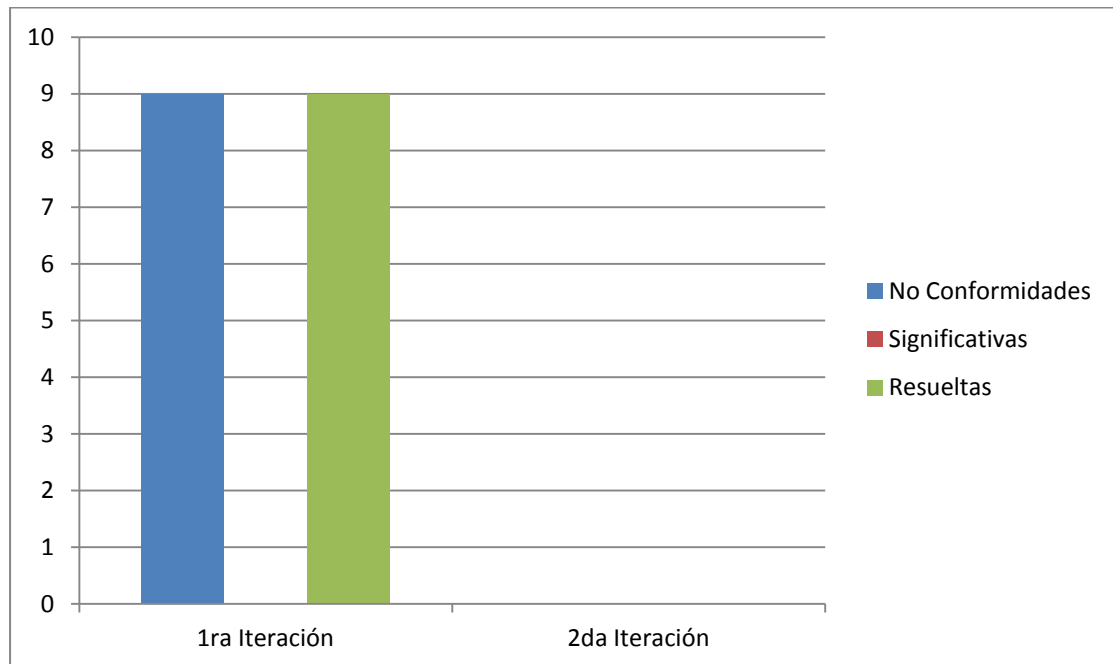


Figura 23: Gráfico de resultados de las pruebas

De los casos de prueba realizados a las funcionalidades de la aplicación, se obtuvieron en una primera iteración nueve no conformidades y de ellas ninguna significativa, todas fueron resueltas. En una 2da iteración se comprobó el correcto funcionamiento de todas las funcionalidades.

### 4.6 Conclusiones

Se puede concluir que:

- El modelo de implementación propuesto permitió una comprensión de la organización de los elementos que forman parte de la aplicación así como de la integración entre los componentes.
- Las pruebas de caja blanca arrojaron resultados satisfactorios, utilizándose la técnica del camino básico que permitió definir los diferentes caminos independientes de las funcionalidades y probar su funcionamiento al menos una vez.
- Las pruebas de unidad desarrolladas a algoritmos arrojaron que la información fluye de forma adecuada hacia y desde la unidad de la aplicación

## *Capítulo 4: Implementación y Prueba*

### *Portal WAP para la plataforma de entrega de contenido*

*Universidad de Las Ciencias Informáticas*

---

que está siendo probada. Se examinaron las estructuras de datos locales para asegurar que los datos que se mantienen temporalmente conservan su integridad durante todos los pasos de ejecución del algoritmo.

- Las pruebas de caja negra realizadas arrojaron resultados satisfactorios, al encontrar un total de sólo 9 no conformidades entre errores de validación de campos de entrada y errores ortográficos en la interfaz de algunos componentes, los cuales fueron corregidos.

### **Conclusiones generales**

Se puede concluir que:

El estudio de varios sistemas existentes arrojó como resultado que no satisfacen las necesidades solicitadas por el cliente. Las herramientas, lenguajes y metodologías utilizadas posibilitaron generar los artefactos necesarios, obteniéndose con estos últimos una idea general de lo que la aplicación debe hacer y cómo debe comportarse, además de facilitar su desarrollo ya que tuvo en cuenta las características del marco de trabajo empleado, que permite a través del módulo MVC una mayor organización, mantenimiento y facilita la interoperabilidad entre los componentes de la aplicación.

El módulo de entrega de contenidos multimedias al usuario, se realizó utilizando métodos y estándares soportados por una amplia gama de dispositivos móviles actuales, ejemplos de ellos es la utilización del estándar de Descarga OMA que posee métodos para realizar una entrega confiable notificando el estado en que esta finalizó. El módulo además permite a la aplicación entregar el contenido mediante el método que sea soportado por el dispositivo móvil que realice la solicitud.

El registro de las descargas permite la posterior facturación de los contenidos, porque se realiza a través los ficheros CDR que contienen un conjunto de parámetros necesarios referentes a la descarga realizada.

Con la utilización de servidor ALEMBIK se logra adaptar los contenidos mediante la transcodificación que este realiza para entregar una multimedia que sea soportada por el dispositivo que la solicite.

Con el empleo de los métodos de protección de los derechos digitales, el módulo desarrollado es capaz de garantizar las restricciones de uso impuestas al contenido por el proveedor de contenidos multimedias.

El subsistema de usuario garantiza que los contenidos disponibles solo sean descargados por los usuarios del portal. Para ello se utiliza el módulo de seguridad un marco de trabajo confiable como es Spring 2.5 que permite la creación de

## *Conclusiones Generales*

### *Portal WAP para la plataforma de entrega de contenido*

*Universidad de Las Ciencias Informáticas*

---

aplicaciones con control de acceso basado en roles, para ello se basa en la aplicación de reglas de las URLs solicitadas.

Con la realización de este trabajo se lograron implementar los servicios de valor agregados como promociones, noticias, estado del tiempo y búsquedas en el directorio telefónico, generando interés y brindando un servicio único a los usuarios que accedan al portal.

En la realización de las pruebas se comprobaron los diferentes caminos independientes de las funcionalidades y su funcionamiento al menos una vez. Las pruebas de caja negra realizadas arrojaron resultados satisfactorios, al encontrar un total de sólo 9 no conformidades entre errores de validación de campos de entrada y errores ortográficos en la interfaz de algunos componentes, los cuales fueron corregidos

Como producto final se obtuvo una aplicación para la plataforma de entrega de contenidos que posee un portal WAP para visualizar los contenidos en los dispositivos móviles. El país no cuenta con una aplicación de este tipo siendo la creación de la misma un avance tecnológico importante además de dar un paso más a la soberanía tecnología del país. También posibilitará brindar un servicio nuevo a los usuarios cubanos de la telefonía celular. Se crea además un nuevo mercado para los proveedores de contenidos.

## **Recomendaciones**

Una vez terminado el desarrollo de la aplicación se recomienda:

- Realizar mejoras a los componentes en función de las nuevas versiones de las tecnologías empleadas, pues en ocasiones se producen nuevas versiones de las herramientas con el objetivo de eliminar brechas de seguridad existentes en versiones anteriores.
- Añadir nuevas funcionalidades en un futuro como la integración con la aplicación de red social que permite el marketing viral de los contenidos (45) ofertados, pago de servicios públicos por teléfono celular, visualización en línea de multimedias, para continuar renovando el portal y mantener el interés de los clientes al incrementarse el número de servicios brindados.
- Utilizar el DRM v2.0 en una posterior versión para incluir mejoras en la protección de los derechos de autor que esta versión trae implementados.