*Universidad de las Ciencias Informáticas*

*Facultad 2*


# ***Título:*** *Plataforma de Gestión de Contenidos para Dispositivos Móviles. Módulo Entrega de Contenido.*

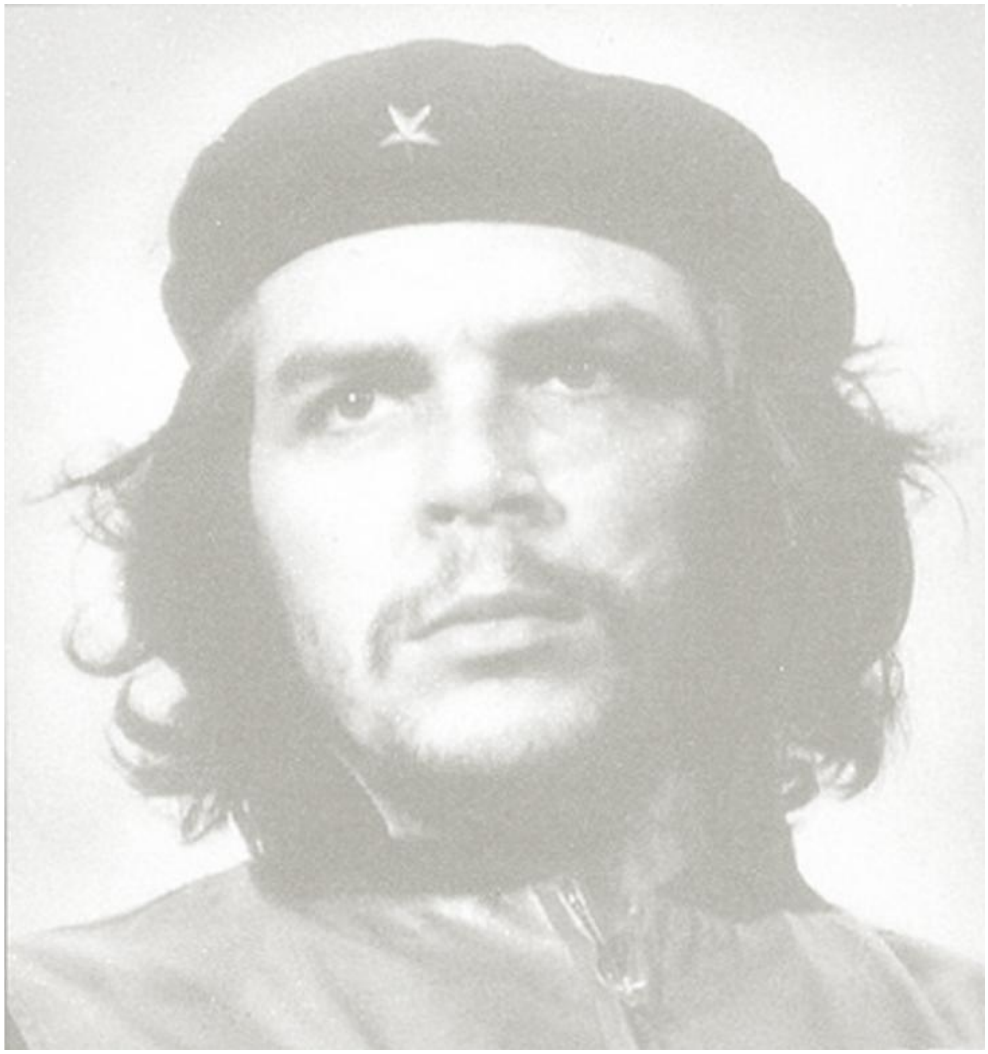
*Trabajo de Diploma para Optar por el Título de Ingeniero en Ciencias Informáticas.*

*Autores: Yoendris Reina Arzuaga.*

*Mayret Sosa Gómez.*

*Tutor: Ing. José Antonio Plá Rodríguez.*

*Ciudad de la Habana, junio de 2009*



*"Todos y cada uno de nosotros paga puntualmente su cuota de sacrificio consciente de recibir el premio en la satisfacción del deber cumplido, conscientes de avanzar con todos hacia el Hombre Nuevo que se vislumbra en el horizonte."*

*CHE*

Agradecimientos:

*Le agradezco a mi mamá Nancy por su eterno amor, comprensión y dedicación, a mi papá Eliseo por ser un excelente padre, gracias Pito, a ustedes que les debo todo lo que soy. A mi hermanito Reyniel por confiar en mí. A mi novio Sandy por entenderme, por sus palabras de apoyo que supieron darme la fuerza cada vez que las necesitaba y por los buenos momentos a lo largo de todo este tiempo, gracias por todo AFU mi amor.*

*A mi tía Norma y mi tío Carlos por quererme como una hija. A mi tía Maruchy y Ernestico por tenerme siempre presente.*

*A mi cuñadita Amarilys porque lo único que has hecho es ayudarme desde que te conocí.*

*A mis amigas Dolores, Elda, Yarima, Duni, Vivi, a mi amigo Veiti gracias a ustedes que han estado cada vez que los necesité.*

*A mi compañero de tesis Yoendris por ser tan dedicado y por aguantarme todo este tiempo.*

*A mi tutor Pla por saber guiarnos y a todos los profesores del proyecto Cubacel por apoyarnos.*

*En fin a todas aquellas personas que se han preocupado por mí.*

*A todos ustedes Gracias*

*Mayret*

Agradecimientos:

*A mi mamá Odalis, por ayudarme tanto en la vida, por estar siempre pendiente de mi, porque se que me quiere mucho y ha sabido entregarme ese infinito amor que caracteriza a una madre.*

*A mi papá Julio por ser mi amigo y apoyarme siempre, por darme todo lo que ha podido y por esa confianza que tiene en si mismo.*

*A mi abuela toña, a mis tías Odaelvis, Odaisy, nina y tío pipi por confiar en mí y ayudarme siempre. A mi abuela Emelia, a mis tíos Arnoldo, Rosendo, chichi y Amaury por preocuparse por mí.A mi hermano Yoandris por ser el mejor hermano del mundo.*

*A Dianellis por comprenderme y apoyarme todo este tiempo.*

*A Mayret por ser tan preocupada y responsable, por nunca perder la fuerza de seguir adelante, por ayudarme y apoyarme en todo momento.*

*A Plá por su excelente labor como tutor de este trabajo, por la inmensa ayuda que me brindó y porque gracias a él obtuve nuevos conocimientos.*

*A Darien por ser tan preocupado, por siempre tener una respuesta a las dudas planteadas y por toda la ayuda que me brindó.*

*A Sandy Noa por su ayuda y apoyo hasta el último momento.*

*A Juan Reyna, Marbelia, Maydelis Milanés, a mis amigos y a todo aquel que de una manera u otra ayudó en mi formación profesional.*

*Yoendris*

Dedicatoria:

*A mis padres que son lo más importante de mi vida.*
*A mi hermano, a mi novio por estar siempre a mi lado.*
*A toda mi familia por apoyarme*
*Mayret*

*A mis Padres por ser lo mejor de este mundo.*
*A mi hermano y a mi familia por confiar en mí.*
*Yoendris*

# Resumen:

El desarrollo de la telefonía celular en Cuba ha experimentado ciertos avances en los últimos años, en consecuencia la empresa Cubacel ha decidido fortalecer e incrementar los servicios brindados, solicitando ayuda a la Universidad de las Ciencias Informáticas (UCI) para su desarrollo. El presente trabajo tiene por objetivo desarrollar el Módulo Entrega de Contenido como componente de la Plataforma de Gestión de Contenidos para Dispositivos Móviles. Un proceso que garantice la entrega de contenidos de manera eficiente hacia los clientes empleando el método OMA Download y aplicando sobre los mismos derechos digitales de autor o DRM es sin lugar a dudas una alternativa viable para la empresa. La opción de aplicar DRM a los contenidos está sustentada en un mecanismo conocido como Forward Lock, encargado de impedir el reenvío de los mismos a otros clientes. Suplir la carencia de dicho servicio constituye el objetivo central de la investigación y el sistema desarrollado.

Teniendo en cuenta la necesidad existente se ha diseñado y elaborado el módulo: Entrega de Contenido, cuyos resultados lograrán mejorar la calidad del servicio de la telefonía móvil en Cuba y dar pasos sólidos que permitan ir colocando paulatinamente al país a la altura de empresas internacionales experimentadas en el sector.

**Palabras claves:**

Telefonía celular en Cuba, Entrega de contenido, *OMA Download*, DRM, *Forward Lock*.

# Índice

# *Índice de Figuras*

# *Índice de Tablas*

# Introducción

La telefonía celular se remonta a los inicios de la Segunda Guerra Mundial, momentos estos donde se apreciaba la necesidad de la comunicación a distancia, debido a ello la compañía Motorola creó un equipo conocido por Handie Talkie H12-16, que permitió el contacto con las tropas mediante ondas radiales. Años después Martin Cooper, pionero y considerado como el padre de la telefonía celular, fabricó el primer radio teléfono, exactamente entre los años 1970 y 1973 en Estados Unidos. (1)

Esta tecnología se caracterizaba por ser analógica (la transmisión y recepción de datos se apoyaba sobre un conjunto de ondas de radio que cambiaban de modo continuo), esto traía consigo una serie de inconvenientes, tales como que solo podían ser utilizados para la transmisión de voz. Posteriormente se convierte en telefonía digital, permitiendo la mejora del manejo de llamadas y más enlaces simultáneos utilizando el mismo ancho de banda.

Esta tecnología brinda un gran espectro de posibilidades, esencialmente para los celulares de últimas generaciones como el envío de SMS (por sus siglas en inglés *Short Message Service*), MMS (por sus siglas en inglés *Multimedia Messaging System*), soporte para la navegación en Internet, además de servicios como la descarga de contenidos, la cual le permite a los usuarios tener en su dispositivo móvil contenidos como tonos, imágenes, fotos, juegos, etc.

Cuba es un país subdesarrollado que independientemente de las restricciones impuestas por el gobierno de Estados Unidos, ha intentado insertarse en los avances tecnológicos que toman mayor auge en el marco internacional, destacando la esfera de la tecnología celular, la cual sentó sus bases por primera vez en el año 1991 con la creación de la Empresa Mixta Cubacel.

La empresa Cubacel es encargada de operar la telefonía celular en Cuba, sin embargo, a pesar de los esfuerzos realizados no ha sido posible que se brinden la totalidad de servicios que en el mundo contemporáneo se consumen, ejemplo de ello es un servicio que garantice la entrega de contenidos hacia los clientes.

Esta problemática conllevó al establecimiento de un acuerdo de colaboración entre Cubacel y la Universidad de las Ciencias Informáticas, asignándole a esta última la responsabilidad del desarrollo del servicio. Para dar solución al problema será elaborada una Plataforma de Gestión de Contenidos para Dispositivos Móviles, encargada de realizar todo el proceso para que cada usuario pueda descargar el contenido que desee, sin necesidad de verificar temas como la compatibilidad del móvil con el archivo solicitado. Para gestionar el envío de contenidos hacia los dispositivos móviles, se desarrollará el Módulo Entrega de Contenido, indispensable para lograr un buen funcionamiento en la descarga de estos.

Con el objetivo de solucionar la problemática existente en la Empresa Cubacel, en cuanto a la descarga de contenidos se plantea el siguiente **problema científico:** ¿Cómo realizar la entrega de contenidos dentro de la Plataforma de Gestión de Contenidos para Dispositivos Móviles?

Como **objetivo general** se plantea**:** desarrollar el Módulo Entrega de Contenido como componente de la Plataforma de Gestión de Contenidos para Dispositivos Móviles (Celulares).

Teniendo en cuenta lo planteado anteriormente como **objeto de estudio** se propone: los procesos asociados a la gestión de contenidos hacia celulares. El **campo de acción** estaría enmarcado en los procesos que se llevan a cabo para la entrega de contenidos hacia los celulares. Se ha determinado establecer la siguiente **pregunta científica**: ¿Cómo desarrollar un módulo de *software* que entregue el contenido vinculando los restantes módulos? En correspondencia con el objetivo planteado y la problemática existente se ha decidido enfocar las **tareas investigativas** a:

- Realizar un análisis del estado del arte relativo a los métodos empleados para garantizar el servicio de entrega de contenidos hacia celulares.
- Analizar las diferentes formas de solicitar una descarga de contenidos por parte de los dispositivos móviles.
- Evaluar las diferentes formas de gestionar el derecho de autor hacia los contenidos que serán entregados a los clientes.

El presente documento está estructurado en 5 Capítulos:

Capítulo 1: Fundamentación Teórica

“Fundamentación Teórica “en este capítulo se encontrarán los principales conceptos que se manejan a lo largo del trabajo, para facilitar información necesaria relacionada con el tema; así como el estado del arte del tema tratado, a nivel internacional y nacional. Además incluye las metodologías, herramientas y lenguajes de programación utilizados para el desarrollo del Módulo Entrega de Contenido y la selección de las tecnologías adecuadas.

Capítulo 2: Características del Sistema

“Características del Sistema” en este capítulo se describe detalladamente el problema existente y las características del sistema a desarrollar como solución a esta problemática. Se identifican los principales conceptos mediante un Modelo de Dominio necesarios para un buen entendimiento del sistema y los requisitos que se deben cumplir el sistema una vez concluido.

Capítulo 3: Análisis y Diseño del Sistema

“Análisis y Diseño del Sistema” este capítulo se enfoca en los diagramas de clases del análisis, diagramas de diseño para cada caso de uso del sistema, y la realización de los diagramas de interacción del sistema específicamente los diagramas de secuencia propios de la metodología seleccionada para el desarrollo del sistema, así como la descripción de los patrones de diseño utilizados.

Capítulo 4: Implementación y Prueba

“Implementación y Prueba” en este capítulo se exponen las principales características del flujo de trabajo de implementación, como el modelo de despliegue y el de componente, así como la realización de casos de prueba para definir los posibles fallos.

Capítulo 5. Estudio de la Factibilidad

**“**Estudio de la Factibilidad” este capítulo muestra la estimación del esfuerzo, tiempo de desarrollo, así como el análisis de costos y los beneficios asociados a la realización del módulo, proporcionando una visión general de la factibilidad.

# Capítulo 1: Fundamentación teórica

## *1.1 Introducción*

En el presente capítulo se manifiestan los términos que están relacionados con el dominio del problema científico; donde se desarrolla una investigación para seleccionar el método de descarga más adecuado para dar solución a la situación planteada anteriormente. Se analizan las principales herramientas, tecnologías y lenguajes que serán utilizados para el desarrollo de la aplicación, argumentando las razones por las cuales se han elegido.

## *1.2 Métodos de descargas empleados en el mundo*

Los teléfonos celulares han representado un accesorio de trascendental importancia para sus usuarios, garantizando una combinación de funcionalidades básicas para la comunicación, con la posibilidad de tener en las manos un producto interesante que ofrezca servicios para el disfrute y entretenimiento de manera eficiente. Con el objetivo de garantizar estas funcionalidades, muchos clientes han hecho evolucionar su dispositivo, dándole la posibilidad de incorporar elementos de imágenes, música, video, entre otros. El acceso a dichos contenidos puede efectuarse a través de vías convencionales como la descarga directa desde un ordenador o vía WAP (por sus siglas en inglés *Wireless Application Protocol*), accediendo desde su propio navegador.

Con el transcurso de los años los celulares se han apoyado en diferentes métodos para descargar los contenidos, algunos de ellos son hoy casi obsoletos; un ejemplo que lo ilustra claramente, está referido a la descarga mediante *Openwave´s Download Fun*. Existen otros métodos para la descarga de contenidos como el conocido *OMA Download* que garantiza un proceso de descarga seguro hacia estos dispositivos móviles. (2). Este último método ha sido creado por la *Open Mobile Alliance,* organización creada en junio del 2002 como respuesta a la proliferación de los foros de la industria encargados de algunos protocolos. Cada uno de estos foros contaba con sus características específicas, sus procedimientos de toma de decisión, sus fechas del lanzamiento, ocasionando cierta sobrecarga en las especificaciones y causando la duplicidad del trabajo. OMA fue creado para recolectar estas iniciativas bajo una idea de

estandarización común.

*Open Mobile Alliance* recoge algunos principios elementales que guían su funcionamiento; su misión es proporcionar servicios interoperables que permitan trabajar a través de países, operadoras y dispositivos móviles. Además, pretende ser un foro para que las principales compañías de la industria se pongan de acuerdo y sigan unas especificaciones comunes para sus productos y servicios. (3)

### 1.2.1 *Openwave´s Download Fun*

*Openwave´s Download Fun* (ODF) requiere tanto de un cliente en el dispositivo, como de un servidor de descarga. La descarga es operada por un proveedor de servicios de comunicación, proporcionando el mecanismo para descargar objetos binarios de un sitio de contenido a un dispositivo móvil de forma segura y trabajando junto con el explorador WAP. Los usuarios pueden navegar a un sitio WAP que ofrece contenido descargable y seleccionar un contenido para su descarga. Una vez que el usuario selecciona el contenido se produce la descarga. Durante la realización de este proceso el servidor tiene la responsabilidad no solo garantizar que exista el espacio necesario en el dispositivo para almacenar el objeto, sino también garantizar que este sea soportado por el celular. (4)

ODF permite descargar archivos de cualquier tamaño, siempre y cuando el teléfono sea capaz de soportar dicho contenido. Algunos de los tipos de contenidos descargables incluyen tonos de llamada, fondos de escritorio, imágenes, juegos, protectores de pantalla, archivos de audio y vídeo, entre otros. (4)

### 1.2.2 *OMA Download*

*OMA Download* surge con el objetivo de determinar si un cliente ha podido descargar o instalar un contenido efectivamente; el término es usualmente conocido por OMA DD, donde DD hace referencia a *Download Descriptor* o Descriptor de Descargas, archivo XML que contiene elementos de configuración específicos. Previamente a la descarga del contenido, el cliente accede al descriptor de archivo, que contendrá toda la información relevante para poder continuar con el proceso.

Debido a ello *OMA Download* proporciona una funcionalidad importante a los dispositivos, lo cual permite comprobar si el contenido se podrá descargar y utilizar en el mismo, así como la confirmación del estado

de la descarga. Es soportado por la mayoría de dispositivos modernos, además constituye un estándar abierto que permite descargas de contenidos fiables.

En la siguiente tabla se describen los posibles elementos a ser incluidos en el descriptor:

| **Nombre** | **Definición** |
|---|---|
| *type* | El tipo de contenido del objeto a descargar. |
| *size* | El número de bytes para ser descargado desde la URL |
| *objectURI* | La URL de la cual el objeto será cargado. |
| *installNotifyURI* | URL que enviará un reporte con el estado de la instalación, ya sea en caso de éxito de la descarga, o en caso de un fracaso |
| *nextURL* | La URL a la que el cliente debe navegar en caso de que el usuario seleccione una acción de exploración después que la descarga de transacciones se ha completado con éxito o fracasado. |
| *DDVersion* | La versión de la tecnología del Descriptor de descargas. |
| *name* | El nombre que identifica el objeto para el usuario. |
| *description* | Una breve descripción textual del objeto. |
| *vendor* | La organización que proporciona el objeto. |
| *infoURL* | Una URL para describir el objeto. |
| *iconURL* | La URL de un icono. |
| *installParam* | Una instalación de parámetros asociados con el objeto a descargar. |

**Tabla 1: Contenidos que posee el descriptor**

En realidad, sólo los 3 primeros elementos (tipo, tamaño y objectURI) son obligatorios en el Descriptor de Descargas. (2)

Por lo expuesto anteriormente se ha seleccionado el método *OMA Download* para la realización de la entrega del contenido al cliente.

## *1.3 DRM*

La Gestión de derechos digitales (por sus siglas en inglés DRM; de *Digital Rights Managemen*t) es el conjunto de tecnologías orientadas a ejercer restricciones sobre los usuarios de un sistema o recurso. Es necesario para prevenir la duplicación sin autorización y así asegurar el flujo continuo de los ingresos. Constituyen procesos que protegen la propiedad intelectual durante las operaciones comerciales realizadas con contenidos digitales. (5)

Actualmente varias empresas implementan su propio mecanismo de DRM, sin embargo de manera general todos tienen características comunes:

- Detectan quién accede a cada obra, cuándo y bajo qué condiciones, y pueden reportar esta información al proveedor de la obra.
- Autorizan o deniegan de manera inapelable el acceso a la obra, de acuerdo a condiciones que pueden ser cambiadas unilateralmente por el proveedor de la obra con total independencia de lo que dicte el marco jurídico.
- Cuando autorizan el acceso, lo hacen bajo condiciones restrictivas que son fijadas unilateralmente por el proveedor de la obra, independientemente de los derechos que la ley otorgue al autor o al público.

En general, se puede afirmar que la aplicación de la tecnología DRM al entorno de las descargas en los dispositivos móviles potencia el negocio de la venta de contenidos digitales, gracias a una base tecnológica en la que se puedan apoyar los modelos de negocio. Mediante la aplicación de esta tecnología se aseguran los derechos de propiedad de los contenidos y se acaba con la distribución de contenidos no regulada por el operador. (5)

### 1.3.1 OMA DRM

La estandarización de la tecnología DRM en el entorno móvil está actualmente liderada por OMA. El proceso de estandarización tiene como objetivo homogeneizar los aspectos relacionados con el consumo controlado de contenidos digitales. Estos aspectos son:

- Permitir a los proveedores de contenidos expresar los derechos de uso sobre los contenidos digitales.
- Gestionar la previsualización de contenidos DRM.
- Evitar que los contenidos DRM sean distribuidos ilegalmente a otros usuarios.
- Permitir la superdistribución de contenidos DRM.
- Definir el procedimiento de autenticación de los agentes DRM.
- Definir mecanismos para el empaquetado y transferencia de derechos y contenidos protegidos.

OMA DRM se centra fundamentalmente en el control de copia, proporcionando mecanismos básicos de gestión de las claves para desbloquear la copia de contenido, haciendo especial hincapié en la seguridad de la protección. Los mecanismos de protección anti-copia definidos son:

- Bloqueo de Envío (*Forward Lock*). Evita el envío de contenido a otro dispositivo deshabilitando las opciones de envío para el contenido protegido en el propio teléfono.
- Distribución Combinada (*Combined Delivery*). Se diferencia del mecanismo Bloqueo de Envío en como se gestiona el bloqueo. En este caso, el contenido se distribuye con una clave de desbloqueo específica y única para un dispositivo.
- Distribución Separada (*Separate Delivery*). En este caso, la clave de desbloqueo se distribuye separadamente del propio contenido, lo que permite ser utilizado como confirmación de la descarga. (5)

## *1.4 Entorno de desarrollo integrado*

IDE o entorno de desarrollo integrado es un programa compuesto por un conjunto de herramientas para un programador. Puede dedicarse a uno o a varios lenguajes de programación tales como C++, C#, Visual Basic, Java, entre otros.

### 1.4.1 Eclipse Ganymede

Eclipse Ganymede es una versión del conjunto de herramientas de desarrollo de *software* del proyecto Eclipse, el cual representa un entorno de desarrollo integrado de código abierto. Dicho IDE ha alcanzado un grado de madurez, así como más robustez y rendimiento. Posee herramientas para desarrollar aplicaciones de escritorio y Web además de Servicios Web utilizando diferentes servidores como el Tomcat. (6)

Con esta nueva versión todos los elementos relacionados con la instalación de *plugins* han sido mejorados notablemente lo que lo hace más extensible, además se instalan nuevos componentes existiendo una mejor ayuda para su utilización. (6)

## *1.5 Plataforma de desarrollo: Java EE*

Java EE (por sus siglas en inglés: *Java Platform, Enterprise Edition*), antes J2EE, es un estándar para el desarrollo de aplicaciones empresariales (portables, robustas, escalables y seguras), utilizando tecnología Java. (7). Esta plataforma facilita el desarrollo de aplicaciones multicapa como; presentación, negocio y acceso a datos. Generalmente requiere de tres componentes de *software* fundamentales en el servidor:

- Contenedor de *servlets.*
- Contenedor de *Web Services.*
- Contenedor de *enterprise* java *beans* (EJB). (8)

## 1.6 Lenguaje de programación: Java

Java es toda una tecnología orientada al desarrollo de *software.* Hoy en día, esta tecnología ha cobrado mucha importancia en el ámbito de Internet gracias a su plataforma Java EE. Está compuesta básicamente por dos elementos: el lenguaje Java y la máquina virtual de Java (*Java Virtual Machine*). (9)

Una de las principales características que ha favorecido el crecimiento y difusión del lenguaje Java es su capacidad de que el código funcione sobre cualquier plataforma de *software* y *hardware* que cuente con una implementación de la maquina virtual. (9). Este lenguaje de programación está entre los más reconocidos en el desarrollo de aplicaciones para dispositivos móviles, por lo que se convierte en un firme candidato para el desarrollo del módulo, además de que posee:

- **Seguridad:** Java ofrece un entorno de ejecución seguro para programas con acceso a red. La máquina virtual de Java lleva a cabo una verificación estricta del código antes de la ejecución, asegurando que éste no trate de saltarse las protecciones impuesta por el lenguaje, utilizar punteros que accedan directamente a memoria o usar el objeto equivocado.
- **Código reutilizable**: Debido a la orientación a objetos de Java, se consiguen características como la facilidad en el desarrollo, la reutilización y la mayor calidad del código. (10)
- **Orientado a objetos:** Lo cual lo hace muy útil para la representación de entidades tal y como las crean las personas.
- **Multihilo:** Lo que permite la realización de muchas actividades simultáneas en un programa.

## 1.7 Frameworks

Para el desarrollo de *software* en ocasiones se utilizan *Frameworks,* debido a que constituyen estructuras de soporte definidas mediante la cual otros proyectos de *software* pueden ser organizados y desarrollados. Son diseñados con el intento de facilitar el desarrollo del producto, permitiendo a los diseñadores e implementadores concentrarse en aspectos de mayor nivel.

### 1.7.1 Spring

*Spring* es un *Frameworks* de código abierto de desarrollo de aplicaciones para la plataforma Java que ha ido revolucionando la manera de programar promoviendo buenas prácticas de diseño e implementación. El mismo se encarga de construir clases una vez que se desee, inyectará las dependencias que ella contenga, construyendo los objetos según sea necesario y de esta forma facilitar el desarrollo.

Ofrece varias opciones de configuración de su aplicación, la más utilizada radica en el uso de ficheros XML; generando notables ventajas, entre las que existe, que una vez diseñado y puesto en producción es posible extender el XML con la adición de nuevas etiquetas, de modo que se pueda continuar utilizando sin complicación, es sencillo de entender su estructura y procesarla. (11)

Una de las grandes ventajas que posee Spring es que está diseñado como una serie de módulos que pueden trabajar independientemente uno de otro. Además intenta mantener un mínimo acoplamiento entre la aplicación y el propio *framework* de forma que podría ser desvinculada de él sin mucha dificultad. (12)

## *1.8 Servidor Web: Apache Tomcat*

Apache Tomcat es un servidor web fácil de obtener desde Internet debido en mayor medida a que es totalmente libre; ocasionado por tal motivo su amplia extensión para el uso de los desarrolladores de software. Los requisitos de software para la utilización de este servidor recaen en la necesidad de disponer de la Maquina Virtual de Java para su adecuado funcionamiento. (13).Resulta realmente sencillo de instalar, con pocos requerimientos de capacidad en disco y compatible con las APIs más recientes de Java.

Tomcat resalta por su fiabilidad, debido a ello innumerables empresas lo utilizan aunque resulta imposible contabilizar su número exactamente. Cuenta con el trabajo de miles de desarrolladores que contribuyen con su código y ponen a disposición de toda la comunidad las últimas actualizaciones. Desplegar aplicaciones para Tomcat se hace en relativamente poco tiempo, el resultado de probar la aplicación web con este servidor asegura de que también pueda desplegarse en otros servidores de aplicaciones web.

(14)

## 1.9 Metodología de desarrollo de software: Rational Unified Process (RUP)

RUP es un proceso de desarrollo de *software* y junto con el Lenguaje Unificado de Modelado (UML), constituye una metodología estándar utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. Posee una forma disciplinada de asignar tareas y responsabilidades que definen quién hace qué, cuándo y cómo. Se caracteriza por estar dirigido por casos de uso, centrado en la arquitectura y por ser iterativo e incremental. Su objetivo es garantizar que la producción de *software* sea de alta calidad y que satisfaga las necesidades de sus usuarios finales, dentro de un calendario y con el presupuesto previsible. (15)

RUP divide el proceso de desarrollo en ciclos, teniendo un producto final al terminar cada ciclo, estos se dividen en fases que finalizan con un hito donde se debe tomar una decisión importante:

- Inicio: Se definen los objetivos y alcance del proyecto.
- Elaboración: Se define la arquitectura del proyecto.
- Construcción: Se enmarca en la elaboración de un producto totalmente operativo.
- Transición: Obtener el release del producto.

Al utilizar RUP como metodología de desarrollo de *software* se podrá verificar continuamente la calidad del mismo desde el propio comienzo del desarrollo, mitigando de forma temprana posibles riesgos. (16)

## 1.10 Lenguaje de Modelado UML

UML (por sus siglas en inglés *Unified Modeling Language*) es el lenguaje de modelado de sistemas de *software* que permite modelar, construir, visualizar y documentar los elementos que conforman un sistema orientado a objetos. Se puede aplicar en el desarrollo de *software* entregando gran variedad de formas para dar soporte a una metodología de desarrollo como el Proceso Unificado Racional (RUP), pero no especifica en sí mismo qué metodología o proceso utilizar (17)

## 1.11 Visual Paradigm

Visual Paradigm es una herramienta CASE (por sus siglas en inglés *Computer Assisted Software Engineering*) que facilita el modelado de artefactos en un Proceso de Desarrollo de *Software* mediante el Lenguaje de Modelado UML. Soporta ingeniería inversa, generación de código, importa proyectos de *Rational Rose*, genera informes, edita detalles de Casos de Uso, genera bases de datos permitiendo la transformación de diagramas de Entidad-Relación en tablas de base de datos. El *software* de modelado UML ayuda a una más rápida construcción de aplicaciones de calidad, mejores y a un menor costo. Permite dibujar todos los tipos de diagramas de clases, código inverso, y generar código desde diagramas. (18)

La herramienta es colaborativa, es decir, soporta múltiples usuarios trabajando sobre el mismo proyecto; genera la documentación del proyecto automáticamente en varios formatos como Web o Pdf y permite control de versiones. Cabe destacar su robustez, usabilidad y portabilidad. (19)

## 1.12 Sistema de control de versiones: SVN

SVN (por sus siglas en inglés *Subversion*) es un sistema de control de versiones, que mantiene los registros de todos los cambios que se han realizado a los archivos de un *software*, lo que permite el trabajo de distintos desarrolladores en un mismo proyecto, está herramienta es usada por los programadores de *software*. (20)

Existen dos razones fundamentales para el uso de esta herramienta:

- Gestiona las modificaciones durante el desarrollo.
- Permite que varias personas trabajen sobre los mismos ficheros.

El cliente a utilizar es TortoiseSVN, donde el estado de cada carpeta y fichero versionado se indica por pequeños iconos y de esta forma se puede ver fácilmente el estado en el que se encuentra la copia de trabajo.

## 1.13 Servicios Web

Un servicio web es un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones. Distintas aplicaciones de software desarrolladas en lenguajes de programación diferentes, y ejecutadas sobre cualquier plataforma, pueden utilizar los servicios web para intercambiar datos en redes. La interoperabilidad se consigue mediante la adopción de estándares abiertos.

Con el objetivo de establecer la comunicación entre subsistemas se pretende emplear servicios web en caso de requerirse; garantizando para ello la seguridad a través de WS-Security, protocolo este que garantiza la autenticación de los actores y la confidencialidad de los mensajes enviados. Empleando el protocolo HTTPS (por sus siglas en inglés *Hypertext Transfer Protocol Secure*) para la transmisión segura de los datos.

## 1.14 Conclusiones

El capítulo ha sido dedicado a la selección de las herramientas y metodologías adecuadas para el desarrollo de un módulo que gestione la entrega de contenidos hacia celulares de manera eficiente; seleccionando *OMA Download* como método para satisfacer dicho objetivo y OMA DRM para la gestión de derechos digitales.

# Capítulo 2. Características del sistema

## *2.1 Introducción*

En el presente capítulo se describe la problemática existente, así como las características del sistema a desarrollar. Debido a la poca definición de los procesos del negocio se manifiestan los principales conceptos tratados y las relaciones existentes entre ellos mediante un Modelo de Dominio o Modelo Conceptual. Además se plantean los requerimientos que el sistema debe cumplir, los casos de uso y los actores del mismo.

## *2.2 Objeto de estudio*

### 2.2.1 Definición del problema

Actualmente la empresa Cubacel, operadora de telefonía celular en Cuba se ha propuesto el desarrollo de un portal WAP para brindar algunos servicios a sus clientes como noticias nacionales e internacionales, partes meteorológicos para las zonas de interés y previsiones para los próximos días, y directorio telefónico de instituciones.

Otro de los servicios que se proponen brindar son los de ocio como es el caso de la descarga de tonos, logos y melodías que el cliente puede comprar. Por tanto se requiere de un mecanismo que aplique el derecho de autor (por sus siglas en inglés DRM; *Digital Rights Management*) hacia estos contenidos; no tenerlo podría ocasionar perdidas económicas una vez realizado el servicio de descarga, debido a que desde el propio instante en que uno de sus usuarios obtenga el producto deseado, puede ser facilitado gratuitamente a otros clientes construyendo de esta forma una cadena de transmisión no factible para la empresa.

Por lo planteado anteriormente surge la siguiente interrogante: ¿Cómo realizar la entrega de contenido y gestión de derechos de autor dentro de la Plataforma de Gestión de Contenidos?

### 2.2.2 Objeto de automatización

Se procederá a la creación de un módulo que gestione de manera eficiente la forma en que se entrega a un cliente el contenido solicitado, además de gestionar el derecho de autor aplicando DRM a estos contenidos.

### 2.2.3 Información que se maneja

La información está relacionada con los contenidos que se manejan por la plataforma y las solicitudes realizadas por los teléfonos celulares como expresión concreta de los clientes, dichas solicitudes poseen:

- Número de teléfono del cliente.
- Fecha de inicio de la solicitud.
- Identificador del contenido que solicitó.
- *User Agent* del teléfono (Identificador del modelo de celular).

## *2.3 Propuesta del sistema*

La Plataforma de Gestión de Contenidos para Dispositivos Móviles estará compuesta por diferentes módulos, entre ellos:

- Portal WAP
- Entrega de Contenido
- Contenido
- Mensajería
- Pago

Dichos módulos se encargarán de gestionar las operaciones asociadas a los contenidos desde la solicitud de un contenido mediante el Portal WAP, hasta su posterior cobro, pasando por procesos que garanticen la descarga y adaptación del contenido adecuándolo a las características del dispositivo móvil (celular).

El presente documento recoge la importancia del Módulo Entrega de Contenido para garantizar el buen

funcionamiento del sistema. El mismo funcionará como un intermediario entre los restantes módulos al poder recibir la solicitud mediante el Portal WAP, enviándole al Módulo de Contenido la orden para ser registrada, y mediante el Módulo de Mensajería le enviará al cliente la URL. En la cual puede encontrarse el descriptor (archivo XML) que posee los datos referentes a la entrega. Una vez que el cliente obtenga el descriptor, podrá acceder a descargar el contenido, por seguridad, el módulo se encargará de verificar que el cliente que está accediendo al mismo es el que lo solicitó; además se comprobará que el tiempo que tardó en hacerlo no supera el tiempo máximo establecido para ello. Antes de que el contenido sea entregado al cliente se le aplicaría DRM, específicamente el método *Forward Lock,* para conservar el derecho de autor, en caso de que la empresa así lo desee. Luego de realizarle la entrega se obtiene la notificación de la misma con la información referente al nivel de efectividad con que se realizó. Se deberá atender la notificación para proceder al cobro del contenido mediante el Módulo de Pago y dar como vendida la solicitud en caso de éxito o esperar hasta que el contenido sea descargado correctamente.

## *2.4 Modelo de Dominio*

Cuando en el entorno del negocio los procesos tienen un bajo nivel de estructuración y no existen flujos de información interconectados y bien definidos, se genera la imposibilidad de llevar a cabo un Modelo de Negocio, haciéndose necesario la realización de un Modelo de Dominio.

El Modelo de Dominio, permite mostrar los principales conceptos que se manejan para el dominio del sistema en desarrollo. El diagrama del Modelo de Dominio se presenta en forma de diagrama de clases donde figuran los principales conceptos y roles del sistema analizado.

**Conceptos:**

Módulo de Contenido: Módulo que contiene la base de datos de los contenidos que el cliente desea descargar como tonos, logos y melodías.

Cliente: Teléfono celular que realiza la solicitud del contenido que desea y luego lo descarga.

Descriptor: Archivo XML que posee los datos referentes a la entrega del contenido al cliente que solicitó, por ejemplo dirección URL donde se le entregará el contenido, dirección URL para notificar el éxito de la

entrega, el tamaño del contenido en bytes, entre otros.

DRM: Seguridad que se aplica a los contenidos para conservar el derecho de autor.

Módulo de Entrega de Contenido: Encargado de gestionar el proceso de entregar el contenido al cliente.

Portal WAP: Portal donde el cliente se conecta para realizar la solicitud del contenido que desea adquirir.

Módulo de Pago: Encargado de realizar todos los procesos relacionados con el cobro del contenido que el cliente descargó.

Módulo de Mensajería: Encargado de enviar al cliente un SMS con la URL donde debe descargar el descriptor.

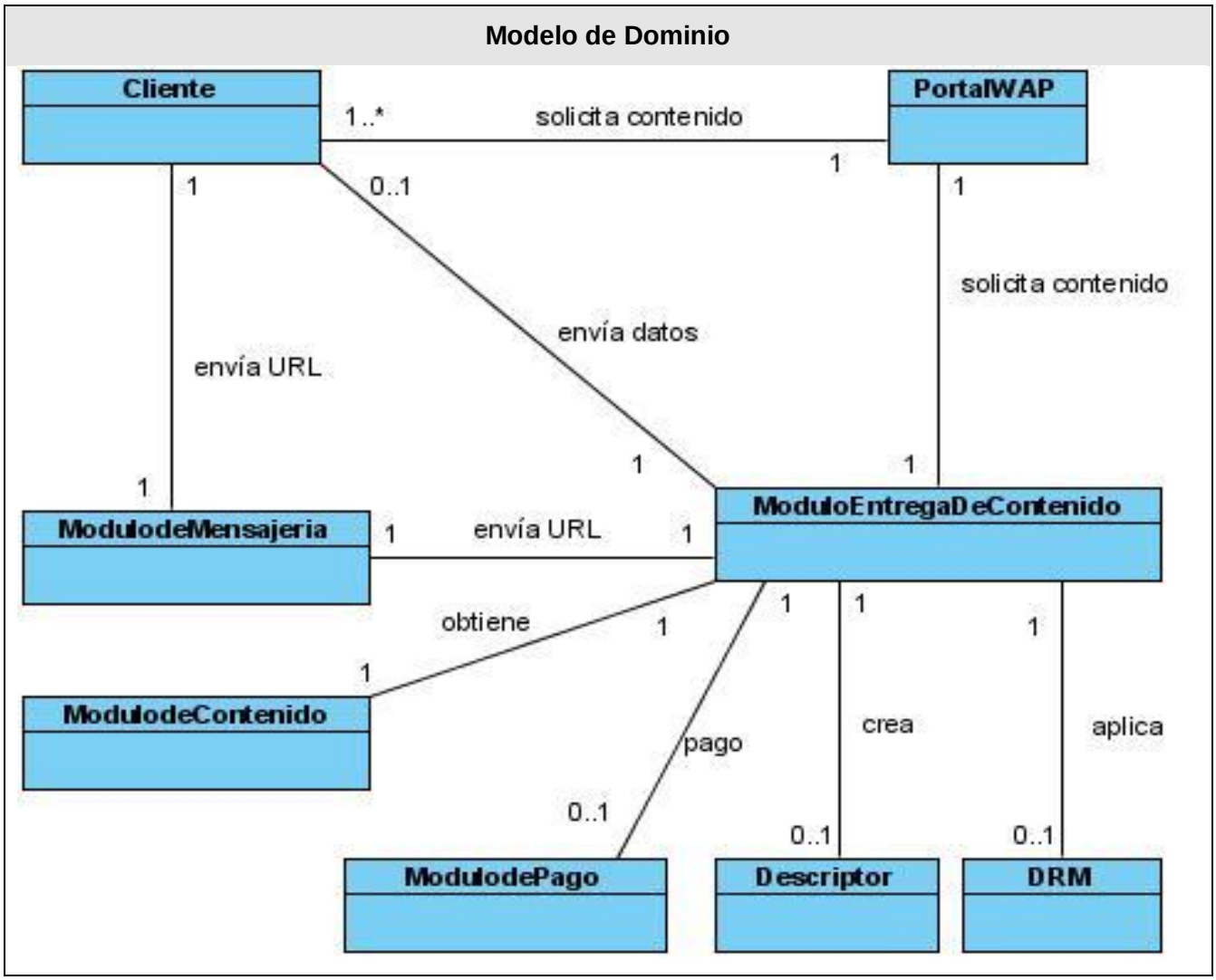



**Figura 1: Modelo de Dominio**

## *2.5 Especificación de los requisitos de software*

Con la especificación de los requisitos de *software* se podrá definir el ámbito del sistema, así como establecer y mantener un acuerdo entre clientes y otros involucrados sobre lo que el sistema debería hacer. Estos requisitos brindan una base para estimar recursos y tiempo de desarrollo del sistema. (21)

### 2.5.1 Requerimientos funcionales

Los requerimientos funcionales van a ser las capacidades o condiciones que el sistema debe cumplir sin alterar la funcionalidad del producto.

**RF1** Atender solicitud del contenido para un cliente.

| **RF1.1** | Registrar datos de la solicitud. |
|---|---|
| Descripción | El módulo debe enviar la solicitud al Módulo de Contenido para ser registrada en la base de datos. |

**Tabla 2: Descripción del requerimiento funcional Registrar datos de la solicitud.**

| **RF1.2** | Enviar URL al cliente. |
|---|---|
| Descripción | El módulo debe crear y enviar al cliente mediante el Módulo de Mensajería la URL donde puede descargar el descriptor que posee los datos de la descarga de tipo *OMA Download.* |

**Tabla 3 Descripción del requerimiento funcional Enviar URL al cliente.**

**RF2** Enviar descriptor al cliente.

| **RF2.1** | Crear descriptor. |
|---|---|
| Descripción | El módulo debe crear el descriptor que posee los datos de la descarga de tipo *OMA Download.* |

**Tabla 4 Descripción del requerimiento funcional Crear Descriptor.**

| **RF2.2** | Enviar Descriptor. |
|---|---|
| Descripción | El módulo debe entregar al cliente el descriptor que posee los datos de la descarga de tipo *OMA Download*. |

**Tabla 5 Descripción del requerimiento funcional Enviar Descriptor.**

**RF3** Atender solicitud de descarga.

| **RF3.1** | Buscar contenido. |
|---|---|
| Descripción | El módulo debe solicitar al Módulo de Contenido el contenido que el cliente desea descargar. |

**Tabla 6 Descripción del requerimiento funcional Buscar contenido.**

| **RF3.2** | Entregar contenido al cliente. |
|---|---|
| Descripción | El módulo debe hacer llegar al cliente el contenido que solicitó. |

**Tabla 7 Descripción del requerimiento funcional Entregar contenido al cliente.**

**RF4** Atender notificación de descarga.

| **RF4** | Atender notificación de descarga. |
|---|---|
| Descripción | El módulo debe obtener la notificación que envía el cliente una vez que se realice la entrega del contenido con el objetivo de conocer el nivel de éxito con que se realizó la misma para proceder al cobro en caso de ser positivo este resultado. |

**Tabla 8 Descripción del requerimiento funcional Atender notificación de descarga.**

**RF5** Aplicar DRM al contenido.

| **RF5.1** | Aplicar método Forward Lock. |
|---|---|
| Descripción | El módulo debe aplicar DRM para conservar el derecho de autor. |

**Tabla 9 Descripción del requerimiento funcional Aplicar método Forward Lock.**

**RF6** Expirar solicitudes.

| **RF6.1** | Eliminar la solicitud expirada. |
|---|---|
| Descripción | El módulo debe a través del Módulo de Contenido eliminar las solicitudes registradas cuyo |

| | |
|---|---|
| | tiempo esperando a ser descargada, supera al tiempo establecido por la empresa para realizar esta operación. |

**Tabla 10 Descripción del requerimiento funcional Expirar solicitudes.**

### 2.5.2 Requerimientos no funcionales

Los requisitos no funcionales son propiedades o cualidades que debe tener el producto. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable. (21)

#### 2.5.2.1 Soporte

El soporte es un aspecto realmente amplio y abarca todas las acciones a tomar una vez que se ha terminado el desarrollo del software con motivos de asistir a los clientes de este, así como lograr su mejoramiento progresivo y evolución en el tiempo. Puede incluir: pruebas, adaptabilidad, mantenimiento, compatibilidad, configuración, servicios, instalación y portabilidad, por solo mencionar algunos.

**RNF** El sistema contará antes de su puesta en marcha con un período de pruebas que permitirá identificar y corregir errores cometidos durante el desarrollo, posteriormente se brindará el servicio de instalación del producto.

**RNF** Se generará la documentación apropiada para agilizar su mantenimiento y configuración.

#### 2.5.2.2 Seguridad

Este es quizás el tipo de requerimiento más difícil, que provocará los mayores riesgos si no se maneja correctamente.

**RNF** El sistema debe ser capaz de permitir que acceda solamente al contenido el cliente que lo solicitó.

**RNF** Se garantizará el tratamiento adecuado de las excepciones y validaciones de las entradas del usuario.

#### 2.5.2.3 Hardware

Los requisitos de hardware están asociados fundamentalmente a la definición específica que determine el proyecto para su despliegue; debido en gran medida a la presencia en el mismo de varios módulos o subsistemas, cada uno con condiciones propias.

#### 2.5.2.4 Software

**RNF** Se requiere la presencia del servidor Web Apache Tomcat.

**RNF** El sistema necesita la máquina virtual de java para su funcionamiento.

**RNF** Sistema multiplataforma.

## *2.6 Modelo del sistema*

### 2.6.1 Actores del sistema

| Actores | Justificación |
|---|---|
| Portal WAP | Realiza la solicitud del contenido que el cliente desea. |
| Módulo de Contenido | Encargado de gestionar toda la información referente a la orden (solicitud). |
| Cliente | Obtiene el contenido solicitado. |
| Factor Tiempo | Gestiona que se eliminen las solicitudes que el tiempo de espera es mayor al tiempo establecido para la descarga. |
| Módulo de Mensajería | Encargado de enviar al cliente la URL donde se encuentra el descriptor. |
| Módulo de Pago | Encargado de cobrar el contenido al cliente. |

**Tabla 11: Actores del sistema**

### 2.6.2 Casos de Uso del sistema

| CU-1 | Atender solicitud |
|---|---|
| **Actor** | Portal WAP |
| **Descripción** | Luego de recibir la solicitud del contenido que el cliente desea, si al mismo hay que aplicarle DRM y el celular no lo soporta se le envía un SMS al cliente informándole que no puede obtener el contenido, de no ser así se guardan los datos referentes a la solicitud, y se le envía un mensaje con la URL donde puede acceder a la descarga del descriptor que contiene los datos referentes al contenido. |
| **Referencia** | RF1 |

**Tabla 12: Descripción general del caso de uso Atender solicitud**

| CU- 2 | Entregar contenido. |
|---|---|
| **Actor** | Cliente |
| **Descripción** | Luego del cliente acceder a la URL obtenida se le envía el descriptor que contiene los datos referentes al contenido que solicitó, se confirma la opción descargar, se busca el contenido correspondiente a este, se le aplica DRM en caso de que se deba y se le entrega. Si la descarga fue con éxito, entonces se cobra el contenido solicitado al cliente y se eliminan los datos de la orden. |
| **Referencia** | RF2, RF3, RF4 |

**Tabla 13: Descripción general del caso de uso Entregar contenido**

| CU- 3 | DRM Forward Lock. |
|---|---|
| **Descripción** | Luego de especificarse que se le debe aplicar DRM al contenido solicitado, se recibe el contenido a proteger para luego armar el mensaje DRM. |
| **Referencia** | RF5 |

**Tabla 14: Descripción general del caso de uso DRM Forward Lock**

| CU- 4 | Expirar solicitudes. |
|---|---|
| **Actor** | Factor Tiempo. |
| **Descripción** | Cada un intervalo de tiempo se levante un proceso que gestiona que se eliminen las solicitudes que han superado el tiempo máximo establecido para descargar el contenido. |
| **Referencia** | RF6 |

**Tabla 15: Descripción general del caso de uso Expirar solicitudes**

### 2.6.3 Diagrama de Casos de Uso del sistema

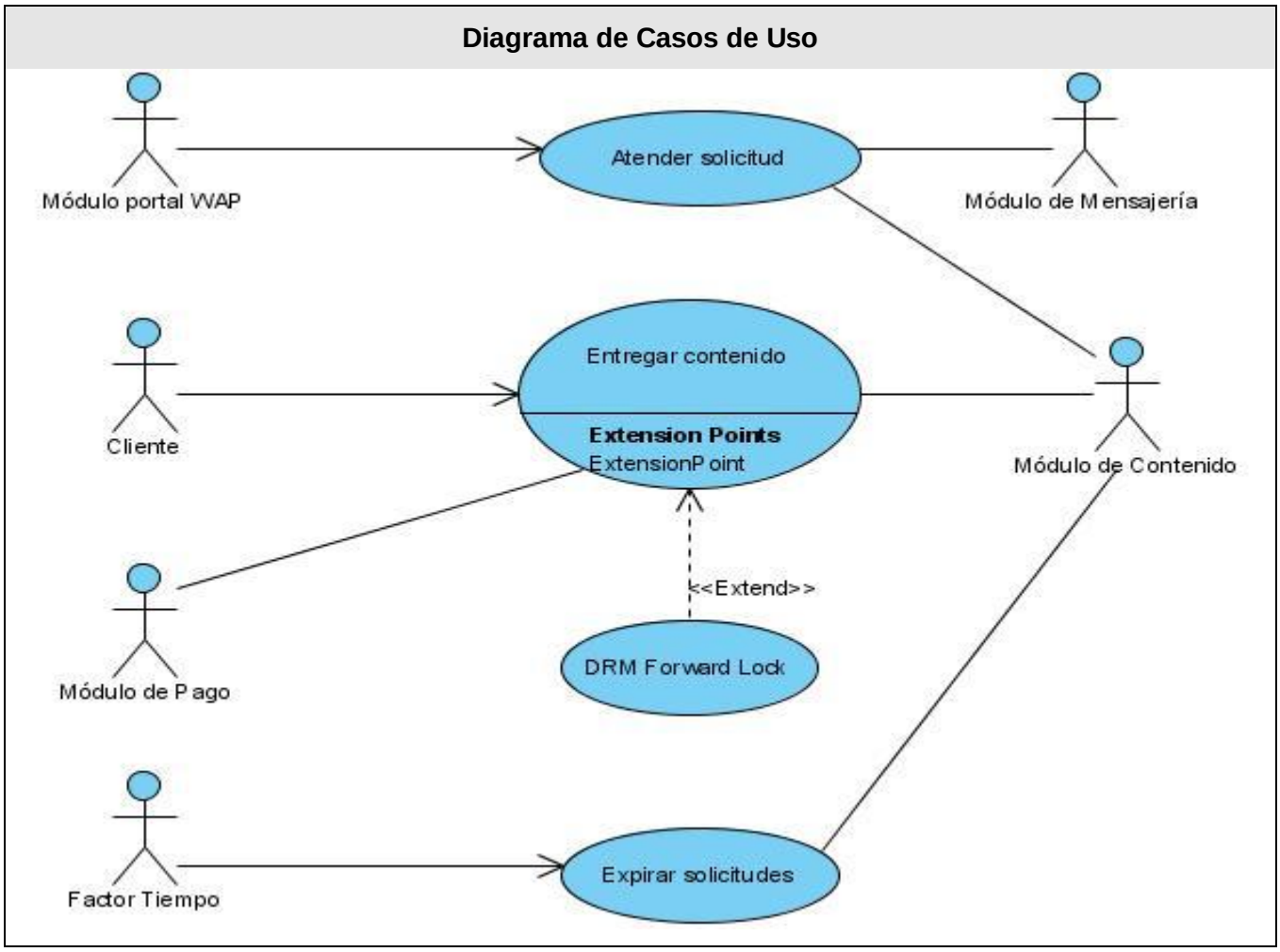



**Figura 2: Diagrama de casos de uso**

### 2.6.4 Descripción de los Casos de Uso del sistema

| **Caso de uso** | |
|---|---|
| CU-1 | Atender solicitud |
| **Propósito** | Dar a conocer la dirección donde el cliente debe descargar el descriptor |
| **Actores:** Portal WAP(inicializa), Módulo de Contenido | |
| **Resumen:** El caso de uso se inicia cuando el Portal WAP realiza la solicitud del contenido que el cliente desea. Es encargado de gestionar que se registren los datos de la solicitud, y enviar al cliente la URL donde debe descargar el descriptor. | |
| **Referencias** | RF1 |
| **Precondición** | El Portal WAP debe poseer los siguientes datos: *User agent*, número de teléfono, fecha de inicio y contenido que se debe entregar al cliente. |

| **Acción del actor** | **Respuesta del sistema** |
|---|---|
| 1) El portal WAP solicita el contenido. | 2) Obtiene los datos a la orden, tales como, *User Agent,* número de teléfono, fecha de inicio y contenido que se debe entregar al cliente. |
| | 3) Solicita al Módulo de Contenido datos referentes al contenido solicitado. |
| 4) El Módulo de Contenido envía datos del contenido. | 5) Comprueba si hay que aplicar o no DRM al contenido. |
| | 6) Registra la solicitud en la base de datos mediante el Módulo de Contenido. |
| | 7) Crea URL donde el cliente puede acceder al descriptor que posee los datos para la entrega del contenido. |
| | 8) Envía al cliente que solicitó el contenido un SMS mediante el Módulo de Mensajería con la URL del descriptor. |

| **Flujo alternativo :**<br><br>Si hay que aplicar DRM y el celular no lo soporta. | |
|---|---|
| **Acción del actor** | **Respuesta del sistema** |
| | 6) Envía al cliente un SMS mediante el Módulo de Mensajería informándole que no puede acceder al contenido solicitado. |
| **Poscondicion** | |
| El cliente debe acceder a la URL obtenida. | |

**Tabla 16: Descripción detallada del caso de uso Atender solicitud**

| **Caso de uso** | |
|---|---|
| CU-2 | Entregar contenido. |
| **Propósito** | Entregar el contenido al cliente. |
| **Actores:** Cliente (Inicializa), Módulo de Contenido. | |
| **Resumen:** El caso de uso inicia cuando el cliente accede a la URL donde se le envía el descriptor, al hacer uso de este se le entrega el contenido de una manera eficiente, se procede al cobro y se eliminan los datos referentes a la orden. | |
| **Referencias** | RF2, RF3,RF4 |
| **Precondición** | El cliente debe haber recibido la URL donde se encuentra el descriptor. |
| **Acción del actor** | **Respuesta del sistema** |
| 1) El cliente realiza la solicitud del descriptor al acceder a la URL. | 2) Con el identificador de la orden que está contenido en la URL se solicita al Módulo de Contenido la orden registrada en la Base de Datos. |
| 3) El Módulo de Contenido envía la orden solicitada. | 4) Verifica que el cliente que está accediendo es el que solicitó el contenido. |
| | 5) Solicita al Módulo de Contenido datos del contenido solicitado como nombre del contenido, |

| | |
|---|---|
| | proveedor. |
| 6) El Módulo de Contenido envía datos del contenido. | 7) Crea descriptor con los datos obtenidos además de la URL donde debe descargar y URL donde se notificará el éxito de la descarga. |
| | 8) Envía descriptor al cliente. |
| 9) El cliente accede a la URL que posee el descriptor para descargar el contenido. | 10) Solicita contenido al Módulo de Contenido. |
| 11) El Módulo de Contenido entrega el contenido. | 12) Verifica si hay que aplicar DRM al contenido. |
| | 13) Ir al caso de uso DRM Forward Look. |
| 14) El cliente envía notificación de la entrega mediante la URL de notificación del descriptor. | 15) Verifica la notificación de la entrega. |
| | 16) Procede al cobro del contenido a través del Módulo de Pago. |
| | 17) Con el identificador de la orden Informa al Módulo de Contenido que elimine los datos referentes a esta. |
| **Flujo alternativo:**<br><br>Si el cliente que está accediendo no fue el que solicitó el contenido: | |
| **Acción del actor** | **Respuesta del sistema** |
| | 5) Informa al cliente. |
| **Flujo alternativo:**<br><br>Si no hay que aplicar DRM al contenido: | |
| **Acción del actor** | **Respuesta del sistema** |
| | 13) Entrega el contenido al cliente. |
| 14) El cliente envía notificación de la entrega mediante la URL de notificación del descriptor. | 15) Verifica la notificación de la entrega. |

| | |
|---|---|
| | 16) Procede al cobro del contenido a través del Módulo de Pago. |
| | 17) Con el identificador de la orden Informa al Módulo de Contenido que elimine los datos referentes a esta. |
| **Flujo alternativo:**<br><br>Si el contenido no fue entregado con éxito: | |
| **Acción del actor** | **Respuesta del sistema** |
| | 18) Espera a que se solicite nuevamente el contenido. |

**Tabla 17: Descripción detallada del caso de uso Entregar contenido**

| **Caso de uso** | |
|---|---|
| CU-3 | DRM Forward Lock. |
| **Propósito** | Aplicar el método Forward Lock. |
| **Resumen:** El caso de uso se inicia cuando el CU-2 realiza la solicitud de aplicar DRM al contenido que será entregado al cliente. Luego se conforma el mensaje para el método de entrega Forward Lock. | |
| **Referencias** | RF1 |
| **Precondición** | Debe haberse realizado la acción 13 en el CU 2. |
| **Acción del actor** | **Respuesta del sistema** |
| | 2) Recibe el contenido y lo convierte en un flujo de bytes. |
| | 3) Forma el mensaje DRM con el contenido convertido. El mensaje solo está relacionado con el contenido, nunca con sus restricciones. |

| | |
|---|---|
| | 4) Entrega el contenido con DRM al cliente. |

**Tabla 18: Descripción detallada del caso de uso DRM Forward Lock**

| **Caso de uso** | | |
|---|---|---|
| CU-4 | Expirar solicitudes. | |
| **Propósito** | Gestiona que se eliminen las solicitudes que hayan superado el tiempo límite para descargar. | |
| **Actores:** Factor Tiempo (inicializa), Módulo de Contenido | | |
| **Resumen:** El caso de uso inicia cuando el Factor Tiempo solicita que se compruebe la expiración de las órdenes registradas, la orden que su tiempo de sea superior al establecido para descargar el contenido será eliminada. | | |
| **Referencias** | RF5 | |
| **Precondición** | Debe haber transcurrido el tiempo establecido para comprobar las órdenes que han expirado. | |
| **Acción del actor** | | **Respuesta del sistema** |
| 1) El Factor Tiempo solicita verificación del tiempo de expiración. | | 2) Solicita al Módulo de Contenido que le entregue las órdenes registradas. |
| 3) El Módulo de Contenido entrega las órdenes registradas. | | 4) Se calcula el tiempo transcurrido desde que se registró la orden hasta la actualidad. |
| | | 5) Se verifica si el tiempo de espera transcurrido es mayor que el tiempo definido para descargar el contenido solicitado. |
| | | 6) Para cada orden que cumpla la condición anterior se solicita al Módulo de Contenido que elimine los datos referentes a la misma. |

**Tabla 19: Descripción detallada del caso de uso Expirar solicitudes**

## 2.7 Conclusiones

En este capítulo se abordaron los principales conceptos y sus relaciones representadas en un modelo de dominio. Además se presentaron los requerimientos funcionales y no funcionales que se deben cumplir, así como la propuesta del sistema. También se realizó la descripción detallada de los casos de uso y de esta forma se da paso al análisis y diseño del módulo.

# Capítulo 3. Análisis y Diseño

## *3.1 Introducción*

El objetivo fundamental a tratar en el presente capítulo es la realización de un diseño consecuente que de respuesta a los requerimientos planteados en la sección anterior. Se obtiene una visión general del sistema mediante los diagramas de clases del análisis, de clases del diseño y de interacción, así como la especificación de los patrones de diseño.

## *3.2 Modelo de Análisis*

Durante el flujo de trabajo de análisis, se analizan los requisitos que se describieron en la captura de requisitos, refinándolos y estructurándolos. El objetivo de hacerlo es conseguir una comprensión más precisa y una descripción de los mismos que sea fácil de mantener y que ayude a estructurar el sistema completo. No se tiene en cuenta el lenguaje de programación a usar en la elaboración, la plataforma en que se ejecutará la aplicación, los componentes prefabricados o reusables de otras aplicaciones, entre otras características que afectan al sistema, ya que el objetivo del análisis es comprender perfectamente los requisitos del *software* y no precisar cómo se implementará la solución. (22)

### 3.2.1 Diagramas de clases del análisis

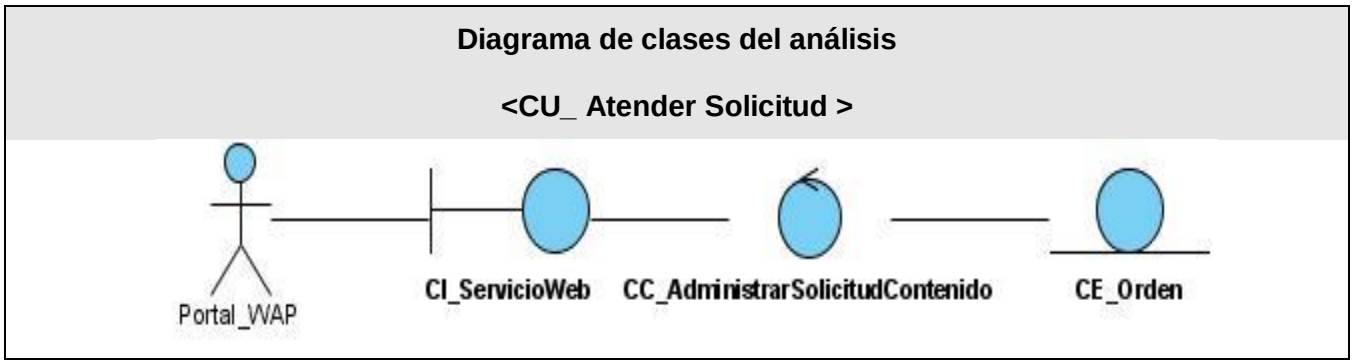



**Figura 3: Diagrama de clases del análisis del caso de uso Atender Solicitud**

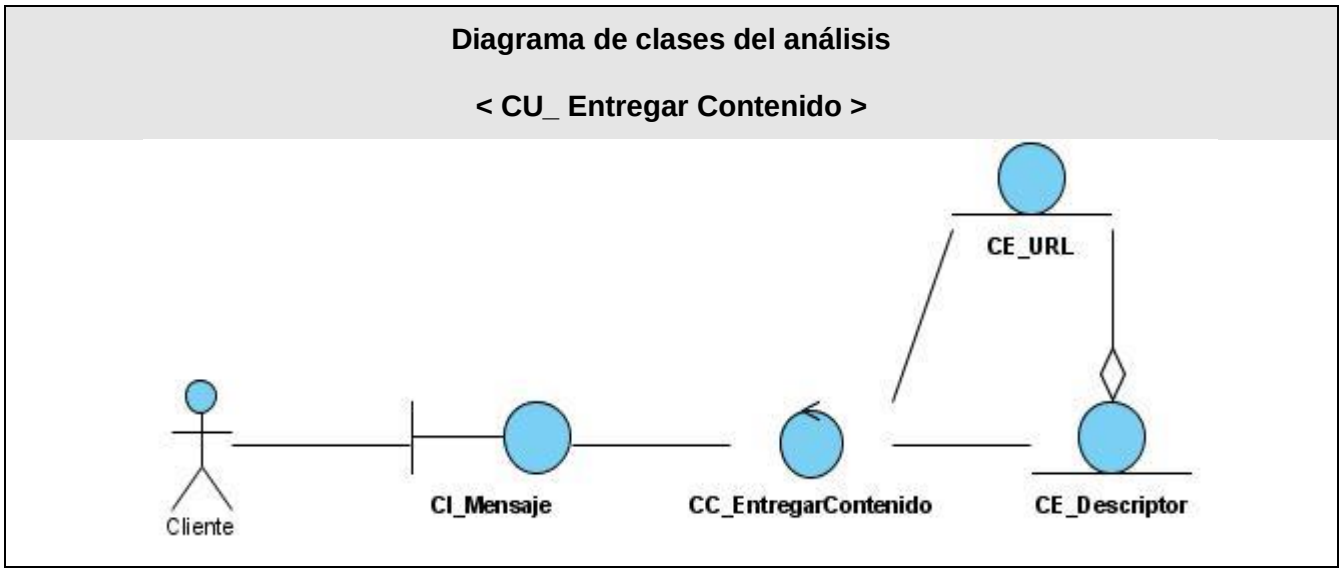



**Figura 4: Diagrama de clases del análisis del caso de uso Entregar Contenido**

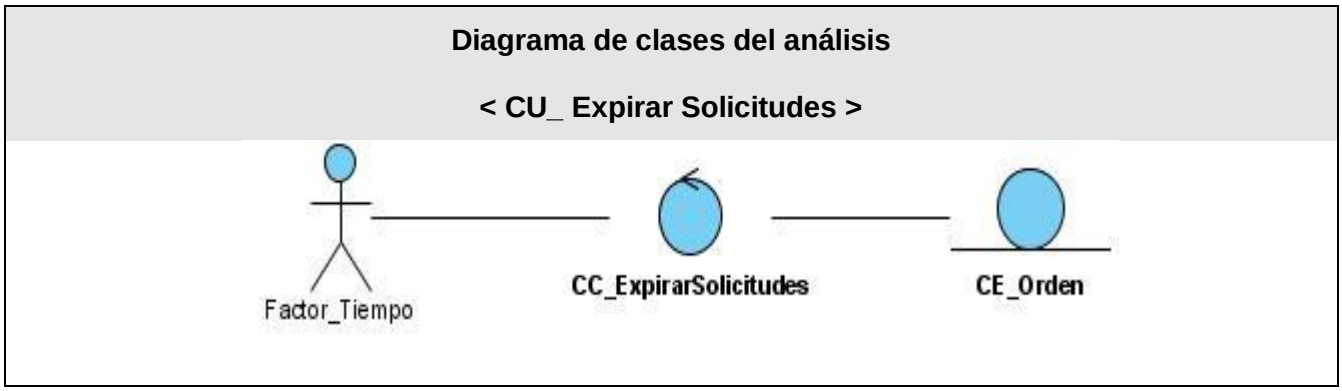



**Figura 5: Diagrama de clases del análisis del caso de uso Expirar Solicitudes**

## *3.3 Modelo de Diseño*

El modelo de diseño es un paso inicial, una primera aproximación conceptual, para una vez comprendidos los requisitos, aumentar el nivel de especificidad en aras de garantizar el cubrimiento de los requisitos funcionales y no funcionales, considerando además el entorno en el cual se realizará la futura implementación. (22)

### 3.3.1 Diagrama de clases del diseño

Los diagramas de clases del diseño describen gráficamente las especificaciones de las clases del *software* y contienen los atributos, métodos, navegabilidad y dependencias existentes entre ellas.

La siguiente tabla muestra las clases y sus relaciones de forma general resaltando con igual color las clases pertenecientes a un mismo caso de uso, para un mejor entendimiento del funcionamiento del módulo.

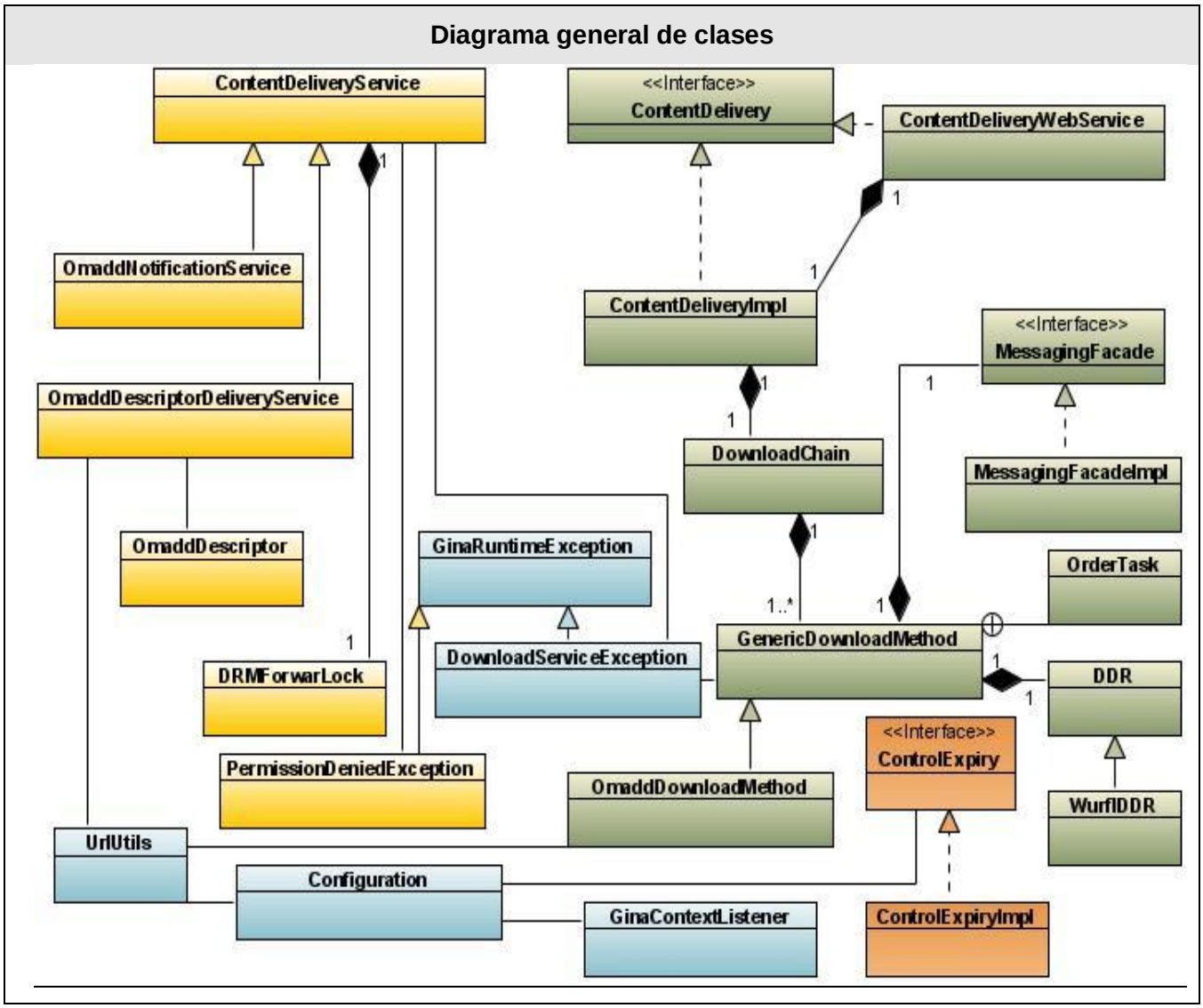



**Figura 6: Diagrama general de clases del Módulo Entrega de Contenido**

**Leyenda:**

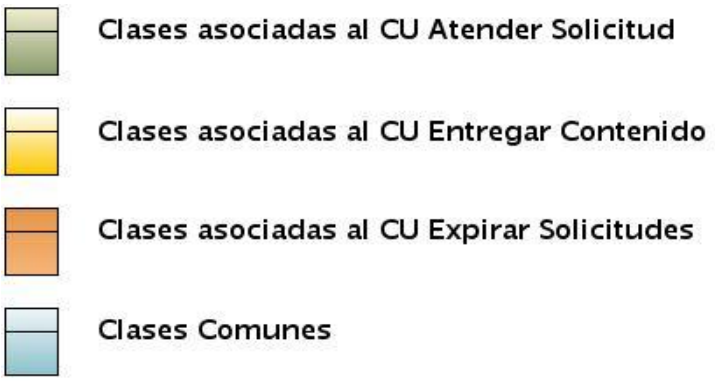



**Figura 7: Definición de los términos fundamentales**

La descripción detallada de las clases pertenecientes al diagrama de diseño de cada caso de uso podrá encontrarse en el anexo 1.

### *3.3.2 Diagramas de secuencia*

Mediante los diagramas de Secuencia se muestra la secuencia de mensajes entre objetos, resaltando el orden temporal de los mensajes que se intercambian en un escenario. A continuación se muestran los diagramas de secuencias referentes a cada caso de uso. Para lograr mejor visualización y entendimiento se procede a representar cada diagrama en fragmentos ordenados.

**Diagrama de secuencia**

**<CU_ Atender Solicitud >**

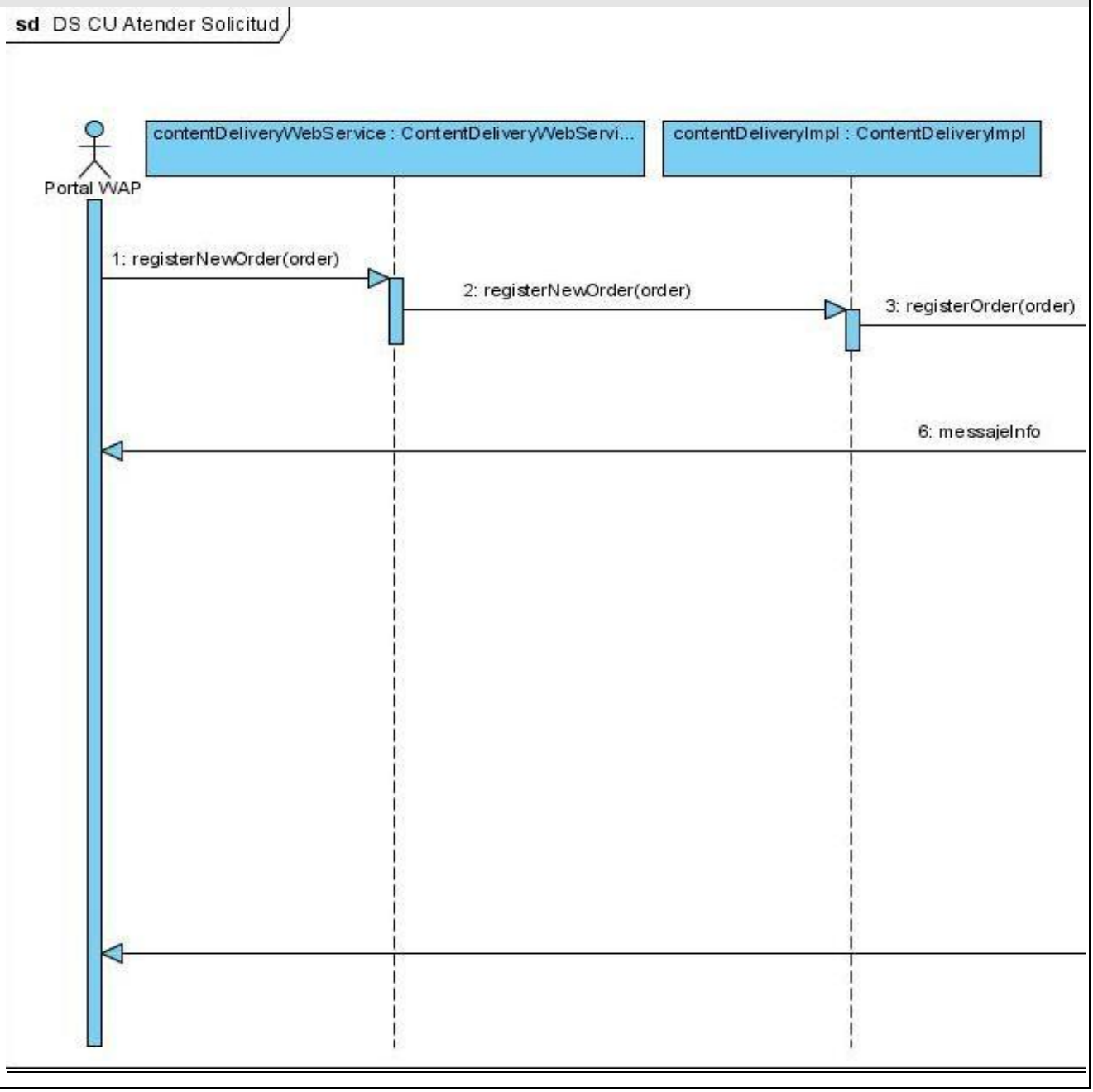



**Figura 8: Diagrama de secuencia del caso de uso Atender Solicitud**

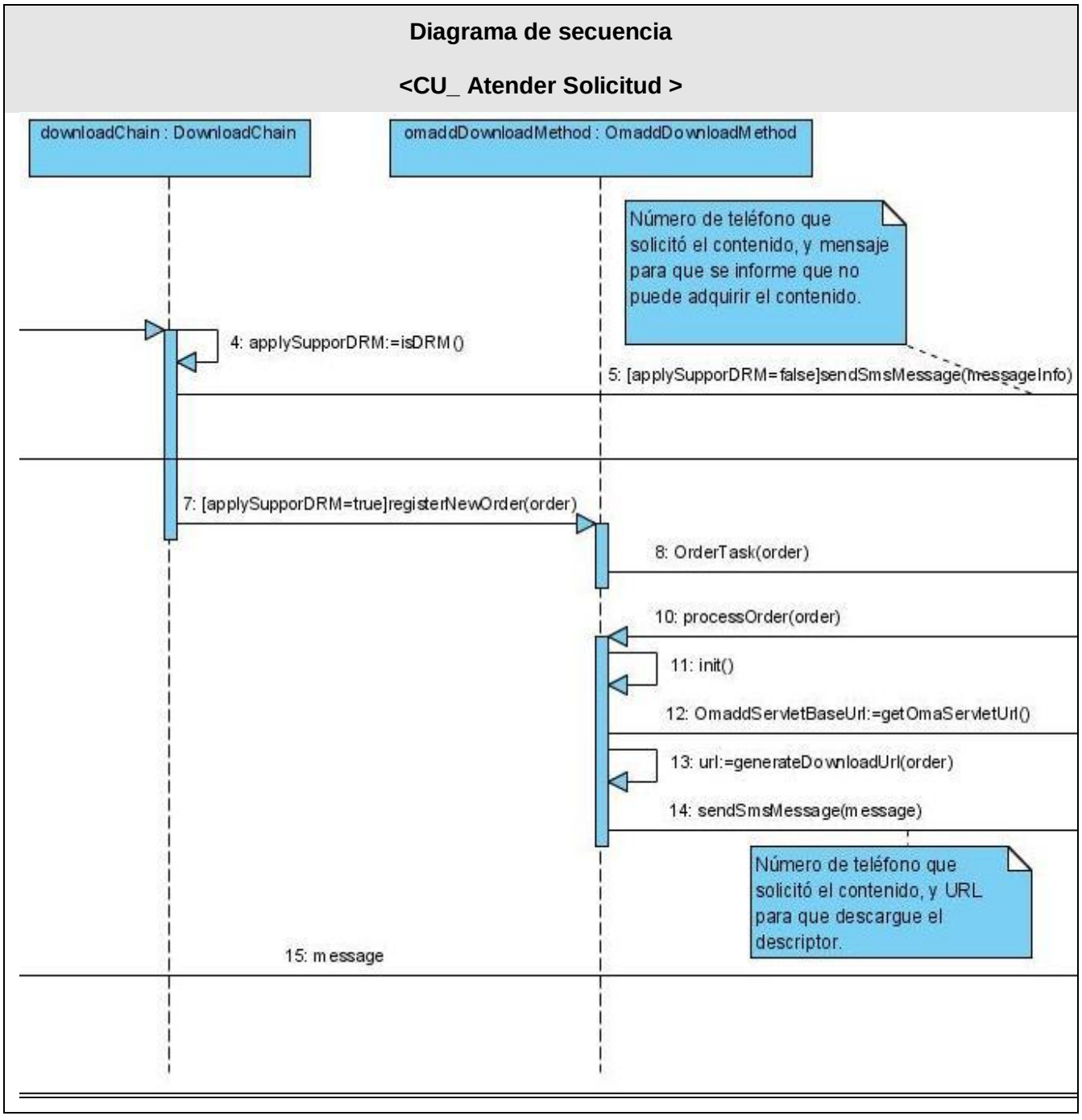



**Figura 9: Diagrama de secuencia del caso de uso Atender Solicitud**

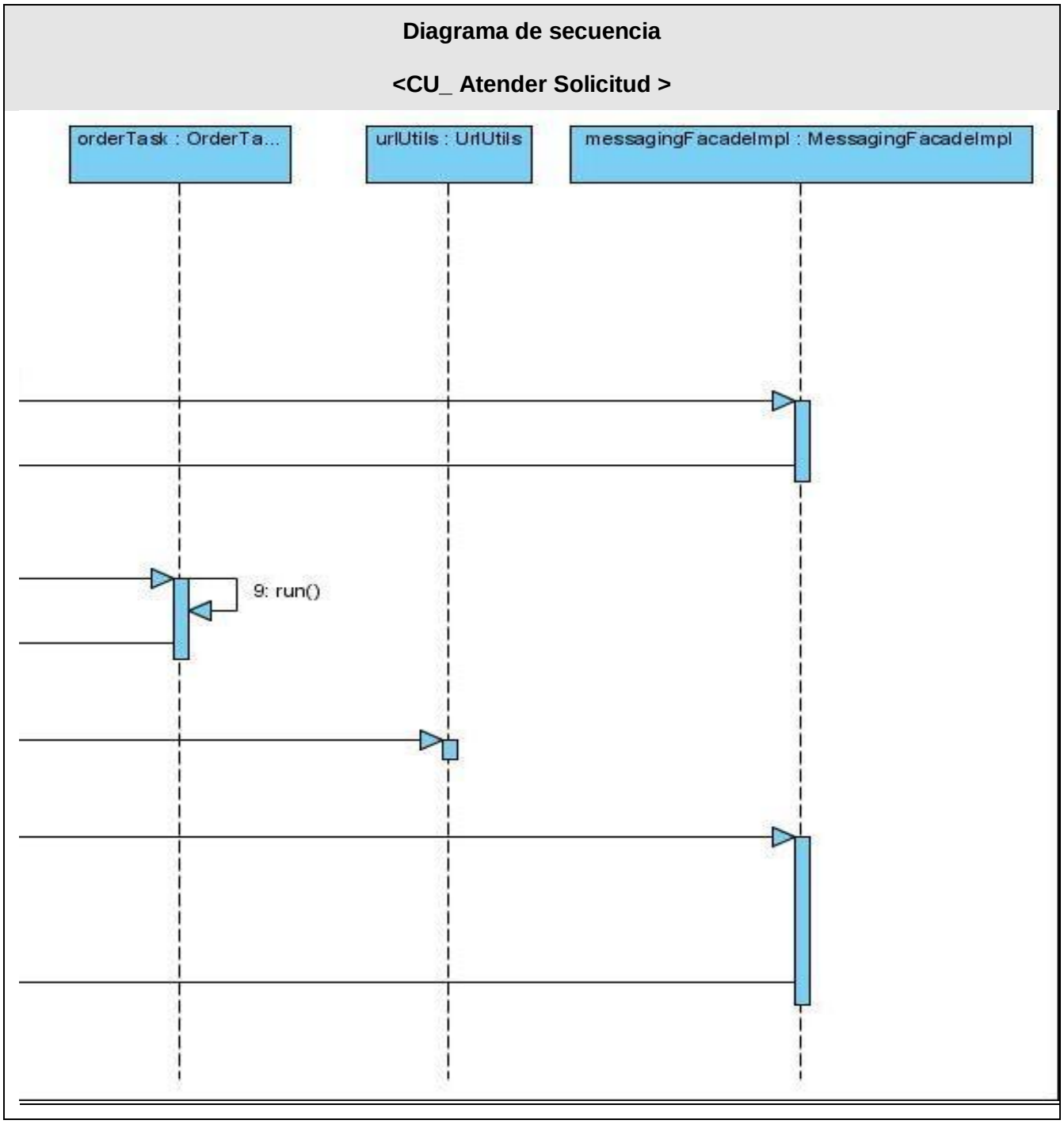



**Figura 10: Diagrama de secuencia del caso de uso Atender Solicitud**

**Diagrama de secuencia**

**<CU_ Entregar Contenido >**

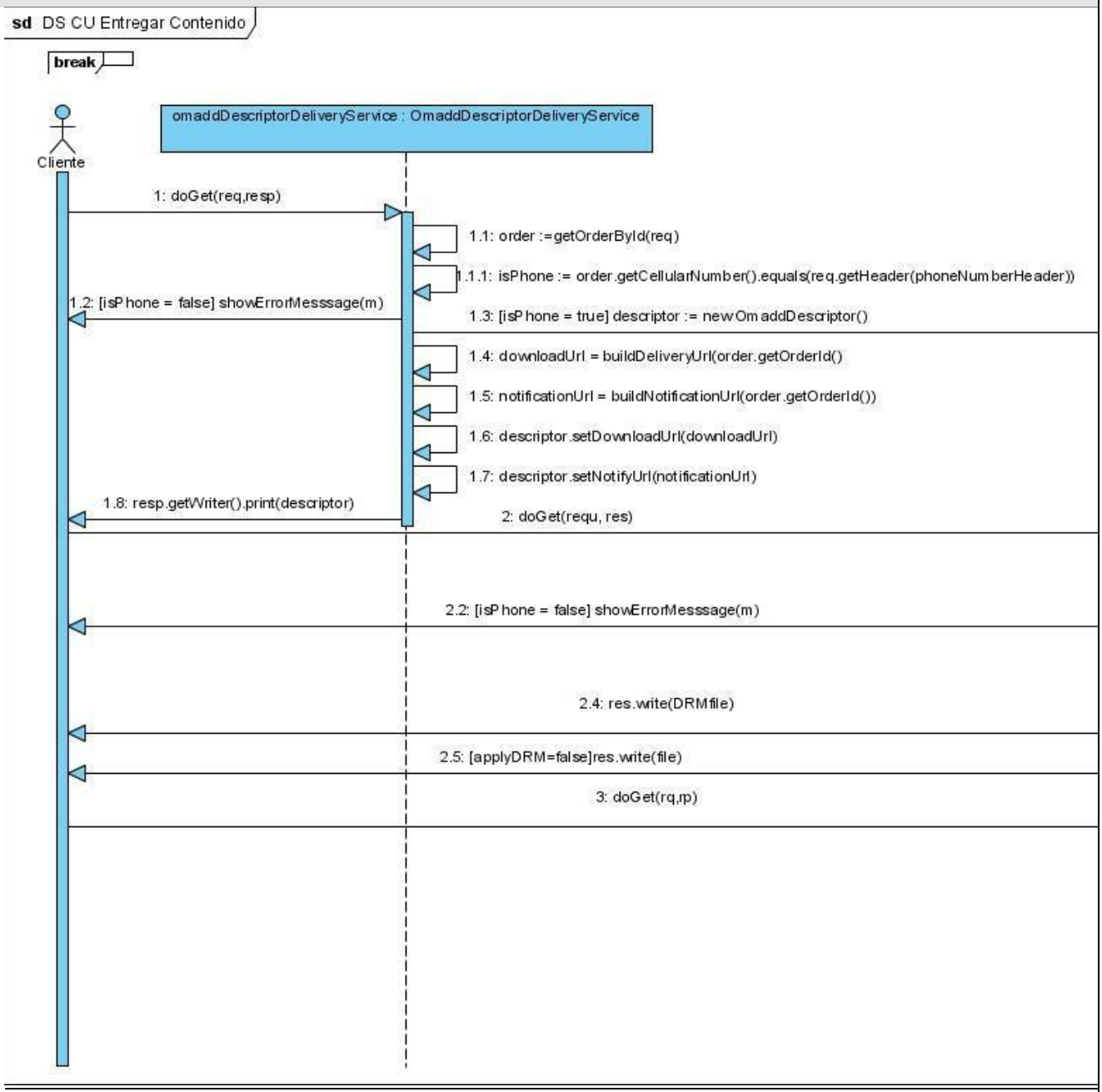



**Figura 11: Diagrama de secuencia del caso de uso Entregar Contenido**

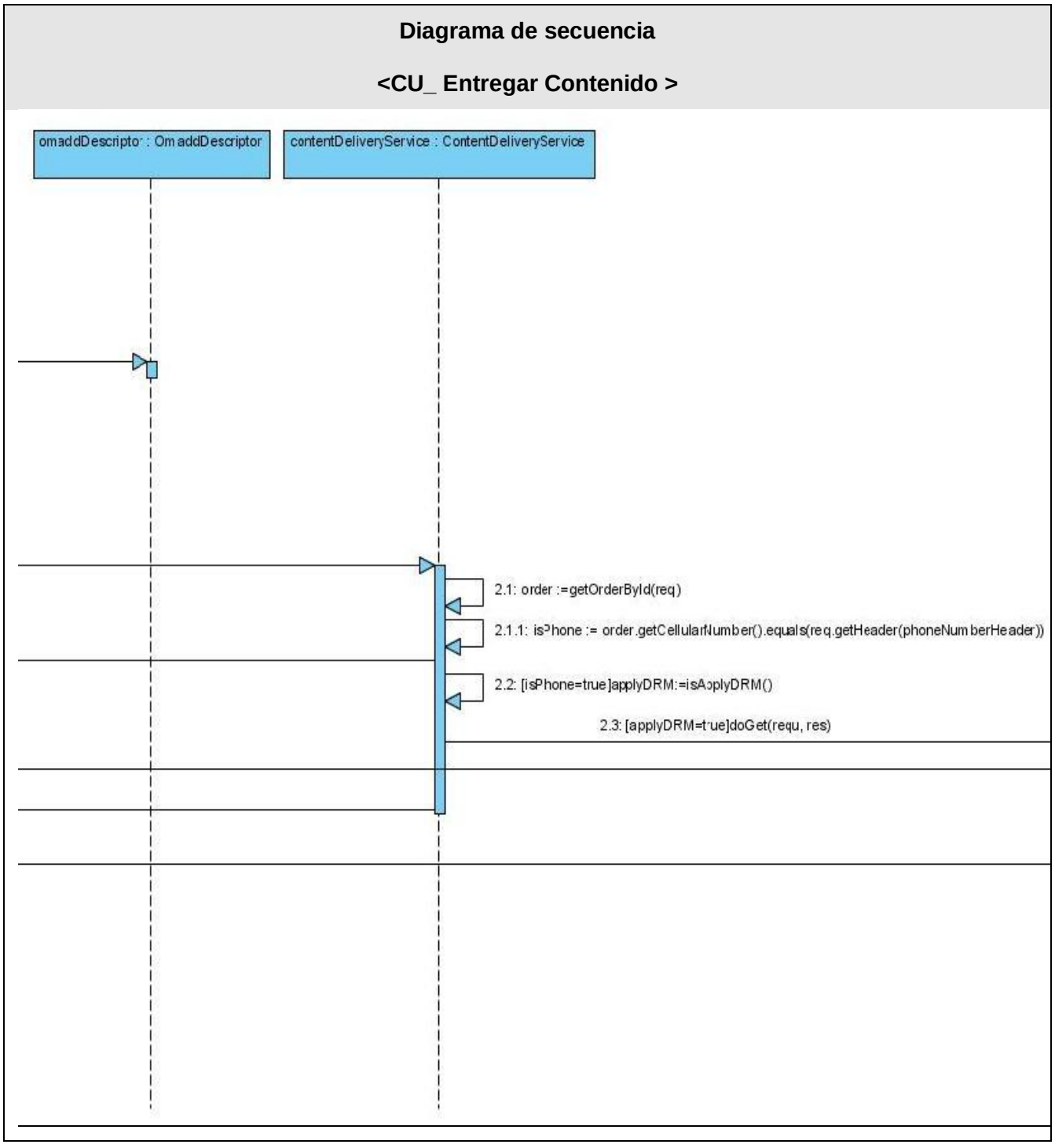



**Figura 12: Diagrama de secuencia del caso de uso Entregar Contenido**

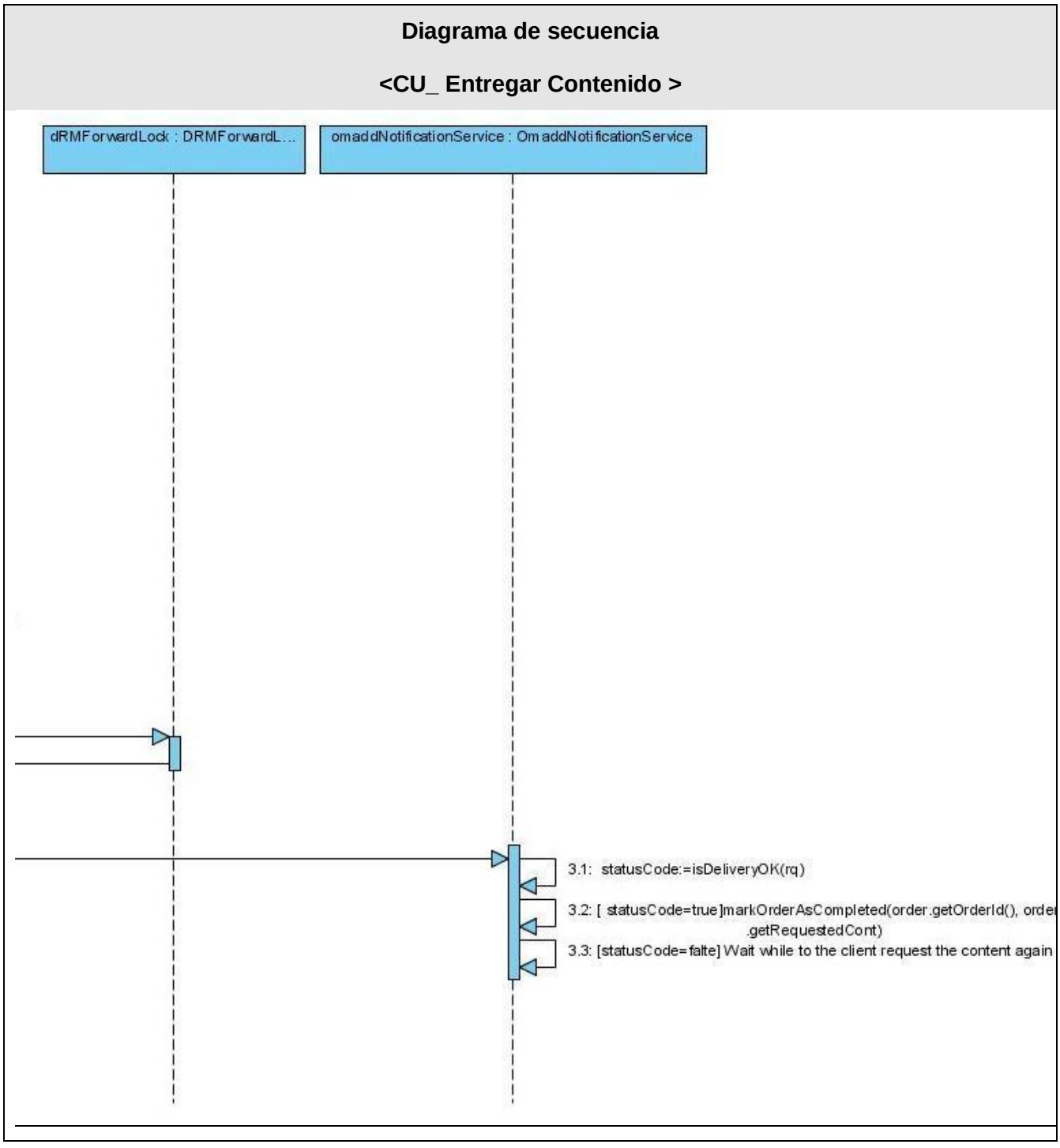



**Figura 13: Diagrama de secuencia del caso de uso Entregar Contenido**

**Diagrama de secuencia**

**<CU_ Expirar Solicitudes >**

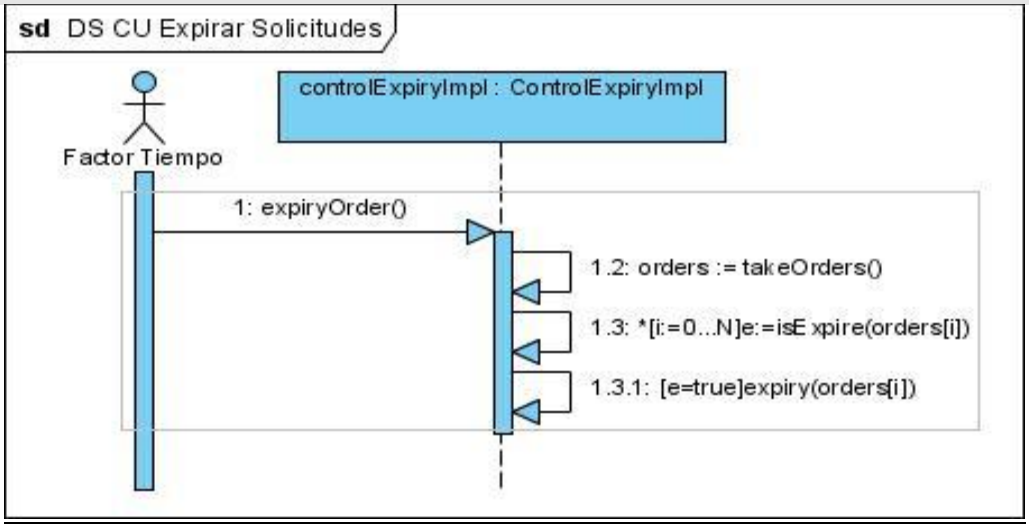



**Figura 14: Diagrama de secuencia del caso de uso Expirar Solicitudes**

## *3.4 Arquitectura*

Un estilo arquitectónico define las reglas generales de organización en términos de un patrón y las restricciones en la forma y la estructura de un grupo numeroso y variado de sistemas de *software*. En una forma más específica, un estilo determina el vocabulario de componentes y conectores que pueden ser utilizados en instancias de este estilo, con un conjunto de restricciones en las descripciones arquitectónicas. (23)

El Módulo Entrega de Contenido emplea una arquitectura Orientada a Objetos la cual se define como un diseño de sistemas que utilizan objetos auto-contenidos y clases de objetos donde:

- Los objetos son abstracciones del mundo real o entidades del sistema que se administran entre ellas mismas.
- Los objetos son independientes y encapsulan el estado y la representación de información.

- La funcionalidad del sistema se expresa en términos de servicios de los objetos.
- Las áreas de datos compartidas son eliminadas. Los objetos se comunican mediante paso de parámetros.
- Los objetos pueden estar distribuidos y pueden ejecutarse en forma secuencial.

Las ventajas de un diseño Orientado a Objetos permiten realizar operaciones de fácil mantenimiento, debido a que los objetos representan entidades auto-contenidas, además son componentes reutilizables y para algunos sistemas, puede haber un mapeo obvio entre las entidades del mundo real y los objetos del sistema.

## *3.5 Fundamentación de los patrones utilizados*

### 3.5.1 Patrones de diseño

Los patrones de diseño son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de *software*, por lo que utilizarlos proporciona una serie de ventajas debido a que:

- Proponen una forma de reutilizar la experiencia de los desarrolladores, para ello clasifica y describe formas de solucionar problemas que ocurren frecuentemente en el desarrollo.
- Están basados en la recopilación del conocimiento de los expertos en el desarrollo de *software*. (23)
- Brindan soluciones eficientes y satisfactoriamente probadas a determinados problemas.

#### 3.5.1.1 Patrones GRASP

Los patrones GRASP (por sus siglas en inglés *General Responsabilty Assignment Software Patterns*) son patrones generales de *software* para la asignación de responsabilidades. Se utilizan el experto, alta cohesión, creador, controlador y bajo acoplamiento, para el desarrollo del Módulo Entrega de Contenido debido a que permiten establecer algunos parámetros útiles para el diseño del producto.

- **Creador:** Se encarga de identificar la clase responsable de la creación de nuevos objetos. Este patrón se aplica generalmente en relaciones asociativas o de composición.
- **Alta cohesión:** Indica que la información que almacena una clase debe ser coherente, de manera que todos sus métodos tengan un comportamiento bien definido. Es utilizado para el desarrollo del módulo debido a que cada clase implementa las funcionalidades que le son correspondidas.
- **Bajo acoplamiento:** Su objetivo es tratar de mantener las clases lo menos ligadas entre sí, de tal forma que en caso de producirse una modificación en alguna de ellas se tenga la mínima repercusión posible en el resto, potenciando la reutilización y disminuyendo sus dependencias.
- **Experto:** Es el encargado de asignar una responsabilidad al experto en información: indica que la creación de un objeto debe recaer sobre la clase que conoce todo lo referente para su realización.
- **Controlador:** Se encarga de gestionar los eventos generados en capas anteriores y a partir de dichos eventos tomar las decisiones apropiadas, pudiendo invocar funcionalidades contenidas en capas más profundas como el acceso a datos. Su función es de mediador o intermediario, del controlador del negocio asociados. (24)

#### 3.5.1.2 Patrones GOF

Dentro de los patrones GOF se utiliza el *Singleton*, su propósito es asegurar que sólo exista una instancia de una clase, proporcionando un punto de acceso global a ésta instancia. Este patrón surge debido de que varios clientes distintos precisan referenciar a un mismo elemento y se necesita asegurar de que no exista más de una instancia de dicho elemento. Se debe utilizar cuando:

- Deba haber exactamente una instancia de una clase y ésta deba ser accesible a los clientes desde un punto de acceso conocido.
- La única instancia debería ser extensible mediante herencia y los clientes deberían ser capaces de utilizar una instancia extendida sin modificar su código. (25)

El patrón *Facade* o Fachada se encuentra dentro de los patrones GOF empleados en la realización del módulo. Este patrón trata de simplificar la interfaz entre dos sistemas o componentes de *software* ocultando un sistema complejo detrás de una clase que hace las veces de pantalla o fachada. La idea principal es la de ocultar todo lo posible la complejidad de un sistema, el conjunto de componentes que lo

forman, que solo se ofrezca un (o unos pocos) punto de entrada al sistema tapado por la fachada. Una de las principales ventajas de utilizarlo, es la de aislar los posibles cambios que se puedan producir en algunas de sus partes. (26)

## *3.6 Tratamiento de excepciones*

El objetivo principal del tratamiento de excepciones es la identificación de los principales errores que pueden surgir durante el funcionamiento del sistema y proporcionarle su tratamiento. En el Módulo Entrega de Contenido el tratamiento de errores estará enfocado principalmente a errores durante la interacción de los clientes con el servicio solicitado, esto puede ocurrir una vez que le envíen a otro usuario la dirección que se le entrega para que puedan obtener los datos del contenido y posteriormente el contenido. Se trata en todo momento de minimizar la ocurrencia de errores de esta índole, aprovechando la posibilidad de enviarle un mensaje de alerta para que rectifique y así garantizar el buen funcionamiento de la aplicación.

## *3.7 Seguridad*

La seguridad del módulo está basada principalmente a la hora de exponer el servicio Web por el cual se registran todas las peticiones debido a que este solo puede ser utilizado por los subsistemas autorizados. Para lograr su buen funcionamiento se utiliza la transmisión segura por el canal HTTPS, además de la utilización de ws-security para la seguridad a nivel de mensajería.

## *3.8 Conclusiones*

En el transcurso de este capítulo se mostró una representación gráfica de los diferentes diagramas del análisis así como los del diseño e interacción por cada caso de uso correspondiente al módulo, proporcionando una comprensión detallada de los requisitos. Se realizó un análisis de los patrones de diseño aplicados, contribuyendo a obtener la calidad requerida y entendimiento, y de esta forma dar paso a la implementación.

# Capítulo 4. Implementación y Prueba

## *4.1 Introducción*

El presente capítulo tiene como objetivo abordar el flujo de trabajo de implementación donde se describe cómo se implementan en términos de componentes los elementos del modelo de diseño. Se modelan los diagramas de componentes y de despliegue, dando una visión de cómo quedará desarrollada y distribuida la aplicación. Además se especifica el tipo de prueba a utilizar para encontrar y documentar los defectos que puedan afectar la calidad del *software* y validar que trabaje como fue diseñado, cumpliendo con los requisitos planteados.

## *4.2 Modelo de implementación*

El modelo de implementación especifica cómo los elementos del modelo de diseño, fundamentalmente las clases, se implementan en términos de componentes como ficheros de código fuente, ejecutables, librerías, entre otros. También describe cómo se organizan los componentes de acuerdo con los mecanismos de estructuración y modularización disponibles en el entorno de implementación y en el lenguaje o lenguajes de programación utilizados, y cómo dependen los componentes unos de otros, así como su organización de acuerdo a los nodos específicos en el modelo de despliegue. (27)

### 4.2.1 Diagrama de despliegue

Un diagrama de despliegue muestra las relaciones físicas entre los componentes *hardware* y *software* en el sistema final, es decir, la configuración de los elementos de procesamiento en tiempo de ejecución y los componentes de *software* (procesos y objetos que se ejecutan en ellos). Se representa mediante un grafo de nodos unidos por conexiones de comunicación. Un nodo es un objeto físico en tiempo de ejecución que representa un recurso computacional. (28)

A continuación se muestran los componentes creados para la solución de la Plataforma de Gestión de Contenidos para Dispositivos Móviles donde se encuentra el Módulo Entrega de Contenido.

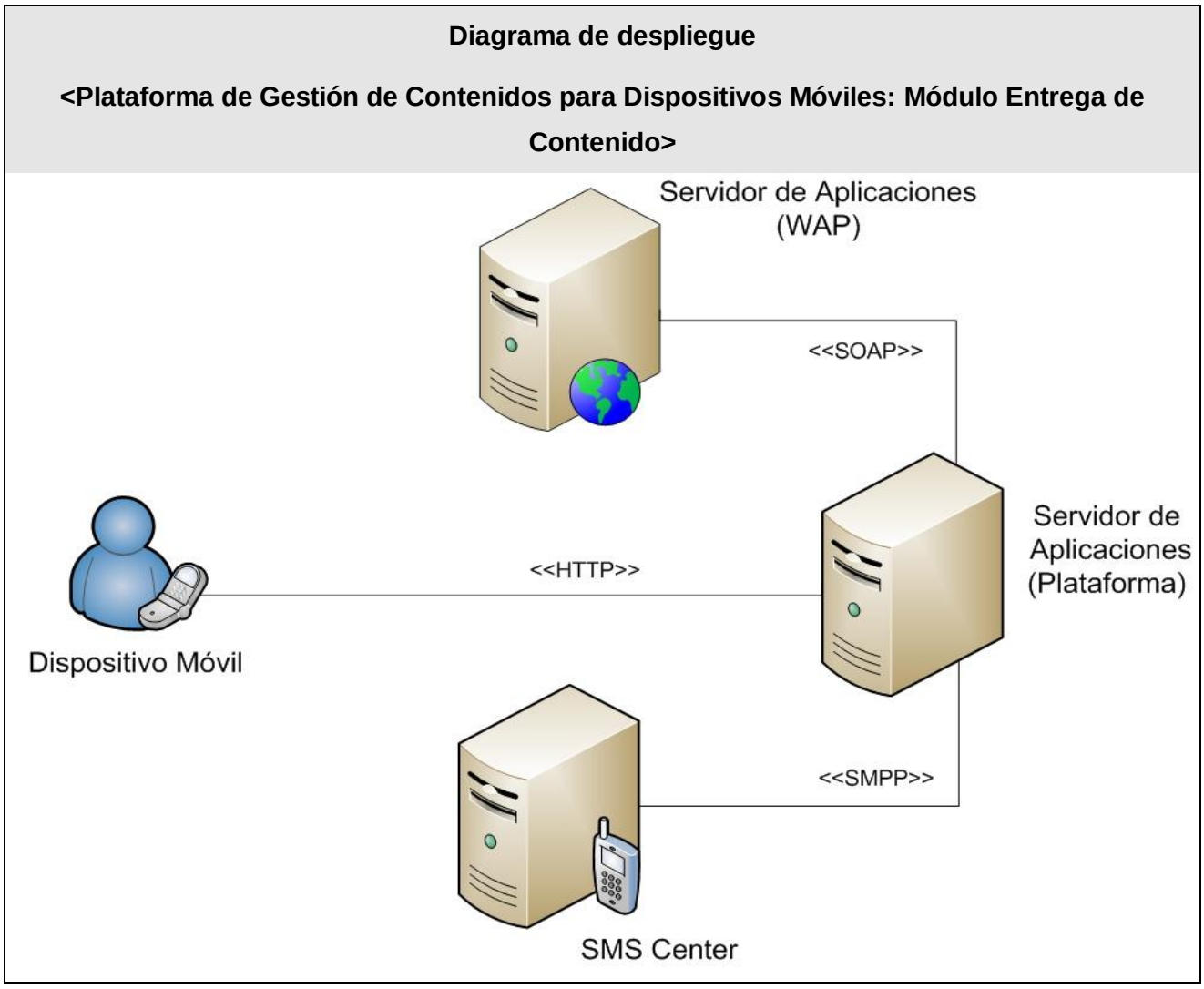



**Figura 15: Diagrama de despliegue de la Plataforma de Gestión de Contenidos para Dispositivos Móviles**

**Descripción de los nodos:**

Dispositivo Móvil: Representa a los usuarios finales de la Plataforma, los cuales se conectan al Portal WAP y al Servidor de Aplicaciones donde se encuentra el Módulo Entrega de Contenido mediante HTTP.

Servidor de Aplicaciones (WAP): En este servidor se encuentra el Portal WAP que atiende la petición del contenido que cada cliente desea descargar. Interactúa con el Módulo Entrega de Contenido mediante

SOAP.

SMS Center: Representa un elemento de la red de telefonía móvil cuya función es la de enviar/recibir mensajes SMS. Se comunica con el servidor de aplicaciones donde está la plataforma con SMPP.

Servidor de Aplicaciones (Plataforma): Constituye el nodo fundamental, aquí se encuentra el Módulo Entrega de Contenido dentro de la Plataforma de Gestión de Contenidos para celulares. Su comunicación con los dispositivos móviles es a través del protocolo HTTP, con el servidor de aplicación donde está el portal WAP, es con SOAP y con el SMS Center es con SMPP.

**Descripción de los protocolos de comunicación:**

HTTP: La transferencia de hipertexto (por sus siglas en inglés *HyperText Transfer Protocol*) es el protocolo usado en cada transacción de la Web. Se utiliza para comunicar los dispositivos móviles con los servidores de aplicaciones (WAP) y el de la Plataforma.

SMPP: El envío de mensajes cortos punto a punto (por sus siglas en inglés *short message peer-to-peer protocol*), es un protocolo estándar de telecomunicaciones pensado para el intercambio de mensajes SMS entre equipos que gestionan los mensajes, se utiliza normalmente para permitir a terceros enviar mensajes. Conecta los nodos del servidor de aplicaciones donde está la plataforma con el SMS Center.

SOAP: El protocolo de acceso a datos (por sus siglas en inglés *Simple Object Access Protocol*) es un protocolo que define cómo dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos XML. SOAP es uno de los protocolos más utilizados en los servicios Web.

### 4.2.2 Diagrama de componentes

Los diagramas de componentes muestran los componentes de *software* que constituyen una parte reusable, sus interfaces y sus interrelaciones, en muchos aspectos se puede considerar que un diagrama de componentes es un diagrama de clases a gran escala. Se representan como un grafo de componentes unidos por medio de relaciones de dependencia (compilación, ejecución). Se utilizan para modelar la vista estática de un sistema y muestra la organización y las dependencias lógicas entre un conjunto de

componentes de *software*, sean éstos componentes de código fuente, librerías, binarios o ejecutables. (27)

El siguiente diagrama muestra la estructura general del sistema en términos de paquetes que agrupan funcionalidades comunes de los componentes utilizados.

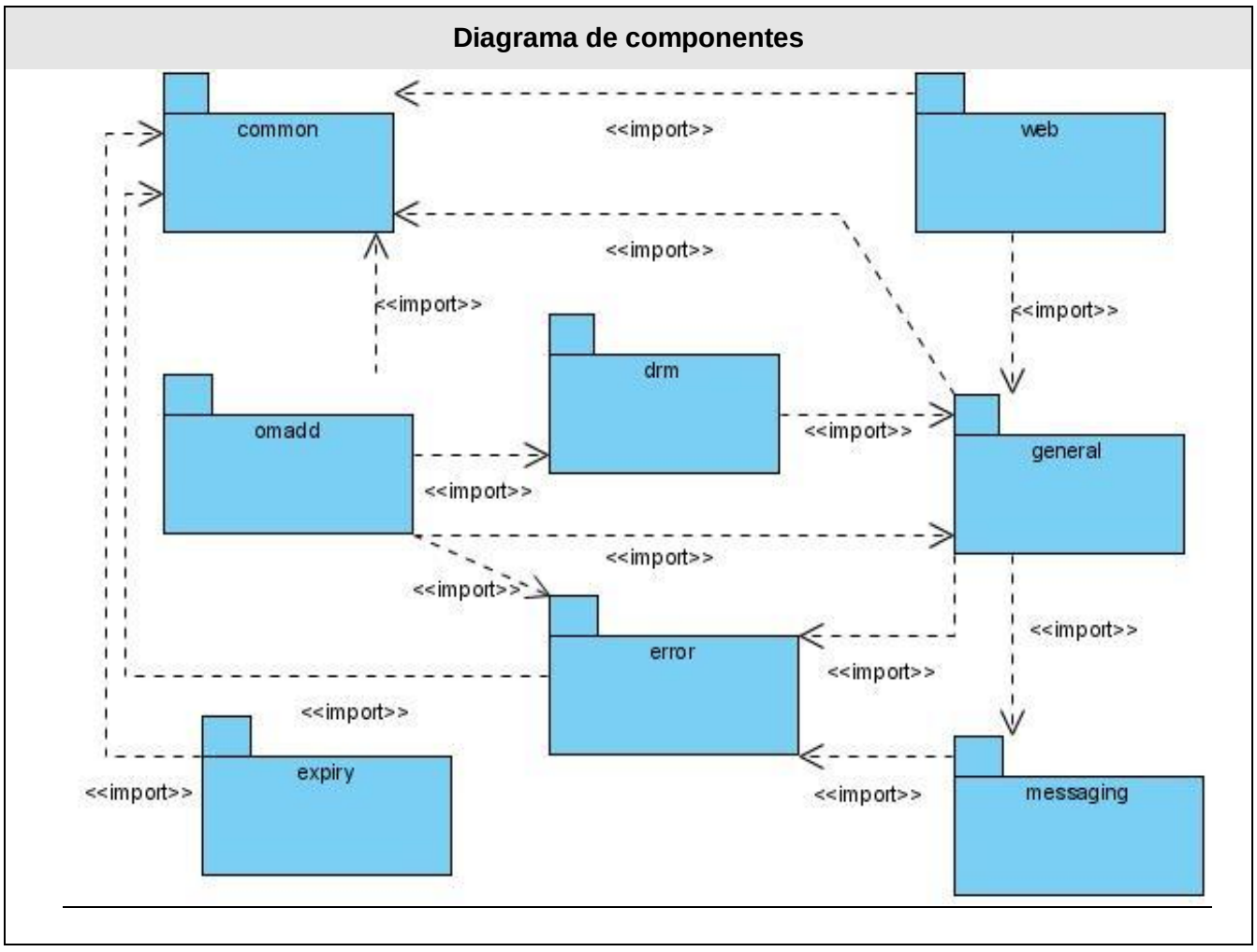



**Figura 16: Diagrama de componentes general del Módulo Entrega de Contenido**

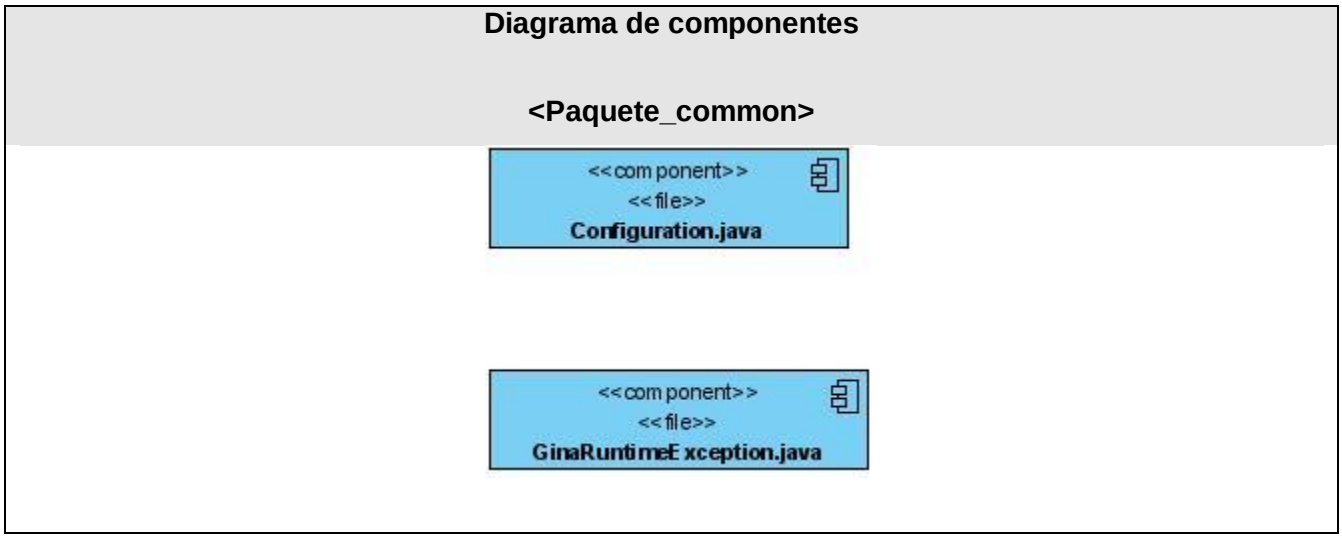



**Figura 17: Diagrama de componentes del Paquete common**

common: Paquete que contiene los componentes utilizados por todos los Módulos de la Plataforma de Gestión de Contenidos para Dispositivos Móviles.

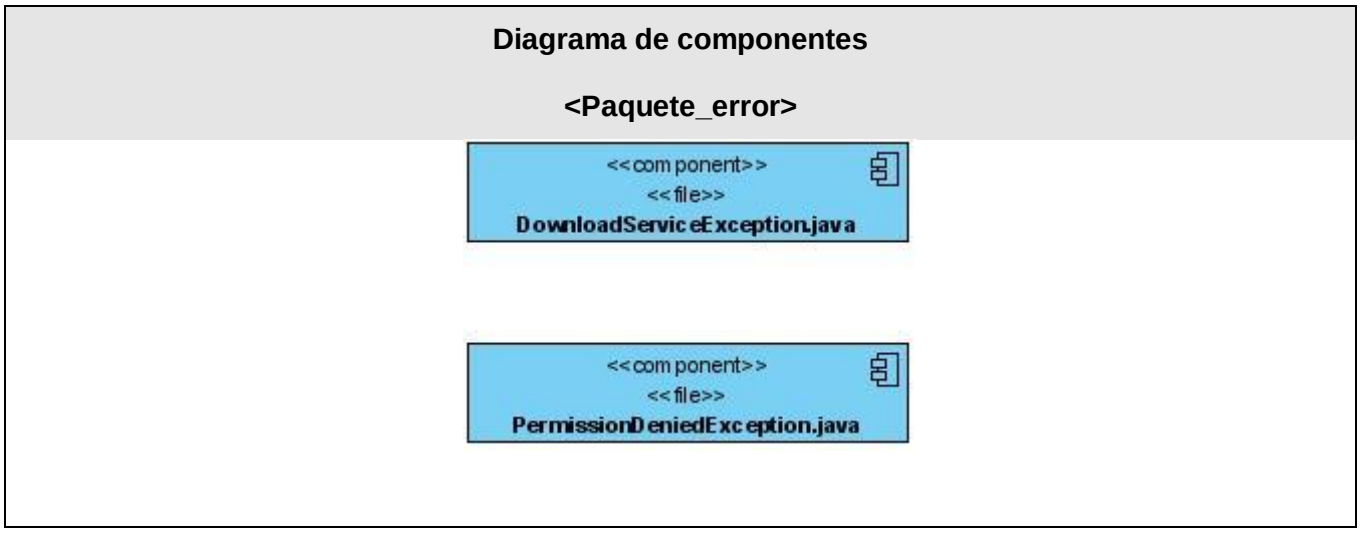



**Figura 18: Diagrama de componentes del Paquete error**

error: Paquete que contiene los componentes que permiten realizar un adecuado tratamiento de

excepciones.

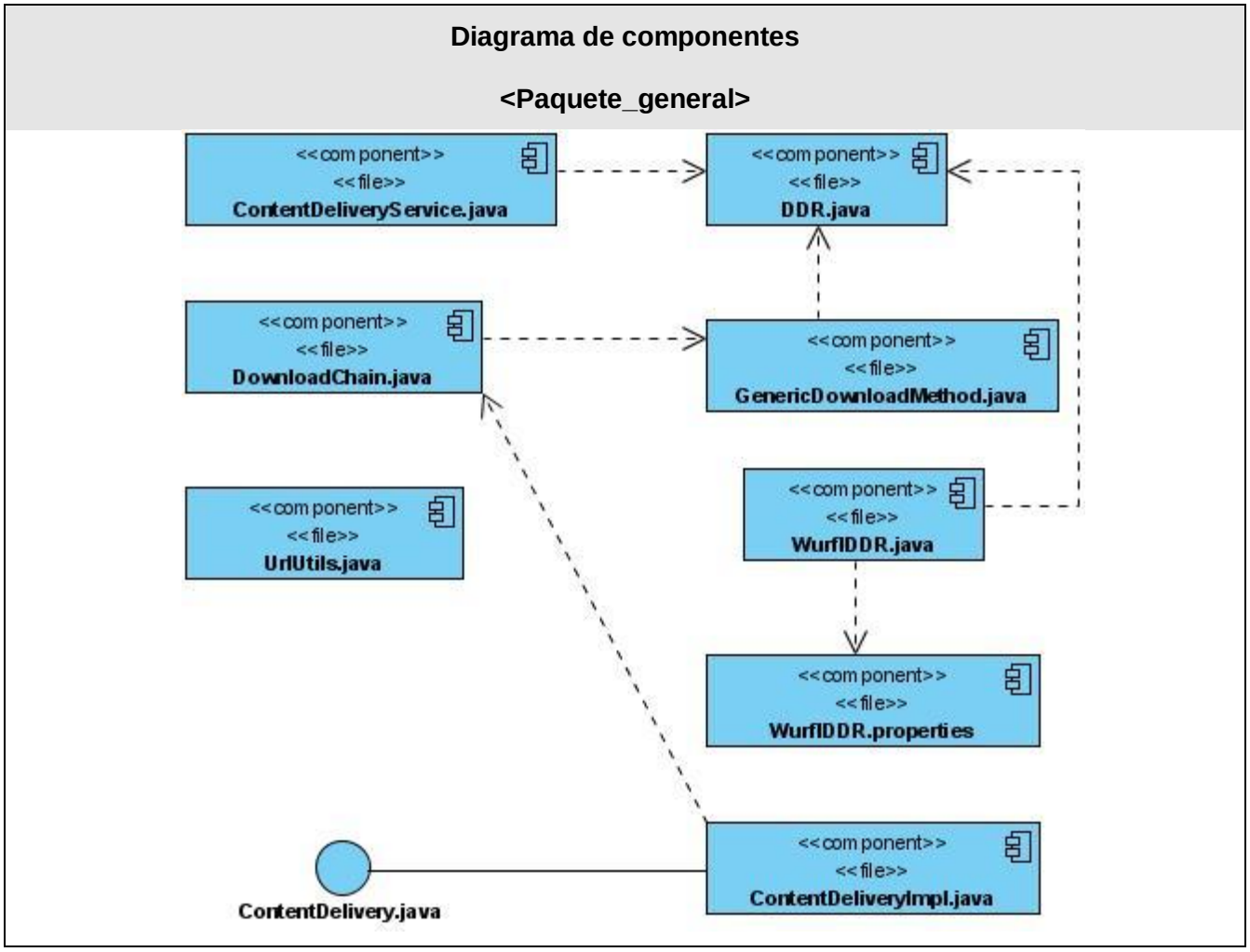



**Figura 19: Diagrama de componentes del Paquete general**

general: Paquete que contiene los componentes encargados crear las URL necesarias para entregar el contenido al cliente, así como los componentes que permiten conocer las características del celular.


**Figura 20: Diagrama de componentes del Paquete drm**

drm: Paquete que contiene los componentes encargados de gestionar el derecho de autor a los contenidos.

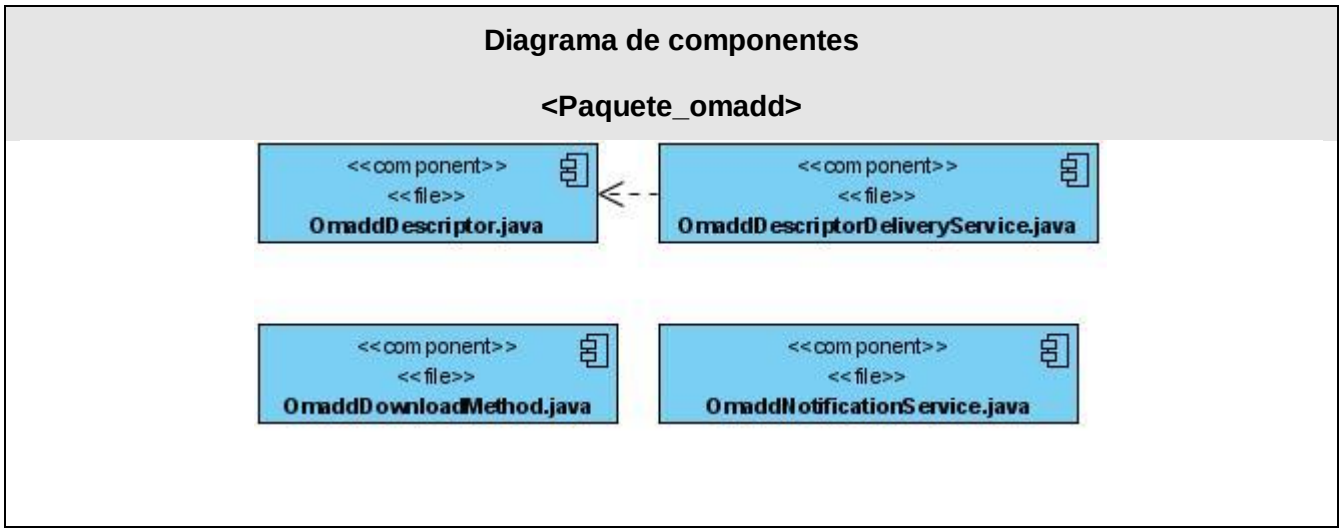



**Figura 21: Diagrama de componentes del Paquete omadd**

omadd: Paquete que contiene los compontes relacionados con la descarga OMA Download.

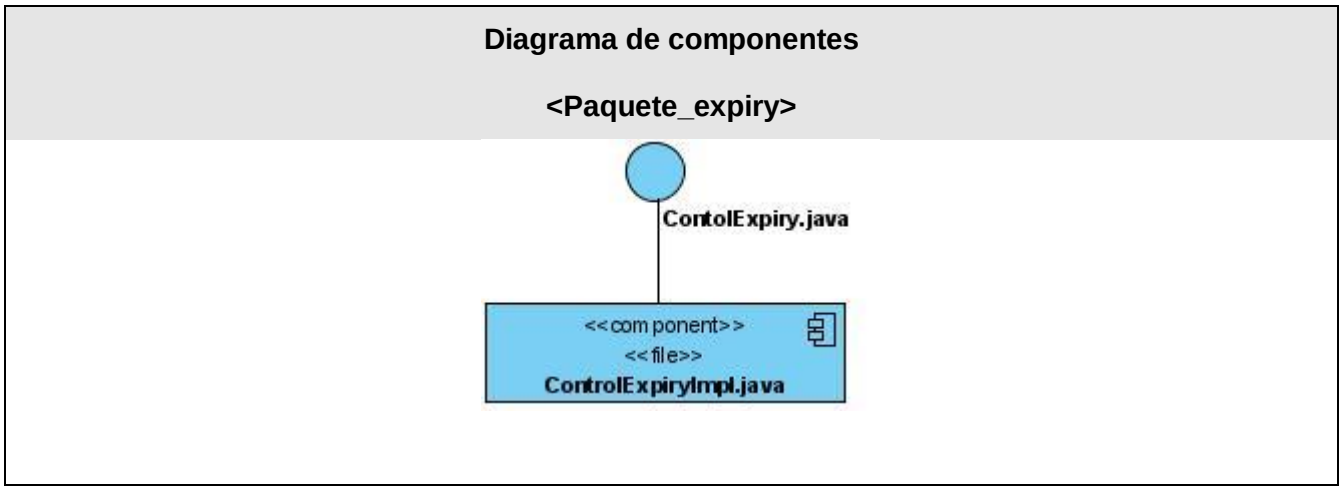



**Figura 22: Diagrama de componentes del Paquete expiry**

expiry: Paquete que contiene los componentes encargados de expirar las solicitudes.

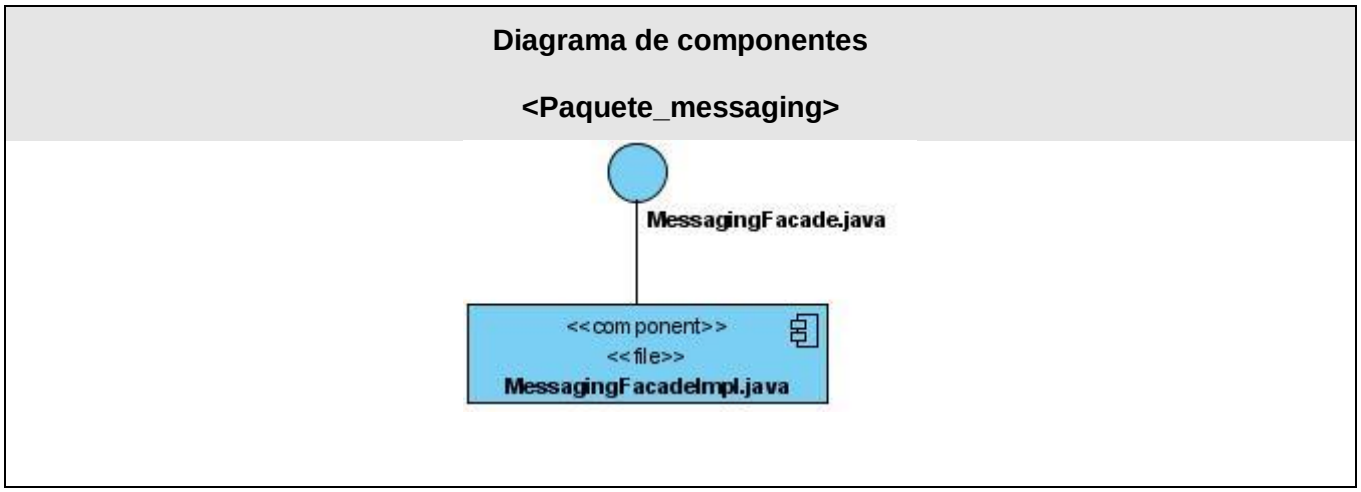



**Figura 23: Diagrama de componentes del Paquete messaging**

messaging: Paquete que contiene los componentes encargados del envío de SMS al cliente.

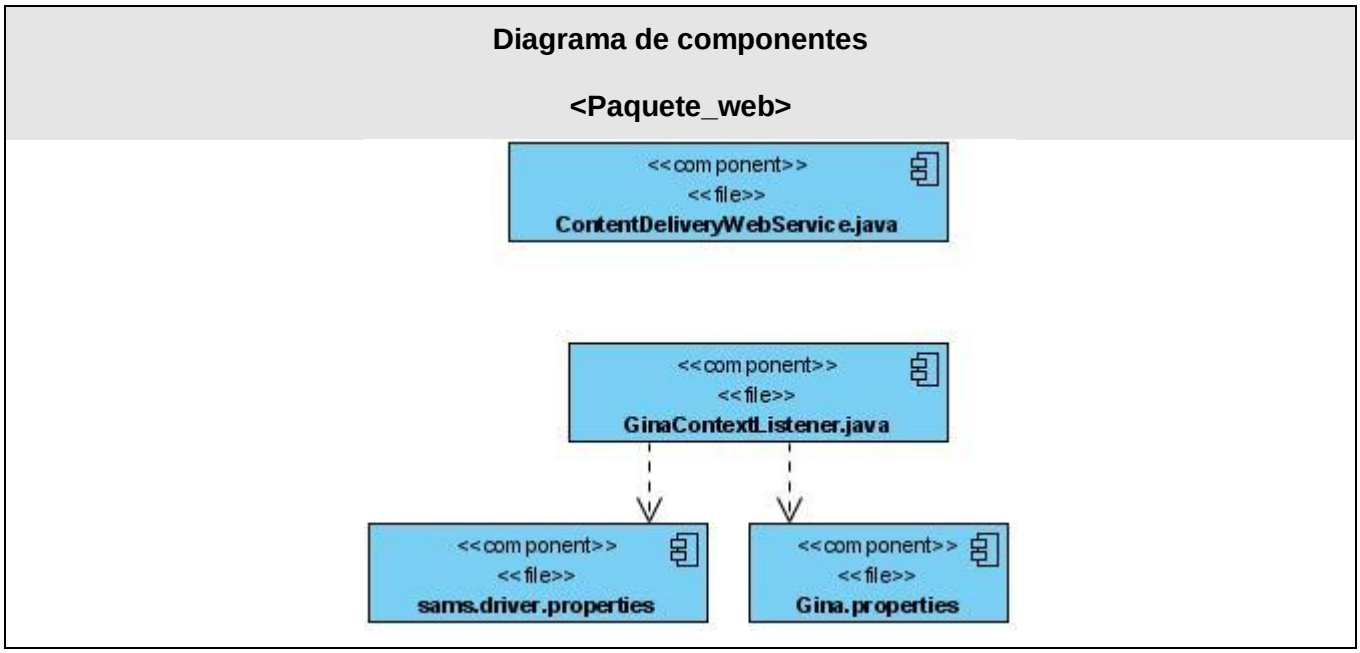



**Figura 24: Diagrama de componentes del Paquete web**

web: Paquete que contiene los componentes relacionados con el servicio web expuesto por el módulo y con la configuración del mismo.

## *4.3 Modelo de pruebas*

Las pruebas son un proceso de ejecución de un programa con la intención de descubrir errores, su principal objetivo es evaluar la calidad del producto a través de la validación del cumplimiento de los requerimientos, además de buscar y documentar los errores. (29). Existen dos métodos fundamentales para lograr su buen funcionamiento: método de caja blanca y método de caja negra.

### 4.3.1 Método de prueba de Caja Negra

En este método las pruebas se llevan a cabo sobre la interfaz del *software*. El objetivo es demostrar que las funciones del *software* son operativas, que las entradas se aceptan de forma adecuada y se produce un resultado correcto, y que la integridad de la información externa se mantiene. Este método ha sido utilizado con el objetivo de realizar pruebas que se centren principalmente en los requisitos funcionales

permitiendo encontrar: (29)

- Funciones incorrectas o ausentes.
- Errores de interfaz.
- Errores de rendimiento.
- Errores de inicialización y terminación.

El proceso de una prueba de caja negra es simple, se ejecuta la unidad de prueba con datos, se observa la salida y se compara con el resultado esperado.

A continuación se presenta la descripción de los casos de prueba de integración, para cada caso de uso.

**Caso de Uso:** Atender solicitud

| **Claves válidas** | **Clases inválidas** | **Resultado esperado** | **Resultado** |
|---|---|---|---|
| Se recibe la solicitud del contenido por el Portal WAP con los datos siguientes:<br><br>**Número de teléfono:** 5352885187<br><br>**Fecha de inicio de la solicitud:** 0009-06-04 02:17:13.014<br><br>**Identificador del contenido solicitado:** 94<br><br>**User Agent:** SAMSUNG- | | Se registran los datos en la base de datos mediante el Módulo de Contenido.<br><br>Se crea la URL para que el cliente descargue el descriptor:<br><br>**URL:** http://waptest.uci.cu/omadd?oid=110<br><br>Se envía la URL al cliente mediante el Módulo de Mensajería. | Satisfactorio. |

| | | | |
|---|---|---|---|
| SGH-D900i/1.0<br>Profile/MIDP-2.0<br>Configuration/CLDC-1.1<br>UP.Browser/6.2.3.3.c.1.101<br>(GUI) MMP/2.0 | | | |

**Tabla 20: Caso de prueba del caso de uso Atender solicitud**

**Caso de uso:** Entregar contenido.

| Claves válidas | Clases inválidas | Resultado esperado | Resultado |
|---|---|---|---|
| El cliente accede a la siguiente URL para obtener el descriptor: http://waptest.uci.cu/omadd?oid=110& x-jinny.cid=5352885187 | | Se verifica que la orden solicitada existe y que el número de teléfono coincide con el registrado. Luego se procede a conformar el descriptor con los siguientes datos:<br><br>type: image/jpeg<br><br>size: 2864<br><br>name: peque<br><br>vendor: Cubacel<br><br>objectURI: http://waptest.uci.cu/delivery?oid=110 | |

| | | | |
|---|---|---|---|
| | | installNotifyURI: http://waptest.uci.cu/notif ?oid=110<br><br>y se procede a la entrega. | |
| El cliente luego de visualizar los datos accede a la URL http://waptest.uci.cu/delivery?oid=110&x-jinny.cid=5352885187 para obtener el contenido. | | Se solicita el contenido al Módulo de Contenido y se le entrega al cliente. | Satisfactorio. |
| El cliente envía la notificación de descarga a la siguiente URL http://waptest.uci.cu/notif?oid=110&x-jinny.cid=5352885187 | | Se obtiene el número de la notificación igual 900(éxito) y se procede al cobro. | Satisfactorio. |
| | El cliente accede a la siguiente URL para obtener el descriptor: http://waptest.uci.cu/omadd?oid=110& x-jinny.cid=5352885185 | Se verifica que la orden solicitada existe y el número de teléfono no coincide con el registrado. Se le muestra al cliente el mensaje siguiente “Invalid request, phone number not included” | Satisfactorio. |

**Tabla 21: Caso de prueba del caso de uso Entregar contenido**

**Caso de uso:** Expirar solicitudes

Para la realización de esta prueba se definieron 10 minutos como tiempo máximo para expirar las solicitudes y 6 minutos como el tiempo establecido para comprobar las solicitudes que han expirado. Se

procede a reiniciar el servidor a las 08:32 PM.

| Claves válidas | Clases inválidas | Resultado esperado | Resultado |
|---|---|---|---|
| A 6 minutos de reiniciar el servidor se levanta un proceso que ejecuta el método encargado de expirar las solicitudes. | | El método pide al Módulo de Contenido las ordenes registradas:<br><br>**Orden 1**: ordenid=112 fechaInicio 0009-06-04 08:20:13.014<br><br>**Orden 2**: ordenid=113 fechaInicio 0009-06-04 08:23:13.014<br><br>**Orden 3**: ordenid=114 fechaInicio 0009-06-04 08:30:13.014<br><br>Se procede a comparar las órdenes quedando eliminadas las dos primeras. | Satisfactorio. |

**Tabla 22: Caso de prueba del caso de uso Expirar solicitudes**

### 4.3.2 Método de prueba de Caja Blanca

La elección de este método permite comprobar los caminos lógicos del *software*, se puede examinar el estado del programa en varios puntos para determinar si el estado real coincide con el esperado. Estas pruebas requieren del conocimiento de la estructura interna del programa y deben garantizar que: (29)

- Se ejerciten por lo menos una vez todos los caminos independientes para cada módulo y todas las decisiones lógicas en sus vertientes verdaderas y falsa.
- Ejecuten todos los bucles en sus límites y con sus límites operacionales.
- Se comprueben las estructuras internas de datos para asegurar su validez.

#### 4.3.2.1 Caso de prueba.

| Método del caso de uso<br><br>< Atender solicitud> |
|---|

```
public void registerOrder(Order order) {1

            Long contentId = order.getRequestedContent();1
            Content content = contentService.findContentById(contentId);1
            boolean support = true;1

            if(content.isDrm()){2
                  String ua = order.getUserAgent();3
                  support = ddr.isSupportedCapability(ua, new DRMForwardLock());3
            }
            if(support){4
                  for (GenericDownloadMethod method : downloadMethods) {5
                        if (method.isSupportedBy(order.getUserAgent())) {6
                              method.registerNewOrder(order);7
                              break;7
                        }
                  }
            }
            else{
                  sendSMSInformation(order);8
            }

                                          }9
```

**Figura 25: Caso de prueba del método perteneciente al caso de uso Atender solicitud**

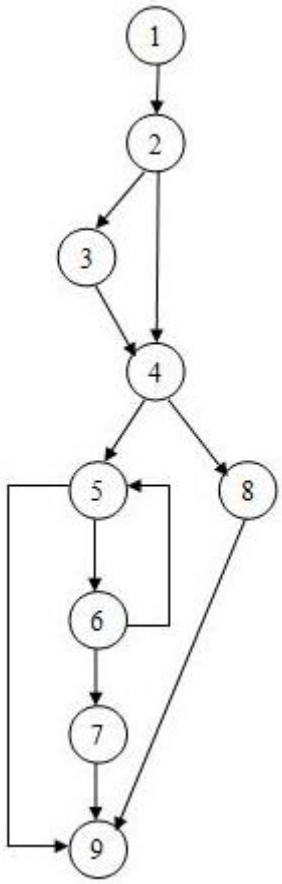



**Figura 26: Grafo correspondiente al método del caso de uso Atender solicitud**

**Complejidad Ciclomática.**

V (G) = A-N+2 = 12 - 9 + 2 = 5

**Caminos Independientes.**

1-2-3-4-5-9

1-2-3-4-5-6-7-9

1-2-3-4-5-6-9

1-2-3-4-5-6-5-9

1-2-4-8-9

## *4.4 Conclusiones*

En este capítulo se representaron los elementos necesarios para la comprensión de cómo el *software* fue implementado, como el diagrama de componente y el de despliegue. Además de algunos ejemplos de las pruebas realizadas para la verificación de las funcionalidades.

# Capítulo 5. Estudio de la factibilidad

## *5.1 Introducción*

El estudio de la factibilidad muestra como resultado final la factibilidad de la ejecución de cualquier proyecto de *software*. En el presente capítulo se realiza, usando el método puntos de casos de uso el cálculo del esfuerzo y el tiempo de desarrollo, así como los beneficios y costos, mostrando de esta forma si es viable o no llevar a cabo la realización del módulo.

## *5.2 Planificación*

### 5.2.1 Puntos de Casos de Uso

Puntos de Casos de Uso es un método de estimación del tiempo de desarrollo de un proyecto mediante la asignación de "pesos" a un cierto número de factores que lo afectan, para finalmente, contabilizar el tiempo total estimado para el proyecto a partir de esos factores. (30)

#### 5.2.1.1 Cálculo de Puntos de Casos de Usos sin ajustar

El cálculo de los Puntos de Caso de Uso sin ajustar constituye el primer paso y se calcula a partir de la siguiente ecuación:

**UUCP = UAW + UUCW**

Donde:

- **UUCP**: Puntos de Casos de Uso sin ajustar.
- **UAW**: Factor de Peso de los Actores sin ajustar.
- **UUCW**: Factor de Peso de los Casos de Uso sin ajustar.

Para obtener el Factor de Peso de los Actores sin ajustar se debe tener en cuenta la cantidad de actores presentes en el sistema y la complejidad de los mismos. Los criterios se muestran en la siguiente tabla:

| Tipo de Actor | Descripción | Peso | Cantidad * Peso |
|---|---|---|---|
| Simple | Otro sistema que interactúa con el sistema a desarrollar mediante una interfaz de programación (*API*, *Application Programming Interface*). | 1 | 0 * 1 |
| Medio | Otro sistema que interactúa con el sistema a desarrollar mediante un protocolo o una interfaz basada en texto. | 2 | 4 * 2 |
| Complejo | Una persona que interactúa con el sistema mediante una interfaz gráfica. | 3 | 0 * 3 |
| **Total** | | | 8 |

**Tabla 23: Factor de Peso de los Actores**

Para obtener el valor del Factor de Peso de los Casos de Uso sin ajustar se debe tener en cuenta la cantidad de casos de uso y la complejidad de cada uno de ellos. Los criterios se muestran en la siguiente tabla:

| Tipo de Caso de Uso | Descripción | Peso | Cantidad * Peso |
|---|---|---|---|
| Simple | El Caso de Uso contiene de 1 a 3 transacciones. | 5 | 3 * 5 |
| Medio | El Caso de Uso contiene de 4 a 7 transacciones. | 10 | 1 * 10 |
| Complejo | El Caso de Uso contiene más de 8 transacciones. | 15 | 0 * 15 |
| **Total** | | | 25 |

**Tabla 24: Factor de Peso de los Casos de Uso**

Por lo que los Puntos de Caso de Uso sin ajustar resultan:

**UUCP**= UAW + UUCW = 8+ 25 = **33**

### 5.2.1.2 Cálculo de Puntos de Casos de Uso ajustados

Una vez obtenidos los Puntos de Caso de Uso sin ajustar se procede al cálculo de los Puntos de Casos de Uso ajustados mediante la siguiente ecuación:

**UCP = UUCP * TCF * EF**

Donde:

- **UCP**: Puntos de Casos de Uso ajustados.
- **UUCP**: Puntos de Casos de Uso sin ajustar.
- **TCF**: Factor de complejidad técnica.
- **EF**: Factor de ambiente.

Factor de complejidad técnica: Este coeficiente se calcula mediante un conjunto de Factores que determinan la complejidad del sistema, cada factor se cuantifica con un valor de 0 a 5.

Significado de los valores:

- 0: No presente o sin influencia.
- 1: Influencia incidental o presencia incidental.
- 2: Influencia moderada o presencia moderada.
- 3: Influencia media o presencia media.
- 4: Influencia significativa o presencia significativa.
- 5: Fuerte influencia o fuerte presencia.

La ecuación para su cálculo es:

**TCF = 0.6 + 0.01 * Σ (Peso i * Valor Asignado i)**

| Factor | Descripción | Peso | Peso * Valor |
|---|---|---|---|
| T1 | Sistema distribuido. | 2 | 2 * 1 |
| T2 | Objetivos de performance o tiempo de respuesta. | 1 | 1 * 4 |
| T3 | Eficiencia del usuario final. | 1 | 1 * 5 |
| T4 | Procesamiento interno complejo. | 1 | 1 * 3 |
| T5 | El código debe ser reutilizable. | 1 | 1 * 3 |
| T6 | Facilidad de instalación. | 0.5 | 0.5 * 2 |
| T7 | Facilidad de uso. | 0.5 | 0.5 * 4 |
| T8 | Portabilidad. | 2 | 2 * 4 |
| T9 | Facilidad de cambio. | 1 | 1 * 3 |
| T10 | Concurrencia. | 1 | 1 * 5 |
| T11 | Incluye objetivos especiales de seguridad. | 1 | 1 * 4 |
| T12 | Provee acceso directo a terceras partes. | 1 | 1 * 5 |
| T13 | Se requieren facilidades especiales de entrenamiento a los usuarios. | 1 | 1 * 0 |
| **Total** | | | 45 |

**Tabla 25: Factor de Complejidad Técnica**

Entonces, **TCF** = 0.6 + 0.01 * 45 = **1.05**

Factor de ambiente: Contempla las habilidades y el entrenamiento del grupo de desarrollo por su importancia en las estimaciones de tiempo. Al igual que el factor de complejidad técnica se cuantifican con valores de 0 a 5. La ecuación para su cálculo es:

EF = 1.4 - 0.03 * Σ (Peso i * Valor Asignado i).

| Factor | Descripción | Peso | Peso * Valor |
|---|---|---|---|

| | | | |
|---|---|---|---|
| E1 | Familiaridad con el modelo de proyecto utilizado. | 1.5 | 1.5 * 3 |
| E2 | Experiencia en la aplicación. | 0.5 | 0.5 * 1 |
| E3 | Experiencia en orientación a objetos. | 1 | 1 * 5 |
| E4 | Capacidad del analista líder. | 0.5 | 0.5 * 4 |
| E5 | Motivación. | 1 | 1 * 5 |
| E6 | Estabilidad de los requerimientos. | 2 | 2 * 4 |
| E7 | Personal part - time. | -1 | -1 * 3 |
| E8 | Dificultad del lenguaje de programación. | -1 | -1 * 3 |
| **Total** | | | 19 |

**Tabla 26: Factor de Ambiente**

Entonces, **EF** = 1.4 - 0.03 * 19 = **0,83**

Finalmente, **UCP (Puntos de Caso de Uso ajustados) =** UUCP * TCF * EF = 33 * 1.05 * 0.83 = **28.7595**

### 5.2.1.3 Estimación del esfuerzo

El esfuerzo en horas - hombre viene dado por:

**E = UCP * CF**

Donde:

- **E**: Esfuerzo estimado en horas-hombre.
- **UCP**: Puntos de Casos de Uso ajustados.
- **CF**: Factor de conversión.

**CF** = 20 horas-hombre (si Total EF ≤ 2)

**CF** = 28 horas-hombre (si Total EF = 3 ó Total EF = 4)

**CF =** abandonar o cambiar proyecto (si Total EF ≥ 5)

**Total EF = Cant EF < 3 (entre E1 –E6) + Cant EF > 3 (entre E7, E8)**

Como **Total EF** = 2 + 0

**Total EF = 2**

**CF = 20 horas-hombre (porque Total EF ≤ 2)**

Luego **E** = 28.7595 * 20 horas-hombre

E = 575.19 horas-hombre

### 5.2.1.4 Distribución del Esfuerzo entre las diferentes actividades

Para llevar a cabo una estimación más completa de la duración total del módulo, se agrega el esfuerzo de las demás actividades. Para ello se plantea la distribución del esfuerzo entre las diferentes actividades:

| Actividad | Porcentaje | Horas-Hombre |
|---|---|---|
| Análisis | 10% | 143.7975 |
| Diseño | 20% | 287.595 |
| Implementación | 40% | 575.19 |
| Pruebas | 15% | 215.69625 |
| Otras actividades | 15% | 215.69625 |
| **Total** | 100% | 1437.975 |

**Tabla 27: Distribución del Esfuerzo**

El Esfuerzo Total sería 1437.975 horas-hombre, si se estima teniendo en cuenta que un mes tiene 176 horas laborables, pues se trabajan 8 horas diarias 22 días al mes, entonces el Esfuerzo Total en mes-hombre sería 8.17 mes-hombre.

### 5.2.1.5 Calcular el costo de todo el proyecto

**Costo = CHM * ET / CH**

Donde:

- **CH**: Cantidad de hombres.
- **CHM**: Costo Hombre – Mes.
- **ET**: Esfuerzo Total.

Si la Cantidad de hombres es 2 y se tiene un Salario Promedio mensual igual a $100.00.

Entonces **CHM** = **CH** * Salario Promedio

**CHM** = 2 * 100

**CHM** = 200.00

Luego **Costo** = **CHM * ET / CH**

**Costo** = 200.00 * 8.17 / 2

**Costo = $ 817**

### 5.2.1.6 Calcular el tiempo de desarrollo de todo el proyecto

A partir de los datos conocidos anteriormente se podría asegurar que una persona puede realizar el trabajo en un tiempo aproximado de 8.17 meses. Pero teniendo en cuenta que la cantidad de hombres para la realización del Módulo Entrega de Contenido es 2 el tiempo va a ser:

**Tiempo = ET / CH**

**Tiempo = 8.17 / 2**

**Tiempo = 4.085 ≈ 4 meses.**

## 5.3 Análisis de costos y beneficios

Para la implementación del sistema se emplearon tecnologías libres, por lo que el pago de licencias de *software* no afectará su costo. Los desarrolladores del sistema son estudiantes de la facultad dos de la Universidad de las Ciencias Informáticas, debido a esto no hay gastos en salario de profesionales. La aplicación va dirigida a la empresa Cubacel y su mayor beneficio se refleja en el mejoramiento de los servicios que brinda a sus clientes. Los medios tecnológicos utilizados son los que se invirtieron en la Facultad por parte de la administración de la Universidad, además de los teléfonos celulares que proporcionó la empresa para las pruebas de la aplicación, por lo que el sistema no supone altos gastos en recursos. Por todo lo planteado se puede llegar a la conclusión que es totalmente factible el desarrollo del sistema.

## 5.4 Conclusiones

Con el estudio de la factibilidad, la estimación, el cálculo del tamaño, el esfuerzo y el costo para el desarrollo del *software* conjuntamente con los beneficios fueron elementos presentados en este capítulo, resultando viable el desarrollo del módulo.

# Conclusiones generales

Al concluir este trabajo, con el desarrollo del Módulo Entrega de Contenido perteneciente a la Plataforma Gestión de Contenidos para Dispositivos Móviles, se solucionó satisfactoriamente el problema existente en cuanto a la descarga de contenidos hacia los celulares dando cumplimiento al objetivo general de la investigación presentada. La puesta en marcha de este sistema permitirá ofrecer mejoras en los servicios que brinda la Empresa Cubacel.

Durante el desarrollo del módulo se realizó un proceso de investigación basado en el estado del arte relativo a los métodos empleados para garantizar el servicio de entrega de contenidos hacia celulares, además se pudieron evaluar las diferentes formas para garantizar los permisos exclusivos del proveedor de contenidos. Dicho estudio condujo hacia la utilización de OMA Download y DRM Forward Lock como método de descarga y de conservación de derechos de autor respectivamente; perteneciente este último a la categoría OMA DRM, todo lo cual proporciona ventajas apreciables para el proveedor del servicio debido a que impide el reenvío de contenidos entre clientes, proporcionando un incremento económico en las ganancias de la empresa. Todo el proceso de desarrollo del software ha estado sustentado en la utilización de metodologías y herramientas eficientes definidas con anterioridad, tales como RUP, para guiar el trabajo del equipo de desarrollo a través de un ciclo que permita de manera organizada traducir los requisitos de software a un sistema que garantice la solución a la problemática existente. Entre las herramientas escogidas figuran el entorno de desarrollo integrado: Eclipse Ganymede debido a sus ventajas para completamiento de código, depuración de errores, diferenciación de sintaxis, entre otras y la herramienta para el modelado UML: Visual Paradigm debido entre otros factores a su soporte multiplataforma. Durante la fase de inicio del proyecto fue necesario realizar un modelo de dominio que permitiera ofrecer una visión clara de los principales conceptos empleados asociados a las entidades implicadas; realizando además descripciones detalladas de todos los casos de uso del sistema, sirviendo de base para los posteriores flujos de Análisis y Diseño. Con la culminación de la fase de Elaboración del software quedó definida una arquitectura Orientada a Objetos; dando paso a la implementación del módulo, todo lo cual es expresado mediante diagramas de componentes que reflejan la relación entre clases y paquetes. El módulo funciona como conector con los módulos: Portal WAP, Contenido, Mensajería y Pago, permitiendo realizar operaciones que van desde la gestión de la solicitud del cliente

hasta la posterior entrega y pago del contenido. Finalmente el producto ha sido objeto de un conjunto de pruebas que contribuyen a garantizar la calidad del mismo, dándole a Cubacel la posibilidad de avanzar en su desarrollo para colocarse a la altura de otras empresas internacionales experimentadas en el sector.

# Recomendaciones

Como resultado del presente trabajo se obtuvo un módulo de un sistema informático que aunque resuelve la problemática planteada inicialmente, puede ser refinado con el objetivo de lograr un incremento en su calidad, a continuación se exponen algunas recomendaciones a tener en cuenta para futuras versiones del producto:

- Extender el sistema con la realización de otros métodos de descargas empleados en el mundo.
- Incluir nuevas técnicas para la Gestión de Derechos Digitales, como los métodos *Combined Delivery* y *Separate Delivery*.
- Analizar después de un período de tiempo el impacto que ha tenido el software en los clientes.

# Bibliografía

1. **López, Jorge Blanco.** Telefonía móvil: Las palabras en el aire. [En línea] junio de 2006. [Citado el: 15 de octubre de 2008.] http://observatorio.cnice.mec.es/modules.php?op=modload&name=News&file=article&sid=354.

2. mobiForge. [En línea] septiembre de 2006. [Citado el: 10 de diciembre de 2008.] http://mobiforge.com/developing/story/content-delivery-mobile-devices..

3. OMA. [En línea] [Citado el: 15 de enero de 2009.] http://www.openmobilealliance.org/.

4. Developer Network. [En línea] 2008. [Citado el: 10 de enero de 2009.] http://developer.openwave.com/dvl/support/faqs/faq_download_fun.htm..

5. **Montero, Norbelis Leyva.** *Análisis y Diseño de una aplicación para la protección de contenido en dispositivos móviles.* Ciudad de la Habana : s.n., 2008.

6. **Denis.** manzana mecánica. [En línea] 2 de junio de 2008. [Citado el: 15 de enero de 2009.] http://www.manzanamecanica.org/2008/07/eclipse_ganymede.html.

7. **Dpto. Ing. Electrónica, Sist. Intormática de la Universidad de Huelva.** Nuevas Tecnologías de la Programación. [En línea] 2006-2007. [Citado el: 20 de enero de 2009.] http://www.uhu.es/josel_alvarez/NvasTecnProg/recursos/tTema1.pdf.

8. **Navón, Jaime.** Introducción a JavaEE(ex-J2EE). [En línea] 2006. [Citado el: 20 de enero de 2009.] http://openpuc.cl/index.php?option=com_docman&task=doc_view&gid=14&Itemid=59.

9. Ciberaula. [En línea] 2008. [Citado el: 20 de enero de 2009.] http://java.ciberaula.com/articulo/que_es_java/.

10. **Luna, Juan Manuel Fernández.** Java. [En línea] septiembre de 2006. [Citado el: 20 de enero de 2009.] http://leo.ugr.es/J2ME/INTRO/intro_1.htm.

11. **Thomas Risberg, Rick Evans, Portia Tung.** Spring Framework. [En línea] [Citado el: 20 de enero de 2009.] http://static.springframework.org/docs/Spring-MVC-step-by-step/.

12. Seam City. [En línea] 2007. [Citado el: 21 de enero de 2009.] http://seamcity.madeinxpain.com/archives/comparativa-spring.

13. Agapea. [En línea] 2008. [Citado el: 21 de enero de 2009.] http://www.agapea.com/libros/Tomcat-6-0-La-guia-definitiva-isbn-8441524319-i.htm.

14. **Ben Alex, Luke Taylor.** Sping Framework. [En línea] [Citado el: 21 de enero de 2009.] http://static.springframework.org/spring-security/site/reference/html/springsecurity.html.

15. Rational Unified Process. [En línea] [Citado el: 22 de enero de 2009.] http://www.ibm.com/developerworks/rational/library/content/03July/1000/1251/1251_bestpractices_TP026B.pdf.

16. **Luis Carlos Díaz, Angela Carrillo y Deicy Alvarado.** IS, RUP y UML en el Contexto de ADOO. Análisis y Diseño OO. [En línea] enero de 2008. [Citado el: 21 de enero de 2009.] http://sophia.javeriana.edu.co/~lcdiaz/ADOO2008-1/IngSoftwareEnADOO(IS-RUP-UML).pdf.

17. UML. Lenguaje de Modelado. [En línea] [Citado el: 22 de enero de 2009.] http://alfa.facyt.uc.edu.ve/computacion/pensum/cs0347/download/exposiciones2006-2007/presentacion%20UML.pdf.

18. Sitio de descargas de software. [En línea] 2007. [Citado el: 23 de enero de 2009.] http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_(Iglesia_Anglicana)_%5Bcuenta_de_Linux_14716_p/.

19. **Pavón, Eduardo León.** Blog de Eduardo León. [En línea] 2008. [Citado el: 23 de enero de 2009.] http://slion2000.blogspot.com/2007/04/visual-paradigm-una-herramienta-de-lo.html.

20. Stefano Salvatori. [En línea] 3 de octubre de 2007. [Citado el: 23 de enero de 2009.]

http://stefano.salvatori.cl/blog/2007/10/03/control-de-versiones-svn/.

21. *Fase de Inicio. Flujo de trabajo de requerimiento.* 2008.

22. *Flujo de trabajo de Análisis y Diseño.* 2008-2009.

23. *Arquitectura y Patrones de diseño.* 2007-2008.

24. un gran inicio. [En línea] 17 de agosto de 2006. [Citado el: 16 de marzo de 2009.] http://jorgeerazo.blogspot.com/2006/08/patrones-grasp.html.

25. msdn Microsoft Developer Network. [En línea] [Citado el: 10 de abril de 2009.] http://msdn.microsoft.com/es-es/library/bb972272.aspx#EEAA.

26. Java en castellano. [En línea] 2007. [Citado el: 10 de abril de 2009.] http://www.programacion.com/java/articulo/joa_patrones4/#joa_patrones3_fachada.

27. *Fase de Construcción.Flujo de trabajo de Implementación.*

28. **Vila, Ana Fernandez.** [En línea] 20 de marzo de 2001. [Citado el: 29 de abril de 2009.] http://tvdi.det.uvigo.es/~avilas/UML/node50.html.

29. *Flujo de trabajo de pruebas.* **Aguilar, Violena Hernández.**

30. *Planificación y Estimación de Proyectos.*

# Anexos

## *Anexo 1. Descripción de las clases*

**Clase ContentDelivery**

**<CU_ Atender Solicitud >**


**Figura 27: Clase ContentDelivery**

| **Nombre:** ContentDelivery | |
|---|---|
| **Para cada responsabilidad**: | |
| **Nombre:** | registerNewOrder(order : Order) : void |
| **Descripción:** | Recibe una orden, y realiza la gestión para que la misma sea registrada en la Base de Datos. |

**Tabla 28: Descripción de la clase ContentDelivery**

**Clase ContentDeliveryWebService**

**<CU_ Atender Solicitud>**

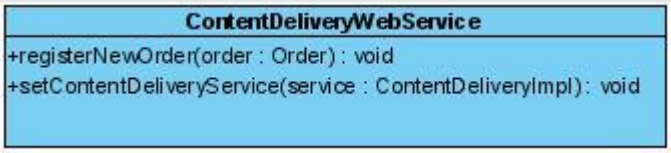



**Figura 28: Clase ContentDeliveryWebService**

| **Nombre:** ContentDeliveryWebService | |
|---|---|
| **Atributo** | **Tipo** |
| contentDeliveryService | ContentDeliveryImpl |
| **Para cada responsabilidad**: | |
| **Nombre:** | registerNewOrder(order: Order): void |
| **Descripción:** | Implementa el método de la Interfaz ContentDelivery. |
| **Nombre:** | setContentDeliveryService(service: ContentDeliveryImpl):void |
| **Descripción:** | Modifica el atributo contentDeliveryService de la clase, por el nuevo valor. |

**Tabla 29: Descripción de la clase ContentDeliveryWebService**

**Clase ContentDeliveryImpl**

**<CU_ Atender Solicitud>**


**Figura 29: Clase ContentDeliveryImpl**

| **Nombre:** ContentDeliveryImpl | |
|---|---|
| **Atributo** | **Tipo** |
| downloadChain | DownloadChain |
| **Para cada responsabilidad**: | |
| **Nombre:** | registerNewOrder(order: Order):void |
| **Descripción:** | Implementa el método de la Interfaz ContentDelivery. |

| | |
|---|---|
| **Nombre:** | setDownloadChain(downloadChain: DownloadChain):void |
| **Descripción:** | Modifica el atributo downloadChain de la clase, por el nuevo valor. |

**Tabla 30: Descripción de la clase ContentDeliveryImpl**

**Clase DownloadChain**

**<CU_ Atender Solicitud>**


**Figura 30: Clase DownloadChain**

| **Nombre:** DownloadChain | |
|---|---|
| **Atributo** | **Tipo** |
| downloadMethods | List<GenericDownloadMethod> |
| **Para cada responsabilidad**: | |
| **Nombre:** | registerOrder(order: Order):void |
| **Descripción:** | Con el user agent del teléfono, verifica si soporta el método de descarga OMA Download y delega la responsabilidad de atender la solicitud. |
| **Nombre:** | setDownloadMethods(downloadMethods :List<GenericDownloadMethod>) :void |
| **Descripción:** | Modifica el atributo downloadMethods de la clase, por el nuevo valor. |

**Tabla 31: Descripción de la clase DownloadChain**

**Clase GenericDownloadMethod**

**<CU_ Atender Solicitud>**

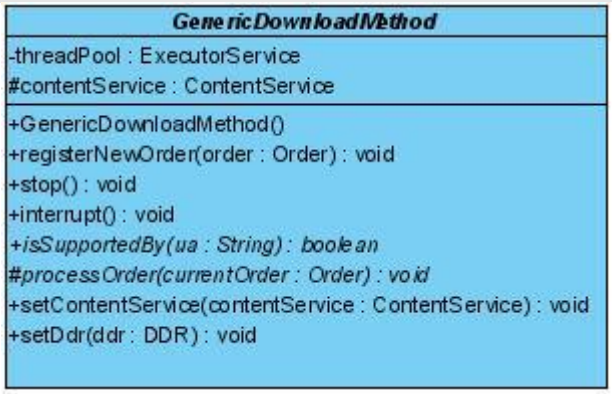



**Figura 31: Clase GenericDownloadMethod**

| **Nombre:** GenericDownloadMethod | |
|---|---|
| **Atributo** | **Tipo** |
| contentService | ContentService |
| ddr | DDR |
| messagingService | MessagingFacade |
| threadPool | ExecutorService |
| **Para cada responsabilidad**: | |
| **Nombre:** | isSupportedBy(ua: String): abstract boolean |
| **Descripción:** | Verifica con el user agent si un teléfono soporta o no la descarga. |
| **Nombre:** | registerNewOrder(order: Order):void |
| **Descripción:** | A través de un hilo de ejecución asigna la responsabilidad de procesar la orden. |
| **Nombre:** | processOrder(currentOrder: Order): abstract void |

| | |
|---|---|
| **Descripción:** | |
| **Nombre:** | interrupt():void |
| **Descripción:** | Interrumpe el hilo de ejecución threadPool. |
| **Nombre:** | stop():void |
| **Descripción:** | Detiene el hilo de ejecución threadPool. |
| **Nombre:** | setContentService(contentService: ContentService):void |
| **Descripción:** | Modifica el atributo contentService de la clase, por el nuevo valor. |
| **Nombre:** | setDdr(ddr :DDR):void |
| **Descripción:** | Modifica el atributo ddr de la clase, por el nuevo valor. |

**Tabla 32: Descripción de la clase GenericDownloadMethod**

**Clase OrderTask**

**<CU_ Atender Solicitud>**

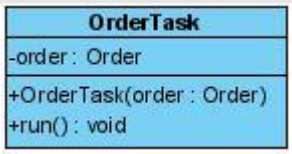



**Figura 32: Descripción de la clase GenericDownloadMethod**

| **Nombre:** OrderTask | |
|---|---|
| **Atributo** | **Tipo** |

| order | Order |
|---|---|
| **Para cada responsabilidad**: | |
| **Nombre:** | run():void |
| **Descripción:** | Encargado de guardar la orden en la Base de Datos mediante el Módulo de Contenido, y solicita que se le informe al cliente donde debe descargar el descriptor. |

**Tabla 33: Descripción de la clase OrderTask**

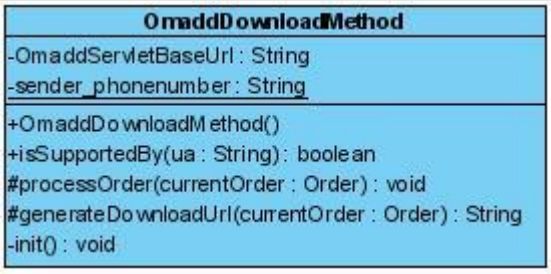
| **Clase OmaddDownloadMethod**<br>**<CU_ Atender Solicitud>** |
|---|
|  |

**Figura 33: Clase OmaddDownloadMethod**

| **Nombre:** OmaddDownloadMethod | |
|---|---|
| **Atributo** | **Tipo** |
| OmaddServletBaseUrl | String |
| sender_phonenumber | String |
| **Para cada responsabilidad**: | |
| **Nombre:** | isSupportedBy(ua: String): boolean |
| **Descripción:** | Verifica si el cliente que realiza la solicitud soporta el tipo de descarga OMA Download. |
| **Nombre:** | processOrder(currentOrder: Order):void |

| | |
|---|---|
| **Descripción:** | Mediante un SMS envía al cliente la URL donde debe descargar el descriptor. En caso de que al contenido se le deba aplicar DRM y el celular no lo soporte se le envía entonces un mensaje informándole que no tienes acceso al contenido solicitado. |
| **Nombre:** | generateDownloadUrl(currentOrder: Order):String |
| **Descripción:** | Genera la URL donde el cliente debe descargar el descriptor con los datos referentes a la entrega del contenido solicitado. |
| **Nombre:** | init():void |
| **Descripción:** | Inicializa los atributos de la clase. |

**Tabla 34: Descripción de la clase OmaddDownloadMethod**

**Clase DDR**

**<CU_ Atender Solicitud>**


**Figura 34: Clase DDR**

| | |
|---|---|
| **Nombre:** DDR | |
| **Para cada responsabilidad**: | |
| **Nombre:** | isSupportedCapability(ua:String,capability: Object):boolean |
| **Descripción:** | Verifica a través del User Agent si un dispositivo celular soporta determinada capability. Por ejemplo, si soporta la descarga OMA Download o el método Forward Lock de DRM. |

**Tabla 35: Descripción de la clase DDR**

**Clase WurflDDR**

**<CU_ Atender Solicitud>**


**Figura 35: Clase WurflDDR**

| **Nombre:** WurflDDR | |
|---|---|
| **Atributo** | **Tipo** |
| config | Properties |
| capabilityMatrix | CapabilityMatrix |
| manager | UAManager |
| configFileName | String |
| **Para cada responsabilidad**: | |
| **Nombre:** | isSupportedCapability(ua:String,capability: Object):boolean |
| **Descripción:** | Verifica a través del User Agent si un dispositivo celular soporta determinada capability. |
| **Nombre:** | getCapabilityForObject(object :Object):String |
| **Descripción:** | Busca en el fichero wurfl.properties el nombre de la capability que representa el objeto que se pasa como parámetro en el fichero wurl.xml. |
| **Nombre:** | getCapabilityForUA(ua: String, capabilityName: String): String |
| **Descripción:** | Con el parámetro ua busca el identificador del dispositivo celular, luego con este y con capabilityName devuelve el valor de la capability, el cual especifica si dicho celular soporta o no la capability solicitada. |

**Tabla 36: Descripción de la clase WurflDDR**

| Clase MessagingFacade<br><CU_ Atender Solicitud> |
|---|
|  |

**Figura 36: Clase MessagingFacade**

| **Nombre:** MessagingFacade | |
|---|---|
| **Para cada responsabilidad**: | |
| **Nombre:** | sendSmsMessage(message: SmsMessage): void |
| **Descripción:** | Envía un SMS al cliente que solicitó el contenido informando acerca de la descarga. |
| **Nombre:** | closeMessagingSession():void |
| **Descripción:** | Cierra la sesión una vez enviado el SMS al cliente. |
| **Nombre:** | createsSmsMessage(Class<?> messageClass): SmsMessage |
| **Descripción:** | Crea un SMS para enviarle información al cliente. |

**Tabla 37: Descripción de la clase MessagingFacade**

**Clase MessagingFacadeImpl**

**<CU_Atender Solicitud>**

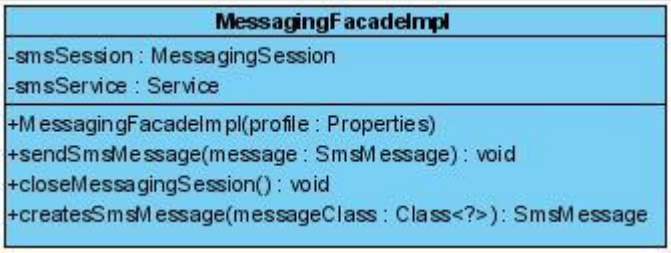



**Figura 37: Clase MessagingFacadeImpl**

| **Nombre:** MessagingFacadeImpl | |
|---|---|
| **Atributo** | **Tipo** |
| smsSession | MessagingSession |
| smsService | Service |
| **Para cada responsabilidad**: | |
| **Nombre:** | sendSmsMessage(message: SmsMessage): void |
| **Descripción:** | Envía un SMS al cliente que solicitó el contenido informando acerca de la descarga. |
| **Nombre:** | closeMessagingSession():void |
| **Descripción:** | Cierra la sesión una vez enviado el SMS al cliente. |
| **Nombre:** | createsSmsMessage(Class<?> messageClass): SmsMessage |
| **Descripción:** | Crea un SMS para enviarle información al cliente. |

**Tabla 38: Descripción de la clase MessagingFacadeImpl**

**Clase ContentDeliveryService**

**<CU_Entregar Contenido>**

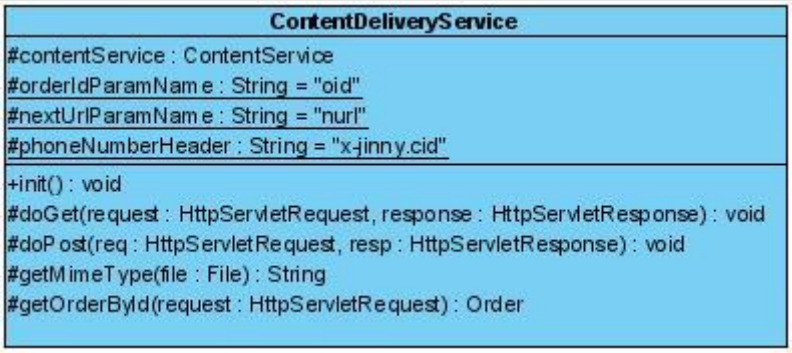



**Figura 38: Clase ContentDeliveryService**

| **Nombre:** ContentDeliveryService | |
|---|---|
| **Atributo** | **Tipo** |
| contentService | ContentService |
| orderIdParamName | String |
| nextUrlParamName | String |
| phoneNumberHeader | String |
| **Para cada responsabilidad**: | |
| **Nombre:** | getOrderById(request: HttpServletRequest): Order |
| **Descripción:** | Busca en el request el parámetro que representa el identificador de e la orden y con este, obtiene mediante el módulo de Contenido la orden registrada. También verifica si el teléfono que está accediendo es el mismo que realizó la solicitud del contenido. |
| **Nombre:** | getMimeType(file: File): String |
| **Descripción:** | Dado un archivo devuelve el mime type. Por el ejemplo el una imagen se obtendría image/jpeg. |

| | |
|---|---|
| **Nombre:** | doGet(request: HttpServletRequest, response:HttpServletResponse): void |
| **Descripción:** | Si al contenido hay que aplicarle DRM delega la responsabilidad e de entrega a la clase DRMFordwarLock, en caso contrario entrega el contenido al celular. |
| **Nombre:** | init():void |
| **Descripción:** | Carga las dependencias de spring Frameworks e inyecta atributos de la clase como contentService y ddr. |

**Tabla 39: Descripción de la clase ContentDeliveryService**

**Clase OmaddNotificationService**

**<CU_Entregar Contenido>**


**Figura 39: Clase OmaddNotificationService**

| | |
|---|---|
| **Nombre:** OmaddNotificationService | |
| **Para cada responsabilidad**: | |
| **Nombre:** | doGet(request: HttpServletRequest, response: HttpServletResponse): void |
| **Descripción:** | Recibe y da tratamiento a la notificación enviada por el dispositivo del usuario una vez realizada la descarga del contenido. |
| **Nombre:** | markOrderAsCompleted(orderId: long, contented: Long): void |
| **Descripción:** | En caso de éxito en la descarga del contenido, incrementa la cantidad de veces que el mismo se ha descargado, manda a cobrar este, y elimina la orden referente |

| | |
|---|---|
| | al mismo. |

**Tabla 40: Descripción de la clase OmaddNotificationService**

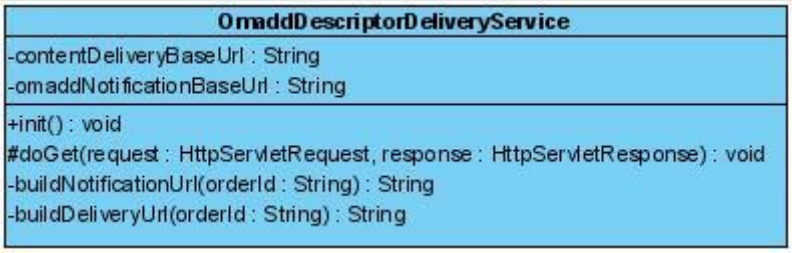
| **Clase OmaddDescriptorDeliveryService**<br>**<CU_Entregar Contenido>** |
|---|
|  |

**Figura 40: Clase OmaddDescriptorDeliveryService**

| **Nombre:** OmaddDescriptorDeliveryService | |
|---|---|
| **Atributo** | **Tipo** |
| contentDeliveryBaseUrl | String |
| omaddNotificationBaseUrl | String |
| **Para cada responsabilidad**: | |
| **Nombre:** | doGet(request: HttpServletRequest,response:HttpServletResponse):void |
| **Descripción:** | Entrega el descriptor referente a la descarga OMA Download al cliente que solicitó el contenido. |
| **Nombre:** | init():void |
| **Descripción:** | Inicializa los atributos correspondientes a las URL. |
| **Nombre:** | buildNotificationUrl(orderId: String):String |
| **Descripción:** | Construye la URL donde el cliente enviará la notificación de la descarga. |

| | |
|---|---|
| **Nombre:** | buildDeliveryUrl (orderId: String):String |
| **Descripción:** | Construye la URL donde el cliente descargará el contenido. |

**Tabla 41: Descripción de la clase OmaddDescriptorDeliveryService**

**Clase OmaddDescriptor**

**<CU_Entregar Contenido>**

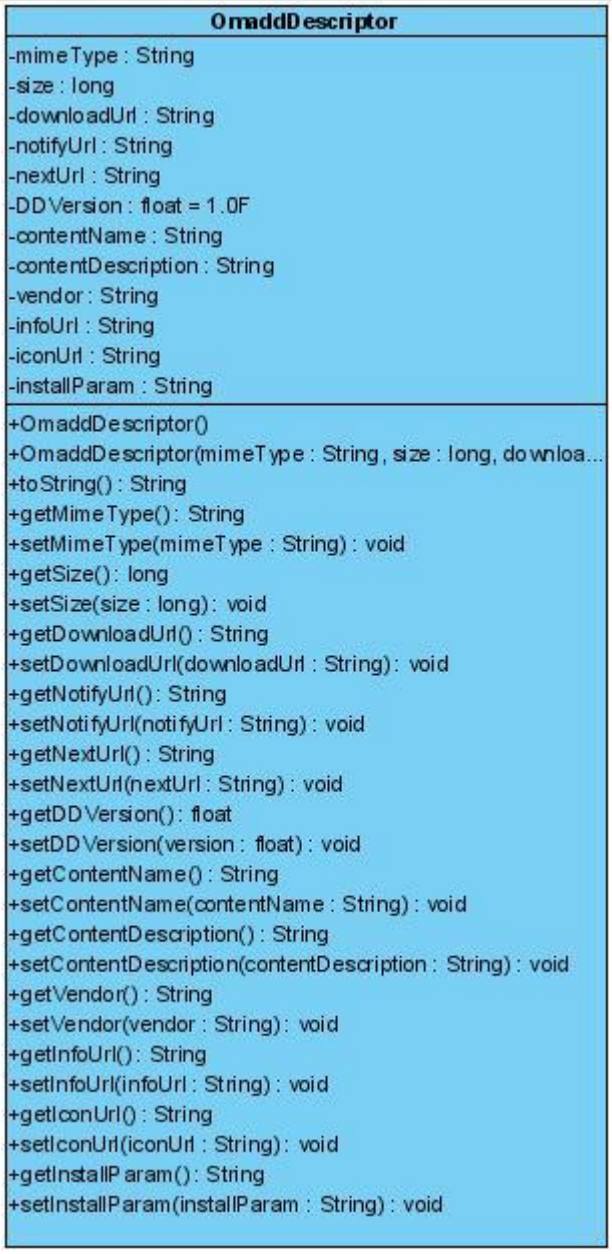



**Figura 41: Clase OmaddDescriptor**

| **Nombre:** OmaddDescriptor | |
|---|---|
| **Atributo** | **Tipo** |
| mimeType | String |
| size | long |
| downloadUrl | String |
| notifyUrl | String |
| nextUrl | String |
| DDVersion = 1.0F | float |
| contentName | String |
| contentDescription | String |
| vendor | String |
| infoUrl | String |
| iconUrl | String |
| installParam | String |
| **Para cada responsabilidad**: | |
| **Nombre:** | toString():void |
| **Descripción:** | Crea un XML que representa el descriptor del método OMA Download |

**Tabla 42: Descripción de la clase OmaddDescriptor**

**Clase DRMForwardLock**

**<CU_Entregar Contenido>**


**Figura 42: Clase DRMForwardLock**

| **Nombre:** DRMForwardLock |
|---|
| **Para cada responsabilidad**: |

| | |
|---|---|
| **Nombre:** | doGet(req: HttpServletRequest,resp:HttpServletResponse):void |
| **Descripción:** | Entrega el contenido con DRM ForwarLock al cliente que lo solicitó. |

**Tabla 43: Descripción de la clase DRMForwardLock**

| **Clase ContolExpiry**<br>**<CU_ Expirar Solicitudes >** |
|---|
|  |

**Figura 43: Clase ControlExpiry**

| **Nombre:** ControlExpiry | |
|---|---|
| **Atributo** | **Tipo** |
| | |
| **Para cada responsabilidad**: | |
| **Nombre:** | expiryOrder():void |
| **Descripción:** | Encargado de eliminar las órdenes cuyo tiempo de espera ha superado el tiempo establecido por la empresa para descargar el contenido. |

**Tabla 44: Descripción de la clase ControlExpiry**

**Clase ControlExpiryImpl**

**<CU_Expirar Solicitudes>**

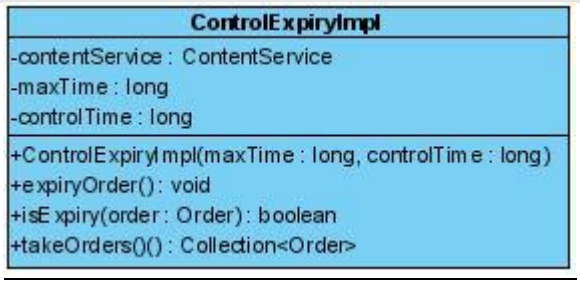



**Figura 44: Clase ControlExpiryImpl**

| **Nombre:** ControlExpiryImpl | |
|---|---|
| **Atributo** | **Tipo** |
| contentService | ContentService |
| maxTime | long |
| controlTime | long |
| **Para cada responsabilidad**: | |
| **Nombre:** | expiryOrder():void |
| **Descripción:** | Encargado de eliminar las órdenes cuyo tiempo de espera ha superado el tiempo establecido por la empresa para descargar el contenido. |
| **Nombre:** | takeOrders():Collection<Order> |
| **Descripción:** | Obtiene todas las órdenes registradas. |
| **Nombre:** | expiry(Long orderId):void |
| **Descripción:** | Elimina las órdenes que superan el tiempo máximo establecido para la descarga. |
| **Nombre:** | isExpiry(Order order): boolean |
| **Descripción:** | Verifica si la orden debe expirar. |

**Tabla 45: Descripción de la clase ControlExpiryImpl**

**Clase UrlUtils**

**<CU_Comunes>**

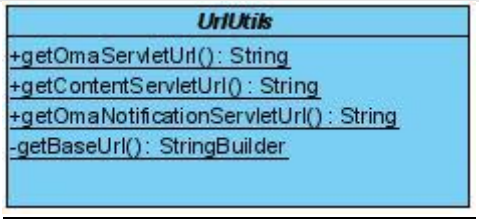



**Figura 45: Clase UrlUtils**

| **Nombre:** UrlUtils | |
|---|---|
| **Para cada responsabilidad**: | |
| **Nombre:** | getOmaServletUrl():String |
| **Descripción:** | Construye la URL que es constante para descargar el descriptor. |
| **Nombre:** | getContentServletUrl():String |
| **Descripción:** | Construye la URL que es constante para descargar el contenido. |
| **Nombre:** | getOmaNotificationServletUrl():String |
| **Descripción:** | Construye la URL que es constante para realizar la notificación de la descarga. |
| **Nombre:** | getBaseUrl():StringBuilder |
| **Descripción:** | Construye la URL base a partir de la cual se crean las demás URL. |

**Tabla 46: Descripción de la clase UrlUtils**

**Clase Configuration**

**<CU_Comunes>**

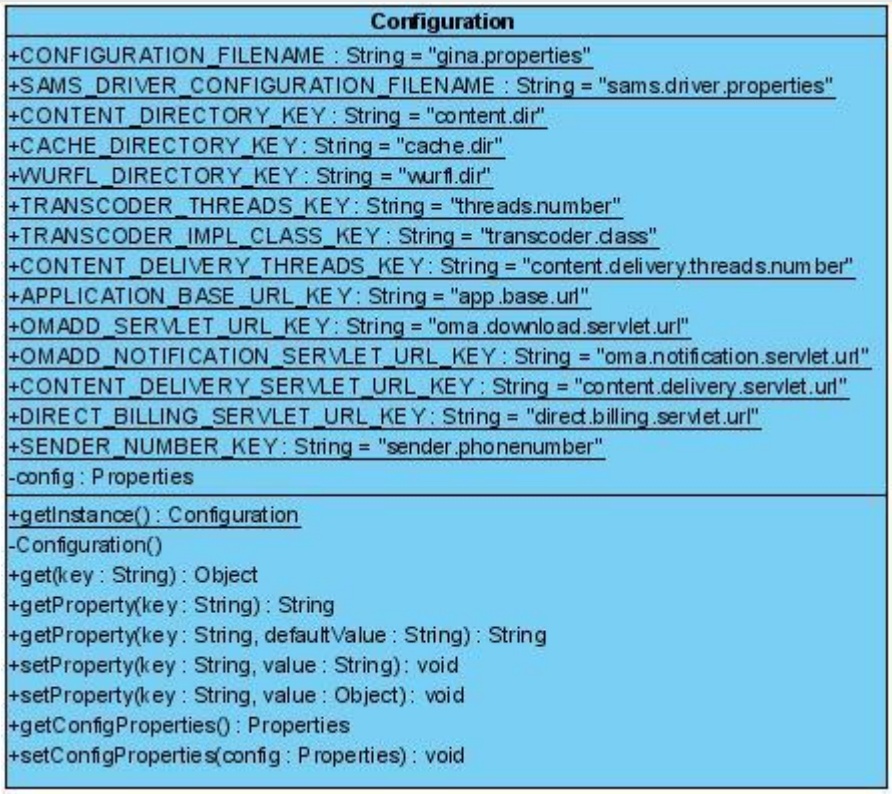



**Figura 46: Clase Configuration**

**Clase GinaContextListener**

**<CU_Comunes>**


**Figura 47: Clase GinaContextListener**

| **Nombre:** GinaContextListener | |
|---|---|
| **Para cada responsabilidad**: | |
| **Nombre:** | contextInitialized(ServletContextEvent event):void |
| **Descripción:** | Carga todos los datos correspondientes al atributo config de la clase Configuration |

**Tabla 47: Descripción de la clase GinaContextListener**

# Glosario de términos

**A**

APIs: Es un conjunto de convenciones internacionales que definen cómo debe invocarse una determinada función de un programa desde una aplicación.

**D**

Diagramas: Facilitan el entendimiento de grandes cantidades de datos y la relación entre diferentes partes de los datos.

**H**

Hardware: Incluye todas las partes físicas del computador, es decir, aquellos dispositivos que se conectan entre sí para formar una sola unidad de trabajo.

Herramienta Case: Son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el coste de las mismas en términos de tiempo y de dinero.

**S**

Software: Se refiere al equipamiento lógico o soporte lógico de un computador digital.

Servlets: Pequeño programa que corre en un servidor. Por lo general son aplicaciones Java que corren en un entorno de servidor web.

**M**

MMS: *Mutimedia Messaging Service* (sistema de mensajería multimedia), extensión lógica de SMS. MMS define un servicio que habilita sonidos, imágenes y video para ser combinados en el mensaje multimedia.

Máquina virtual de Java: Es responsable de la independencia de hardware y sistema operativo de la plataforma Java.

**P**

Plugins: Programa que puede anexarse a otro para aumentar sus funcionalidades (generalmente sin afectar otras funciones).

**S**

SMS**:** *Short Message Service*. Mensaje de hasta 160 caracteres, para ser enviados y recibidos a un teléfono móvil, la cantidad de caracteres la impone el operador celular.

**U**

URL: *Uniform Resource Locator,* es la dirección global de cualquier documento o recurso en la Web, visto como localizador.

**X**

XML: *Extensible Markup Language* (lenguaje extensible de etiquetas) Es un meta-lenguaje que permite definir lenguajes de marcado adecuado a usos determinados. Su función principal es describir datos y no mostrarlos.

**W**

WAP: *Wireless Application Protocol* (protocolo de aplicaciones inalámbricas), un estándar seguro que permite que los usuarios accedan a información de forma instantánea a través de dispositivos inalámbricos como teléfonos móviles.

Web Services: Es un sistema de *software* diseñado para apoyar interoperable máquina a máquina la interacción a través de una red.

WS-Security: Define cómo utilizar XML de cifrado y firma XML en SOAP para garantizar el intercambio de mensajes, como una alternativa al uso o la ampliación de HTTPS para asegurar el canal.