

Universidad de las Ciencias Informáticas



Universidad de las Ciencias
Informáticas

**“Plataforma de Gestión de Contenidos para
Dispositivos Móviles. Versión 2.0.”**

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

Autor(es): Dainelys Espinosa Marrero
Daniel Reinier Llanes Knight

Tutor: Ing. Lex Karel Zayas Hernández

Co-Tutor: Ing. Darién Jesús Álvarez de la Cruz

Ciudad de la Habana. Junio de 2010

“Año 52 de la Revolución”

Declaración de Autoría

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

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

Dainelys Espinosa Marrero

Firma del Autor

Daniel Reinier Llanes Knight

Firma del Autor

Ing. Lex Karel Zayas Hernández

Firma del Tutor

Ing. Darién Jesús Álvarez de la Cruz

Firma del Co-Tutor

Agradecimientos

Le agradezco a mi mamá Melba y a mi papá Ramón por su amor incondicional, por siempre saberme guiar y estar pendientes de mí, por haberme inculcado el amor al estudio, por enseñarme muchísimo de la vida y no dejarme caer ante los problemas. De ellos he aprendido todo lo que se hasta ahora, gracias por su protección, su apoyo y por ser excelentes padres, los amo muchísimo, a ustedes les debo todo lo que soy.

A mi abuela Tita por estar conmigo desde el primer día de mi vida, por seguirme en todos mis pasos. Por estar ahí cada vez que la he necesitado y por quererme mucho como yo también la quiero a ella.

A mi padrastro Humberto y mi tío Aleixi por quererme como una hija, por apoyarme y confiar en mí.

A mi primo Aleixito por quererme mucho y estar siempre ahí como el hermano que no tengo. A Leyanis por ser la hermanita grande con la que crecí.

A mi otra mamá Garde por estar orgullosa de mí y apoyar a mi mamá en estos momentos que no he podido estar a su lado.

A mis amigos y maestros del pre por enseñarme a ser una mejor persona. A todas las amistades y profesores de la universidad que no los menciono porque son muchos y puede ser que se me quede alguno, a ellos que se convirtieron en mi familia durante estos 5 años y que juntos pasamos momentos que nunca olvidaré aunque ya nos separemos.

A mi compañero de tesis por su dedicación y empeño en lograr la culminación de la tesis. A mi tutor Lex Karel y co-tutor Darién por su capacidad para guiarnos, por su ayuda y porque junto a ellos obtuvimos nuevos conocimientos.

En fin a todos aquellos que siempre me han querido y me han apoyado.

Dainelys

Agradecimientos

Le agradezco a mi mamá Ileana y a mi papá Reiniel por su eterno amor, por siempre estar junto a mí cuando más los necesitaba, por darme una excelente educación, por guiarme siempre por el camino correcto, a ustedes le debo en gran medida lo que soy hoy, por aguantarme tantas malcriadeces, los amo muchísimo . A mi hermana Daima por ese amor tan grande que me tiene, por aconsejarme y estar junto a mí en todo momento, te quiero muchísimo.

A mi abuela Iris y mi abuelo Reno por siempre confiar en mí, por estar a mi lado en cada momento, por quererme y apoyarme tanto, los quiero muchísimo también. A todos mis tíos en especial a Iyoleixis y Daniel por tratarme como un hijo, gracias por su amor incondicional. También destacar a mi tía Martha María por apoyarme y preocuparse tanto, los quiero mucho a todos.

A mi novia Marlis por apoyarme tanto, por ser esa persona que estuvo a mi lado siempre dándome amor y consejo, por guiarme en todo momento, por soportarme mis malacrianzas, por entregarse a mi, te amo muchísimo. A mis queridos suegros por confiar en mí y por darme todo su apoyo y cariño. A mis primos por siempre estar todos unidos, por poder contar con su apoyo y amor.

A mis amigos tanto los del barrio como los que tuve el privilegio de conocer en la escuela, resaltar entre ellos a Yamil, David Alejandro y Sandor que fueron como hermanos míos en estos 5 años de vida universitaria.

A mi compañera de tesis por todo su empeño y sacrificio durante el desarrollo de la tesis y por aguantarme todo este tiempo, gracias por todo. A mi tutor Lex Karel y co-tutor Darién por su capacidad para guiarnos, por su ayuda y porque junto a ellos obtuvimos nuevos conocimientos.

En fin a todos aquellos que siempre me han querido y me han apoyado para convertirme en un gran profesional.

Daniel

Dedicatoria

A mis padres Melba y Ramón, porque a ellos les debo todo lo que soy.

A mi abuela Tita por estar siempre a mi lado.

A toda mi familia por su apoyo y amor.

Dainelys

A mi abuelo Sini y mi abuela Martha que no se encuentran hoy junto a mí pero que hubiesen estado orgullosos con mis resultados.

A mis padres que son lo más importante de mi vida.

A mi hermana y a mi novia por estar siempre a mi lado.

A toda mi familia por su apoyo y amor.

Daniel

Resumen

Las tecnologías inalámbricas y fundamentalmente la telefonía celular ha presentado en el transcurso de los últimos años un gran avance lo cual ha convertido a los teléfonos celulares en una herramienta primordial para la comunicación. En la actualidad la mayoría de los operadores de telefonía móvil ofrecen un gran número de servicios, dentro de ellos la descarga de contenidos como imágenes, videos, tonos y logos es un ejemplo de que la telefonía celular se encuentra en la etapa expresiva. La unidad de negocios móvil de ETECSA (Cubacel) es la encargada de operar dicha tecnología en Cuba, sin embargo a pesar de los esfuerzos realizados no cuenta con una plataforma desplegada capaz de gestionar la descarga de contenidos multimedia para dispositivos móviles.

Por estas razones se propone el desarrollo de la Plataforma de Gestión de Contenidos para Dispositivos Móviles que tiene como objetivo proporcionar el espacio de trabajo necesario para cumplir con las funciones relacionadas con el proceso de descarga de contenidos, realizando una serie de actividades que garanticen la descarga segura y adaptación del contenido de acuerdo a las características del dispositivo móvil que solicite el servicio. Además brinda la posibilidad de poder visualizar dichos contenidos a través de un Portal WEB y destaca la gestión y almacenamiento de la información asociada a los contenidos mantenidos por el sistema.

Palabras claves:

Telefonía celular, Cubacel, Servicio de descarga de contenidos, Plataforma de Gestión de Contenidos para Dispositivos Móviles.

Índice

INTRODUCCIÓN	1
CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA.....	4
1.1. INTRODUCCIÓN.....	4
1.2. GESTIÓN DE CONTENIDOS PARA DISPOSITIVOS MÓVILES	4
1.3. PLATAFORMA DE GESTIÓN DE CONTENIDOS PARA DISPOSITIVOS MÓVILES	6
1.4. ESTADO DEL ARTE.....	6
1.4.1. <i>MOMA CDP</i>	7
1.4.2. <i>Be Downloads</i>	7
1.4.3. <i>PlayNow</i>	8
1.5. TECNOLOGÍAS Y HERRAMIENTAS A UTILIZAR	8
1.5.1. <i>Método de descarga empleado</i>	9
1.5.1.1. Descarga OMA	9
1.5.2. <i>DRM</i>	10
1.5.2.1. OMA DRM.....	10
1.5.3. <i>Transcoding</i>	11
1.5.3.1. Alembik.....	11
1.5.4. <i>Metodología de desarrollo de software</i>	12
1.5.4.1. Rational Unified Process (RUP).....	13
1.5.5. <i>Lenguaje de Modelado UML</i>	14
1.5.6. <i>Herramienta CASE para la modelación del sistema</i>	14
1.5.6.1. Visual Paradigm	14
1.5.7. <i>Plataforma de desarrollo</i>	15
1.5.7.1. Plataforma Java EE.....	15
1.5.8. <i>Máquina virtual de Java (JVM)</i>	16
1.5.9. <i>Lenguaje de programación: Java</i>	16
1.5.10. <i>Entorno de Desarrollo Integrado</i>	18
1.5.10.1. Eclipse.....	18
1.5.11. <i>Framework</i>	19
1.5.11.1. Spring.....	19

Índice de Contenidos

1.5.11.2.	Hibernate	21
1.5.12.	Especificaciones	21
1.5.12.1.	Java Persistence API (JPA)	21
1.5.13.	Sistema de control de versiones: SVN	22
1.5.14.	PostgreSQL servidor de base de datos	22
1.5.15.	Servidor Web: Apache Tomcat	23
1.6.	CONCLUSIONES DEL CAPÍTULO	24
CAPÍTULO II: CARACTERÍSTICAS DEL SISTEMA		25
2.1.	INTRODUCCIÓN	25
2.2.	PROPUESTA DEL SISTEMA	25
2.3.	MODELO DE DOMINIO	27
2.3.1.	Diagrama de Clases del dominio	28
2.4.	ESPECIFICACIONES DE REQUISITOS DEL SISTEMA	29
2.4.1.	Requisitos funcionales	29
2.4.2.	Requisitos no funcionales	31
2.5.	MODELO DE CASOS DE USOS DEL SISTEMA	34
2.5.1.	Actores del sistema	34
2.5.2.	Casos de Uso del sistema	35
2.5.3.	Diagrama de Paquetes de Casos de Usos del sistema	42
2.6.	CONCLUSIONES DEL CAPÍTULO	43
CAPÍTULO III: ANÁLISIS Y DISEÑO DEL SISTEMA		44
3.1.	INTRODUCCIÓN	44
3.2.	MODELO DE DISEÑO	44
3.2.1.	Diagramas de Interacción	45
3.2.2.	Diagrama de Clases del Diseño	45
3.3.	DESCRIPCIÓN DE LA ARQUITECTURA	50
3.3.1.	Patrones utilizados	50
3.3.2.	Patrón arquitectónico	50
3.3.3.	Patrones de diseño. Patrones GRASP	52
3.3.4.	Patrones de diseño. Patrones GOF	53

Índice de Contenidos

3.4.	MODELO DE DATOS.....	53
3.4.1.	<i>Modelo lógico de datos</i>	54
3.4.2.	<i>Modelo físico de datos</i>	55
3.5.	TRATAMIENTO DE EXCEPCIONES	56
3.6.	SEGURIDAD	56
3.7.	CONCLUSIONES DEL CAPÍTULO.....	56
CAPÍTULO IV: IMPLEMENTACIÓN.....		57
4.1.	INTRODUCCIÓN.....	57
4.2.	MODELO DE IMPLEMENTACIÓN.....	57
4.2.1.	<i>Diagrama de despliegue</i>	57
4.2.2.	DIAGRAMA DE COMPONENTES.....	58
4.3.	CONCLUSIONES DEL CAPÍTULO.....	62
CONCLUSIONES GENERALES		63
RECOMENDACIONES		65
REFERENCIAS BIBLIOGRÁFICAS		66

Introducción

Las tecnologías inalámbricas han presentado un gran auge en estos últimos años, principalmente en la rama de la telefonía celular. Martin Cooper fue el pionero de la misma, a él se le considera como "el padre de la telefonía celular " al introducir el primer radioteléfono, en 1973, en Estados Unidos. (1)

Desde finales de los 70 con el inicio de la telefonía celular las actividades que realizamos diariamente han revolucionado enormemente. Los teléfonos celulares se han convertido en una herramienta primordial para la gente común y de negocios a nivel mundial; las hace sentir más seguras y más productivas.

Esta tecnología brinda un amplio número de posibilidades como es el caso del envío de SMS (por sus siglas en inglés Short Message Service), MMS (por sus siglas en inglés Multimedia Messaging Service), 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, juegos y videos.

Ningún país puede desarrollarse hoy de espaldas a las Nuevas Tecnologías de la Información y las Comunicaciones (TICs), las cuales han significado una verdadera revolución en el decursar de la humanidad.

Cuba cuenta con una recién modernizada red de telefonía digital que cubre todo el país y a pesar de que todavía dichos avances no pueden llegar a todas partes y de que el número de líneas tanto fijas como móviles sea aún bastante bajo, lo cierto es que se está intentando avanzar en esta rama.

La unidad de negocios móvil de ETECSA (Cubacel) es la encargada de operar la telefonía celular en Cuba, sin embargo, a pesar de los esfuerzos realizados no ha sido posible que se brinden en su totalidad los servicios que en el mundo contemporáneo se consumen, en la isla todavía nos encontramos atrasados en dicho aspecto pues solamente contamos con los mínimos servicios que se les pueden brindar a los clientes, nos falta mucho para poder llegar al desarrollo que ha alcanzado la telefonía digital a nivel mundial, ejemplo de ello es un servicio que garantice la entrega de contenidos hacia los clientes.

Capítulo I: Fundamentación Teórica

Como parte del desarrollo que se está llevando a cabo en Cuba en cuanto al tema de las redes de telecomunicaciones así como los servicios que sobre estas se pueden brindar, la Empresa de Telecomunicaciones de Cuba S.A le asignó la responsabilidad a la Universidad de las Ciencias Informáticas del desarrollo de un software que brindara los servicios necesarios para la gestión de contenidos en teléfonos celulares.

Para darle solución a dicho pedido se elaboró en el año 2008-2009 una Plataforma de Gestión de Contenidos para Dispositivos Móviles, encargada de gestionar los procesos de descarga de contenidos. Sin embargo, dicho proyecto presentó algunas deficiencias que conllevaron a desarrollar otra aplicación donde se corrigieran los errores llevados a cabo anteriormente, dado esto surge la versión 2.0 de dicho producto.

A partir de dicha decisión se plantea como **Problema Científico**: ¿Cómo proveer contenidos multimedia a dispositivos móviles?

Para darle solución al problema planteado se define como **objetivo general** desarrollar una plataforma que almacene contenidos multimedia de varios proveedores y que realice las operaciones necesarias para adaptar estos contenidos a las características de los dispositivos de los clientes, protegiendo los derechos de los proveedores sobre el contenido.

Teniendo en cuenta lo planteado anteriormente el **objeto de estudio** se centra en los procesos de gestión de contenidos, siendo el **campo de acción** los procesos de gestión de contenidos multimedia para dispositivos móviles.

Para cumplir con el objetivo propuesto y resolver la situación problémica mencionada, se plantea un grupo de **tareas de investigación**:

- Desarrollar un nuevo método de descarga.
- Integrar la nueva versión de Alembik.
- Desarrollar los métodos de gestión de derechos digitales (por sus siglas en inglés DRM; de Digital Rights Management)

Capítulo I: Fundamentación Teórica

- Modificar la BD para incluir las nuevas funcionalidades.
- Desarrollar la seguridad del sistema.
- Complementar las funcionalidades de la Plataforma.

Con el cumplimiento de estas tareas se pretende proporcionarle a la unidad de negocios móvil de ETECSA una plataforma capaz de administrar contenidos multimedia como imágenes, videos, música, juegos y temas de dispositivos móviles a partir de un portal Web y un portal Wap. Además debe ser capaz de aplicarle derechos digitales en los contenidos que sean posibles.

El presente documento está compuesto por 4 Capítulos:

En el **capítulo 1** “Fundamentación Teórica” se exponen los fundamentos generales que sirven de soporte teórico en la solución del problema. Se analizan las herramientas y lenguajes de programación, para comprobar que son las idóneas para la implementación de la Plataforma. Además se plantea la metodología a emplear en el desarrollo del mismo.

En el **capítulo 2** “Características del Sistema” se describe detalladamente el problema existente y las características del sistema a desarrollar. Contiene la descripción de los principales conceptos mediante un Modelo de Dominio, la captura de requisitos funcionales y no funcionales, así como el diagrama de casos de usos del sistema y las descripciones de los mismos.

En el **capítulo 3** “Análisis y Diseño” se construye la solución propuesta y se modelan diagramas de clases que representan las funcionalidades de la Plataforma de Gestión de Contenidos aplicando los patrones de arquitectura y diseño seleccionados.

En el **capítulo 4** “Implementación” se exponen las principales características del flujo de trabajo de implementación, representando el diagrama de despliegue y el de componentes.

Capítulo I: Fundamentación Teórica

1.1. Introducción

El objetivo fundamental de este capítulo es realizar un estudio valorativo sobre el proceso de Gestión de Contenidos para Dispositivos Móviles. Se incluye el estado actual en el mundo de dicho proceso, así como las tendencias tecnológicas y metodológicas más usadas para su solución. Además se abordan aspectos de los distintos tipos de herramientas y tecnologías que se emplean en el desarrollo del sistema propuesto.

1.2. Gestión de contenidos para dispositivos móviles

Actualmente los teléfonos móviles se caracterizan entre otras cuestiones por la operación de voz y datos en el mismo dispositivo, lo cual permitirá la sincronización de agendas, tener acceso a Internet, recibir archivos, mensajes escritos y correos electrónicos; en otras palabras, está apta para aplicaciones multimedia y altas transmisiones de datos. Los protocolos empleados en los sistemas de tercera generación, han traído consigo la aparición de tecnologías de comunicación que permitirán llevar voz, gráficos, videos y otros tipos de datos de banda ancha directamente al usuario, independientemente de la red o dispositivo móvil utilizado.

Los dispositivos móviles son aparatos de pequeño tamaño, con algunas capacidades de procesamiento, con conexión permanente o intermitente a una red, con memoria limitada, diseñados específicamente para una función, pero que pueden llevar a cabo otras funciones más generales. Una característica importante es el concepto de movilidad: dado que son lo suficientemente pequeños para ser transportados y empleados durante su transporte. Su función esencial es la comunicación de voz, como el teléfono convencional y su rápido desarrollo ha incorporado otras funciones como son cámara fotográfica, agenda, acceso a Internet, reproducción de video e incluso GPS y reproductor mp3.

Capítulo I: Fundamentación Teórica

Otras de sus características son: pantalla significativamente más pequeña que la de un ordenador personal, teclados más limitados que los de un ordenador, limitaciones en la memoria disponible, tanto memoria RAM como memoria para almacenamiento persistente y limitaciones en la capacidad del procesador, en comparación con la memoria y procesador de un ordenador personal típico.

Los servicios de valor agregado (VAS) son todos aquellos servicios adicionales a los servicios de voz dentro de la telefonía celular. Generan entre el 20% y 25% de los ingresos de las compañías de telefonía celular a nivel mundial. (2) En la actualidad la mayoría de los operadores de telefonía móvil ofrecen un gran número de nuevos servicios: promociones, meteorológicos, noticias, juegos, acceso a bancos, localización, información de tráfico, información turística, restaurantes, entre otros. De ellos, la descarga de contenidos como imágenes, videos, tonos y logos es el ejemplo más claro de que la comunicación por el móvil se encuentra en la etapa expresiva.

Dentro de los dispositivos un contenido se define como un objeto que se le ofrece al cliente móvil. Son entregados al mismo como resultado del acceso a un servicio brindado por los operadores de la telefonía celular, así puede mencionarse por ejemplo, como resultado del servicio de meteorología un parte meteorológico para un momento determinado; del servicio de noticias una noticia nacional, internacional, deportiva o de otra índole y del servicio de descarga un tono, un logo u otro contenido.

Los operadores necesitan coordinar todos los recursos disponibles, para hacerlos llegar hasta el cliente móvil. Esto implica fuertes interacciones entre el entorno (cliente móvil), las estructuras (aplicaciones para la gestión), los productos (contenidos para dispositivos móviles), y el proceso (se encarga de unir el entorno, las estructuras y los productos). Toda la secuencia de actividades desarrollada para lograr este objetivo, se conoce como proceso de gestión de contenidos para dispositivos móviles.

1.3. Plataforma de Gestión de Contenidos para Dispositivos Móviles

La Plataforma de Gestión de Contenidos para Dispositivos Móviles tiene como objetivo proporcionar el espacio de trabajo necesario para cumplir con las funciones relacionadas con el proceso de descarga de contenidos hacia el dispositivo móvil.

Permitirá llevar a cabo la descarga de un contenido hacia un dispositivo realizando una serie de actividades que garanticen la descarga y adaptación del contenido al dispositivo móvil que solicite el servicio.

Además brinda la posibilidad de poder visualizar dichos contenidos a través de un Portal WEB y destaca la gestión y almacenamiento de la información asociada a los contenidos mantenidos por el sistema. Toda esa información queda almacenada en el repositorio, bien como registros de la base de datos o bien como ficheros estáticos accesibles por el sistema.

1.4. Estado del arte

En el mundo existen innumerables empresas que se han dedicado a la gestión de entrega de contenidos para móviles a partir de plataformas on-line, aunque cada una cuenta con sus particularidades muchas han logrado un producto con gran calidad que las ha llevado a liderar en este negocio. Dentro de estas plataformas podemos mencionar “MOMA CDP” (<http://www.m-wise.com/7.html>), de la empresa M-Wise, “Play Now” (<http://www.playnow-arena.com>) un sistema creado por Sony Ericsson y “Be Downloads” (<http://spiralis.com.mx/plataformas.php>), perteneciente a la empresa Spiralís.

En Cuba, el uso de dispositivos móviles ha contado con un pobre y menos vertiginoso desarrollo que en el resto del mundo, por esta razón la demanda de descarga de contenidos para dichos dispositivos es muy poca y la VPSM (“*Vicepresidencia de Servicios Móviles de ETECSA*”) no cuenta con una plataforma de descarga de contenidos para móviles.

1.4.1. MOMA CDP

MOMA Content Management and Delivery Platform ("MOMA CDP") es un sistema integrado que permite la fácil gestión, administración, comercialización, publicación, entrega, facturación e inventario de contenidos digitales.

La plataforma incluye una amplia gama de servicios y beneficios comerciales para sus proveedores de contenidos. Posee interfaces para socios comerciales como Operadores Móviles, Proveedores de Contenidos, Proveedores de Servicios y Proveedores de Medios de Facturación, para de conjunto gestionar y entregar contenido a los usuarios finales a través de múltiples canales, entre ellos las redes móviles e Internet.

Cuenta con una infraestructura modular que permite la creación de reportes, además facilita la clasificación de los contenidos, la atención al cliente y la gestión de derechos digitales sobre los contenidos.

Incluye además facilidades en la gestión de los contenidos por parte de los proveedores y permite la transcodificación de los contenidos en correspondencia con el usuario que lo solicita.

1.4.2. Be Downloads

La Empresa Spiralís cuenta con varias plataformas, una de ellas es Be Downloads que permite la administración, venta y entrega de contenidos variados, innovadores y personalizados. Al mantener una actualización constante en sus bases de terminales, garantizan que el contenido sea entregado en el formato adecuado.

Una de las características fundamentales con que cuenta la plataforma es su potente herramienta de estadísticas, además permite administrar el contenido para diferentes operadores (esquema proveedor o agregador) por medio de ofertas de contenidos propios e independientes.

Facilita el manejo de cualquier canal de venta en el esquema agregador (SMS, WAP, SIM Browsing). Cuenta con un manejo de reportes por perfil de usuario, una herramienta de atención a clientes donde

muestra todas sus transacciones y también permite determinar el formato que deberá llevar el contenido a la hora de la descarga teniendo en cuenta el modelo del dispositivo que realizó la petición. Posee un alto grado de seguridad a la hora de la descarga y un cobro inteligente.

1.4.3. PlayNow

Sony Ericsson es una prestigiosa compañía de teléfonos móviles que cuenta con su propia plataforma de descarga de contenidos para los teléfonos de su propia marca. La plataforma cuenta con servicios de transcodificación de contenidos permitiendo adaptar el contenido acorde con las características del destino final.

Permite la comercialización, entrega y facturación de los contenidos que gestiona, además otorga gestión de derechos digitales a los contenidos de sus proveedores. Se encuentra integrado con un sistema de descarga denominado PlayNow Plus. Además como parte de los servicios que brinda están también PlayNowTrackID y PlayNow movies, el primero permite el reconocimiento de música y el segundo consiste en un sistema de descarga de hasta 60 películas que le permiten al usuario visualizar el contenido solamente en su dispositivo y por el tiempo de 90 días, actualmente este servicio no se ofrece en todos los países.

1.5. Tecnologías y herramientas a utilizar

Para el desarrollo de la Plataforma de Gestión de Contenidos para Dispositivos Móviles se hace necesaria la automatización de varios procesos. Por dicha razón haciendo un análisis exhaustivo se llegaron a seleccionar un grupo de herramientas necesarias para el desarrollo del sistema, las cuales se detallan con claridad a continuación.

1.5.1. Método de descarga empleado

El acceso a los 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. Este último método ha sido creado por la *Open Mobile Alliance* (<http://www.openmobilealliance.org/>) organización creada como respuesta a la proliferación de los foros de la industria encargados de algunos protocolos.

1.5.1.1. Descarga OMA

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

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.

Entre los posibles elementos a ser incluidos en el descriptor se encuentran tipo, tamaño y object URI (URL de la cual el objeto será cargado).

1.5.2. DRM

La Gestión de derechos digitales (por sus siglas en inglés DRM; de Digital Rights Management) 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.

Existen diferentes mecanismos de DRM, diseñados por distintas empresas, pero en general todos tienen en común algunas características:

- detectan quién accede a cada obra, cuándo y bajo qué condiciones, y reportan 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.
- 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. (3)

1.5.2.1. OMA DRM

El mecanismo de protección de contenido más difundido en Internet Móvil actualmente es el de la Open Mobile Alliance (OMA). El proceso de estandarización tiene como objetivo homogeneizar los aspectos relacionados con el consumo controlado de contenidos digitales.

OMA DRM se centra primordialmente en el control de copia y proporciona mecanismos básicos de gestión de las claves para desbloquear la copia de contenido.

Los tres 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.
(4)

Puede afirmarse que el DRM ha alcanzado la madurez para garantizar la adecuada protección de los contenidos y los derechos de propiedad intelectual. (5)

1.5.3. Transcoding

Transcoding es un servicio que se brinda para adaptar los archivos digitales de manera que los contenidos se puedan ver en diferentes dispositivos de reproducción. Trabajando como un intérprete, un transcoder traduce los archivos a un formato adecuado para el usuario final. Las traducciones se basan en complicados cálculos algorítmicos y requieren importantes recursos de procesamiento. (6)

En el país no existe una herramienta capaz de realizar el proceso de transcoding para servicios de telefonía móvil, que sea capaz de adaptarse a las necesidades correspondientes del proyecto que se está analizando, por lo que se hace necesario escoger la que se utilizará entre: Media Coder, Transcoder 4.0.1.40, VoDKABatchTranscoder y Alembik que son las que se utilizan a nivel mundial.

1.5.3.1. Alembik

Alembik es una aplicación servidor J2EE que presta servicios de transcodificación a una variedad de clientes. Es una herramienta de código abierto, libre, totalmente compatible con la especificación de la Interfaz de Transcodificación Estándar de OMA. Su arquitectura por capas es independiente de contenedor, extensible y altamente escalable (clúster de servidores, almacén de ficheros compartido).

La aplicación ofrece una amplia gama de transformaciones predefinidas de contenido media, resolución automática del User Agent (Agente de Usuario) a los parámetros de transcodificación, procesamiento y control de tareas asíncronas, acceso sincronizado a ficheros, transmisión al punto de video y muchas otras características. También incluye un motor de renderización web completamente desarrollado (transcodificación de HTML). Toda la funcionalidad se pone a disposición de los clientes a través de cuatro API (del inglés application programming interface) diferentes:

- la fachada del EJB (*Enterprise JavaBean*) de sesión
- el servicio web SOAP (*Simple Object Access Protocol*)
- el servicio HTTP
- la librería de etiquetas JSP (7)

1.5.4. Metodología de desarrollo de software

Las metodologías de desarrollo de software son un conjunto de procedimientos, técnicas, herramientas y un soporte documental que ayuda a los desarrolladores a realizar un software. Imponen un proceso disciplinado sobre el desarrollo de software con el fin de hacerlo más predecible y eficiente. Lo hacen desarrollando un proceso detallado con un fuerte énfasis en planificar.

En un proyecto de desarrollo de software la metodología define Quién debe hacer Qué, Cuándo y Cómo debe hacerlo. Una metodología es un proceso.

No existe una metodología de software universal. Las características de cada proyecto (equipo de desarrollo, recursos, etc.) exigen que el proceso sea configurable.

En la actualidad existen varias metodologías orientadas a objetos basadas en UML (*Unified Modeling Language*):

- Rational Unified Process (RUP)
- OPEN

- MÉTRICA 3

1.5.4.1. Rational Unified Process (RUP)

Para controlar, planificar, organizar y guiar el desarrollo de la Plataforma se decidió utilizar como metodología de desarrollo del software RUP (Proceso Unificado de Desarrollo de Software).

RUP es una infraestructura flexible de desarrollo de software que proporciona prácticas recomendadas probadas y una arquitectura configurable. Es el resultado de varios años de desarrollo y uso práctico en el que se han unificado técnicas de desarrollo, a través del UML y trabajo de muchas metodologías utilizadas por los clientes.

Como RUP es un proceso, en su modelación define como sus principales elementos: Trabajadores (“quién”), Actividades (“cómo”), Artefactos (“qué”) y Flujo de actividades (“Cuándo”).

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

Cada una de estas etapas es desarrollada mediante el ciclo de iteraciones, la cual consiste en reproducir el ciclo de vida en cascada a menor escala. Los objetivos de una iteración se establecen en función de la evaluación de las iteraciones precedentes.

Una particularidad de esta metodología es que, en cada ciclo de iteración, se hace exigente el uso de artefactos, siendo por este motivo, una de las metodologías más importantes para alcanzar un grado de certificación en el desarrollo del software. (8)

La Metodología RUP es más adaptable para proyectos de largo plazo. Está pensado para proyectos y equipos grandes o muy largos, genera gran cantidad de documentación y es considerado un proceso pesado. Ambas características aportan estabilidad al equipo de desarrollo, debido al tamaño del equipo y la poca interacción entre ellos se utiliza la documentación generada por RUP para el estudio y actualización del proceso de desarrollo del sistema.

1.5.5. Lenguaje de Modelado UML

Se utilizó como lenguaje de modelado UML (*Unified Modeling Language*) con el que se puede especificar, construir, visualizar y documentar los artefactos de un sistema de software orientado a objetos (OO). Un artefacto es una información que es utilizada o producida mediante un proceso de desarrollo de software.

El UML está compuesto por diversos elementos gráficos que se combinan para conformar diagramas. Debido a que el UML es un lenguaje, cuenta con reglas para combinar tales elementos.

Es importante destacar que un modelo UML describe lo que supuestamente hará un sistema, pero no dice cómo implementar dicho sistema.

1.5.6. Herramienta CASE para la modelación del sistema.

Las herramientas 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. (9)

Estas herramientas permitirán organizar y manejar la información de un proyecto informático. Permittedole a los participantes de un proyecto, que los sistemas (especialmente los complejos), se tornen más flexibles, más comprensibles y además mejorar la comunicación entre los participantes.

1.5.6.1. Visual Paradigm

La herramienta case que se utilizará será Visual Paradigm (<http://www.visual-paradigm.com>). La misma es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El software de modelado UML ayuda a una más rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. La herramienta UML CASE también proporciona abundantes tutoriales de UML, demostraciones interactivas de UML y proyectos UML. (10)

Se decidió utilizar esta herramienta CASE para el modelado del diseño de la aplicación, ya que entre sus principales características y facilidades se encuentran: multiplataforma, producto de calidad, varios

idiomas, generación de código para Java y exportación como HTML y se integra con herramientas Java como Eclipse.

1.5.7. Plataforma de desarrollo

Una plataforma de desarrollo es el entorno común en el cual se desenvuelve la programación de un grupo definido de aplicaciones.

1.5.7.1. Plataforma Java EE

La plataforma Java es el nombre de un entorno o plataforma de computación originaria de Sun Microsystems, capaz de ejecutar aplicaciones desarrolladas usando el Lenguaje de programación Java

La plataforma Java incluye:

- Plataforma Java, Edición Estándar (Java Platform, Standard Edition), o Java SE.
- Plataforma Java, Edición Empresa (Java Platform, Enterprise Edition), o Java EE.
- Plataforma Java, Edición Micro (Java Platform, Micro Edition), o Java ME.

Para el desarrollo de la aplicación se seleccionó la Edición Empresarial: J2EE o Java EE (Java Platform, Enterprise Edition).

Java Platform, Enterprise Edition o Java EE, es una plataforma de programación para desarrollar y ejecutar software de aplicaciones en el lenguaje de programación Java con arquitectura de n niveles distribuidos basándose ampliamente en componentes de software modulares ejecutados sobre un servidor de aplicaciones. Incluye varias especificaciones de API, tales como EJB, Servlets (código en Java que se ejecuta en un servidor web), Portlets, JSP y varias tecnologías de servicios web.

Java Enterprise Edition (Java EE) es el estándar de la industria para desarrollar aplicaciones Java portables, robustas, escalables y seguras en el lado del servidor (server-side). Java EE proporciona APIs para servicios web, modelo de componentes, gestión y comunicación que hacen lo convierten en el estándar de la industria para implementar aplicaciones web y web 2.0 y aplicaciones con arquitectura orientada a servicios (SOA). (11)

Java EE permite al desarrollador crear una aplicación de empresa que es portable entre plataformas y escalable. Otros beneficios añadidos son, por ejemplo, que el servidor de aplicaciones puede manejar las transacciones, seguridad, escalabilidad, concurrencia y gestión de los componentes que son desplegados, significando que los desarrolladores pueden concentrarse más en la lógica de negocio de los componentes en lugar de las tareas de mantenimiento de bajo nivel.

Los componentes principales de la plataforma son la Máquina Virtual de Java (JVM) y el lenguaje de programación Java.

1.5.8. Máquina virtual de Java (JVM)

El término JVM se refiere a la especificación abstracta de una máquina de software para ejecutar programas Java. Es el núcleo de ese lenguaje de programación, por lo que resulta imposible ejecutar cualquier programa Java sin la ejecución de alguna implantación de la JVM, o sea, el código no se ejecuta directamente sobre un procesador físico, sino sobre un procesador virtual Java. Es la encargada de traducir los bytecode (código resultante de la compilación del código fuente) en las instrucciones nativas. La gran ventaja de la máquina virtual de java es aportar portabilidad a las aplicaciones.

1.5.9. Lenguaje de programación: Java

El lenguaje de programación Java, fue diseñado por la compañía Sun Microsystems Inc, con el propósito de crear un lenguaje que pudiera funcionar en redes computacionales heterogéneas (redes de computadoras formadas por más de un tipo de computadora, ya sean PC, MAC's, estaciones de trabajo, etc), y que fuera independiente de la plataforma en la que se vaya a ejecutar. Esto significa que un programa de Java puede ejecutarse en cualquier máquina o plataforma. El lenguaje fue diseñado con las siguientes características en mente:

El lenguaje de programación Java es el único soportado por la inmensa mayoría de dispositivos móviles. (12) Esto ha provocado que se haya convertido en la Plataforma de Desarrollo preferida por programadores de todo el mundo.

- **Simple:** Elimina la complejidad de los lenguajes como "C" y da paso al contexto de los lenguajes modernos orientados a objetos. La filosofía de programación orientada a objetos es diferente a la programación convencional.
- **Familiar:** Como la mayoría de los programadores están acostumbrados a programar en C o en C++, la sintaxis de Java es muy similar al de estos.
- **Robusto:** El sistema de Java maneja la memoria de la computadora por ti. No te tienes que preocupar por apuntadores, memoria que no se esté utilizando, etc. Java realiza todo esto sin necesidad de que uno se lo indique.
- **Seguro:** El sistema de Java tiene ciertas políticas que evitan se puedan codificar virus con este lenguaje. Existen muchas restricciones, especialmente para los applets, que limitan lo que se puede y no puede hacer con los recursos críticos de una computadora.
- **Portable:** Como el código compilado de Java (conocido como bytecode) es interpretado, un programa compilado de Java puede ser utilizado por cualquier computadora que tenga implementado el intérprete de Java.
- **Independiente a la arquitectura:** Al compilar un programa en Java, el código resultante es un tipo de código binario conocido como bytecode. Este código es interpretado por diferentes computadoras de igual manera, solamente hay que implementar un intérprete para cada plataforma. De esa manera Java logra ser un lenguaje que no depende de una arquitectura computacional definida.
- **Multithreaded:** Un lenguaje que soporta múltiples hilos es un lenguaje que puede ejecutar diferentes líneas de código al mismo tiempo.
- **Interpretado:** Java corre en máquina virtual, por lo tanto es interpretado.
- **Dinámico:** Java no requiere que compile todas las clases de un programa para que este funcione. Si realizas una modificación a una clase Java se encarga de realizar un Dynamic Binding o un Dynamic Loading para encontrar las clases. (13)

Por todas las características anteriormente mencionadas, es que se propone el empleo de esta plataforma que integra todo lo necesario para que el desarrollo de la aplicación en cuestión sea favorable, sin costes elevados y ajustado a cronogramas.

1.5.10. Entorno de Desarrollo Integrado.

UN IDE (Integrated Development Environment) es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica (GUI).

Los IDEs son un conjunto de herramientas para el programador, que suelen incluir en una misma suite, un buen editor de código, administrador de proyectos y archivos, enlace transparente a compiladores y debuggers e integración con sistemas controladores de versiones o repositorios. (14)

1.5.10.1. Eclipse

Eclipse es una plataforma de desarrollo open source basada en Java. Es un desarrollo de IBM cuyo código fuente fue puesto a disposición de los usuarios. En sí mismo Eclipse es un marco y un conjunto de servicios para construir un entorno de desarrollo a partir de componentes conectados (plug-in). (15)

Eclipse contiene una serie de perspectivas. Cada perspectiva proporciona una serie de funcionalidades para el desarrollo de un tipo específico de tarea. Por ejemplo la perspectiva Java combina un conjunto de vistas que permiten ver información útil cuando se está escribiendo código fuente, mientras que la perspectiva de depuración contiene vistas que muestran información útil para la depuración de los programas Java. (15)

Para el trabajo con Spring se utilizó Spring Source Tool Suite el cual provee el mejor y el más completo conjunto de herramientas para la creación de aplicaciones Spring.

Spring Source Tool Suite es completamente gratuito e incorpora una serie de asistentes y herramientas visuales orientadas a facilitar y agilizar el desarrollo de aplicaciones que utilizan tecnologías como Spring Framework y Spring Web Flow, además es una herramienta mucho más completa que el pluging Spring IDE. (16)

1.5.11. Framework

Es una estructura de software compuesta de componentes personalizables e intercambiables para el desarrollo de una aplicación. En otras palabras, un framework se puede considerar como una aplicación genérica incompleta y configurable a la que podemos añadirle las últimas piezas para construir una aplicación concreta.

Los objetivos principales que persigue un framework son: acelerar el proceso de desarrollo, reutilizar código ya existente y promover buenas prácticas de desarrollo como el uso de patrones.

Son diseñados con la intención de facilitar el desarrollo de software, permitiendo a los diseñadores y programadores pasar más tiempo identificando requerimientos de software que tratando con los tediosos detalles de bajo nivel de proveer un sistema funcional.

1.5.11.1. Spring

El Spring Framework (también conocido como Spring) es un framework de código abierto de desarrollo de aplicaciones para la plataforma Java. Por su diseño el framework ofrece mucha libertad a los desarrolladores en Java y soluciones muy bien documentadas y fáciles de usar para las prácticas comunes en la industria.

Spring contiene muchas características que le dan una funcionalidad muy basta; dichas características están organizadas en siete grandes módulos.

- El módulo Core o "Núcleo" es la parte fundamental del framework ya que provee toda la funcionalidad de Inyección de Dependencias permitiéndote administrar la funcionalidad del contenedor de beans. El concepto básico de este módulo es el BeanFactory, que implementa el patrón de diseño Factory (fábrica) eliminando la necesidad de crear singletons programáticamente permitiéndote desligar la configuración y especificación de las dependencias de tu lógica de programación.
- Encima del módulo core se encuentra el módulo Context (Contexto), el cual te provee de herramientas para acceder a los beans de una manera elegante, similar a un registro JNDI (Java

Naming and Directory Interface). El paquete de contexto hereda sus características del paquete de beans y añade soporte para mensajería de texto, como son resource bundles (para internacionalización), propagación de eventos, carga de recursos y creación transparente de contextos por contenedores (como el contenedor de servlets, por ejemplo).

- El paquete DAO provee una capa de abstracción de JDBC que elimina la necesidad de teclear código JDBC tedioso y redundante así como el parseo de códigos de error específicos de cada proveedor de base de datos. También, el paquete JDBC provee de una manera de administrar transacciones tanto declarativas como programáticas, no solo para clases que implementen interfaces especiales, pero para todos tus POJOs (por sus siglas en inglés, Viejos y simples objetos java).
- El paquete ORM provee capas de integración para APIs de mapeo objeto - relacional, incluyendo, JDO, Hibernate e iBatis. Usando el paquete ORM tú puedes usar esos mapeadores en conjunto con otras características que Spring ofrece, como la administración de transacciones mencionada con anterioridad.
- El paquete AOP provee una implementación de programación orientada a aspectos compatibles con AOP Alliance, permitiéndote definir pointcuts e interceptores de métodos para desacoplar el código de una manera limpia implementando funcionalidad que por lógica y claridad debería estar separada. Usando metadatos a nivel de código fuente se pueden incorporar diversos tipos de información y comportamiento al código, un poco similar a los atributos de .NET
- El paquete Web provee características básicas de integración orientadas a la web, como funcionalidad multipartes (para realizar la carga de archivos), inicialización de contextos mediante servlet listeners y un contexto de aplicación orientado a web. Cuando se usa Spring junto con WebWork o Struts, este es el paquete que te permite una integración sencilla.
- El paquete Web MVC provee de una implementación Modelo - Vista - Controlador para las aplicaciones web. La implementación de Spring MVC permite una separación entre código de modelo de dominio y las formas web y permite el uso de otras características de Spring Framework como lo es la validación. (17)

Por todas las características mencionadas anteriormente, es posible afirmar que Spring es un framework muy simple, conveniente y flexible, pero al mismo tiempo muy poderoso; facilita la manipulación de objetos, elimina la necesidad de usar distintos y variados tipos de ficheros de configuración, mejora la práctica de programación y suaviza la curva de aprendizaje favorablemente para el desarrollador; es por ello que se propone como una alternativa viable para el desarrollo de la aplicación.

1.5.11.2. Hibernate

Hibernate es una herramienta que realiza el mapeo entre el mundo orientado a objetos de las aplicaciones y el mundo entidad-relación de las bases de datos en entornos Java. El término utilizado es ORM (object/relational mapping) y consiste en la técnica de realizar la transición de una representación de los datos de un modelo relacional a un modelo orientado a objetos y viceversa.

Hibernate no solo realiza esta transformación sino que nos proporciona capacidades para la obtención y almacenamiento de datos de la base de datos que nos reducen el tiempo de desarrollo. (17)

Generador de sentencias sql, permite diseñar objetos persistentes que podrán incluir polimorfismo, relaciones, colecciones, y un gran número de tipos de datos. De una manera muy rápida y optimizada puede generarse la base de datos en cualquiera de los entornos soportados. (18)

1.5.12. Especificaciones

1.5.12.1. Java Persistence API (JPA)

Java Persistence API (JPA) proporciona un estándar para gestionar datos relacionales en aplicaciones Java SE o Java EE, de forma que además se simplifique el desarrollo de la persistencia de datos.

En su definición, ha combinado ideas y conceptos de los principales frameworks de persistencia, como Hibernate, Toplink y JDO, y de las versiones anteriores de EJB. Todos estos cuentan actualmente con una implementación JPA.

El mapeo objeto-relacional (es decir, la relación entre entidades Java y tablas de la base de datos, *queries* con nombre, etc) se realiza mediante anotaciones en las propias clases de entidad. No se requieren ficheros descriptores XML. También pueden definirse transacciones como anotaciones JPA. (19)

1.5.13. Sistema de control de versiones: SVN

Subversion es uno de los sistemas de control de versiones más modernos y utiliza un sistema con repositorio centralizado. El control de versiones se basa en una serie de acciones más o menos estándar de comunicación entre la copia de trabajo y el repositorio. Estas acciones son precisamente las que permite el cliente Subversion.

Es software libre bajo una licencia de tipo Apache/BSD (*Berkeley Software Distribution*) y se le conoce también como svn por ser ese el nombre de la herramienta de línea de comandos. Una característica importante de Subversion es que, a diferencia de CVS, los archivos versionados no tienen cada uno un número de revisión independiente. En cambio, todo el repositorio tiene un único número de versión que identifica un estado común de todos los archivos del repositorio en cierto punto del tiempo.

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

1.5.14. PostgreSQL servidor de base de datos.

Postgre SQL es un servidor de base de datos relacional libre, liberado bajo la licencia BSD (*Berkeley Software Distribution*). Es una alternativa a otros sistemas de bases de datos de código abierto (como MySQL, Firebird y MaxDB), así como sistemas propietarios como Oracle o DB2. (20)

Postgre SQL está ampliamente considerado como el sistema de bases de datos de código abierto más avanzado del mundo. Posee muchas características que tradicionalmente solo se podían ver en productos comerciales de alto calibre. (21)

Funciona en todos los sistemas operativos importantes, incluyendo Linux, UNIX y Windows. Soporta distintos tipos de datos: además del soporte para los tipos base, también soporta datos de tipo fecha, monetarios, elementos gráficos, datos sobre redes (MAC, IP...), cadenas de bits, etc. Incorpora una estructura de datos array y además permite la creación de tipos propios

Incluye características de la orientación a objetos, como puede ser la herencia, tipos de datos y operaciones geométricas, funciones, restricciones, disparadores, reglas e integridad transaccional.

1.5.15. Servidor Web: Apache Tomcat

El servidor de aplicaciones Tomcat de Apache y las tecnologías afines proporcionan a los programadores de Java un completo conjunto de herramientas para crear de forma rápida sofisticadas aplicaciones web. (22)

Apache Tomcat soporta las tecnologías Java HTTP Servlets y JSP. Es un servidor web fácil de obtener desde Internet debido en mayor medida a que es totalmente libre. Los requisitos de software para la utilización de este servidor recaen en la necesidad de disponer de la Máquina Virtual de Java para su adecuado funcionamiento. 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. Cuenta con el trabajo de miles de desarrolladores que contribuyen con su código y ponen a disposición de toda la comunidad las últimas actualizaciones. Apache Tomcat y las tecnologías afines proporcionan a los programadores de Java un completo conjunto de herramientas para crear de forma rápida sofisticadas aplicaciones web.

1.6. Conclusiones del capítulo

En este capítulo se realizó un análisis de los principales conceptos que deben conocerse para comprender la solución que se propone. Incluye además un estudio de algunos sistemas existentes para la gestión de contenidos de dispositivos móviles y la fundamentación de la elección de las herramientas y tecnologías a utilizar en la elaboración de la aplicación como es el caso de Oma Download como método de descarga empleado, Oma DRM para la protección de los contenidos y Alembik como herramienta de transcodificación; así como el uso de RUP como metodología de desarrollo de software, como lenguaje de modelado UML y herramienta case Visual Paradigm. Se utilizará además lenguaje de programación Java, entorno de desarrollo integrado Eclipse y como Framework Spring. Para el desarrollo de la base de datos se utilizó Hibernate con la implementación de JPA y como servidor de base de datos PostgreSQL.

Capítulo II: Características del Sistema

2.1. Introducción

A continuación se dará a conocer la propuesta del sistema a desarrollar y 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. Además se describirán los procesos existentes con el objetivo de comprenderlos y se tratarán los requisitos funcionales y no funcionales del sistema, así como el diagrama de casos de uso y las descripciones de los mismos teniendo en cuenta sus actores y casos de uso respectivamente.

2.2. Propuesta del Sistema

La solución que se propone se limita al desarrollo de la Plataforma COMCEL en su versión número 2. Dicha plataforma estará compuesta por diferentes módulos:

- Portal WAP
- Portal Web
- Módulo de Administración
- Módulo de Entrega de Contenido
- Módulo de Contenido
- Módulo de Mensajería
- Módulo de Transcoder
- Módulo de Facturación

Capítulo II: Características del Sistema

La plataforma se encargará de gestionar todas las operaciones asociadas en el proceso de entrega de contenidos. Realizando una serie de actividades que garanticen la descarga y adaptación del contenido adecuándolo a las características del dispositivo móvil que solicite el servicio. Además se permitirá darle a conocer los contenidos almacenados al Portal WEB.

Como inicio del proceso mencionado anteriormente el Portal WAP genera una orden, la cual es enviada al Módulo de Contenido, el mismo es el encargado de registrar dicha orden ya que administrará una base de datos relacional y un repositorio de contenidos donde se almacenarán los datos referentes a los contenidos. Luego el Módulo de Entrega de Contenido verifica si se le aplicó DRM al contenido solicitado y si el User Agent (Agente de Usuario) admite DRM. A continuación se pasa a adaptar el contenido a las características del dispositivo que hizo la solicitud por lo que se envía el contenido y el User Agent del teléfono al Módulo de Transcoding para pasar a realizar la transformación. Posteriormente se comprueba que el cliente admita descarga OMA para crear el descriptor (archivo XML que posee los datos referentes al contenido); de ser así se crea la URL donde se encuentra el contenido, se crea el descriptor y finalmente se le envía el mismo al cliente; de no admitir descarga OMA se pasa a realizar la descarga mediante un MMS. Una vez que el cliente obtenga el descriptor podrá acceder a descargar el contenido, por seguridad se verificará que el cliente que está accediendo sea el indicado; además se comprobará que el tiempo que tardó en hacerlo no supera el tiempo máximo establecido para ello. Luego se pasará al proceso de descargar el contenido obteniendo el mismo de la caché, donde fue enviado luego de haberle realizado el transcoding. Después de terminado el proceso se obtiene la notificación del mismo 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 Facturación y dar como atendida la solicitud en caso de éxito o esperar hasta que el contenido sea solicitado nuevamente.

Otra vía para descargar el contenido sería atender la solicitud realizada por el cliente al enviar un SMS con el código del contenido que desea descargar. La forma de atender dicho pedido es similar al descrito anteriormente comenzado por el Portal WAP; solo que con el SMS que envía el cliente no se puede obtener el User Agent por lo que se espera a que acceda a la URL para actualizar la orden realizada.

2.3. Modelo de Dominio

En el sistema que se desea realizar no se identifican de forma clara los actores, los trabajadores, ni los procesos del negocio por lo cual se decide modelar el negocio mediante un Modelo de Dominio.

El Modelo de Dominio constituye un subconjunto del modelo de objeto del negocio y en él se representan los conceptos y eventos fundamentales que se expresan como clases con la cardinalidad que existe entre ellas.

Conceptos

User: Entidad que descarga los contenidos.

Provider: Representa la entidad que se encarga de proveer los contenidos.

Country: Representa el país del proveedor de contenidos.

Content: Representa los contenidos que se gestionan.

Music: Representa los contenidos de tipo música.

Aplication: Representa los contenidos de tipo aplicaciones.

Order: Representa una petición para adquirir un contenido.

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

M_Administración: Representa el subsistema encargado de la comunicación entre el administrador y la plataforma.

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

M_ContentDelivery: Encargado de gestionar el proceso de entregar el contenido al cliente.

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

M_Transcoder: Representa el subsistema de modificación de formato del contenido.

M_Facturación: Módulo que efectúa el cobro por la descarga de los contenidos.

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

2.3.1. Diagrama de Clases del dominio

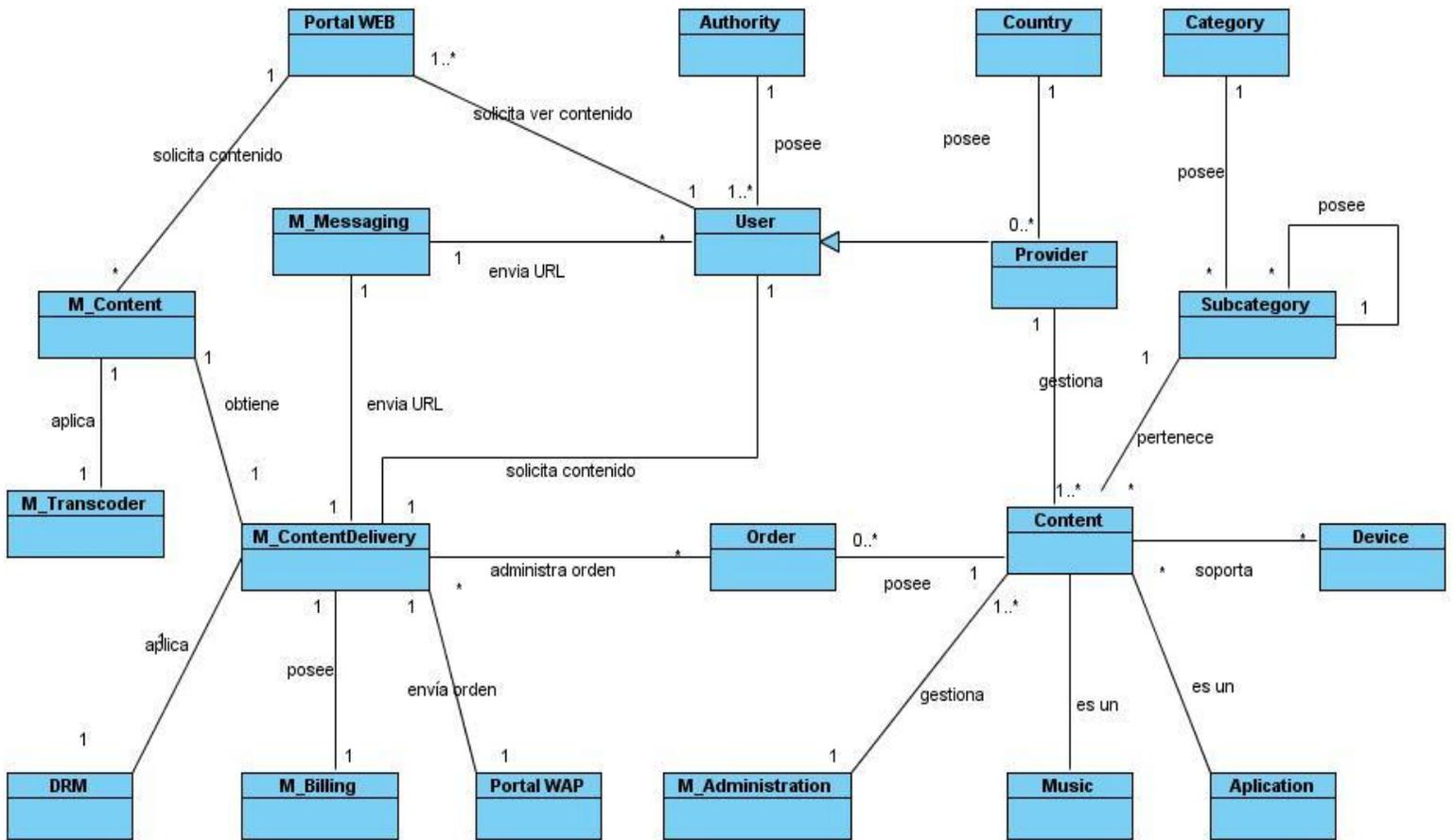


Figura 1: Modelo de dominio

2.4. Especificaciones de requisitos del sistema

2.4.1. Requisitos funcionales

- RF 1. Autenticar Usuario.
- RF 2. Buscar todos los países.
- RF 3. Buscar todas las categorías.
- RF 4. Buscar todos los usuarios.
- RF 5. Buscar usuarios por países.
- RF 6. Buscar usuario por ID.
- RF 7. Buscar usuarios por rol.
- RF 8. Adicionar Usuario.
- RF 9. Modificar Usuario.
- RF 10. Eliminar Usuario.
- RF 11. Buscar todas las subcategorías.
- RF 12. Adicionar Subcategoría.
- RF 13. Modificar Subcategoría.
- RF 14. Eliminar Subcategoría.
- RF 15. Buscar contenido por ID.
- RF 16. Buscar todos los contenidos.

Capítulo II: Características del Sistema

RF 17. Buscar contenidos disponibles por categoría.

RF 18. Buscar todos los contenidos por categoría.

RF 19. Buscar contenidos disponibles por proveedor.

RF 20. Buscar contenidos por proveedor.

RF 21. Buscar los 10 contenidos más descargados.

RF 22. Buscar los 10 contenidos más descargados por ID de categoría.

RF 23. Obtener los últimos contenidos.

RF 24. Obtener los últimos contenidos por ID de categoría.

RF 25. Adicionar contenido.

RF 26. Modificar contenido.

RF 27. Eliminar contenido.

RF 28. Obtener todos los dispositivos móviles.

RF 29. Obtener los dispositivos móviles por nombre de la marca.

RF 30. Obtener todas las marcas de los dispositivos móviles.

RF 31. Adicionar Orden.

RF 32. Modificar Orden.

RF 33. Aplicar DRM (método Forward Lock) para proteger el contenido.

RF 34. Transformar el contenido en correspondencia con el móvil que está realizando la petición.

RF 35. Crear un mecanismo de caché de forma tal que cuando se realice un proceso de transcoding a un dispositivo móvil y después se realice otra petición que contenga el mismo tipo de contenido y el mismo User Agent no sea necesario volver a realizarle el proceso.

RF 36. Enviar descriptor al cliente.

RF 36.1. Crear URL del descriptor.

RF 36.2. Crear descriptor.

RF 36.3. Enviar descriptor al cliente.

RF 37. Enviar URL del contenido al cliente.

RF 38. Expirar solicitud.

RF 39. Entregar contenido al cliente.

RF 40. Atender notificación de descarga.

RF 41. Realizar el cobro de la descarga.

2.4.2. Requisitos 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.

(23)

Soporte

RNF 1. Se elaborará un manual de usuario que contendrá aspectos relacionados al uso del sistema y de todos los objetos generados por este. En el manual se abordarán sobre todo las definiciones principales del sistema, el trabajo con los archivos que se generan, como LOGs y los posibles errores que pudieran ocurrir, especificando las posibles soluciones del mismo.

Capítulo II: Características del Sistema

RNF 2. Se generará el manual de usuario creado por java(JavaDoc) de todas las estructuras implementadas en el sistema

RNF 3. La UCI debe brindar soporte tanto a administradores como a clientes que necesiten una capacitación para operar el sistema, en dependencia de las demandas del comprador, puede ser mantenimiento, entrenamiento de personal, integración con otro software u operar directamente el sistema.

Portabilidad

RNF 4. El sistema será multiplataforma.

RNF 5. El sistema debe ser desarrollado en forma modular.

Restricciones en el diseño y la implementación

RNF 6. El lenguaje de programación a ser usado para la implementación será el lenguaje de programación orientado a objetos: JAVA.

RNF 7. La herramienta Case a utilizar será: Visual Paradigm con soporte para UML 6.0.

RNF 8. El sistema debe cumplir con estándares internacionales de transcoding de contenidos para móviles como el STI 1.0 de OMA.

Usabilidad

RNF 9. Debe ser escalable para poder agregar nuevos servicios sin afectar los que ya se encuentran en funcionamiento.

RNF 10. Debe permitir la interacción con diferentes proveedores de contenidos independientemente de las plataformas que estos posean.

Capítulo II: Características del Sistema

RNF 11. Debe ser capaz de adaptarse a cualquier sistema de gestión de contenidos que implementen otros operadores de telefonía móvil, ya que la plataforma puede ser comercializable con otros operadores.

Seguridad

RNF 12. El sistema deberá registrar una cantidad finita de eventos normales o anormales del sistema en Logs, para poder desarrollar una auditoría.

RNF 13. El sistema debe ser capaz de permitir que acceda solamente al contenido el cliente que lo solicitó.

RNF 14. Se garantizará el tratamiento adecuado de las excepciones y validaciones de las entradas del usuario.

RNF 15. Se le proporcionará una seguridad basada en la autenticación y en el control de acceso.

RNF 16. Se definirán roles a los usuarios registrados en la aplicación.

Software

RNF 17. Se requiere la presencia del servidor Web Apache Tomcat.

RNF 18. El sistema necesita la máquina virtual de java para su funcionamiento.

RNF 19. Para transformar videos y audio es necesario tener instalado el software FFMpeg.

RNF 20. Para transformar imágenes se recomienda la instalación del software ImageMagick.

Hardware

RNF 21. El sistema se desplegará en un servidor con 4gb de memoria RAM como requerimiento mínimo, procesador Core 2 Duo a 2.6 ghz o superior y debe contar además con un almacenamiento en disco de 500gb.

Apariencia o Interfaz interna

RNF 22. La implementación de las funcionalidades debe seguir los lineamientos de código establecidos por la dirección del proyecto, para garantizar la comprensión del código por desarrolladores de futuras versiones.

RNF 23. Los servicios prestados por la plataforma serán publicados en una interfaz para ser accedidos por el resto de los componentes del sistema, garantizando la genericidad de la plataforma y que sea capaz de adaptarse a cualquier sistema que implementen los operadores de telefonía móvil. Los restantes componentes de la plataforma podrán acceder a los servicios del Módulo Contenidos mediante objetos.

2.5. Modelo de Casos de Usos del sistema

2.5.1. Actores del sistema

Los actores del sistema representan a las personas o entidades que realizarán los casos de uso. En el sistema propuesto se identificó los siguientes actores:

Actores	Justificación
Usuario	Representa una generalización de los actores del sistema para las tareas más generales.
Cliente	Representa al cliente que solicita los servicios brindados.

Portal_WEB	Representa al Portal WEB encargado de mostrarle los contenidos brindados a los clientes.
Portal_WAP	Representa al Portal Wap, encargado de la interacción entre los clientes y la Plataforma para descargar el contenido.
M_Administracion	Representa el Módulo de Administración, encargado de la interacción entre el personal autorizado para administrar la información y la Plataforma.
Factor Tiempo	Gestiona que se eliminen las solicitudes que el tiempo de espera es mayor al tiempo establecido para la descarga.

Tabla 1: Definición de los actores del sistema.

2.5.2. Casos de Uso del sistema

Los casos de uso son funcionalidades que el sistema debe cumplir para aportar un resultado de valor para sus actores. En ellos se representan comportamientos del sistema desde el punto de vista del usuario al interactuar con la aplicación.

CU-1	Autenticar Usuario
Actor	Usuario

Capítulo II: Características del Sistema

Descripción	Permite autenticar un usuario en el sistema.
Referencia	RF 1

Tabla 2: Caso de Uso Autenticar Usuario

CU-2	Buscar País
Actor	M_Administracion
Descripción	Permite buscar todos los países almacenados en la Base de Datos.
Referencia	RF 2

Tabla 3: Caso de Uso Buscar País

CU-3	Buscar Categoría
Actor	M_Administracion
Descripción	Permite buscar todas las categorías almacenadas en la Base de Datos.
Referencia	RF 3

Tabla 4: Caso de Uso Buscar Categoría

CU-4	Buscar Usuario
------	----------------

Capítulo II: Características del Sistema

Actor	M_Administracion
Descripción	Permite buscar uno o varios usuarios según el parámetro solicitado.
Referencia	RF 4, RF 5, RF 6, RF 7

Tabla 5: Caso de Uso Buscar Usuario

CU-5	Gestionar Usuario
Actor	M_Administracion
Descripción	Permite gestionar los usuarios de la plataforma.
Referencia	RF 8, RF 9, RF 10

Tabla 6: Caso de Uso Gestionar Usuario

CU-6	Buscar Subcategoría
Actor	M_Administracion
Descripción	Permite buscar todas las subcategorías almacenadas en la Base de Datos.
Referencia	RF 11

Tabla 7: Caso de Uso Buscar Subcategoría

CU-7	Gestionar Subcategoría
------	------------------------

Capítulo II: Características del Sistema

Actor	M_Administracion
Descripción	Permite gestionar las subcategorías de los contenidos
Referencia	RF 12, RF 13, RF 14

Tabla 8: Caso de Uso Gestionar Subcategoría

CU-8	Buscar Contenido
Actor	Usuario
Descripción	Permite buscar uno o varios contenidos según el parámetro solicitado.
Referencia	RF 15, RF 16, RF 17, RF 18, RF 19, RF 20, RF 21, RF 22, RF 23, RF 24

Tabla 9: Caso de Uso Buscar Contenido

CU-9	Gestionar Contenido
Actor	M_Administracion
Descripción	Permite gestionar los contenidos.
Referencia	RF 25, RF 26, RF 27

Tabla 10: Caso de Uso Gestionar Contenido

CU-10	Buscar dispositivo
-------	--------------------

Capítulo II: Características del Sistema

Actor	Usuario
Descripción	Permite buscar los dispositivos según el parámetro solicitado.
Referencia	RF 28, RF 29, RF 30

Tabla 11: Caso de Uso Buscar dispositivo

CU-11	Atender Solicitud Vía WAP
Actor	Portal_WAP
Descripción	Gestiona que se adicione una orden a la base de datos para luego continuar con el proceso de entrega de contenido.
Referencia	RF 31

Tabla 12: Caso de Uso Atender Solicitud Vía WAP

CU-12	Atender Solicitud Vía SMS
Actor	Cliente
Descripción	Atender la solicitud de un contenido vía SMS hasta actualizar completamente la orden en la base de datos para luego continuar con el proceso.

Capítulo II: Características del Sistema

Referencia	RF 31, RF 32
------------	--------------

Tabla 13: Caso de Uso Atender Solicitud Vía SMS

CU-13	Atender Solicitud
Actor	CU Atender Solicitud Vía WAP, CU Atender Solicitud Vía WEB
Descripción	Dar a conocer la dirección de donde el cliente debe descargar el contenido.
Referencia	RF 36, RF 37

Tabla 14: Caso de Uso Atender Solicitud

CU-14	Entregar Contenido
Actor	Cliente
Descripción	Permite entregar el contenido al cliente.
Referencia	RF 39, RF 40

Tabla 15: Caso de Uso Entregar Contenido

CU-15	Realizar Transcoding
Actor	CU Entregar Contenido
Descripción	Se solicita llevar a cabo el proceso de transcoding de un contenido, dado un

Capítulo II: Características del Sistema

	identificador de contenido y el User Agent del móvil.
Referencia	RF 34, RF 35

Tabla 16: Caso de Uso Realizar Transcoding

CU-16	Aplicar DRM
Actor	CU Atender Solicitud
Descripción	Se solicita aplicar el algoritmo Forward Lock, el cual recibe el contenido a proteger para luego armar el mensaje DRM y realizar la solicitud de su entrega.
Referencia	RF 33

Tabla 17: Caso de Uso Aplicar DRM

CU-17	Realizar Cobro
Actor	CU Entregar Contenido
Descripción	Permite realizar el cobro por la descarga de contenidos.
Referencia	RF 41

Tabla 18: Caso de Uso Realizar Cobro

CU-18	Expirar solicitudes
-------	---------------------

Actor	Factor Tiempo
Descripción	Gestiona que se eliminen las solicitudes que hayan superado el tiempo límite para descargar.
Referencia	RF 38

Tabla 19: Caso de Uso Expirar solicitudes

2.5.3. Diagrama de Paquetes de Casos de Usos del sistema



Figura 2: Diagrama de paquetes de caso de uso del sistema

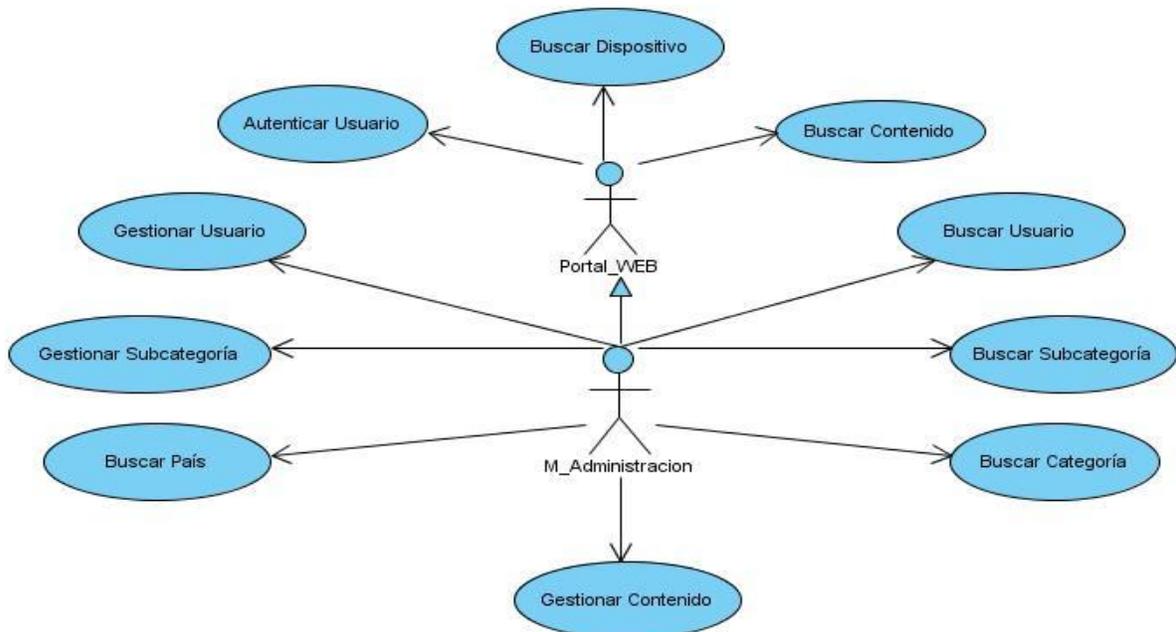


Figura 3: Diagrama de caso de uso del sistema paquete gestionar contenido

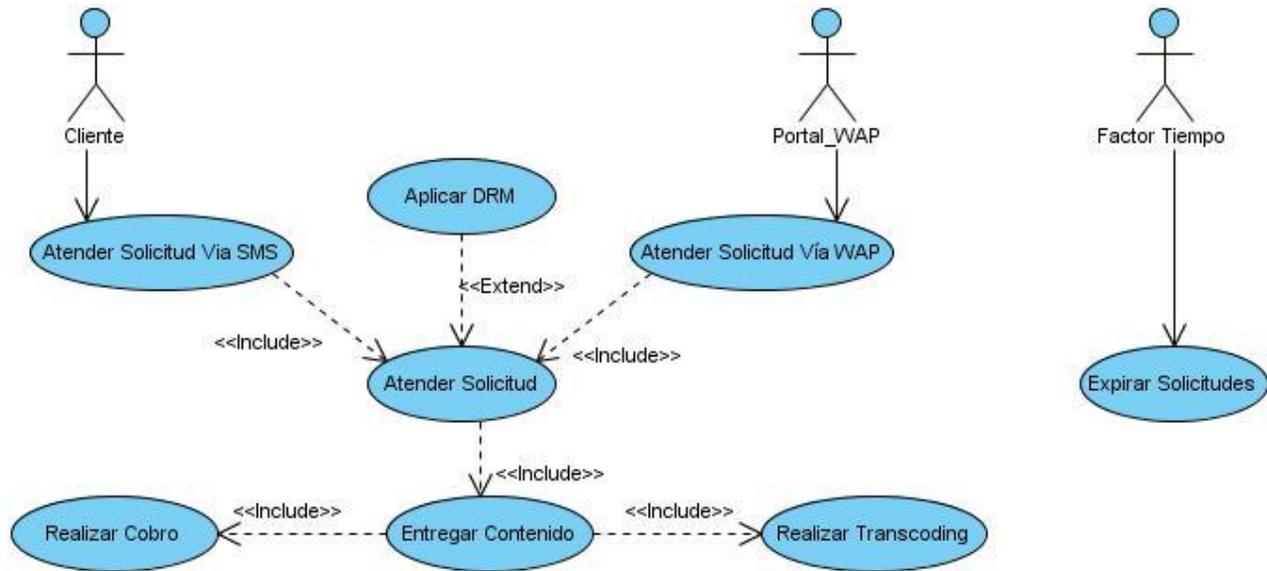


Figura 4: Diagrama de caso de uso del sistema paquete descargar contenido

2.6. Conclusiones del capítulo

Este capítulo permitió crear el modelo del sistema que será implementado, utilizando los casos de usos y su descripción para ese propósito. También posibilitó llegar a un acuerdo con los clientes sobre las condiciones y funcionalidades que debe cumplir el sistema. Como resultado del trabajo realizado se decide priorizar el desarrollo de los casos de uso Gestionar Contenido, Buscar Contenido, Atender Solicitud Vía Wap, Atender Solicitud Vía SMS, Atender Solicitud y Entregar Contenido por contar con funcionalidades críticas para el ciclo de vida del software.

Capítulo III: Análisis y Diseño del Sistema

3.1. Introducción

El objetivo fundamental a tratar en el presente capítulo es la realización de un diseño consecuente que dé respuesta a los requerimientos planteados en el capítulo anterior. El análisis fundamentalmente consiste en obtener una visión que se preocupa de ver que hace el sistema de software a desarrollar mientras que el diseño es un refinamiento que toma en cuenta los requerimientos no funcionales, por lo cual se centra en como el sistema cumple sus objetivos.

Los objetivos específicos del análisis y diseño son:

1. Transformar los requerimientos al diseño del futuro sistema.
2. Desarrollar una arquitectura para el sistema.
3. Adaptar el diseño para que sea consistente con el entorno de implementación.

3.2. Modelo de Diseño

El modelo de diseño es una abstracción de la implementación del sistema. Se utiliza para concebir y para documentar el diseño del sistema de software. Es un producto de trabajo integral y compuesto que abarca todas las clases de diseño, subsistemas, paquetes, colaboraciones y las relaciones entre ellos. Diagrama de Clases del Diseño. (24)

3.2.1. Diagramas de Interacción

El diagrama de interacción, representa la forma en cómo un Cliente (Actor) u Objetos (Clases) se comunican entre sí en petición a un evento. Esto implica recorrer toda la secuencia de llamadas, de donde se obtienen las responsabilidades claramente. (25)

3.2.2. 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.

El Modelo de Diseño de la Plataforma ha sido dividido en 4 subsistemas de diseño mientras que el subsistema Content dado su complejidad fue desglosado en 3 subsistemas más. Estos son una forma de organizar el Modelo de Diseño en fragmentos más manejables y fáciles de entender. Cada subsistema posee varias clases del diseño e interfaces que se encuentran fuertemente asociados según sus funciones. Se realizó un Diagrama de Clases del Diseño para cada subsistema identificado y un Diagrama de Paquetes que muestra la relación entre los subsistemas, así como su organización teniendo en cuenta el patrón arquitectónico basado en componentes.

Diagrama de Paquetes

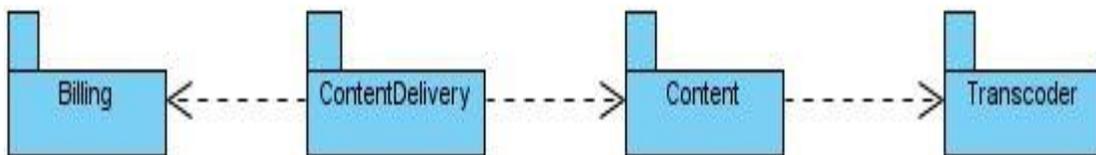


Figura 5: Diagrama de paquetes

Subsistema Billing

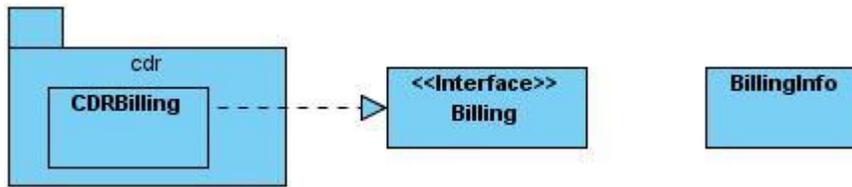


Figura 7: Diagrama de clases del diseño. Subsistema Billing

Subsistema Transcoder

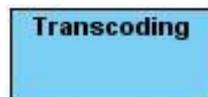


Figura 8: Diagrama de clases del diseño. Subsistema Transcoder

Subsistema Content (Diagrama de paquetes)

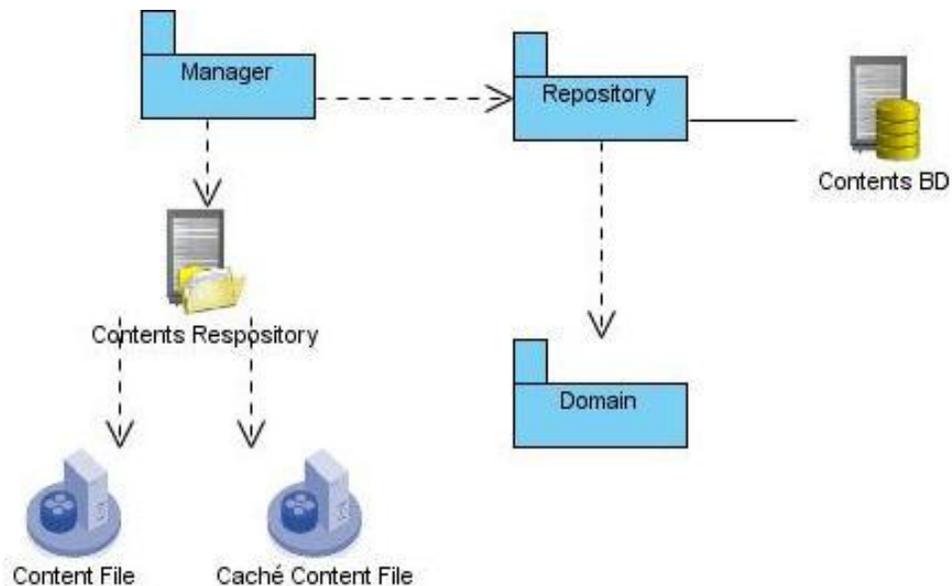


Figura 9: Diagrama de paquetes. Subsistema Content

Subsistema Content (Subsistema Manager)

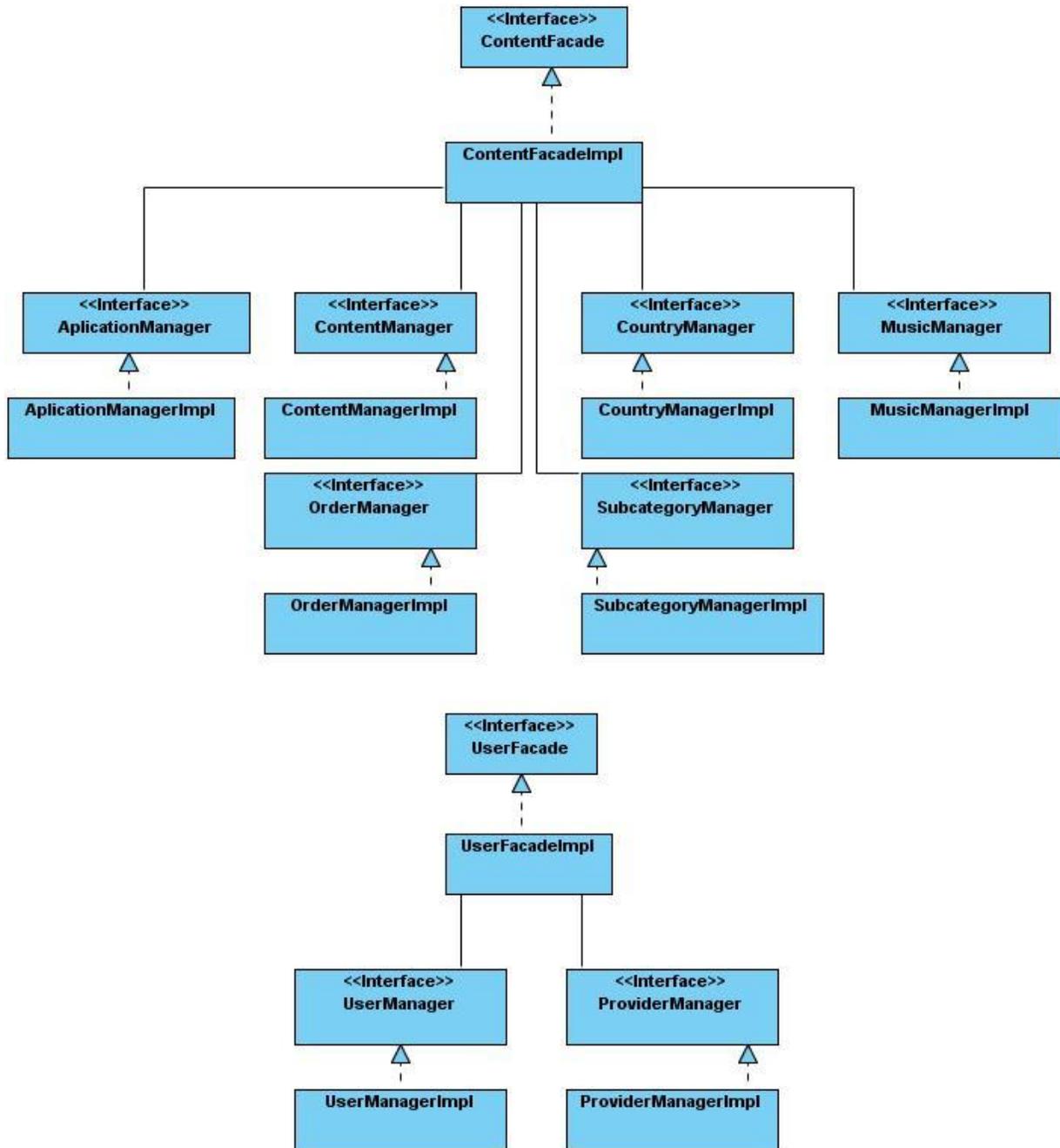


Figura 10: Diagrama de clases del diseño. Subsistema Manager

Subsistema Content (Subsistema Repository)

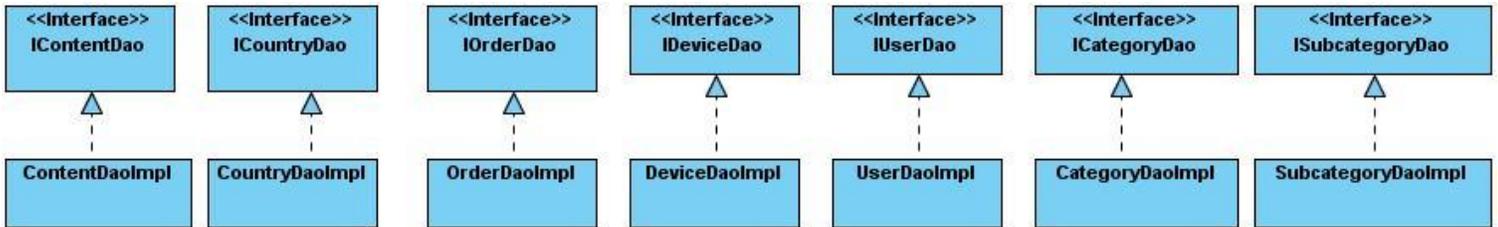


Figura 11: Diagrama de clases del diseño. Subsistema Repository

Subsistema Content (Subsistema Domain)

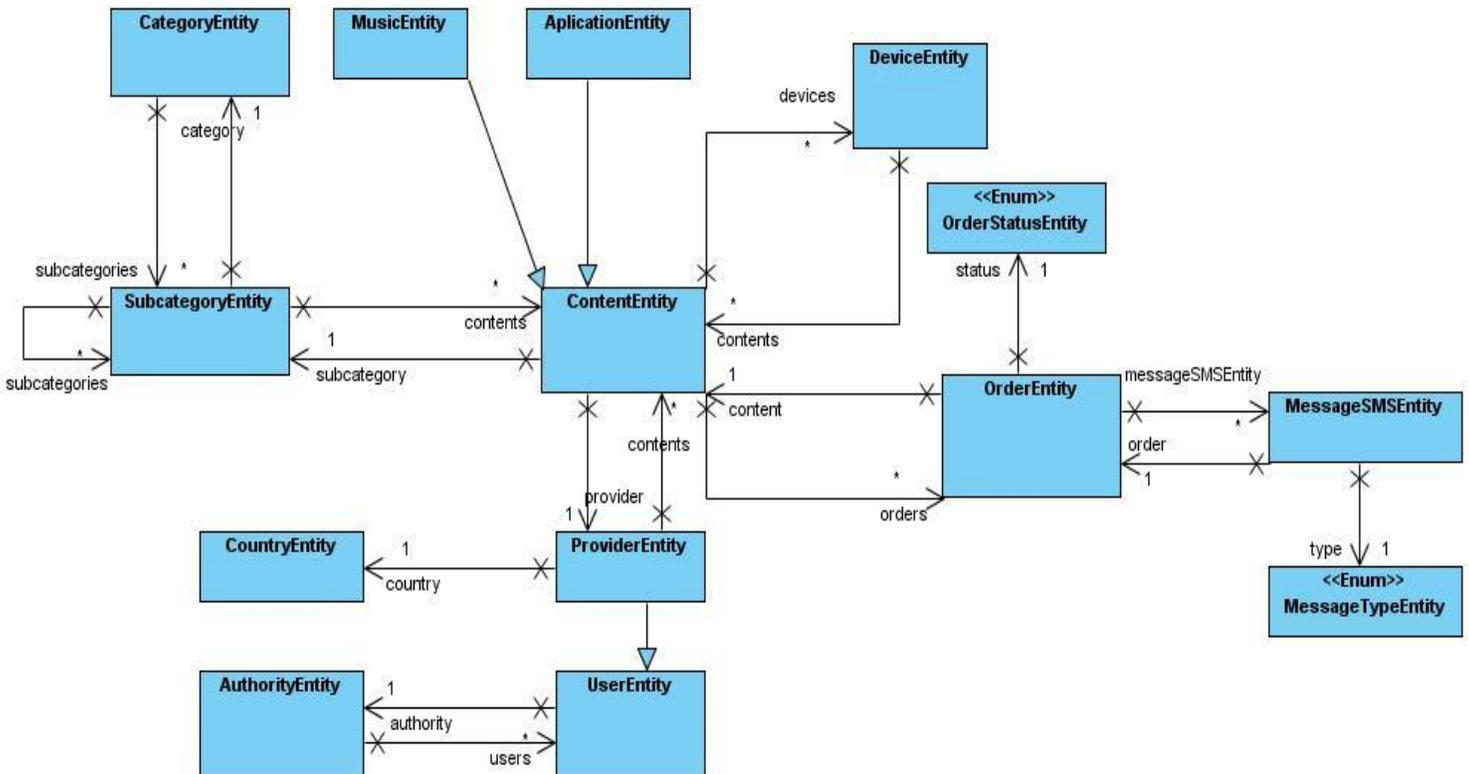


Figura 12: Diagrama de clases del diseño. Subsistema Domain

3.3. Descripción de la Arquitectura

3.3.1. Patrones utilizados

En ingeniería de software, un patrón es una solución ya probada y aplicable a un problema que se presenta una y otra vez en el desarrollo de distintas aplicaciones y en distintos contextos. Es importante destacar que un patrón no es en general una solución en forma de código directamente, sino una descripción de cómo resolver el problema y ante qué circunstancias es aplicable.

Patrones arquitectónicos: Aquellos que expresan un esquema organizativo estructural fundamental para sistemas software.

Patrones de diseño: Aquellos que expresan esquemas para definir estructuras de diseño (o sus relaciones) con las que construir sistemas software.

3.3.2. Patrón arquitectónico

Una arquitectura basada en componentes describe una aproximación de ingeniería de software al diseño y desarrollo de un sistema. Esta arquitectura se enfoca en la descomposición del diseño en componentes funcionales o lógicos que expongan interfaces de comunicación bien definidas. Esto provee un nivel de abstracción mayor que los principios de programación orientada a objetos y no se enfoca en asuntos específicos de los objetos como los protocolos de comunicación y la forma como se comparte el estado.

El estilo de arquitectura basado en componentes tiene las siguientes características:

- Es un estilo de diseño para aplicaciones compuestas de componentes individuales.
- Pone énfasis en la descomposición del sistema en componentes lógicos o funcionales que tienen interfaces bien definidas.
- Define una aproximación de diseño que usa componentes discretos, los que se comunican a través de interfaces que contienen métodos, eventos y propiedades.

Principios Fundamentales

Un componente es un objeto de software específicamente diseñado para cumplir con cierto propósito. Los principios fundamentales cuando se diseña un componente es que estos deben ser:

- **Reusable:** Los componentes son usualmente diseñados para ser utilizados en escenarios diferentes por diferentes aplicaciones, sin embargo, algunos componentes pueden ser diseñados para tareas específicas.
- **Sin contexto específico:** Los componentes son diseñados para operar en diferentes ambientes y contextos. Información específica como el estado de los datos deben ser pasadas al componente en vez de incluirlos o permitir al componente acceder a ellos.
- **Extensible:** Un componente puede ser extendido desde un componente existente para crear un nuevo comportamiento.
- **Encapsulado:** Los componentes exponen interfaces que permiten al programa usar su funcionalidad. Sin revelar detalles internos, detalles del proceso o estado.
- **Independiente:** Los componentes están diseñados para tener una dependencia mínima de otros componentes. Por lo tanto, los componentes pueden ser instalados en el ambiente adecuado sin afectar otros componentes o sistemas. (26)

En esencia, un componente es una pieza de código preelaborado que encapsula alguna funcionalidad expuesta a través de interfaces estándar. Los componentes son los "ingredientes de las aplicaciones", que se juntan y combinan para llevar a cabo una tarea.

El paradigma de ensamblar componentes y escribir código para hacer que estos componentes funcionen se conoce como Desarrollo de Software Basado en Componentes. El uso de este paradigma posee algunas ventajas:

- **Reutilización del software:** Nos lleva a alcanzar un mayor nivel de reutilización de software.
- **Simplifica las pruebas:** Permite que las pruebas sean ejecutadas probando cada uno de los componentes antes de probar el conjunto completo de componentes ensamblados.

- **Simplifica el mantenimiento del sistema:** Cuando existe un débil acoplamiento entre componentes, el desarrollador es libre de actualizar y/o agregar componentes según sea necesario, sin afectar otras partes del sistema.
- **Mayor calidad:** Dado que un componente puede ser construido y luego mejorado continuamente por un experto u organización, la calidad de una aplicación basada en componentes mejorará con el paso del tiempo. (27)

El desarrollo de software basado en componentes se ha convertido actualmente en uno de los mecanismos más efectivos para la construcción de grandes sistemas y aplicaciones de software. (28)

3.3.3. Patrones de diseño. Patrones GRASP

Son soluciones simples y elegantes a problemas específicos y comunes del diseño orientado a objetos. Son soluciones basadas en la experiencia y que se ha demostrado que funcionan. (29)

Patrones GRASP (General Responsibility Assignment Software Patterns): Los patrones GRASP son patrones de diseño que describen los principios generales para asignar responsabilidades a objetos, expresados en forma de patrón.

- **Experto:** Asignar una responsabilidad al experto en información: la clase que cuenta con la información necesaria para cumplir la responsabilidad.
- **Creador:** El patrón Creador guía la asignación de responsabilidades relacionadas con la creación de objetos, tarea muy frecuente en los sistemas orientados a objetos.
- **Bajo Acoplamiento:** Acoplamiento bajo significa que una clase no depende de muchas clases.
- **Alta Cohesión:** La cohesión es una medida de cuán relacionadas y enfocadas están las responsabilidades de una clase.

Una alta cohesión caracteriza a las clases con responsabilidades estrechamente relacionadas que no realicen un trabajo enorme.

3.3.4. Patrones de diseño. Patrones GOF

- **Facade:** La idea principal es la de ocultar todo lo posible la complejidad de un sistema, el conjunto de clases o componentes que lo forman, de forma que solo se ofrezca un (o unos pocos) punto de entrada al sistema tapado por la fachada. (30)
- **Singleton:** El patrón Singleton garantiza que una clase solo tenga una instancia y proporciona un punto de acceso global a esta instancia. (31)
- **Inyección de Dependencia:** Es un patrón de diseño orientado a objetos, en el que se inyectan objetos a una clase en lugar de ser la propia clase quien cree el objeto.

3.4. Modelo de Datos

La necesidad de definir una estructura que permita almacenar datos, reconocer el contenido, y recuperar la información hacen del modelo de datos una actividad esencial en el desarrollo del software. La estructura tiene que estar en correspondencia con las necesidades de las aplicaciones que la usarán.

El modelado de los datos responde a interrogantes importantes para el procesamiento de datos sobre las entidades que procesará el sistema, los atributos y las relaciones entre las mismas. Para responder estas interrogantes se realizan un conjunto de actividades que tienen como resultado primario un Modelo Lógico de Datos, a través de las clases persistentes y un Modelo Físico de Datos, mediante el Diagrama Entidad-Relación.

3.4.1. Modelo lógico de datos

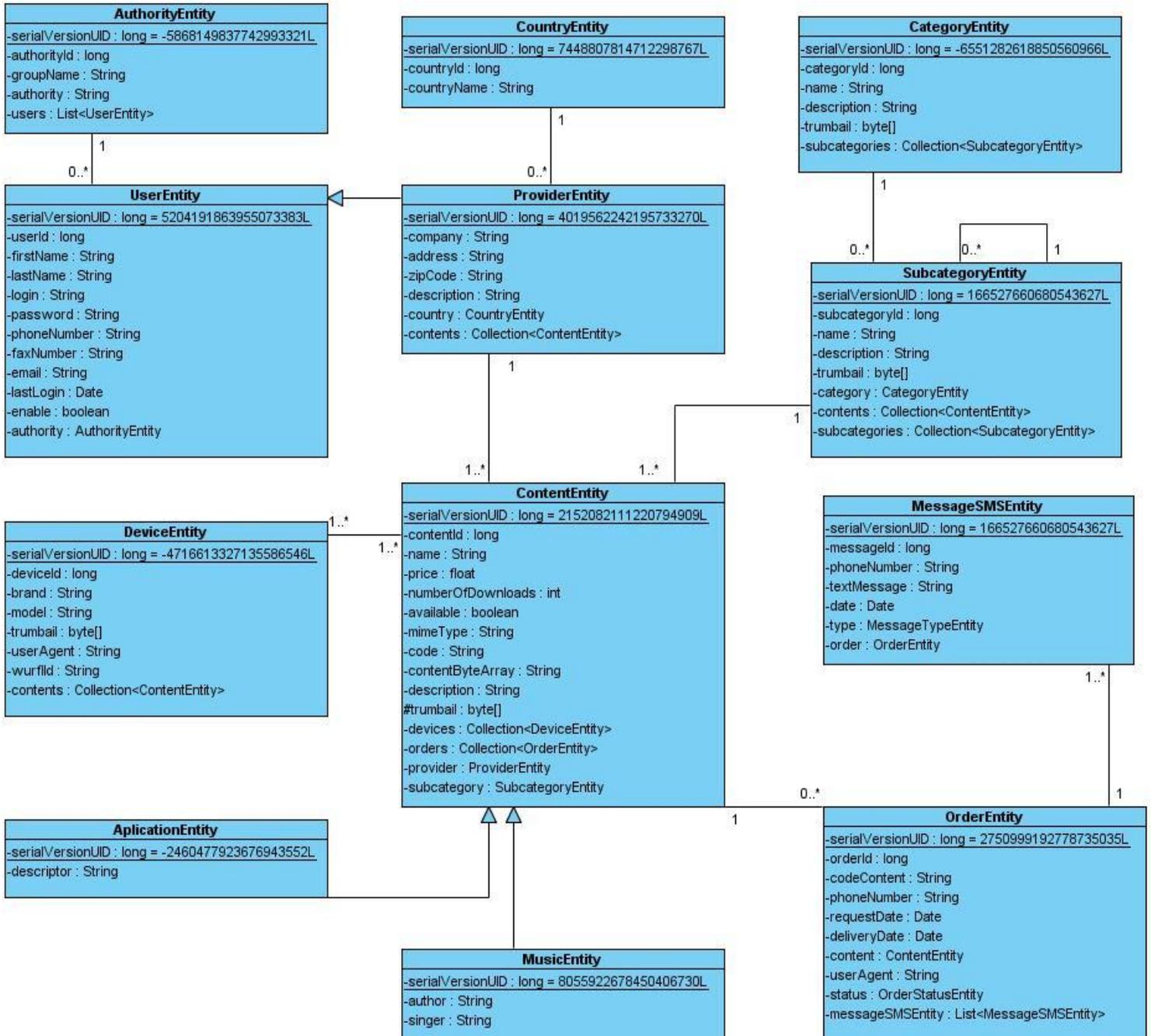


Figura 13: Diagrama de clases persistentes

3.4.2. Modelo físico de datos

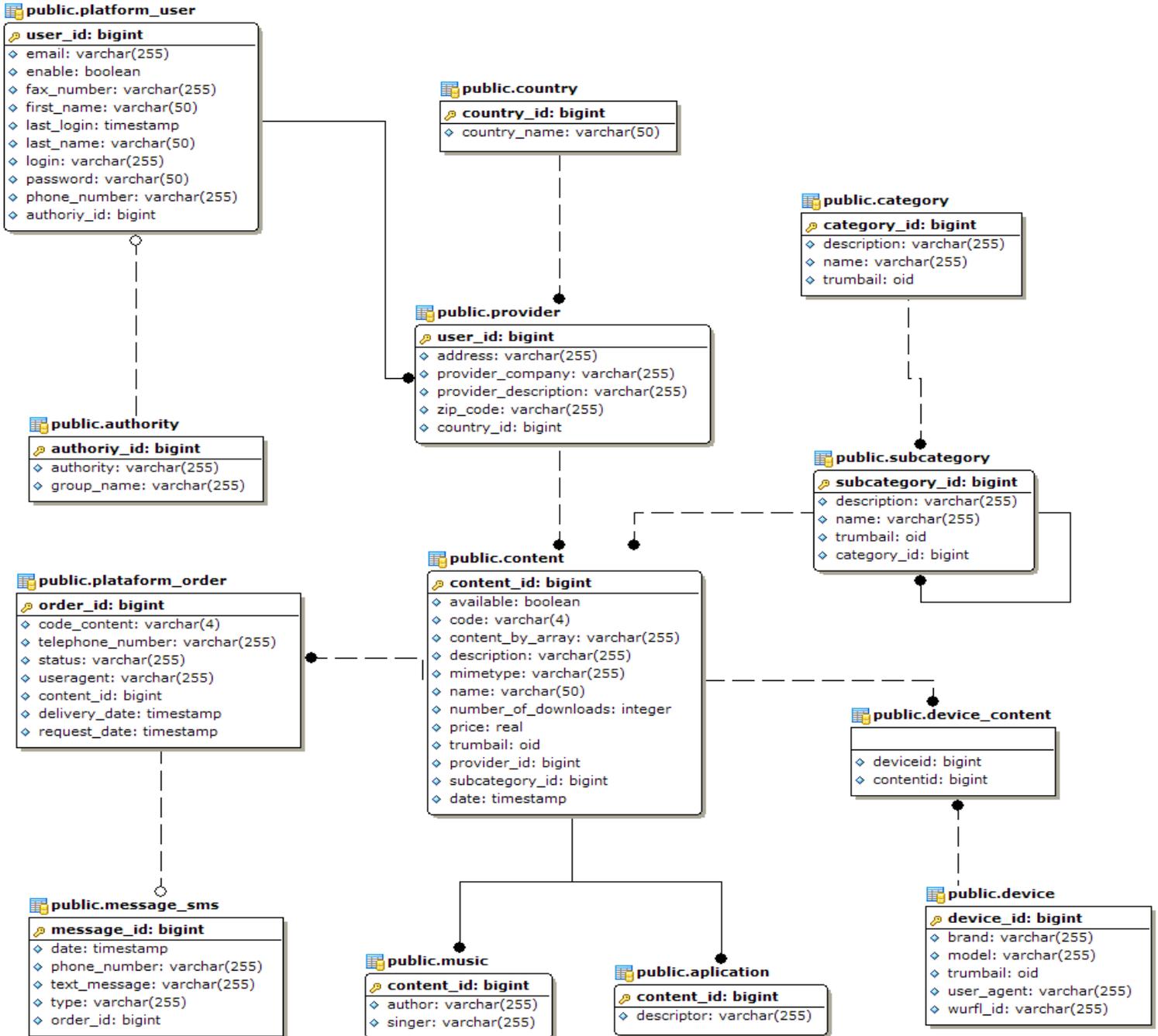


Figura 14: Modelo físico de datos

3.5. 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. El tratamiento de errores estará enfocado principalmente en detectar posibles irregularidades en el sistema que puedan provocarle un mal funcionamiento y con ello dañar el resultado esperado de la operación, para esto contaremos con clases que extiendan de la clase Excepción y que permitan identificar el problema en todos los niveles de la aplicación dígame en el acceso a los datos como en su procesamiento, además estos errores identificados se deberán guardar en log para posibilitar una revisión minuciosa del comportamiento del sistema en determinados momentos y ante de determinadas circunstancias. Cuando ocurra un error en los servicios que presta la plataforma y este sea tratado se le deberá informar al usuario en un mensaje de texto lo suficientemente claro para él.

3.6. Seguridad

Poder brindarle confianza y seguridad al cliente siempre es una de las principales características con que deben contar los productos informáticos, es por ello que la plataforma no está ajena a ello y utiliza para la parte de seguridad Spring Security que posibilita brindar una autenticación y autorización al usuario cuando va a acceder a un determinado recurso, además permite poder verificar que un único usuario tenga una sola sección abierta en la aplicación.

3.7. Conclusiones del capítulo

El Análisis y Diseño contribuye a una arquitectura estable y sólida, un diseño robusto de la base de datos y a crear un plano del modelo de implementación. La peculiaridad del diseño es modelar el sistema, encontrar su forma (incluida la arquitectura) para que soporte todos sus requisitos.

Capítulo IV: Implementación

4.1. Introducción

En el presente capítulo se representan en términos de componentes los elementos identificados en el Modelo de Diseño. A continuación se modela un diagrama de componentes, con los componentes principales que constituyen la Plataforma, dando paso luego a presentar la situación física de los componentes lógicos desarrollados en el Diagrama de Despliegue.

4.2. Modelo de Implementación

El modelo de implementación representa la composición física de la implementación en términos de subsistemas de implementación, y elementos de implementación (directorios y archivos, incluyendo código fuente, datos y archivos ejecutables). (32)

Los componentes son la parte física y reemplazable de un sistema que está compuesto por un conjunto de interfaces y proporciona la realización de dicho conjunto. Se usan para modelar los elementos físicos que pueden hallarse en un nodo por lo que empaquetan elementos como clases, colaboraciones e interfaces. Son independientes entre ellos y tienen su propia estructura e implementación. Tienen relaciones de traza con los elementos del modelo que implementan.

4.2.1. Diagrama de despliegue

Un diagrama de Despliegue muestra cómo y dónde se desplegará el sistema. Las máquinas físicas y los procesadores se representan como nodos, y la construcción interna puede ser representada por nodos o artefactos embebidos. Como los artefactos se ubican en los nodos para modelar el despliegue del sistema, la ubicación es guiada por el uso de las especificaciones de despliegue. (33)



Figura 15: Diagrama de despliegue

Descripción de los nodos:

Dispositivo Móvil: Representa a los usuarios finales de la Plataforma, los cuales se conectan al Servidor de Aplicaciones donde se encuentra la Plataforma de Gestión de Contenidos.

Servidor de Aplicaciones:

Servidor de Aplicaciones (Plataforma): Constituye el nodo fundamental, aquí se encuentra la Plataforma de Gestión de Contenidos para Dispositivos Móviles. Su comunicación con los dispositivos móviles es a través del protocolo HTTP.

Protocolo de comunicación:

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

4.2.2. Diagrama de componentes

Un diagrama de Componentes ilustra los fragmentos de software, controladores embebidos, etc. que conformarán un sistema. Un diagrama de componentes tiene un nivel de abstracción más elevado que un diagrama de clase; usualmente un componente se implementa por una o más clases (u objetos) en tiempo de ejecución. Estos son bloques de construcción, como así eventualmente un componente puede comprender una gran porción de un sistema. (34)

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

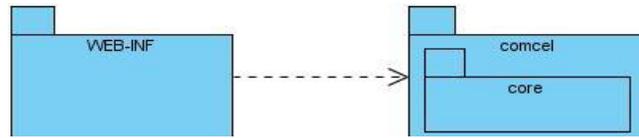


Figura 16: Diagrama de componentes general de la Plataforma de Gestión de Contenidos

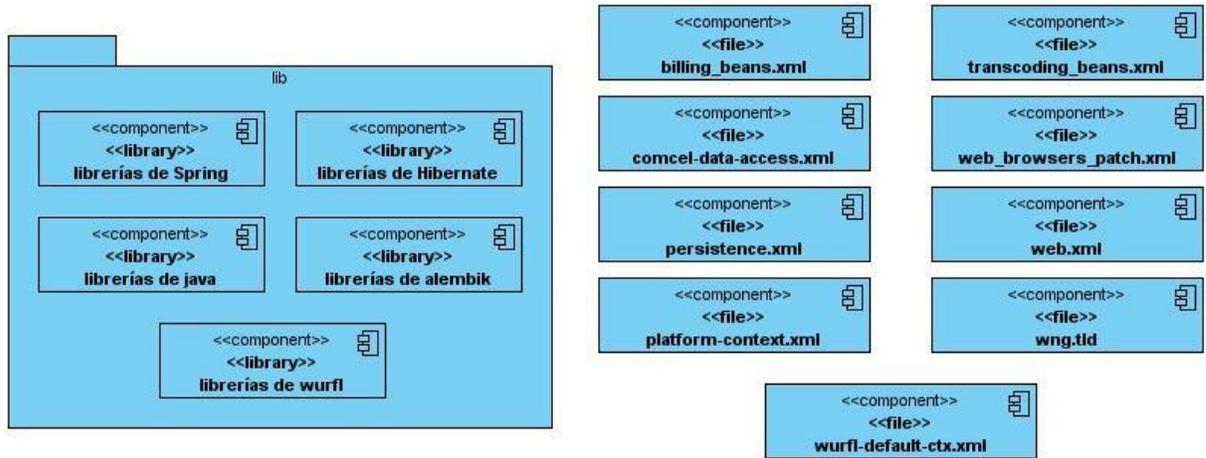


Figura 17: Diagrama de componentes del paquete WEB-INF

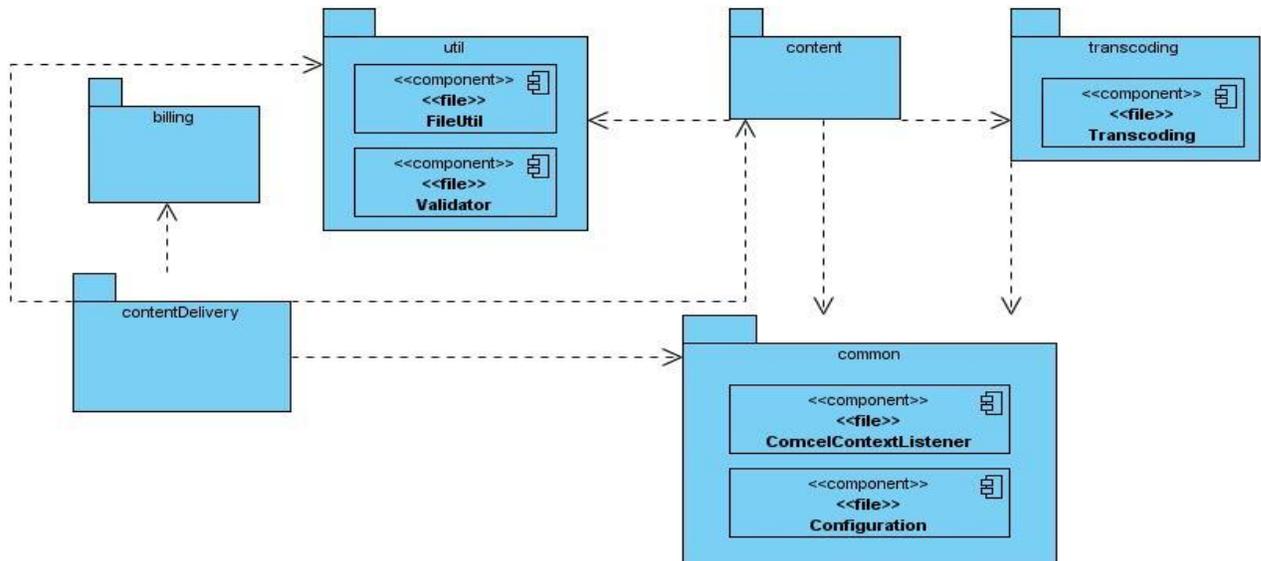


Figura 18: Diagrama de componentes del paquete core

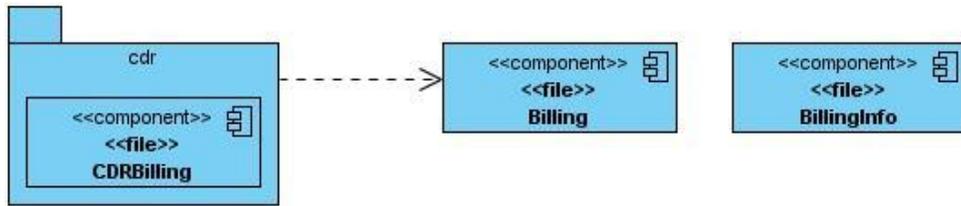


Figura 19: Diagrama de componentes del paquete billing

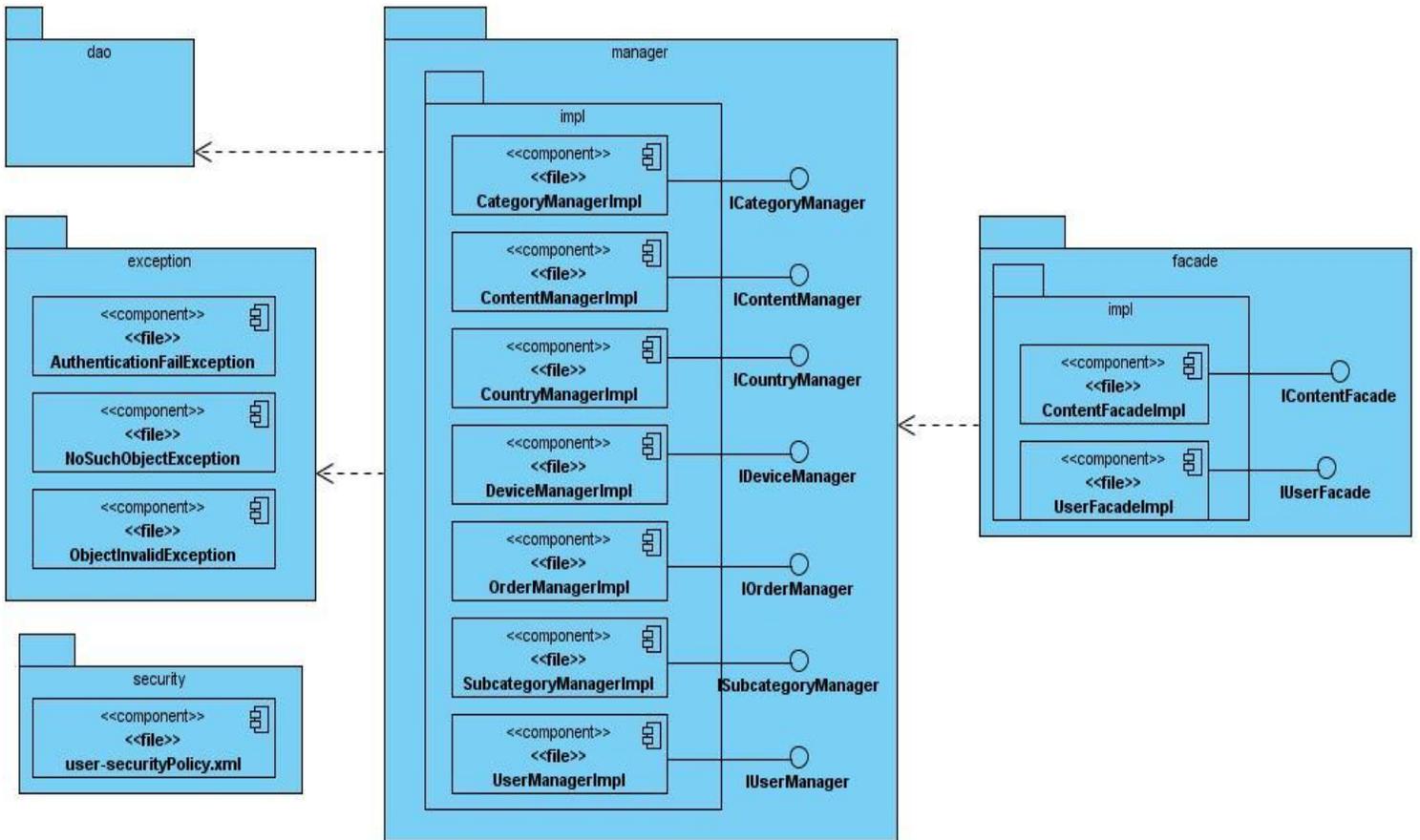


Figura 20: Diagrama de componentes del paquete content

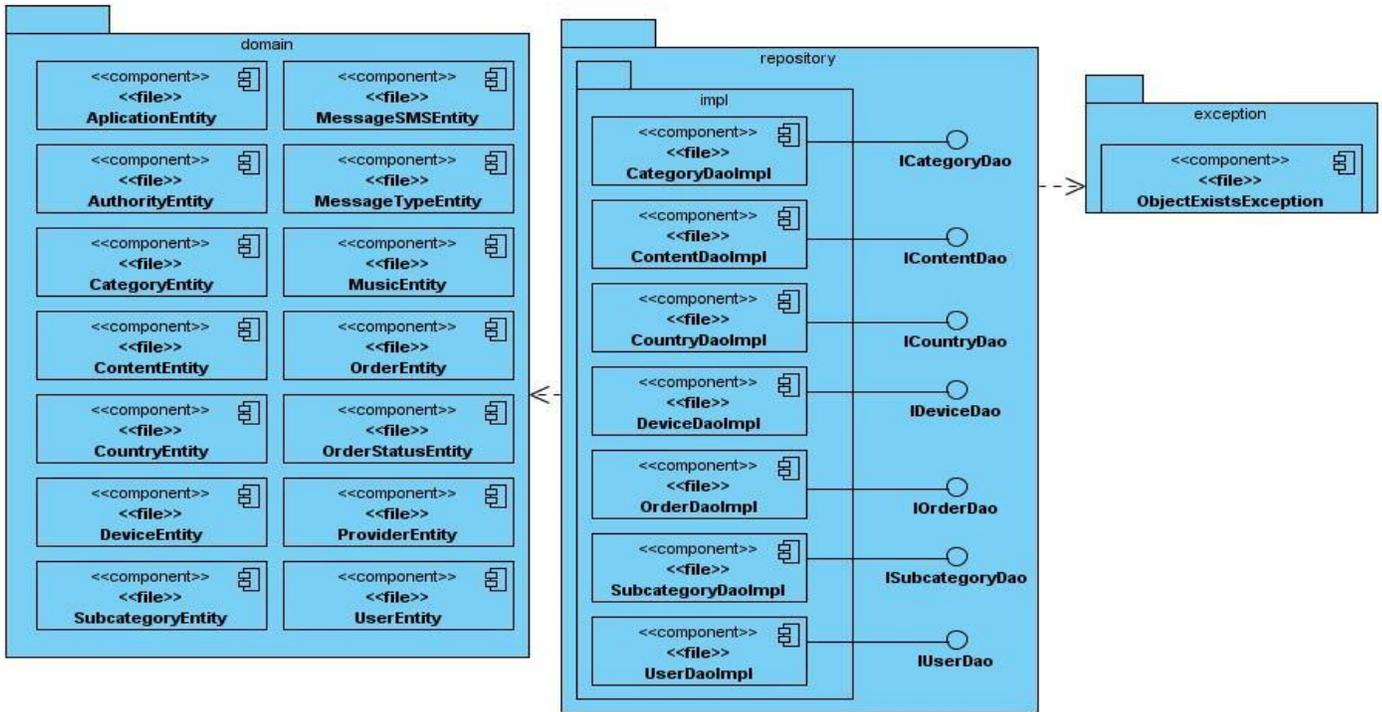


Figura 21: Diagrama de componentes del paquete dao

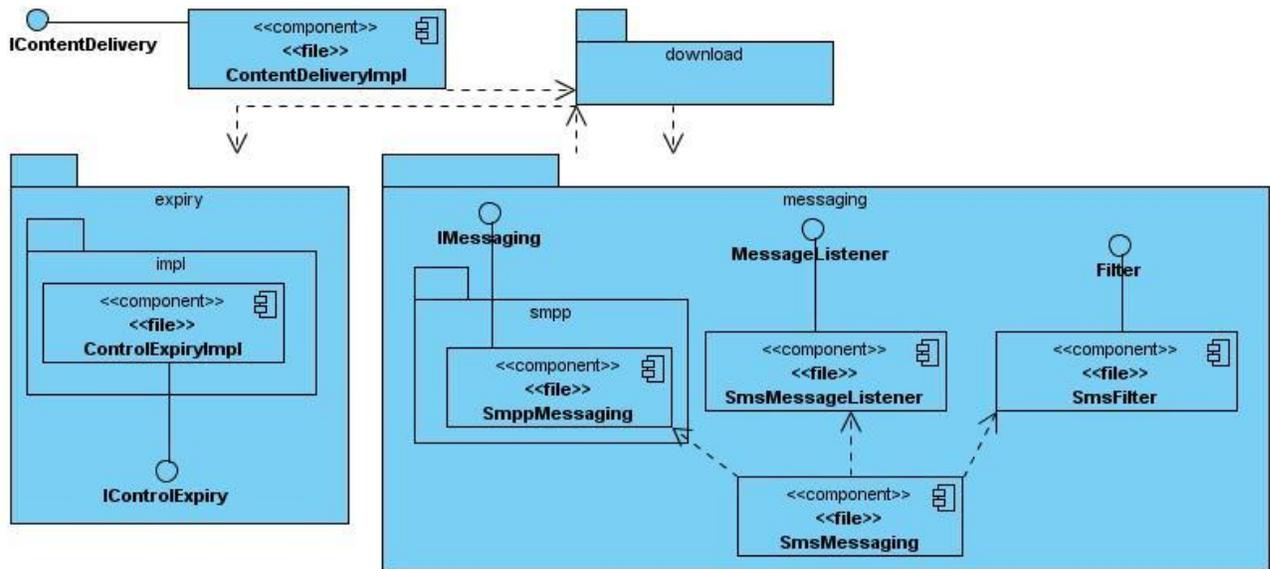


Figura 22: Diagrama de componentes del paquete contentDelivery

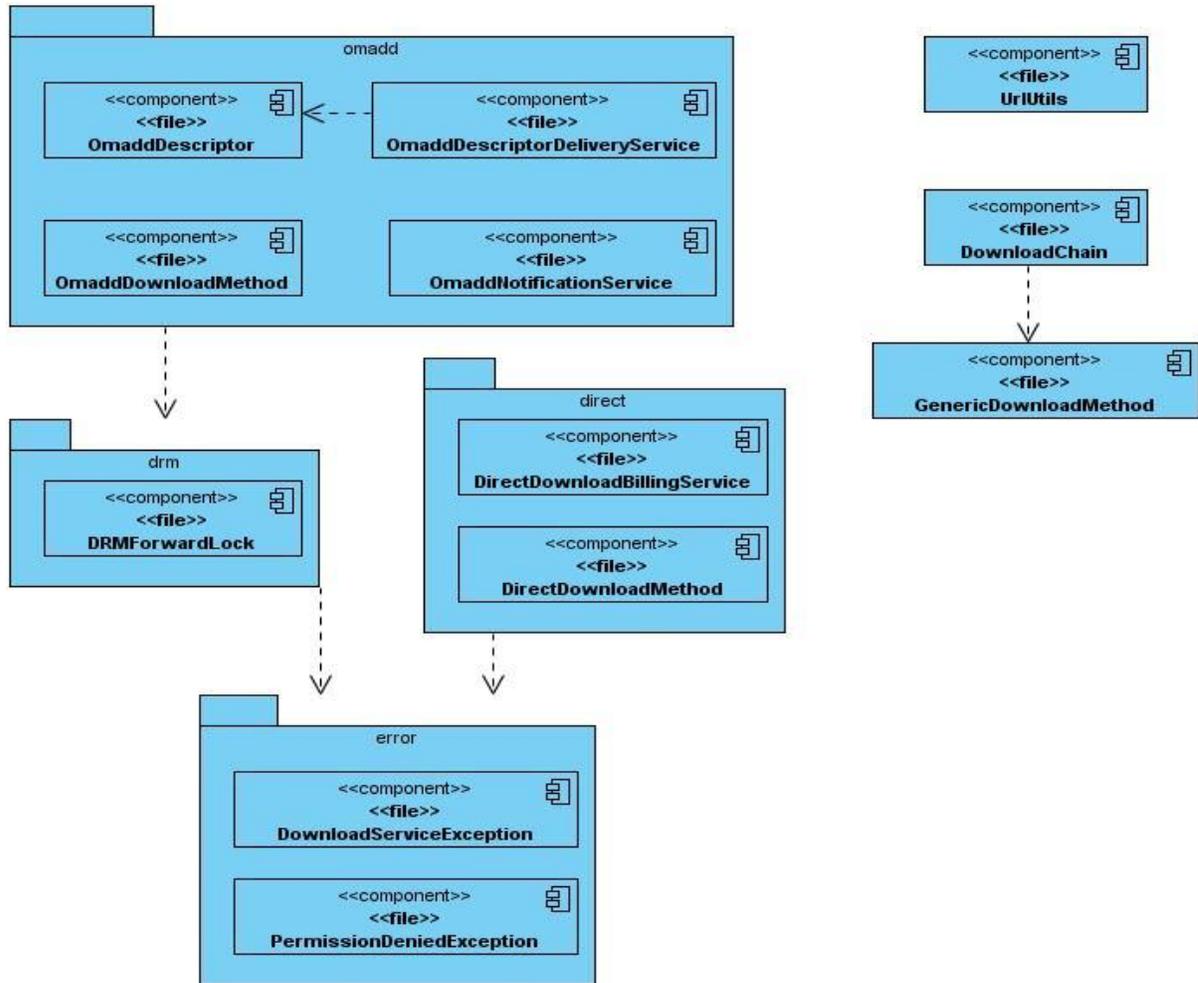


Figura 23: Diagrama de componentes del paquete download

4.3. Conclusiones del capítulo

Los diferentes diagramas de componentes confeccionados en este capítulo, representan en términos de componentes y subsistemas de implementación los elementos del modelo de diseño obtenidos en el capítulo anterior y establecen las dependencias entre uno y otro.

Conclusiones generales

La telefonía móvil es una de las oportunidades de mercado emergente más grandes de los últimos tiempos, lo cual unido al crecimiento de la tecnología en general y la conectividad convierten a las comunicaciones móviles en un fenómeno social de gran magnitud.

En este fenómeno radica la importancia de esta investigación, puesto que el sector del ocio electrónico es uno de los mercados que con más rapidez adoptó el uso de las tecnologías móviles para ampliar su campo de explotación. En este sentido vale destacar la descarga de tonos y logos, servicios que permiten a los usuarios personalizar los móviles con distintos elementos gráficos y melodías de aviso de llamada.

Durante el desarrollo de la plataforma se realizó un proceso de investigación basado en el estado del arte relativo a los métodos empleados para garantizar el servicio de descarga de contenidos hacia el dispositivo móvil, además se pudieron evaluar las diferentes formas para garantizar los permisos exclusivos del proveedor de contenidos y la forma de lograr la compatibilidad del contenido descargado con el dispositivo móvil. 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 y como herramienta de transcodificación se usó Alembik, 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 y una adaptación satisfactoria del contenido al dispositivo móvil.

La plataforma funciona como contenedor con los módulos: Contenido, Mensajería, Facturación, Transcoder, Entrega de Contenido, Portal WAP, Portal WEB, Administración permitiendo realizar operaciones que van desde la gestión de la solicitud del cliente hasta la posterior entrega y pago del contenido.

Se brinda la posibilidad de interactuar con distintos proveedores de contenidos, permitiendo a la VPSM establecer contratos con los mismos para adicionar, modificar y eliminar contenidos que luego serán descargados por los clientes de telefonía celular. Además tiene la posibilidad de darle promoción a los

Conclusiones generales

contenidos a través del Portal WEB donde los clientes pueden ver los contenidos ofertados y hacer pedido del que deseen.

De manera general el desarrollo de la Plataforma de Gestión de Contenidos para Dispositivos Móviles permitirá a la Vicepresidencia de Servicios Móviles de ETECSA (VPSM) solucionar 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.

Recomendaciones

Como resultado del presente trabajo se obtuvo una plataforma de gestión de contenidos 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 la descarga utilizando MMS.
- ✓ 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.

Referencias bibliográficas

1. BuenasTareas.com. *Cuando Se Ivento El Telefono*. [En línea] 02 de 10 de 2009. [Citado el: 25 de 04 de 2010.] <http://www.buenastareas.com/ensayos/Cuando-Se-Ivento-El-Telefono/22157.html>. 2
2. **Spiralis**. Spiralis. [En línea] [Citado el: 05 de 10 de 2009.] <http://www.spiralis.com.mx/serviciosva.php>. 2
3. DRM: Deliberadamente Defectuosos. [En línea] 05 de 05 de 2008. [Citado el: 09 de 02 de 2010.] <http://www.fsfla.org/svnwiki/texto/drm-deliberdefect.es.html>. 3.
4. **Arienza, Jorge L.** El Observatorio de Internet Móvil. *Protección Anti-Copia para Internet Movil*. [En línea] 21 de 11 de 2006. [Citado el: 09 de 02 de 2010.] <http://larienza.blogspot.com/2006/11/proteccion-anti-copia-para-internet.html>. 4.
5. **Alonso, José Antonio y Jiménez, Juan Carlos.** revista TELOS. *Aspectos conceptuales y analíticos.El valor de la lengua como capital social*. [En línea] 2009. [Citado el: 09 de 02 de 2010.] <http://sociedadinformacion.fundacion.telefonica.com/telos/articulocuaderno.asp?idarticulo=3&rev=69.htm>.
6. **Whatis.com**. What is transcoding? - Definition from Whatis.com. [En línea] 10 de 12 de 2008. [Citado el: 09 de 02 de 2010.] http://searchsoa.techtarget.com/sDefinition/0,,sid26_gci515589,00.html. 6.
7. **Danilewicz, Jakub.** portfolio-es. [En línea] [Citado el: 09 de 02 de 2010.] <http://jay-dee.atspace.com/portfolio-es.html>. 7.
8. **Mendoza Sanchez, María A.** <http://www.informatizate.net>. *Metodologías De Desarrollo De Software*. [En línea] 07 de 06 de 2004. [Citado el: 30 de 01 de 2010.] http://www.informatizate.net/articulos/metodologias_de_desarrollo_de_software_07062004.html. 8.
9. HERRAMIENTAS CASE. [En línea] [Citado el: 30 de 01 de 2010.] <http://www.cyta.com.ar/biblioteca/bddoc/bdlibros/proyectoinformatico/libro/c5/c5.htm>. 9.
10. **Visual Paradigm International Ltd.** Visual Paradigm for UML (ME) - (Paradigma Visual para UML (ME)) (Visual Paradigm for UML (ME)) 6.0. [En línea] 05 de 03 de 2007. [Citado el: 30 de 1 de 2010.] [http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_\(M%C3%8D\)_14720_p/](http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_(M%C3%8D)_14720_p/). 10.
11. **Universidad de Alicante.** Especialista Universitario en Java Enterprise. [En línea] 2009. [Citado el: 30 de 01 de 2010.] <http://www.jtech.ua.es/j2ee/index.html>. 11.
12. Programación Java para dispositivos móviles. [En línea] [Citado el: 30 de 01 de 2010.] http://www.red21.com/programacion_java_moviles.htm. 12.

Referencias bibliográficas

13. **Mendez, Justo.** Lenguajes de programación - Monografias.com. [En línea] [Citado el: 30 de 01 de 2010.] <http://www.monografias.com/trabajos/lengprog/lengprog.shtml>. 13.
14. **Luciano.** Entornos de Desarrollo Integrado para Java. [En línea] [Citado el: 30 de 01 de 2010.] <http://luauf.com/2008/05/13/entornos-de-desarrollo-integrado-para-java/>. 14.
15. **Dept. Informatica, Universitat de Valencia.** El entorno de desarrollo Eclipse. [En línea] 2004. [Citado el: 03 de 02 de 2010.] http://www.uv.es/~jgutierrez/MySQL_Java/TutorialEclipse.pdf. 15.
16. **Dupuis, Christian.** SpringSource. [En línea] 2009. [Citado el: 03 de 02 de 2010.] http://www.springsource.com/files/2009-05_STS_v2.1_Feature_Comparison.pdf. 16.
17. **Chololo.** SpringHispano.org | Spring Framework para la comunidad hispanoamericana. [En línea] 28 de 04 de 2006. [Citado el: 04 de 02 de 2010.] <http://springhispano.org/?q=node/35>. 17.
18. **Hibernate.** Relational Persistence for Java and .NET. [En línea] 31 de 01 de 2010. [Citado el: 04 de 02 de 2010.] <https://www.hibernate.org/>. 18.
19. **el holgazán.** JPA: Java Persistence API. [En línea] 30 de 08 de 2007. [Citado el: 04 de 02 de 2010.] <http://www.elholgazan.com/2007/08/jpa-java-persistence-api.html>. 19.
20. PostgreSQL - Servidor de base de datos relacional libre. [En línea] 29 de 07 de 2006. [Citado el: 05 de 02 de 2010.] <http://www.solodisenio.com/postgresql-servidor-de-base-de-datos-relacional-libre/>. 20.
21. **González, Carlos D.** Curso Base de Datos PostgreSQL, SQL avanzado y PHP. [En línea] 02 de 2010. [Citado el: 05 de 02 de 2010.] <http://www.usabilidadweb.com.ar/postgre.php>. 21.
22. **agapea.com.** Libro: Apache Tomcat 6. [En línea] 29 de 01 de 2008. [Citado el: 06 de 02 de 2010.] <http://www.agapea.com/libros/Apache-Tomcat-6-isbn-8441523770-i.htm>. 22.
23. Ayuda del Rational. Fase de Inicio. Flujo de trabajo de requerimiento. 2008. [En línea] 23.
24. Ayuda del Rational. Artefacto: Modelo de diseño. [En línea] 24.
25. Tutorial de UML - Diagrama de Interacción. [En línea] [Citado el: 27 de 04 de 2010.] <http://www.dcc.uchile.cl/~psalinas/uml/interaccion.html>. 25.
26. **Peláez, Juan Carlos.** ARQUITECTURA BASADA EN COMPONENTES. [En línea] 14 de 04 de 2009. [Citado el: 27 de 04 de 2010.] <http://www.dcc.uchile.cl/~psalinas/uml/interaccion.html>. 26.
27. **Casal, Julio.** Desarrollo de Software Basado en Componentes. [En línea] 08 de 05 de 2006. [Citado el: 06 de 02 de 2010.] <http://msguayaquil.com/blogs/julioc/archive/2006/05/08/Desarrollo-de-Software-Basado-en-Componentes.aspx>. 27.

Referencias bibliográficas

28. **Soto, Lauro.** Modelo Basado En Componentes Diseño De Sistemas. [En línea] [Citado el: 06 de 02 de 2010.] <http://www.mitecnologico.com/Main/ModeloBasadoEnComponentesDise%F1oDeSistemas>. 28.
29. **Gracia, Joaquin.** Modelo Basado En Componentes Diseño De Sistemas. [En línea] 27 de 05 de 2005. [Citado el: 06 de 02 de 2010.]
<http://www.mitecnologico.com/Main/ModeloBasadoEnComponentesDise%F1oDeSistemas>. 29.
30. **Manuel B.** Modelo Basado En Componentes Diseño De Sistemas. [En línea] 24 de 02 de 2002. [Citado el: 06 de 02 de 2010.]
<http://www.mitecnologico.com/Main/ModeloBasadoEnComponentesDise%F1oDeSistemas>. 30.
31. **Welicki, León.** El Patrón Singleton. [En línea] [Citado el: 27 de 04 de 2010.]
<http://msdn.microsoft.com/es-es/library/bb972272.aspx#EDAALeo%20Welicki>. 31.
32. Ayuda del Rational. Artefacto: Modelo de implementación. [En línea] 32.
33. Guía de Usuario de Enterprise Architect 7.0. *Diagrama de Despliegue*. [En línea] [Citado el: 06 de 05 de 2010.] <http://www.sparxsystems.com.ar/download/ayuda/index.html?deploymentdiagram.htm>. 33.
34. Guía de Usuario de Enterprise Architect 7.0. *Diagrama de Componentes*. [En línea] [Citado el: 06 de 05 de 2010.] <http://www.sparxsystems.com.ar/download/ayuda/index.html?componentdiagram.htm>. 34.