



Universidad de las Ciencias Informáticas

Polo de Telecomunicaciones

Facultad 2

# **“Plataforma de Gestión de Contenidos para Dispositivos Móviles. Módulo Contenidos”**

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

Autor:

Raudel José Rosquete Martínez

Tutores:

Ing. Rosa María Rente Labrada

Ing. Lex Karel Zayas Hernández

Ciudad de la Habana, Junio 2009

# Declaración de Autoría

Declaro que soy el único autor de este trabajo y autorizo al Polo de Telecomunicaciones, perteneciente a la Facultad 2 de la Universidad de Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmo la presente a los 24 días del mes de Junio de 2009.

---

Raudel J. Rosquete Martínez  
Autor

---

Ing. Rosa M. Renté Labrada  
Tutor

---

Ing. Lex K. Zayas Hernández.  
Tutor



Para ñeñá, quien siempre soñó verme convertido en un profesional. Espero que ahora vea su sueño hecho realidad donde quiera que esté.

Mi agradecimiento para todos los amigos que me acompañaron en estos 5 años de estudio, nombrarlos a todos haría una lista interminable y cometería el error de dejar fuera algunos, no porque fueran menos importantes, sino porque mi memoria me traicionaría a la hora de escribir estas líneas.

En especial para los que llegaron a convertirse en mi familia, Linet, Alejandro, los 101 dálmatas, los profesores y trabajadores del Centro Cultural y Recreación, el Grupo Infodanz, Silvia Rita, Ludia, los profesores y trabajadores de la facultad 2, Julio, Marino y Polanco.

Muchas gracias también para los que colaboraron de alguna manera en la elaboración de este trabajo, Yanet, Leo, Evelyn, Areanne, mis tutores Rosa y Lex Karel y a Plá, Darien y Gerardo por su ayuda desinteresada.

Por último agradecer a mi familia por todo lo que representan y por el apoyo de siempre.

*"El trabajo es un premio en ciertos casos, un instrumento de educación en otros, jamás un castigo"*

# RESUMEN

La telefonía móvil o celular se conoce desde mediados del siglo XX, aunque no fue hasta la década del '70 que comenzó a popularizarse. Desde entonces, la permanente investigación y adelantos tecnológicos permiten enmarcar etapas bien definidas dentro de las comunicaciones móviles, denominadas generaciones, de manera que un teléfono móvil de última generación permite no solo realizar llamadas, sino reproducir música y videos, tomar fotos y pequeñas películas, navegar por internet, enviar y recibir mensajes de texto y gráficos, acceder a prestaciones como juegos electrónicos e incluso ya se puede ver por ellos televisión digital. (1)

Esta evolución tecnológica es un facilitador de nuevos modelos de negocio donde los operadores de telefonía móvil se esfuerzan en buscar servicios de valor añadido que aporten un carácter diferenciador en su catálogo de productos. Demuestra que las comunicaciones por redes móviles junto con la banda ancha son los servicios con mayor crecimiento económico dentro de las Tecnologías de la Información y las Comunicaciones (TIC) en los últimos años. (2)

Inmersos en esta revolución económica, los sistemas de gestión de contenidos para dispositivos móviles aportan a los operadores móviles mayor capacidad al nivel de aplicación y de servicio, cubriendo las necesidades requeridas para fortalecerse en un mercado tan transformador. Se encargan de centralizar y gestionar una serie de procesos que comienzan con la captación del contenido desde cualquier fuente hasta su presentación final al cliente. Entre estos procesos pueden distinguirse algunos como la gestión de contenidos, la gestión de conversión de formatos, la gestión de facturación y la gestión de descarga y derechos digitales.

La telefonía móvil cubana también se enfrenta a los retos de la revolución tecnológica. La Vicepresidencia de Servicios Móviles (VPSM) de la Empresa de Telecomunicaciones de Cuba (ETECSA), único operador de telefonía móvil del país, se propone ofrecer servicios y contenidos para los móviles, llevando a cabo acciones como el desarrollo de una Plataforma de Gestión de Contenidos para Dispositivos Móviles.

Este trabajo tiene como objetivo el desarrollo del Módulo Contenidos de dicha plataforma, el cual es el encargado del aprovisionamiento de contenidos desde cualquier fuente y de la gestión de sus datos, lo que permite a la VPSM una rápida incorporación al mercado de los contenidos para dispositivos móviles.

# TABLA DE CONTENIDOS

Resumen.....	IV
Introducción .....	1
Capítulo I: Fundamentación teórica .....	4
1.1.    Introducción .....	4
1.2.    Gestión de contenidos para dispositivos móviles.....	4
1.3.    Módulo Contenidos.....	6
1.4.    Estado del arte. ....	7
1.4.1.    Content Delivery Platform .....	7
1.4.2.    MProvision Mobile Content Platform.....	7
1.4.3.    Mobile Web & Content Delivery Platform.....	8
1.5.    Tecnologías y herramientas a utilizar.....	8
1.5.1.    Metodología de desarrollo de software.....	8
1.5.1.1.    Rational Unified Process (RUP).....	9
1.5.2.    UML como lenguaje de modelado.....	10
1.5.3.    Herramienta CASE para la modelación del sistema.....	10
1.5.3.1.    Visual Paradigm .....	10
1.5.4.    Java como plataforma de desarrollo.....	11
1.5.5.    Entorno de Desarrollo Integrado.....	13
1.5.5.1.    Eclipse Ganymede .....	13
1.5.6.    Framework.....	13
1.5.6.1.    Spring.....	14
1.5.6.2.    Hibernate.....	14
1.5.6.3.    JUnit.....	15
1.5.7.    Especificaciones.....	15
1.5.7.1.    Java Persistence API (JPA) .....	15
1.5.8.    Web Service.....	16
1.5.9.    PostgreSQL servidor de base de datos.....	16
1.5.10.    Apache Tomcat como servidor web.....	16
1.6.    Conclusiones.....	17

Capítulo II: Características del sistema .....	18
2.1. Introducción .....	18
2.2. Propuesta del sistema .....	18
2.3. Modelo de Dominio.....	19
2.3.1. Diagrama de Clases del dominio .....	19
2.4. Relación de requerimientos.....	20
2.4.1. Requisitos funcionales. ....	20
2.4.2. Requisitos no funcionales .....	23
2.5. Modelo de Casos de Usos del sistema .....	24
2.5.1. Diagrama de Casos de Usos .....	24
2.5.2. Definición de los actores .....	25
2.5.3. Listado de casos de usos.....	26
2.6. Conclusiones .....	29
Capítulo III: Análisis y Diseño del sistema .....	30
3.1. Introducción .....	30
3.2. Análisis y Diseño del sistema .....	30
3.2.1. Definición del Modelo del Análisis .....	30
3.2.1. Diagrama de Clases del Análisis.....	30
3.2.2. Definición del Modelo de Diseño .....	30
3.2.3. Diagrama de Clases del Diseño .....	31
3.2.4. Descripción del Diagrama de Clases del Diseño .....	33
3.2.5. Diagramas de Interacción .....	33
3.3. Descripción de la Arquitectura.....	35
3.3.1. Patrones utilizados .....	35
3.3.2. Patrón arquitectónico .....	35
3.3.3. Patrones de diseño. Patrones GRASP .....	35
3.4. Modelo de Datos .....	36
3.4.1. Definición del Modelo de Datos.....	36
3.5. Conclusiones .....	37
Capítulo IV: Implementación y Pruebas.....	38
4.1. Introducción .....	38
4.2. Modelo de Implementación .....	38
4.2.1. Diagrama de Componentes .....	38

4.2.2. Diagrama de Despliegue .....	41
4.3. Modelo de Prueba .....	42
4.4. Conclusiones .....	43
Capítulo V: Estudio de factibilidad .....	44
5.1. Introducción .....	44
5.2. Método de estimación “Puntos de Casos de Uso” .....	44
5.2.1. Definición y conceptos fundamentales .....	44
5.2.2. Cálculo de Puntos de Casos de Uso .....	45
5.2.3. Estimación del esfuerzo .....	48
5.2.4. Análisis y estimación del costo .....	50
5.3. Beneficios tangibles e intangibles .....	50
5.4. Conclusiones .....	51
Conclusiones .....	52
Recomendaciones .....	53
Referencias bibliográficas .....	54
Bibliografía .....	56
Glosario de términos .....	60
Anexos .....	67
Anexo I: Descripción de los casos de uso .....	67
Anexo II: Descripción de las clases del diseño .....	81
Anexo III: Diagramas de Secuencia por casos de uso .....	107

# TABLA DE FIGURAS

Figura 1: Diagrama de clases del dominio.....	19
Figura 2: Diagrama de Casos de Uso.....	25
Figura 3: Diagrama de clases del análisis .....	30
Figura 4: Diagrama de paquetes. Vista arquitectónica .....	31
Figura 5: Diagrama de clases del diseño. Subsistema Service .....	31
Figura 6: Diagrama de clases del diseño. Subsistema Resolver.....	32
Figura 7: Diagrama de clases del diseño. Subsistema Repository.....	32
Figura 8: Diagrama de clases del diseño. Subsistema Domain .....	32
Figura 9: Diagrama de Secuencia. Escenario Obtener ID del Contenido.....	34
Figura 10: Diagrama de clases persistentes.....	37
Figura 11: Diagrama Entidad-Relación.....	37
Figura 12: Diagrama de componentes. Subsistema Service .....	39
Figura 13: Diagrama de componentes. Subsistema Resolver.....	39
Figura 14: Diagrama de componentes. Subsistema Repository.....	40
Figura 15: Diagrama de componentes. Subsistema Domain .....	40
Figura 16: Diagrama de despliegue.....	41

# INTRODUCCIÓN

La telefonía móvil se ha caracterizado, desde su aparición, por un progreso vertiginoso en la tecnología que la sustenta. Ha transitado por varias generaciones, todas caracterizadas por el incremento de nuevos servicios, de la velocidad de transmisión y de las posibilidades de negocios.

En la última década, las comunicaciones móviles se han convertido en la tecnología más determinante para el intercambio entre las personas y los negocios. El cambio tecnológico también se manifiesta en la creciente oferta de contenidos, servicios y aplicaciones para dispositivos móviles.

El negocio de la telefonía móvil viene desarrollándose alrededor de dos servicios básicos, la voz y los mensajes cortos. En la mayor parte de los países desarrollados estos dos servicios, con unas tasas elevadas de penetración y uso entre la población, están alcanzando el nivel de saturación. La llamada tercera generación y los sistemas integrados de banda ancha han sido la gran esperanza y es el gran desafío ante el cual los operadores necesitan encontrar la forma de obtener rentabilidad de los nuevos servicios.

Según un estudio de la Consultora Global para el Crecimiento Empresarial Frost & Sullivan titulado: Mercado de Contenido Móvil en América Latina, el mercado de contenido móvil y servicios en América Latina creció cerca de 120 por ciento de 2004 a 2005 en la región. (3)

El estudio de Frost & Sullivan demostró que el mercado latinoamericano de los servicios de contenido móvil sumó \$594.2 millones en ingresos en 2005 y se espera que alcance \$2.61 mil millones antes de 2012. (3)

"El contenido móvil es cada vez más popular porque los operadores móviles están comprendiendo que el contenido móvil es un negocio provechoso..."<sup>1</sup> por lo que es responsabilidad de estos aumentar el mercado móvil de contenidos y servicios.

Se hace de vital importancia, teniendo en cuenta el aumento acelerado de los ingresos por servicios de contenido móvil y la imponente penetración en la población mundial de estos servicios, que los operadores móviles se replanteen la forma de consolidarse en el mercado.

Teniendo en cuenta estas observaciones, los operadores telefónicos se ven en la obligación de adentrarse en el negocio de los servicios móviles y para ello tienen la necesidad de implementar Sistemas de Gestión que sean capaces de englobar una serie de procesos, que van desde la captación de los servicios, contenidos y aplicaciones, hasta la forma que adoptan estos cuando son

---

<sup>1</sup> Según declaraciones de Víctor Casiano, analista de Frost & Sullivan.

ofrecidos finalmente al usuario. Este sistema también deberá encargarse de proporcionar políticas de control de acceso y gestión de los distintos roles definidos en el sistema.

En Cuba, la VPSM desplegó un Portal WAP desarrollado en la plataforma .NET con funcionalidades básicas como promoción, estado del tiempo y directorio telefónico de entidades, pero no cuenta con una herramienta que provea prestaciones suficientes para administrar estos servicios; además no se ofrecen servicios de valor agregado tales como la descarga de tonos, melodías, logos y fondos.

A partir de la instalación de nuevos servicios por parte de esta entidad y en el marco de los acuerdos de colaboración entre ETECSA y la Universidad de las Ciencias Informáticas (UCI), se decide dar solución a esta **Situación Problemática** con la implementación de la Plataforma de Gestión de Contenidos para Dispositivos Móviles (GINA). Este sistema contará con tres componentes; un Portal WAP, un Sistema de Administración de Contenidos (CMS) y una Plataforma de Servicios.

El primer componente permitirá a los clientes móviles acceder a los servicios que se pondrán a su disposición, el segundo será la interfaz utilizada por los usuarios de la VPSM para administrar las distintas funcionalidades de GINA y el tercer componente encargado de desarrollar dichas funcionalidades.

Entre estas funcionalidades se encuentra el proceso de gestión de los contenidos para dispositivos móviles, por esta razón se plantea como **Problema Científico** de esta investigación ¿Cómo facilitar la gestión de los contenidos para dispositivos móviles en la Plataforma GINA de la VPSM?

Se define como **Objeto de Estudio** el proceso de gestión de contenidos para dispositivos móviles y como **Campo de Acción** la gestión de contenidos para dispositivos móviles en la Plataforma de Servicios de la Plataforma GINA.

El **Objetivo General** de la investigación es desarrollar el Módulo Contenidos para la Plataforma de Servicios de la Plataforma GINA.

Para dar cumplimiento a los Objetivos presentados, se definen como **Tareas de la Investigación**:

- Estudio de los Sistemas Existentes para la Gestión de Contenidos.
- Estudio sobre el proceso de gestión de los Contenido para dispositivos móviles, quienes participan, como se lleva a cabo y las diferentes formas que se utilizan según los proveedores involucrados.
- Encuentros periódicos con el personal calificado de la VPSM.
- Estudio de las Herramientas y Tecnologías utilizadas para el desarrollo de la Plataforma GINA.
- Estudio de la organización para obtener una visión adecuada de la misma y de los procesos que requieren automatización.
- Análisis, Diseño, Implementación y Prueba de la solución propuesta.
- Estudio de Factibilidad.

El desarrollo del Módulo Contenidos es una pieza esencial para el funcionamiento de GINA, la que permitirá a la VPSM posicionarse en un nuevo modelo de negocio en el cual los operadores dejan de ser, para los clientes, solo un proveedor de telefonía móvil para convertirse en un proveedor de soluciones completas, que incluye servicios, contenidos y aplicaciones. Se incrementa además la posibilidad de iniciar negocios con proveedores como Empresas Discográficas, ampliando el mercado de la VSPM.

Este trabajo ha sido estructurado de la siguiente forma:

**Capítulo 1 Fundamentación Teórica:** Contiene una descripción de la Gestión de Contenidos, los procesos que serán automatizados y un breve estudio de los principales conceptos tratados en el trabajo. Cuenta con una reseña de otros sistemas utilizados en el mundo para iguales propósitos y un estudio de las tecnologías y herramientas actuales, haciendo énfasis en las utilizadas en el desarrollo de la Plataforma GINA.

**Capítulo 2 Características del Sistema:** Contiene la descripción de los tipos de objetos más importantes que existen en el dominio. Se definen como se interrelacionan estos objetos. Contiene además la captura de los requisitos funcionales y no funcionales, el diagrama de casos de usos del sistema, así como las descripciones de los casos de usos.

**Capítulo 3 Análisis y Diseño del Sistema:** Se define el modelo de clases del análisis y el diseño, los diagramas de interacción, así como las descripciones de las clases identificadas. Se describe además el diseño de la Base de Datos utilizada.

**Capítulo 4 Implementación y Prueba:** Contiene una descripción y representación gráfica del modelo de implementación y despliegue del sistema. Cuenta además, con la descripción de los casos de prueba de integración.

**Capítulo 5 Estudio de Factibilidad:** Contiene una estimación del tiempo de duración del proyecto de software utilizando el método Puntos de Casos de Uso y una estimación del costo del mismo. Además cuenta con un breve análisis de los beneficios tangibles e intangibles que se esperan del mismo.

## 1.1. Introducción

Este capítulo contiene un estudio crítico y valorativo sobre el proceso de Gestión de Contenidos para Dispositivos Móviles. Incluye el estado actual de este proceso, así como las tendencias tecnológicas y metodológicas más utilizadas por los desarrolladores de soluciones para este fin.

Se definen las herramientas y tecnologías que se utilizarán en el desarrollo del sistema propuesto y en capítulos posteriores será la base para la construcción de la arquitectura del sistema.

## 1.2. Gestión de contenidos para dispositivos móviles.

La Tercera Generación de las comunicaciones móviles se caracteriza por contener la convergencia de voz y datos con acceso inalámbrico a Internet; en otras palabras, está apta para aplicaciones multimedia y altas transmisiones de datos. Los protocolos empleados en los sistemas de tercera generación soportan altas velocidades de información, han traído consigo la aparición de tecnologías de comunicación que permitirán llevar voz, gráficos, vídeo y otros tipos de datos de banda ancha directamente al usuario, independientemente de la red o dispositivo móvil utilizado.

Un dispositivo móvil se define como un dispositivo inalámbrico electrónico que permite tener acceso a la red de telefonía celular o móvil. Su principal característica es su portabilidad y su función esencial es la comunicación de voz, como el teléfono convencional. Su rápido desarrollo ha incorporado otras funciones como son cámara fotográfica, agenda, acceso a Internet, reproducción de video e incluso GPS y reproductor mp3.

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.

En la actualidad la mayoría de los operadores de telefonía móvil ofrecen un gran número de nuevos servicios a través de otras entidades: meteorológicos, noticias, juegos, acceso a bancos, localización, información de tráfico, información turística, restaurantes, etc. De ellos, la descarga de tonos y logos es el ejemplo más claro de que la comunicación por el móvil se encuentra en la etapa expresiva, como ya fue descrito anteriormente.

## Capítulo I: Fundamentación Teórica

Un contenido para dispositivos móviles puede definirse como un objeto que se ofrece al cliente móvil para satisfacer sus deseos y necesidades. Son entregados como resultado del acceso a un servicio ofrecido por los operadores de telefonía móvil, 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 etc.

Dentro de los contenidos se pueden especificar contenidos de valor agregados (CVA): son aquellos que aportan un valor monetario para el operador y que el cliente puede comprar para descargar en su dispositivo móvil, por ejemplo los logo, tonos, melodías, wallpaper y juegos entre otros.

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.

Los procesos involucrados dentro del flujo de trabajo clásico de una herramienta de gestión de contenidos engloban una serie de fases que abarcan el flujo completo de los servicios soportados. A continuación se presentan algunas de estas fases:

**Captación:** En esta fase los datos se encuentran inicialmente “en bruto” y pueden proceder de varias fuentes; bases de datos de proveedores, servicios on-line, agencias de información, etc. así como tener diferentes vías de comunicación. El principal cometido de esta fase es, por tanto, capturar esta información y transformarla de tal forma que quede almacenada en un formato unificado que permita su procesado posterior.

La tendencia actual, buscando facilitar esta tarea, es usar como estructura de intercambio documentos basados en XML (Extensible Markup Language), para los que se define de antemano una DTD (Document Type Definition) común a proveedores y receptores.

**Categorización:** En esta fase los contenidos son caracterizados y validados, de modo que posteriormente puedan ser localizados y usados de modo lógico y ordenado. En la medida de lo posible este proceso suele automatizarse, aunque tiene sus limitaciones; no se conoce el contenido de las imágenes o desconoce la semántica de los textos, y por tanto en muchos casos necesita de una supervisión manual.

## Capítulo I: Fundamentación Teórica

**Distribución:** Esta fase tiene como objetivo proporcionar a los clientes, de manera organizada y flexible, aquellos contenidos a los que se han suscrito o han solicitado. Para que esta difusión de información se realice con éxito, ambas partes deben poseer un vocabulario y protocolo de intercambio común.

**Publicación:** En esta fase finaliza el flujo de trabajo definido y es la encargada de poner los contenidos a disposición de los usuarios, tras recibir la debida autorización.

Para facilitar la complejidad asociada a las tecnologías de descarga (protocolo, codificación, características de los terminales, etc.) siendo estas funciones transparentes a los servicios y al usuario, la gestión de contenidos a dispositivos móviles de cualquier operadora está compuesta por un conjunto de módulos principales, entre los que se pueden mencionar *El módulo de gestión de contenidos*, *El módulo de gestión de conversión de formatos*, *El módulo de gestión de facturación*, *El módulo de gestión de usuarios*, *El módulo de gestión de descarga y de tecnologías de descarga*, entre otros. Cada desarrollador puede agregar o eliminar módulos diferentes según la información y actividades que se necesite manejar en el sistema que se proponga implementar.

Todos estos módulos son componentes autocontrolados del sistema, por lo que debe ser sencillo su ensamblaje con el resto de los módulos. O sea, serán los encargados de realizar una o varias tareas del total de actividades automatizadas. Por esa razón, se hace necesario implementar un mecanismo de integración e intercambio de información que permita la interoperabilidad y extensibilidad entre los módulos.

### 1.3. Módulo Contenidos

Tiene como objetivo proporcionar el espacio de trabajo necesario para cumplir con las funciones relacionadas con los contenidos, desde la captación de los mismos hasta su presentación al usuario final.

El Módulo Contenidos permitirá realizar la provisión de cualquier tipo de contenidos mediante diversas vías de suministro. Además coordina las actividades de conversión de formato de los contenidos para terminales móviles con características distintivas, así como de las actividades de búsqueda, modificación, entrega y actualización de los contenidos en el repositorio de contenido.

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.

Existen muchas empresas que se han dedicado a la gestión de contenidos para los dispositivos móviles y que han hecho productos con una gran calidad. Líderes en esta esfera, dentro de los productos podemos mencionar “Content Delivery Platform”, de la empresa Ginger Alliance, “MProvision Mobile Content Platform” un sistema creado por la empresa Mgenie y “Mobile Web & Content Delivery Platform” creado por la empresa BeeWeb.

En Cuba, el uso de teléfonos celulares ha tenido un desarrollo lento y menos vertiginoso que en el resto del mundo, por esta razón la demanda por estos contenidos es muy poca y no existe ningún sistema de gestión de contenidos.

#### 1.4.1. Content Delivery Platform

GA-CDP (Ginger Alliance-Content Delivery Platform) es una solución completa para una fiable presentación, entrega y facturación de los contenidos digitales adecuada al dispositivo móvil y al canal de comunicación que utiliza.

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, Internet, servicios VoIP, entre otros.

GA-CDP incluye la Plataforma de facturación móvil que permite la incorporación de servicios de facturación directa en la cadena de suministro. Además implementa políticas DRM y una plataforma donde se aplica el formato de presentación necesario a los contenidos según la interfaz de usuario y el dispositivo final. (4)

#### 1.4.2. MProvision Mobile Content Platform.

MProvision es un sistema creado para facilitar la comercialización, facturación y entrega de contenido móvil a los usuarios finales. Le permite elegir qué grupos de contenido, métodos de facturación y países desea destacar.

El operador móvil debe comunicarse con los administradores de este sistema y escoger el grupo de contenidos que desea vender, el país en el que desea hacerlo y cómo desea que el usuario final pague los contenidos. Luego se crea una cuenta MProvision para el operador en un formato visual básico o de diseño propio para adaptarse a cualquier sitio web que este posea.

Mgenie se ocupa entonces de todas las facturas, la entrega del producto, la concesión de licencias, regalías y apoyo a los usuarios finales dejando al operador concentrarse en la comercialización de sus

servicios. Además se ocupa de hacer el pago mensual en la cuenta bancaria del operador según sus ingresos. (5)

### 1.4.3. Mobile Web & Content Delivery Platform.

BeeWeb ha madurado una gran experiencia en el suministro de soluciones para la administración y entrega de contenidos y el resultado es MWCDP (Mobile Web & Content Delivery Platform).

El MWCDP tiene una arquitectura modular que puede garantizar un rápido despliegue y una fácil integración con los operadores de instalaciones internas (facturación, mensajería y otros).

La plataforma se basa en un sistema de centralización de funcionalidades. De esta forma, la integración, publicación de contenidos y prestación de servicios a los usuarios, en la parte superior de los módulos de salida personalizables se encargan de adaptar los contenidos para cualquiera de los canales de distribución soportados.

Todos los módulos de salida son manejados por un módulo de administración de servicios central.

Cuenta además con una Librería de Media Flexible para garantizar que todos los contenidos manejados por la plataforma (imágenes, textos, audio, videos, aplicaciones, flash, etc.) sean almacenados con un adecuado formato, tamaño y nivel de compresión.

Además de ser flexible y modular, la plataforma BeeWeb ha sido diseñada y desarrollada para garantizar un nivel de escalabilidad al portador. (6)

### 1.5. Tecnologías y herramientas a utilizar.

El Módulo Contenidos responde a la necesidad de automatización de procesos análogos dentro de la Plataforma de Gestión de Contenido para Dispositivos Móviles, perteneciente al Polo de Telecomunicaciones de la Facultad 2 en la UCI, producto que se desarrolla para la Vicepresidencia de Servicios Móviles de ETECSA.

Por esta razón, en análisis y estudio conjunto entre la dirección y el grupo de arquitectura del polo productivo y del proyecto con la dirección y especialistas de ETECSA, se seleccionaron un grupo de herramientas y tecnologías necesarias para el desarrollo de este sistema, las cuales se detallan a continuación.

#### 1.5.1. Metodología de desarrollo de software.

Una metodología de desarrollo es un conjunto de procedimientos, técnicas, herramientas y un soporte documental que ayuda a los desarrolladores a realizar un software.

## Capítulo I: Fundamentación Teórica

En un proyecto de desarrollo de software la metodología define Quién debe hacer Qué, Cuándo y Cómo debe hacerlo. No existe una metodología de software universal. Las características de cada proyecto exigen que el proceso sea configurable.

### 1.5.1.1. Rational Unified Process (RUP)

Un proyecto realizado siguiendo RUP se divide en 4 fases: Inicio, Elaboración, Construcción y Transición. En cada fase se ejecutarán una o varias iteraciones (de tamaño variable según el proyecto) y dentro de cada una de ellas seguirá un modelo de cascada para los flujos de trabajo que requieren las actividades anteriormente citadas.

RUP define 9 Flujos de Trabajo:

- **Modelamiento del negocio:** Describe los procesos de negocio, identificando quiénes participan y las actividades que requieren automatización.
- **Requerimientos:** Define qué es lo que el sistema debe hacer, para lo cual se identifican las funcionalidades requeridas y las restricciones que se imponen.
- **Análisis y diseño:** Describe cómo el sistema será realizado a partir de la funcionalidad prevista y las restricciones impuestas (requerimientos), por lo que indica con precisión lo que se debe programar.
- **Implementación:** Define cómo se organizan las clases y objetos en componentes, cuáles nodos se utilizarán y la ubicación en ellos de los componentes y la estructura de capas de la aplicación.
- **Prueba (Testeo):** Busca los defectos a lo largo del ciclo de vida.
- **Instalación:** Produce versiones del producto y realiza actividades (empaquete, instalación, asistencia a usuarios, etc.) para entregar el software a los usuarios finales.
- **Administración del proyecto:** Involucra actividades con las que se busca producir un producto que satisfaga las necesidades de los clientes.
- **Administración de configuración y cambios:** Describe cómo controlar los elementos producidos por todos los integrantes del equipo de proyecto en cuanto a: utilización/actualización concurrente de elementos, control de versiones, etc.
- **Ambiente:** Contiene actividades que describen los procesos y herramientas que soportarán el equipo de trabajo del proyecto; así como el procedimiento para implementar el proceso en una organización.

RUP se basa en Casos de Usos para describir lo que se espera del software y está orientado a la arquitectura del sistema. Es un proceso iterativo e incremental que utiliza UML (Unified Modeling Language) como Lenguaje de Modelado. (7)

Está pensado para proyecto 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 de la Plataforma GINA 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.

RUP es fácilmente adaptable al ambiente de desarrollo de software deseado, por lo que puede configurarse como una metodología ágil. (8)

### 1.5.2. UML como lenguaje de modelado

**UML** son las siglas de **Unified Modeling Language** (Lenguaje Unificado para la Construcción de Modelos), notación con que se construyen sistemas por medio de conceptos orientados a objetos.

Es el lenguaje de modelado más conocido y usado en la actualidad, es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software orientado a objetos. Ofrece un estándar para describir un plano del sistema, incluyendo aspectos conceptuales tales como procesos de negocio y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables.

Ha sido adoptado por muchos proveedores de herramientas CASE y es el lenguaje de modelado soportado por RUP, metodología escogida para el desarrollo del sistema.

### 1.5.3. 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)

#### 1.5.3.1. Visual Paradigm

Es una poderosa herramienta CASE, por excelencia para ser utilizada en un ambiente de software libre, debido a la posibilidad de ejecutarse sobre cualquier sistema operativo, lo que la convierte en una herramienta multiplataforma. Permite crear tipos diferentes de diagramas en un ambiente totalmente visual.

Soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Permite graficar todos los tipos de diagramas de clases, código

inverso, generar código desde diagramas y generar documentación. También proporciona tutoriales demostraciones interactivas y proyectos UML.

Dentro de las principales características 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. (10)

### 1.5.4. Java como plataforma de desarrollo.

La **plataforma Java** es originaria de Sun Microsystems, orientada al desarrollo de software, permite implementar una gran variedad de aplicaciones. Dentro de la plataforma Java se pueden destacar tres subestándares, (11) cada una es un mundo de tecnología disponible para desarrollar familias de aplicaciones:

- JSE (Java Standard Edition). Plataforma base del lenguaje Java donde puedes hacer aplicaciones Stand Alone. Tanto JEE y JME se basan en JSE. Es el conjunto de herramientas software que permite el desarrollo y la ejecución de programas Java. Es gratuito y de libre distribución. Dentro del JSE se incluyen el compilador y la JVM (Java Virtual Machine), que también se conoce como Intérprete de Java, Java Standard Edition, comprende las librerías básicas del lenguaje, así como las APIs más comunes.
- JME (Java Micro Edition). Plataforma del lenguaje Java destinada al desarrollo de aplicaciones para dispositivos móviles. JME es una versión reducida del conjunto de clases de JSE aunque con algunas librerías extra para el desarrollo de aplicaciones para dispositivos móviles. Se trata de una edición especial debido a las limitaciones de recursos en diversas familias de dispositivos (celulares, PDAs, Pockets, PCs, televisiones, relojes, sistemas de ayuda para automóviles, tarjetas, robótica).
- JEE (Java Enterprise Edition). Plataforma del lenguaje Java que provee una especificación de cómo debe construirse una aplicación empresarial. Esta especificación describe cómo los servidores de aplicación deben proporcionar seguridad, escalabilidad, portabilidad, consistencia, manejo transaccional robusto e independencia de la plataforma tanto de hardware como de sistemas operativos, y al mismo tiempo para que el desarrollador final pueda desplegar aplicaciones empresariales con menor esfuerzo.

El desarrollo de aplicaciones en Java nunca empieza desde cero. Java implementa consigo un gran número de clases, incluidas gratuitamente en su entorno de desarrollo, para realizar variadas tareas

## Capítulo I: Fundamentación Teórica

que permiten al desarrollador centrarse en el negocio y no en la implementación. Permite además, la incorporación de un gran número de componentes reutilizables.

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.

Un programa destinado a la Plataforma Java necesita dos componentes en el sistema donde se va a ejecutar: una Máquina Virtual de Java (JVM), y un conjunto de librerías para proporcionar los servicios que pueda necesitar la aplicación. La JVM junto con su implementación de las librerías estándar, se conocen como Java Runtime Environment (JRE) o Entorno en tiempo de ejecución para Java. El JRE es lo mínimo que debe contener un sistema para poder ejecutar una aplicación Java sobre el mismo. (13)

La característica más conocida de Java es su total independencia respecto a los distintos tipos de plataformas utilizadas. Los programas de Java están compilados en ByteCode que, a su vez, es traducido por el intérprete incluido en la Máquina Virtual de Java. Esto significa que el ByteCode compilado (en cierto sentido, código binario ejecutable) puede ejecutarse en cualquier plataforma que disponga de una JVM (Java Virtual Machines) y sobre equipos que no tienen que ser necesariamente computadoras, como son los dispositivos móviles.

Los programas de Java son mucho más seguros que los de C o C++; es decir, existen menos posibilidades de que generen errores graves. Java no tiene punteros. La Máquina Virtual de Java se ocupa además de administrar la memoria, por lo que no existen pérdidas de memoria, uno de los problemas más difíciles de detectar. Esto significa que la mayoría de errores fatales que se producen habitualmente en los programas de C y C++ nunca suceden en los programas de Java. Los programas de Java contienen menos errores y resultan por tanto más seguros que los programas de C y C++ desarrollados con el mismo nivel de esfuerzo y dificultad.

La programación en Java suele ser más productiva que la escrita en C y C++. En su diseño, se incorporaron los últimos resultados en Ingeniería de Software orientado a objetos. Las bibliotecas estándar de Java también recurren con frecuencia a los patrones de diseño, que incrementan su flexibilidad.

Java es Open Source (Software libre o Código abierto). La infraestructura para desarrollar en JAVA es gratuita. Tiene licencia de software libre GPL. Por esta y por las ventajas presentadas anteriormente,

en el desarrollo del proyecto Plataforma GINA se utiliza la plataforma Java, específicamente Java Platform Enterprise Edition (JEE).

### 1.5.5. Entorno de Desarrollo Integrado.

Un Entorno de Desarrollo Integrado IDE (*por sus siglas en inglés 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).

#### 1.5.5.1. Eclipse Ganymede

Eclipse es un entorno de desarrollo integrado de código abierto, multiplataforma, para integrar herramientas de desarrollo, con una arquitectura abierta y basada en plug-ins. Da soporte a todo tipo de proyectos abarcando el ciclo de vida completo en el desarrollo de aplicaciones, incluyendo soporte para modelado. La arquitectura de plug-ins permite integrar diversos lenguajes sobre un mismo IDE e introducir otras aplicaciones accesorias.

Eclipse es soportado por los principales sistemas operativos. Tiene además una comunidad de desarrolladores, realmente activa. Continuamente se están desarrollando nuevos plug-ins y revisando los anteriores. Al igual que en cualquier otro software, Eclipse permite la instalación de plug-ins destinados a mejorar las funcionalidades del propio IDE y a extenderse en cada vez más tecnologías. Eclipse Ganymede es la versión 3.4 de este IDE presentada en el año 2008. (14)

### 1.5.6. Framework.

Es un esquema para el desarrollo de un sistema integrando componentes personalizables e intercambiables. 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 el código ya existente y promover buenas prácticas de desarrollo como el uso de patrones.

Diseñados con el intento 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. Son soluciones completas que contemplan herramientas de apoyo a la construcción (ambiente de trabajo o desarrollo) y motores de ejecución (ambiente de ejecución).

### 1.5.6.1. Spring

Es un framework de código abierto para el desarrollo de aplicaciones en la plataforma Java. Ofrece mucha libertad a los desarrolladores, gracias a su diseño en módulos. Facilita el desarrollo de funcionalidades específicas en diferentes paquetes entre los que pueden mencionarse DAO, JDBC, ORM y Web MVC.

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.

El paquete JDBC provee una manera de administrar transacciones tanto declarativas como programáticas, no solo para clases que implementen interfaces especiales.

El paquete ORM provee capas de integración para APIs de mapeo objeto - relacional, incluyendo, JDO, Hibernate e iBatis. Usando el paquete ORM se pueden usar esos mapeadores en conjunto con otras características que Spring ofrece, como la administración de transacciones mencionada con anterioridad.

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 la validación.

El diseño de Spring está pensado para ofrecer un modelo de cómo debe trabajar la aplicación y cómo se comunican sus partes. Está expresamente concebido para que deba ser extensible y acoplable con otros frameworks, ya que no ofrece una solución completa que abarque desde la presentación al modelo. (15)

### 1.5.6.2. Hibernate

Es una herramienta de Mapeo objeto-relacional para la plataforma Java que facilita el mapeo de atributos entre una base de datos relacional tradicional y el modelo de objetos de una aplicación, mediante archivos declarativos (XML) que permiten establecer estas relaciones. Es software libre, distribuido bajo los términos de la licencia GNU LGPL.

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.

Hibernate convertirá los datos entre los tipos utilizados por Java y los definidos por SQL, libera al desarrollador del manejo manual de los datos que resultan de la ejecución de dichas sentencias, manteniendo la portabilidad entre todos los motores de bases de datos. Está diseñado para ser flexible en cuanto al esquema de tablas utilizado, para poder adaptarse a su uso sobre una base de datos ya existente. (16)

### 1.5.6.3. JUnit

JUnit es un framework de Open Source para la automatización de prueba tanto unitarias como de integración. Permite probar la ejecución de clases Java de manera controlada para de esta forma comprobar si el funcionamiento de cada uno de los métodos de la clase se comporta de la forma esperada.

Según un valor de entrada, se evalúa el retorno de la prueba; si cumple con los valores esperados, entonces JUnit devolverá que el método de la clase pasó la prueba, en caso contrario JUnit devolverá un fallo en el método correspondiente.

JUnit es también un medio de controlar las pruebas de regresión, que se utilizan cuando una parte del código ha sido modificado y se desea saber si el nuevo código cumple con los requerimientos y no se ha alterado su funcionalidad con el cambio realizado, incluye además formas de ver los resultados, puede ser en modo texto, gráfico o como tarea.

En la actualidad el Eclipse cuenta con plug-ins que permiten generar las plantillas necesarias para la creación de las pruebas de una clase de forma automática. (17)

### 1.5.7. Especificaciones

#### 1.5.7.1. Java Persistence API (JPA)

Proporciona un estándar para gestionar datos relacionales en aplicaciones Java simplificando el trabajo con la persistencia de datos. Se basa en las ideas y conceptos de tecnologías de persistencia como Hibernate, Toplink y JDO. 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) se realiza mediante anotaciones de Java o descriptores XML que definen el mapeo entre objetos Java y la base de datos relacional. (18) JPA con su mecanismo de persistencia se encarga de crear las tablas, si no existen y de usarlas en caso contrario. Además da igual la base de datos, se puede por ejemplo escribir la aplicación en MySQL y después con un ligero cambio llevarla a Oracle.

### 1.5.8. Web Service.

Los servicios web son sistemas de software que permiten el intercambio de datos y funcionalidad entre aplicaciones sobre una red. Está soportado en diferentes estándares que garantizan la interoperabilidad de los servicios. Aportan interoperabilidad entre aplicaciones de software independientemente de sus propiedades o de las plataformas sobre las que se instalen. Los servicios Web fomentan los estándares y protocolos basados en texto, que hacen más fácil acceder a su contenido y entender su funcionamiento. Al apoyarse en HTTP, los servicios Web pueden aprovecharse de los sistemas de seguridad firewall sin necesidad de cambiar las reglas de filtrado.

Permiten que servicios y software de diferentes compañías ubicadas en diferentes lugares geográficos puedan ser combinados fácilmente para proveer servicios integrados. (19)

### 1.5.9. PostgreSQL servidor de base de datos.

Es un gestor de base de datos relacional orientada a objetos de software libre, publicado bajo la licencia BSD. Funciona en todos los sistemas operativos importantes, incluyendo Linux, UNIX y Windows. Incluye la mayoría de los tipos de datos como son INTEGER, NUMERIC, BOOLEAN, CHAR, VARCHAR, DATE y INTERVAL.

Es ampliamente considerado como una de las alternativas de sistemas de bases de datos de código abierto. El código fuente está disponible para todos sin costo, permitiendo personalizarlo o extenderlo.

PostgreSQL usa una estrategia de almacenamiento de filas llamada MVCC para conseguir una mejor respuesta en ambientes de grandes volúmenes. Ofrece varios métodos de bloqueos para evitar el acceso concurrente a los datos en las tablas. (20)

Incluye características de la orientación a objetos, como puede ser la herencia, tipos de datos, funciones, restricciones, disparadores, reglas e integridad transaccional. (21)

### 1.5.10. Apache Tomcat como servidor web.

Es un servidor web que funciona como un contenedor de servlet. Implementa las especificaciones de los servlet y las Java Server Page (JSP). 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.

Tomcat es mantenido y desarrollado por miembros de la Apache Software Foundation y voluntarios independientes. Los usuarios disponen de libre acceso a su código fuente y a su forma binaria en los términos establecidos en la *Apache Software Licence*.

## Capítulo I: Fundamentación Teórica

Puede ser ejecutado en cualquier sistema operativo que cuente con una JVM, debido a que fue escrito utilizando esta plataforma; por lo que puede ser considerado multiplataforma. (22)

### 1.6. Conclusiones.

Este capítulo recogió los conceptos fundamentales relacionados con la Gestión de Contenidos para Dispositivos Móviles. Se especifican las condiciones actuales del objeto de estudio y de los procesos existentes. Incluye además un estudio de algunos sistemas existentes para la gestión de contenidos para dispositivos móviles, aunque ninguno de ellos implementa el 100% de las funcionalidades requeridas por la VPSM, además de ser software propietario por lo que se evidencia la necesidad de desarrollar una solución propia. Por último se describen brevemente las herramientas y tecnologías que serán utilizadas en la creación del sistema propuesto.

## 2.1. Introducción

Este capítulo permitirá conocer el contexto en el que se emplaza la solución propuesta. Se describirán los procesos existentes u observados con el objetivo de comprenderlos y se especificarán aquellos que soportará el sistema. Además guiará el desarrollo del software hacia el sistema adecuado, mediante la captura y descripción de las condiciones o capacidades que el sistema debe cumplir y de las propiedades del mismo.

## 2.2. Propuesta del sistema

La solución que se propone se limita a la Plataforma GINA que estará diseñada para clientes móviles de la VPSM de ETECSA. En su versión número 1, el Módulo Contenidos tendrá como propósito fundamental la gestión de los contenidos de valor agregado. Estos se restringirán solo a las categorías imágenes y audio. Dentro de su ámbito también se encuentra el control sobre las tareas de conversión de formato realizadas por el Módulo Transcoder.

En un futuro se agregarán funcionalidades que permitan la gestión de contenidos de valor agregado de otro tipo, como los juegos y videos. El módulo administrará una base de datos relacional y un repositorio de contenidos; en la primera se almacenarán los datos referentes a los contenidos, proveedores, autores y otros objetos del dominio, en la segunda se almacenarán los archivos de los contenidos. Los archivos de los contenidos contienen un arreglo de byte que se obtiene del proceso de serialización de un contenido.

Una vez transformados los contenidos según las especificaciones del cliente móvil que lo solicite, estos se almacenarán de manera caché en el repositorio.

Los objetos contenidos pueden representar una categoría de contenido o un contenido específico, conformando de esta forma un árbol donde los contenidos que representan categorías conforman los diferentes niveles y los que representan un contenido en sí, son las hojas del árbol. De esta forma los contenidos poseerán un contenido padre que representa la categoría inmediata superior a la que pertenecen.

## Capítulo II: Características del Sistema

El acceso a los servicios prestados por el Módulo Contenidos se realiza mediante una interfaz para los elementos internos de la Plataforma de Servicios de GINA y mediante web service para los elementos externos a dicha Plataforma.

### 2.3. Modelo de Dominio

Al ser la gestión de contenidos para dispositivos móviles una actividad que se realiza por primera vez dentro del campo de acción de esta investigación y debido a la poca experiencia de los conocedores del tema, fue necesario modelar el negocio mediante un Modelo de Dominio.

Un Modelo de Dominio describe y recoge los tipos más importantes de objetos o clases dentro del contexto del sistema, mientras un Modelo de Negocio describe además los procesos que se desarrollan, quienes participan en ellos y como interactúan entre si.

Los procesos de negocio en la Plataforma de Gestión de Contenidos no están bien definidos y algunos todavía no tienen claras las actividades que desarrollan ni los involucrados en las mismas. Por ese motivo, la Modelación del Dominio utiliza las clases del dominio, las que se obtienen del conocimiento de especialistas de la VPSM de ETECSA y del estudio de sistemas similares realizado en el capítulo 1.

#### 2.3.1. Diagrama de Clases del dominio

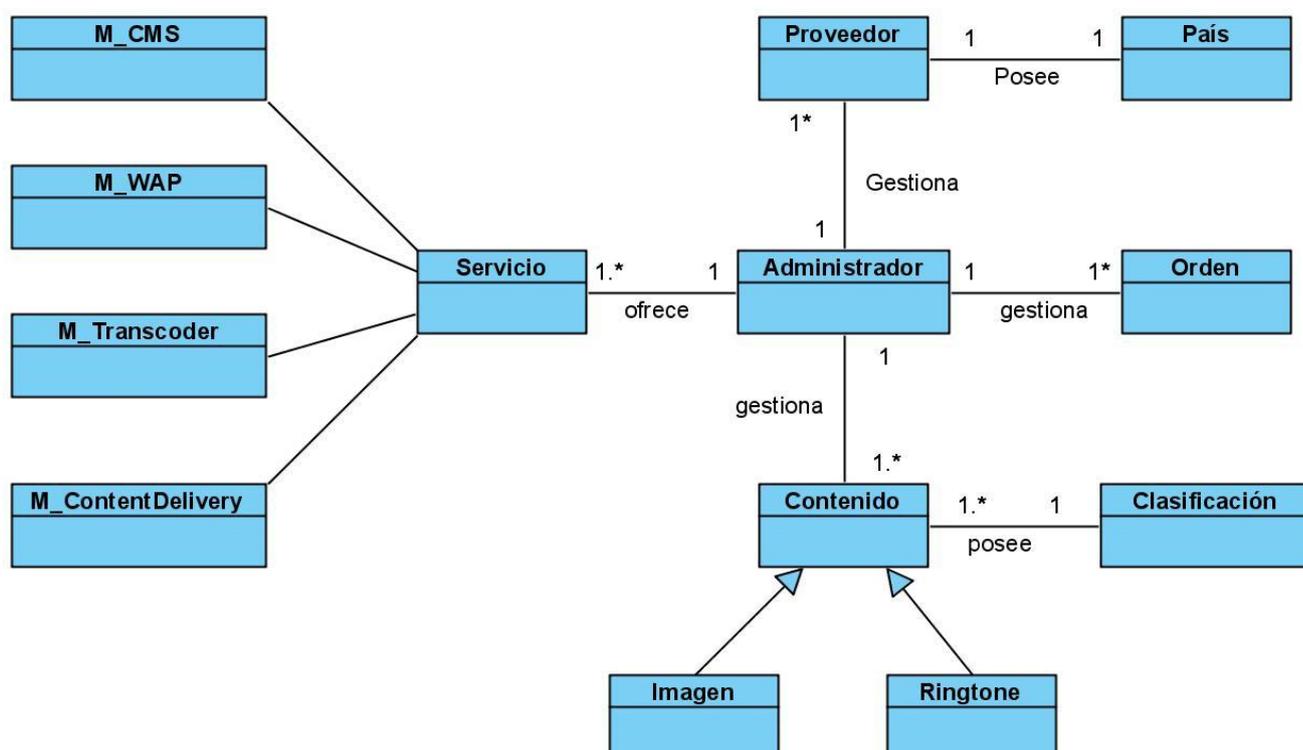


FIGURA 1: Diagrama de clases del dominio

## Capítulo II: Características del Sistema

<b>Administrador:</b>	Controla y dirige las actividades que realizan el resto de los objetos.
<b>Contenido:</b>	Representa los contenido que se gestionan.
<b>Imagen:</b>	Representa los contenidos de tipo imagen.
<b>Ringtone:</b>	Representa los contenidos de tipo ringtone.
<b>Clasificación:</b>	Representa la clasificación de los contenidos según el tema que desarrollan.
<b>Orden:</b>	Representa una petición para adquirir un contenido.
<b>Proveedor:</b>	Representa las entidades o personas proveedores de contenidos.
<b>País:</b>	Representa el país del proveedor de contenidos.
<b>Servicio:</b>	Representa los diferentes servicios que se prestan a los diferentes subsistemas.
<b>M_ContentDelivery:</b>	Representa el subsistema de entrega de contenidos.
<b>M_Transcoder:</b>	Representa el subsistema de modificación de formato del contenido.
<b>M_CMS:</b>	Representa el subsistema encargado de la comunicación entre el administrador y la plataforma.
<b>M_WAP:</b>	Representa el subsistema encargado de la comunicación entre el cliente y la plataforma.

### 2.4. Relación de requerimientos.

#### 2.4.1. Requisitos funcionales.

- RF 1. Adicionar país.  
Permitirá adicionar al sistema los datos de un nuevo país para determinados proveedores.
- RF 2. Modificar país.  
Permitirá modificar los datos de un país en el sistema.
- RF 3. Eliminar país.  
Permitirá eliminar un país del sistema.
- RF 4. Buscar país por ID.  
Permitirá buscar los datos de un país según el identificador del mismo.
- RF 5. Buscar todos los países.  
Permitirá buscar todos los países almacenados en el sistema.
- RF 6. Adicionar autor.  
Permitirá adicionar al sistema los datos de un nuevo autor de determinados contenidos.
- RF 7. Modificar autor.  
Permitirá modificar los datos de un autor.
- RF 8. Eliminar autor.  
Permitirá eliminar los datos de un autor del sistema

## Capítulo II: Características del Sistema

- RF 9. Buscar autor por ID.  
Permitirá buscar los datos de un autor según el identificador del mismo.
- RF 10. Buscar todos los autores.  
Permitirá buscar todos los autores almacenados en el sistema.
- RF 11. Adicionar proveedor.  
Permitirá adicionar al sistema los datos de un nuevo proveedor de determinados contenidos.
- RF 12. Modificar proveedor.  
Permitirá modificar los datos de un proveedor.
- RF 13. Eliminar proveedor.  
Permitirá eliminar los datos de un proveedor.
- RF 14. Buscar proveedor por ID.  
Permitirá buscar los datos de un proveedor según el identificador del mismo.
- RF 15. Buscar todos los proveedores.  
Permitirá buscar todos los proveedores almacenados en el sistema.
- RF 16. Buscar proveedor por países.  
Permitirá buscar todos los proveedores de un país identificado por su identificador.
- RF 17. Adicionar clasificación.  
Permitirá adicionar los datos de una clasificación al sistema para determinados contenidos.
- RF 18. Modificar clasificación.  
Permitirá modificar los datos de una clasificación.
- RF 19. Eliminar clasificación.  
Permitirá eliminar una clasificación del sistema.
- RF 20. Buscar clasificación por ID.  
Permitirá buscar los datos de una clasificación según el identificador del mismo.
- RF 21. Buscar todas las clasificaciones.  
Permitirá buscar todas las clasificaciones almacenadas en el sistema.
- RF 22. Adicionar contenido.  
Permitirá adicionar un contenido al sistema. Los contenidos pueden contener todos los datos del mismo o ser un contenido vacío que representa una categoría de contenido.
- RF 23. Modificar contenido.  
Permitirá modificar los datos de un contenido.
- RF 24. Eliminar contenido.

## Capítulo II: Características del Sistema

- Permitirá eliminar un contenido.
- RF 25. Actualizar número de descargas.  
Permitirá actualizar el número de descargas de un contenido incrementándolo en 1.
- RF 26. Buscar contenido por ID.  
Permitirá buscar los datos de un contenido según el identificador del mismo.
- RF 27. Buscar todos los contenidos por categoría.  
Permitirá buscar todos los contenidos de una categoría determinada.
- RF 28. Buscar contenidos disponibles por proveedor.  
Permitirá buscar los contenidos disponibles por proveedor.
- RF 29. Buscar contenidos por proveedor.  
Permitirá buscar los contenidos por proveedor.
- RF 30. Buscar todos los contenidos disponibles por autor.  
Permitirá buscar los contenidos disponibles por autor.
- RF 31. Buscar todas las categorías disponibles.  
Permitirá buscar todas las categorías disponibles en el sistema.
- RF 32. Buscar todas las categorías.  
Permitirá buscar todas las categorías almacenadas en el sistema.
- RF 33. Buscar todos los contenidos por autor.  
Permitirá buscar todos los contenidos por autor.
- RF 34. Buscar todos los contenidos disponibles por clasificación.  
Permitirá buscar todos los contenidos disponibles por clasificación.
- RF 35. Buscar todos los contenidos por clasificación.  
Permitirá buscar todos los contenidos por clasificación.
- RF 36. Buscar contenido con arreglo de byte.  
Permitirá buscar un contenido con el arreglo de byte del contenido.
- RF 37. Adicionar orden  
Permitirá almacenar una orden de descarga con el identificador del contenido solicitado y el user agente del cliente móvil solicitante.
- RF 38. Modificar orden  
Permitirá modificar los datos de una orden de descarga.
- RF 39. Eliminar orden  
Permitirá eliminar una orden del sistema. La orden eliminada se almacena en un historial.
- RF 40. Buscar orden por ID

## Capítulo II: Características del Sistema

- Permitirá buscar una orden por el identificador de la misma.
- RF 41. Buscar todas las órdenes  
Permitirá buscar todas las órdenes.
- RF 42. Obtener Contenido.  
Permitirá obtener un contenido de la base de datos según el identificador del contenido asociado en la carpeta caché. .
- RF 43. Obtener ID del contenido  
Permitirá obtener el identificador de un contenido en la carpeta caché según el identificador del contenido almacenado en una orden de descarga.  
Si el contenido no posee el formato adecuado según las especificaciones del cliente móvil que solicitó la descarga, permite coordinar la modificación del formato del contenido, almacenarlo en la carpeta caché y obtener el identificador del contenido una vez almacenado.
- RF 44. Obtener archivo de la caché.  
Permitirá obtener el archivo de un contenido en la carpeta caché según el identificador del mismo en dicha carpeta.

### 2.4.2. Requisitos no funcionales

#### Apariencia o Interfaz interna.

- RNF 1. 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 2. Los servicios prestados por el módulo serán publicados en un web service 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 componente de la plataforma podrán acceder a los servicios del Módulo Contenidos mediante objetos.

#### Usabilidad

- RNF 3. Debe ser escalable para poder agregar nuevos servicios sin afectar los que ya se encuentran en funcionamiento.
- RNF 4. Debe permitir la interacción con diferentes proveedores de contenidos independientemente de las plataformas que estos posean.

## Capítulo II: Características del Sistema

- RNF 5. 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.

### Soporte

- RNF 6. La UCI debe brindar soporte tanto a administradores como a clientes que necesiten un breve entrenamiento 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 7. No será capaz de ajustarse automáticamente a cualquier repositorio de contenido que posea un proveedor, pero se suministrarán las interfaces necesarias para desarrollar las implementaciones específicas de cada proveedor; estas implementaciones específicas no forman parte del alcance de este proyecto.

### Seguridad

- RNF 8. El sistema deberá registrar una cantidad finita de eventos normales o anormales del sistema, para poder desarrollar una auditoría.

### Confiabilidad

- RNF 9. El sistema deberá ser capaz de recuperarse ante la ocurrencia de un fallo, de no ser posible, emitir alertas al personal encargado de la administración del mismo, así como proteger la información y contenidos.

### Ayuda y Documentación en línea

- RNF 10. 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.

- RNF 11. Se generará el Javadoc de todas las estructuras implementadas en el sistema

## 2.5. Modelo de Casos de Usos del sistema

### 2.5.1. Diagrama de Casos de Usos

## Capítulo II: Características del Sistema

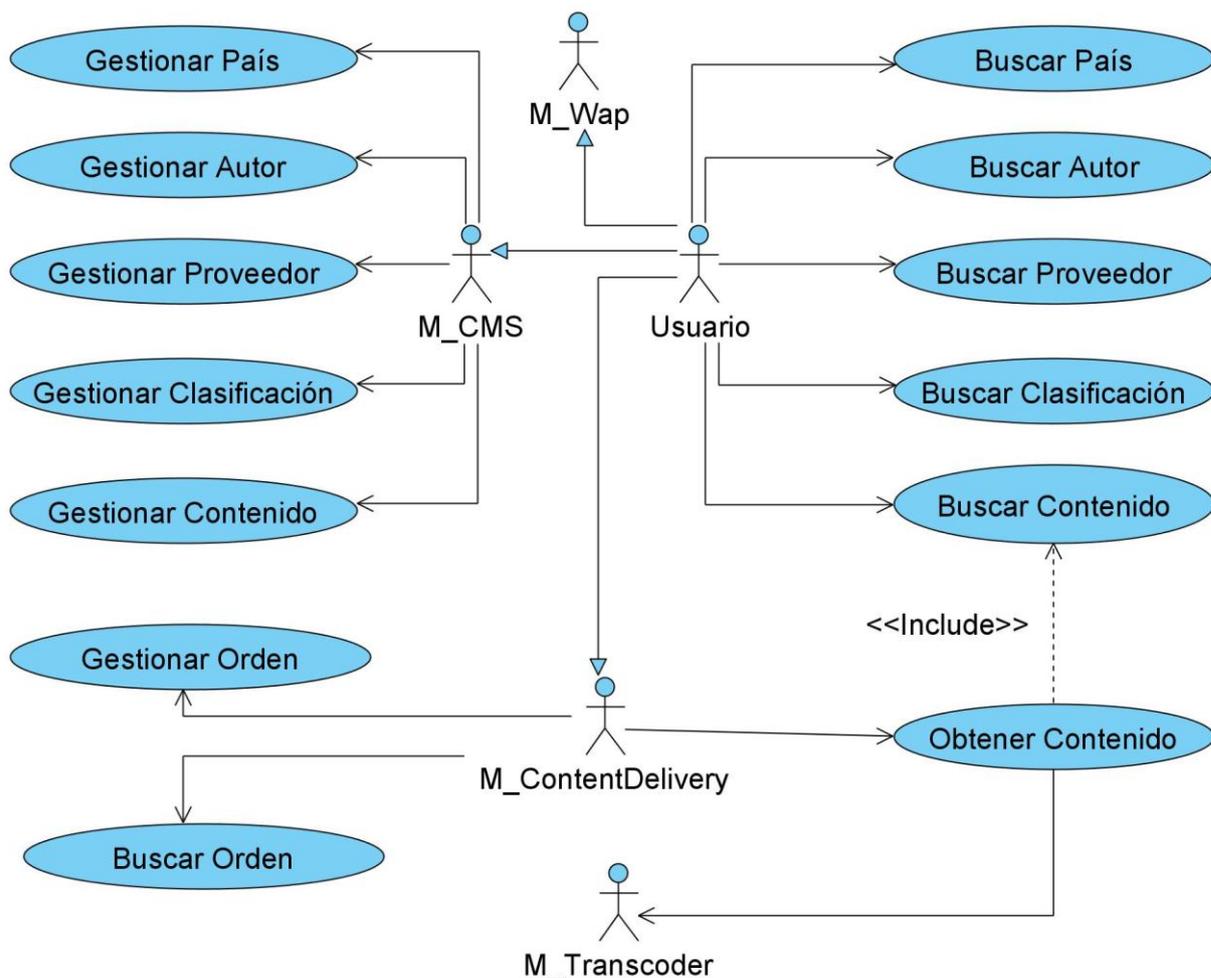


FIGURA 2: Diagrama de Casos de Uso

### 2.5.2. Definición de los actores

Actor	Descripción
<b>Usuario</b>	Representa una generalización de los actores del sistema para las tareas más generales.
<b>M_CMS</b>	Representa el Módulo CMS, encargado de la interacción entre el personal autorizado para administrar la información y la Plataforma.
<b>M_ContentDelivery</b>	Representa el Módulo Descargas, encargado de gestionar la descarga de los contenidos y la aplicación de derechos digitales.
<b>M_Wap</b>	Representa el Módulo Portal Wap, encargado de la interacción entre los clientes y la Plataforma.
<b>M_Transcoder</b>	Representa el Módulo Transcoder, encargado de transformar los contenidos al formato adecuado según las especificaciones de los clientes móviles.

## Capítulo II: Características del Sistema

### 2.5.3. Listado de casos de usos

<b>Caso de Uso:</b>	Gestionar País
<b>Actores:</b>	M_CMS
<b>Resumen:</b>	Permite gestionar los países de los proveedores.
<b>Precondiciones:</b>	-
<b>Referencias:</b>	RF 1; RF 2; RF 3
<b>Prioridad:</b>	Media

<b>Caso de Uso:</b>	Buscar País
<b>Actores:</b>	Usuario
<b>Resumen:</b>	Permite buscar uno o varios países según el parámetro solicitado.
<b>Precondiciones:</b>	-
<b>Referencias:</b>	RF 4; RF 5
<b>Prioridad:</b>	Media

<b>Caso de Uso:</b>	Gestionar Autor
<b>Actores:</b>	M_CMS
<b>Resumen:</b>	Permite gestionar los autores de los contenidos.
<b>Precondiciones:</b>	-
<b>Referencias:</b>	RF 6; RF 7; RF 8
<b>Prioridad:</b>	Media

<b>Caso de Uso:</b>	Buscar Autor
<b>Actores:</b>	Usuario
<b>Resumen:</b>	Permite buscar uno o varios autores según el parámetro solicitado.
<b>Precondiciones:</b>	-
<b>Referencias:</b>	RF 9; RF 10
<b>Prioridad:</b>	Media

<b>Caso de Uso:</b>	Gestionar Proveedor
<b>Actores:</b>	M_CMS
<b>Resumen:</b>	Permite gestionar los proveedores de los contenidos.
<b>Precondiciones:</b>	-

## Capítulo II: Características del Sistema

<b>Referencias:</b>	RF 11; RF 12; RF 13
<b>Prioridad:</b>	Media

<b>Caso de Uso:</b>	Buscar Proveedor
<b>Actores:</b>	Usuario
<b>Resumen:</b>	Permite buscar uno o varios proveedores según el parámetro solicitado.
<b>Precondiciones:</b>	-
<b>Referencias:</b>	RF 14; RF 15; RF 16
<b>Prioridad:</b>	Media

<b>Caso de Uso:</b>	Gestionar Clasificación
<b>Actores:</b>	M_CMS
<b>Resumen:</b>	Permite gestionar las clasificaciones de los contenidos.
<b>Precondiciones:</b>	-
<b>Referencias:</b>	RF 17; RF 18; RF 19
<b>Prioridad:</b>	Media

<b>Caso de Uso:</b>	Buscar Clasificación
<b>Actores:</b>	Usuario
<b>Resumen:</b>	Permite buscar una o varias clasificaciones según el parámetro solicitado.
<b>Precondiciones:</b>	-
<b>Referencias:</b>	RF 20; RF 21
<b>Prioridad:</b>	Media

<b>Caso de Uso:</b>	Gestionar Contenido
<b>Actores:</b>	M_CMS
<b>Resumen:</b>	Permite gestionar contenidos.
<b>Precondiciones:</b>	Los contenidos deben estar en formatos de audio o imagen
<b>Referencias:</b>	RF 22; RF 23; RF 24; RF 25
<b>Prioridad:</b>	Alta

<b>Caso de Uso:</b>	Buscar Contenido
---------------------	------------------

## Capítulo II: Características del Sistema

<b>Actores:</b>	Usuario
<b>Resumen:</b>	Permite buscar uno o varios contenidos según el parámetro solicitado.
<b>Precondiciones:</b>	-
<b>Referencias:</b>	RF26; RF 27; RF 28; RF 29; RF 30; RF 31; RF 32; RF 33; RF 34; RF 35; RF 36
<b>Prioridad:</b>	Alta

<b>Caso de Uso:</b>	Gestionar Orden
<b>Actores:</b>	M_ContentDelivery
<b>Resumen:</b>	Permite gestionar las órdenes de descarga.
<b>Precondiciones:</b>	-
<b>Referencias:</b>	RF 37; RF 38; RF 39
<b>Prioridad:</b>	Media

<b>Caso de Uso:</b>	Buscar Orden
<b>Actores:</b>	M_ContentDelivery
<b>Resumen:</b>	Permite buscar una o varias órdenes según los parámetros solicitados.
<b>Precondiciones:</b>	-
<b>Referencias:</b>	RF 40; RF 41
<b>Prioridad:</b>	Media

<b>Caso de Uso:</b>	Obtener Contenido
<b>Actores:</b>	M_ContentDelivery (inicia), M_Transcoder
<b>Resumen:</b>	Permite obtener un contenido desde la base de datos y el repositorio de contenidos
<b>Precondiciones:</b>	-
<b>Referencias:</b>	RF 42; RF 43; RF 44
<b>Prioridad:</b>	Alta

La descripción detallada de los Casos de Uso se encuentra en el Anexo I: Descripción textual de Casos de Uso

### 2.6. Conclusiones

Este capítulo permitió crear el modelo del sistema que será implementado, utilizando los casos de usos para ese propósito. También permitió 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, Solicitar Contenido y Verificar Formato por contar con funcionalidades críticas para el ciclo de vida del software.

Los casos de usos identificados dirigirán el trabajo a lo largo del resto de los flujos de trabajos propuestos por la metodología de desarrollo de software utilizada, por lo tanto serán los encargados de enlazar directamente los diferentes flujos de trabajo.

### 3.1. Introducción

Analizar los requisitos funcionales descritos en el capítulo anterior, refinarlos y estructurarlos para obtener una comprensión más certera de los mismos es uno de los objetivos de este capítulo.

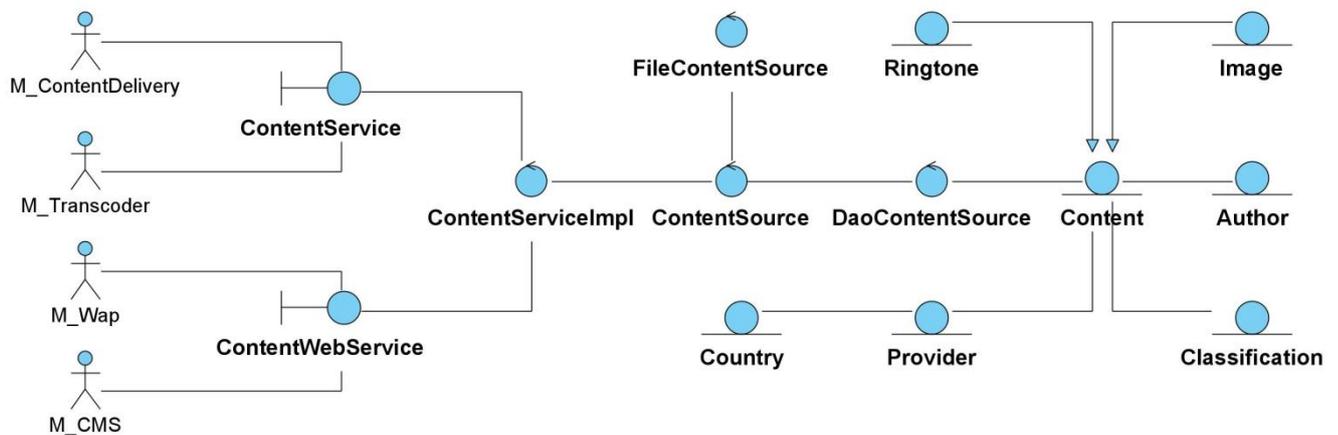
Permitirá además comprender en profundidad aspectos relacionados con los requisitos no funcionales, lenguajes de programación, componentes reutilizables, tecnologías, etc. y examinar brevemente la arquitectura del sistema, la comunicación y el acoplamiento entre subsistemas.

### 3.2. Análisis y Diseño del sistema

#### 3.2.1. Definición del Modelo del Análisis

Es un modelo de objetos conceptual que ayuda a refinar los requisitos y permite razonar sobre los aspectos internos del sistema. Proporciona una estructura centrada en el mantenimiento, en aspectos como la flexibilidad ante los cambios y la reutilización, además se utiliza como entrada en las actividades de diseño e implementación. Generalmente se centra en los requisitos funcionales.

#### 3.2.1. Diagrama de Clases del Análisis



**FIGURA 3: Diagrama de clases del análisis**

#### 3.2.2. Definición del Modelo de Diseño

Permite modelar todo el sistema para que soporte no sólo los requisitos funcionales, sino también los requisitos no funcionales. Contribuye a una arquitectura más estable y sólida. Es un modelo de objetos que describe la realización física de los casos de usos centrándose en cómo los requisitos funcionales

## Capítulo III: Análisis y Diseño del Sistema

y no funcionales, junto con otras restricciones relacionadas con el entorno de implementación, tienen impacto en el sistema a considerar.

### 3.2.3. Diagrama de Clases del Diseño

El Modelo de Diseño del Módulo Contenidos ha sido dividido en 4 subsistemas de diseño. 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, además proporciona interfaces que representan las funcionalidades que exportan en términos de operaciones.

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 Modelo en tres capas.

### Diagrama de Paquetes

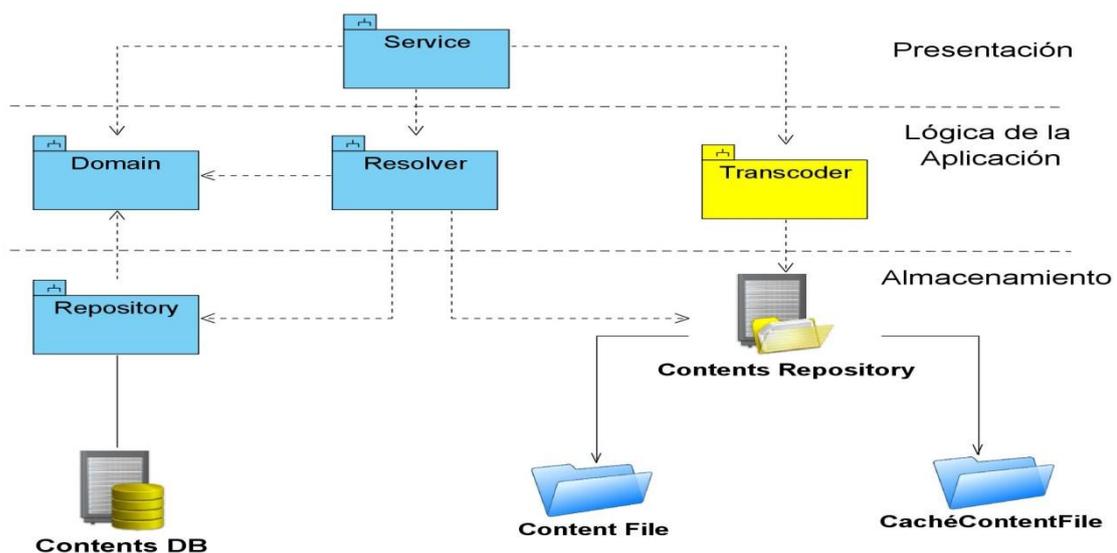


FIGURA 4: Diagrama de paquetes. Vista arquitectónica

### Subsistema Service

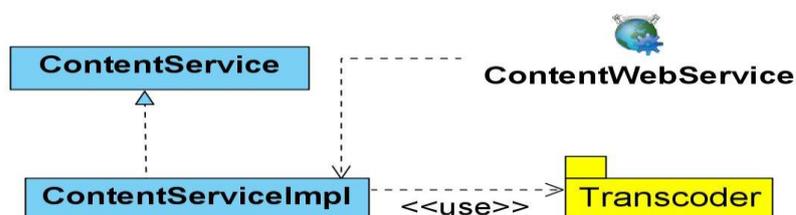


FIGURA 5: Diagrama de clases del diseño. Subsistema Service

## Subsistema Resolver

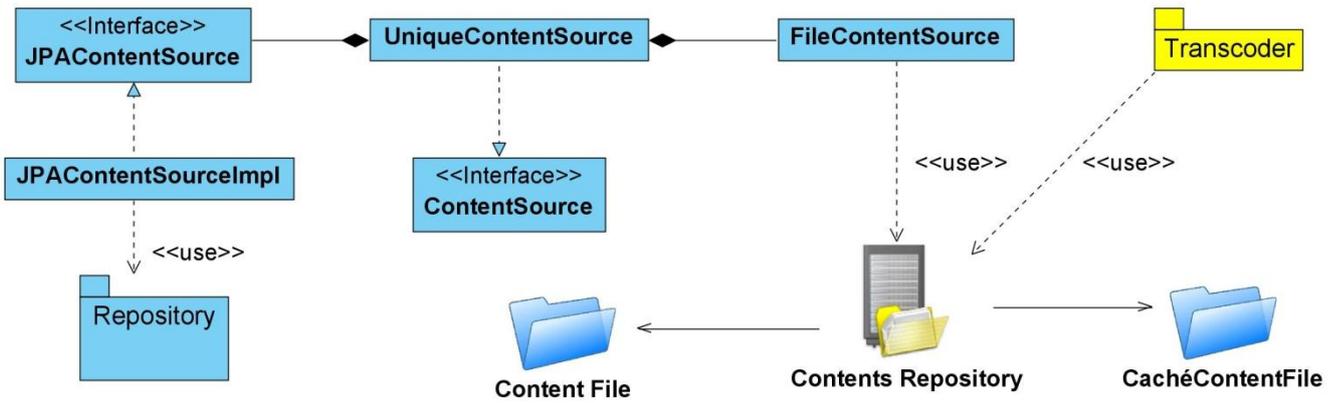


FIGURA 6: Diagrama de clases del diseño. Subsistema Resolver

## Subsistema Repository

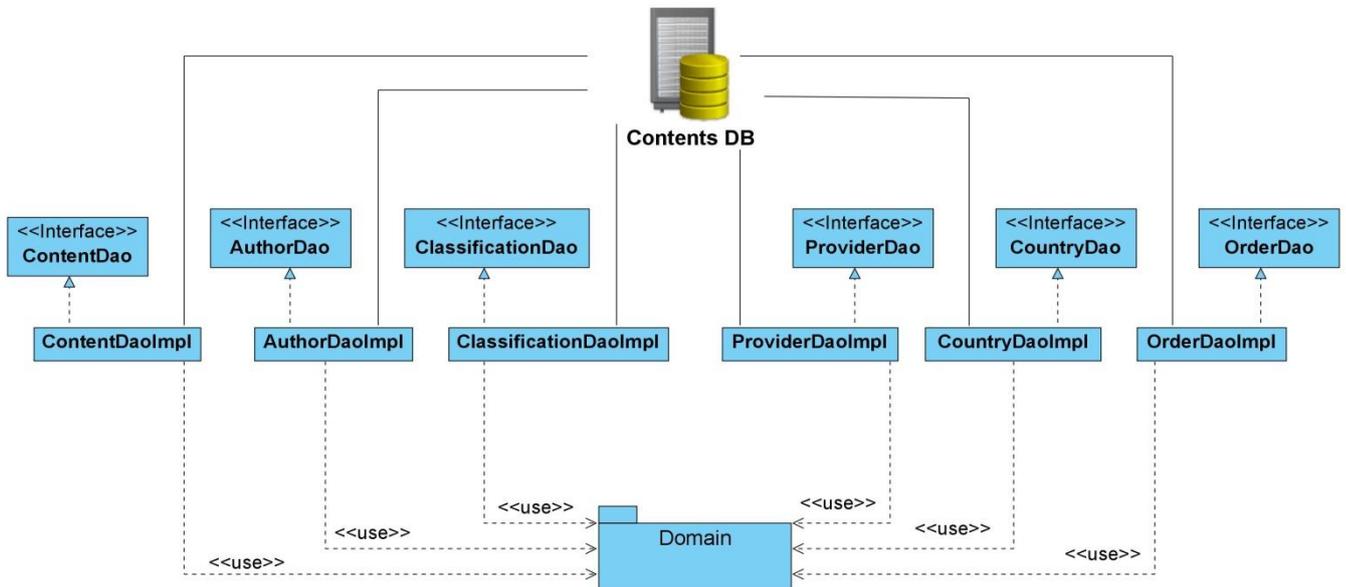


FIGURA 7: Diagrama de clases del diseño. Subsistema Repository

## Subsistema Domain

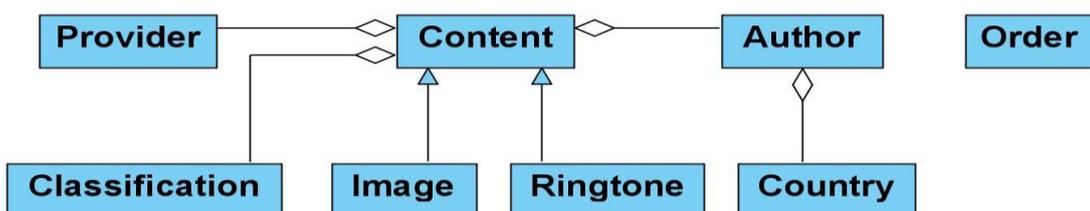


FIGURA 8: Diagrama de clases del diseño. Subsistema Domain

### 3.2.4. Descripción del Diagrama de Clases del Diseño

La descripción detallada de cada clase, sus métodos y atributos se encuentran en el anexo II Descripción de las Clases del Diseño.

### 3.2.5. Diagramas de Interacción

Los diagramas de interacción se utilizan para exponer las especificidades del diseño, estos revelan cómo los objetos se relacionan por medio de mensajes. Entre estos diagramas se encuentra el diagrama de secuencia, el cual muestra claramente la secuencia u ordenación en el tiempo de los mensajes.

Para lograr una comprensión precisa de los eventos que tienen lugar en los casos de usos identificados en el capítulo anterior, se elaboró un diagrama de secuencia para cada escenario de los diferentes casos de usos.

De los casos de usos identificados puede destacarse por su importancia Obtener Contenido. Este caso de usos permite al M\_ContentDelivery obtener un contenido, el identificador del mismo o el archivo donde se encuentra almacenado en la carpeta caché según una petición de descarga de un cliente móvil almacenada en una orden de descarga.

El escenario Obtener ID del Contenido del caso de uso mencionado permite dado el identificador de un contenido y el user agent de un cliente móvil almacenados en una orden, obtener el id del contenido en la carpeta caché. Posteriormente el M\_ContentDelivery envía al cliente móvil una url construida con el identificador del contenido obtenido. Cuando el cliente accede a esta url el M\_ContentDelivery obtiene el contenido con el identificador del mismo.

Si el contenido no se encuentra en la carpeta caché se solicita el Transcoder del mismo según el user agent del cliente móvil y se almacena el contenido en la carpeta caché, luego se devuelve el identificador del contenido en la carpeta caché.

El resto de los diagramas se encuentran en el anexo III Diagramas de Secuencias del Módulo Contenidos.

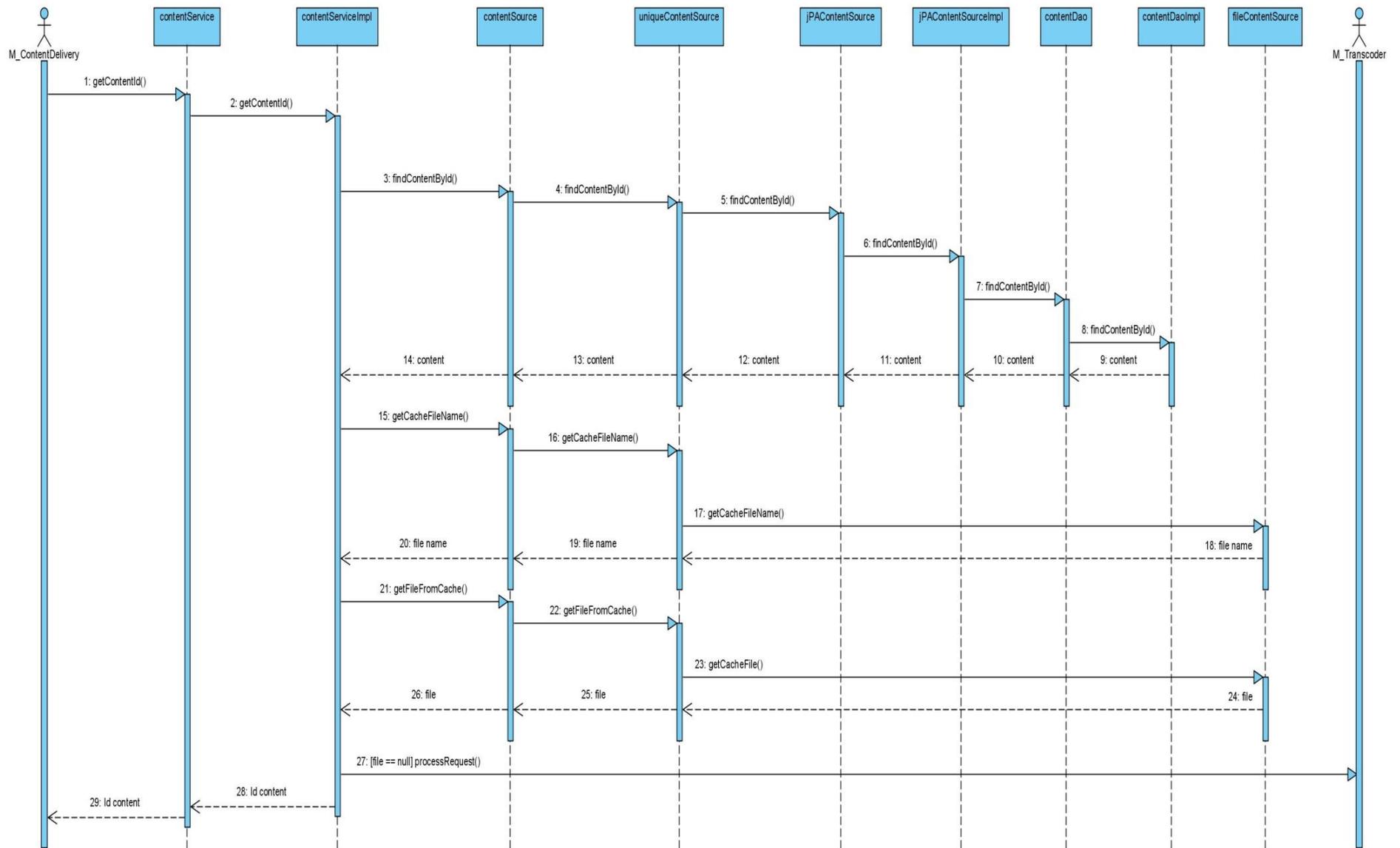


Figura 9: Diagrama de Secuencia. Escenario Obtener ID del Contenido

### 3.3. Descripción de la Arquitectura

#### 3.3.1. Patrones utilizados

Un patrón define una solución a un problema en un contexto determinado e indica cómo puede ser utilizado en diferentes circunstancias. Pueden organizarse por categorías según la escala o nivel de abstracción entre las que pueden mencionarse:

**Patrones arquitecturales:** 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

La **Arquitectura en tres capas** es un patrón arquitectónico que consiste en aislar la lógica de la aplicación y convertirla en una capa intermedia bien definida. (9)

Se definen por tanto 3 capas dentro de la aplicación. En la capa de presentación se realiza poco procesamiento de la aplicación. Sus elementos envían a la capa intermedia peticiones de trabajo y este se comunica con la capa de almacenamiento para atender estas peticiones.

Esta arquitectura se adecua a los sistemas de información orientados a objetos e incluye la división de las responsabilidades asignadas a los objetos de software.

En el Módulo Contenidos se ubica en la capa de presentación el subsistema de diseño “**Service**”, encargado de gestionar el acceso de los usuarios externos e internos a los servicios del módulo.

En la capa de lógica de la aplicación se ubican dos subsistemas, el “**Resolver**” y el “**Domain**”. La división de las responsabilidades de esta capa en dos subsistemas permite hacer la misma menos densa. En el subsistema “**Resolver**” se integran las clases que gestionan los servicios prestados y en el “**Domain**” las clases que representan los conceptos del dominio.

Por último, en la capa de almacenamiento el subsistema “**Repository**” con las clases que permitirán el acceso a la base de datos y el repositorio de contenidos.

#### 3.3.3. Patrones de diseño. Patrones GRASP

Los patrones GRASP son patrones de diseño que describen los principios generales para asignar responsabilidades a objetos, expresados en forma de patrón.

En el diseño del Módulo Contenidos se utilizaron los siguientes patrones:

## Capítulo III: Análisis y Diseño del Sistema

- a) Patrón Experto: Consiste en asignar una responsabilidad a la clase que cuenta con toda la información para cumplirla. Permite conservar el encapsulamiento y soporta un bajo acoplamiento logrando sistemas más robustos y de fácil mantenimiento. Brinda además un soporte a la alta cohesión definiendo clases más sencillas y más cohesivas.
- b) Patrón Creador: Consiste en asignar la responsabilidad de la creación de la instancia de una clase a aquella que agrega o contiene objetos de la misma, registra instancias de ella, utiliza específicamente sus objetos o tiene los datos de inicialización que le serán transmitidos cuando sea creada su instancia. Brinda soporte a un bajo acoplamiento suponiendo menos dependencias respecto al mantenimiento y mejores oportunidades de reutilización.
- c) Patrón 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. Es un principio que debe estar presente durante el diseño, es un patrón evaluativo que se aplica al juzgar decisiones de diseño.
- d) Patrón Bajo Acoplamiento: El acoplamiento es una medida de la fuerza con que una clase está conectada a otra. Clases con fuerte acoplamiento son más difíciles de entender cuando están aisladas, son más difíciles de reutilizar porque se requiere la presencia de otras clases de las que dependen. El Bajo Acoplamiento al igual que la Alta Cohesión es un principio, un patrón evaluativo.

### 3.4. Modelo de Datos

#### 3.4.1. Definición del 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.

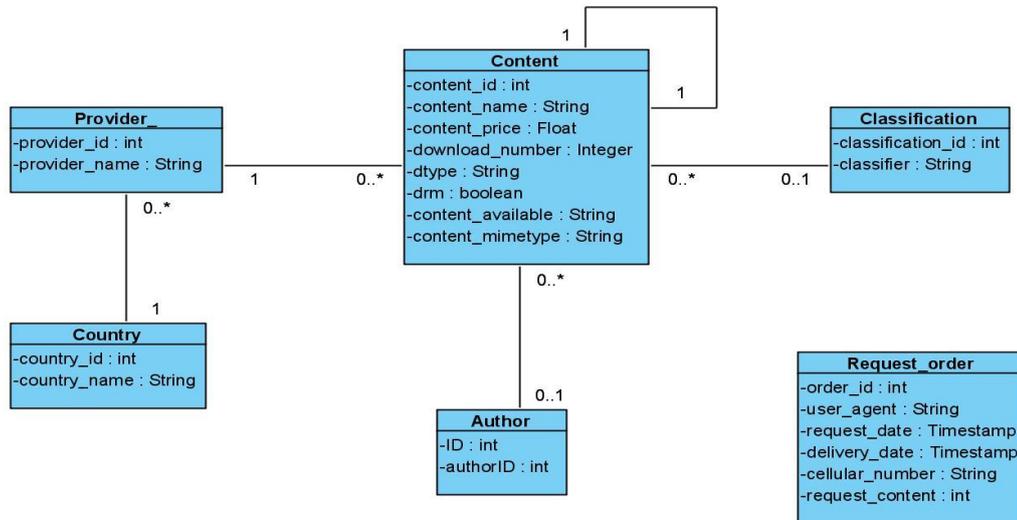


FIGURA 10: Diagrama de clases persistentes

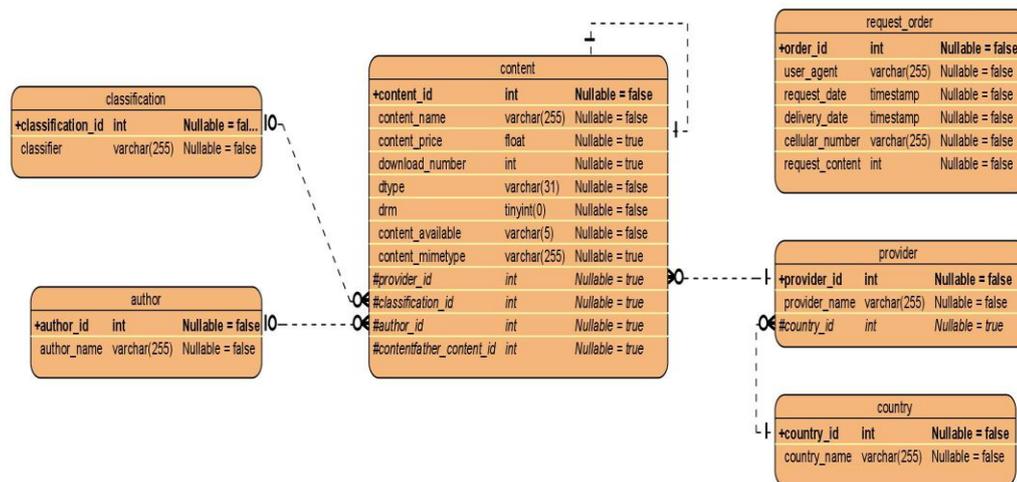


FIGURA 11: Diagrama Entidad-Relación

### 3.5. Conclusiones

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. Una entrada esencial en el diseño es el resultado del análisis. El modelo de análisis proporciona una comprensión detallada de los requisitos e impone una estructura del sistema que debe conservarse lo más fielmente posible cuando se da forma al sistema.

#### 4.1. Introducción

El objetivo primario de este capítulo es describir cómo los elementos identificados en el Modelo de Diseño, tales como Clases del Diseño y Subsistemas del Diseño se implementan en términos de componentes.

También se describirá cómo se relacionan y organizan los componentes de acuerdo con los mecanismos de estructuración y modularización en el entorno de implementación y en el lenguaje de programación utilizado. Se indicará la situación física de los componentes lógicos desarrollados en el Diagrama de Despliegue.

Por último se explicará la planificación de las pruebas realizadas a la implementación; creando casos de pruebas que especifican qué probar y los procedimientos que indican cómo realizar las pruebas, además de un resumen con los resultados obtenidos en las mismas.

#### 4.2. Modelo de Implementación

##### 4.2.1. Diagrama de Componentes

Un componente es el empaquetamiento físico de los elementos de un modelo (23). Dentro de los estereotipos estándares pueden mencionarse <<file>>: un fichero que contiene código fuente o datos y <<library>>: es una librería estática o dinámica.

Se definen 4 subsistemas de implementación con el objetivo de proporcionar una forma de organizar los artefactos del modelo de implementación en trozos más manejables. Para cada uno de estos subsistemas se desarrolló un Diagrama de Componentes.

## Subsistema Service

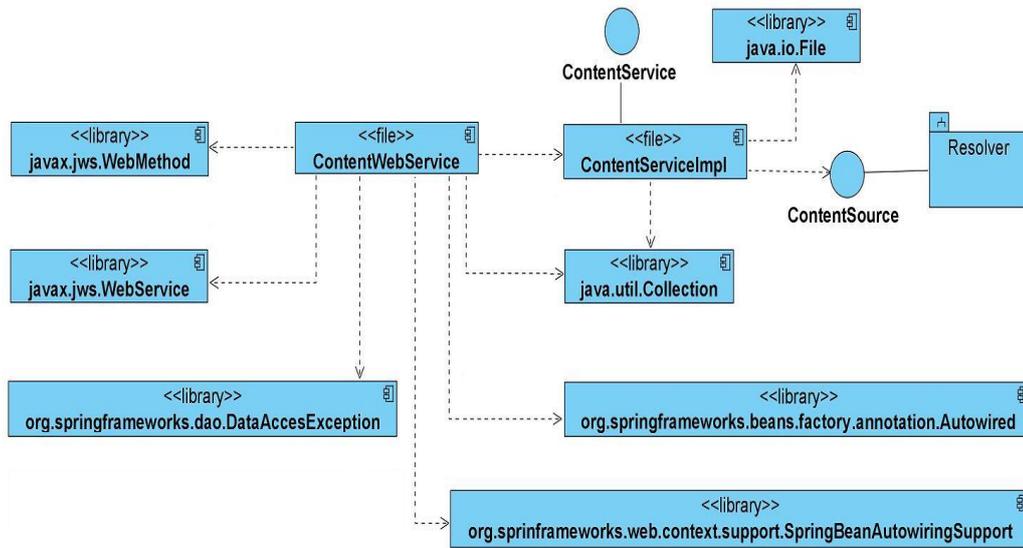


FIGURA 12: Diagrama de componentes. Subsistema Service

## Subsistema Resolver

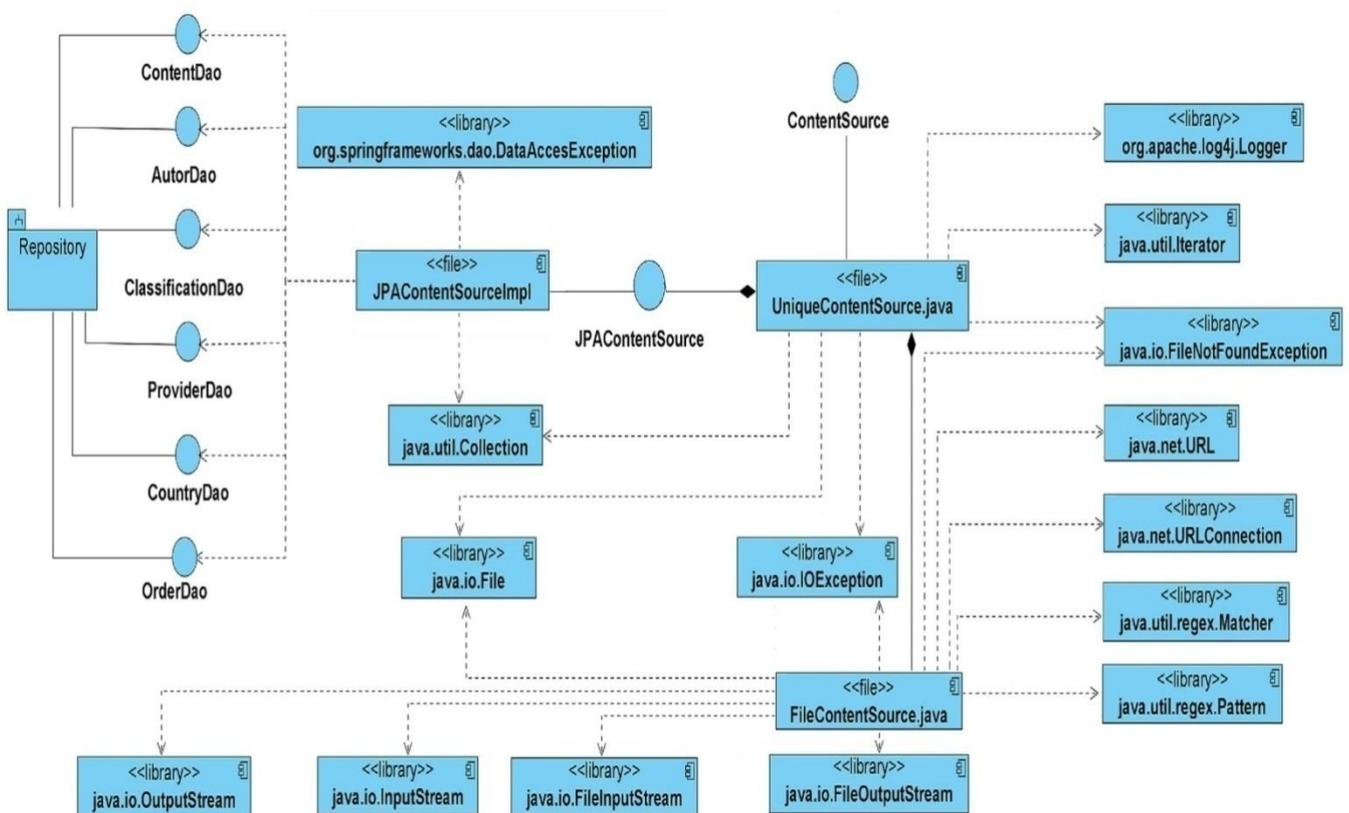


FIGURA 13: Diagrama de componentes. Subsistema Resolver

## Subsistema Repository

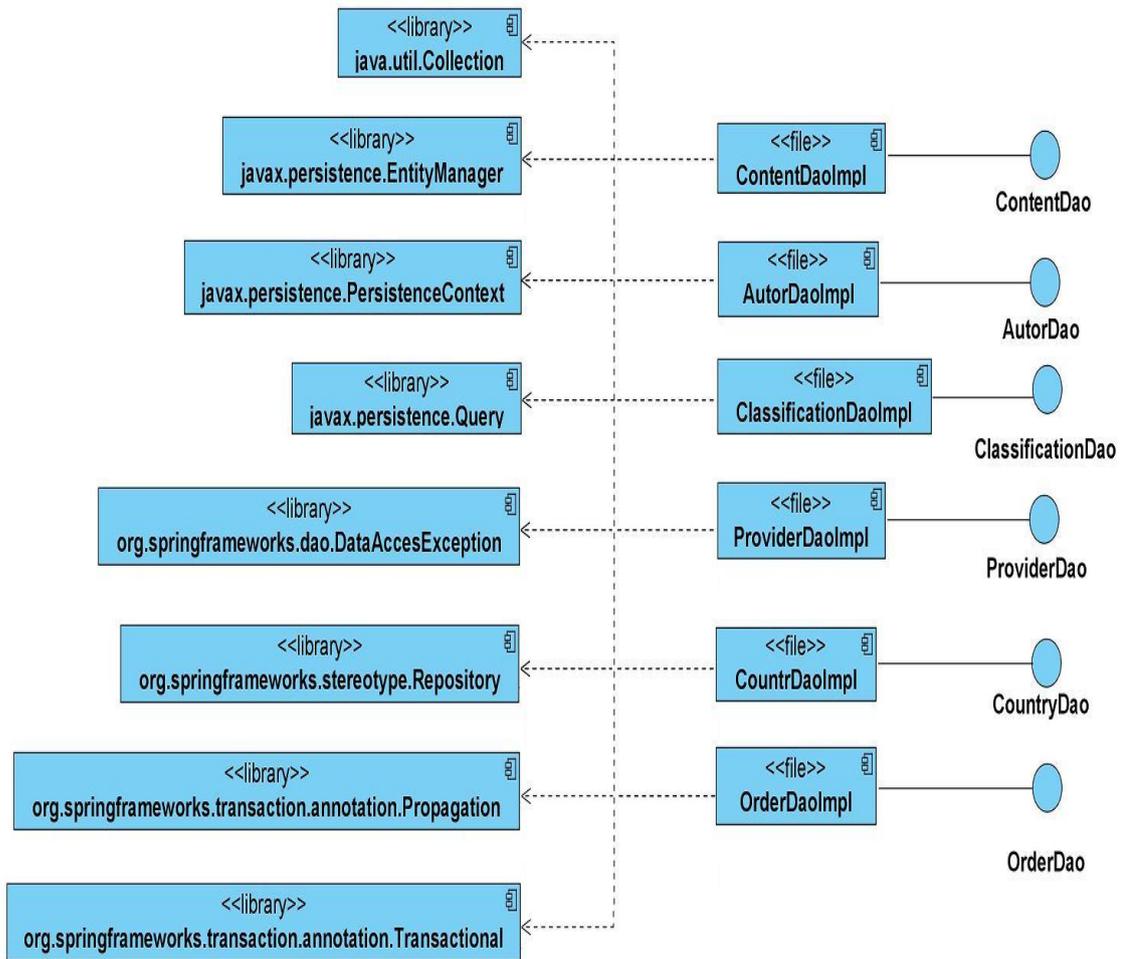


FIGURA 14: Diagrama de componentes. Subsistema Repository

## Subsistema Domain

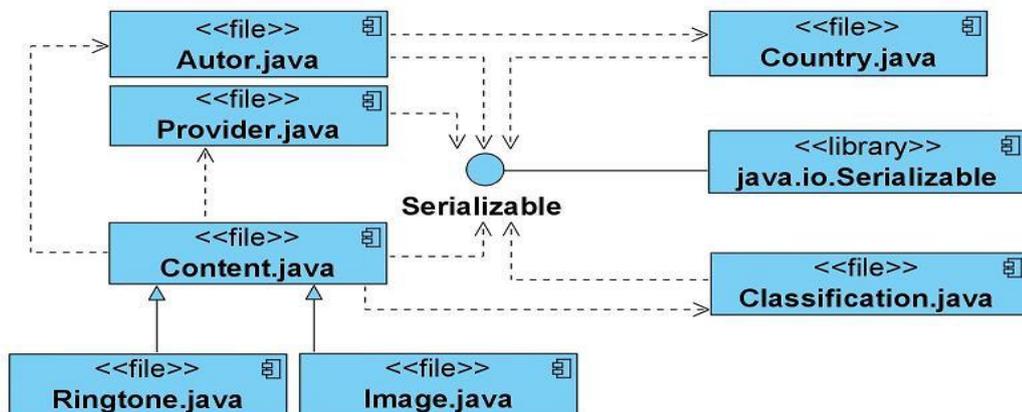


FIGURA 15: Diagrama de componentes. Subsistema Domain

### 4.2.2. Diagrama de Despliegue

Es un modelo de objetos que describe la distribución física del sistema. Muestra como se distribuyen las funcionalidades entre los nodos. Cada nodo representa un recurso de cómputo, normalmente un procesador o un dispositivo hardware similar.

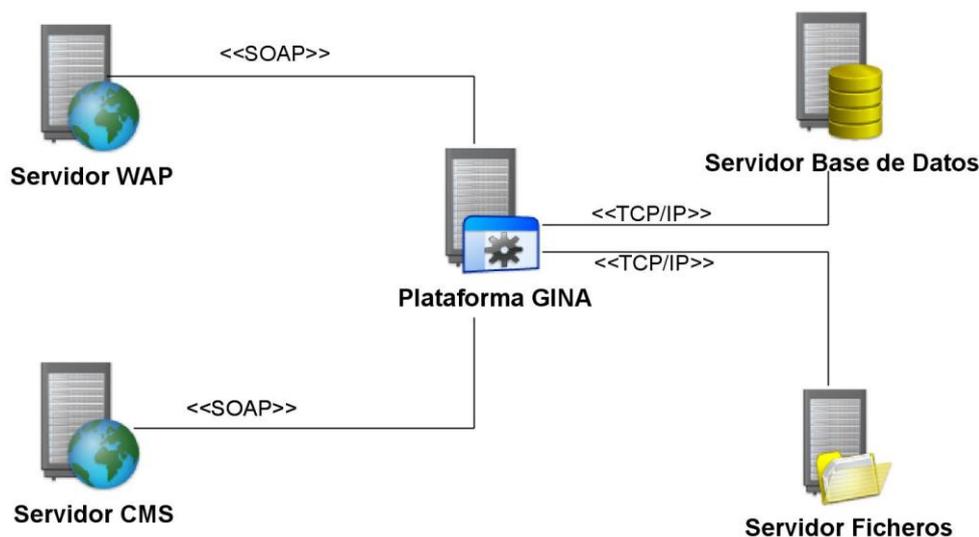


FIGURA 16: Diagrama de despliegue

#### Descripción de los elementos principales:

PC WAP: En este servidor se encuentra el Portal WAP, por el cual navegan los usuarios finales para obtener noticias y llegar a la plataforma de gestión de contenido.

PC CMS: En este servidor se encuentra la interfaz de administración por la cual los proveedores y el personal autorizado de ETECSA pueden interactuar con los diferentes parámetros de configuración y con los contenidos de la plataforma y el portal WAP.

PC Plataforma de Servicios: En este servidor se ubica la Plataforma de Servicios de GINA.

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

TCP/IP: es un conjunto de protocolos de red en la que se basa Internet y que permiten la transmisión de datos entre redes de computadoras.

### 4.3. Modelo de Prueba

En la Plataforma GINA, se planificaron y realizaron dos tipos de pruebas, de unidad y de integración. Realizar pruebas al código permite observar con que frecuencia podría fallar el mismo, encontrar errores y solucionarlos.

Las **pruebas de unidad** permiten probar la correcta funcionalidad del módulo como si actuara independientemente de los demás, mientras las **pruebas de integración** permiten probar, como su nombre lo indica, la correcta integración del módulo ya probado con pruebas unitarias, con el resto de los módulos.

A continuación se muestra un Caso de Prueba correspondiente al Caso de Uso Obtener Contenido, Sección Obtener ID del contenido.

#### CP: Obtener ID contenido

##### Descripción de la funcionalidad:

Permite obtener el identificador de un contenido en la carpeta caché dado una orden de descarga. Esta orden contiene el id del contenido solicitado y el user agent del cliente que los solicita. Si en la carpeta caché se encuentra el contenido con el formato adecuado para el cliente móvil devuelve el id del contenido en dicha carpeta. Si no está solicita el transcoder del contenido, luego devuelve el id del contenido en la carpeta caché ya transformado y almacenado en dicha carpeta.

##### Flujo de Eventos:

Sección Obtener ID del contenido	
Acción del Actor	Respuesta del Sistema
1. Solicita el ID de un contenido en la carpeta caché	2. Busca el contenido según el id del mismo.
	3. Busca el nombre del archivo asociado en la carpeta caché.
	4. Obtiene el archivo en la carpeta caché.
	5. Devuelve el ID del contenido en la carpeta caché.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	3.1 Muestra un mensaje de error si la operación no puede realizarse.
	4.1 Muestra un mensaje de error si la operación no puede realizarse.
	5.1. Si el archivo no existe en la carpeta caché, solicita la transformación del contenido al M_Transcoder.
	5.2. Devuelve el id del contenido en la carpeta

## Capítulo IV: Implementación y Prueba

	caché.
	5.3. Muestra un mensaje de error si la operación no puede realizarse.

### Iteraciones:

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
CV 1		Devuelve el id del contenido en la carpeta caché.	Devuelve el id del contenido en la carpeta caché.	Asegurar que el contenido está transformado en la carpeta caché.
CV 2		Solicita el transcoder del contenido y devuelve el id del mismo en la carpeta caché luego de transformado y almacenado.	Solicita el transcoder del contenido y devuelve el id del mismo en la carpeta caché luego de transformado y almacenado.	Comprobar que el contenido no está en la carpeta caché antes de comenzar la prueba y que se almacena luego.
	CNV 3	Se muestra un mensaje de error indicando que el contenido no existe.	Se muestra un mensaje de error indicando que el contenido no existe.	Comprobar que el contenido no existe.

CV 1: El contenido se encuentra en la carpeta caché.

CV 2: El contenido no se encuentra en la carpeta caché. Se solicita Transcoder.

CNV 3: El contenido solicitado no existe.

#### 4.4. Conclusiones

El propósito fundamental de la implementación es desarrollar la arquitectura del sistema capturada durante el diseño. En esta fase no solo se distribuyeron los componentes y se asignaron a nodos en el diagrama de despliegue, también se implementaron las clases y subsistemas encontrados durante el diseño.

También se probaron los componentes individualmente y fueron integrados con el resto de los componentes de la Plataforma de Gestión de Contenidos para Dispositivos Móviles de la VPSM de ETECSA.

### 5.1. Introducción

La estimación de recursos, costes y planificación temporal de un esfuerzo en el desarrollo de software requiere confiar en predicciones cuantitativas cuando todo lo que existe son datos cualitativos. Estas estimaciones se hacen dentro de un límite de tiempo al comienzo de una iteración del proyecto de software y debe actualizarse a medida que se pasa de una iteración a otra.

La estimación del proyecto nunca será una ciencia exacta, pero la combinación de buenos datos históricos y de técnicas sistemáticas puede mejorar la precisión de la estimación.

### 5.2. Método de estimación "Puntos de Casos de Uso"

#### 5.2.1. Definición y conceptos fundamentales

Existe una posibilidad de predecir el tamaño de un sistema a partir de las características de sus requisitos, expresados en los casos de uso.

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

El método utiliza los actores y casos de uso identificados para calcular el esfuerzo que costará desarrollarlos. A los casos de uso se les asigna una complejidad basada en transacciones, de los escenarios de los casos de uso. A los actores se les asigna una complejidad basada en el tipo de actor, es decir, si son interfaces con usuarios o si son interfaces con otros sistemas. También se utilizan factores de entorno y de complejidad técnica para afinar el resultado.

Una vez asignada complejidad a actores y casos de uso y establecidos los factores técnicos y de entorno, se calculan los puntos de caso de uso no ajustados o UUCP, el TCF (factor de complejidad técnica) y el EF (factor del entorno). Con ellos, se calculan los puntos de caso de uso o UCP, que finalmente se traducen a esfuerzo en horas-hombre con un sencillo cálculo.

### 5.2.2. Cálculo de Puntos de Casos de Uso

Los Puntos de Casos de Uso sin ajustar (**UUCP**) se calcula sumando el Factor de Peso de los Actores sin ajustar (**UAW**) y el Factor de Peso de los Casos de Uso sin ajustar (**UUCW**).

$$\mathbf{UUCP = UAW + UUCW [1]}$$

El Factor de Peso de los Actores sin ajustar (**UAW**) se calcula analizando la cantidad de actores presentes en el sistema y la complejidad de cada uno de ellos. Los criterios para evaluar la complejidad pueden ser los siguientes.

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

Puede definirse

$$\mathbf{UAW = \sum(\text{actores} * \text{peso}) [2]}$$

Donde actores representa la cantidad de actores de un tipo presente en el sistema y peso el Factor de Peso de ese Tipo de Actor.

En el Módulo Contenidos se determinaron 4 actores simples, quedando **UAW** como:

$$\mathbf{UAW = (4 * 1)}$$

$$\mathbf{UAW = 4}$$

El Factor de Peso de los Casos de Uso sin ajustar (**UUCW**) se calcula analizando la cantidad de Casos de Uso presentes en el sistema y la complejidad de cada uno de ellos. La complejidad de los Casos de Uso se establece teniendo en cuenta la cantidad de transacciones efectuadas en el mismo, donde una **transacción** se entiende como una secuencia de actividades atómica. Los criterios se muestran en la siguiente tabla:

Tipo de Caso de Uso	Descripción	Factor de Peso
Simple	El Caso de Uso contiene de 1 a 3 Transacciones.	5
Medio	El Caso de Uso contiene de 4 a 7 Transacciones.	10
Complejo	El Caso de Uso contiene más de 8 Transacciones.	15

Puede definirse

## Capítulo V: Estudio de Factibilidad

$$\text{UUCW} = \sum (\text{CU} * \text{peso}) \text{ [3]}$$

Donde CU representa la cantidad de Casos de Usos de un tipo presentes en el sistema y peso el Factor de Peso correspondiente a ese Tipo de Caso de Uso.

En el Módulo Contenidos se determinaron 13 Casos de Uso; 5 simple y 8 medio, quedando **UUCW** como:

$$\text{UUCW} = (5*5) + (8*10)$$

$$\text{UUCW} = 25 + 80$$

$$\text{UUCW} = 105$$

Sustituyendo las ecuaciones 2 y 3 en la ecuación 1:

$$\text{UUCP} = \text{UAW} + \text{UUCW}$$

$$\text{UUCP} = 4 + 105$$

$$\text{UUCP} = 109$$

Luego de calcular los Puntos de Casos de Uso sin ajustar (**UUCP**), se hace necesario ajustar este valor multiplicándolo por el Factor de complejidad técnica (**TCF**) y el Factor de Ambiente (**EF**), obteniendo como resultado los Puntos de Casos de Uso ajustados (**UCP**)

$$\text{UCP} = \text{UUCP} \times \text{TCF} \times \text{EF} \text{ [4]}$$

El Factor de complejidad técnica (TCF) es un coeficiente que se calcula cuantificando un conjunto de factores que determinan la complejidad técnica del sistema. Cada uno de estos factores se cuantifica con un valor de 0 a 5, donde 0 significa irrelevante y 5 un aporte muy importante.

La siguiente tabla muestra los factores cuantificados en el Módulo Contenidos y el valor asignado a cada uno.

Factor	Descripción	Peso	Comentarios	Valor Asignado
T1	Sistema distribuido	2	El sistema es centralizado	0
T2	Objetivos de performance o tiempo de respuesta	1	No se necesitan altas velocidades de respuesta.	1
T3	Eficiencia del usuario final	1	Escasas restricciones de eficiencia	1
T4	Procesamiento interno complejo	1	No hay cálculos complejos	0
T5	El código debe ser reutilizable	1	No se necesita que el código sea reutilizable	0
T6	Facilidad de instalación	0.5	Escasos requerimientos de instalación	1

## Capítulo V: Estudio de Factibilidad

T7	Facilidad de uso	0.5	Normal	3
T8	Portabilidad	2	No es necesario que sea portable	0
T9	Facilidad de cambio	1		3
T10	Concurrencia	1	No hay concurrencia	0
T11	Incluye objetivos especiales de seguridad	1	La seguridad es manejada por otro sistema	1
T12	Provee acceso directo a terceras partes	1	No provee acceso a terceras partes	0
T13	Se requieren facilidades especiales de entrenamiento a usuarios	1	Sistema fácil de usar	1

El Factor de complejidad técnica, se calcula utilizando la ecuación:

$$TCF = 0.6 + 0.01 * \sum(\text{peso}_i * \text{valor asignado}_i) [5]$$

Donde  $\text{peso}_i$  representa el peso del factor  $i$  y  $\text{valor asignado}_i$  el valor que se le asignó según el aporte del mismo al sistema.

$$TCF = 0.6 + 0.01 * 9$$

$$TCF = 0.6 + 0.09$$

$$TCF = 0.69$$

El Factor de ambiente (**EF**) se calcula similar al Factor de complejidad técnica. Los factores que se cuantifican están relacionados con las habilidades y el entrenamiento del grupo involucrado en el desarrollo del sistema.

La siguiente tabla muestra los factores cuantificados en el Módulo Contenidos y el valor asignado a cada uno.

Factor	Descripción	Peso	Valor Asignado
E1	Familiaridad con el modelo de proyecto utilizado	1.5	5
E2	Experiencia en la aplicación	0.5	4
E3	Experiencia en orientación a objetos	1	5
E4	Capacidad del analista líder	0.5	5
E5	Motivación	1	5
E6	Estabilidad de los requerimientos	2	4
E7	Personal part-time	-1	0
E8	Dificultad del lenguaje de programación	-1	1

Para los factores E1 al E4, un valor asignado de 0 significa sin experiencia, 3 experiencia media y 5 amplia experiencia (experto).

## Capítulo V: Estudio de Factibilidad

Para el factor E5, 0 significa sin motivación para el proyecto, 3 motivación media y 5 alta motivación.

Para el factor E6, 0 significa requerimientos extremadamente inestables, 3 estabilidad media y 5 requerimientos estables sin posibilidad de cambios.

Para el factor E7, 0 significa que no hay personal part-time (es decir todos son full-time), 3 significa mitad y mitad, y 5 significa que todo el personal es part-time (nadie es full-time).

Para el factor E8, 0 significa que el lenguaje de programación es fácil de usar, 3 medio y 5 que el lenguaje es extremadamente difícil.

El Factor de ambiente (EF) se calcula mediante la siguiente ecuación:

$$EF = 1.4 - 0.03 \times \Sigma (\text{peso}_i \times \text{valor asignado}_i) \quad [6]$$

Donde  $\text{peso}_i$  representa el peso del factor  $i$  y  $\text{valor asignado}_i$  el valor que se le asignó según el aporte del mismo al sistema.

$$EF = 1.4 - 0.03 * 29$$

$$EF = 1.4 - 0.87$$

$$EF = 0.53$$

Sustituyendo las ecuaciones 3, 5 y 6 en 4:

$$UCP = UUCP * TCF * EF$$

$$UCP = 109 * 0.69 * 0.53$$

$$UCP = 39.861 \approx 39.9$$

### 5.2.3. Estimación del esfuerzo

El primer paso es definir el factor de conversión horas-hombre/Punto de Casos de Uso que se debe utilizar para calcular el esfuerzo. Para ello se contabilizan cuántos factores de los que afectan al Factor de ambiente están por debajo del valor medio (3), para los factores E1 a E6, luego se contabilizan cuántos factores de los que afectan al Factor de ambiente están por encima del valor medio (3), para los factores E7 y E8.

- Si el total es 2 ó menos, se utiliza el **factor de conversión 20 horas-hombre/Punto de Casos de Uso**, es decir, un Punto de Caso de Uso toma 20 horas-hombre.
- Si el total es 3 ó 4, se utiliza el **factor de conversión 28 horas-hombre/Punto de Casos de Uso**, es decir, un Punto de Caso de Uso toma 28 horas-hombre.

## Capítulo V: Estudio de Factibilidad

- Si el total es mayor o igual que 5, se recomienda efectuar cambios en el proyecto, ya que se considera que el riesgo de fracaso del mismo es demasiado alto.

En el Módulo Contenidos se utilizará el factor de conversión 20 horas-hombre/Puntos de Caso de Uso.

El esfuerzo en horas-hombre (**E**) viene dado por la multiplicación de los Puntos de Casos de Uso ajustados (**UCP**) por el factor de conversión (**CF**)

$$E = UCP * CF$$

$$E = 39.9 * 20$$

$$E = 798$$

Este método proporciona una estimación del esfuerzo en horas-hombre contemplando sólo el desarrollo de la funcionalidad especificada en los casos de uso. Para una estimación más completa de la duración total del proyecto, hay que agregar a la estimación del esfuerzo obtenida por los Puntos de Casos de Uso, las estimaciones de esfuerzo de las demás actividades relacionadas con el desarrollo de software.

Para ello se puede tener en cuenta el siguiente criterio, que estadísticamente se considera aceptable. El criterio plantea la distribución del esfuerzo entre las diferentes actividades de un proyecto, según la siguiente aproximación:

Actividad	Porcentaje	Horas Hombre
Análisis	10 %	199.5
Diseño	20 %	399
Programación	40 %	798
Pruebas	15 %	299.25
Sobrecarga (otras actividades)	15 %	299.25
Total	100 %	1995

Estos valores no son absolutos sino que pueden variar de acuerdo a las características de la organización, del proyecto y de la tecnología utilizada para el desarrollo del sistema.

Tomando como entrada la estimación de tiempo calculada a partir de los Puntos de Casos de Uso, se pueden calcular las demás estimaciones para obtener la duración total del proyecto.

Evaluando la utilización de aproximadamente 8 horas de trabajo diarias y de 24 días laborables en el mes, se estima para la elaboración del Módulo Contenidos una duración de 10,3 meses.

### 5.2.4. Análisis y estimación del costo

Para estimar el costo del proyecto, teniendo en cuenta solo la mano de obra se utiliza la siguiente ecuación:

$$\text{Costo} = \text{CHM} * \text{ET} / \text{CH} \quad (1)$$

Donde **CHM**: Costo Hombre – Mes, **ET**: Esfuerzo Total (mes-hombre) y **CH**: Cantidad de hombres.

Para el Módulo Contenidos:

$$\text{CH} = 1; \text{ET} = 10.3$$

$$\text{CHM} = \text{CH} * \text{Salario Promedio} \quad (2)$$

$$\text{CHM} = 1 * 100$$

$$\text{CHM} = 100$$

Sustituyendo la ecuación 2 en 1:

$$\text{Costo} = 100 * 10.3 / 1$$

$$\text{Costo} = 1030.0 \$$$

### 5.3. Beneficios tangibles e intangibles

El Módulo Contenidos es un sistema que tiene como objetivo desarrollar actividades afines para dar soporte a las funcionalidades de la Plataforma GINA. Esta plataforma es un producto que tiene dentro de sus premisas brindar el servicio de descargas de contenidos a los clientes móviles de CUBACEL implicando una ganancia financiera para la empresa.

Debido a que el producto aún no se pone a disposición de los clientes es impreciso nombrar beneficios tangibles del mismo. Un breve análisis de la situación del mercado al que va dirigido permite reconocer oportunidades únicas para el éxito comercial del mismo.

ETECSA comenzó la comercialización del servicio de telefonía celular en moneda convertible el pasado año, siendo activadas un total de 7 400 líneas a solo 10 días de la liberación de este servicio, (24), número que ha seguido en crecimiento hasta la fecha y al que deben agregarse más de 300 000 clientes que ya existían en el país. (25) CUBACEL, único operador de telefonía móvil en el país, pretende aumentar la gama de prestaciones que ofrece a sus clientes incluyendo la descarga de contenidos, a los cuales podrán acceder tanto los clientes nacionales como los visitantes que hacen uso del servicio de roaming en la isla, teniendo la posibilidad de descargar contenidos con precios que oscilan entre los 0.60 y 1.00 dólares por unidad. Lo antes expuesto permite estimar los aportes

económicos de la plataforma GINA al país, los que son destinados, en su mayor parte, al desarrollo de programas sociales referentes a la telefonía móvil y fija.

La plataforma GINA permitirá además lograr una mejor relación entre CUBACEL y sus clientes, al poner a disposición de los mismos, una amplia gama de servicios y facilidades y permitir la futura agregación de otras prestaciones sin afectar las ya implementadas. También representa una oportunidad de nuevos convenios con empresas nacionales e internacionales; como proveedores de contenidos, facilitando la gestión de sus contenidos dentro de GINA y la fácil agregación de nuevos proveedores con interfaces específicas.

### 5.4. Conclusiones

La estimación permite determinar si el producto o el proyecto es viable. Gracias a la información obtenida en la fase de inicio se pueden definir las funcionalidades que el sistema debe cumplir, los actores que interactúan con el y los factores que pueden significar un riesgo para el mismo.

Con esta información se puede estimar no sólo los recursos humanos, materiales, etc. sino también las fechas de entrega y culminación del proyecto. Se pueden planificar además las actividades de los desarrolladores.

La estimación por Puntos de Casos de Uso resulta muy efectiva para estimar el esfuerzo requerido en el desarrollo de los primeros Casos de Uso de un sistema, si se sigue una aproximación iterativa como el Proceso Unificado de Rational. En éste tipo de aproximación, los primeros Casos de Uso a desarrollar son los que ejercitan la mayor parte de la arquitectura del software y los que a su vez ayudan a mitigar los riesgos más significativos.

En el caso del Módulo Contenidos se puede concluir que una persona puede desarrollar el mismo en un tiempo de 10,3 meses y un costo de \$1030,0.

# CONCLUSIONES

La web para móviles es una de las oportunidades de mercado emergente más grandes de los últimos tiempos, lo cual unido al crecimiento de la tecnología y la conectividad convierten 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.

El desarrollo del Módulo Contenidos permitirá a la Vicepresidencia de Servicios Móviles de ETECSA (VPSM) desplegar la Plataforma GINA logrando que los clientes con servicio de navegación wap puedan acceder a prestaciones sin precedentes en la gama de servicios ofrecidos por el operador telefónico cubano.

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.

Puede ser comercializado con otras empresas de telefonía móvil como parte de la Plataforma GINA o como producto independiente gracias al diseño genérico de sus prestaciones y a la escalabilidad que presenta permitiendo agregar nuevos servicios sin verse afectados los que ya están en funcionamiento.

De manera general es evidente el aporte económico del desarrollo del Módulo Contenidos al analizar la situación económica actual del mercado de contenidos para dispositivos móviles el cual ingresó 13 000 millones de dólares en el año 2005 y pronostica 36 000 millones para el 2009. (26) En el año 2007 fueron responsables del 3.1% del total de ingresos de telefonía móvil. En ese mismo año los usuarios de contenidos móviles llegaron a un 57,1% que representa el 31,1% del total de abonados de telefonía móvil, pronosticándose un crecimiento de esta cifra en los próximos años. (27)

# RECOMENDACIONES

Actualmente la tecnología es un obstáculo para que los operadores de telefonía móvil de muchos países ofrezcan a sus usuarios servicios avanzados, como las descargas de juegos y videos, puesto que las redes y los teléfonos no son capaces de administrar estos contenidos. Por esa razón, el mayor segmento de contenido móvil es el de la música, donde se incluyen los ringtones, porque no requiere teléfonos más avanzados o redes de alta velocidad.

No obstante en el mercado de contenidos para dispositivos móviles, la música es el contenido con mayor demanda, seguido por los juegos y los videos. Esto se debe a la penetración de teléfonos más avanzados en el mercado y el aumento del interés de los usuarios por las nuevas tecnologías comenzando a exigir este tipo de contenidos de sus operadores.

Por esta razón se recomienda implementar en la Plataforma GINA servicios que permitan gestionar juegos y videos provenientes de cualquier proveedor por diferentes vías; permitiendo a la VPSM ofrecer este tipo de contenidos a sus clientes en la medida que las capacidades tecnológicas lo permitan.

Por último se recomienda aumentar el número de subsistemas para la captación, edición y almacenamiento de contenidos desde diferentes fuentes. En esta versión, sólo es posible el aprovisionamiento de contenidos desde el CMS, pero existen muchas vías recomendadas en el mercado internacional, entre las que puede mencionarse la captación de contenidos desde servidores externos pertenecientes a diferentes proveedores.

Un estudio detallado de estas vías y de los posibles proveedores de la VPSM, para conocer la forma en que estos proveedores hacen llegar los contenidos a sus clientes permitiría implementar un mayor número de los subsistemas mencionados, así como la reutilización de las funcionalidades ya implementadas en los subsistemas existentes para el desarrollo de futuros componentes.

# REFERENCIAS BIBLIOGRÁFICAS

1. **Del Valle, Amaury E.** El futuro pasa por el móvil. *Diario Juventud Rebelde*. Suplemento Científico Técnico en RED 145, 08 febrero 2009.
2. **Comisión del Mercado de las Telecomunicaciones CMT.** *Reporte Anual del 2006*. Barcelona, España : ., 2007.
3. **Pearce, Jeame.** Frecuencia Online.com. *Mercado de contenido móvil de América Latina es pequeño pero creciente*. [En línea]  
<http://www.espanol.frecuenciaonline.com/servicios/CMASite/contenidos.php?id=104&identificaArticulo=1307>.
4. **Ginger Alliance.** Content Delivery Plattform. *Ginger Alliance*. [En línea] 2006.  
<http://www.gingerall.com/ga-cdp.html>.
5. **Mgenie.** mProvision Mobile Content Platform. *Mgenie. Mobile phone content delivery platform*. [En línea] 2007. <http://www.mgenie.co.uk/b2b/mgenie/mprovision/>.
6. **Beeweeb.** Mobile Web & Content Delivery Platform. *beeweeb technologies*. [En línea] 2007.  
<http://www.beeweeb.com/mwt/index.php/products/mobile-web-content-delivery-platform/>.
7. **Jacobson, Ivar, Booch, Grady y Rumbaugh, James.** *El Proceso Unificado de Desarrollo de Software*. La Habana : Editorial Félix Varela, 2004.
8. **Molpeceres, Alberto.** *Procesos de Desarrollo RUP, SP y FDD*. Alemania : AT javahispano DOT org, 2002.
9. **Larman, Craig.** *UML y Patrones*. La Habana : Editorial Felix Varela, 2004.
10. **Colectivo de Autores.** *Visual Paradigm for UML User's Guide*. 2007.
11. **Qualitrain.** ¿Por qué seleccionar java como tecnología de desarrollo? *Qualitrain México*. [En línea] Empresa especializada en capacitación., 2009. <http://www.qualitrain.com.mx/index.php/Tecnologia-y-desarrollo/Por-que-seleccionar-a-Java-como-plataforma-de-desarrollo/Page-1.html>.
12. **Iborra Baeza, Ignacio y Lozano Ortega, Miguel Angel.** Presentación del curso "Java y Dispositivos Móviles". *Universidad de Alicante*. [En línea] 2008. <http://www.jtech.ua.es/jdm/present.htm>.
13. **Zukowski, John.** *Programación Java 2 J2SE 1.4*. La Habana : Editorial Félix Varela, 2007.
14. **Colectivo de Autores.** Help - Eclipse SDK. *eclipse*. [En línea] 2008.  
<http://help.eclipse.org/ganymede/index.jsp>.

15. **Ladd, Seth y Donald, Keith.** *Expert Spring MVC and Web Flows*. New York, Estados Unidos : Apress, 2006. ISBN-13 978-1-59059-584-8.
16. **Colectivo de Autores.** Página oficial del proyecto Hibernate. *www.hibernate.org*. [En línea]
17. **Iborra Baeza, Ignacio.** *JUnit*. España : Departamento de Ciencia de la Computación e Inteligencia Artificial, Universidad de Alicante, 2004.
18. **Sun Microsystem.** Java Persistence API FAQ. *Sun Developer Network*. [En línea] 2009. <http://java.sun.com/javaee/overview/faq/persistence.jsp>.
19. **Oficina Española del W3C.** Guía Breve de Servicios Web. *W3C World Wide Web consortium. Oficina Española*. [En línea] <http://www.w3c.es/divulgacion/guiasbreves/ServiciosWeb>.
20. **Espinoza, Humberto.** *PostgreSQL. Una alternativa de DBMS Open Source*. Venezuela : Open Word. Consultores Informáticos Integrales, 2005.
21. **Colectivo de Autores.** Frequently Asked Questions (FAQ) for PostgreSQL. *PostgreSQL*. [En línea] 2008. <http://www.postgresql.org/docs/faqs.FAQ.html>.
22. **Chopra, Vivek, Li, Sing y Genender, Jeff.** *Apache Tomcat 6*. s.l. : Anaya Multimedia, 2008. ISBN 8441523770.
23. **Rational Software Corp.** The Unified Modeling Language for Object-Oriented Development. Septiembre 1997. Vols. Documentation Set, ver. 1.1.
24. **Mateu Francés, Lianis.** Comercializadas en Cuba más de 7 mil 400 líneas de teléfonos celulares. *Radio Rebelde*. [En línea] 24 de Abril de 2008. <http://www.radiorebelde.com.cu/noticias/economia/economia%201-240408.html>.
25. **del Sol Reyes, Aday.** Celulares en CUC. *Cubasi.cu*. [En línea] 8 de Mayo de 2008. <http://www.cubasi.cu/DesktopDefault.aspx?SPK=160&CLK=197159&LK=1&CK=101211&SPKA=35>.
26. **Blázquez, Susana.** Un mercado de millones de euros. La lucha por los contenidos llega a los móviles. *Revista TELOS*. [En línea] 2006. <http://www.campusred.net/telos/articulonoticia.asp?idArticulo=4&rev=68>.
27. **Infobaeprofesional.com.** El teléfono móvil se convierte en un escenario estratégico para las empresas. *Marketing Marcas y Tendencias de consumo*. [En línea]

# BIBLIOGRAFÍA

**Apache Software Foundation.** The Apache Software Foundation. *Apache Tomcat*. [En línea] 2008. <http://tomcat.apache.org/index.html>.

**Ballestero, Maria P.** *Evolución de la Telefonía Móvil en el Mercado de las Telecomunicaciones*. Madrid : CEPREDE Centro de Predicción Económica, 2002.

**Bartolomé, Tatiana y Herrero, Jesús.** *Sistema de Gestión de Contenidos y Visualización de TV Digital en Terminales Móviles*. Turitec : VI Congreso "Turismo y Tecnologías de la Información y las Comunicaciones" Turitec 2006, 2006.

**Beeweeb.** Mobile Web & Content Delivery Platform. *beeweeb technologies*. [En línea] 2007. <http://www.beeweeb.com/mwt/index.php/products/mobile-web-content-delivery-platform/>.

**Bermejo Sanz, Laura y Gómez Monreal, Enrique.** Eclipse como IDE. URJC-ITIS Herramientas CASE y 4GL. España : Universidad Rey Juan Carlos.

**Blázquez, Susana.** Un mercado de millones de euros. La lucha por los contenidos llega a los móviles. *Revista TELOS*. [En línea] 2006. <http://www.campusred.net/telos/articulonoticia.asp?idArticulo=4&rev=68>.

**CATA Alliance Canadian Advance Technology Alliance.** *Mobile Content & Service*. Canada : Desjardins, 2007.

**Chopra, Vivek, Li, Sing y Genender, Jeff.** *Apache Tomcat 6*. s.l. : Anaya Multimedia, 2008. ISBN 8441523770.

**CITEL Comisión Interamericana de Telecomunicaciones.** *Informe Anual de la Comisión Interamericana de Telecomunicaciones a la Asamblea General*. Washington DC, Estados Unidos : Organización de los Estados Americanos, 2005.

**Colectivo de Autores.** *Comunicaciones Móviles e Inalámbricas*. Grupo de Análisis y Prospectiva del Sector de las Telecomunicaciones GAPTEL : Red.es, Septiembre 2005.

**Colectivo de Autores.** Contenidos aumentan ingresos de los operadores de telefonía móvil en AL. *esemanal.com.mx*. [En línea] 2007. [http://www.esemanal.com.mx/articulos.php?id\\_sec=2&id\\_art=6757](http://www.esemanal.com.mx/articulos.php?id_sec=2&id_art=6757).

**Colectivo de Autores.** Frequently Asked Questions (FAQ) for PostgreSQL. *PostgreSQL*. [En línea] 2008. <http://www.postgresql.org/docs/faqs.FAQ.html>.

**Colectivo de Autores.** Help - Eclipse SDK. *eclipse*. [En línea] 2008. <http://help.eclipse.org/ganymede/index.jsp>.

- Colectivo de Autores.** Página oficial del proyecto Hibernate. *www.hibernate.org*. [En línea]
- Colectivo de Autores.** Visual Paradigm for UML User's Guide. 2007.
- Comisión del Mercado de las Telecomunicaciones CMT.** *Reporte Anual del 2006*. Barcelona, España : ., 2007.
- Comisión del Mercado de las Telecomunicaciones CMT.** *Reporte Anual del 2007*. Barcelona, España : ., 2008.
- del Pozo Paumard, Cesar.** *Servicios de Gestión en Telefonía Móvil*. Madrid, España : Universidad Politécnica de Madrid, Abril 2006.
- del Sol Reyes, Aday.** Celulares en CUC. *Cubasi.cu*. [En línea] 8 de Mayo de 2008.  
<http://www.cubasi.cu/DesktopDefault.aspx?SPK=160&CLK=197159&LK=1&CK=101211&SPKA=35>.
- Del Valle, Amaury E.** El futuro pasa por el móvil. *Diario Juventud Rebelde*. Suplemento Científico Técnico en RED 145, 08 febrero 2009.
- Espinoza, Humberto.** *PostgreSQL. Una alternativa de DBMS Open Source*. Venezuela : Open Word. Consultores Informáticos Integrales, 2005.
- Frick, Oliver, Schmidt, Albrecht y Schröder, Henning.** *WAP - Designing for Small User Interfaces*. Karlsruhe, Alemania : universidad de Karlsruhe.
- Ginger Alliance.** Content Delivery Plattform. *Ginger Alliance*. [En línea] 2006.  
<http://www.gingerall.com/ga-cdp.html>.
- Iborra Baeza, Ignacio y Lozano Ortega, Miguel Angel.** Presentación del curso "Java y Dispositivos Móviles". *Universidad de Alicante*. [En línea] 2008. <http://www.jtech.ua.es/jdm/present.htm>.
- Iborra Baeza, Ignacio.** *JUnit*. España : Departamento de Ciencia de la Computación e Inteligencia Artificial, Universidad de Alicante, 2004.
- Infobaeprofesional.com.** El teléfono móvil se convierte en un escenario estratégico para las empresas. *Marketing Marcas y Tendencias de consumo*. [En línea]
- Institute of Electrical and Electronics Engineers (IEEE).** *IEEE Recomendad Practice for Software Requirements Specifications*. ANSI 830-1993.
- Jacobson, Ivar, Booch, Grady y Rumbaugh, James.** *El Proceso Unificado de Desarrollo de Software*. La Habana : Editorial Félix Varela, 2004.
- Küng, Stefan, Onken, Lübbe y Large, Simon.** *TortoiseSVN: Un cliente de Subversion para Windows: Version 1.4.1*. Alemania : s.n., 2006.
- Ladd, Seth y Donald, Keith.** *Expert Spring MVC and Web Flows*. New York, Estados Unidos : Apress, 2006. ISBN-13 978-1-59059-584-8.
- Larman, Craig.** *UML y Patrones*. La Habana : Editorial Felix Varela, 2004.

- Margain Alvarez, Ing. Jorge.** Internet en los celulares se llama "Wap". *Microasist. Soluciones humanas a problemas técnicos.* [En línea] 2007. <http://microasist.com.mx/noticias/mo/mo0503.shtml>.
- Mateu Francés, Lianis.** Comercializadas en Cuba más de 7 mil 400 líneas de teléfonos celulares. *Radio Rebelde.* [En línea] 24 de Abril de 2008. <http://www.radiorebelde.com.cu/noticias/economia/economia%201-240408.html>.
- McLaughlin, Brett D., Pollice, Gary y West, David.** *Head First Object-Oriented Analysis and Design.* Estados Unidos : O'Reilly Media, Inc., 2007. ISBN-10: 0-596-00867-8.
- Mgenie.** mProvision Mobile Content Platform. *Mgenie. Mobile phone content delivery platform.* [En línea] 2007. <http://www.mgenie.co.uk/b2b/mgenie/mprovision/>.
- Microsoft Corporation.** Microsoft Visual Studio en MSDN. *Manual del programador de .NET Framework. Serialización.* [En línea] 2009. [http://msdn.microsoft.com/es-es/library/7ay27kt9\(VS.80\).aspx](http://msdn.microsoft.com/es-es/library/7ay27kt9(VS.80).aspx).
- Molina Gasca, Laura.** *Personalización de servicios WAP a través de dispositivos móviles utilizando M-commerce.* Distrito Federal, México : Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional, Octubre 2004. Departamento de Ingeniería Eléctrica. Sección de Computación.
- Molpeceres, Alberto.** *Procesos de Desarrollo RUP, SP y FDD.* Alemania : AT javahispano DOT org, 2002.
- Morales Machuca, Carlos Andrés.** *Estado del Arte: Servicios Web.* s.l. : Universidad Nacional de Colombia.
- Oficina Española del W3C.** Guía Breve de Servicios Web. *W3C World Wide Web consortium. Oficina Española.* [En línea] <http://www.w3c.es/divulgacion/guiasbreves/ServiciosWeb>.
- Pearce, Jeame.** Frecuencia Online.com. *Mercado de contenido móvil de América Latina es pequeño pero creciente.* [En línea] <http://www.espanol.frecuenciaonline.com/servicios/CMASite/contenidos.php?id=104&identificaArticulo=1307>.
- Pilone, Dan y Miles, Russ.** *Head First Software Development.* Estados Unidos : O'Reilly Media, Inc., 2008. ISBN-10: 0-596-52735-7.
- Pressman, Roger S.** *Ingeniería del Software. Un enfoque práctico.* La Habana : Editorial Félix Varela, 2005.
- Qualitrain.** ¿Por qué seleccionar java como tecnología de desarrollo? *Qualitrain México.* [En línea] Empresa especializada en capacitación., 2009. <http://www.qualitrain.com.mx/index.php/Tecnologia-y-desarrollo/Por-que-seleccionar-a-Java-como-plataforma-de-desarrollo/Page-1.html>.
- Ramirez, José Antonio.** Evolución de lo dispositivos Móviles . *RED, la comunidad de expertos en TICs.* [En línea] <http://www.red.com.mx/tema5.php>.

**Rational Software Corp.** The Unified Modeling Language for Object-Oriented Development. Septiembre 1997. Vols. Documentation Set, ver. 1.1.

**Sánchez González, Carlos.** *Seguridad no intrusiva con Acegi Security for Spring*. España : Dpto. Consultoría Soluciones e-Bussines SOFTGAL, Diciembre 2004.

**Sánchez González, Carlos.** *Seguridad no intrusiva con Acegi Security System for Spring*. Coruña, España : Softgal.

**Soto Pérez, Gloria.** Importancia de la WAP. *Microasist. Soluciones humanas a problemas técnicos*. [En línea] 2007. <http://microasist.com.mx/noticias/mo/gspmo100804.shtml>.

**SOURCEFORGE.NET.** Spring Security 2.0.0 Released. *Acegi security System for spring*. [En línea] <http://www.acegisecurity.org/>.

**Sun Microsystem.** Java Persistence API FAQ. *Sun Developer Network*. [En línea] 2009. <http://java.sun.com/javaee/overview/faq/persistence.jsp>.

**Telefónica I+D.** *Informe Anual. Telefónica Investigación y Desarrollo*. España : Telefónica I+D, 2007.

**Telefónica I+D.** *Las Telecomunicaciones y la Movilidad en la Sociedad de la Información*. España : Telefónica I+D.

**Telefónica Investigación y Desarrollo.** Gestión de Contenidos para portales móviles UMTS. *Telefónica I+D*. [En línea] 2002. <http://www.tid.es/presencia/publicaciones/comsid/esp/24/art10.pdf>.

**Tigris.** Subversion FAQ. *Tigris.org Open Source Software Engineering Tools*. [En línea] 2008. <http://subversion.tigris.org/faq.html>.

**Tigris.** TortoiseSVN FAQ. *Tigris.org Open Source Spftware Engineering Tools*. [En línea] 2008. <http://tortoisesvn.tigris.org/faq.html>.

**Unión Internacional de Telecomunicaciones(UIT).** *digital.life ITU Internet Report 2006*. Geneva, España : ., 2006.

**Unión Internacional de Telecomunicaciones (UIT).** *UIT Informe Anual de la Unión 2007*. Suiza, Ginebra : ., 2008.

**Zukowski, John.** *Programación Java 2 J2SE 1.4*. La Habana : Editorial Félix Varela, 2007.

# GLOSARIO DE TÉRMINOS

## A

---

**Arquitectura:** Descripción de la organización y estructura de un sistema.

## B

---

**ByteCode:** Es un código intermedio más abstracto que el código máquina. Habitualmente es tratado como un fichero binario que contiene un programa ejecutable similar a un módulo objeto, que es un fichero binario producido por el compilador cuyo contenido es el código objeto o código máquina.

**Bean:** Componente de software reutilizable que sigue una convención de nombres concreta para definir las propiedades y eventos que comprende. Puede manipularse visualmente mediante una herramienta de construcción.

## C

---

**Caso de Uso:** Descripción narrativa textual de la secuencia de eventos y acciones que ocurren cuando un usuario parte o divide en un diálogo con un sistema durante un proceso significativo.

**Ciclo de vida del software:** Ciclo que cubre cuatro fases en el siguiente orden: inicio, elaboración, construcción y transición.

**Clase Persistente:** Clases en una aplicación que implementan las entidades del problema.

**Cliente Móvil:** Se refiere a un dispositivo móvil conectado a la red celular y que accede a los servicios prestados por el operador móvil al que está suscrito.

**Cliente:** Persona, organización o grupo de personas que encarga la construcción de un sistema, ya sea empezado desde cero, o mediante el refinamiento de versiones sucesivas.

**Confiabilidad:** Capacidad de un producto de realizar su función de la manera prevista. De otra forma, la confiabilidad se puede definir también como la probabilidad en que un producto realizará su función prevista sin incidentes por un período de tiempo especificado y bajo condiciones indicadas.

**CVS:** también conocido como *Concurrent Versioning System*, es una aplicación informática que implementa un sistema de control de versiones: mantiene el registro de todo el trabajo y los cambios en los ficheros (código fuente principalmente) que forman un proyecto (de programa) y permite que distintos desarrolladores (potencialmente situados a gran distancia) colaboren. CVS se ha hecho popular en el mundo del software libre. Sus desarrolladores difunden el sistema bajo la licencia GPL.

### D

---

**DAO:** Es un componente de software que suministra una interfaz común entre la aplicación y uno o más dispositivos de almacenamiento de datos, tales como una Base de datos o un archivo.

**Desarrolladores:** Personas participantes en los flujos de trabajo fundamentales. Por ejemplo, ingeniero de casos de usos, programador, etc.

**Dominio:** Área de conocimiento o actividad caracterizada por un conjunto de conceptos y terminología comprendidos por los practicantes de ese dominio.

**DRM:** Es un término genérico que se refiere a las tecnologías de control de acceso usada por editoriales y dueños de derechos de autor para limitar el uso de medios o dispositivos digitales. También se puede referir a las restricciones asociadas a instancias específicas de obras digitales o dispositivos.

### E

---

**EJB (Enterprise JavaBean):** Son una de las API que forman parte del estándar de construcción de aplicaciones empresariales J2EE de Sun Microsystems. Su especificación detalla cómo los servidores de aplicaciones proveen objetos desde el lado del servidor que son, precisamente, los EJB.

### F

---

**Framework:** Microarquitectura que proporciona una plantilla incompleta para sistemas de un determinado dominio. Puede tratarse por ejemplo, de un subsistema construido para ser ampliado y reutilizado.

### G

---

**GNU GPL (GNU General Public Licence):** Es una licencia creada por la Free Software Foundation a mediados de los 80, y está orientada principalmente a proteger la libre distribución, modificación y uso

de software. Su propósito es declarar que el software cubierto por esta licencia es software libre y protegerlo de intentos de apropiación que restrinjan esas libertades a los usuarios.

**GPRS:** Servicio general de paquetes vía radio. Es una extensión del GSM para la transmisión de datos no conmutada (o por paquetes).

**GPS (Global Positioning System):** Es un Sistema Global de Navegación por Satélite (GNSS) que permite determinar en todo el mundo la posición de un objeto, una persona, un vehículo o una nave, con una precisión hasta de centímetros.

**GSM:** Sistema Global para las comunicaciones móviles. Es un servicio ofrecido por las empresas operadoras de telefonía móvil que permite determinar, con una cierta precisión, donde se encuentra físicamente un terminal móvil determinado.

**GUI (Interfaz gráfica de usuario):** Interfaz de iconos de los sistemas operativos actuales.

## H

---

**HTTP (Protocolo de transferencia de hipertexto):** Protocolo de aplicación utilizado por la World Wide Web para solicitar, transmitir y recibir documentos web.

## I

---

**International Telecommunication Union (ITU):** La Unión Internacional de Telecomunicaciones, es una organización internacional del sistema de las Naciones Unidas en la cual los gobiernos y el sector privado coordinan los servicios y redes mundiales de telecomunicaciones.

## J

---

**Javadoc:** es una utilidad de Sun Microsystems para la generación de documentación de APIs en formato HTML a partir de código fuente Java. Javadoc es el estándar de la industria para documentar clases de Java. La mayoría de los IDEs los generan automáticamente.

**JDBC (Java Database Connectivity):** Es una API que permite la ejecución de operaciones sobre bases de datos desde el lenguaje de programación Java, independientemente del sistema operativo donde se ejecute o de la base de datos a la cual se accede, utilizando el dialecto SQL del modelo de base de datos que se utilice.

**JDK (Kit de desarrollo de java):** Herramienta de Sun que ayuda a los desarrolladores a crear componentes reutilizables incluyendo la información introductoria, un tutorial, beans de muestra, así como un contenedor de muestra.

**JRE (Java Runtime Environment):** Se corresponde con un conjunto de utilidades que permite la ejecución de programas java sobre todas las plataformas soportadas.

**JVM (Java Virtual Machine):** Sistema que abre, verifica y ejecuta bytecodes de java.

### L

---

**Licencia Apache:** Es una licencia de software libre creada por la Apache Software Foundation (ASF). La licencia Apache (con versiones 1.0, 1.1 y 2.0) requiere la conservación del aviso de copyright pero no es una licencia copyleft, ya que permite el uso y distribución del código fuente para software libre y software propietario.

**Licencia BSD:** Es la licencia de software otorgada principalmente para los sistemas BSD (*Berkeley Software Distribution*). Pertenece al grupo de licencias de software Libre. Esta licencia tiene menos restricciones en comparación con otras como la GPL estando muy cercana al dominio público. La licencia BSD al contrario que la GPL permite el uso del código fuente en software no libre.

**LOG:** Es un registro oficial de eventos durante un periodo de tiempo en particular.

### M

---

**Memoria RAM:** Memoria de acceso aleatorio o memoria de acceso directo. Es un tipo de memoria temporal que pierde sus datos cuando se queda sin energía (por ejemplo, al apagar la computadora), por lo cual es una memoria volátil.

**MP3:** Formato de audio digital comprimido con pérdida desarrollado por el Moving Picture Experts Group.

### O

---

**Open source:** Es el término con el que se conoce al software distribuido y desarrollado libremente.

**Operadores:** Se refiere a los operadores de telefonía móvil. Empresas u otras instituciones.

**ORM:** El mapeo objeto-relacional (más conocido por su nombre en inglés, Object-Relational mapping, o sus siglas O/RM, ORM, y O/R mapping) es una técnica de programación para convertir datos entre el sistema de tipos utilizado en un lenguaje de programación orientado a objetos y el utilizado en una base de datos relacional. En la práctica crea una base de datos orientada a objetos virtual, sobre la base de datos relacional. Esto posibilita el uso de las características propias de la orientación a objetos (básicamente herencia y polimorfismo).

### **P**

---

**Persistencia:** Habilidad de un objeto para grabar su propio estado para que pueda reproducirse en el futuro en un ambiente distinto.

**Plataforma .NET:** Es un proyecto de Microsoft para crear una nueva plataforma de desarrollo de software con énfasis en transparencia de redes, con independencia de plataforma de hardware y que permita un rápido desarrollo de aplicaciones.

**Plug-ins:** Elemento añadido al buscador para reemplazar el ambiente de ejecución de java del mismo.

**Portabilidad:** Se define como la característica que posee un software para ejecutarse en diferentes plataformas, el código fuente del software es capaz de reutilizarse en vez de crearse un nuevo código cuando el software pasa de una plataforma a otra. A mayor portabilidad menor es la dependencia del software con respecto a la plataforma.

**Primera Generación:** La primera generación (1G) de la telefonía celular se caracterizó por ser analógica, es decir, sólo para voz. La gran demanda y su evolución dieron origen a la tecnología CDMA por sus siglas en inglés (Acceso Múltiple por División de Código), utilizada en proyectos bélicos de la Segunda Guerra Mundial y donde la forma de acceso en cuanto al ancho de banda fue aprovechada de manera más eficiente, permitiendo más usuarios en el mismo canal de frecuencia. Su única limitante era la calidad de servicio, pues se degradaba conforme aumentaban los abonados

**Pruebas de Regresión:** cualquier tipo de pruebas de software que intentan descubrir las causas de nuevos errores, carencias de funcionalidad, o divergencias funcionales con respecto al comportamiento esperado del software, inducidos por cambios recientemente realizados en partes de la aplicación que anteriormente al citado cambio no eran propensas a este tipo de error. Esto implica que el error tratado se reproduce como consecuencia inesperada del citado cambio en el programa.

### **R**

---

**Repositorio de Contenido:** Es un sitio centralizado donde se almacena y mantiene información digital como los contenidos para dispositivos móviles.

**Roaming:** Concepto utilizado en comunicaciones inalámbricas que está relacionado con la capacidad de un dispositivo para moverse de una zona de cobertura a otra. Es la capacidad de hacer y recibir llamadas en redes móviles fuera del área de servicio local de su compañía; es decir, dentro de la zona de servicio de otra empresa del mismo país, o bien durante una estancia en otro país diferente, con la red de una empresa extranjera.

## S

---

**Segunda Generación:** La segunda generación (2G) aparece con el estándar GSM Europeo, que utiliza tecnología de Acceso Múltiple por División de Tiempo (TDMA); caracterizada por ser digital, empleó protocolos que soportan velocidades de información más altas para voz, pero limitados en comunicación de datos. Se ofrecieron, además, servicios auxiliares como transmisión de datos, fax y SMS (Short Message Service).

**Serialización:** Proceso en el que un objeto se almacena a sí mismo, escribiendo los valores que describen su estado.

**Servlet:** Programa de java del lado del servidor incluido dentro del servidor web.

**Stand Alone:** Por lo general se aplica a un aparato o sistema que puede funcionar por sí mismo. Por ejemplo, una computadora que tiene todo lo necesario para funcionar independientemente, a diferencia de terminales que necesitan estar conectadas a una computadora central. Una aplicación Stand Alone se ejecuta en una estación de trabajo normal, no se conecta a servidores, ni utiliza servicios web.

**SOA (Arquitectura Orientada a Servicios):** Es un concepto de arquitectura de software que define la utilización de servicios para dar soporte a los requisitos del negocio. Permite la creación de sistemas altamente escalables que reflejan el negocio de la organización, a su vez brinda una forma estándar de exposición e invocación de servicios (comúnmente pero no exclusivamente servicios web), lo cual facilita la interacción entre diferentes sistemas propios o de terceros.

## T

---

**TDMA:** Acceso múltiple por divisiones de tiempo. Es una técnica de multiplexación que distribuye las unidades de información en ranuras alternas de tiempo, proveyendo acceso múltiple a un reducido número de frecuencias.

**Tercera Generación:** La tercera generación 3G se caracteriza por converger voz y datos con acceso inalámbrico a Internet; en otras palabras, es apta para aplicaciones multimedia: voz, acceso a Internet, e-mail, transferencia de documentos e imágenes fijas, servicios de ubicación, video de alta definición y altas transmisiones de datos, entre otras funciones. Los protocolos empleados en los sistemas 3G soportan altas velocidades de información y están enfocados para aplicaciones más allá de la voz, como audio (mp3), video en movimiento, videoconferencia y acceso rápido a Internet, sólo por nombrar algunas.

### U

---

**Usabilidad:** Se refiere a la capacidad de un software de ser comprendido, aprendido, usado y ser atractivo para el usuario, en condiciones específicas de uso.

### W

---

**WAP (Wireless Application Protocol):** Un estándar abierto internacional para aplicaciones que utilizan las comunicaciones inalámbricas, p.ej. acceso a servicios de Internet desde un teléfono móvil.

**Web Service:** Conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones. Distintas aplicaciones de software desarrolladas en lenguajes de programación diferentes, y ejecutadas sobre cualquier plataforma, pueden utilizar los servicios web para intercambiar datos en redes de ordenadores como Internet. La interoperabilidad se consigue mediante la adopción de estándares abiertos.

### X

---

**XML:** Lenguaje de marcado utilizado para definir y utilizar las etiquetas necesarias para describir datos jerárquicos.

## Anexo I: Descripción de los casos de uso

### CU Gestionar Contenido

<b>Caso de Uso:</b>	Gestionar Contenido	
<b>Actores:</b>	M_CMS	
<b>Flujo Normal de Eventos</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	
1. Decide gestionar un contenido. Para ello puede: Adicionar un Contenido. Modificar un Contenido. Incrementar el número de descargas. Eliminar un Contenido.	2. Si decide Adicionar un Contenido ir a la <b>Sección Adicionar Contenido</b> . Modificar un Contenido ir a la <b>Sección Modificar Contenido</b> . Incrementar el número de descargas ir a la <b>Sección Incrementar número de Descargas</b> . Eliminar un Contenido ir a la <b>Sección Eliminar Contenido</b> .	
<b>Sección Adicionar Contenido</b>		
2. El M_CMS envía un contenido al sistema.	3. Verifica que el contenido no representa una categoría.	
	4. El sistema almacena los datos del contenido en la base de datos.	
	5. Almacena el arreglo de byte en un archivo dentro del repositorio de contenido.	
<b>Flujos Alternos</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	
	3.1. Almacena los datos del contenido en la base de datos y finaliza el caso de uso.	
	4.1. Envía un mensaje de error si los datos no pudieron ser almacenados en la base de datos.	
	5.1. Envía un mensaje de error si el archivo no pudo ser almacenado en el repositorio.	
	5.2. Almacena un log con el identificador del contenido que no pudo ser almacenado en el repositorio.	
<b>Sección Modificar Contenido</b>		
2. El M_CMS solicita el servicio Buscar Contenido.	3. Se invoca el Caso de Uso Buscar Contenido.	
4. Selecciona el contenido que desea modificar.		
5. Envía al sistema el contenido modificado.		
	6. Modifica los datos del contenido en la	

## Anexo I: Descripción de los Casos de Uso

	base de datos.
	7. Modifica los datos del contenido en el repositorio.
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	6.1 Envía un mensaje de error si los datos no pudieron ser modificados en la base de datos.
	7.1. Envía un mensaje de error si los datos no pudieron ser modificados en el repositorio.
	7.2. Almacena un log con el identificador el archivo que no pudo ser modificado en el repositorio.
<b>Sección Incrementar número de descargas</b>	
2. El M_CMS solicita el servicio Buscar Contenido.	3. Se invoca el Caso de Uso Buscar Contenido.
4. Selecciona el contenido al que desea incrementar el número de descargas.	
5. Envía al sistema el contenido seleccionado.	
	6. Obtiene el número de descargas del contenido.
	7. Incrementa el número de descargas en 1.
	8. Modifica los datos del contenido en la base de datos.
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	8.1 Envía un mensaje de error si los datos no pudieron ser modificados en la base de datos.
<b>Sección Eliminar Contenido</b>	
2. El M_CMS solicita el servicio Buscar Contenido.	3. Se invoca el Caso de Uso Buscar Contenido.
4. Selecciona el contenido que desea eliminar.	5. Elimina los datos del contenido de la base de datos.
	6. Elimina el contenido del repositorio.
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	5.1 Envía un mensaje de error si los datos no pudieron eliminarse de la base de datos.
	6.1 Envía un mensaje de errores si los datos no pudieron eliminarse del repositorio.
	7. Almacena un log con el identificador del archivo que no se pudo eliminar en el repositorio
<b>Poscondiciones</b>	Quedan gestionados los datos del contenido

## Anexo I: Descripción de los Casos de Uso

### CU Gestionar País

<b>Caso de Uso:</b>	Gestionar País	
<b>Actores:</b>	M_CMS	
<b>Flujo Normal de Eventos</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	
1. Decide gestionar un país. Para ello puede: Adicionar un País ( <b>Sección Adicionar País</b> ) Modificar un País ( <b>Sección Modificar País</b> ) Eliminar un País ( <b>Sección Eliminar País</b> )		
<b>Sección Adicionar País</b>		
2. El M_CMS envía un país al sistema.	3. El sistema almacena los datos del país en la base de datos.	
<b>Flujos Alternos</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	
	3.1. Envía un mensaje de error si los datos no pudieron ser almacenados en la base de datos.	
<b>Sección Modificar País</b>		
2. El M_CMS solicita el servicio Buscar País.		
3. Selecciona el país que desea modificar.		
4. Envía al sistema el país modificado.	5. Modifica el país en la base de datos.	
<b>Flujos Alternos</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	
	5.1. Envía un mensaje de error si los datos no pudieron ser modificados en la base de datos.	
<b>Sección Eliminar País</b>		
2. El M_CMS solicita el servicio Buscar País.		
3. Selecciona el país que desea eliminar.		
4. Envía al sistema el país eliminado.	5. Elimina el país en la base de datos.	
<b>Flujos Alternos</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	
	6.1 Envía un mensaje de error si los datos no pudieron ser eliminados en la base de datos”.	
<b>Poscondiciones</b>	Quedan gestionados los datos del país	

### CU Gestionar Autor

<b>Caso de Uso:</b>	Gestionar Autor	
<b>Actores:</b>	M_CMS	
<b>Flujo Normal de Eventos</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	
1. Decide gestionar un Autor. Para ello puede: Adicionar un Autor ( <b>Sección Adicionar Autor</b> ) Modificar un Autor ( <b>Sección Modificar Autor</b> ) Eliminar un Autor ( <b>Sección Eliminar Autor</b> )		

## Anexo I: Descripción de los Casos de Uso

<b>Sección Adicionar Autor</b>	
2. El M_CMS envía un Autor al sistema.	3. El sistema almacena los datos del Autor en la base de datos.
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	3.1 Envía un mensaje de error si los datos no pudieron ser almacenados en la base de datos.
<b>Sección Modificar Autor</b>	
2. El M_CMS solicita el servicio Buscar Autor.	
3. Selecciona el Autor que desea modificar.	
4. Envía al sistema el Autor modificado.	5. Modifica el Autor en la base de datos.
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	6.1. Envía un mensaje de error si los datos no pudieron ser modificados en la base de datos.
<b>Sección Eliminar Autor</b>	
2. El M_CMS solicita el servicio Buscar Autor.	
3. Selecciona el Autor que desea eliminar.	
4. Envía al sistema el Autor eliminado.	5. Elimina el Autor en la base de datos.
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	6.1 Envía un mensaje de error si los datos no pudieron ser eliminados en la base de datos”.
<b>Poscondiciones</b>	Quedan gestionados los datos del Autor

### CU Gestionar Proveedor

<b>Caso de Uso:</b>	Gestionar Proveedor
<b>Actores:</b>	M_CMS
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. Decide gestionar un proveedor. Para ello puede: Adicionar un Proveedor ( <b>Sección Adicionar Proveedor</b> ) Modificar un Proveedor ( <b>Sección Modificar Proveedor</b> ) Eliminar un Proveedor ( <b>Sección Eliminar Proveedor</b> )	
<b>Sección Adicionar Proveedor</b>	
2. El M_CMS envía un proveedor al sistema.	3. El sistema almacena los datos del proveedor en la base de datos.
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	3.1 Envía un mensaje de error si los datos no pudieron ser almacenados en la base de datos.

## Anexo I: Descripción de los Casos de Uso

<b>Sección Modificar Proveedor</b>	
2. El M_CMS solicita el servicio Buscar Proveedor.	
3. Selecciona el proveedor que desea modificar.	
4. Envía al sistema el proveedor modificado.	5. Modifica el proveedor en la base de datos.
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	6.1. Envía un mensaje de error si los datos no pudieron ser modificados en la base de datos.
<b>Sección Eliminar Proveedor</b>	
2. El M_CMS solicita el servicio Buscar Proveedor.	
3. Selecciona el proveedor que desea eliminar.	
4. Envía al sistema el proveedor eliminado.	5. Elimina el proveedor en la base de datos.
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	6.1 Envía un mensaje de error si los datos no pudieron ser eliminados en la base de datos”.
<b>Poscondiciones</b>	Quedan gestionados los datos del proveedor

### CU Gestionar Clasificación

<b>Caso de Uso:</b>	Gestionar Clasificación
<b>Actores:</b>	M_CMS
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. Decide gestionar una clasificación. Para ello puede: Adicionar una Clasificación ( <b>Sección Adicionar Clasificación</b> ) Modificar una Clasificación ( <b>Sección Modificar Clasificación</b> ) Eliminar una Clasificación ( <b>Sección Eliminar Clasificación</b> )	
<b>Sección Adicionar Clasificación</b>	
2. El M_CMS envía una clasificación al sistema.	3. El sistema almacena los datos de la clasificación en la base de datos.
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	3.1 Envía un mensaje de error si los datos no pudieron ser almacenados en la base de datos.
<b>Sección Modificar Clasificación</b>	
2. El M_CMS solicita el servicio Buscar Clasificación.	
3. Selecciona la clasificación que desea	

## Anexo I: Descripción de los Casos de Uso

modificar.	
4. Envía al sistema la clasificación modificada.	5. Modifica la clasificación en la base de datos.
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	6.1. Envía un mensaje de error si los datos no pudieron ser modificados en la base de datos.
<b>Sección Eliminar Clasificación</b>	
2. El M_CMS solicita el servicio Buscar Clasificación.	
3. Selecciona la clasificación que desea eliminar.	
4. Envía al sistema la clasificación eliminada.	5. Elimina la clasificación en la base de datos.
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	6.1 Envía un mensaje de error si los datos no pudieron ser eliminados en la base de datos”.
<b>Poscondiciones</b>	Quedan gestionados los datos de la clasificación

### CU Gestionar Orden

<b>Caso de Uso:</b>	Gestionar Orden
<b>Actores:</b>	M_ContentDelivery
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. Decide gestionar una orden. Para ello puede: Adicionar una Orden ( <b>Sección Adicionar Orden</b> ) Modificar una Orden ( <b>Sección Modificar Orden</b> ) Eliminar una Orden ( <b>Sección Eliminar Orden</b> )	
<b>Sección Adicionar Orden</b>	
2. El M_CMS envía una orden al sistema.	3. El sistema almacena los datos de la orden en la base de datos.
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	3.1 Envía un mensaje de error si los datos no pudieron ser almacenados en la base de datos.
<b>Sección Modificar Orden</b>	
2. El M_CMS solicita el servicio Buscar Orden.	
3. Selecciona la orden que desea modificar.	
4. Envía al sistema la orden modificada.	5. Modifica la orden en la base de datos.
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	6.1. Envía un mensaje de error si los datos no pudieron ser modificados en la base

## Anexo I: Descripción de los Casos de Uso

	de datos.
<b>Sección Eliminar Orden</b>	
2. El M_CMS solicita el servicio Buscar Orden.	
3. Selecciona la orden que desea eliminar.	
4. Envía al sistema la orden eliminada.	5. Elimina la orden en la base de datos.
	6. Almacena la orden eliminada en un historial.
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	5.1 Envía un mensaje de error si los datos no pudieron ser eliminados en la base de datos”.
<b>Poscondiciones</b>	Quedan gestionados los datos de la orden.

### CU Buscar Contenido

<b>Caso de Uso:</b>	Buscar Contenido
<b>Actores:</b>	Externo
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
<p>1. El M_CMS solicita al sistema una de las siguientes opciones:</p> <ul style="list-style-type: none"> <li>Buscar todos los contenidos por categoría.</li> <li>Buscar todos los contenidos por proveedor.</li> <li>Buscar todos los contenidos por autor.</li> <li>Buscar todos los contenidos por clasificación.</li> <li>Buscar todas las categorías.</li> <li>Buscar contenido por ID.</li> <li>Buscar contenido con arreglo de datos.</li> <li>Buscar todos los contenidos disponibles por autor.</li> <li>Buscar todos los contenidos disponibles por proveedor.</li> <li>Buscar todos los contenidos disponibles por clasificación.</li> <li>Buscar todas las categorías disponibles.</li> </ul>	<p>2. Si solicita</p> <ul style="list-style-type: none"> <li>Buscar todos los contenidos por categoría ir a la <b>Sección Buscar todos los contenidos por categoría.</b></li> <li>Buscar todos los contenidos por proveedor ir a la <b>Sección Buscar todos los contenidos por proveedor.</b></li> <li>Buscar todos los contenidos por autor ir a la <b>Sección Buscar todos los contenidos por autor.</b></li> <li>Buscar todos los contenidos por clasificación ir a la <b>Sección Buscar todos los contenidos por clasificación.</b></li> <li>Buscar todas las categorías ir a la <b>Sección Buscar todas las categorías.</b></li> <li>Buscar contenido por ID ir a la <b>Sección Buscar contenido por ID.</b></li> <li>Buscar contenido con arreglo de datos ir a la <b>Sección Buscar contenido con arreglo de datos.</b></li> <li>Buscar todos los contenidos disponibles por autor ir a la <b>Sección Buscar todos los contenidos disponibles por autor.</b></li> <li>Buscar todos los contenidos disponibles por proveedor ir a la <b>Sección Buscar todos los contenidos disponibles por proveedor.</b></li> <li>Buscar todos los contenidos</li> </ul>

## Anexo I: Descripción de los Casos de Uso

	<p>disponibles por clasificación ir a la <b>Sección Buscar todos los contenidos disponibles por clasificación.</b></p> <p>Buscar todas las categorías disponibles ir a la <b>Sección Buscar todas las categorías disponibles.</b></p>
<b>Sección Buscar todos los contenidos por categoría</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	3. Devuelve un listado con todos los contenidos de una categoría dada almacenados en el sistema.
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	3.1 Muestra un mensaje de error si la operación no puede realizarse.
<b>Sección Buscar todos los contenidos por proveedor</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	3. Devuelve un listado con todos los contenidos de un proveedor dado almacenados en el sistema.
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	3.1 Muestra un mensaje de error si la operación no puede realizarse.
<b>Sección Buscar todos los contenidos por autor</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	3. Devuelve un listado con todos los contenidos de un autor dado almacenados en el sistema.
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	3.1 Muestra un mensaje de error si la operación no puede realizarse.
<b>Sección Buscar todos los contenidos por clasificación</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	3. Devuelve un listado con todos los contenidos de una clasificación dada almacenados en el sistema.
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	3.1 Muestra un mensaje de error si la operación no puede realizarse.
<b>Sección Buscar todas las categorías</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	3. Devuelve un listado con todas las categorías almacenadas en el sistema.
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	3.2 Muestra un mensaje de error si la

## Anexo I: Descripción de los Casos de Uso

	operación no puede realizarse.
<b>Sección Buscar contenido por ID</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	3. Devuelve los datos de un contenido almacenado en el sistema.
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	3.1 Muestra un mensaje de error si la operación no puede realizarse.
<b>Sección Buscar contenido con arreglo de datos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	3. Busca el contenido según el identificador del mismo.
	4. Busca el archivo del contenido asociado en el repositorio.
	5. Obtiene el arreglo de datos del archivo asociado.
	6. Devuelve los datos del contenido junto al arreglo de datos obtenidos del archivo asociado en el repositorio.
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	3.1 Muestra un mensaje de error si la operación no puede realizarse.
	4.1 Muestra un mensaje de error si la operación no puede realizarse.
	5. Almacena un log con el identificador del archivo que no se encontró en el repositorio.
<b>Sección Buscar todos los contenidos disponibles por autor</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	3. Devuelve un listado de todos los contenidos disponibles de un autor dado almacenados en el sistema.
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	3.1 Muestra un mensaje de error si la operación no puede realizarse.
<b>Sección Buscar todos los contenidos disponibles por proveedor</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	3. Devuelve un listado de todos los contenidos disponibles de un proveedor dado almacenados en el sistema.
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	3.1 Muestra un mensaje de error si la operación no puede realizarse.
<b>Sección Buscar todos los contenidos disponibles por clasificación</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	3. Devuelve un listado de todos los

## Anexo I: Descripción de los Casos de Uso

	contenidos disponibles de una clasificación dada almacenados en el sistema
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	3.1 Muestra un mensaje de error si la operación no puede realizarse.
<b>Sección Buscar todas las categorías disponibles</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	3. Devuelve un listado de todas las categorías disponibles almacenadas en el sistema.
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	3.1 Muestra un mensaje de error si la operación no puede realizarse.
<b>Poscondiciones</b>	

### CU Buscar Autor

<b>Caso de Uso:</b>	Buscar Autor
<b>Actores:</b>	Externo
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
4. El M_CMS solicita al sistema una de las siguientes opciones: Buscar todos los autores Buscar autor por ID	5. Si solicita Buscar todos los autores ir a la <b>Sección Buscar todos los autores.</b> Buscar autor por ID ir a la <b>Sección Buscar autor por ID.</b>
<b>Sección Buscar todos los autores</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	6. Devuelve un listado con todos los autores almacenados en el sistema.
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	6.1 Muestra un mensaje de error si la operación no puede realizarse.
<b>Sección Buscar autor por ID</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	3. Devuelve los datos de un autor almacenado en el sistema.
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	3.1 Muestra un mensaje de error si la operación no puede realizarse.
<b>Poscondiciones</b>	

### CU Buscar País

<b>Caso de Uso:</b>	Buscar País
---------------------	-------------

## Anexo I: Descripción de los Casos de Uso

<b>Actores:</b>	Externo
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. El M_CMS solicita al sistema una de las siguientes opciones: Buscar todos los países. Buscar país por ID.	2. Si solicita Buscar todos los países ir a la <b>Sección Buscar todos los países.</b> Buscar país por ID ir a la <b>Sección Buscar país por ID.</b>
<b>Sección Buscar todos los países</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	3. Devuelve un listado con todos los países almacenados en el sistema.
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	3.1 Muestra un mensaje de error si la operación no puede realizarse.
<b>Sección Buscar país por ID</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	3. Devuelve los datos de un país almacenado en el sistema.
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	3.1 Muestra un mensaje de error si la operación no puede realizarse.
<b>Poscondiciones</b>	

### CU Buscar Proveedor

<b>Caso de Uso:</b>	Buscar Proveedor
<b>Actores:</b>	Externo
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. El M_CMS solicita al sistema una de las siguientes opciones: Buscar todos los proveedores. Buscar proveedor por ID. Buscar todos los proveedores por país	2. Si solicita Buscar todos los proveedores ir a la <b>Sección Buscar todos los proveedores.</b> Buscar proveedor por ID ir a la <b>Sección Buscar proveedor por ID.</b> <b>Buscar todos los proveedores por país</b> ir a la <b>Sección Buscar todos los proveedores por país</b>
<b>Sección Buscar todos los proveedores</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	3. Devuelve un listado con todos los autores almacenados en el sistema.
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	3.1 Muestra un mensaje de error si la operación no puede realizarse.
<b>Sección Buscar proveedor por ID</b>	

## Anexo I: Descripción de los Casos de Uso

Acción del Actor		Respuesta del Sistema
		3. Devuelve los datos de un proveedor almacenado en el sistema.
Flujos Alternos		
Acción del Actor		Respuesta del Sistema
		3.1 Muestra un mensaje de error si la operación no puede realizarse.
Sección Buscar todos los proveedores por país		
Acción del Actor		Respuesta del Sistema
		3. Devuelve un listado con todos los proveedores de un país dado almacenado en el sistema.
Flujos Alternos		
Acción del Actor		Respuesta del Sistema
		3.1 Muestra un mensaje de error si la operación no puede realizarse.
Poscondiciones		

### CU Buscar Clasificación

<b>Caso de Uso:</b>	Buscar Clasificación	
<b>Actores:</b>	Externo	
Flujo Normal de Eventos		
Acción del Actor		Respuesta del Sistema
1. El M_CMS solicita al sistema una de las siguientes opciones: Buscar todas las clasificaciones. Buscar clasificación por ID.		2. Si solicita Buscar todos las clasificaciones ir a la <b>Sección Buscar todos los clasificaciones.</b> Buscar clasificación por ID ir a la <b>Sección Buscar clasificación por ID.</b>
Sección Buscar todas las clasificaciones		
Acción del Actor		Respuesta del Sistema
		3. Devuelve un listado con todas las clasificaciones almacenadas en el sistema.
Flujos Alternos		
Acción del Actor		Respuesta del Sistema
		3.1 Muestra un mensaje de error si la operación no puede realizarse.
Sección Buscar clasificación por ID		
Acción del Actor		Respuesta del Sistema
		3. Devuelve los datos de una clasificación.
Flujos Alternos		
Acción del Actor		Respuesta del Sistema
		3.1 Muestra un mensaje de error si la operación no puede realizarse.
Poscondiciones		

### CU Buscar Orden

## Anexo I: Descripción de los Casos de Uso

<b>Caso de Uso:</b>	Buscar Orden
<b>Actores:</b>	M_ContentDelivery
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El M_CMS solicita al sistema una de las siguientes opciones: Buscar todas las órdenes. Buscar orden por ID.	2. Si solicita Buscar todos las ordenes ir a la <b>Sección Buscar todos los órdenes.</b> Buscar orden por ID ir a la <b>Sección Buscar orden por ID.</b>
Sección Buscar todas las órdenes	
Acción del Actor	Respuesta del Sistema
	3. Devuelve un listado con todas las órdenes almacenadas en el sistema.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	3.1 Muestra un mensaje de error si la operación no puede realizarse.
Sección Buscar orden por ID	
Acción del Actor	Respuesta del Sistema
	3. Devuelve los datos de una orden almacenada en el sistema.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	3.1 Muestra un mensaje de error si la operación no puede realizarse.
<b>Poscondiciones</b>	

### CU Obtener Contenido

<b>Caso de Uso:</b>	Obtener Contenido
<b>Actores:</b>	M_ContentDelivery (inicia), M_Transcoder
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
6. El M_CMS solicita al sistema una de las siguientes opciones: Obtener ID del contenido. Obtener contenido. Obtener archivo de la carpeta caché	7. Si solicita Obtener ID del contenido ir a la <b>Sección Obtener ID del contenido.</b> Obtener contenido ir a la <b>Sección Obtener contenido.</b> Obtener archivo de la carpeta caché ir a la <b>Sección Obtener archivo de la carpeta caché.</b>
Sección Obtener ID del contenido	
Acción del Actor	Respuesta del Sistema
	8. Busca el contenido según el id del mismo.
	9. Busca el nombre del archivo asociado en la carpeta caché.
	10. Obtiene el archivo en la carpeta caché.
	11. Devuelve el ID del contenido en la carpeta caché.

## Anexo I: Descripción de los Casos de Uso

<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	7.1 Muestra un mensaje de error si la operación no puede realizarse.
	8.1 Muestra un mensaje de error si la operación no puede realizarse.
	10.1. Si el archivo no existe en la carpeta caché, solicita la transformación del contenido al M_Transcoder.
	10.2. Devuelve el id del contenido en la carpeta caché.
	10.3. Muestra un mensaje de error si la operación no puede realizarse.
<b>Sección Obtener contenido</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	3. Obtiene el id del contenido asociado al id del contenido en la carpeta caché.
	4. Busca el contenido según el id obtenido.
	5. Devuelve el contenido.
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	3.1 Muestra un mensaje de error si la operación no puede realizarse.
	4.1 Muestra un mensaje de error si la operación no puede realizarse.
<b>Sección Obtener archivo de la carpeta caché</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	3. Devuelve un archivo de la carpeta caché.
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	3.1 Muestra un mensaje de error si la operación no puede realizarse.
	3.2 Almacena el identificador del archivo que no se encontró en la carpeta caché.
<b>Poscondiciones</b>	

## Anexo II: Descripción de las clases del diseño

Clase del Diseño ContentService <Interface>

Atributos	Tipo
Operaciones	Descripción
+getContentId (order :Order) :String	Hace referencia al método getContentId() de la clase <b>ContentServiceImpl</b> .
+getContent (cacheId : String) :Content	Hace referencia al método getContent() de la clase <b>ContentServiceImpl</b> .
+getFileFromCache (cacheId :String) :File	Hace referencia al método getFileFromCache() de la clase <b>ContentServiceImpl</b> .
+updateAuthor (author : Author) :void	Hace referencia al método updateAuthor() de la clase <b>ContentServiceImpl</b> .
+removeAuthor (authorId : Long) :void	Hace referencia al método removeAuthor() de la clase <b>ContentServiceImpl</b> .
+findAuthorById (authorId : Long) : Author	Hace referencia al método findAuthorById() de la clase <b>ContentServiceImpl</b> .
+saveContentClassification (classification : Classification) : void	Hace referencia al método saveContentClassification() de la clase <b>ContentServiceImpl</b> .
+updateContentClassification (classification : Classification) : void	Hace referencia al método updateContentClassification() de la clase <b>ContentServiceImpl</b> .
+removeContentClassification (classificationId : Long) : void	Hace referencia al método removeContentClassification() de la clase <b>ContentServiceImpl</b> .
+findClassificationById (classificationId : Long) :Classification	Hace referencia al método findClassificationById() de la clase <b>ContentServiceImpl</b> .
+findAllClassifications ( ) :Collection<Classification>	Hace referencia al método findAllClassifications() de la clase <b>ContentServiceImpl</b> .
+saveContent (content : Content) : void	Hace referencia al método saveContent() de la clase <b>ContentServiceImpl</b> .
+updateContent (content : Content) : void	Hace referencia al método updateContent() de la clase <b>ContentServiceImpl</b> .
+removeContent (contentId : Long) : void	Hace referencia al método removeContent() de la clase <b>ContentServiceImpl</b> .
+findContentById (contentId : Long) : Content	Hace referencia al método findContentById() de la clase <b>ContentServiceImpl</b> .
+findContentWithByteArrayById (contentId : Long) : Content	Hace referencia al método findContentWithByteArrayById() de la clase <b>ContentServiceImpl</b> .
+findAllContentByCategory (contentId : Long) :Collection<Content>	Hace referencia al método findAllContentByCategory() de la clase <b>ContentServiceImpl</b> .
+findAllContentByCategory (contentId : Long) :Collection<Content>	Hace referencia al método findAllContentByCategory() de la clase <b>ContentServiceImpl</b> .
+findAllAvailableContentFather ( ) :Collection<Content>	Hace referencia al método findAllAvailableContentFather() de la clase <b>ContentServiceImpl</b> .
+findAllContentFather ( ) :Collection<Content>	Hace referencia al método findAllContentFather() de la clase <b>ContentServiceImpl</b> .
+saveCountry	Hace referencia al método saveCountry() de la clase

## Anexo II: Descripción de las Clases del Diseño

(country : Country) : void	<b>ContentServiceImpl.</b>
+updateCountry (country : Country) : void	Hace referencia al método updateCountry() de la clase <b>ContentServiceImpl.</b>
+removeCountry (countryId : Long) : void	Hace referencia al método removeCountry() de la clase <b>ContentServiceImpl.</b>
+findCountryById (countryId : Long) : Country	Hace referencia al método findCountryById() de la clase <b>ContentServiceImpl.</b>
+findAllCountries ( ) :Collection<Country>	Hace referencia al método findAllCountries() de la clase <b>ContentServiceImpl.</b>
+saveOrder (order :Order) : Long	Hace referencia al método saveOrder() de la clase <b>ContentServiceImpl.</b>
+updateOrder (order :Order) : void	Hace referencia al método updateOrder() de la clase <b>ContentServiceImpl.</b>
+removeOrder (orderId : Long) : void	Hace referencia al método removeOrder() de la clase <b>ContentServiceImpl.</b>
+findOrderById (orderId : Long) : Order	Hace referencia al método findOrderById() de la clase <b>ContentServiceImpl.</b>
+findAllOrders ( ) :Collection<Order>	Hace referencia al método findAllOrders() de la clase <b>ContentServiceImpl.</b>
+saveProvider (provider : Provider) : void	Hace referencia al método saveProvider() de la clase <b>ContentServiceImpl.</b>
+updateProvider (provider : Provider) : void	Hace referencia al método updateProvider() de la clase <b>ContentServiceImpl.</b>
+removeProvider (providerId : Long) : void	Hace referencia al método removeProvider() de la clase <b>ContentServiceImpl.</b>
+findProviderById (providerId : Long) : Provider	Hace referencia al método findProviderById() de la clase <b>ContentServiceImpl.</b>
+findAllProviders ( ) :Collection<Provider>	Hace referencia al método findAllProviders() de la clase <b>ContentServiceImpl.</b>
+findAllAuthors ( ) : Collection<Author>	Hace referencia al método findAllAuthors() de la clase <b>ContentServiceImpl.</b>
+increasingNumberOfDownloads (contentId : Long) :void	Hace referencia al método increasingNumberOfDownloads() de la clase <b>ContentServiceImpl.</b>
+findAllProvidersByCountryId (countryId : Long) :Collection<Provider>	Hace referencia al método findAllProvidersByCountryId() de la clase <b>ContentServiceImpl.</b>
+findAllAvailableContentByProvider (providerId : Long) : Collection<Content>	Hace referencia al método findAllAvailableContentByProvider() de la clase <b>ContentServiceImpl.</b>
+findAllContentByProvider (providerId : Long) :Collection<Content>	Hace referencia al método findAllContentByProvider() de la clase <b>ContentServiceImpl.</b>
+findAllAvailableContentBy Classification (classificationId : Long) :Collection<Content>	Hace referencia al método findAllAvailableContentByClassification() de la clase <b>ContentServiceImpl.</b>
+findAllContentByClassification (classificationId : Long) :Collection<Content>	Hace referencia al método findAllContentByClassification() de la clase <b>ContentServiceImpl.</b>
+findAllAvailableContentByAuthor	Hace referencia al método

## Anexo II: Descripción de las Clases del Diseño

(authorId : Long) :Collection<Content>	findAllAvailableContentByAuthor() de la clase <b>ContentServiceImpl</b> .
+findAllContentByAuthor (authorId : Long) :Collection<Content>	Hace referencia al método findAllContentByAuthor() de la clase <b>ContentServiceImpl</b> .

Clase del Diseño ContentWebService

Atributos	Tipo
-contentService	ContentServiceImpl
Operaciones	Descripción
+getContentId (order :Order) :String	Hace referencia al método getContentId() de la clase <b>ContentServiceImpl</b> .
+getContent (cacheId : String) :Content	Hace referencia al método getContent() de la clase <b>ContentServiceImpl</b> .
+getFileFromCache (cacheId :String) :File	Hace referencia al método getFileFromCache() de la clase <b>ContentServiceImpl</b> .
+updateAuthor (author : Author) :void	Hace referencia al método updateAuthor() de la clase <b>ContentServiceImpl</b> .
+removeAuthor (authorId : Long) :void	Hace referencia al método removeAuthor() de la clase <b>ContentServiceImpl</b> .
+findAuthorById (authorId : Long) : Author	Hace referencia al método findAuthorById() de la clase <b>ContentServiceImpl</b> .
+saveContentClassification (classification : Classification) : void	Hace referencia al método saveContentClassification() de la clase <b>ContentServiceImpl</b> .
+updateContentClassification (classification : Classification) : void	Hace referencia al método updateContentClassification() de la clase <b>ContentServiceImpl</b> .
+removeContentClassification (classificationId : Long) : void	Hace referencia al método removeContentClassification() de la clase <b>ContentServiceImpl</b> .
+findClassificationById (classificationId : Long) :Classification	Hace referencia al método findClassificationById() de la clase <b>ContentServiceImpl</b> .
+findAllClassifications () :Collection<Classification>	Hace referencia al método findAllClassifications() de la clase <b>ContentServiceImpl</b> .
+saveContent (content : Content) : void	Hace referencia al método saveContent() de la clase <b>ContentServiceImpl</b> .
+updateContent (content : Content) : void	Hace referencia al método updateContent() de la clase <b>ContentServiceImpl</b> .
+removeContent (contentId : Long) : void	Hace referencia al método removeContent() de la clase <b>ContentServiceImpl</b> .
+findContentById (contentId : Long) : Content	Hace referencia al método findContentById() de la clase <b>ContentServiceImpl</b> .
+findContentWithByteArrayById (contentId : Long) : Content	Hace referencia al método findContentWithByteArrayById() de la clase <b>ContentServiceImpl</b> .
+findAllContentByCategory (contentId : Long) :Collection<Content>	Hace referencia al método findAllContentByCategory() de la clase <b>ContentServiceImpl</b> .
+findAllContentByCategory (contentId : Long) :Collection<Content>	Hace referencia al método findAllContentByCategory() de la clase <b>ContentServiceImpl</b> .
+findAllAvailableContentFather	Hace referencia al método findAllAvailableContentFather()

## Anexo II: Descripción de las Clases del Diseño

() :Collection<Content>	de la clase <b>ContentServiceImpl</b> .
+findAllContentFather () :Collection<Content>	Hace referencia al método findAllContentFather() de la clase <b>ContentServiceImpl</b> .
+saveCountry (country : Country) : void	Hace referencia al método saveCountry() de la clase <b>ContentServiceImpl</b> .
+updateCountry (country : Country) : void	Hace referencia al método updateCountry() de la clase <b>ContentServiceImpl</b> .
+removeCountry (countryId : Long) : void	Hace referencia al método removeCountry() de la clase <b>ContentServiceImpl</b> .
+findCountryById (countryId : Long) : Country	Hace referencia al método findCountryById() de la clase <b>ContentServiceImpl</b> .
+findAllCountries () :Collection<Country>	Hace referencia al método findAllCountries() de la clase <b>ContentServiceImpl</b> .
+saveOrder (order :Order) : Long	Hace referencia al método saveOrder() de la clase <b>ContentServiceImpl</b> .
+updateOrder (order :Order) : void	Hace referencia al método updateOrder() de la clase <b>ContentServiceImpl</b> .
+removeOrder (orderId : Long) : void	Hace referencia al método removeOrder() de la clase <b>ContentServiceImpl</b> .
+findOrderById (orderId : Long) : Order	Hace referencia al método findOrderById() de la clase <b>ContentServiceImpl</b> .
+findAllOrders () :Collection<Order>	Hace referencia al método findAllOrders() de la clase <b>ContentServiceImpl</b> .
+saveProvider (provider : Provider) : void	Hace referencia al método saveProvider() de la clase <b>ContentServiceImpl</b> .
+updateProvider (provider : Provider) : void	Hace referencia al método updateProvider() de la clase <b>ContentServiceImpl</b> .
+removeProvider (providerId : Long) : void	Hace referencia al método removeProvider() de la clase <b>ContentServiceImpl</b> .
+findProviderById (providerId : Long) : Provider	Hace referencia al método findProviderById() de la clase <b>ContentServiceImpl</b> .
+findAllProviders () :Collection<Provider>	Hace referencia al método findAllProviders() de la clase <b>ContentServiceImpl</b> .
+findAllAuthors () : Collection<Author>	Hace referencia al método findAllAuthors() de la clase <b>ContentServiceImpl</b> .
+increasingNumberOfDownloads (contentId : Long) :void	Hace referencia al método increasingNumberOfDownloads() de la clase <b>ContentServiceImpl</b> .
+findAllProvidersByCountryId (countryId : Long) :Collection<Provider>	Hace referencia al método findAllProvidersByCountryId() de la clase <b>ContentServiceImpl</b> .
+findAllAvailableContentByProvider (providerId : Long) : Collection<Content>	Hace referencia al método findAllAvailableContentByProvider() de la clase <b>ContentServiceImpl</b> .
+findAllContentByProvider (providerId : Long) :Collection<Content>	Hace referencia al método findAllContentByProvider() de la clase <b>ContentServiceImpl</b> .
+findAllAvailableContentBy Classification (classificationId : Long) :Collection<Content>	Hace referencia al método findAllAvailableContentByClassification() de la clase <b>ContentServiceImpl</b> .

## Anexo II: Descripción de las Clases del Diseño

+findAllContentByClassification (classificationId : Long) :Collection<Content>	Hace referencia al método findAllContentByClassification() de la clase <b>ContentServiceImpl</b> .
+findAllAvailableContentByAuthor (authorId : Long) :Collection<Content>	Hace referencia al método findAllAvailableContentByAuthor() de la clase <b>ContentServiceImpl</b> .
+findAllContentByAuthor (authorId : Long) :Collection<Content>	Hace referencia al método findAllContentByAuthor() de la clase <b>ContentServiceImpl</b> .

Clase del Diseño ContentServiceImpl

Atributos	Tipo
- contentSource	ContentSource
Operaciones	Descripción
+getContentId (order :Order) :String	Dada una orden de descarga, que contiene el id del contenido solicitado y el user agent del cliente, verifica que el archivo del contenido existe en la carpeta caché y devuelve el id del contenido en la misma. Si el archivo del contenido existente en el repositorio no corresponde al user agent del cliente en la orden de descarga, solicita la transformación del mismo y devuelve el id del archivo contenido en la carpeta caché una vez transformado y almacenado en la misma.
+getContent (cacheId : String) :Content	Dado el id de un contenido en la carpeta caché busca el contenido asociado y devuelve el mismo.
+getFileFromCache (cacheId :String) :File	Hace referencia al método getFileFromCache() de la clase <b>ContentSource&lt;Interface&gt;</b> .
+updateAuthor (author : Author) :void	Hace referencia al método updateAuthor() de la clase <b>ContentSource&lt;Interface&gt;</b> .
+removeAuthor (authorId : Long) :void	Hace referencia al método removeAuthor() de la clase <b>ContentSource&lt;Interface&gt;</b> .
+findAuthorById (authorId : Long) : Author	Hace referencia al método findAuthorById() de la clase <b>ContentSource&lt;Interface&gt;</b> .
+saveContentClassification (classification : Classification) : void	Hace referencia al método saveContentClassification() de la clase <b>ContentSource&lt;Interface&gt;</b> .
+updateContentClassification (classification : Classification) : void	Hace referencia al método updateContentClassification() de la clase <b>ContentSource&lt;Interface&gt;</b> .
+removeContentClassification (classificationId : Long) : void	Hace referencia al método removeContentClassification() de la clase <b>ContentSource&lt;Interface&gt;</b> .
+findClassificationById (classificationId : Long) :Classification	Hace referencia al método findClassificationById() de la clase <b>ContentSource&lt;Interface&gt;</b> .
+findAllClassifications ( ) :Collection<Classification>	Hace referencia al método findAllClassifications() de la clase <b>ContentSource&lt;Interface&gt;</b> .
+saveContent (content : Content) : void	Hace referencia al método saveContent() de la clase <b>ContentSource&lt;Interface&gt;</b> .
+updateContent (content : Content) : void	Hace referencia al método updateContent() de la clase <b>ContentSource&lt;Interface&gt;</b> .
+removeContent (contentId : Long) : void	Hace referencia al método removeContent() de la clase <b>ContentSource&lt;Interface&gt;</b> .

## Anexo II: Descripción de las Clases del Diseño

+findContentById (contentId : Long) : Content	Hace referencia al método findContentById() de la clase <b>ContentSource&lt;Interface&gt;</b> .
+findContentWithByteArrayById (contentId : Long) : Content	Hace referencia al método findContentWithByteArrayById() de la clase <b>ContentSource&lt;Interface&gt;</b> .
+findAllContentByCategory (contentId : Long) :Collection<Content>	Hace referencia al método findAllContentByCategory() de la clase <b>ContentSource&lt;Interface&gt;</b> .
+findAllContentByCategory (contentId : Long) :Collection<Content>	Hace referencia al método findAllContentByCategory() de la clase <b>ContentSource&lt;Interface&gt;</b> .
+findAllAvailableContentFather ( ) :Collection<Content>	Hace referencia al método findAllAvailableContentFather() de la clase <b>ContentSource&lt;Interface&gt;</b> .
+findAllContentFather ( ) :Collection<Content>	Hace referencia al método findAllContentFather() de la clase <b>ContentSource&lt;Interface&gt;</b> .
+saveCountry (country : Country) : void	Hace referencia al método saveCountry() de la clase <b>ContentSource&lt;Interface&gt;</b> .
+updateCountry (country : Country) : void	Hace referencia al método updateCountry() de la clase <b>ContentSource&lt;Interface&gt;</b> .
+removeCountry (countryId : Long) : void	Hace referencia al método removeCountry() de la clase <b>ContentSource&lt;Interface&gt;</b> .
+findCountryById (countryId : Long) : Country	Hace referencia al método findCountryById() de la clase <b>ContentSource&lt;Interface&gt;</b> .
+findAllCountries ( ) :Collection<Country>	Hace referencia al método findAllCountries() de la clase <b>ContentSource&lt;Interface&gt;</b> .
+saveOrder (order :Order) : Long	Hace referencia al método saveOrder() de la clase <b>ContentSource&lt;Interface&gt;</b> .
+updateOrder (order :Order) : void	Hace referencia al método updateOrder() de la clase <b>ContentSource&lt;Interface&gt;</b> .
+removeOrder (orderId : Long) : void	Hace referencia al método removeOrder() de la clase <b>ContentSource&lt;Interface&gt;</b> .
+findOrderById (orderId : Long) : Order	Hace referencia al método findOrderById() de la clase <b>ContentSource&lt;Interface&gt;</b> .
+findAllOrders ( ) :Collection<Order>	Hace referencia al método findAllOrders() de la clase <b>ContentSource&lt;Interface&gt;</b> .
+saveProvider (provider : Provider) : void	Hace referencia al método saveProvider() de la clase <b>ContentSource&lt;Interface&gt;</b> .
+updateProvider (provider : Provider) : void	Hace referencia al método updateProvider() de la clase <b>ContentSource&lt;Interface&gt;</b> .
+removeProvider (providerId : Long) : void	Hace referencia al método removeProvider() de la clase <b>ContentSource&lt;Interface&gt;</b> .
+findProviderById (providerId : Long) : Provider	Hace referencia al método findProviderById() de la clase <b>ContentSource&lt;Interface&gt;</b> .
+findAllProviders ( ) :Collection<Provider>	Hace referencia al método findAllProviders() de la clase <b>ContentSource&lt;Interface&gt;</b> .
+findAllAuthors ( ) : Collection<Author>	Hace referencia al método findAllAuthors() de la clase <b>ContentSource&lt;Interface&gt;</b> .
+increasingNumberOfDownloads (contentId : Long) :void	Incrementa en uno el atributo número de descargas de n contenido luego que se entrega el mismo al cliente.
+findAllProvidersByCountryId (countryId : Long)	Hace referencia al método findAllProvidersByCountryId() de la clase <b>ContentSource&lt;Interface&gt;</b> .

## Anexo II: Descripción de las Clases del Diseño

:Collection<Provider>	
+findAllAvailableContentByProvider (providerId : Long) : Collection<Content>	Hace referencia al método findAllAvailableContentByProvider() de la clase <b>ContentSource&lt;Interface&gt;</b> .
+findAllContentByProvider (providerId : Long) :Collection<Content>	Hace referencia al método findAllContentByProvider() de la clase <b>ContentSource&lt;Interface&gt;</b> .
+findAllAvailableContentBy Classification (classificationId : Long) :Collection<Content>	Hace referencia al método findAllAvailableContentByClassification() de la clase <b>ContentSource&lt;Interface&gt;</b> .
+findAllContentByClassification (classificationId : Long) :Collection<Content>	Hace referencia al método findAllContentByClassification() de la clase <b>ContentSource&lt;Interface&gt;</b> .
+findAllAvailableContentByAuthor (authorId : Long) :Collection<Content>	Hace referencia al método findAllAvailableContentByAuthor() de la clase <b>ContentSource&lt;Interface&gt;</b> .
+findAllContentByAuthor (authorId : Long) :Collection<Content>	Hace referencia al método findAllContentByAuthor() de la clase <b>ContentSource&lt;Interface&gt;</b> .
+getContentSource ( ) :ContentSource	Devuelve el atributo contentSource de la clase.
+setContentSource (contentSource :ContentSource) : void	Cambia el atributo contentSource de la clase por un nuevo objeto de tipo ContentSource.

### Clase del Diseño ContentSource<Interface>

Atributos	Tipo
Operaciones	Descripción
+getFileFromCache (cacheId :String) :File	Hace referencia al método getFileFromCache() de la clase <b>UniqueContentSource</b> .
+updateAuthor (author : Author) :void	Hace referencia al método updateAuthor() de la clase <b>UniqueContentSource</b> .
+saveAuthor (author : Author) : void	Hace referencia al método saveAuthor() de la clase <b>UniqueContentSource</b> .
+removeAuthor (authorId : Long) : void	Hace referencia al método removeAuthor() de la clase <b>UniqueContentSource</b> .
+findAuthorById (authorId : Long) : Author	Hace referencia al método findAuthorById() de la clase <b>UniqueContentSource</b> .
+saveContentClassification (classification : Classification) : void	Hace referencia al método saveContentClassification() de la clase <b>UniqueContentSource</b> .
+updateContentClassification (classification : Classification) : void	Hace referencia al método updateContentClassification() de la clase <b>UniqueContentSource</b> .
+removeContentClassification (classificationId : Long) : void	Hace referencia al método removeContentClassification() de la clase <b>UniqueContentSource</b> .
+findClassificationById (classificationId : Long) : Classification	Hace referencia al método findClassificationById() de la clase <b>UniqueContentSource</b> .
+findAllClassifications	Hace referencia al método findAllClassifications() de la

## Anexo II: Descripción de las Clases del Diseño

() : Collection<Classification>	clase <b>UniqueContentSource</b> .
+saveContent (content : Content) : void	Hace referencia al método saveContent() de la clase <b>UniqueContentSource</b> .
+updateContent (content : Content) : void	Hace referencia al método updateContent() de la clase <b>UniqueContentSource</b> .
+removeContent (contentId : Long) : void	Hace referencia al método removeContent() de la clase <b>UniqueContentSource</b> .
+findContentById (contentId : Long) : Content	Hace referencia al método findContentById() de la clase <b>UniqueContentSource</b> .
+findContentWithByteArrayById (contentId : Long) : Content	Hace referencia al método findContentWithByteArrayById() de la clase <b>UniqueContentSource</b> .
+findAllContentByCategory (contentId : Long) :Collection<Content>	Hace referencia al método findAllContentByCategory() de la clase <b>UniqueContentSource</b> .
+findAllAvailableContentByCategory (contentId : Long) :Collection<Content>	Hace referencia al método findAllAvailableContentByCategory() de la clase <b>UniqueContentSource</b> .
+findAllAvailableContentFather ( ) :Collection<Content>	Hace referencia al método findAllAvailableContentFather() de la clase <b>UniqueContentSource</b> .
+findAllContentFather ( ) : Collection<Content>	Hace referencia al método findAllContentFather() de la clase <b>UniqueContentSource</b> .
+saveCountry (country : Country) : void	Hace referencia al método saveCountry() de la clase <b>UniqueContentSource</b> .
+updateCountry (country : Country) : void	Hace referencia al método updateCountry() de la clase <b>UniqueContentSource</b> .
+removeCountry (countryId : Long) : void	Hace referencia al método removeCountry() de la clase <b>UniqueContentSource</b> .
+findCountryById (countryId : Long) : Country	Hace referencia al método findCountryById() de la clase <b>UniqueContentSource</b> .
+findAllCountries ( ) : Collection<Country>	Hace referencia al método findAllCountries() de la clase <b>UniqueContentSource</b> .
+saveOrder (order :Order) : Long	Hace referencia al método saveOrder() de la clase <b>UniqueContentSource</b> .
+updateOrder (order :Order) : void	Hace referencia al método updateOrder() de la clase <b>UniqueContentSource</b> .
+removeOrder (orderId : Long) : void	Hace referencia al método removeOrder() de la clase <b>UniqueContentSource</b> .
+findOrderById (orderId : Long) : Order	Hace referencia al método findOrderById() de la clase <b>UniqueContentSource</b> .
+findAllOrders ( ) :Collection<Order>	Hace referencia al método findAllOrders() de la clase <b>UniqueContentSource</b> .
+saveProvider (provider : Provider) : void	Hace referencia al método saveProvider() de la clase <b>UniqueContentSource</b> .
+updateProvider (provider : Provider) : void	Hace referencia al método updateProvider() de la clase <b>UniqueContentSource</b> .
+removeProvider (providerId : Long) : void	Hace referencia al método removeProvider() de la clase <b>UniqueContentSource</b> .
+findProviderById (providerId : Long) : Provider	Hace referencia al método findProviderById() de la clase <b>UniqueContentSource</b> .

## Anexo II: Descripción de las Clases del Diseño

+findAllProviders ( ) : Collection<Provider>	Hace referencia al método findAllProviders() de la clase <b>UniqueContentSource</b> .
+findURLFileById (contentId : String) : String	Hace referencia al método findURLFileById() de la clase <b>UniqueContentSource</b> .
+getFileName (cacheId : String) : String	Hace referencia al método getFileName() de la clase <b>UniqueContentSource</b> .
+getCacheFileName (contentId : String, UA : String) : String	Hace referencia al método getCacheFileName() de la clase <b>UniqueContentSource</b> .
+getContentIdAssociatedToACacheId (cacheId : String) : String	Hace referencia al método getContentIdAssociatedToACacheId() de la clase <b>UniqueContentSource</b> .
+findAllAuthors ( ) : Collection<Author>	Hace referencia al método findAllAuthors() de la clase <b>UniqueContentSource</b> .
+findAllProvidersByCountryId (countryId : Long) :Collection<Provider>	Hace referencia al método findAllProvidersByCountryId() de la clase <b>UniqueContentSource</b> .
+findAllAvailableContentByProvider (providerId : Long) :Collection<Content>	Hace referencia al método findAllAvailableContentByProvider() de la clase <b>UniqueContentSource</b> .
+findAllContentByProvider (providerId : Long) :Collection<Content>	Hace referencia al método findAllContentByProvider() de la clase <b>UniqueContentSource</b> .
+findAllAvailableContentByClassification (classificationId : Long) :Collection<Content>	Hace referencia al método findAllAvailableContentByClassification() de la clase <b>UniqueContentSource</b> .
+findAllContentByClassification (classificationId : Long) :Collection<Content>	Hace referencia al método findAllContentByClassification() de la clase <b>UniqueContentSource</b> .
+findAllAvailableContentByAuthor (authorId : Long) : Collection<Content>	Hace referencia al método findAllAvailableContentByAuthor() de la clase <b>UniqueContentSource</b> .
+findAllContentByAuthor (authorId : Long) : Collection<Content>	Hace referencia al método findAllContentByAuthor() de la clase <b>UniqueContentSource</b> .

Clase del Diseño UniqueContentSource

Atributos	Tipo
-jpaContentSource	JPAContentSource
Operaciones	Descripción
+getFileFromCache (cacheId :String) :File	Devuelve el archivo de un contenido almacenado en la carpeta caché según el ID del mismo.
+updateAuthor (author : Author) :void	Hace referencia al método updateAuthor() de la clase <b>JPAContentSource &lt;Interface&gt;</b> .
+getJpaContentSource ( ) : JPAContentSource	Devuelve el atributo jpaContentSource de la clase.
+setJpaContentSource (jpaContentSource : JPAContentSource) : void	Cambia el atributo jpaContentSource de la clase por un nuevo objeto de tipo JPAContentSource.

## Anexo II: Descripción de las Clases del Diseño

+findAllAvailableContentByCategory (contentId : Long) : Collection<Content>	Hace referencia al método findAllAvailableContentByCategory() de la clase <b>JPAContentSource &lt;Interface&gt;</b> .
+findAllAvailableContentFather ( ) : Collection<Content>	Hace referencia al método findAllAvailableContentFather() de la clase <b>JPAContentSource &lt;Interface&gt;</b> .
+findAllClassifications ( ) : Collection<Classification>	Hace referencia al método findAllClassifications() de la clase <b>JPAContentSource &lt;Interface&gt;</b> .
+findAllContentByCategory (contentId : Long) :Collection<Content>	Hace referencia al método findAllContentByCategory() de la clase <b>JPAContentSource &lt;Interface&gt;</b> .
+findAllContentFather ( ) : Collection<Content>	Hace referencia al método findAllContentFather() de la clase <b>JPAContentSource &lt;Interface&gt;</b> .
+findAllCountries ( ) : Collection<Country>	Hace referencia al método findAllCountries() de la clase <b>JPAContentSource &lt;Interface&gt;</b> .
+findAllOrders ( ) :Collection<Order>	Hace referencia al método findAllOrders() de la clase <b>JPAContentSource &lt;Interface&gt;</b> .
+findAllProviders ( ) : Collection<Provider>	Hace referencia al método findAllProviders() de la clase <b>JPAContentSource &lt;Interface&gt;</b> .
+findAuthorById (authorId : Long) : Author	Hace referencia al método findAuthorById() de la clase <b>JPAContentSource &lt;Interface&gt;</b> .
+findClassificationById (classificationId : Long) : Classification	Hace referencia al método findClassificationById() de la clase <b>JPAContentSource &lt;Interface&gt;</b> .
+findContentById (contentId : Long) : Content	Hace referencia al método findContentById() de la clase <b>JPAContentSource &lt;Interface&gt;</b> .
+findContentWithByteArrayById (contentId : Long) : Content	Dado el id de un contenido devuelve el contenido; haciendo referencia al método findContentById() de la clase <b>JPAContentSource&lt;Interface&gt;</b> , junto con el arreglo de byte del archivo contenido asociado en la carpeta caché haciendo referencia al método getByteArrayFromFile() de la clase <b>FileContentSource</b> .
+findCountryById (countryId : Long) : Country	Hace referencia al método findCountryById() de la clase <b>JPAContentSource &lt;Interface&gt;</b> .
+findOrderById (orderId : Long) : Order	Hace referencia al método findOrderById() de la clase <b>JPAContentSource &lt;Interface&gt;</b> .
+findProviderById (providerId : Long) : Provider	Hace referencia al método findProviderById() de la clase <b>JPAContentSource &lt;Interface&gt;</b> .
+removeAuthor (authorId : Long) : void	Hace referencia al método removeAuthor() de la clase <b>JPAContentSource &lt;Interface&gt;</b> .
+removeContentHelper (content : Content) : void	Hace referencia al método removeFile() de la clase <b>FileContentSource()</b> para eliminar el archivo de un contenido. Si el contenido representa una categoría hace referencia a la clase findAllContentByCategory() de la clase <b>JPAContentSource</b> para obtener un listado de los contenidos de dicha categoría, recorre el listado y hace referencia al método removeFile() de la clase <b>FileContentSource()</b> para eliminar el archivo de cada contenido.
+removeContent (contentId : Long) : void	Hace referencia al método removeContentHelper(), luego Hace referencia al método removeContent() de la clase <b>JPAContentSource &lt;Interface&gt;</b> .

## Anexo II: Descripción de las Clases del Diseño

+removeContentClassification (classificationId : Long) : void	Hace referencia al método removeContentClassification() de la clase <b>JPAContentSource &lt;Interface&gt;</b> .
+removeCountry (countryId : Long) : void	Hace referencia al método removeCountry() de la clase <b>JPAContentSource &lt;Interface&gt;</b> .
+removeOrder (orderId : Long) : void	Hace referencia al método removeOrder() de la clase <b>JPAContentSource &lt;Interface&gt;</b> .
+removeProvider (providerId : Long) : void	Hace referencia al método removeProvider() de la clase <b>JPAContentSource &lt;Interface&gt;</b> .
+saveAuthor (author : Author) : void	Hace referencia al método saveAuthor() de la clase <b>JPAContentSource &lt;Interface&gt;</b> .
+saveContent (content : Content) : void	Hace referencia al método saveContent() de la clase <b>JPAContentSource &lt;Interface&gt;</b> y luego hace referencia al método saveFile() de la clase <b>FileContentSource</b> para almacenar el archivo contenido asociado en la carpeta caché.
+saveContentClassification (classification : Classification) : void	Hace referencia al método saveContentClassification() de la clase <b>JPAContentSource &lt;Interface&gt;</b> .
+saveCountry (country : Country) : void	Hace referencia al método saveCountry() de la clase <b>JPAContentSource &lt;Interface&gt;</b> .
+saveOrder (order : Order) : Long	Hace referencia al método saveOrder() de la clase <b>JPAContentSource &lt;Interface&gt;</b> .
+saveProvider (provider : Provider) : void	Hace referencia al método saveProvider() de la clase <b>JPAContentSource &lt;Interface&gt;</b> .
+updateContent (content : Content) : void	Hace referencia la método updateFile() de la clase <b>FileContentSource</b> y luego al método saveContent() de la clase <b>JPAContentSource</b> .
+updateContentClassification (classification : Classification) : void	Hace referencia al método updateContentClassification() de la clase <b>JPAContentSource &lt;Interface&gt;</b> .
+updateCountry (country : Country) : void	Hace referencia al método updateCountry() de la clase <b>JPAContentSource &lt;Interface&gt;</b> .
+updateOrder (order :Order) : void	Hace referencia al método updateOrder() de la clase <b>JPAContentSource &lt;Interface&gt;</b> .
+updateProvider (provider : Provider) : void	Hace referencia al método updateProvider() de la clase <b>JPAContentSource &lt;Interface&gt;</b> .
+findURLFileById (contentId : String) : String	Hace referencia al método getFile() de la clase <b>FileContentSource</b> para obtener el archivo contenido dado el id del contenido y devuelve el URL del mismo.
+getFileName (cacheId : String) : String	Hace referencia al método getCacheFile de la clase <b>FileContentSource()</b> y devuelve el nombre del archivo obtenido.
+getCacheFileName (contentId : String, UA : String) : String	Hace referencia al método getCacheFileName() de la clase <b>FileContentSource</b> .
+getContentIdAssociatedToACacheId (cacheId : String) : String	Hace referencia al método getContentIdAssociatedToACacheId() de la clase <b>FileContentSource</b> .
+findAllAuthors ( ) : Collection<Author>	Hace referencia al método findAllAuthors() de la clase <b>JPAContentSource &lt;Interface&gt;</b> .
+findAllProvidersByCountryId (countryId : Long) : Collection<Provider>	Hace referencia al método findAllProvidersByCountryId() de la clase <b>JPAContentSource &lt;Interface&gt;</b> .

## Anexo II: Descripción de las Clases del Diseño

+findAllAvailableContentByAuthor (authorId : Long) : Collection<Content>	Hace referencia al método findAllAvailableContentByAuthor() de la clase <b>JPAContentSource &lt;Interface&gt;</b> .
+findAllAvailableContentBy Classification (classificationId : Long) :Collection<Content>	Hace referencia al método findAllAvailableContentByClassification() de la clase <b>JPAContentSource &lt;Interface&gt;</b> .
+findAllAvailableContentByProvider (providerId : Long) : Collection<Content>	Hace referencia al método findAllAvailableContentByProvider() de la clase <b>JPAContentSource &lt;Interface&gt;</b> .
+findAllContentByAuthor (authorId : Long) : Collection<Content>	Hace referencia al método findAllContentByAuthor() de la clase <b>JPAContentSource &lt;Interface&gt;</b> .
+findAllContentByClassification (classificationId : Long) :Collection<Content>	Hace referencia al método findAllContentByClassification() de la clase <b>JPAContentSource &lt;Interface&gt;</b> .
+findAllContentByProvider (providerId : Long) : Collection<Content>	Hace referencia al método findAllContentByProvider() de la clase <b>JPAContentSource &lt;Interface&gt;</b> .

### Clase del Diseño FileContentSource

Atributos	Tipo
<b>Operaciones</b>	<b>Descripción</b>
+getFile (contentId : String) : File	Devuelve un archivo
+getBytesFromArrayFromFile (cacheId : String) : byte []	Devuelve el arreglo de byte de un archivo.
+removeFile (contentId : String) : Boolean	Elimina un archivo.
+saveFile (contentId : String, file : byte []) : void	Almacena un archivo.
+saveFileFromUrl (address : String, contentId : String) : void	Almacena un archivo desde un URL.
+updateFile (contentId : String, file : byte []) : void	Actualiza un archivo.
+getCacheFile (cacheId : String) : File	Obtiene un archivo dado el identificador en la carpeta caché.
+getCacheFileName (contentId : String, UA : String) : String	Devuelve el nombre de un archivo en la carpeta caché dado el identificador del contenido y el user agent del cliente móvil.
+getContentIdAssociatedToA CacheId (cacheId : String) : String	Devuelve el id de un contenido dado el id del contenidos asociado en la carpeta caché.

### Clase del Diseño JPAContentSource<Interface>

## Anexo II: Descripción de las Clases del Diseño

Atributos	Tipo
<b>Operaciones</b>	<b>Descripción</b>
+saveAuthor (author : Author) : void	Hace referencia al método saveAuthor() de la clase <b>JPAContentSourceImpl</b> .
+updateAuthor (author : Author) : void	Hace referencia al método updateAuthor() de la clase <b>JPAContentSourceImpl</b> .
+removeAuthor (authorId : Long) : void	Hace referencia al método removeAuthor() de la clase <b>JPAContentSourceImpl</b> .
+findAuthorById (authorId : Long) : Author	Hace referencia al método findAuthorById() de la clase <b>JPAContentSourceImpl</b> .
+saveContentClassification (classification : Classification) : void	Hace referencia al método saveContentClassification() de la clase <b>JPAContentSourceImpl</b> .
+updateContentClassification (classification : Classification) : void	Hace referencia al método updateContentClassification() de la clase <b>JPAContentSourceImpl</b> .
+removeContentClassification (classificationId : Long) : void	Hace referencia al método removeContentClassification() de la clase <b>JPAContentSourceImpl</b> .
+findClassificationById (classificationId : Long) : Classification	Hace referencia al método findClassificationById() de la clase <b>JPAContentSourceImpl</b> .
+findAllClassifications ( ) : Collection<Classification>	Hace referencia al método findAllClassifications() de la clase <b>JPAContentSourceImpl</b> .
+saveContent (content : Content) : Long	Hace referencia al método saveContent() de la clase <b>JPAContentSourceImpl</b> .
+updateContent (content : Content) : void	Hace referencia al método updateContent() de la clase <b>JPAContentSourceImpl</b> .
+removeContent (contentId : Long) : void	Hace referencia al método removeContent() de la clase <b>JPAContentSourceImpl</b> .
+findContentById (contentId : Long) : Content	Hace referencia al método findContentById() de la clase <b>JPAContentSourceImpl</b> .
+findAllContentByCategory (contentId : Long) : Collection<Content>	Hace referencia al método findAllContentByCategory() de la clase <b>JPAContentSourceImpl</b> .
+findAllAvailableContentBy Category (contentId : Long) : Collection<Content>	Hace referencia al método findAllAvailableContentByCategory() de la clase <b>JPAContentSourceImpl</b> .
+findAllAvailableContentFather ( ) : Collection<Content>	Hace referencia al método findAllAvailableContentFather() de la clase <b>JPAContentSourceImpl</b> .
+findAllContentFather ( ) : Collection<Content>	Hace referencia al método findAllContentFather() de la clase <b>JPAContentSourceImpl</b> .
+saveCountry (country : Country) : void	Hace referencia al método saveCountry() de la clase <b>JPAContentSourceImpl</b> .
+updateCountry (country : Country) : void	Hace referencia al método updateCountry() de la clase <b>JPAContentSourceImpl</b> .
+removeCountry (countryId : Long) : void	Hace referencia al método removeCountry() de la clase <b>JPAContentSourceImpl</b> .
+findCountryById	Hace referencia al método findCountryById () de la clase

## Anexo II: Descripción de las Clases del Diseño

(countryId : Long) : Country	<b>JPAContentSourceImpl.</b>
+findAllCountries ( ) : Collection<Country>	Hace referencia al método findAllCountries() de la clase <b>JPAContentSourceImpl.</b>
+saveOrder (order : Order) : Long	Hace referencia al método saveOrder() de la clase <b>JPAContentSourceImpl.</b>
+updateOrder (order : Order) : void	Hace referencia al método updateOrder() de la clase <b>JPAContentSourceImpl.</b>
+removeOrder (orderId : Long) : void	Hace referencia al método removeOrder() de la clase <b>JPAContentSourceImpl.</b>
+findOrderById (orderId : Long) : Order	Hace referencia al método findOrderById() de la clase <b>JPAContentSourceImpl.</b>
+findAllOrders ( ) : Collection<Order>	Hace referencia al método findAllOrders() de la clase <b>JPAContentSourceImpl.</b>
+saveProvider (provider : Provider) : void	Hace referencia al método saveProvider() de la clase <b>JPAContentSourceImpl.</b>
+updateProvider (provider : Provider) : void	Hace referencia al método updateProvider() de la clase <b>JPAContentSourceImpl.</b>
+removeProvider (providerId : Long) : void	Hace referencia al método removeProvider() de la clase <b>JPAContentSourceImpl.</b>
+findProviderById (providerId : Long) : Provider	Hace referencia al método findProviderById() de la clase <b>JPAContentSourceImpl.</b>
+findAllProviders ( ) : Collection<Provider>	Hace referencia al método findAllProviders() de la clase <b>JPAContentSourceImpl.</b>
+findAllAuthors ( ) : Collection<Author>	Hace referencia al método findAllAuthors() de la clase <b>JPAContentSourceImpl.</b>
+findAllProvidersByCountryId (countryId : Long) : Collection<Provider>	Hace referencia al método findAllProvidersByCountryId() de la clase <b>JPAContentSourceImpl.</b>
+findAllAvailableContentBy Provider (providerId : Long) : Collection<Content>	Hace referencia al método findAllAvailableContentByProvider() de la clase <b>JPAContentSourceImpl.</b>
+findAllContentByProvider (providerId : Long) : Collection<Content>	Hace referencia al método findAllContentByProvider() de la clase <b>JPAContentSourceImpl.</b>
+findAllAvailableContentBy Classification (classificationId : Long) :Collection<Content>	Hace referencia al método findAllAvailableContentByClassification() de la clase <b>JPAContentSourceImpl.</b>
+findAllContentByClassification (classificationId : Long) :Collection<Content>	Hace referencia al método findAllContentByClassification() de la clase <b>JPAContentSourceImpl.</b>
+findAllAvailableContentBy Author (authorId : Long) : Collection<Content>	Hace referencia al método findAllAvailableContentByAuthor() de la clase <b>JPAContentSourceImpl.</b>
+findAllContentByAuthor (authorId : Long) : Collection<Content>	Hace referencia al método findAllContentByAuthor() de la clase <b>JPAContentSourceImpl.</b>

Clase del Diseño JPAContentSourceImpl

## Anexo II: Descripción de las Clases del Diseño

Atributos	Tipo
-authorDao	AuthorDao
-classificationDao	ClassificationDao
-contentDao	ContentDao
-countryDao	CountryDao
-providerDao	ProviderDao
-orderDao	OrderDao
Operaciones	Descripción
+getAuthorDao ( ) : AuthorDao	Devuelve el atributo authorDao de la clase.
+setAuthorDao (authorDao : AuthorDao) : void	Cambia el atributo authorDao de la clase por un nuevo objeto de tipo AuthorDao.
+getClassificationDao ( ) : ClassificationDao	Devuelve el atributo classificationDao de la clase.
+setClassificationDao (classificationDao : ClassificationDao) :void	Cambia el atributo classificationDao de la clase por un nuevo objeto de tipo ClassificationDao.
+getContentDao ( ) : ContentDao	Devuelve el atributo contentDao de la clase.
+setContentDao (contentDao : ContentDao) : void	Cambia el atributo contentDao de la clase por un nuevo objeto de tipo ContentDao.
+getCountryDao ( ) : CountryDao	Devuelve el atributo countryDao de la clase.
+setCountryDao (countryDao : CountryDao) : void	Cambia el atributo countryDao de la clase por un nuevo objeto de tipo CountryDao.
+getProviderDao ( ) : ProviderDao	Devuelve el atributo providerDao de la clase.
+setProviderDao (providerDao : ProviderDao) : void	Cambia el atributo providerDao de la clase por un nuevo objeto de tipo ProviderDao.
+getOrderDao ( ) : OrderDao	Devuelve el atributo orderDao de la clase.
+setOrderDao (orderDao : OrderDao) : void	Cambia el atributo orderDao de la clase por un nuevo objeto de tipo OrderDao.
+findAllAvailableContentFather ( ) : Collection<Content>	Hace referencia al método findAllAvailableContentFather() de la clase <b>ContentDao&lt;Interface&gt;</b> .
+findAllContentFather ( ) : Collection<Content>	Hace referencia al método findAllContentFather() de la clase <b>ContentDao&lt;Interface&gt;</b> .
+findAllAvailableContentBy Category (contentId : Long) : Collection<Content>	Hace referencia al método findAllAvailableContentByCategory() de la clase <b>ContentDao&lt;Interface&gt;</b> .
+findAllClassifications ( ) : Collection<Classification>	Hace referencia al método findAllClassifications() de la clase <b>ClassificationDao&lt;Interface&gt;</b> .
+findAllContentByCategory (contentId : Long) : Collection<Content>	Hace referencia al método findAllContentByCategory() de la clase <b>ContentDao&lt;Interface&gt;</b> .
+findAllCountries	Hace referencia al método findAllCountries() de la clase

## Anexo II: Descripción de las Clases del Diseño

() : Collection<Country>	<b>CountryDao&lt;Interface&gt;</b> .
+findAllOrders () : Collection<Order>	Hace referencia al método findAllOrders() de la clase <b>OrderDao&lt;Interface&gt;</b> .
+findAllProviders () : Collection<Provider>	Hace referencia al método findAllProviders() de la clase <b>ProviderDao&lt;Interface&gt;</b> .
+findAuthorById (authorId : Long) : Author	Hace referencia al método findAuthorById() de la clase <b>AuthorDao&lt;Interface&gt;</b> .
+findClassificationById (classificationId : Long) : Classification	Hace referencia al método findClassificationById() de la clase <b>ClassificationDao&lt;Interface&gt;</b> .
+findContentById (contentId : Long) : Content	Hace referencia al método findContentById() de la clase <b>ContentDao&lt;Interface&gt;</b> .
+findCountryById (countryId : Long) : Country	Hace referencia al método findCountryById() de la clase <b>CountryDao&lt;Interface&gt;</b> .
+findOrderById (orderId : Long) : Order	Hace referencia al método findOrderById() de la clase <b>OrderDao&lt;Interface&gt;</b> .
+findProviderById (providerId : Long) : Provider	Hace referencia al método findProviderById() de la clase <b>ProviderDao&lt;Interface&gt;</b> .
+removeAuthor (authorId : Long) : void	Hace referencia al método removeAuthor() de la clase <b>AuthorDao&lt;Interface&gt;</b> .
+removeContent (contentId : Long) : void	Hace referencia al método removeContent() de la clase <b>ContentDao&lt;Interface&gt;</b> .
+removeContentClassification (classificationId : Long) : void	Hace referencia al método removeContentClassification() de la clase <b>ClassificationDao&lt;Interface&gt;</b> .
+removeCountry (countryId : Long) : void	Hace referencia al método removeCountry() de la clase <b>CountryDao&lt;Interface&gt;</b> .
+removeOrder (orderId : Long) : void	Hace referencia al método removeOrder() de la clase <b>OrderDao&lt;Interface&gt;</b> .
+removeProvider (providerId : Long) : void	Hace referencia al método removeProvider() de la clase <b>ProviderDao&lt;Interface&gt;</b> .
+saveAuthor (author : Author) : void	Hace referencia al método saveAuthor() de la clase <b>AuthorDao&lt;Interface&gt;</b> .
+saveContent (content : Content) : Long	Hace referencia al método saveContent() de la clase <b>ContentDao&lt;Interface&gt;</b> .
+saveContentClassification (classification : Classification) : void	Hace referencia al método saveContentClassification() de la clase <b>ClassificationDao&lt;Interface&gt;</b> .
+saveCountry (country : Country) : void	Hace referencia al método saveCountry() de la clase <b>CountryDao&lt;Interface&gt;</b> .
+saveOrder (order : Order) : Long	Hace referencia al método saveOrder() de la clase <b>OrderDao&lt;Interface&gt;</b> .
+saveProvider (provider : Provider) : void	Hace referencia al método saveProvider() de la clase <b>ProviderDao&lt;Interface&gt;</b> .
+updateAuthor (author : Author) : void	Hace referencia al método updateAuthor() de la clase <b>AuthorDao&lt;Interface&gt;</b> .
+updateContent (content : Content) : void	Hace referencia al método updateContent() de la clase <b>ContentDao&lt;Interface&gt;</b> .
+updateContentClassification (classification : Classification) : void	Hace referencia al método updateContentClassification() de la clase <b>ClassificationDao&lt;Interface&gt;</b> .

## Anexo II: Descripción de las Clases del Diseño

+updateCountry (country : Country) : void	Hace referencia al método updateCountry() de la clase <b>CountryDao&lt;Interface&gt;</b> .
+updateOrder (order : Order) : void	Hace referencia al método updateOrder() de la clase <b>OrderDao&lt;Interface&gt;</b> .
+updateProvider (provider : Provider) : void	Hace referencia al método updateProvider() de la clase <b>ProviderDao&lt;Interface&gt;</b> .
+findAllAuthors () : Collection<Author>	Hace referencia al método findAllAuthors() de la clase <b>AuthorDao&lt;Interface&gt;</b> .
+findAllProvidersByCountryId (countryId : Long) : Collection<Provider>	Hace referencia al método findAllProvidersByCountryId() de la clase <b>ProviderDao&lt;Interface&gt;</b> .
+findAllAvailableContentBy Author (authorId : Long) : Collection<Content>	Hace referencia al método findAllAvailableContentByAuthor() de la clase <b>ContentDao&lt;Interface&gt;</b> .
+findAllAvailableContentBy Classification (classificationId : Long) : Collection<Content>	Hace referencia al método findAllAvailableContentByClassification() de la clase <b>ContentDao&lt;Interface&gt;</b> .
+findAllAvailableContentBy Provider (providerId : Long) : Collection<Content>	Hace referencia al método findAllAvailableContentByProvider() de la clase <b>ContentDao&lt;Interface&gt;</b> .
+findAllContentByAuthor (authorId : Long) : Collection<Content>	Hace referencia al método findAllContentByAuthor() de la clase <b>ContentDao&lt;Interface&gt;</b> .
+findAllContentByClassification (classificationId : Long) : Collection<Content>	Hace referencia al método findAllContentByClassification() de la clase <b>ContentDao&lt;Interface&gt;</b> .
+findAllContentByProvider (providerId : Long) : Collection<Content>	Hace referencia al método findAllContentByProvider() de la clase <b>ContentDao&lt;Interface&gt;</b> .

### Clase del Diseño ContentDao<Interface>

Atributos	Tipo
Operaciones	Descripción
+saveContent (content : Content) : Long	Hace referencia al método saveContent() en la clase <b>ContentDaoImpl</b> .
+updateContent (content : Content) : void	Hace referencia al método updateContent() en la clase <b>ContentDaoImpl</b> .
+removeContent (contentId : Long) : void	Hace referencia al método removeContent() en la clase <b>ContentDaoImpl</b> .
+findContentById (contentId : Long) : Content	Hace referencia al método findContentById() en la clase <b>ContentDaoImpl</b> .
+findAllContentByCategory (contentId : Long) : Collection<Content>	Hace referencia al método findAllContentByCategory() en la clase <b>ContentDaoImpl</b> .
+findAllAvailableContentBy Category	Hace referencia al método findAllAvailableContentByCategory() en la clase <b>ContentDaoImpl</b> .

## Anexo II: Descripción de las Clases del Diseño

(contentId : Long) : Collection<Content>	
+findAllAvailableContentFather ( ) : Collection<Content>	Hace referencia al método findAllAvailableContentFather() en la clase <b>ContentDaolmpl</b> .
+findAllContentFather ( ) : Collection<Content>	Hace referencia al método findAllContentFather() en la clase <b>ContentDaolmpl</b> .
+findAllAvailableContentBy Provider (providerId : Long) : Collection<Content>	Hace referencia al método findAllAvailableContentByProvider() en la clase <b>ContentDaolmpl</b> .
+findAllContentByProvider (providerId : Long) : Collection<Content>	Hace referencia al método findAllContentByProvider() en la clase <b>ContentDaolmpl</b> .
+findAllAvailableContentBy Classification (classificationId : Long) :Collection<Content>	Hace referencia al método findAllAvailableContentByClassification() en la clase <b>ContentDaolmpl</b> .
+findAllContentByClassification (classificationId : Long) :Collection<Content>	Hace referencia al método findAllContentByClassification() en la clase <b>ContentDaolmpl</b> .
+findAllAvailableContentBy Author (authorId : Long) : Collection<Content>	Hace referencia al método findAllAvailableContentByAuthor() en la clase <b>ContentDaolmpl</b> .
+findAllContentByAuthor (authorId : Long) : Collection<Content>	Hace referencia al método findAllContentByAuthor() en la clase <b>ContentDaolmpl</b> .

### Clase del Diseño ContentDaolmpl

Atributos	Tipo
-em	EntityManager
Operaciones	Descripción
+findAllAvailableContentBy Category (contentId : Long) : Collection<Content>	Devuelve una lista con todos los contenidos disponibles dado el id de una categoría.
+findAllContentByCategory (contentId : Long) : Collection<Content>	Devuelve todos los contenidos dado el id de una categoría.
+findContentById (contentId : Long) : Content	Devuelve un contenido según el id.
+removeContent (contentId : Long) : void	Elimina un contenido.
+saveContent (content : Content) : Long	Guarda un contenido.
+updateContent (content : Content) : void	Actualiza un contenido.
+findAllAvailableContentFather ( ) : Collection<Content>	Devuelve un listado de todos los contenidos.
+findAllContentFather	Devuelve un listado con todos los contenidos que representan

## Anexo II: Descripción de las Clases del Diseño

() : Collection<Content>	categorías.
+findAllAvailableContentBy Author (authorId : Long) : Collection<Content>	Devuelve un listado de todos los contenidos disponibles dado el id de un autor.
+findAllAvailableContentBy Classification (classificationId : Long) : Collection<Content>	Devuelve un listado de todos los contenidos disponibles por clasificación.
+findAllAvailableContentBy Provider (providerId : Long) : Collection<Content>	Devuelve un listado de todos los contenidos disponibles por proveedor.
+findAllContentByAuthor (authorId : Long) : Collection<Content>	Devuelve un listado de todos los contenidos dado el id de un autor.
+findAllContentByClassification (classificationId : Long) : Collection<Content>	Devuelve un listado de todos los contenidos disponibles por clasificación.
+findAllContentByProvider (providerId : Long) : Collection<Content>	Devuelve un listado de todos los contenidos disponibles por proveedor.

Clase del Diseño AuthorDao<Interface>

Atributos	Tipo
Operaciones	Descripción
+saveAuthor (author : Author) : void	Hace referencia al método saveAuthor() en la clase <b>AuthorDaoImpl</b> .
+updateAuthor (author : Author) : void	Hace referencia al método updateAuthor() en la clase <b>AuthorDaoImpl</b> .
+removeAuthor (authorId : Long) : void	Hace referencia al método removeAuthor() en la clase <b>AuthorDaoImpl</b> .
+findAuthorById (authorId : Long) : Author	Hace referencia al método findAuthorById() en la clase <b>AuthorDaoImpl</b> .
+findAllAuthors ( ) : Collection<Author>	Hace referencia al método findAllAuthors() en la clase <b>AuthorDaoImpl</b> .

Clase del Diseño AuthorDaoImpl

Atributos	Tipo
-em	EntityManager
Operaciones	Descripción
+findAllAuthors ( ) : Collection<Author>	Devuelve un listado con todos los autores.
+findAuthorById (authorId : Long) : Author	Devuelve un autor dado un id.
+removeAuthor (authorId : Long) : void	Elimina un autor.
+saveAuthor	Guarda un autor.

## Anexo II: Descripción de las Clases del Diseño

(author : Author) : void	
+updateAuthor (author : Author) : void	Actualiza un autor.

Clase del Diseño ClassificationDao<Interface>

Atributos	Tipo
Operaciones	Descripción
+saveContentClassification( classification : Classification) : void	Hace referencia al método saveContentClassification() en la clase <b>ClassificationDaoImpl</b> .
+updateContentClassification (classification : Classification) : void	Hace referencia al método updateContentClassification() en la clase <b>ClassificationDaoImpl</b> .
+removeContentClassification (classificationId : Long) : void	Hace referencia al método removeContentClassification() en la clase <b>ClassificationDaoImpl</b> .
+findClassificationById (classificationId : Long) : Classification	Hace referencia al método findClassificationById() en la clase <b>ClassificationDaoImpl</b> .
+findAllClassifications ( ) : Collection<Classification>	Hace referencia al método findAllClassifications() en la clase <b>ClassificationDaoImpl</b> .

Clase del Diseño ClassificationDaoImpl

Atributos	Tipo
-em	EntityManager
Operaciones	Descripción
+findAllClassifications ( ) : Collection<Classification>	Devuelve un listado de todas las clasificaciones.
+findClassificationById (classificationId : Long) : Classification	Devuelve una clasificación dado un id.
+removeContentClassification (classificationId : Long) : void	Elimina una clasificación.
+saveContentClassification (classification : Classification) : void	Guarda una clasificación.
+updateContentClassification (classification : Classification) : void	Elimina una clasificación.

Clase del Diseño ProviderDao<Interface>

Atributos	Tipo
Operaciones	Descripción
+saveProvider (provider : Provider) : void	Hace referencia al método saveProvider() en la clase <b>ProviderDaoImpl</b> .
+updateProvider (provider : Provider) : void	Hace referencia al método updateProvider() en la clase <b>ProviderDaoImpl</b> .

## Anexo II: Descripción de las Clases del Diseño

+removeProvider (providerId : Long) : void	Hace referencia al método removeProvider() en la clase <b>ProviderDaoImpl</b> .
+findProviderById (providerId : Long) : Provider	Hace referencia al método findProviderById() en la clase <b>ProviderDaoImpl</b> .
+findAllProviders ( ) : Collection<Provider>	Hace referencia al método findAllProviders() en la clase <b>ProviderDaoImpl</b> .
+findAllProvidersByCountryId (countryId : Long) : Collection<Provider>	Hace referencia al método findAllProvidersByCountryId() en la clase <b>ProviderDaoImpl</b> .

Clase del Diseño ProviderDaoImpl

Nombre de la Clase	ProviderDaoImpl
Atributos	Tipo
-em	EntityManager
Operaciones	Descripción
+findAllProviders ( ) : Collection<Provider>	Devuelve un listado de todos los proveedores.
+findProviderById (providerId : Long) : Provider	Devuelve un proveedor dado un id.
+removeProvider (providerId : Long) : void	Elimina un proveedor.
+saveProvider (provider : Provider) : void	Guarda un proveedor.
+updateProvider (provider : Provider) : void	Actualiza un proveedor.
+findAllProvidersByCountryId (countryId : Long) : Collection<Provider>	Devuelve un listado de todos los proveedores dado el id de un país.

Clase del Diseño CountryDao<Interface>

Atributos	Tipo
Operaciones	Descripción
+saveCountry (country : Country) : void	Hace referencia al método saveCountry() en la clase <b>CountryDaoImpl</b> .
+updateCountry (country : Country) : void	Hace referencia al método updateCountry() en la clase <b>CountryDaoImpl</b> .
+removeCountry (countryId : Long) : void	Hace referencia al método removeCountry() en la clase <b>CountryDaoImpl</b> .
+findCountryById (countryId : Long) : Country	Hace referencia al método findCountryById() en la clase <b>CountryDaoImpl</b> .
+findAllCountries ( ) : Collection<Country>	Hace referencia al método findAllCountries() en la clase <b>CountryDaoImpl</b> .

Clase del Diseño CountryDaoImpl

Atributos	Tipo
-em	EntityManager
Operaciones	Descripción

## Anexo II: Descripción de las Clases del Diseño

+findAllCountries ( ) :Collection<Country>	Devuelve un listado de todos los países.
+findCountryById (countryId : Long) : Country	Devuelve un país dado un id.
+removeCountry (countryId : Long) : void	Elimina un país.
+saveCountry (country : Country) : void	Guarda un país.
+updateCountry (country : Country) : void	Actualiza un país.

Clase del Diseño OrderDao<Interface>

Atributos	Tipo
Operaciones	Descripción
+saveOrder (order : Order) : Long	Hace referencia al método saveOrder() en la clase <b>OrderDaoImpl</b> .
+updateOrder (order : Order) : void	Hace referencia al método updateOrder() en la clase <b>OrderDaoImpl</b> .
+removeOrder (orderId : Long) : void	Hace referencia al método removeOrder() en la clase <b>OrderDaoImpl</b> .
+findOrderById (orderId : Long) : Order	Hace referencia al método findOrderById() en la clase <b>OrderDaoImpl</b> .
+findAllOrders ( ) : Collection<Order>	Hace referencia al método findAllOrders() en la clase <b>OrderDaoImpl</b> .

Clase del Diseño OrderDaoImpl

Atributos	Tipo
-em	EntityManager
Operaciones	Descripción
+findAllOrders ( ) :Collection<Order>	Devuelve un listado con todas las órdenes.
+findOrderById (orderId : Long) : Order	Devuelve una orden dada un id.
+saveOrder(order : Order) : Long	Guarda una orden.
+updateOrder(order : Order) : void	Actualiza una orden.
+removeOrder(orderId: Long): void	Elimina una orden.

Clase del Diseño Content

Atributos	Tipo
-contentId	Long
-name	String
-price	float
-numberOfDownloads	int
-available	boolean

## Anexo II: Descripción de las Clases del Diseño

-drm	boolean
-mimeType	String
-contentFather	Contenido
-classification	Clasificación
-provider	Provider
-author	Author
-contentByteArray	byte[]
<b>Operaciones</b>	<b>Descripción</b>
+Content()	Permite inicializar un contenido vacío.
+Content(name : String, classification : Clasificación, available : boolean)	Permite inicializar un contenido con nombre, clasificación y disponibilidad.
+Content(name : String, classification : Clasificación, available : boolean, contentFather : Contenido)	Permite inicializar un contenido con nombre, clasificación, disponibilidad y contenido padre.
+Content(name : String, mimeType : String, price : float, available : boolean, contentFather : Contenido, classification : Clasificación, provider : Provider, author : Author, contentByteArray : byte [])	Permite inicializar un contenido con nombre, tipo, precio, disponibilidad, contenido padre, clasificación, proveedor, autor y arreglo de datos.
+getContentId() : Long	Permite obtener el identificador del contenido.
+setContentId(contentId : Long) : void	Permite cambiar el identificador del contenido.
+getName() : String	Permite obtener el nombre del contenido.
+setName(name : String) : void	Permite cambiar el nombre del contenido.
+getPrice() : float	Permite obtener el precio del contenido.
+setPrice(price : float) : void	Permite cambiar el precio del contenido.
+getNumberOfDownloads() : int	Permite obtener el número de descargas del contenido.
+setNumberOfDownloads(numberOfDownloads : int) : void	Permite cambiar el número de descargas del contenido.
+isAvailable() : Boolean	Permite conocer si el contenido está disponible.
+setAvailable(available : boolean) : void	Permite cambiar el estado de la disponibilidad del contenido.
+getMimeType() : String	Permite obtener el tipo del contenido.
+setMimeType(mimeType : String) : void	Permite cambiar el tipo del contenido.
+getContentFather() : Contenido	Permite obtener el contenido padre.
+setContentFather(contentFather : Contenido) : void	Permite cambiar el contenido padre.
+getClassification() : Clasificación	Permite obtener la clasificación.
+setClassification(classification : Clasificación) : void	Permite cambiar la clasificación.
+getProvider() : Provider	Permite obtener un proveedor.

## Anexo II: Descripción de las Clases del Diseño

+setProvider(provider : Provider) : void	Permite cambiar el proveedor.
+getAuthor() : Author	Permite obtener un autor.
+setAuthor(author : Author) : void	Permite cambiar un autor.
+getContentByteArray() : byte []	Permite obtener el arreglo de datos de un contenido.
+setContentByteArray (contentByteArray : byte []) : void	Permite cambiar el arreglo de datos de un contenido.
+isDrm (): Boolean	Permite saber si el contenido necesita drm.
+setDrm(drm: Boolean): void	Permite cambiar el drm.

Clase del Diseño Author

Atributos	Tipo
-authorId	Long
-authorName	String
Operaciones	Descripción
+Author()	Permite inicializar un autor vacío.
+Author(authorName : String)	Permite inicializar un autor con su nombre.
+getAuthorId() : Long	Permite obtener el identificar de un autor
+setAuthorId(authorId : Long) : void	Permite cambiar el identificar de un autor.
+getAuthorName() : String	Permite obtener el nombre de un autor.
+setAuthorName(authorName : String) : void	Permite cambiar el nombre de un autor.

Clase del Diseño Clasificación

Atributos	Tipo
-classificationId	Long
-classifier	String
Operaciones	Descripción
+Classification()	Permite inicializar una clasificación vacía.
+Classification(classifier : String)	Permite inicializar una clasificación con el nombre.
+getClassificationId() : Long	Permite obtener el identificador de una clasificación.
+setClassificationId (classificationId : Long) : void	Permite cambiar el identificador de una clasificación.
+getClassifier() : String	Permite obtener el nombre de una clasificación.
+setClassifier(classifier : String) : void	Permite cambiar el nombre de una clasificación.

Clase del Diseño Country

Atributos	Tipo
-countryId	Long
-countryName	String
Operaciones	Descripción
+Country()	Permite inicializar un país vacío.

## Anexo II: Descripción de las Clases del Diseño

+Country(countryName : String)	Permite inicializar un país con el nombre.
+getCountryId() : Long	Permite obtener el identificador de un país.
+setCountryId(countryId : Long) : void	Permite cambiar el identificador de un país.
+getCountryName() : String	Permite obtener el nombre de un país.
+setCountryName(countryName : String) : void	Permite cambiar el nombre de un país.

Clase del Diseño Provider

Atributos	Tipo
-providerId	Long
-providerName	String
-providerCountry	Country
Operaciones	Descripción
+Provider()	Permite inicializar un proveedor vacío.
+Provider(providerName : String, providerCountry : Country)	Permite inicializar un proveedor con el nombre y el país.
+getProviderId() : Long	Permite obtener el identificador de un proveedor.
+setProviderId(providerId : Long) : void	Permite cambiar el identificador de un proveedor.
+getProviderName() : String	Permite obtener el nombre de un proveedor.
+setProviderName(providerName : String) : void	Permite cambiar el nombre de un proveedor.
+getProviderCountry() : Country	Permite obtener el país de un proveedor.
+setProviderCountry(providerCountry : Country) : void	Permite cambiar el proveedor de un país.

Clase del Diseño Order

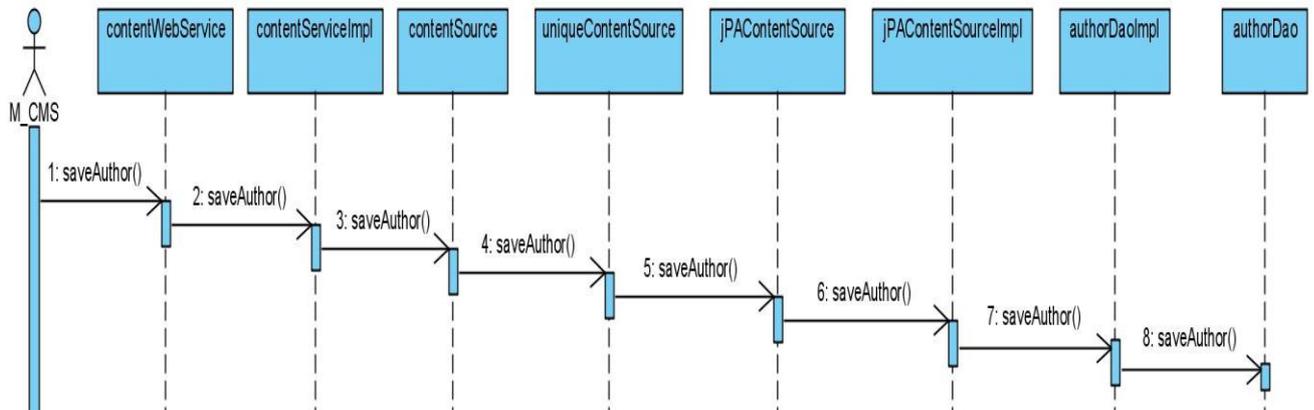
Atributos	Tipo
-orderId	Long
-cellularNumber	String
-userAgent	String
-requestDate	Date
-deliveryDate	Date
-requestedContent	Long
Operaciones	Descripción
+Order()	Permite inicializar una orden de descarga vacía.
+Order(orderId : Long, cellularNumber : String, userAgent : String, requestDate : Date, deliveryDate : Date, requestedContent : Long)	Permite inicializar una orden de descarga con el identificador, el número del celular, el user agent, la fecha de solicitud, la fecha de entrega y el identificador del contenido solicitado.
+getOrderId() : Long	Permite obtener el identificador de una orden de descarga.
+setOrderId(orderId : Long) : void	Permite cambiar el identificador de una orden de descarga.
+getCellularNumber() : String	Permite obtener el número del celular.

## Anexo II: Descripción de las Clases del Diseño

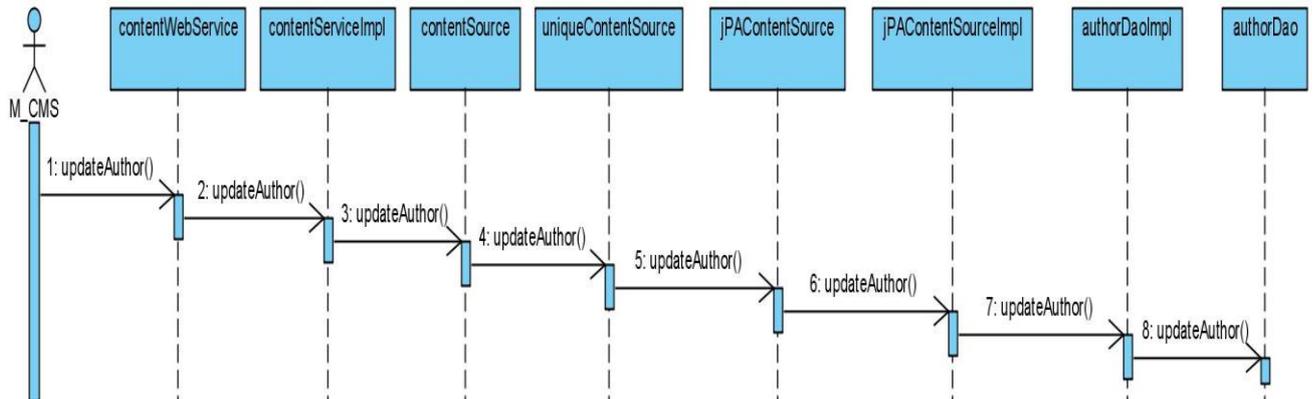
+setCellularNumber (cellularNumber : String) : void	Permite cambiar el número del celular.
+getUserAgent() : String	Permite obtener el user agent.
+setUserAgent(userAgent : String) : void	Permite cambiar el user agent.
+getRequestDate() : java.util.Date	Permite obtener la fecha de solicitud.
+setRequestDate(requestDate : java.util.Date) : void	Permite cambiar la fecha de solicitud.
+getDeliveryDate() : java.util.Date	Permite obtener la fecha entrega.
+setDeliveryDate(deliveryDate : java.util.Date) : void	Permite cambiar la fecha de entrega.
+getRequestedContent() : Long	Permite obtener el identificador del contenido solicitado.
+setRequestedContent(request edContent : Long) : void	Permite cambiar el identificador del contenido solicitado.

## Anexo III: Diagramas de Secuencia por casos de uso.

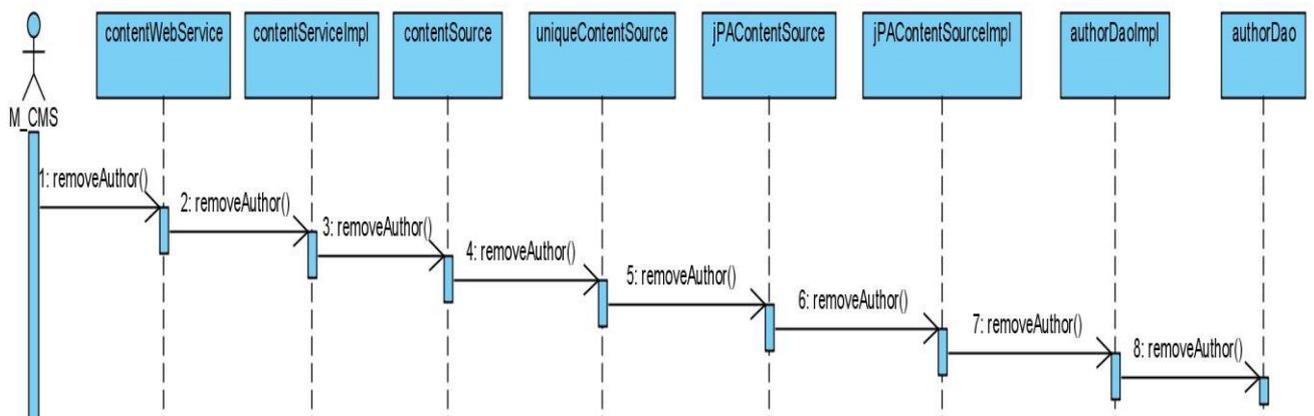
### CU Gestionar autor. Escenario Adicionar autor



### CU Gestionar autor. Escenario Modificar autor

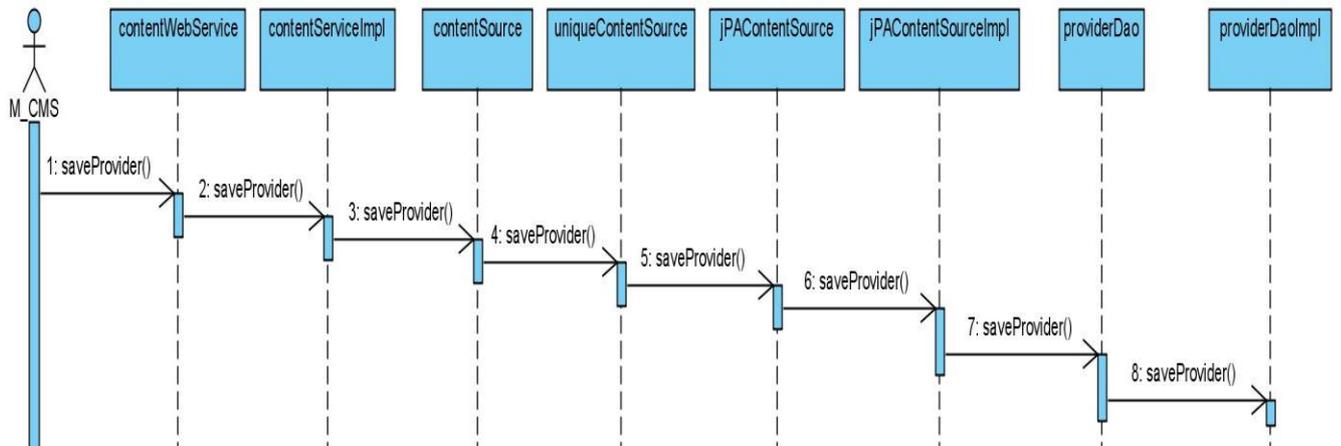


### CU Gestionar autor. Escenario Eliminar autor

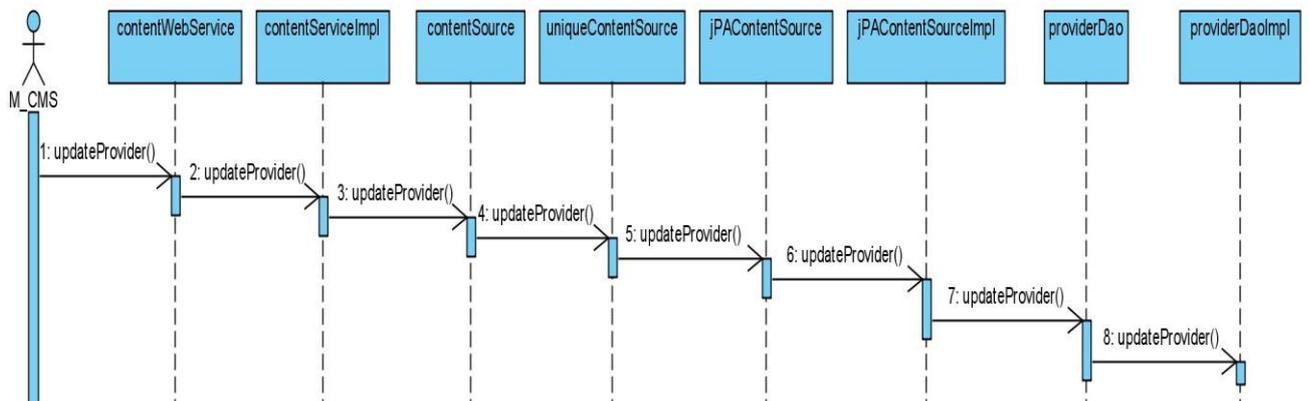


## Anexo III: Diagramas de Secuencias por Caso de Uso

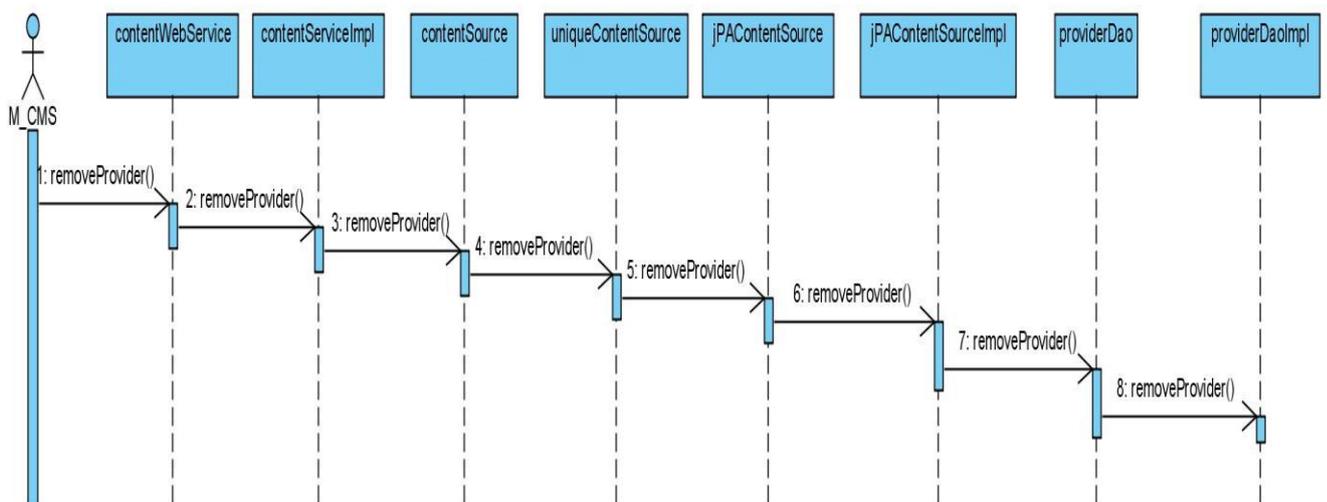
### CU Gestionar proveedor. Escenario Adicionar proveedor



### CU Gestionar proveedor. Escenario Modificar proveedor

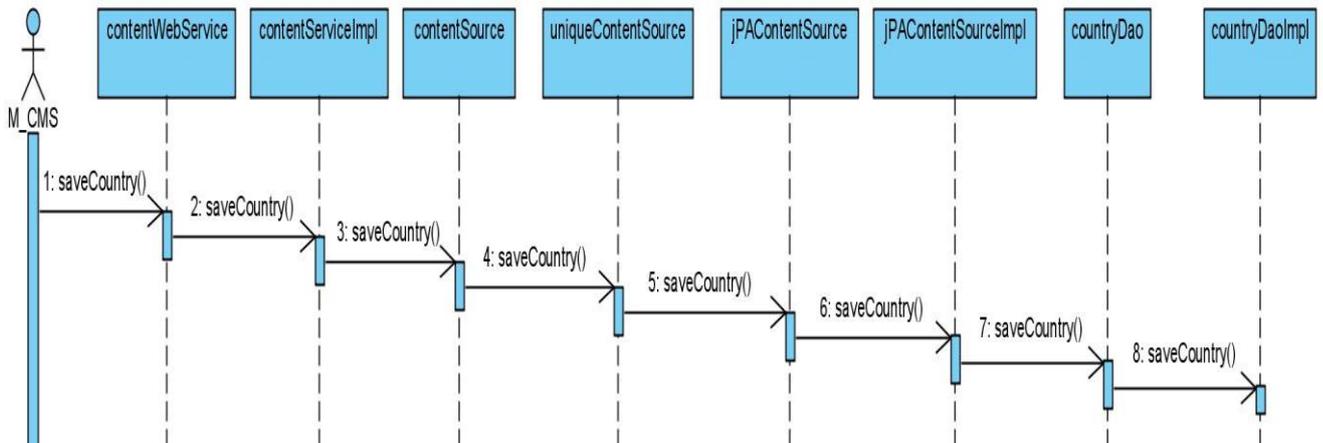


### CU Gestionar proveedor. Escenario Eliminar proveedor

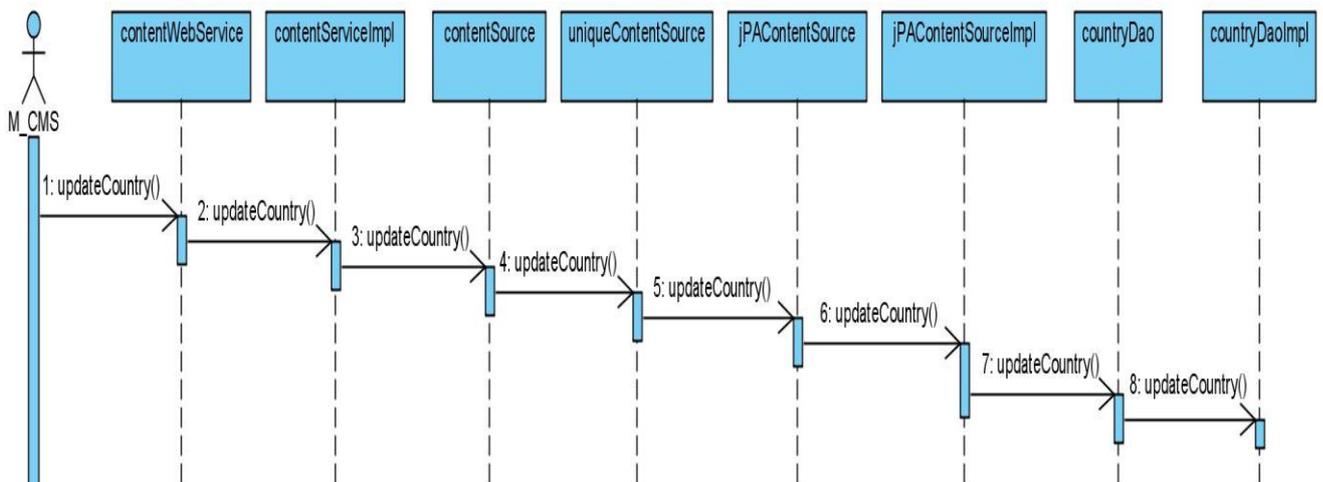


## Anexo III: Diagramas de Secuencias por Caso de Uso

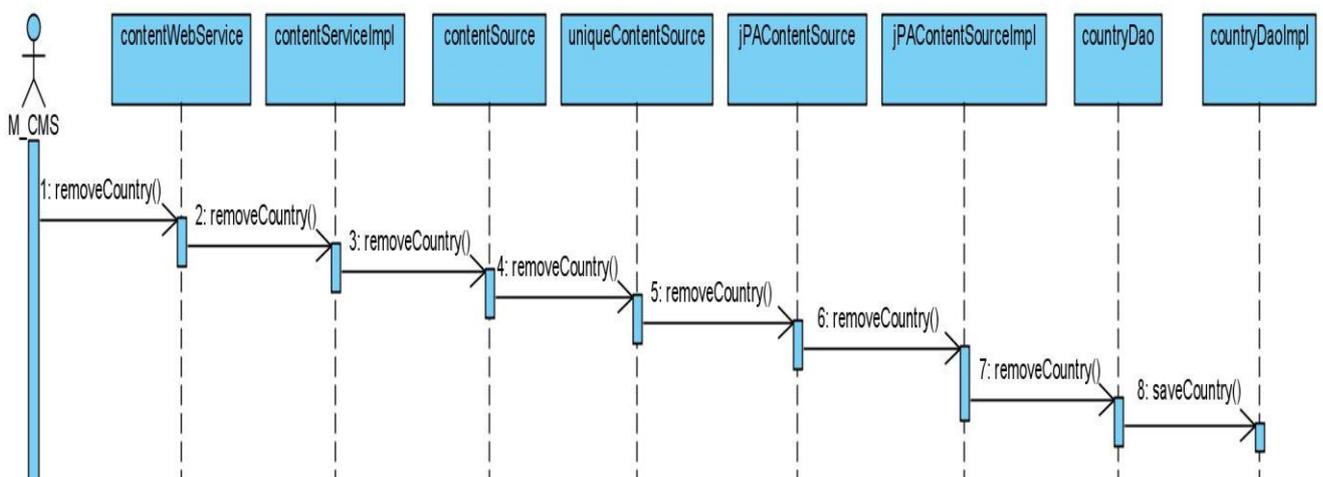
### CU Gestionar país. Escenario Adicionar país



### CU Gestionar país. Escenario Modificar país

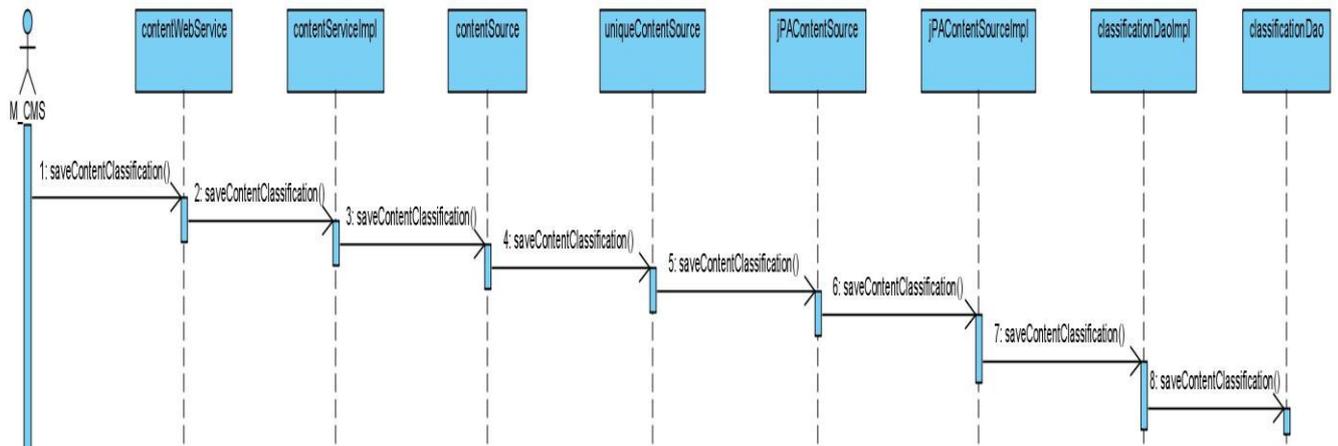


### CU Gestionar país. Escenario Eliminar país

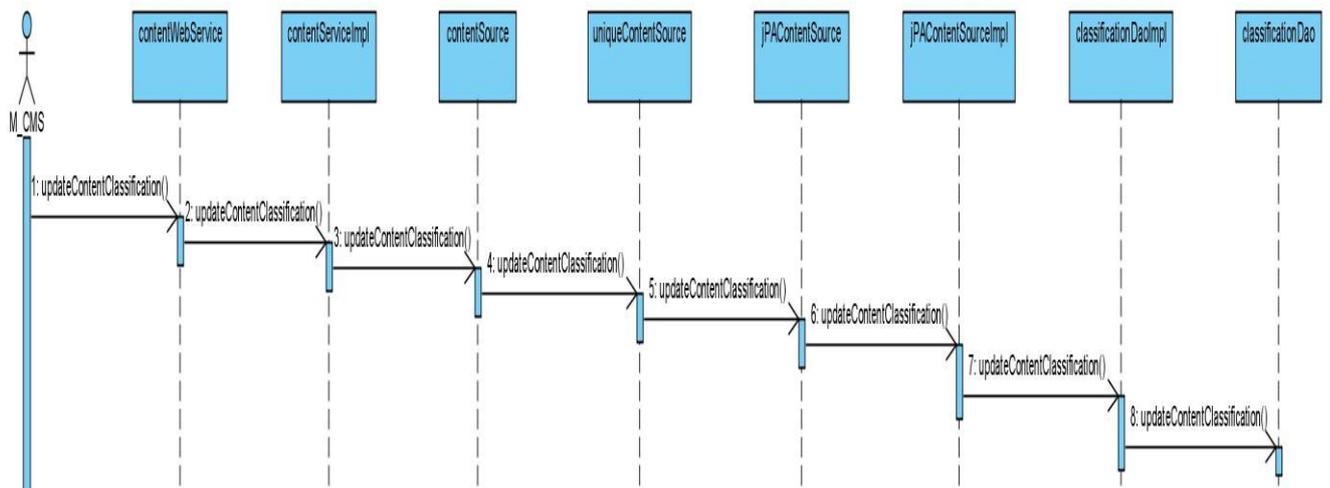


## Anexo III: Diagramas de Secuencias por Caso de Uso

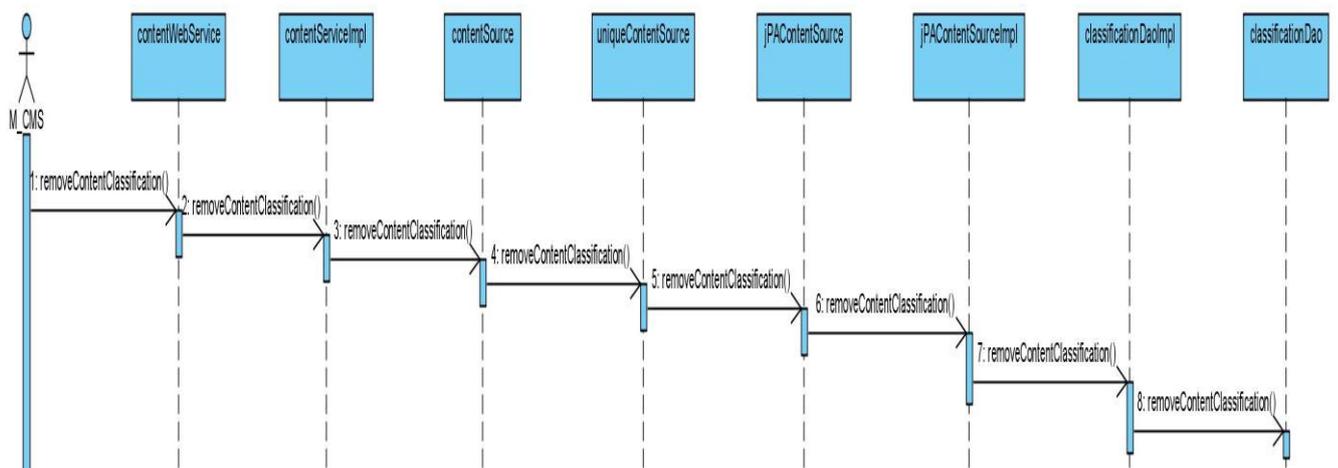
### CU Gestionar clasificación. Escenario Adicionar clasificación



### CU Gestionar clasificación. Escenario Modificar clasificación

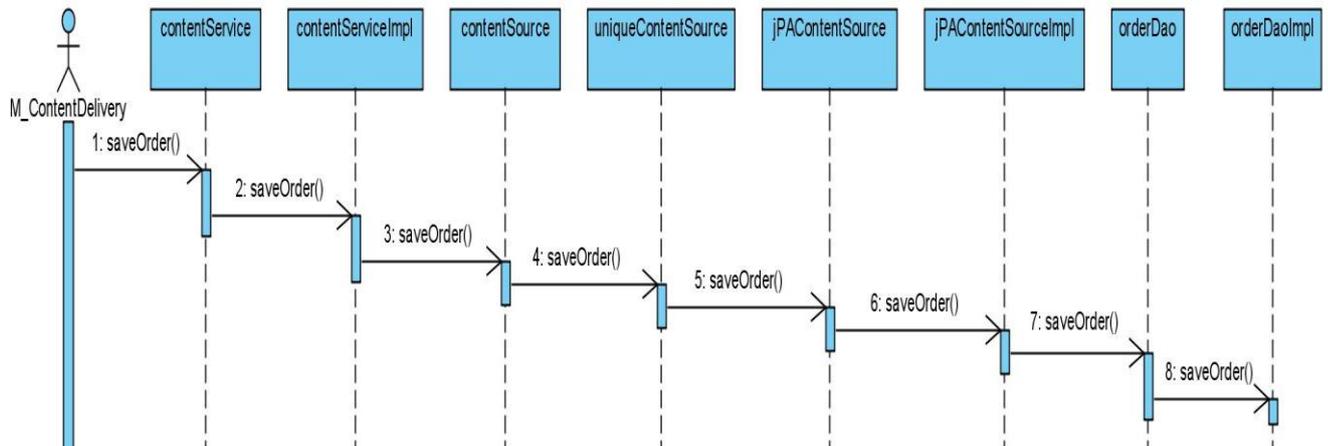


### CU Gestionar clasificación. Escenario Eliminar clasificación

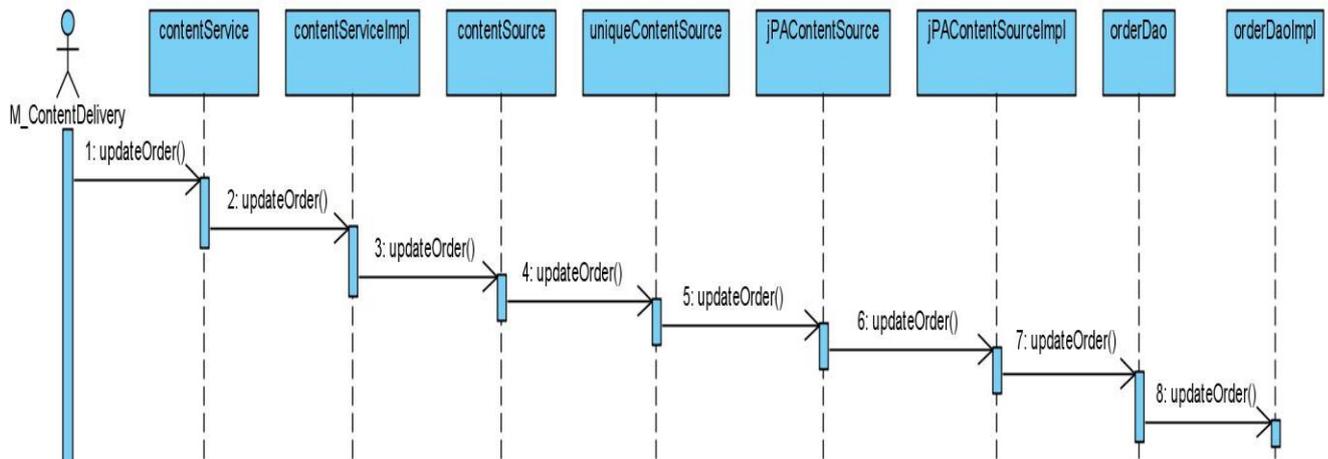


## Anexo III: Diagramas de Secuencias por Caso de Uso

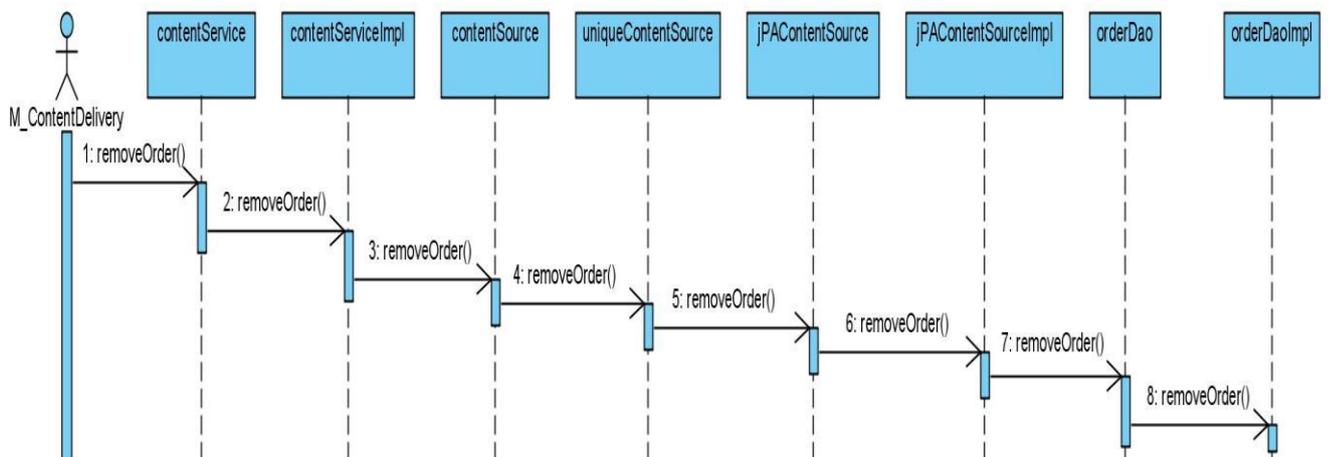
### CU Gestionar orden. Escenario Adicionar orden



### CU Gestionar orden. Escenario Modificar orden

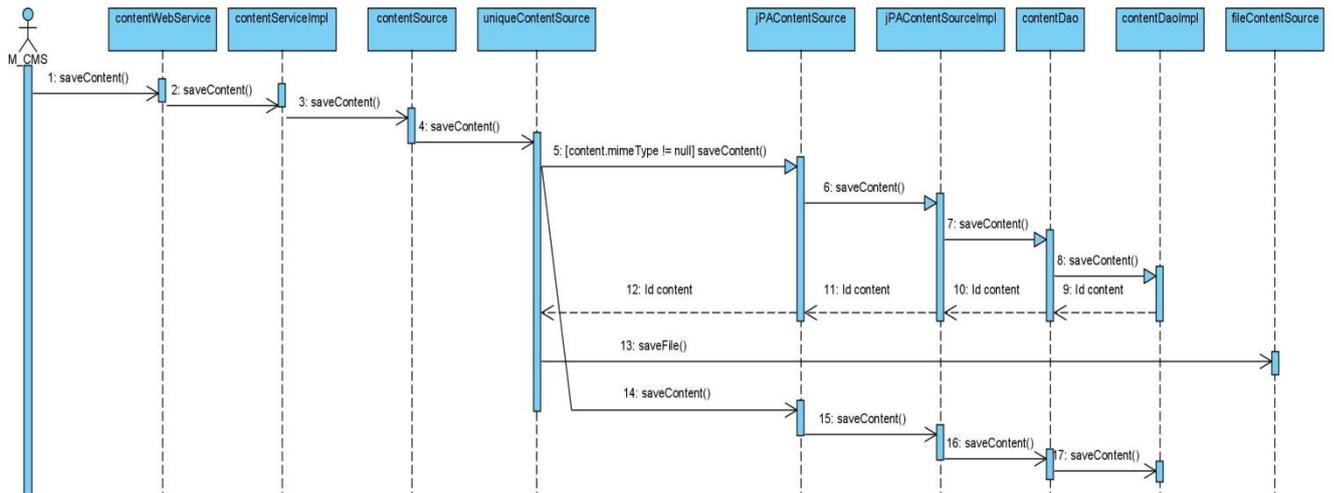


### CU Gestionar orden. Escenario Eliminar orden

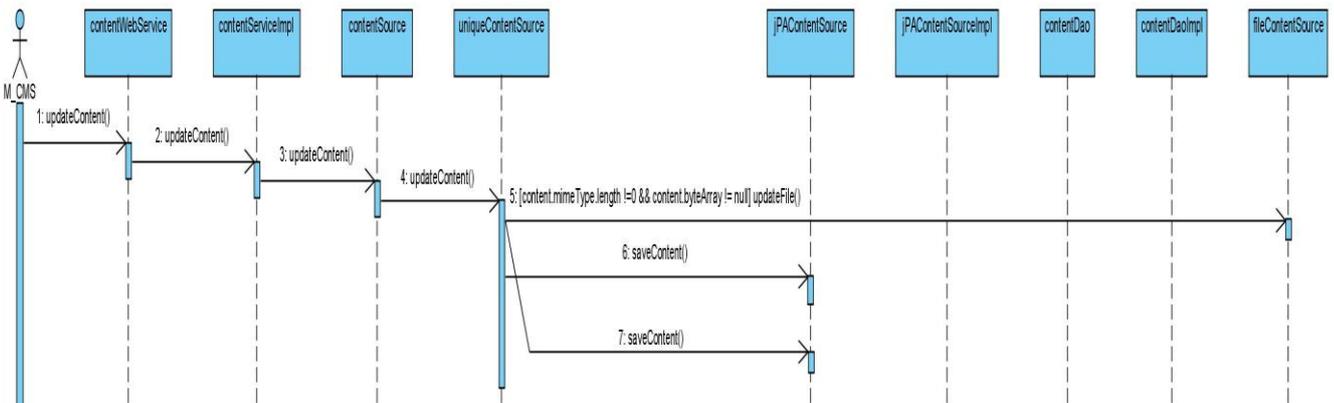


## Anexo III: Diagramas de Secuencias por Caso de Uso

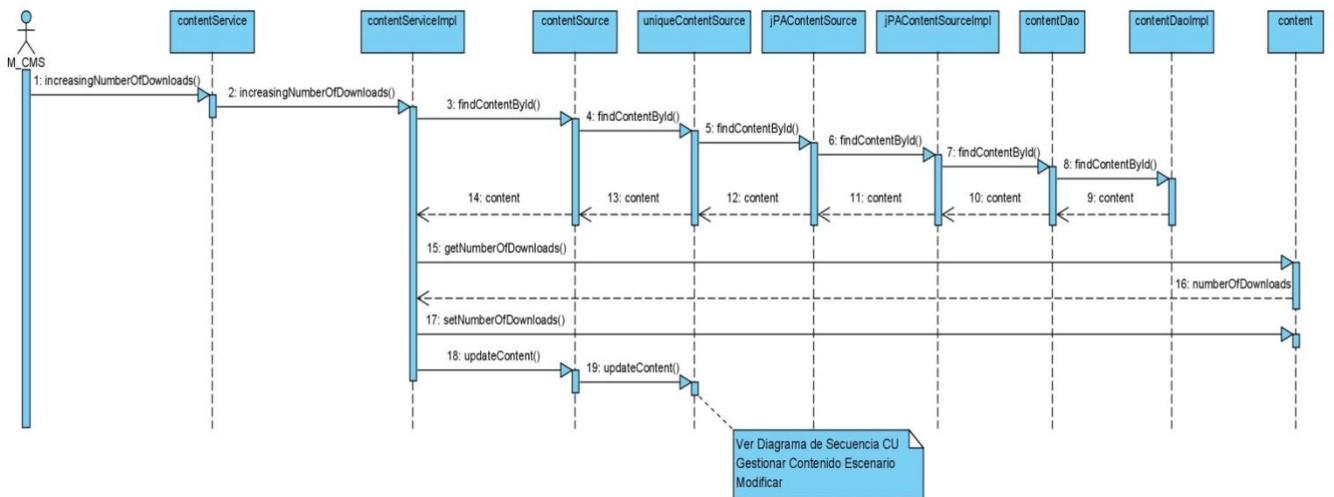
### CU Gestionar contenido. Escenario Adicionar contenido



### CU Gestionar contenido. Escenario Modificar contenido

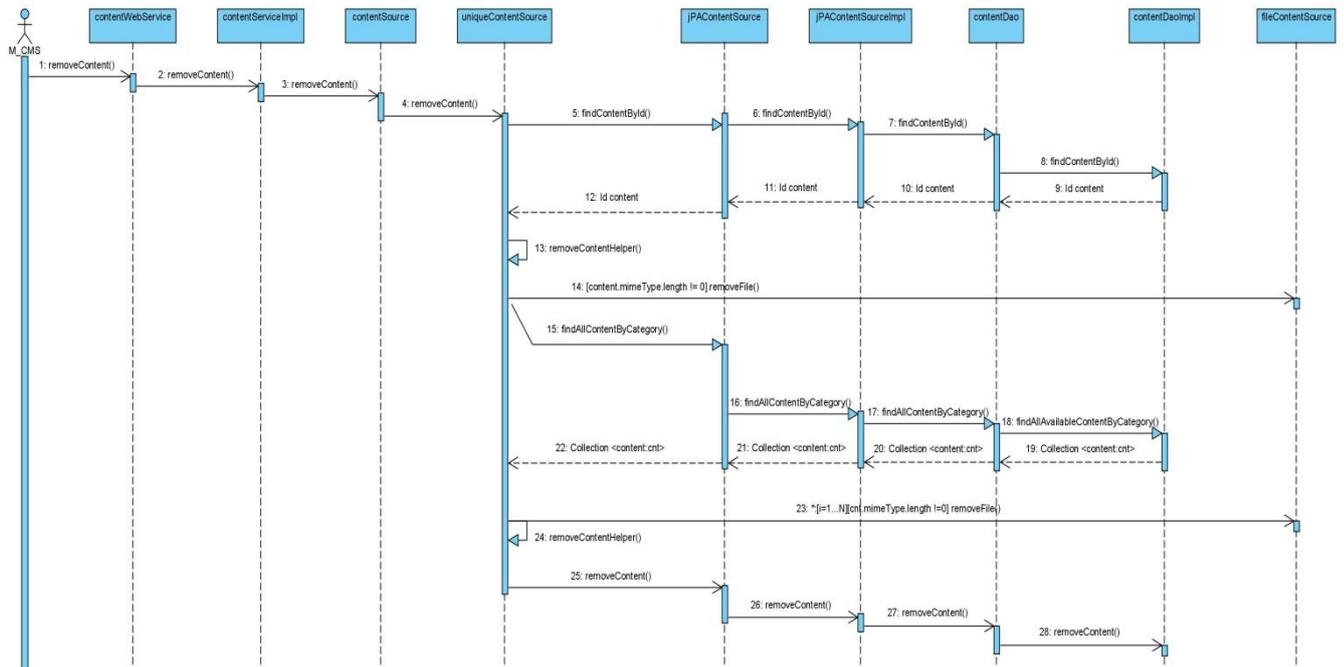


### CU Gestionar contenido. Escenario Incrementar número de descargas

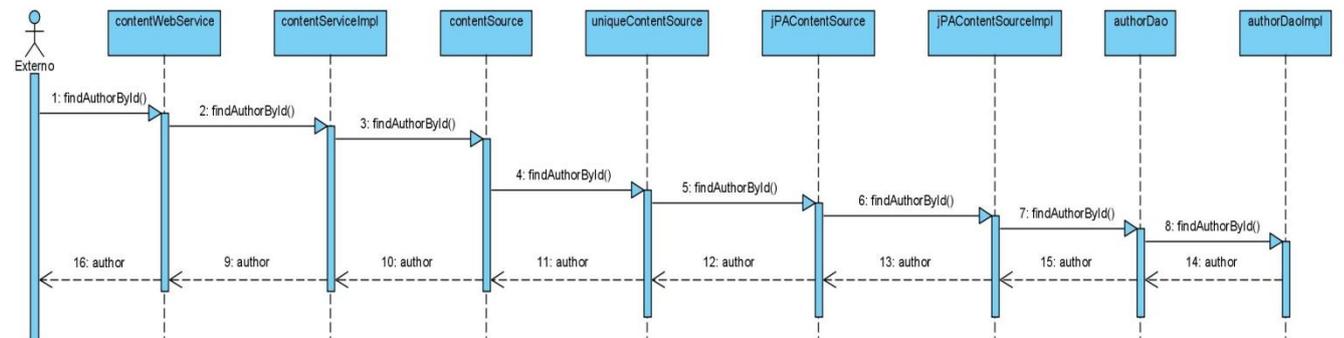


## Anexo III: Diagramas de Secuencias por Caso de Uso

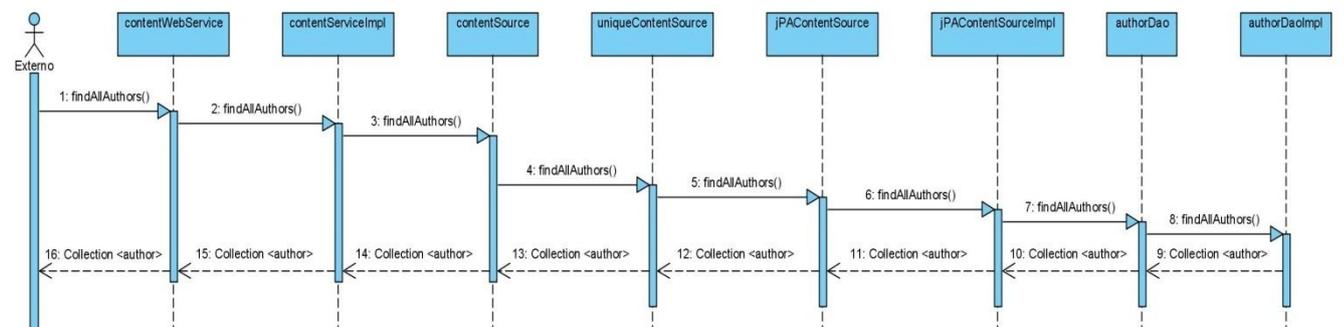
### CU Gestionar contenido. Escenario Eliminar contenido



### CU Buscar autor. Escenario Buscar autor por id

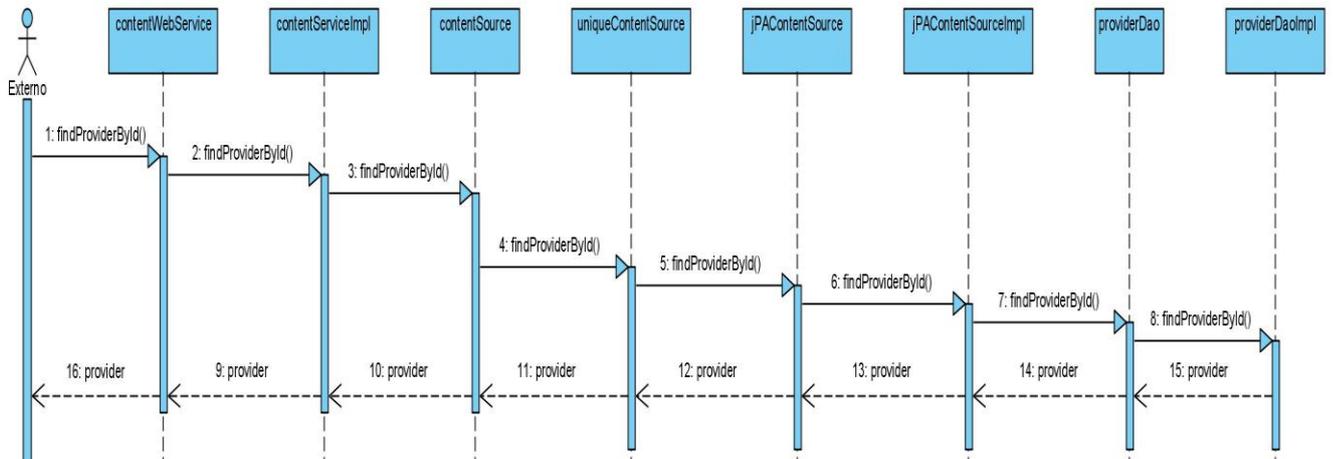


### CU Buscar autor. Escenario Buscar todos los autores

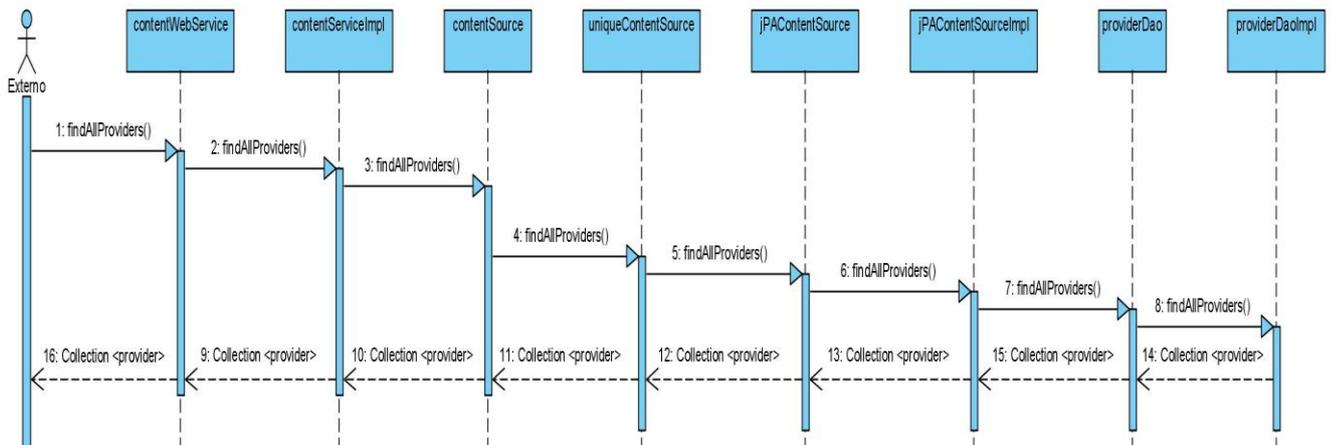


## Anexo III: Diagramas de Secuencias por Caso de Uso

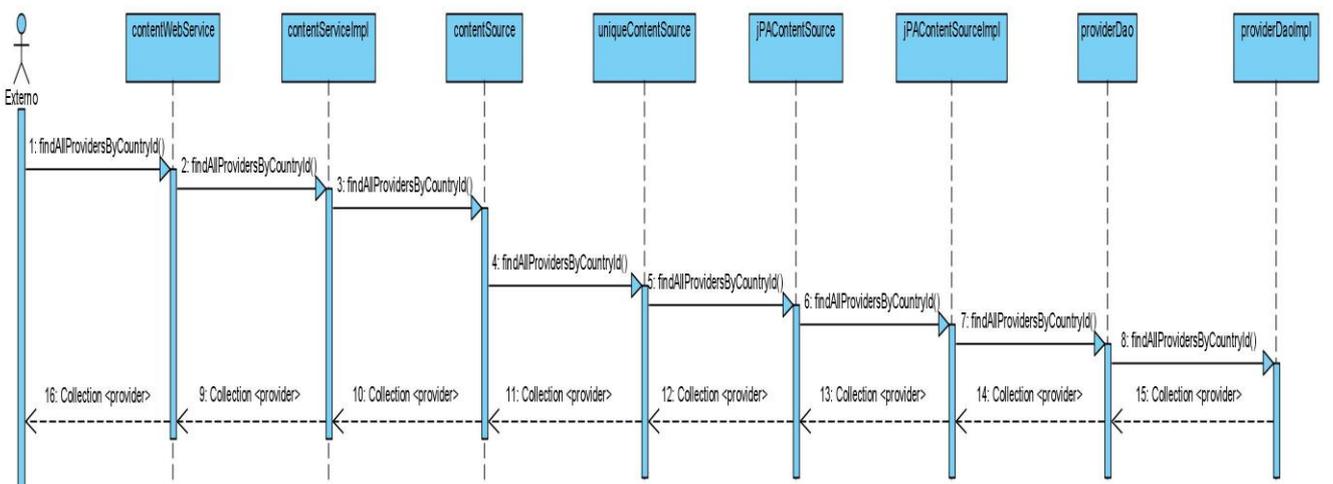
### CU Buscar proveedor. Escenario Buscar proveedor por ID



### CU Buscar proveedor. Escenario Buscar todos los proveedores

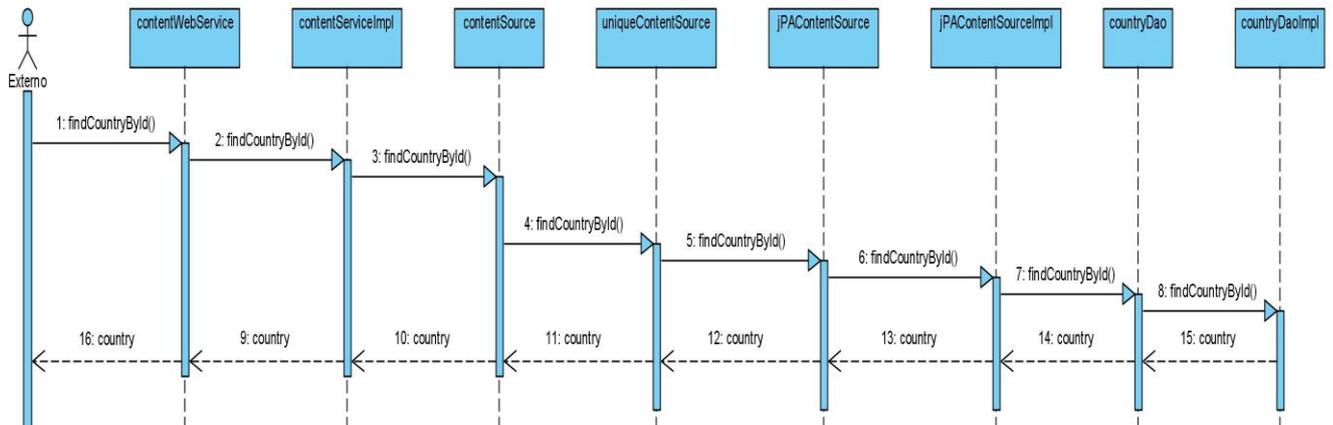


### CU Buscar proveedor. Escenario Buscar proveedor por país

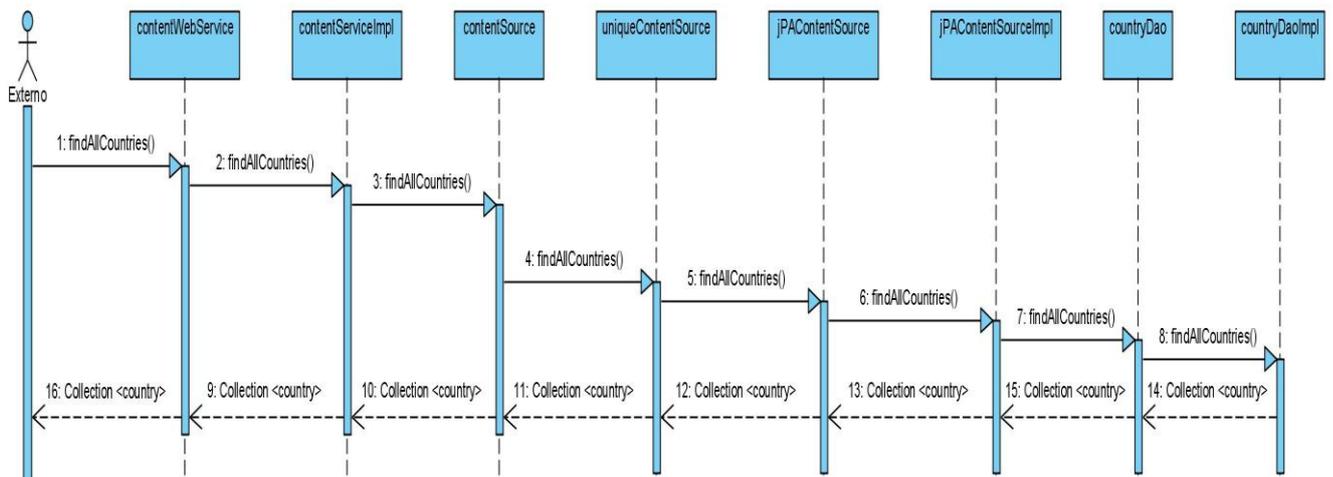


## Anexo III: Diagramas de Secuencias por Caso de Uso

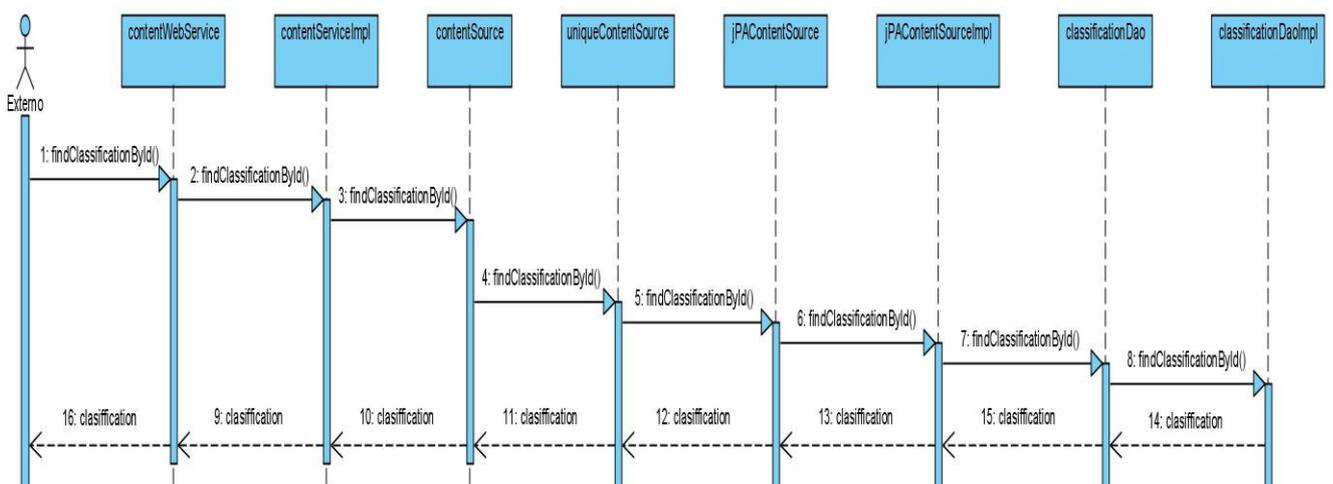
### CU Buscar país. Escenario Buscar países por ID



### CU Buscar país. Escenario Buscar todos los países

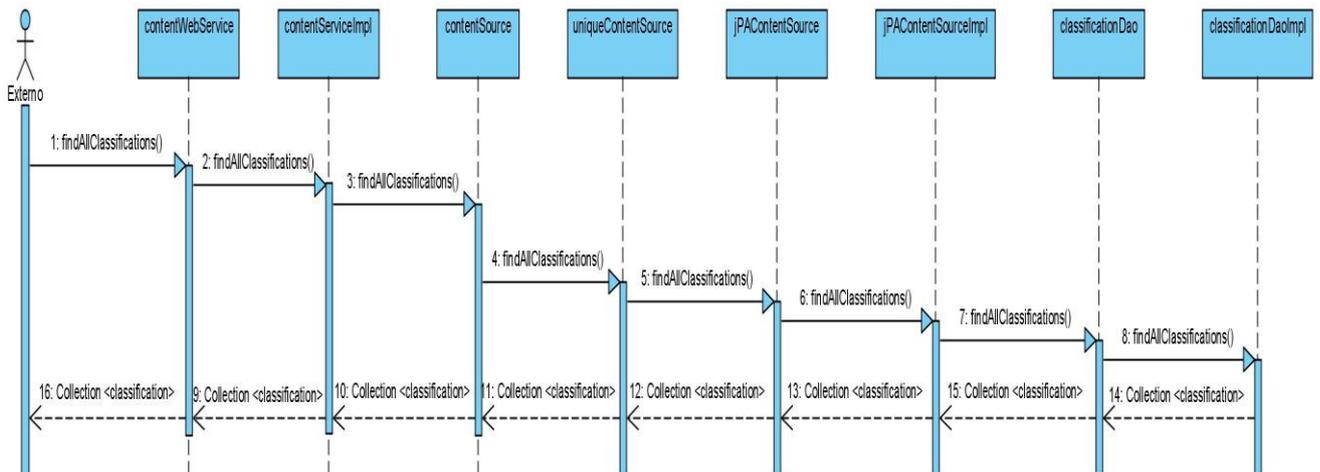


### CU Buscar clasificación. Escenario Buscar clasificación por ID

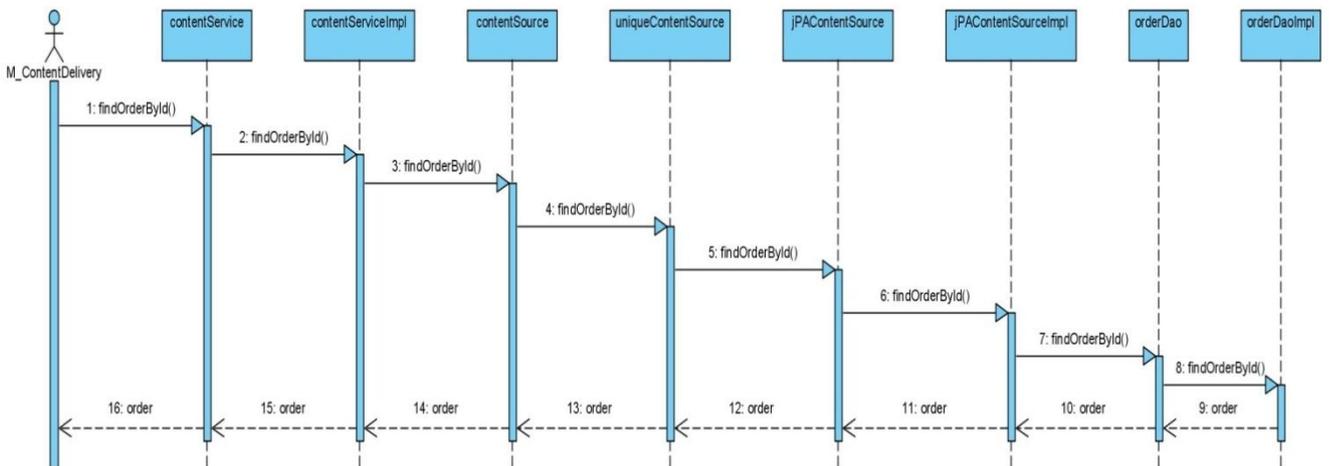


## Anexo III: Diagramas de Secuencias por Caso de Uso

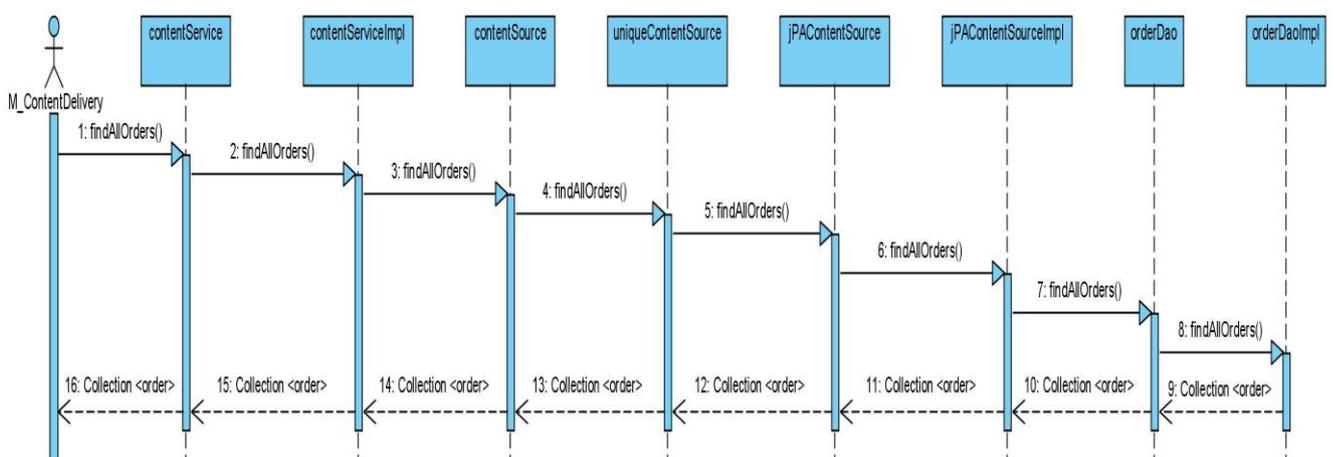
CU Buscar clasificación. Escenario Buscar todas las clasificaciones



CU Buscar orden. Escenario Buscar orden por ID

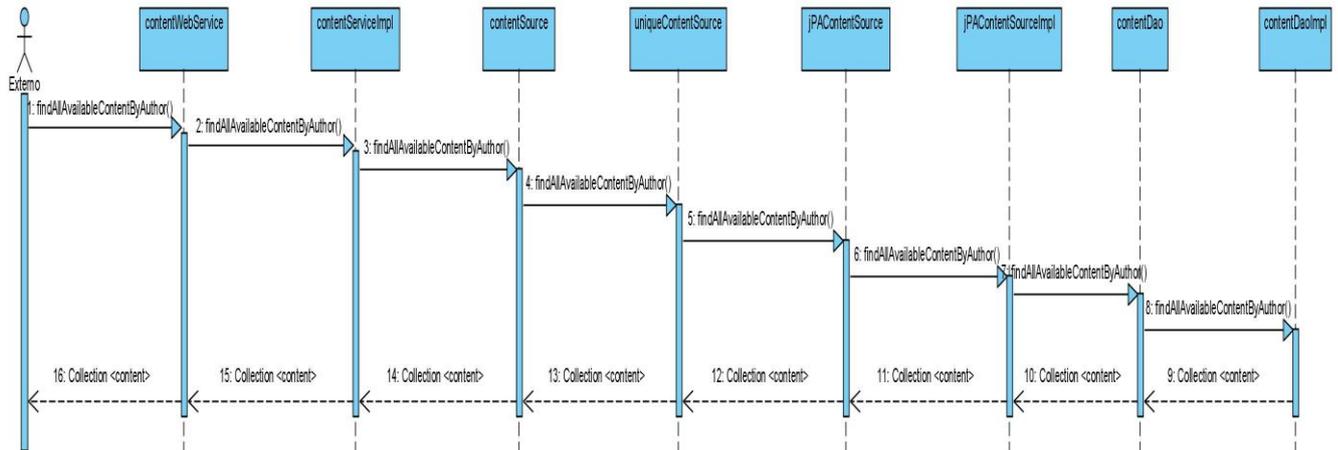


CU Buscar orden. Escenario Buscar todas las órdenes

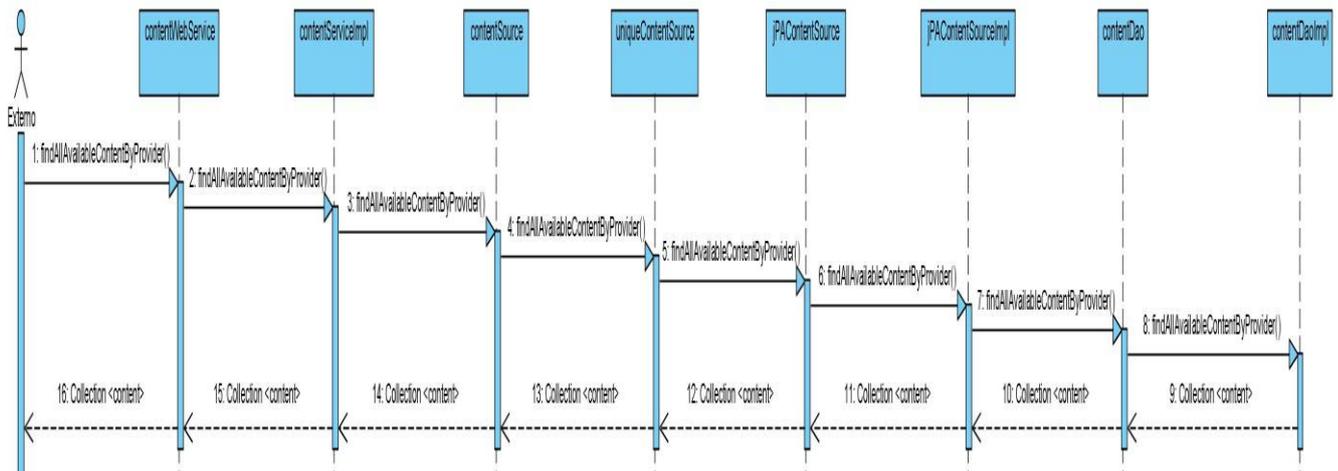


## Anexo III: Diagramas de Secuencias por Caso de Uso

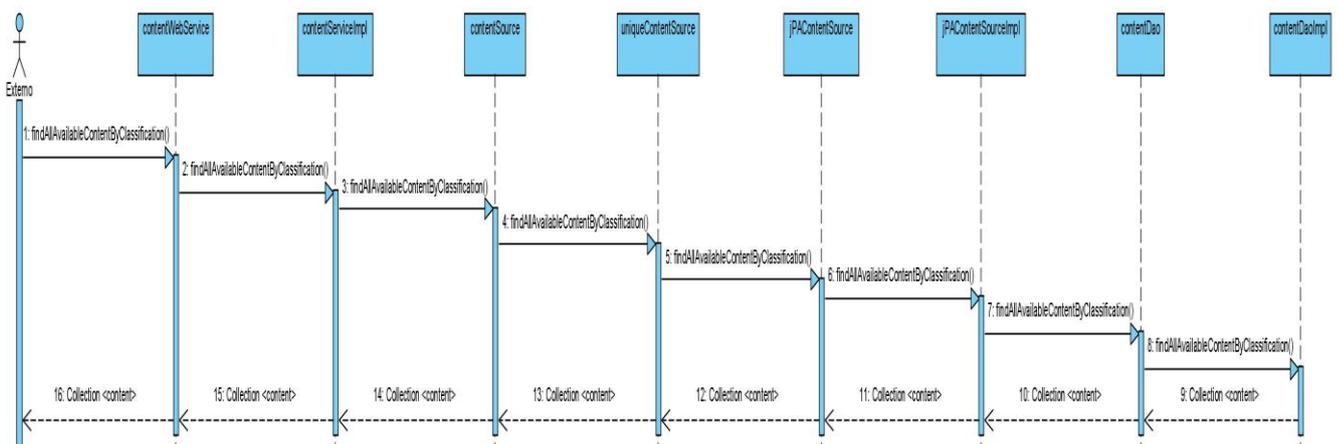
CU Buscar Contenido. Escenario Buscar todos los contenidos disponibles por autor



CU Buscar Contenido. Escenario Buscar todos los contenidos disponibles por proveedor

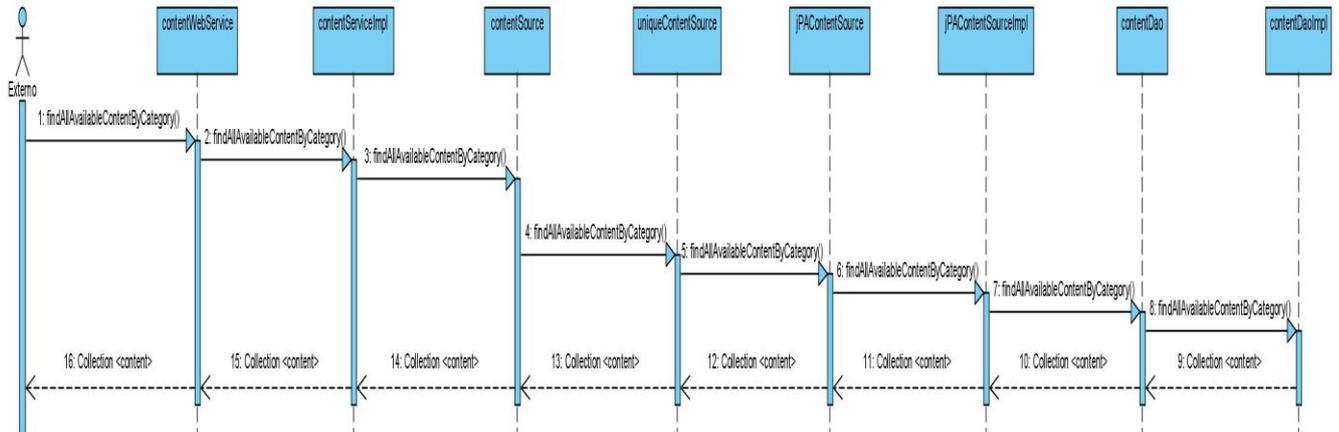


CU Buscar Contenido. Escenario Buscar todos los contenidos disponibles por clasificación

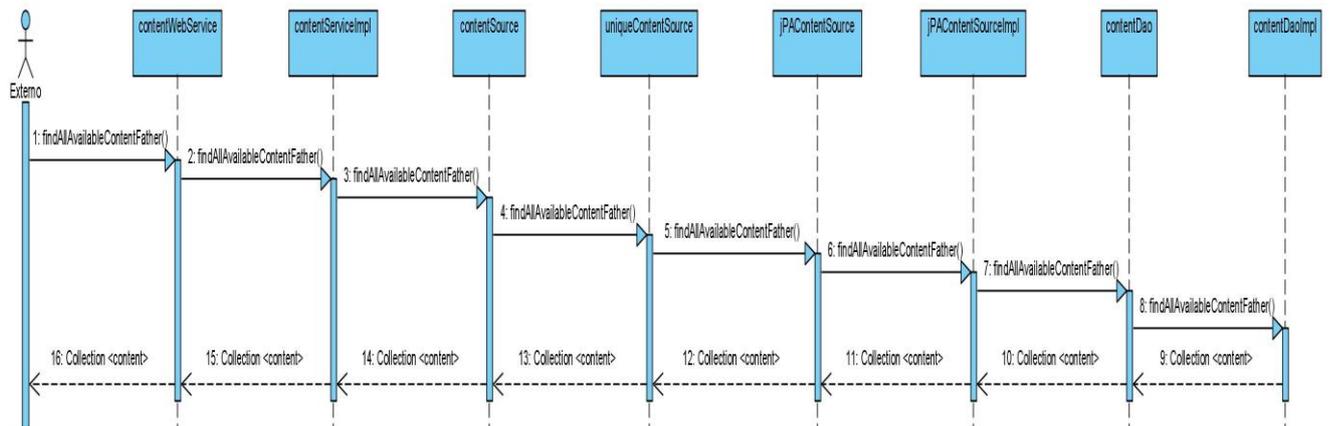


## Anexo III: Diagramas de Secuencias por Caso de Uso

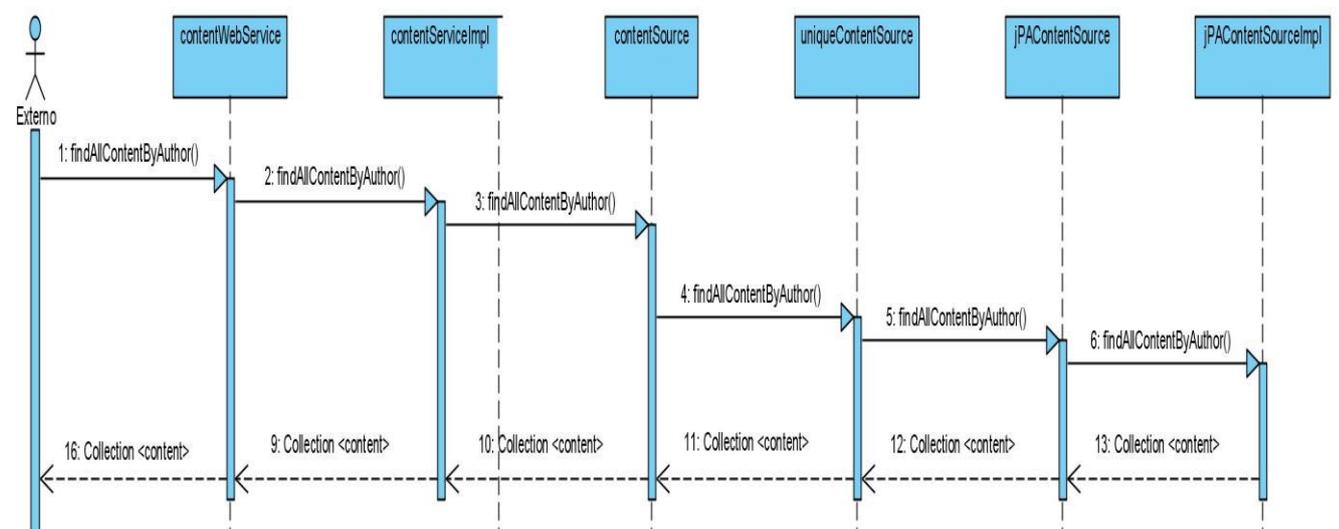
CU Buscar Contenido. Escenario Buscar todos los contenidos disponibles por categoría



CU Buscar Contenido. Escenario Buscar todas las categorías disponibles

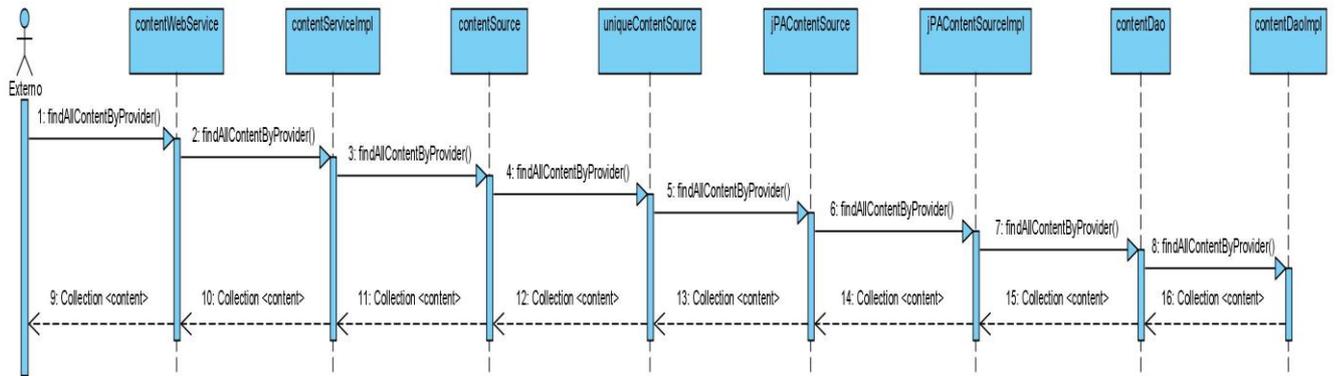


CU Buscar Contenido. Escenario Buscar todos los contenidos por autor

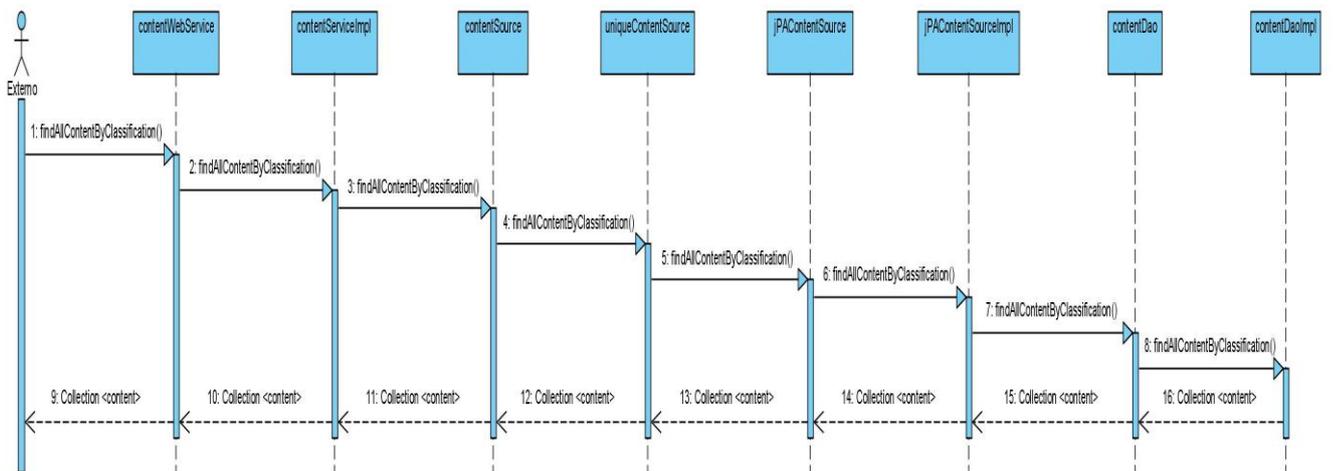


## Anexo III: Diagramas de Secuencias por Caso de Uso

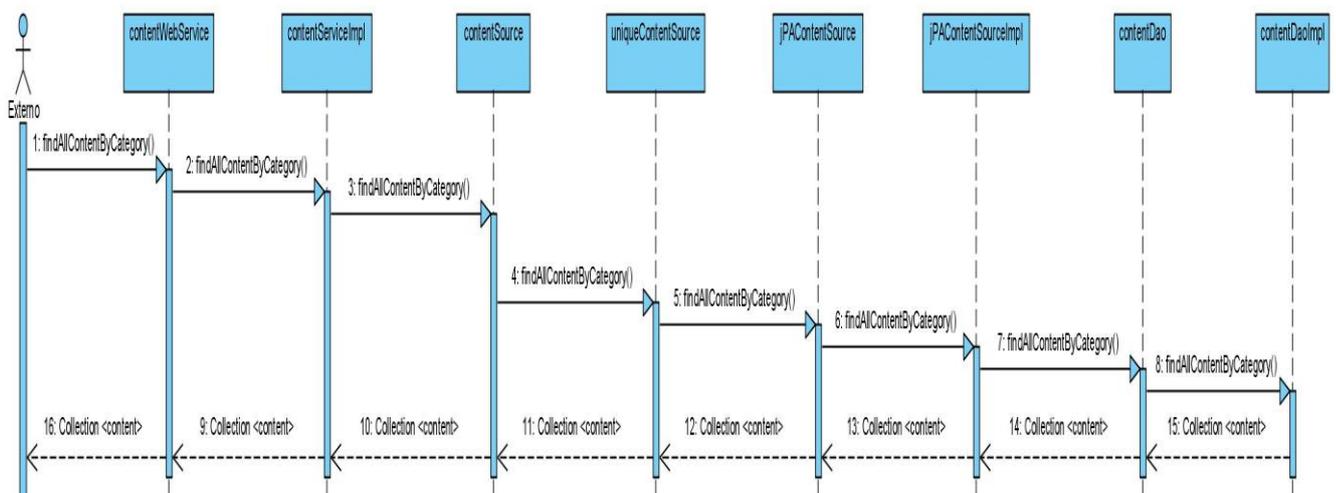
CU Buscar Contenido. Escenario Buscar todos los contenidos por proveedor



CU Buscar Contenido. Escenario Buscar todos los contenidos por clasificación

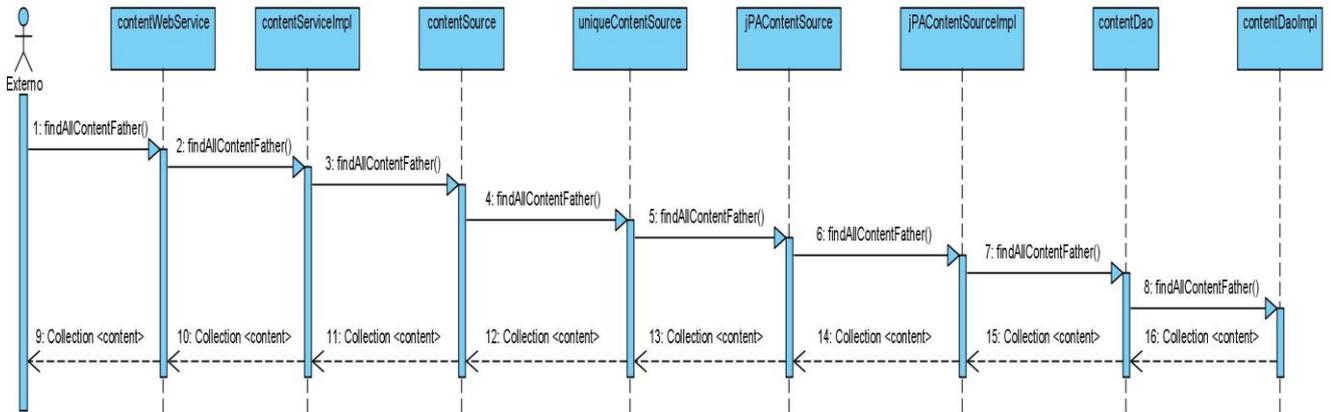


CU Buscar Contenido. Escenario BuscarSource todos los contenidos por categoría

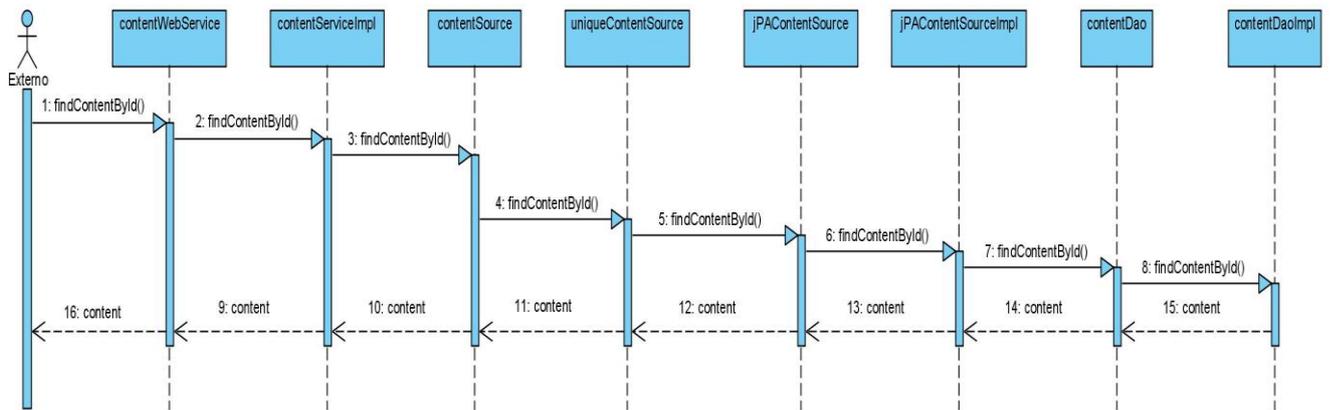


## Anexo III: Diagramas de Secuencias por Caso de Uso

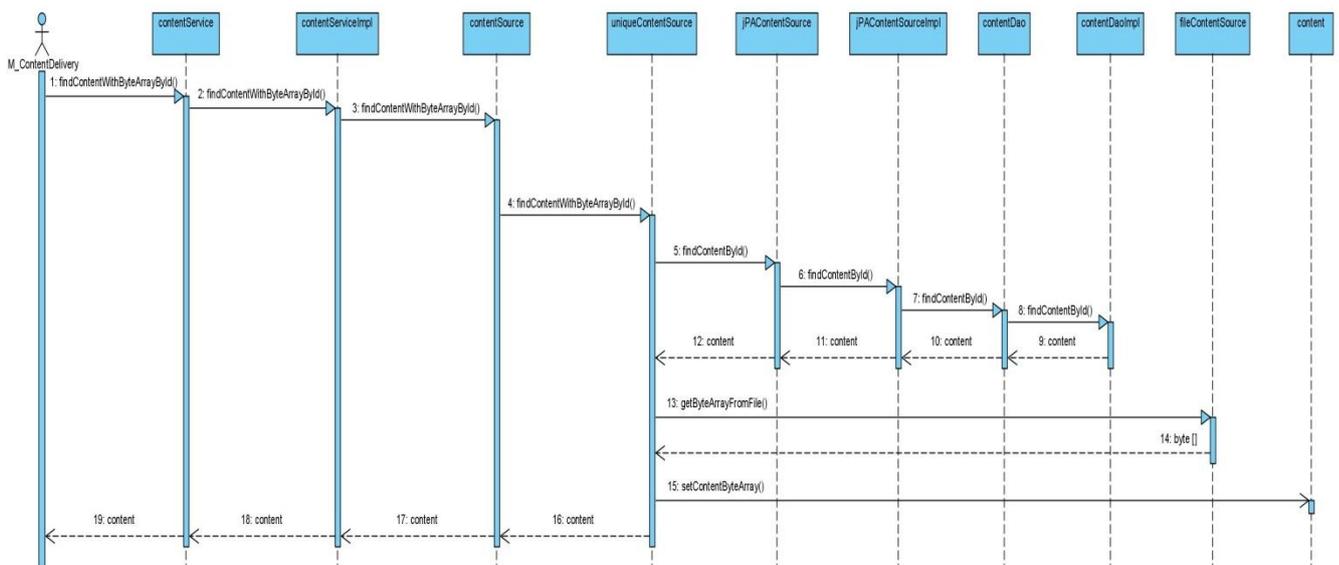
### CU Buscar Contenido. Escenario Buscar todas las categorías



### CU Buscar Contenido. Escenario Buscar contenido por ID

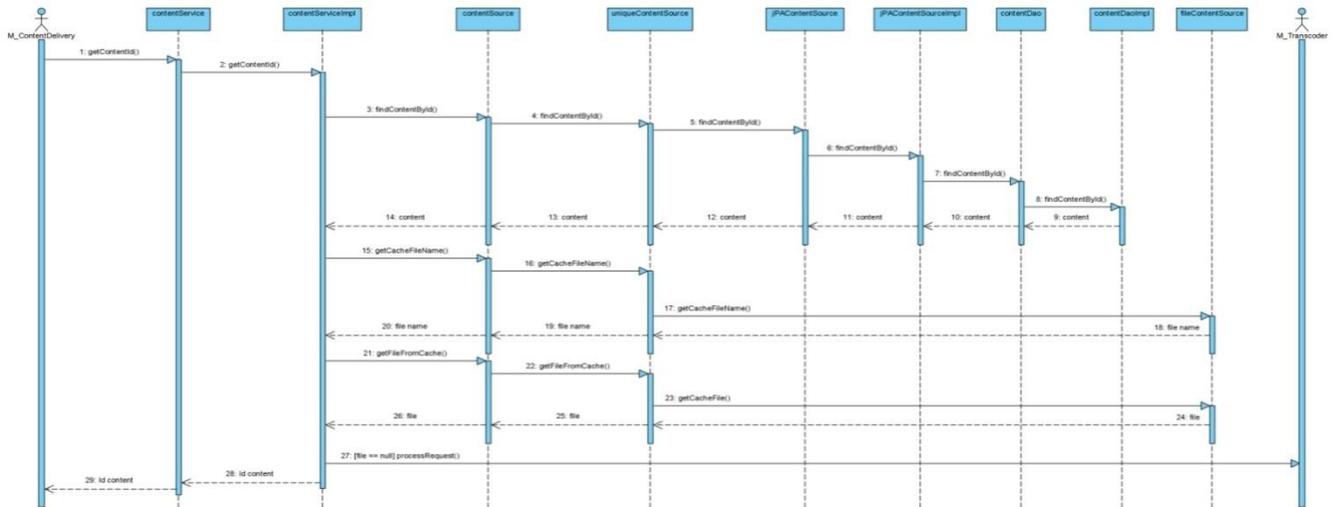


### CU Buscar Contenido. Escenario Buscar contenido con arreglo de datos

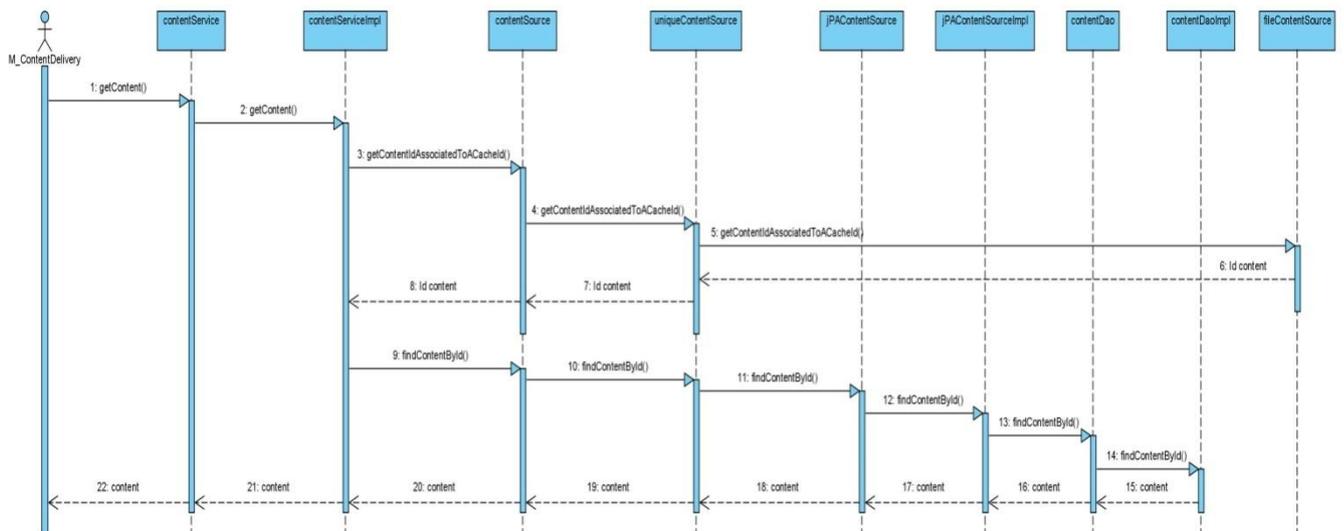


## Anexo III: Diagramas de Secuencias por Caso de Uso

### CU Obtener contenido. Escenario Obtener ID del contenido



### CU Obtener contenido. Escenario Obtener contenido por ID



### CU Obtener contenido. Escenario Obtener archivo de la caché

