

Universidad de las Ciencias Informáticas

Facultad 2



Título: Sistema de Administración de Contenido para móviles.

Trabajo de Diploma para optar por el título de
Ingeniero Informático

Autores: Kariné Ramos Blanco.
Dariena Ramirez Luján.

Tutor: Ing. Damián Fernández López.

Ciudad de la Habana, 5 de Julio de 2007

DECLARACIÓN DE AUTORÍA

Declaramos ser autoras de este trabajo y autorizamos a la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmo la presente a los 5 días del mes de julio del año 2007.

Autoras:

Tutor:

Kariné Ramos Blanco

Dariena Ramirez Luján

Ing. Damián Fernández López

Nunca consideres el estudio como una obligación, sino como una oportunidad para penetrar en el bello y maravilloso mundo del saber.

Albert Einstein

AGRADECIMIENTOS

Quiero agradecer a mi familia por estar siempre a mi lado, apoyándome.

A Alejandro por enseñarme a amar mi vocación.

A Dariena (mi compañera de tesis y mi mejor amiga), por las peleas fraternales que permitieron el éxito de este trabajo.

A mi tutor, que siempre nos apoyó, y que más que tutor es un amigo.

A los amigos que me apoyaron y ayudaron a la realización de este trabajo.

Kariné Ramos Blanco.

A mi madre, por enseñarme a luchar cuando se quiere algo de verdad.

A mis abuelos, Estrella y Eugenio, por creer en mí, por quererme tanto.

A mi familia, tía Estrellita, Liudmila, Karina, Ramoncito y Miguel por el apoyo y la confianza.

A Kariné, mi mejor amiga, mi hermana, mi consejera, la persona más correcta y sensata del mundo.

A mi amigo y tutor, Damián, por enseñarme tanto, por tantas conversaciones.

A mis amigos Raida, Del Toro, Adis, Katty, Ely, Isis, Yuni y Damiancito, por apoyarme, por estar, por reír y llorar juntos.

A todos mis amigos del pre y la universidad, porque han sido compañeros en esta lucha.

Dariena Ramirez Luján.

DEDICATORIA

A mis padres, que les debo tanto... , la vida.

A mi mamá por ser una mujer espectacular, mi ejemplo a seguir, la mejor hija, la mejor esposa y la mejor madre del mundo.

A mi papá, por su firmeza, por su apoyo en todos los tiempos, por forjar mi carácter, por hacerme una mujer de bien, por su amor.

A mi hermanito, por ser mi amigo, mi apoyo, y mi motivo para seguir adelante para trazarle un camino honroso y digno, quiero que siempre esté orgulloso de su hermana que lo ama tanto.

A mi tía Ondina, que siempre ha sido otra madre para mí.

A Alejandro, por entenderme poco y quererme tanto.

Kariné Ramos Blanco.

A Siry, mi "mami", por darme amor, confianza, sueños, esperanza, vida.

Por creer en mí y ser la fuerza que mueve mi mundo.

Por formarme con esfuerzo, dedicación y paciencia.

Por ser mi madre y mi mejor amiga.

Dariena Ramirez Luján.

RESUMEN

La tecnología de teléfonos móviles ha tomado un gran auge que va en crecimiento a través de los años, por lo que la demanda de contenido en este ámbito ha provocado que la administración de estos se haya convertido en una rama importante de desarrollo en el mercado mundial.

En la empresa Procyon se hicieron investigaciones en las que se obtuvo como resultado, que existe la necesidad de administrar los contenidos para móviles y a partir de esto se decidió desarrollar un Sistema de Administración de Contenido para móviles basado en servicios web.

Los servicios web representan la solución con más futuro para la integración de aplicaciones en Internet, aportan interoperabilidad entre aplicaciones de software independientemente de sus propiedades o de las plataformas sobre las que se instalen, además, permiten que servicios y software de diferentes compañías ubicadas en diferentes lugares geográficos puedan ser combinados fácilmente para proveer servicios integrados.

El sistema consta de 3 servicios web separados por funcionalidad: Management, Provider y Reseller. El servicio web de Management se encarga de la administración general de la aplicación; el Provider está dirigido a los proveedores de contenidos, dándole permiso para adicionar, editar, eliminar y publicar contenido y el Reseller permite al usuario final (móvil) acceder al contenido.

El sistema cuenta actualmente con una aplicación web que permite interactuar con el servicio web Management, siendo este módulo (el servicio web Management y la aplicación web) el tema del presente trabajo. Estos forman parte de la totalidad del sistema que incluye los servicios web Provider y Reseller y las aplicaciones web que accedan a estos servicios.

ÍNDICE

Introducción	1
Capítulo 1: Fundamentación Teórica	5
Introducción	5
1.1 Introducción a los Sistemas de Administración de Contenidos.	5
1.2 Estado del arte de los Sistemas de Administración de Contenidos.....	6
1.3 Lenguajes de Programación	7
1.3.1 Java.....	7
1.3.2 C#.....	8
1.4 Tecnologías a utilizar	8
1.4.1 J2EE (Java 2 Enterprise Edition)	8
1.4.2 Servicios Web	9
1.4.3 ASP.NET.....	10
1.5 Metodología de desarrollo.....	10
1.5.1 RUP (Rational Unified Process).....	12
1.5.2 UML: Unified Modeling Language.....	13
Conclusiones	14
Capítulo 2: Características del Sistema	15
Introducción	15
2.1 Propuesta de sistema.	15
2.2 Modelo de dominio.....	16
2.3 Relación de los Requerimientos.	18
2.3.1 Requerimientos Funcionales.....	18
2.3.2 Requerimientos no Funcionales.	20
2.4 Modelo de Casos de Uso del Sistema.	22
2.4.1 Definición de los Actores.....	22
2.4.2 Listado de Casos de Uso. Descripción breve.	22
2.4.3 Diagrama de Casos de Uso.	24
2.4.4 Casos de Uso expandidos.	25
Conclusiones	26

Capítulo 3: Análisis y Diseño del Sistema	27
Introducción	27
3.1 Análisis.....	27
3.1.1 Diagramas de Clases del Análisis.....	27
3.2 Diseño.....	33
3.2.1 Diagramas de Interacción.....	33
3.2.2 Diagramas de Clases del Diseño.....	34
Conclusiones	46
Capítulo 4: Implementación	47
Introducción	47
4.1 Implementación.....	47
4.1.1 Diagrama de Despliegue.....	47
4.1.2 Diagramas de Componentes.....	48
Conclusiones	55
Capítulo 5. Estudio de Factibilidad	57
Introducción	57
5.1 Estimación por Puntos de Casos de Uso.....	57
5.1.1 Cálculo de Puntos de Casos de Uso sin ajustar	58
5.1.2 Cálculo de Puntos de Casos de Uso ajustados.....	59
5.1.3 De los Puntos de Casos de Uso a la estimación del esfuerzo.....	60
5.2 Beneficios tangibles e intangibles.....	63
5.3 Análisis de costo	63
Conclusiones	64
Conclusiones	65
Recomendaciones	66
Bibliografía	67
Referencias Bibliográficas	70
Glosario de Términos	72

INTRODUCCIÓN

La tecnología celular ha aumentado notablemente a través de los años, comenzó su difusión cerca de 1995 y ya en el año 2001 había alcanzado un índice de penetración social cerca del 60-80%, más que lo que el teléfono fijo había alcanzado en cientos de años desde su inicio (1876). Así, se puede observar un proceso de difusión extremadamente rápido, necesitando solamente cerca de seis años para finalizar. Esta impresionante velocidad no se puede comparar con la velocidad de difusión de ningún otro dispositivo tecnológico durante el mismo período o en cualquier momento antes (incluyendo la expansión de las PC y las conexiones a Internet).

La evolución del teléfono celular ha permitido disminuir su tamaño y peso, desde ese primer teléfono en 1983 que pesaba 780 gramos, a los actuales más compactos y con mayores prestaciones de servicio. El desarrollo de baterías más pequeñas y de mayor duración, pantallas más nítidas y de colores y la incorporación de software más amigable, hacen del teléfono celular un elemento muy apreciado en la vida moderna.

El avance de la tecnología ha hecho que estos aparatos incorporen funciones que no hace mucho parecían futuristas, como juegos, reproducción de música MP3, correo electrónico, SMS (del inglés Short Message Service), agenda electrónica PDA (del inglés Personal Digital Assistant), fotografía digital, navegación por Internet y hasta televisión digital.

Todo lo expuesto anteriormente tiene como consecuencia que la demanda de contenido incremente exponencialmente y a su vez la administración de estos se convierta en una necesidad requiriendo la creación de sistemas que automaticen este proceso. Las empresas comercializadoras de contenidos para celulares administran millones de estos, por lo que se hace necesario un proceso de organización y categorización. Estas empresas cuentan ya con una infraestructura que gestiona sus procesos de negocio, por lo que adquirir un nuevo software podría chocar con los estándares ya implementados.

Procyon es una empresa que se dedica a producir contenidos y software para teléfonos móviles y pretende entrar al mercado internacional. Al ser una empresa joven en este tipo de desarrollo, ha analizado las necesidades que se están presentando en el mundo actual de la telefonía móvil y uno de los

problemas detectados es la administración de los contenidos (tonos, imágenes, animaciones, juegos, temas, etc.).

A partir de esta situación se plantea el siguiente **problema**: cómo crear un Sistema de Administración de Contenidos que cumpla los requerimientos de las empresas comercializadoras de contenidos.

La empresa Procyon decidió crear un Sistema de Administración de Contenidos para teléfonos móviles capaz de administrar sus proveedores, revendedores, usuarios, contenidos, categorías de contenidos, categorías de precio, entre otros. El sistema utilizaría la tecnología de servicios web para facilitar la integración de las aplicaciones en uso por aquellas empresas que se dedican a este tipo de negocio. Además, se lograría que tuviera funcionalidades de interoperabilidad, permitiendo que se puedan brindar los servicios a más de una empresa a la vez, manteniendo la administración del sistema en Procyon. De esta forma estas empresas no tendrían que preocuparse ni por el soporte ni por el mantenimiento.

Como antecedente, en la empresa se implementó un Sistema de Administración de Contenidos en el cual la capa de presentación y la lógica de negocio estaban montadas sobre JSP (Java Server Page) y Porlets (objetos que producen fragmentos de código dinámicamente), la capa de servicio o acceso a datos usaba EJB (del inglés Enterprise Java Bean) y como gestor de base de datos MySQL. Esta plataforma inicial era muy abarcadora y obligaba a modificar las aplicaciones que podrían tener los operadores o a comprar funcionalidades que no necesitaban. Este sistema tuvo limitaciones comerciales, por lo que no tuvo éxitos en el mercado.

Con el desarrollo de esta aplicación se espera cumplir con las expectativas trazadas inicialmente de integrarse al mercado internacional, construyendo un sistema que permita eficazmente administrar contenidos y desarrollar el resto de las funcionalidades propuestas.

Objetivo General:

- Creación de servicios web para la administración de contenidos y de una aplicación web que permita interactuar con estos.

Objetivos Específicos:

- Estudiar el estado del arte de los sistemas de administración de contenidos.
- Estudiar metodologías de desarrollo de software.
- Estudiar tecnologías y lenguajes que se adapten a la posible solución.
- Diseñar e implementar un servicio web que permita:
 - Gestionar contenidos.
 - Gestionar proveedores.
 - Gestionar revendedores.
 - Gestionar usuarios.
 - Gestionar marcas de agua.
 - Gestionar categorías de contenido y de precio.
- Diseñar e implementar una aplicación web que permita interactuar con el servicio web.

Tareas de la investigación:

- Selección de una metodología de desarrollo que se adecue al desarrollo del sistema.
- Selección del lenguaje de programación para el desarrollo del sistema.
- Selección de las herramientas a utilizar para el desarrollo del sistema.
- Diseño e implementación del servicio web y la aplicación web.

Estructuración del contenido.

Capítulo 1: Fundamentación Teórica

El capítulo 1 trata principalmente los aspectos metodológicos del trabajo, los objetivos generales y específicos, el campo de acción, objeto de estudio y las tareas de la investigación que dan solución a los objetivos.

Capítulo 2: Características del Sistema

El capítulo 2 presenta la propuesta de sistema, los requerimientos funcionales y los no funcionales. Se desarrolla el modelo de dominio y se definen los casos de uso.

Capítulo 3: Análisis y Diseño del Sistema

El capítulo 3 se centra en el análisis y diseño del sistema, obteniendo como resultado principal, los diagramas de clases del diseño y los diagramas de interacción.

Capítulo 4: Implementación

El capítulo 4 expone los diagramas de componentes y la distribución física del sistema a través del diagrama de despliegue conformando así la implementación.

Capítulo 5: Estudio de Factibilidad

En el capítulo 5 se hace una estimación del costo del sistema, calculando un aproximado del tiempo de desarrollo y la cantidad de desarrolladores necesarios para cumplir ese tiempo planificado.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

Introducción

En este capítulo se describen las tecnologías y metodología a utilizar, los lenguajes de programación elegidos y se expone el estado del arte de los sistemas de administración de contenidos.

1.1 Introducción a los Sistemas de Administración de Contenidos.

Un Sistema de Administración de Contenido es una herramienta que permite a compañías o individuos crear y modificar el contenido de su página web, con poco o nada de conocimiento técnico. Aún cuando la mayoría de los Sistemas de Administración de Contenido comparten algunas características básicas, principalmente la capacidad para corregir y publicar contenido a través de un navegador o una aplicación y proporcionar el acceso a usuarios múltiples.

Los sitios que proveen información a partir de bases de datos, son ahora el estándar en Internet, y los Sistemas de Administración de Contenido utilizan este mismo principio. Las herramientas para manejo de contenido típicamente trabajan en conjunto con una base de datos que es utilizada para almacenar documentos, texto e imágenes.

El Sistema de Administración de Contenido es en sí una interfase a una base de datos que permite a los usuarios autorizados adicionar, editar, publicar y eliminar un contenido. Un aspecto crucial en el uso de esta tecnología, es que el contenido que se ingresa a un Sistema de Administración de Contenido es reutilizable. Almacenar información en una base de datos significa que su contenido puede ser reutilizado en muchos lugares dentro de su sitio web y ajustado a diversos formatos para cualquier dispositivo: teléfono móvil, PDA e incluso para impresión.

Un sistema de administración de contenidos es muy confiable, está centralizado y en la mayoría de los casos se opera a través de navegadores. La información puede ser accedida propiamente desde cualquier computadora. Esto significa que el contenido no tiene que ser transferido de un departamento a otro para

ser aprobado. Puede ser accedido directamente por los diferentes responsables a medida que el nuevo material se agrega. [1]

1.2 Estado del arte de los Sistemas de Administración de Contenidos.

En Cuba, un país del 3er mundo, no abundan los teléfonos celulares, por lo que la demanda de contenido para estos dispositivos es bastante baja y no existe ningún Sistema de Administración de Contenido para móviles.

En el sector del mercado Internacional que abastece a las medianas y grandes empresas, existen varios competidores. Entre los productos líderes en el mercado de soluciones de los Sistemas de Administración de Contenido enfocados a los grandes consorcios internacionales, se encuentran Vignette, FatWire Content Server y Microsoft. [2]

Las soluciones de gestión de contenidos empresariales de Vignette permiten a las empresas aprovechar al máximo sus archivos, documentos, páginas web, imágenes, e-mails, elementos multimedia y otros contenidos no estructurados. [3]

FatWire Content Server es otra solución, 100% J2EE, que debido a su larga trayectoria en el mercado de gestión de contenidos presenta gran estabilidad y madurez como herramienta. [4]

Algunos de estos competidores permiten que se adquiera solamente los componentes que se necesiten. El precio básico de estos productos comienza en los \$30,000 USD, pero los sistemas más avanzados pueden llegar a costar más de \$100,000 USD. Desde luego, hay soluciones confeccionadas a base de módulos que son más baratas. Existen en el mercado una serie de Sistema de Administración de Contenido, cuyo costo puede comenzar alrededor de los \$1,000 USD por una licencia de 10 usuarios para la administración de un solo dominio.

1.3 Lenguajes de Programación

La empresa Procyon tiene varios principios para el desarrollo de software. Entre ellos están: que el producto desarrollado sea multiplataforma, logrando un despliegue del sistema sin que el tipo de sistema operativo constituya un obstáculo; implementado con código abierto, que permite su comercialización sin trabas legales y que facilite el desarrollo de sistemas empresariales, por lo que se decidió utilizar como plataforma de implementación a Java.

Para el desarrollo de la capa de presentación del sistema se decidió usar la plataforma .NET, que es una plataforma propietaria pero tiene grandes facilidades, una de ellas es proporcionar un breve tiempo de desarrollo, característica fundamental que motivó a la empresa a utilizarla.

1.3.1 Java

La tecnología Java está compuesta básicamente por 2 elementos: el lenguaje Java y su plataforma. Java es un lenguaje de alto nivel, orientado a objetos, multiplataforma, es decir, que permite la ejecución de un mismo programa en múltiples sistemas operativos, como dice el axioma de Java, “write once, run everywhere”. El sistema de Java es seguro porque tiene ciertas políticas que evita que se puedan codificar virus. Es un lenguaje que soporta múltiples hilos, capaz de ejecutar diferentes líneas de código al mismo tiempo. Es distribuido ya que contiene un conjunto de clases que permiten la comunicación a través de la red facilitando así la creación de aplicaciones distribuidas. Es interpretado y compilado a la vez, su código fuente se transforma en código de máquina (bytecode), y este a su vez en código nativo. Es un lenguaje robusto pues fue diseñado para crear software altamente fiable, en él se ha prescindido por completo de los punteros, y el recolector de basura (garbage collector) elimina la necesidad de liberación explícita de memoria.

Tiene una plataforma, que a diferencia de las demás que son una combinación de hardware y software, está basada en software, que corre sobre cualquier hardware. Consta de 2 componentes: la máquina virtual de java (JVM) y la interfaz de programación de aplicaciones (API). [3] [4]

1.3.2 C#

C# es el lenguaje insignia de la plataforma .NET, es un lenguaje muy bien diseñado y con muchas virtudes, posee herramientas de programación para todos los gustos. Es un lenguaje moderno y orientado a objetos, con una sintaxis muy similar a la de C++ y Java. Combina la alta productividad de Visual Basic con el poder y la flexibilidad de C++. La misma aplicación que se ejecuta bajo Windows podría funcionar en un dispositivo móvil de tipo PDA. C# gestiona automáticamente la memoria, y de este modo evita los problemas de programación tan típicos en lenguajes como C o C++. Mediante la plataforma .NET, desde la cual se ejecuta, es posible interactuar con otros componentes realizados en otros lenguajes .NET de manera muy sencilla. También es posible interactuar con componentes no gestionados fuera de la plataforma .NET. Por ello, puede ser integrado con facilidad en sistemas ya creados. Desde C# se puede acceder a una librería de clases muy completa y muy bien diseñada, que permite disminuir en gran medida los tiempos de desarrollo. [5]

1.4 Tecnologías a utilizar

En el mundo de la programación existe una enorme cantidad de tecnologías que satisfacen los requerimientos de cualquier proyecto por complejos que estos sean.

En el sistema se usó ASP.NET para el desarrollo de la aplicación cliente pues inicialmente es una aplicación con fines demostrativos, con el objetivo de reflejar el funcionamiento del servicio web. Una de las razones más influyentes para la utilización de esta tecnología es el breve tiempo de desarrollo que este proporciona, pues el plazo de entrega al cliente era corto.

Se usó J2EE como plataforma para el desarrollo del servicio web. Cada una de estas tecnologías se acopla exactamente a las necesidades de cada módulo del proyecto.

1.4.1 J2EE (Java 2 Enterprise Edition)

Java 2 Enterprise Edition es una plataforma de programación para desarrollar y ejecutar software de aplicaciones en el lenguaje de programación Java con arquitectura de n niveles, distribuida, basándose ampliamente en componentes de software modulares ejecutados sobre un servidor de aplicaciones.

Incluye varias especificaciones de API, tales como EJB, Servlets (código en Java que se ejecuta en un servidor web), Portlets, JSP y varias tecnologías de servicios web.

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

Uno de los beneficios de J2EE como plataforma es que es posible empezar con poco o ningún coste. La implementación J2EE de Sun Microsystems puede ser descargada gratuitamente, y hay muchas herramientas de código abierto, disponibles para extender la plataforma o para simplificar el desarrollo.

1.4.2 Servicios Web

Los servicios web, son servicios ofertados vía web, que representan la solución con más futuro para la integración de aplicaciones en Internet. Permiten una total independencia entre un servicio y su consumidor. De esta manera el cliente se abstrae de toda la tecnología que subyace detrás del servicio. La conexión al servicio, el mapeo de XML y otros aspectos son ignorados por el desarrollador encargado de alimentarse de estos servicios.

Funcionan de la siguiente manera: una aplicación cliente envía una petición vía HTTP a un servicio situado en una URL, el servicio recibe la petición, la procesa y devuelve una respuesta también sobre HTTP. Para esto se requiere un protocolo de intercambio de mensajes petición/respuesta sobre HTTP y una forma de que clientes y proveedores puedan interactuar a través de los mensajes.

Los servicios web utilizan SOAP (del inglés Simple Object Access Protocol) como estándar de comunicación, que mundialmente se ha utilizado como protocolo de intercambio de mensajes. Este es un protocolo sencillo basado en XML y estandarizado por el W3C (del inglés World Wide Web Consortium).

El lenguaje de especificación de interfaces utilizado en los servicios web es WSDL (del inglés Web Services Description Language). WSDL permite especificar en XML las operaciones y tipos de datos de un

servicio web. Así, aunque el cliente y el servidor estén escritos en lenguajes distintos pueden interactuar al utilizar un lenguaje neutral para comunicarse. [6]

1.4.3 ASP.NET

ASP.NET es un conjunto de tecnologías de desarrollo de aplicaciones web comercializado por Microsoft. Es usado por programadores para construir sitios web domésticos, aplicaciones web y servicios XML. Forma parte de la plataforma .NET de Microsoft y es la tecnología sucesora de la tecnología Active Server Pages (ASP).

ASP.NET incorpora un modelo declarativo a la programación Web: los controles de servidor funcionan en una página Web simplemente declarándolos. Cuando se carga la página ASP.NET, se instancian los controles listados en la página ASP y es responsabilidad del control emitir código HTML que el navegador pueda entender. Impone un cierto orden sobre el modelo de programación estándar ASP además de claramente separar la porción basada en script de una página web de su contenido.

ASP.NET posee:

- Rendimiento: la aplicación se compila una sola vez al lenguaje nativo, y luego, en cada petición tiene una compilación “Just In Time”, es decir, se compila desde el código nativo, lo que permite mucho mejor rendimiento. También permite el almacenamiento del caché en el servidor.
- Rapidez en programación: mediante diversos controles, se puede con unas pocas líneas y en menos de 5 minutos mostrar toda una base de datos y hacer rutinas complejas.
- Seguridad: tiene diversas herramientas que garantizan la seguridad de las aplicaciones.

1.5 Metodología de desarrollo

Para el desarrollo de este sistema se utilizó RUP como metodología de desarrollo.

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

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 (equipo de desarrollo, recursos, etc.) exigen que el proceso sea configurable.

Metodologías más utilizadas en el mundo:

- ✓ Rational Unified Process (RUP)
- ✓ XP (eXtreme Programming)
- ✓ MSF(Microsoft Solution Framework)

Uno de los aspectos más importantes a la hora de determinar que metodología utilizar es definir las características del proyecto y a partir de ahí seleccionar la que mejor se acoja a estas.

XP (Extreme Programming) pertenece a las Metodologías Ágiles, está basada en heurísticas provenientes de prácticas de producción de código. Está orientada a proyectos de corto plazo de entrega y pequeños equipos de desarrollo donde se esté especialmente preparado para cambios durante el proyecto. Posee procesos poco controlados, con pocos principios. No existe contrato tradicional o al menos es bastante flexible. El cliente es parte del equipo de desarrollo, se desarrollan pocos artefactos y roles y se hace menos énfasis en la arquitectura del software. [7]

Estas son algunas características de esta metodología, que no satisfacen las necesidades del proyecto debido a que se tiene un contrato prefijado que no permite esa flexibilidad que brinda XP, además el usuario final no forma parte del equipo de trabajo.

MSF es una metodología que consta de cinco fases:

- Visión y Alcances.
- Planificación.
- Desarrollo.
- Estabilización.
- Implantación.

Aunque mediante estas fases se logra desarrollar un proyecto con éxito, tienen poca interacción con el cliente debido a que en la primera fase se hace énfasis en la unificación del equipo detrás de una visión común donde es el equipo el que debe tener una visión clara de lo que quisiera lograr para el cliente, también se definen los líderes y responsables del proyecto, se identifican las metas y objetivos a alcanzar y se realiza la evaluación inicial de riesgos del proyecto. Todo esto provoca que en esta primera fase se lleven a cabo una serie de actividades que desvían la atención de lo que realmente importa, que es lo que el cliente necesita (enfaticando que en la mayoría de las ocasiones ni siquiera el propio cliente sabe lo que quiere), el dedicar poco tiempo a la captura de requisitos provoca que el resultado final no sea lo que realmente se necesitaba.

En este punto RUP es una metodología para proyectos de largo alcance, pero es estructurable y puede ser modificado en dependencia de las necesidades del proyecto y de la experiencia del desarrollador en el uso de esta metodología, logrando que se ajuste también a proyectos con un corto tiempo de desarrollo. Es fácil de entender tanto por usuarios avanzados como por clientes con pocos conocimientos técnicos. Está basado en normas provenientes de estándares seguidos por el entorno de desarrollo. Tiene procesos mucho más controlados, con numerosas políticas/normas. Existe un contrato prefijado donde el cliente interactúa con el equipo de desarrollo mediante reuniones. La arquitectura del software es esencial y se expresa mediante modelos. Por todo lo expuesto anteriormente se usa RUP como metodología de desarrollo.

1.5.1 RUP (Rational Unified Process)

El Proceso Unificado de desarrollo de Software (RUP) es el resultado de varios años de desarrollo y uso práctico en el que se han unificado técnicas de desarrollo, a través del lenguaje UML (del inglés Unified Modeling Language), y el trabajo de muchas metodologías utilizadas por los clientes. La versión que se ha estandarizado se creó en 1998 y se conoció en sus inicios como Proceso Unificado de Rational 5.0, de ahí las siglas con las que se identifica a este proceso de desarrollo.

RUP está conformado por fases y flujos de trabajo. Las fases son inicio, elaboración, construcción y transición, los flujos de trabajo están comprendidos en 2 grupos, 6 de ingeniería: modelación del negocio,

requerimientos, análisis y diseño, implementación, prueba e instalación y 3 de soporte: administración del proyecto, administración de configuración y cambios y por último ambiente.

Características de RUP:

1. **Dirigido por casos de uso:** Los casos de uso reflejan lo que los usuarios futuros necesitan y desean, lo cual se capta cuando se modela el negocio y se representa a través de los requerimientos. A partir de aquí los casos de uso guían el proceso de desarrollo debido a que los modelos que se obtienen, como resultado de los diferentes flujos de trabajo, representan la realización de los casos de uso.
2. **Centrado en la arquitectura:** La arquitectura muestra la visión común del sistema completo en la que el equipo de proyecto y los usuarios deben estar de acuerdo, por lo que describe los elementos del modelo que son más importantes para su construcción. RUP se desarrolla mediante iteraciones, comenzando por los casos de uso relevantes desde el punto de vista de la arquitectura.
3. **Iterativo e Incremental:** RUP propone que cada fase se desarrolle en iteraciones. Una iteración involucra actividades de todos los flujos de trabajo, aunque desarrolla fundamentalmente algunos más que otros.

RUP establece actividades y criterios que conducen a un sistema desde su máximo nivel de abstracción (la idea en la cabeza del cliente), hasta su nivel más concreto (un programa ejecutándose en las instalaciones del cliente). UML ofrece la notación gráfica necesaria a RUP para representar los sucesivos modelos que se obtienen en este proceso. [8] [9]

1.5.2 UML: Unified Modeling Language

Para el modelado se usa UML. Este lenguaje de modelado aparece en 1995 como el Método Unificado, que posteriormente se reorienta para dar lugar al Unified Modeling Language (UML), la notación orientada a objetos más popular en la actualidad. UML es un lenguaje de modelado gráfico para visualizar, especificar, construir y documentar los artefactos de un sistema que involucra una gran cantidad de software.

Está compuesto por diversos elementos gráficos que se combinan para conformar diagramas. La finalidad de estos es presentar diversas perspectivas de un sistema, a las cuales se les conoce como modelo. Un modelo UML describe lo que supuestamente hará un sistema, pero no dice cómo implementar dicho sistema. [10]

Conclusiones

En este capítulo se analizó el estado del arte de los Sistemas de Administración de Contenido para móviles a nivel internacional, nacional y de la Universidad. Se analizaron las últimas tendencias tecnológicas, haciendo una investigación exhaustiva de las que pueden adaptarse al desarrollo del sistema, eligiendo finalmente la que suplía completamente las necesidades del proyecto. Este mismo proceso se llevó a cabo en la elección de los lenguajes y metodologías de desarrollo de software que sería preciso usar para dar solución al problema que se enfrenta.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

Introducción

A continuación se describe la propuesta de sistema. Se presentan los requerimientos funcionales y no funcionales, se definen los actores y casos de uso y se hace una descripción breve y otra detallada de estos.

2.1 Propuesta de sistema.

Descripción general.

Se propone el desarrollo de un sistema multicapa y distribuido que administre contenidos para móviles. La capa de presentación se desarrolló sobre ASP.NET y encapsula la lógica de navegación y el control de eventos de la interfaz, inicialmente se construyó con fines demostrativos, para visualizar las funcionalidades de la lógica del negocio.

La capa de la lógica del negocio se implementó a través del servicio web, que proporciona interoperabilidad al sistema. La capa de acceso a datos se implementó sobre EJBs, que proporcionan un modelo de componentes distribuido estándar para el lado del servidor, además, permiten comunicación remota. El gestor de base de datos utilizado es MySQL.

Se desarrolló un servicio web: Management. Desde este se administran todos los procesos, la gestión de usuarios, de proveedores, revendedores, palabras claves, marcas de agua, categorías de precio, categorías de contenido, los contenidos específicamente y los clientes (dispositivos móviles). Las palabras claves son un conjunto de caracteres que identifican un tipo de contenido. Por otra parte las marcas de aguas son imágenes semitransparentes que se sitúan delante de las imágenes a vender cuando son mostradas al cliente, estas se relacionan además con los revendedores. Las categorías de contenido

permiten agrupar el contenido por el tema que traten, por ejemplo: amor, deporte, naturaleza. El sistema cuenta también con las categorías de precio, que permiten enmarcar los precios en barato, medio y caro. También controla el acceso de todos los usuarios al sistema mediante las transacciones que recogen el historial de todo lo que sucede en este.

El sistema cuenta con una aplicación web como cliente del servicio web Management, que permite interactuar con el servicio y desde el cual, como se ha explicado, se pueden ejecutar todas las acciones.

Análisis comparativo

Anteriormente en procyon se había implementado un Sistema de Administración de Contenidos multicapa, la capa de presentación estaba implementada sobre JSP, Servlets (código en Java que se ejecuta en un servidor web) y Portlets. La capa de servicios se implementó a través de EJBs que utiliza MySQL como gestor de almacenamiento de datos.

Esta propuesta está basada en una idea novedosa, la creación de servicios web para proporcionar las funcionalidades anteriormente implementadas. Esto permite vender un sistema que provee interoperabilidad, permitiendo a cualquier tipo de aplicación utilizar los servicios web, lo que hace de esta propuesta una solución mucho más madura y adaptable a las necesidades del mundo actual en este tipo de aplicaciones. El nuevo sistema usa la misma capa de servicio del sistema anterior. Cuenta también con una aplicación web que funciona como interfaz del servicio web Management aunque el cliente, si lo desea, puede implementar su propia interfaz gráfica.

2.2 Modelo de dominio

En el sistema los procesos del negocio no están bien definidos, no son visibles y las fronteras no están bien establecidas. Sin embargo podemos identificar personas, eventos, transacciones y objetos involucrados en ese entorno que no está bien delimitado. Estos se relacionan en un diagrama que conforma el modelo de dominio. Ver figura 1.

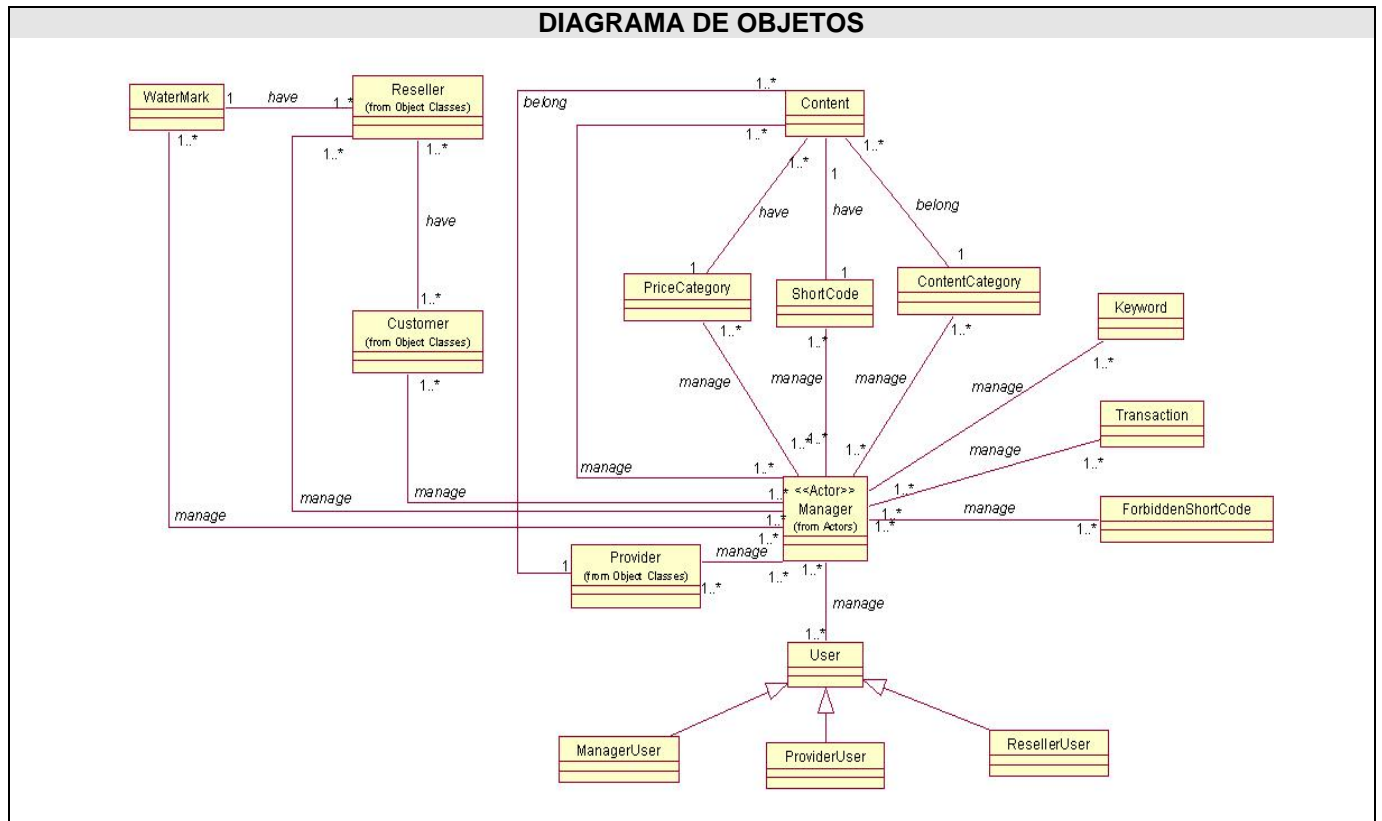


Figura 1. Diagrama de Objetos.

Descripción de los objetos del dominio

Manager: actor que gestiona todos los objetos.

Content: objeto que representa los contenidos que administrará el sistema.

Provider: objeto que representa la compañía proveedora de contenidos.

Reseller: objeto que representa la compañía que va a vender los contenidos.

User: objeto que representa a los usuarios del sistema, que se especializan en tres tipos, **ManagerUser**, **ProviderUser** y **ResellerUser**.

Content Category: objeto que representa una categoría de contenido, puede ser amor, deporte, naturaleza, etc.

Price Category: objeto que representa una categoría de precio, que enmarca un rango de precios, ejemplo: barato de cero a tres, medio de tres a cinco, etc.

Keyword: objeto que representa una palabra clave que está asociada a un tipo de contenido y tiene un conjunto de palabras con las que el cliente final puede referirse a un tipo de contenido, ejemplo: image, img, imag.

Watermark: objeto que representa una marca de agua, está formado por tres imágenes (media, clara, oscura) que se superponen sobre las imágenes a vender por determinado revendedor.

Customer: objeto que representa al cliente final que es un móvil.

Shortcode: es un código de cuatro caracteres creado dinámicamente que identifica un contenido cuando se va a descargar para el móvil.

Forbidden Shortcode: es un shortcode que no debe ser creado pues pueden formarse palabras desagradables.

Transaction: objeto que representa todas las acciones que se realizan sobre el sistema.

2.3 Relación de los Requerimientos.

Un requerimiento es una condición o capacidad que necesita un usuario para resolver un problema o lograr un objetivo. Todas las ideas que los clientes, usuarios y miembros del equipo de proyecto tengan acerca de lo que debe hacer el sistema, son candidatas a requisitos. Los requisitos se pueden clasificar en: funcionales y no funcionales. A continuación se relacionan los requerimientos funcionales y no funcionales capturados para el desarrollo del sistema.

2.3.1 Requerimientos Funcionales.

R1. Gestionar Contenido.

- 1.1 Permitir a los administradores adicionar contenido.
- 1.2 Permitir a los administradores modificar contenido.
- 1.3 Permitir a los administradores eliminar contenido.
- 1.4 Permitir a los administradores publicar contenido.

R2. Gestionar Usuario.

- 2.1 Permitir a los administradores adicionar un usuario.
- 2.2 Permitir a los administradores modificar un usuario.
- 2.3 Permitir a los administradores eliminar un usuario.

R3. Autenticar.

- 3.1 Permitir a los usuarios del sistema autenticarse.

R4. Gestionar Proveedor.

- 4.1 Permitir a los administradores adicionar un proveedor.
- 4.2 Permitir a los administradores modificar un proveedor.
- 4.3 Permitir a los administradores eliminar un proveedor.

R5. Gestionar Revendedor.

- 5.1 Permitir a los administradores adicionar revendedor.
- 5.2 Permitir a los administradores modificar revendedor.
- 5.3 Permitir a los administradores eliminar revendedor.

R6. Gestionar Palabra Clave.

- 6.1 Permitir a los administradores adicionar palabra clave.
- 6.2 Permitir a los administradores modificar palabra clave.
- 6.3 Permitir a los administradores eliminar palabra clave.

R7. Gestionar Shortcode no permitidos.

- 7.1 Permitir a los administradores adicionar un shortcode que no debe ser usado.
- 7.2 Permitir a los administradores modificar un shortcode que no debe ser usado.
- 7.3 Permitir a los administradores eliminar un shortcode que no debe ser usado.

R8. Gestionar Categoría de contenido.

8.1 Permitir a los administradores adicionar una categoría de contenido.

8.2 Permitir a los administradores modificar una categoría de contenido

8.3 Permitir a los administradores eliminar una categoría de contenido

R9. Gestionar Marca de Agua.

9.1 Permitir a los administradores adicionar una marca de agua.

9.2 Permitir a los administradores modificar una marca de agua.

9.3 Permitir a los administradores eliminar una marca de agua.

R10. Gestionar Categoría de Precio.

10.1 Permitir a los administradores adicionar una categoría de precio.

10.2 Permitir a los administradores modificar una categoría de precio.

10.3 Permitir a los administradores eliminar una categoría de precio.

R11. Gestionar Cliente.

11.1 Permitir a los administradores adicionar un cliente.

11.3 Permitir a los administradores eliminar un cliente.

2.3.2 Requerimientos no Funcionales.

Apariencia o interfaz externa.

El sistema debe implementar una interfaz web con un diseño sencillo y comunicarse a través del protocolo de comunicación http.

Usabilidad.

El sistema debe ser usado por cualquier administrador sin necesidad de un entrenamiento costoso.

Rendimiento.

El sistema debe ser eficiente, lograr un tiempo de respuesta mínimo, ya que de por si las aplicaciones web son desconectadas y tienen que acceder al servidor en busca de cualquier información por lo que la velocidad de procesamiento es un parámetro fundamental.

Soporte.

La empresa debe brindar soporte en dependencia de las demandas del comprador, puede ser mantenimiento, entrenamiento de personal, integración con otros softwares o a operar directamente el sistema.

Portabilidad.

El sistema debe ser independiente de la plataforma debido a que el servicio web está implementado en Java y la aplicación cliente es Web.

Seguridad

El sistema debe proteger la integridad de la información y los contenidos. No permitir el acceso no autorizado al sistema.

Para que se realice cualquier operación el usuario debe estar registrado. Todos los servicios web deben tener seguridad a través de cuentas de usuario con roles que definen el nivel de acceso.

Requerimientos Legales

Inicialmente no hay requerimientos legales pues se trabaja con código abierto precisamente para que a la hora de vender el sistema orientado a servicios no haya problemas de este tipo. Sin embargo la aplicación web que constituye la capa de presentación, implementada con tecnología propietaria, puede ser vendida si el comprador posee licencia.

Confiabilidad.

El sistema debe proteger la información y contenidos así como brindar un proceso de recuperación ante cualquier fallo.

2.4 Modelo de Casos de Uso del Sistema.

A continuación se presenta el modelo de casos de uso del sistema que comprende los actores del sistema, casos de uso que estos inicializan y sus descripciones y el diagrama de casos de uso del sistema.

2.4.1 Definición de los Actores.

Tabla 1. Definición de Actores.

Actores	Justificación
Manager	Representa a la persona que administra todos los procesos del sistema.

2.4.2 Listado de Casos de Uso. Descripción breve.

Tabla 2. Descripción del Caso de Uso Manage Content.

CU-1	Manage Content
Actor	Manager
Descripción	El actor decide que acción va a realizar (adicionar, modificar, eliminar). En el caso de adicionar o modificar, llena o actualiza los datos. Si es eliminar selecciona el contenido y lo elimina.
Referencia	R1

Tabla 3. Descripción del Caso de Uso Manage User.

CU-2	Manage User
Actor	Manager
Descripción	El actor decide que acción va a realizar (adicionar, modificar, eliminar). En el caso de adicionar o modificar, llena o actualiza los datos. Si es eliminar selecciona el usuario y lo elimina.
Referencia	R2

Tabla 4. Descripción del Caso de Uso Authenticate.

CU-3	Authenticate
Actor	Manager
Descripción	El actor se autentica en el sistema.

Referencia	R3
-------------------	----

Tabla 5. Descripción del Caso de Uso Manage Provider.

CU-4	Manage Provider
Actor	Manager
Descripción	El actor decide que acción va a realizar (adicionar, modificar, eliminar). En el caso de adicionar o modificar, llena o actualiza los datos. Si es eliminar selecciona el proveedor y lo elimina.
Referencia	R4

Tabla 6. Descripción del Caso de Uso Manage Reseller.

CU-5	Manage Reseller
Actor	Manager
Descripción	El actor decide que acción va a realizar (adicionar, modificar, eliminar). En el caso de adicionar o modificar, llena o actualiza los datos. Si es eliminar selecciona el revendedor y lo elimina.
Referencia	R5

Tabla 7. Descripción del Caso de Uso Manage Keyword.

CU-6	Manage Keyword
Actor	Manager
Descripción	El actor decide que acción va a realizar (adicionar, modificar, eliminar). En el caso de adicionar o modificar, llena o actualiza los datos. Si es eliminar selecciona la palabra clave a eliminar y la elimina.
Referencia	R6

Tabla 8. Descripción del Caso de Uso Manage Forbidden Shortcode.

CU-7	Manage Forbidden Shortcode
Actor	Manager
Descripción	El actor inserta los shortcode que no deben ser usados o elimina uno ya creado.
Referencia	R7

Tabla 9. Descripción del Caso de Uso Manage Content Category.

CU-8	Manage Content Category
Actor	Manager
Descripción	El actor decide que acción va a realizar (adicionar, modificar, eliminar). En el caso de adicionar o modificar, llena o actualiza los datos. Si es eliminar selecciona la categoría de contenido y la elimina.

Referencia	R8
-------------------	----

Tabla 10. Descripción del Caso de Uso Manage Watermark.

CU-9	Manage Watermark
Actor	Manager
Descripción	El actor decide que acción va a realizar (adicionar, modificar, eliminar). En el caso de adicionar o modificar, llena o actualiza los datos. Si es eliminar selecciona la marca de agua y la elimina.
Referencia	R9

Tabla 11. Descripción del Caso de Uso Manage Price Category.

CU-10	Manage Price Category
Actor	Manager
Descripción	El actor decide que acción va a realizar (adicionar, modificar, eliminar). En el caso de adicionar o modificar, llena o actualiza los datos. Si es eliminar selecciona la categoría de precio a eliminar y la elimina.
Referencia	R10

Tabla 12. Descripción del Caso de Uso Manage Customer.

CU-11	Manage Customer
Actor	Manager
Descripción	El actor decide que acción va a realizar (adicionar o eliminar). En el caso de adicionar llena los datos. Si es eliminar selecciona el cliente (móvil) y lo elimina.
Referencia	R11

2.4.3 Diagrama de Casos de Uso.

El siguiente diagrama de casos de uso representa al actor Manager y todos los casos de uso del sistema que este inicializa.

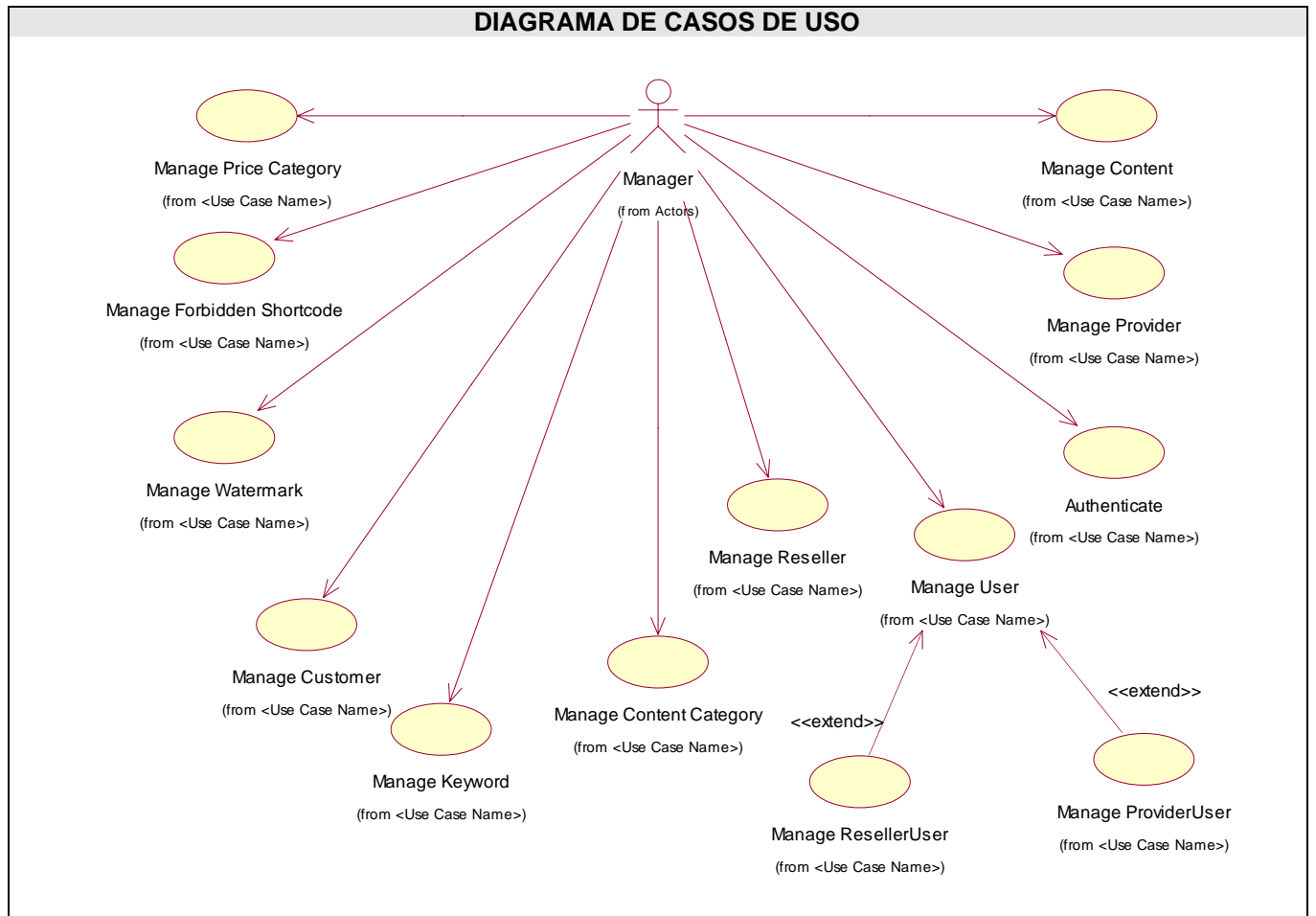


Figura 2. Diagrama de Casos de Uso del Sistema

2.4.4 Casos de Uso expandidos.

La descripción detallada de los casos de uso del sistema permite entender la funcionalidad asociada a cada caso de uso. Estas descripciones se representan con sus respectivas pantallas asociadas. (Ver anexo 1).

Conclusiones

En este capítulo se realizó una descripción de la solución propuesta, se presentó el modelo de dominio y las condiciones o capacidades que debe cumplir el sistema, resumidas en los requisitos funcionales y las propiedades o cualidades que el producto debe tener, recogidas en los requisitos no funcionales. Se desarrolló el Diagrama de Casos de Uso, y su descripción detallada.

CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

Introducción

En este capítulo se describen los procesos que se llevan a cabo en el flujo de trabajo de análisis y diseño. Se organizan los diagramas que describen el análisis y diseño de las clases, así como los que representan la interacción entre objetos, siguiendo el criterio de distribución por caso de uso.

3.1 Análisis.

El modelo de análisis ofrece un poder expresivo y una formalización para describir los aspectos del sistema, proporciona una estructura centrada en el mantenimiento, específicamente la flexibilidad ante los cambios y la reutilización. El modelo de análisis puede considerarse una primera aproximación al modelo de diseño aunque es un modelo por si mismo. Es importante también hacer notar que en el modelo de análisis se hacen abstracciones para evitar resolver algunos problemas que es mejor posponer al diseño y a la implementación. [11]

3.1.1 Diagramas de Clases del Análisis.

Los diagramas de clases del análisis representan los conceptos en un dominio del problema, sin llegar al nivel de detalle necesario que se persigue en el diseño, constituyendo una primera aproximación de este.

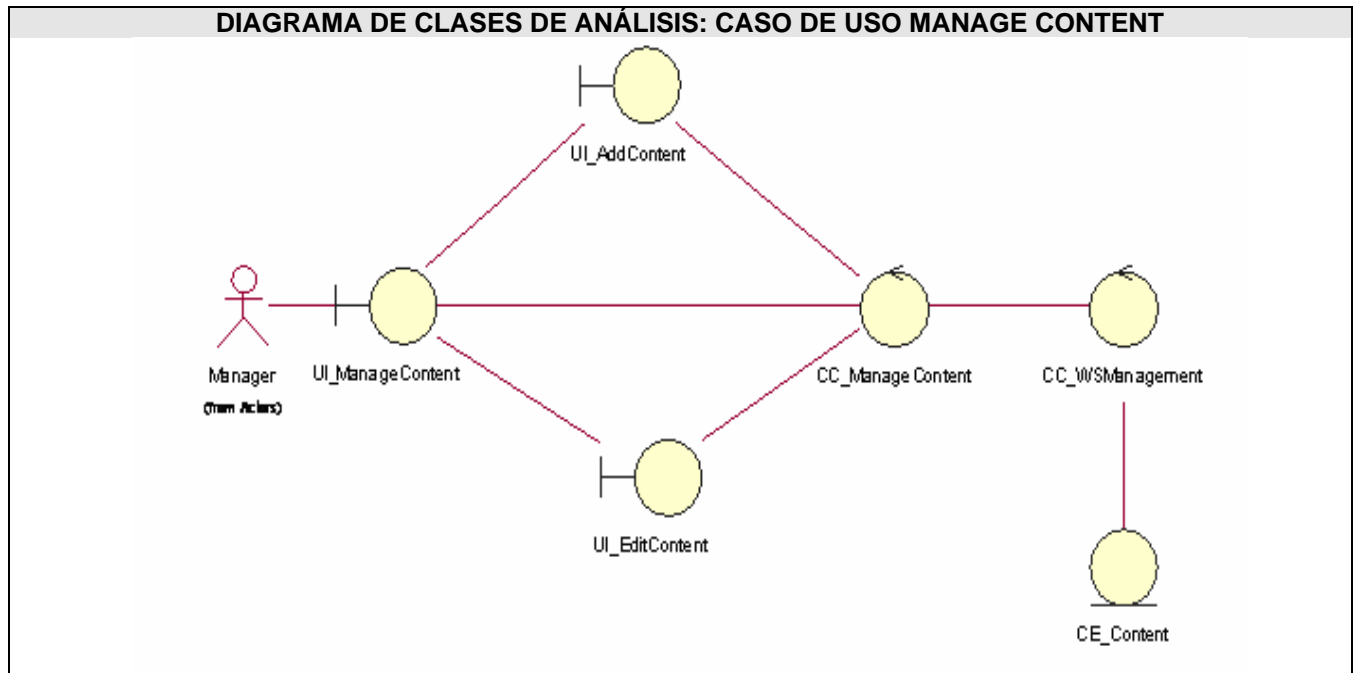


Figura 3. Diagrama de Clases del Análisis del Caso de Uso Manage Content.

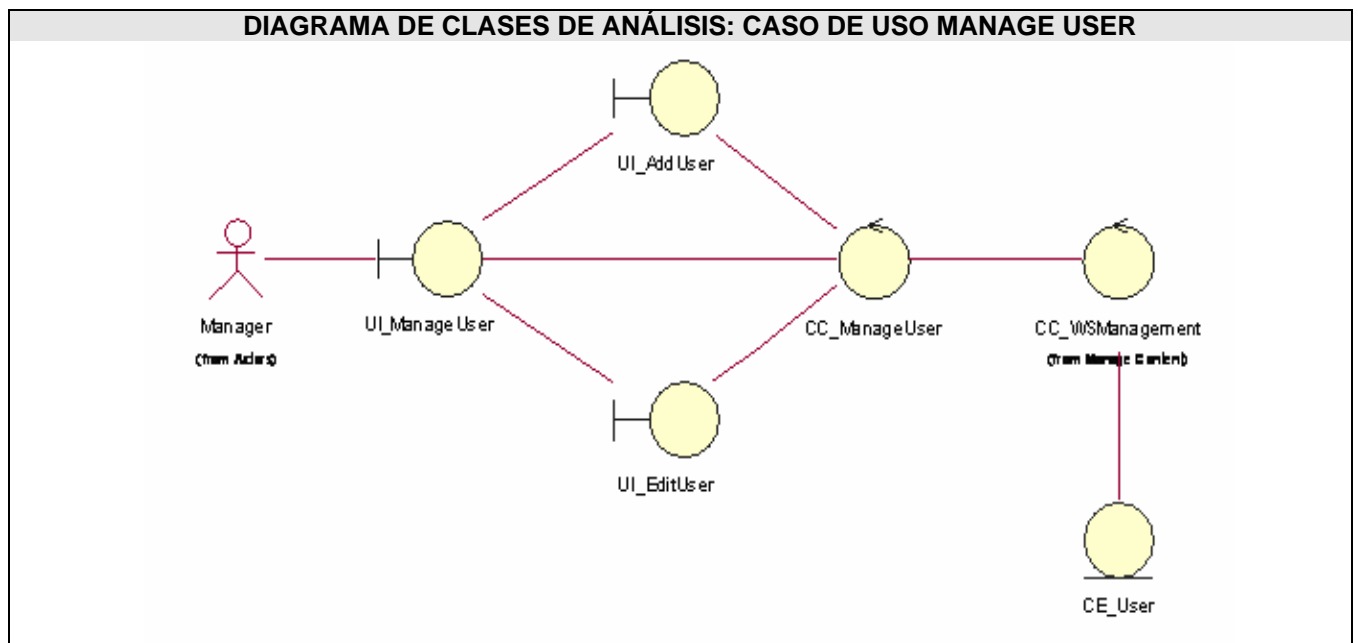


Figura 4. Diagrama de Clases del Análisis del Caso de Uso Manage User.

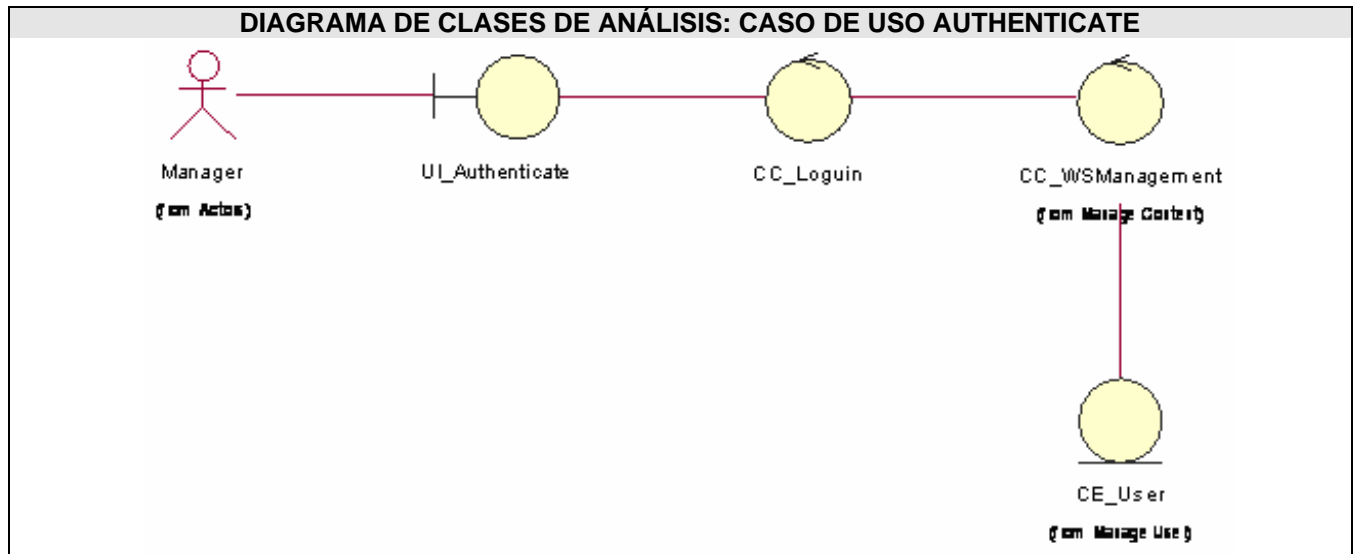


Figura 5. Diagrama de Clases del Análisis del Caso de Uso Authenticate.

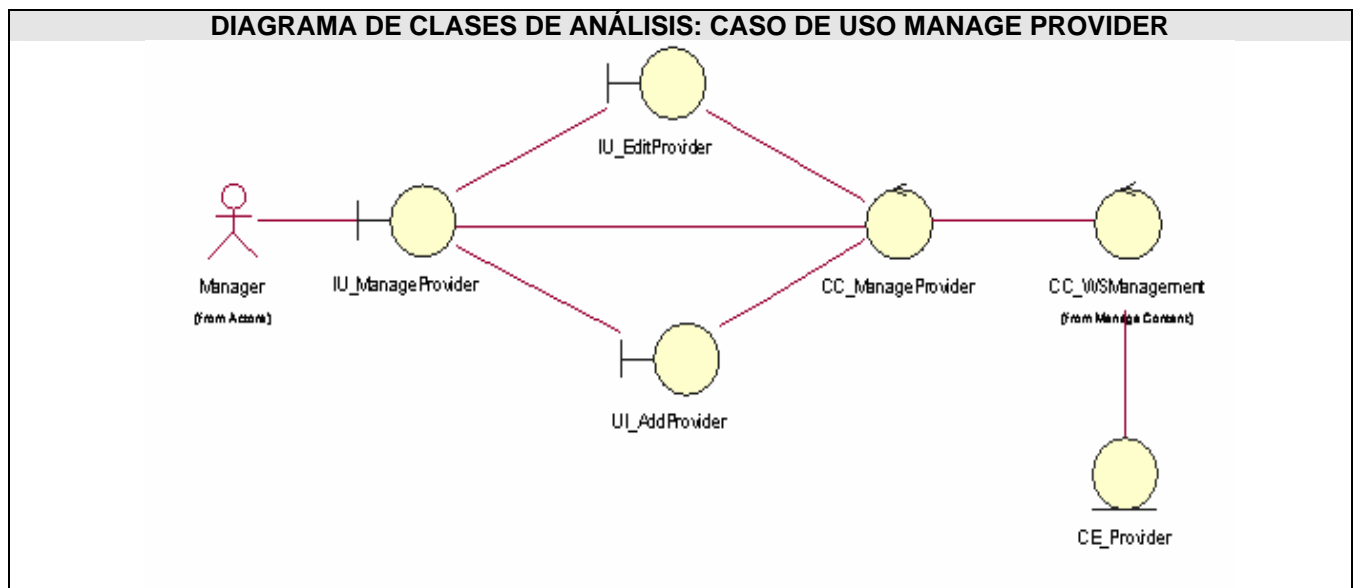


Figura 6. Diagrama de Clases del Análisis del Caso de Uso Manage Provider.

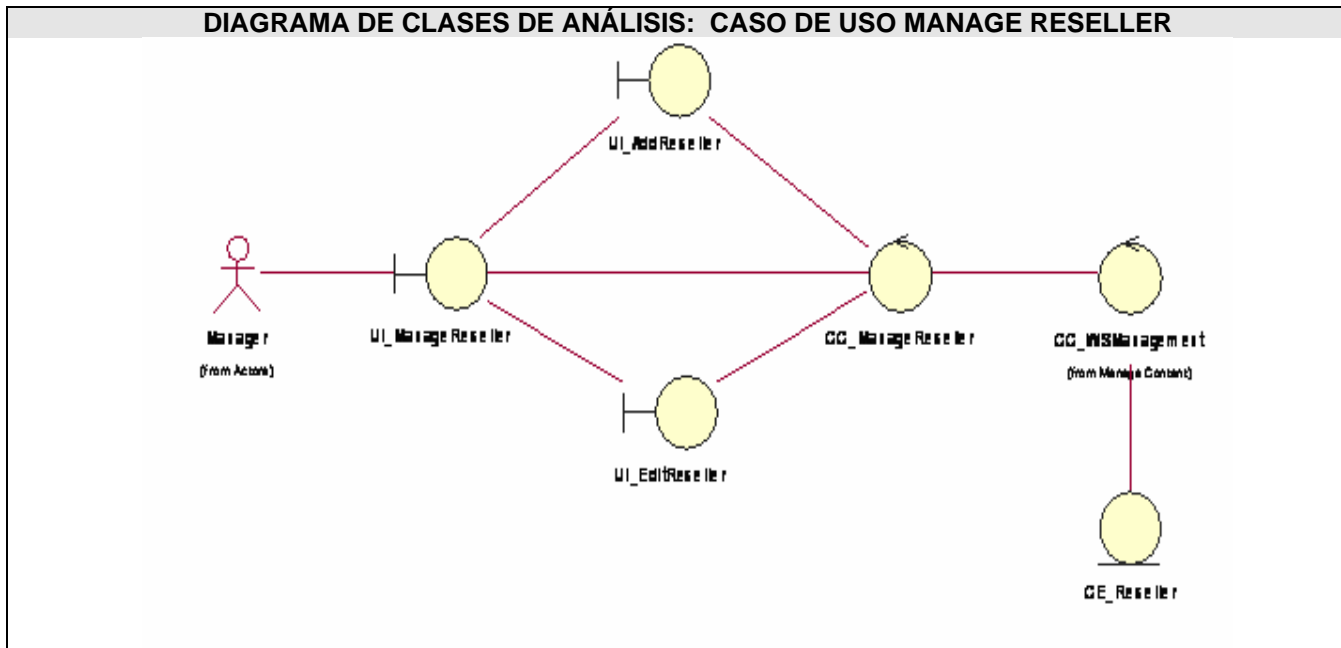


Figura 7. Diagrama de Clases del Análisis del Caso de Uso Manage Reseller.

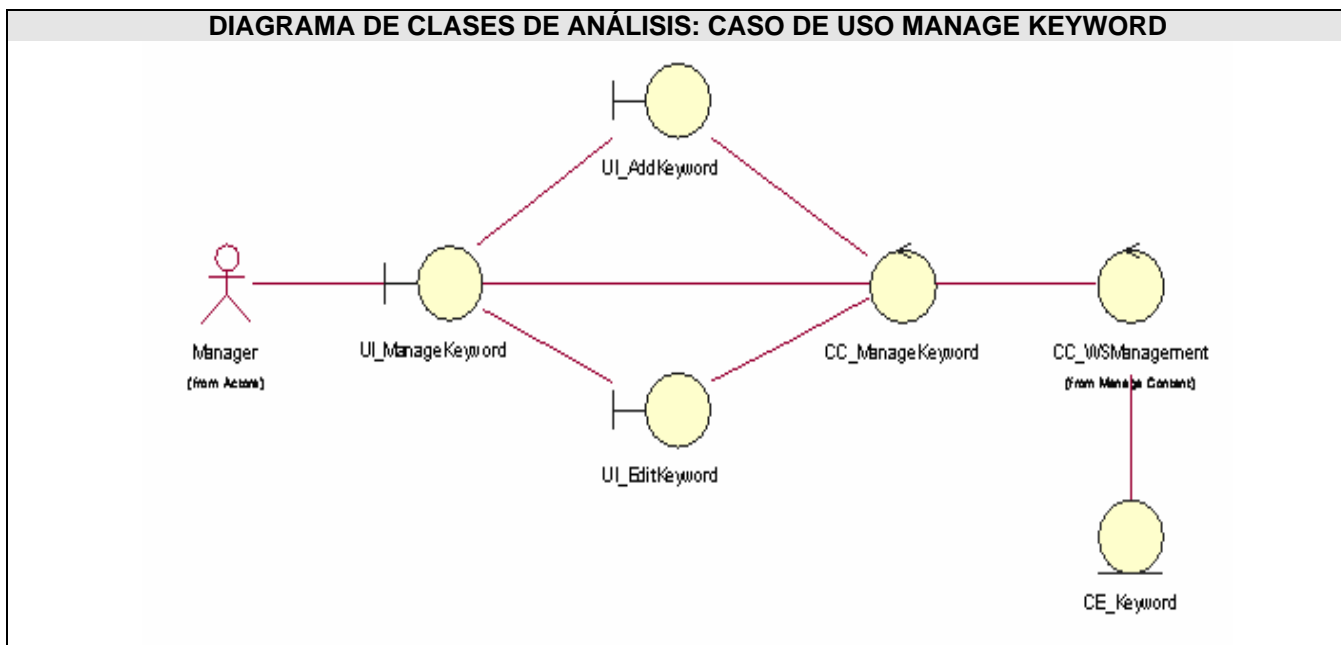


Figura 8. Diagrama de Clases del Análisis del Caso de Uso Manage Keyword.

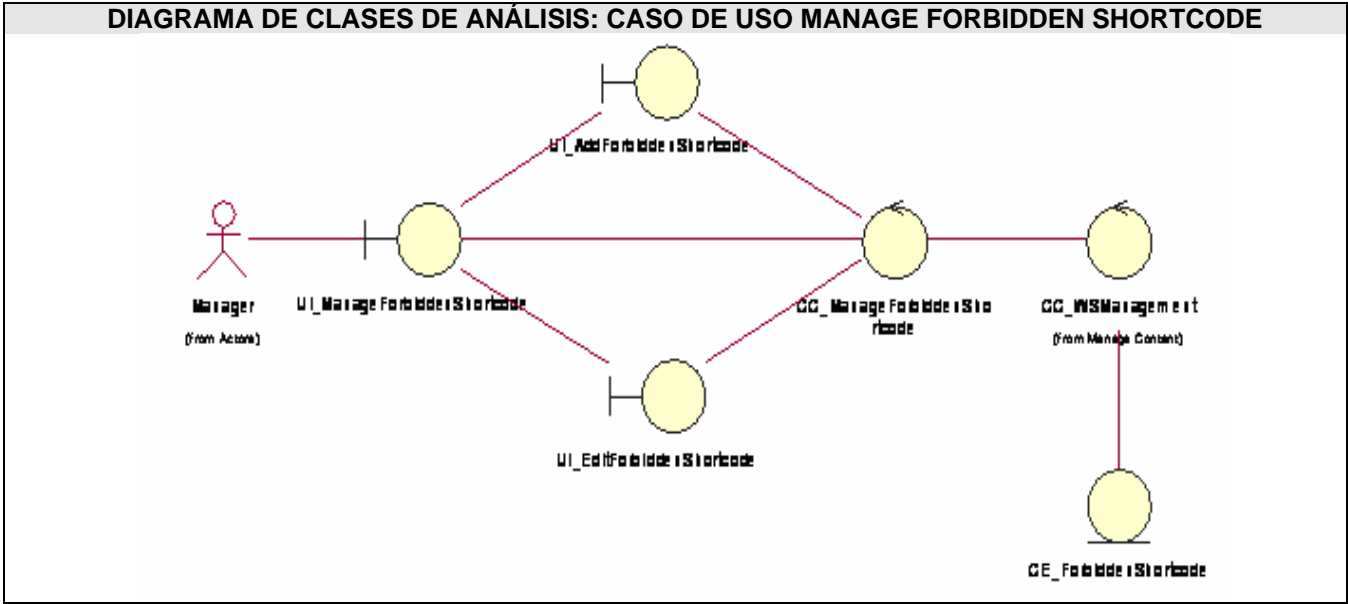


Figura 9. Diagrama de Clases del Análisis del Caso de Uso Manage Forbidden Shortcode.

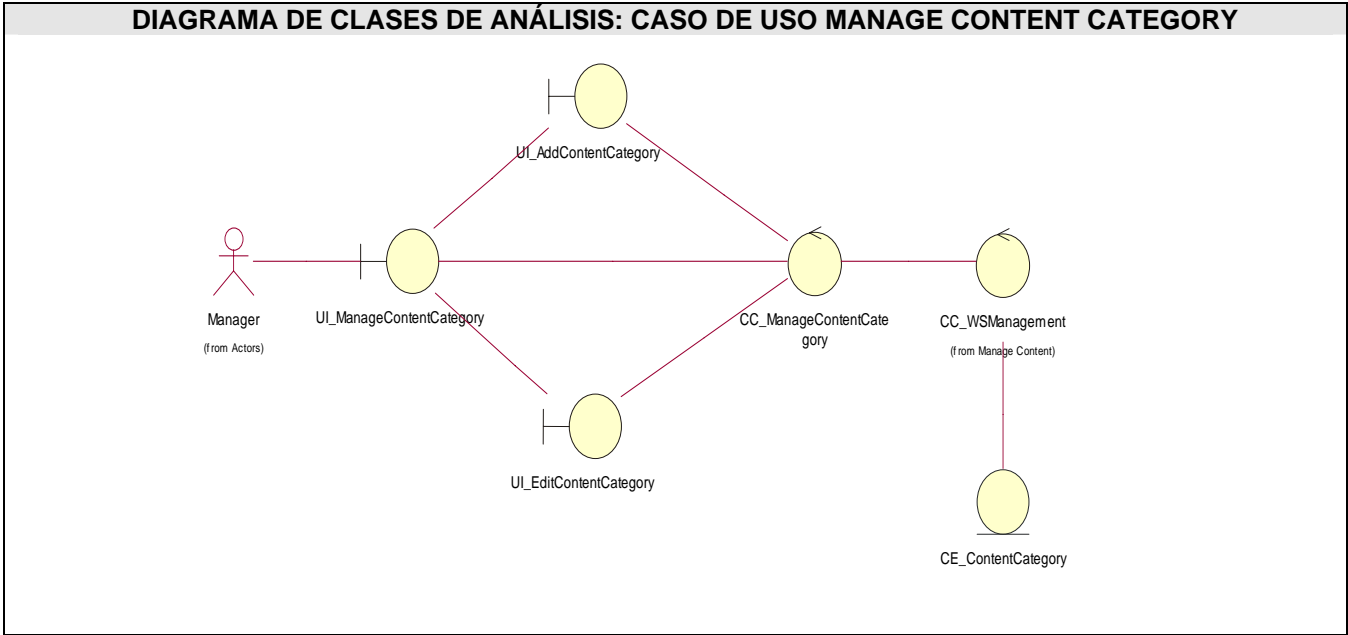


Figura 10. Diagrama de Clases del Análisis del Caso de Uso Manage Content Category.

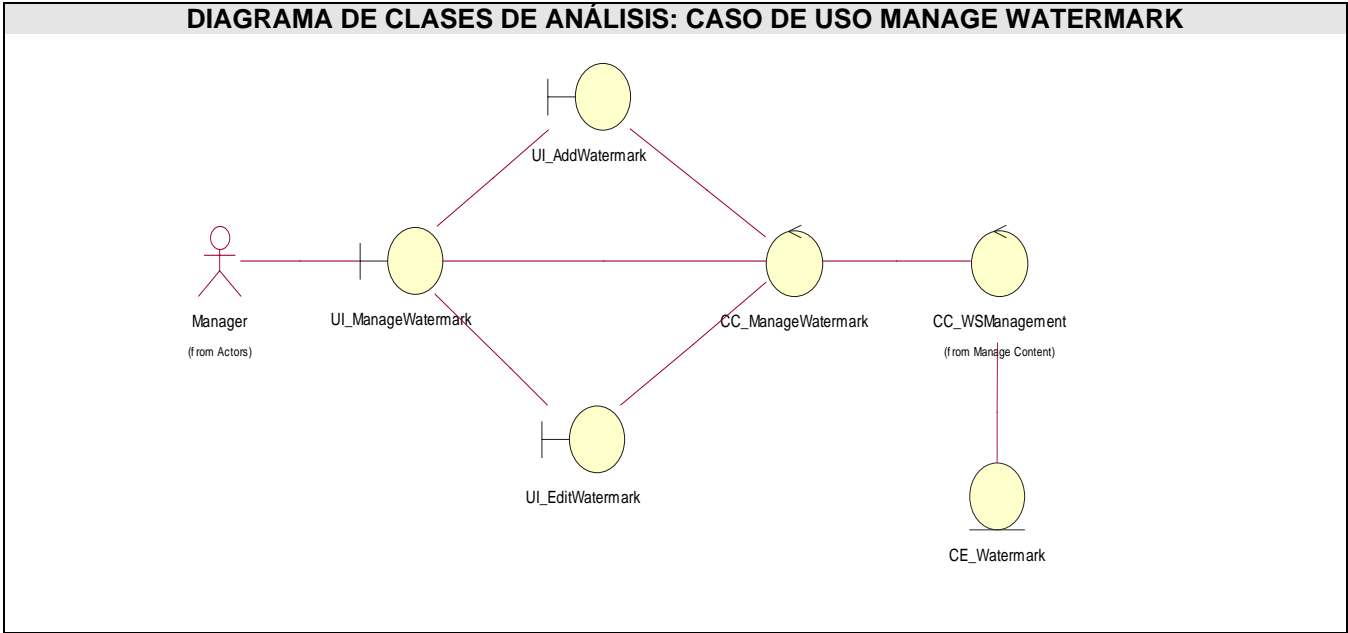


Figura 11. Diagrama de Clases del Análisis del Caso de Uso Manage Watermark.

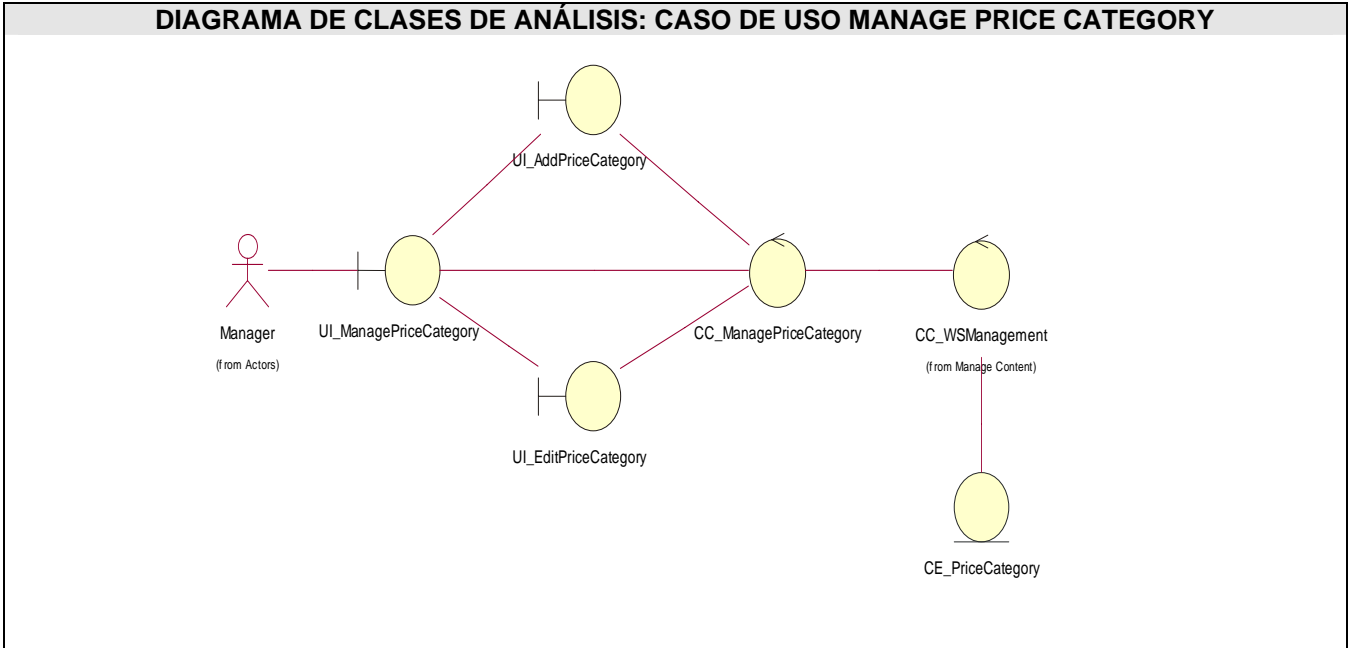


Figura 12. Diagrama de Clases del Análisis del Caso de Uso Manage Price Category.

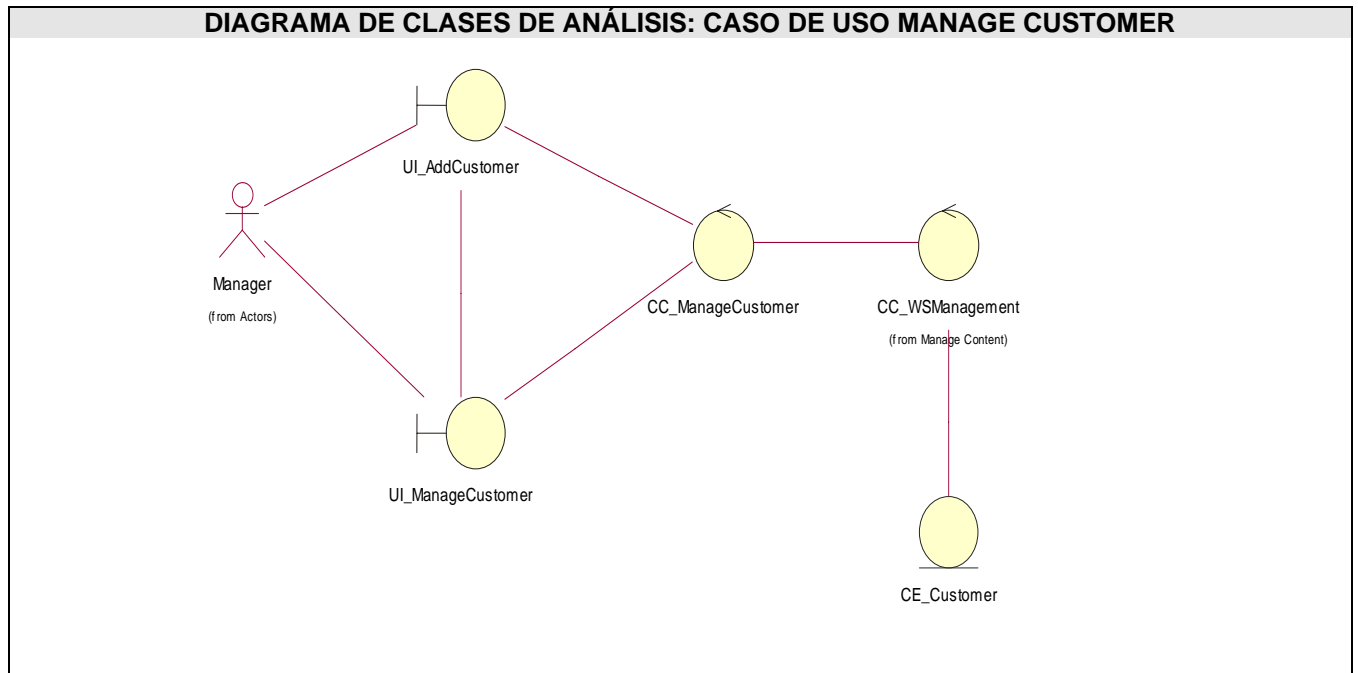


Figura 13. Diagrama de Clases del Análisis del Caso de Uso Manage Customer.

3.2 Diseño.

En el diseño se modela el sistema de manera que se cumplan los requisitos funcionales y no funcionales. Dentro de los propósitos del diseño están: adquirir una comprensión de los aspectos relacionados con los requisitos no funcionales y restricciones relacionadas con los lenguajes de programación, componentes reutilizables, sistemas operativos, tecnologías de distribución y concurrencia, ser capaces de descomponer los trabajos de implementación en partes más manejables que puedan ser llevadas a cabo por diferentes equipos de desarrollo. [12]

3.2.1 Diagramas de Interacción.

Los diagramas de interacción describen secuencias de intercambios de mensajes entre los objetos que implementan el comportamiento del sistema. Muestran el flujo de control a través de muchos objetos.

Un diagrama de secuencia muestra una interacción ordenada según la secuencia temporal de eventos. En particular, muestra los objetos participantes en la interacción y los mensajes que intercambian, ordenados según su secuencia en el tiempo. Ver Anexo 2.

3.2.2 Diagramas de Clases del Diseño.

Un diagrama de clases representa las clases que serán utilizadas dentro del sistema y las relaciones que existen entre ellas.

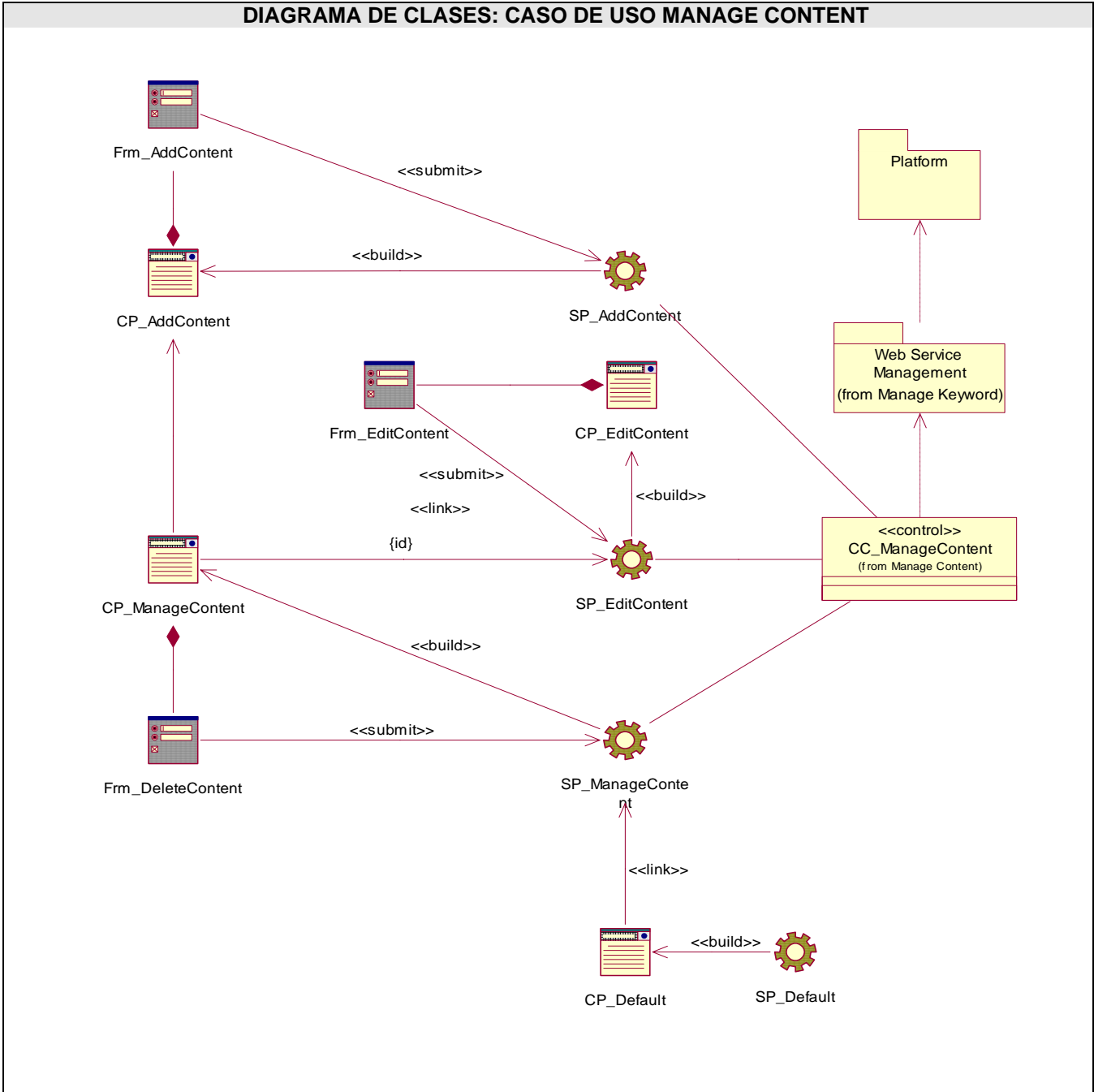


Figura 14. Diagrama de Clases del Diseño del Caso de Uso Manage Content.

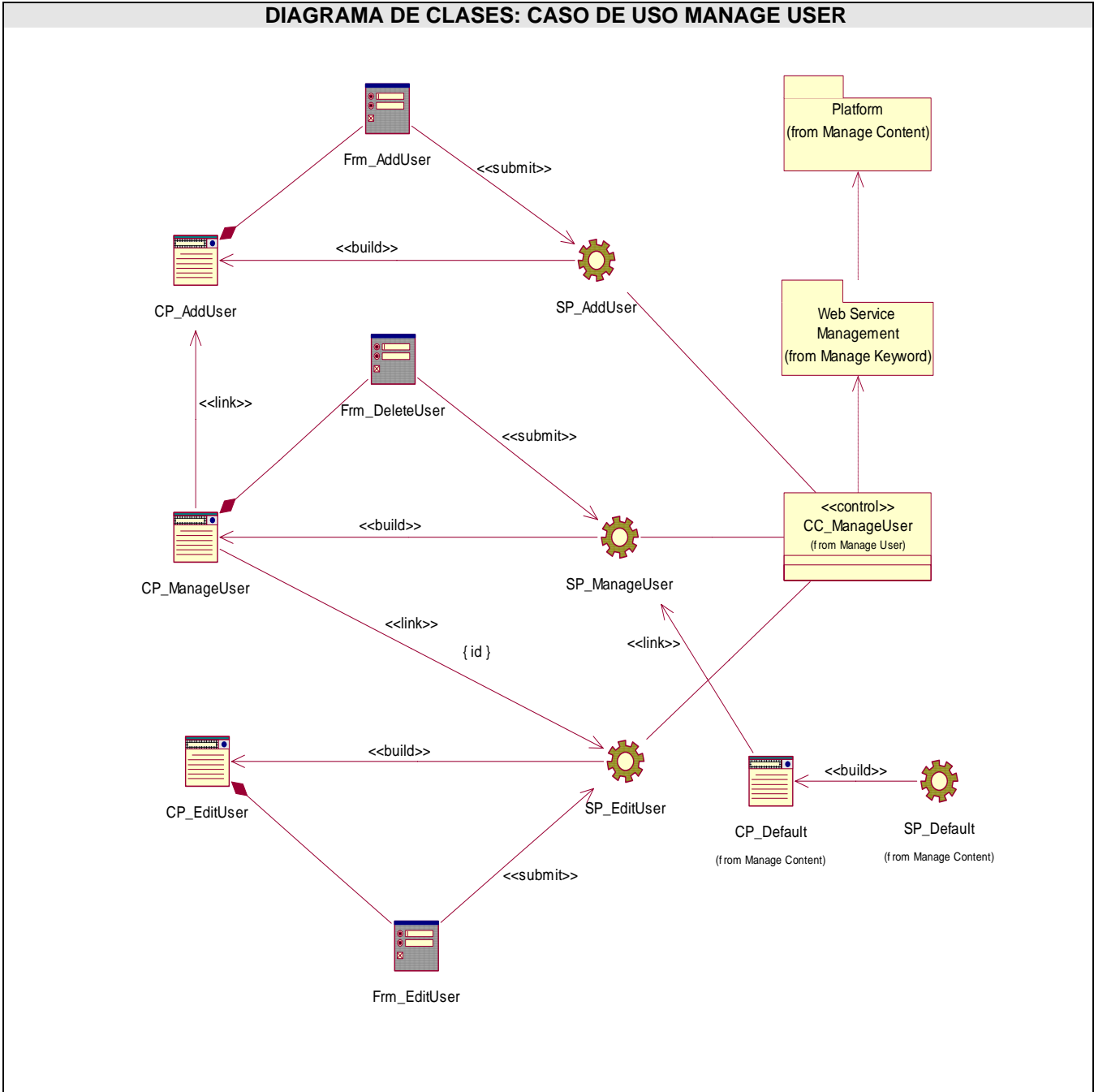


Figura 15. Diagrama de Clases del Diseño del Caso de Uso Manage User.

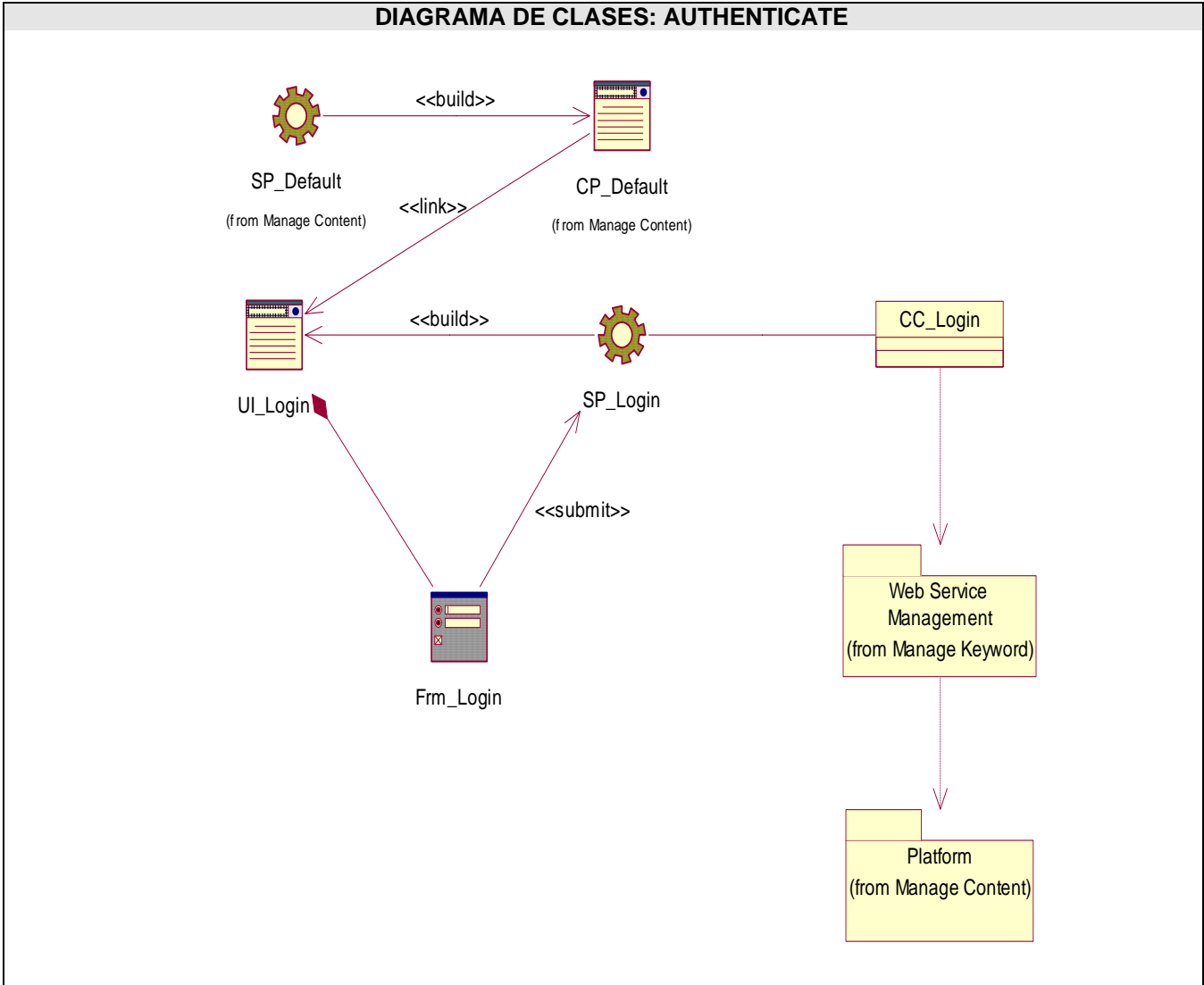


Figura 16. Diagrama de Clases del Diseño del Caso de Uso Authenticate.

DIAGRAMA DE CLASES: CASO DE USO MANAGE PROVIDER

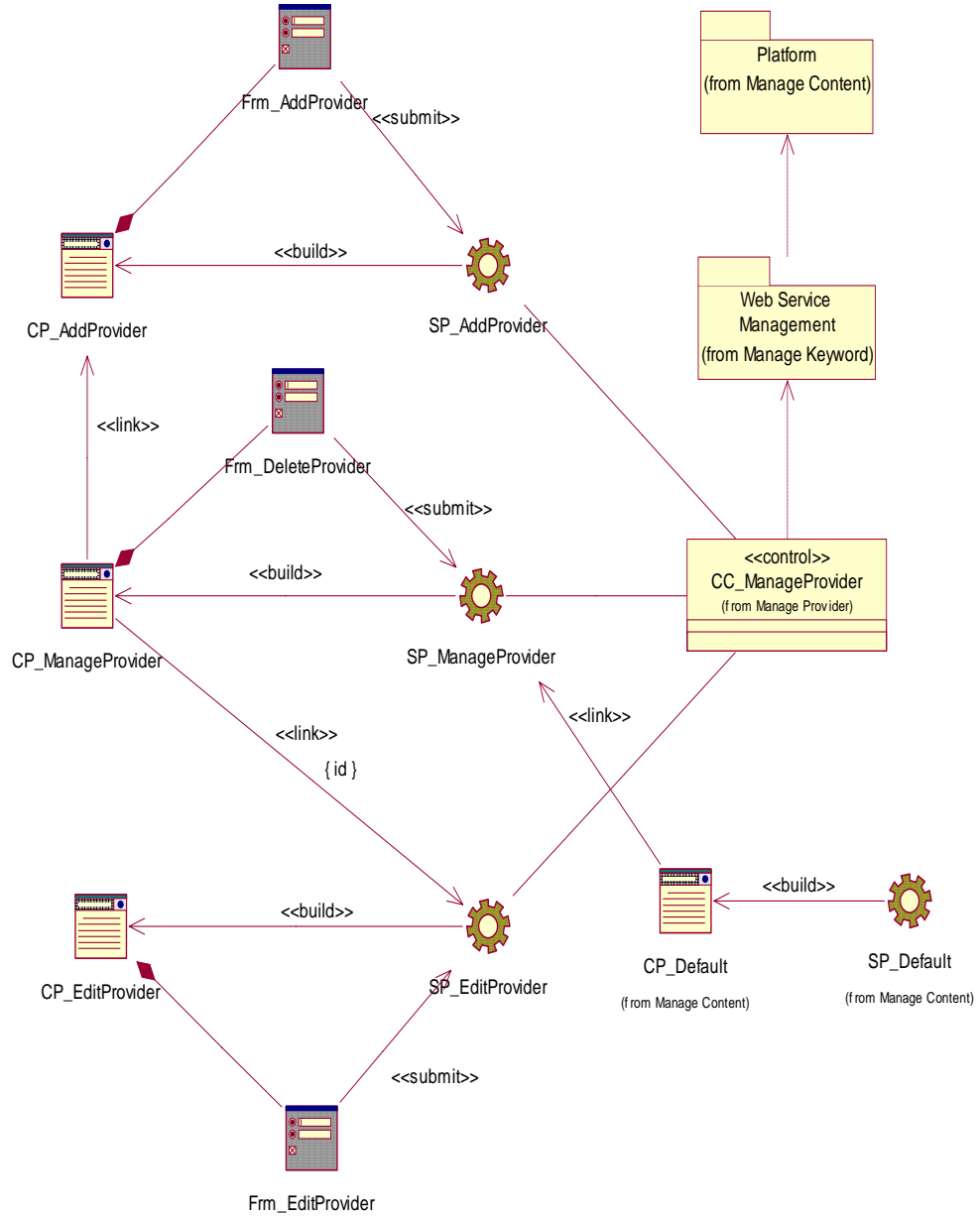


Figura 17. Diagrama de Clases del Diseño del Caso de Uso Manage Provider.

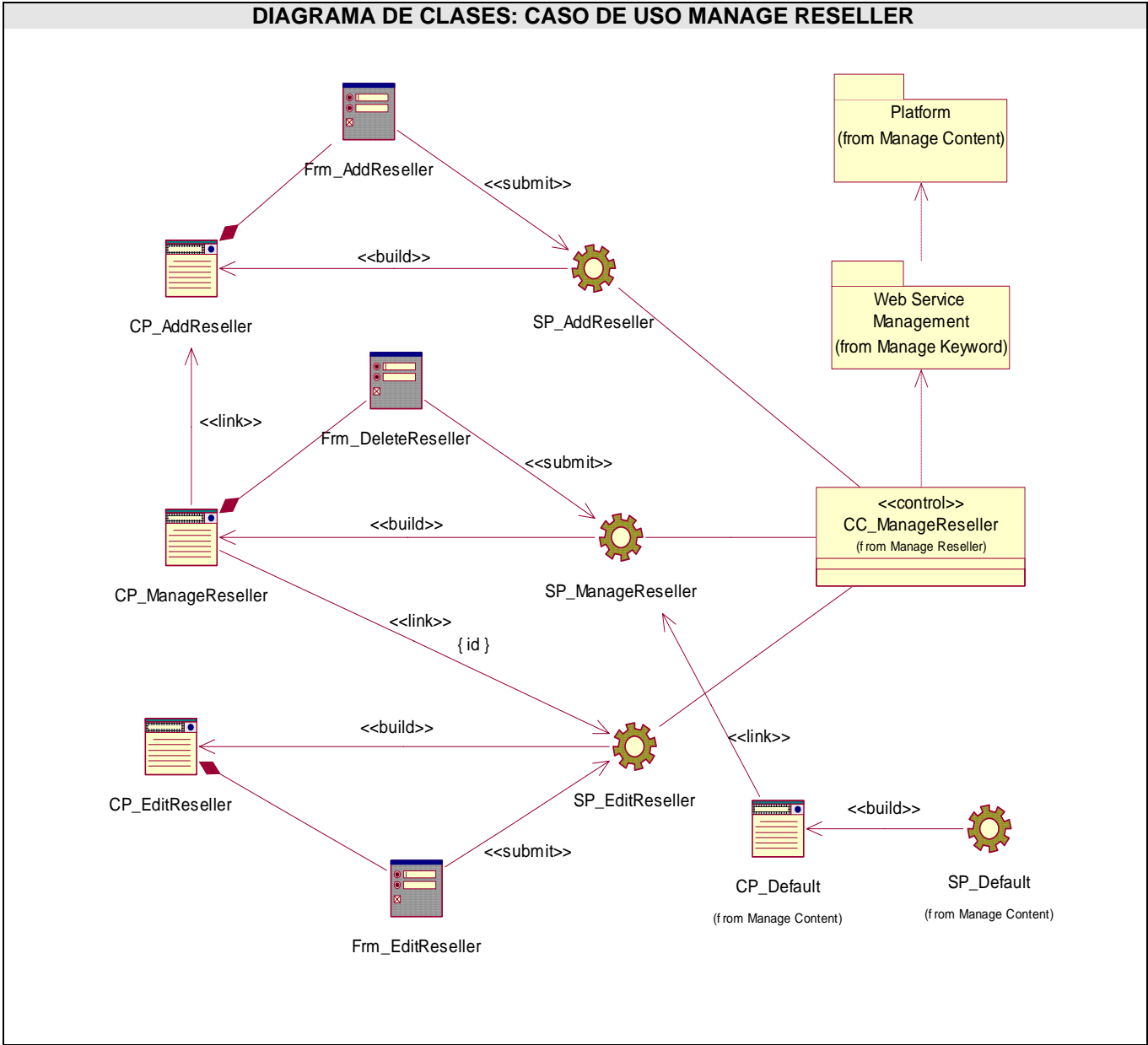


Figura 18. Diagrama de Clases del Diseño del Caso de Uso Manage Reseller.

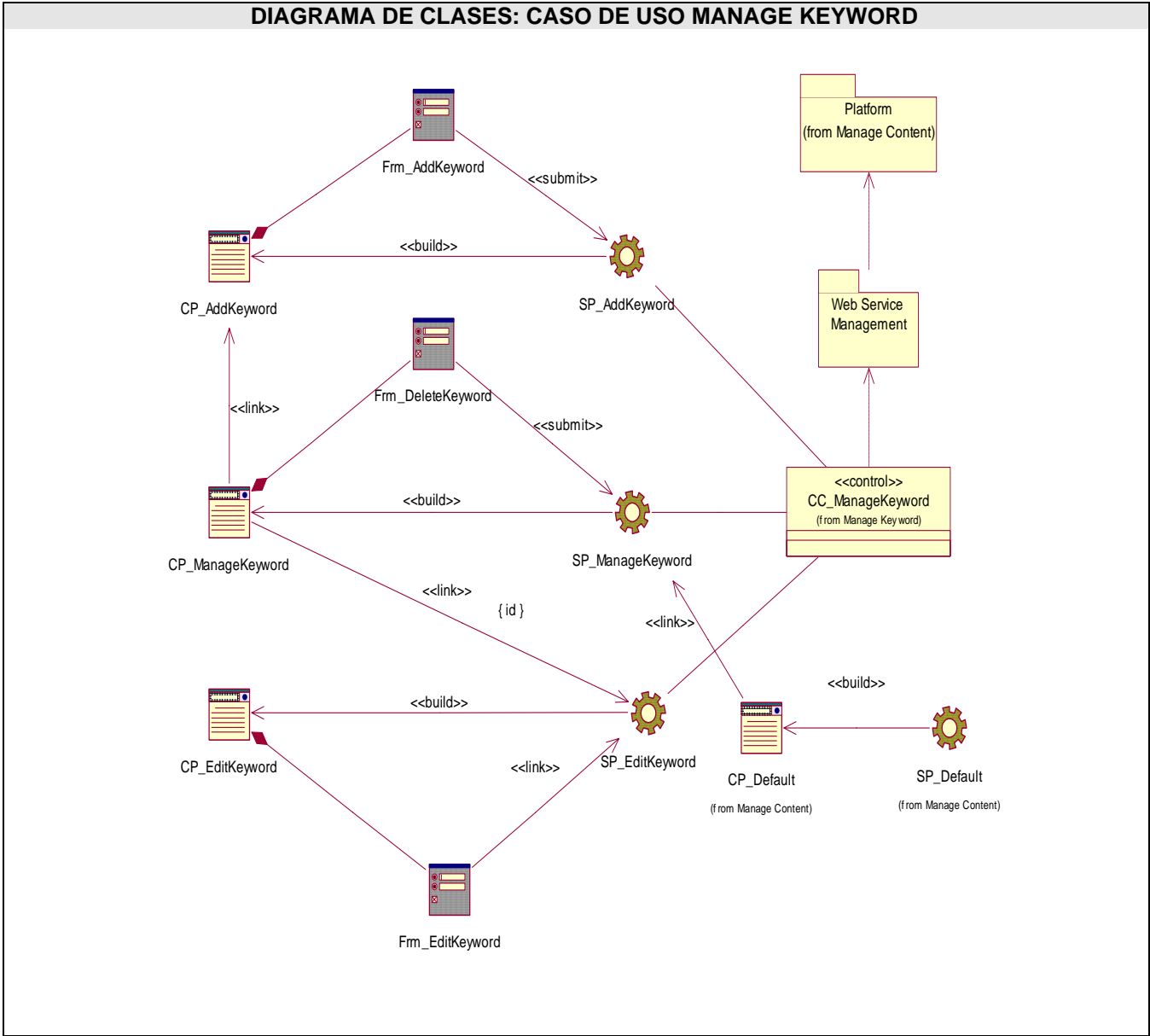


Figura 19. Diagrama de Clases del Diseño del Caso de Uso Manage Keyword.

DIAGRAMA DE CLASES: CASO DE USO MANAGE FORBIDDEN SHORTCODE

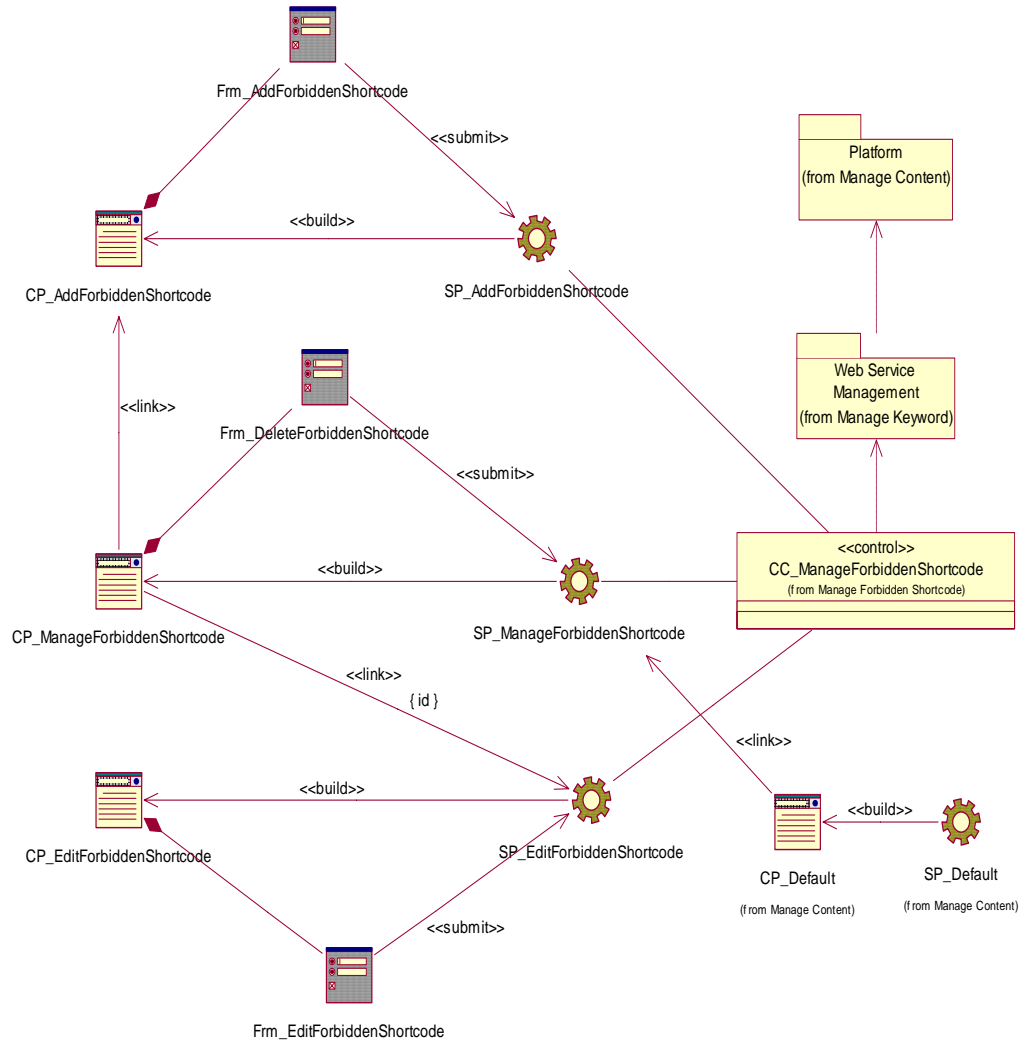


Figura 20. Diagrama de Clases del Diseño del Caso de Uso Manage Forbidden Shortcode.

DIAGRAMA DE CLASES: CASO DE USO MANAGE CONTENT CATEGORY

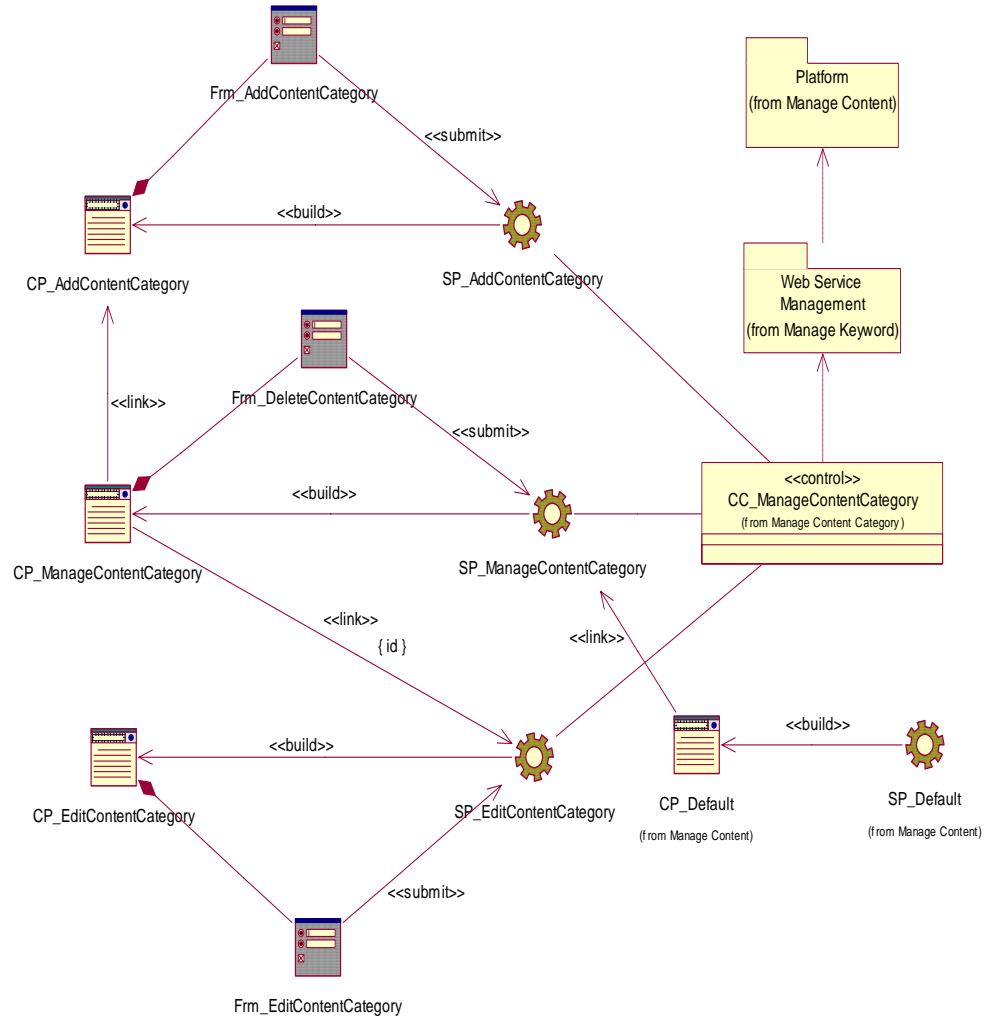


Figura 21. Diagrama de Clases del Diseño del Caso de Uso Manage Content Category.

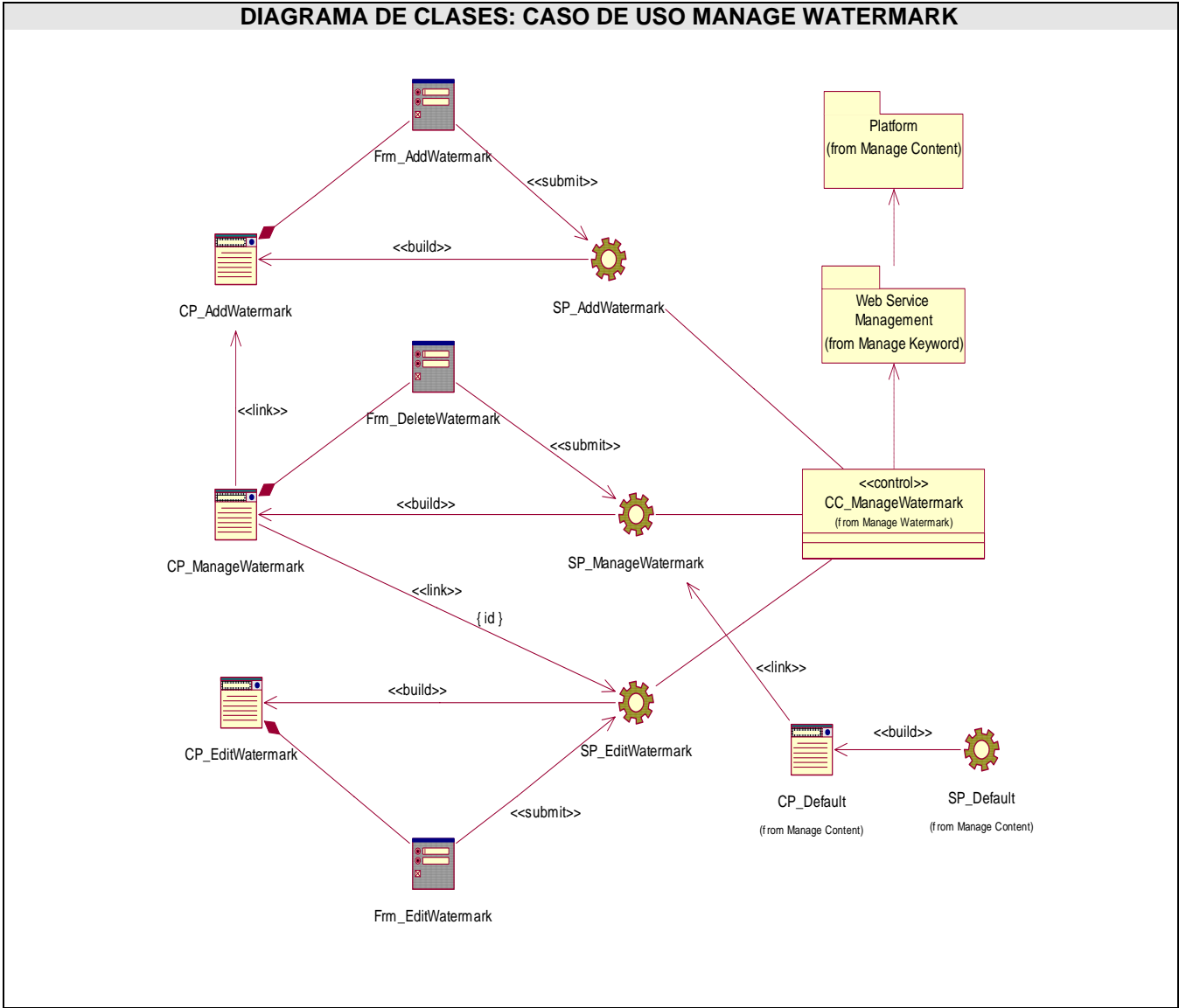


Figura 22. Diagrama de Clases del Diseño del Caso de Uso Manage Watermark.

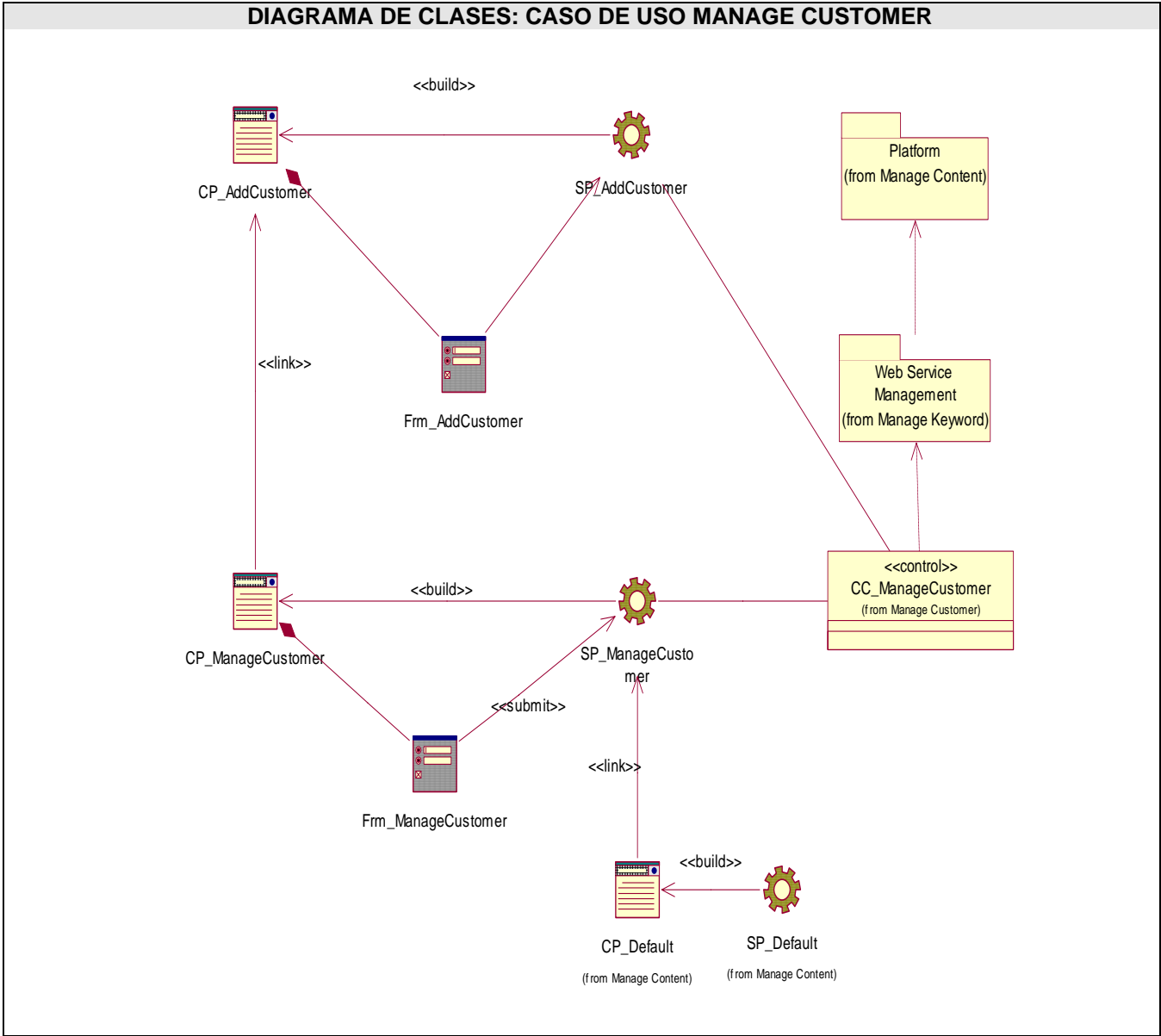


Figura 24. Diagrama de Clases del Diseño del Caso de Uso Manage Customer.

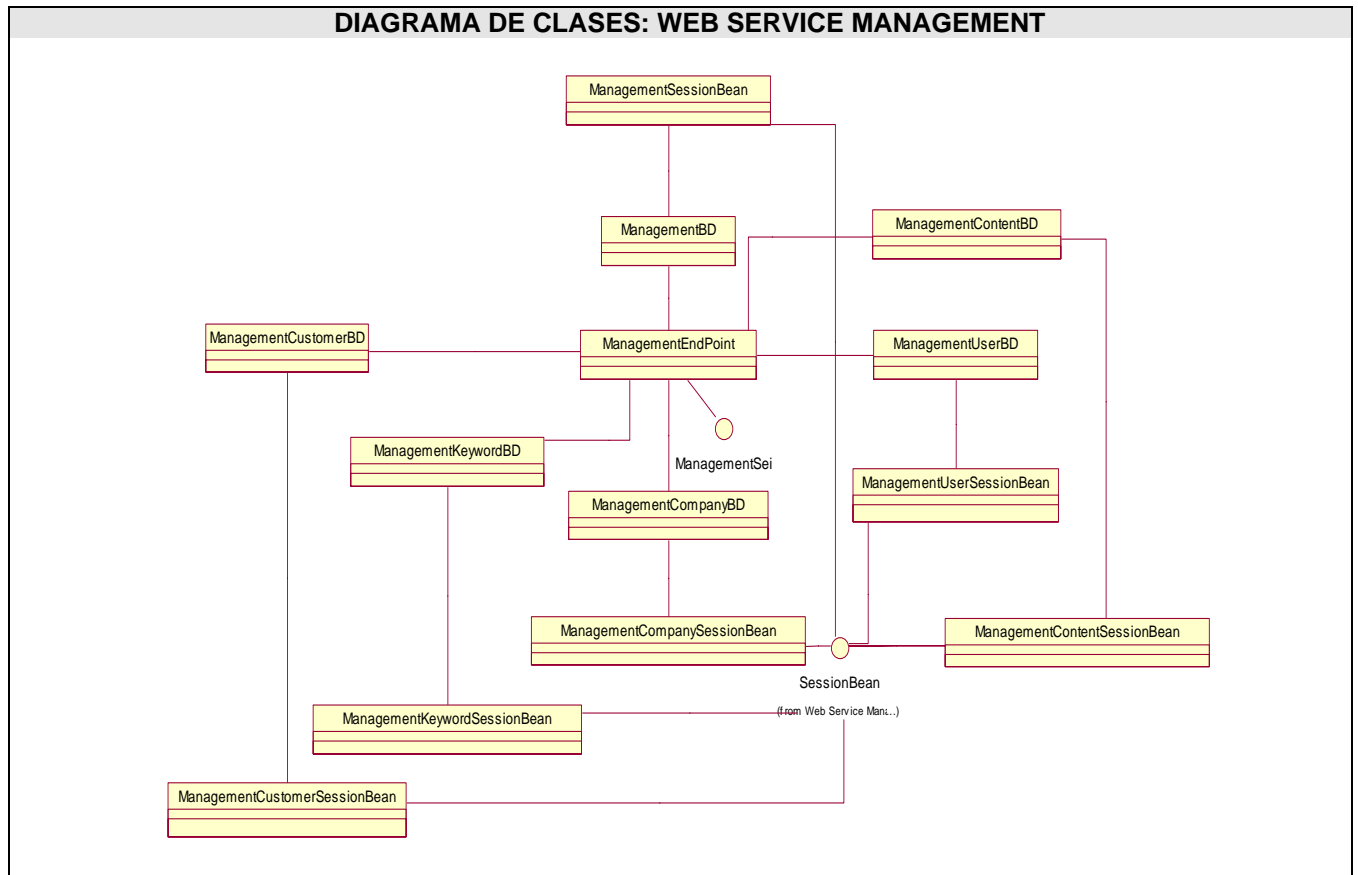


Figura 25. Diagrama de Clases del Diseño del Paquete Web Service Management.

Conclusiones

En este capítulo se expuso el análisis del sistema a través de los diagramas de clases del análisis. Además, se describieron las secuencias de intercambios de mensajes entre los objetos a través de los diagramas de secuencia. Por último, se ilustraron los diagramas de clases del diseño para cada caso de uso y para el paquete de clases del servicio web Management necesarios para la implementación.

CAPÍTULO 4: IMPLEMENTACIÓN

Introducción

En este capítulo se expone la distribución del sistema en nodos y los protocolos de comunicación entre cada uno de ellos. Se explica la trazabilidad de las clases del diseño a componentes de implementación mediante los diagramas de componentes del modelo de implementación.

4.1 Implementación.

El modelo de implementación representa la composición física de la implementación. En él está incluido el diagrama de despliegue que comenzó a desarrollarse en el flujo de trabajo de análisis y diseño y que se perfecciona en implementación. Contiene además, los diagramas de componentes que muestran un conjunto de elementos del modelo tales como componentes, paquetes de implementación y sus relaciones.

4.1.1 Diagrama de Despliegue.

Un diagrama de despliegue representa la distribución física del sistema distribuido. Es inicializado en el flujo de trabajo de análisis y diseño y perfeccionado en el de implementación.

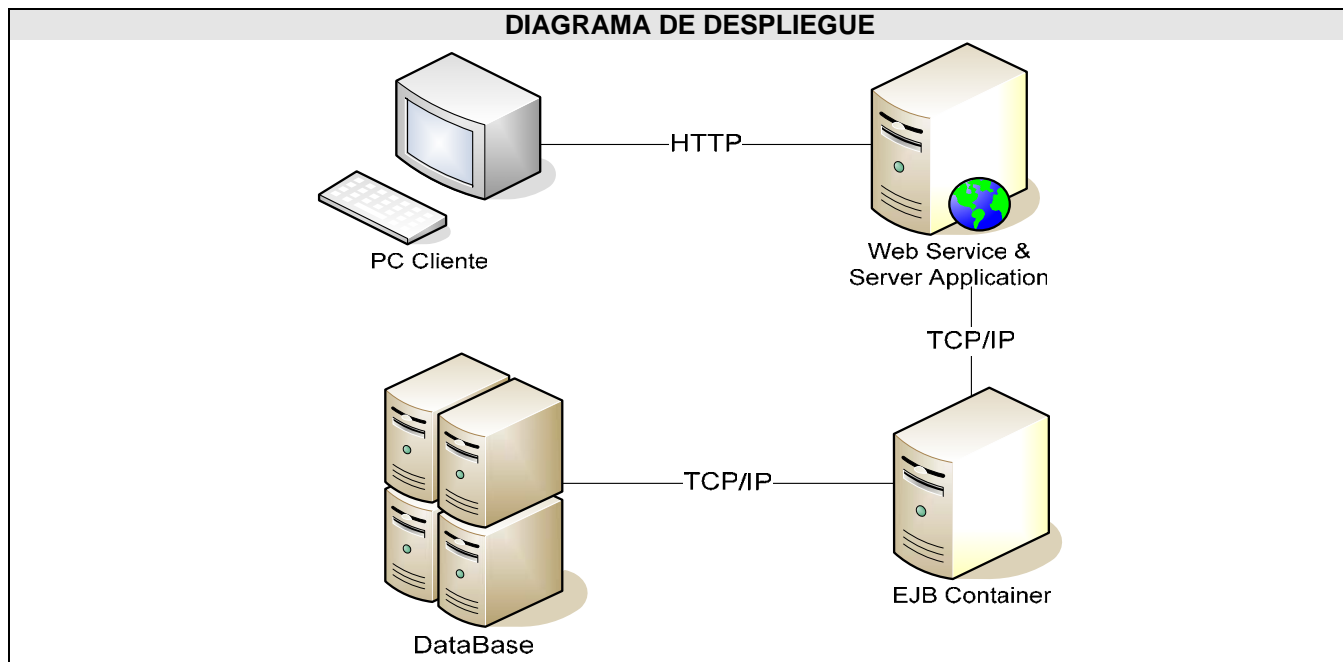


Figura 26. Diagrama de Despliegue.

La computadora cliente mediante un buscador se conecta a través del protocolo de conexión http al servidor de aplicación donde pueden encontrarse además los servicios web. Estos últimos se conectan por TCP/IP al contenedor de EJBs, quienes finalmente interactúan con el servidor o el cluster de la base de datos.

4.1.2 Diagramas de Componentes.

Los diagramas de componentes han sido organizados por paquetes siguiendo el criterio de organización por caso de uso. El diagrama de componentes es usado para estructurar el modelo de implementación y mostrar las relaciones entre los elementos de este.

En el siguiente diagrama se describe la distribución en capas del sistema y la dependencia entre los paquetes de implementación. La primera es la capa de presentación dividida en paquetes organizados por casos de uso, esta depende de la capa de la lógica del negocio que está representada por el paquete del servicio web Management y este a su vez de la plataforma que es la capa de acceso a datos.

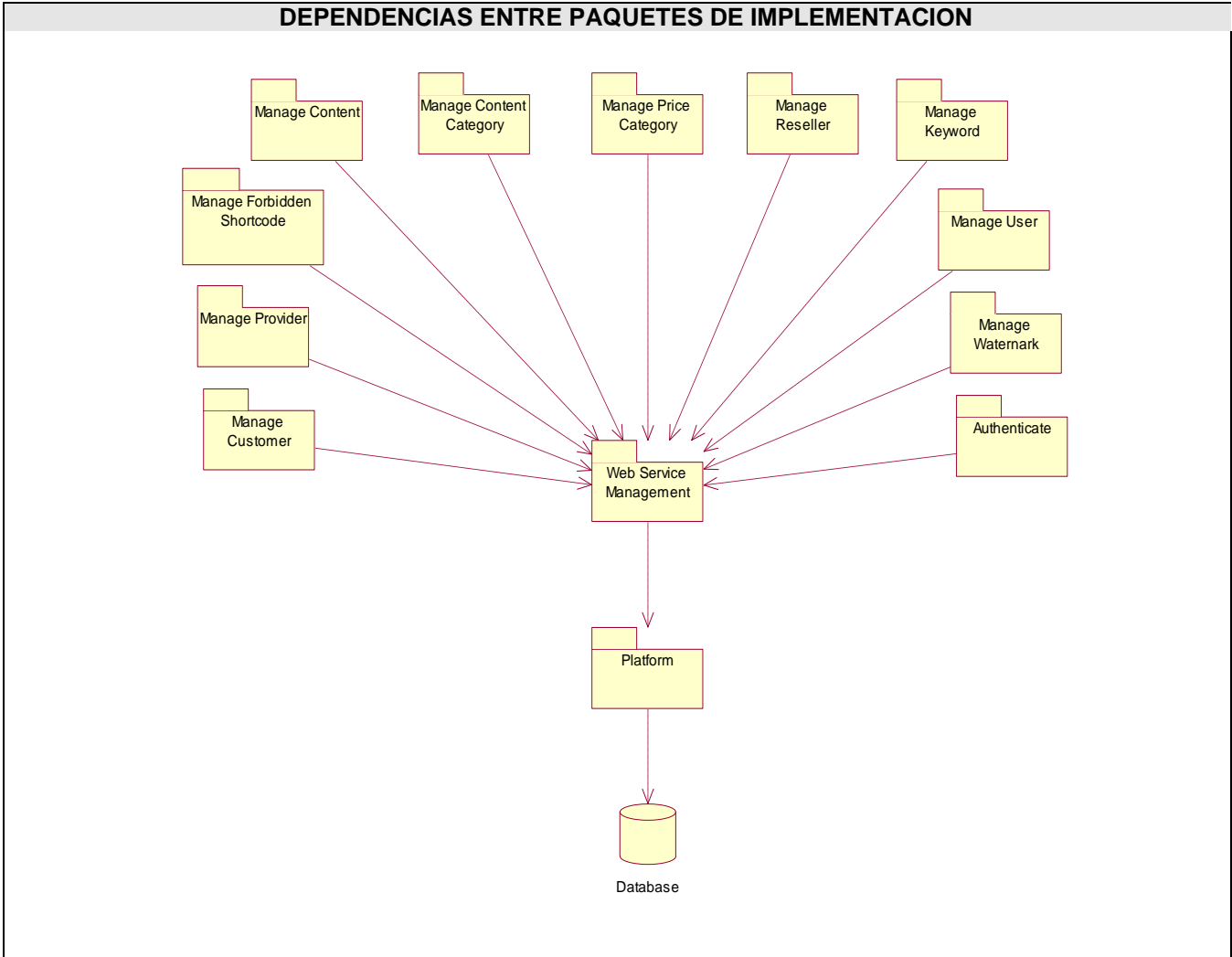


Figura 27. Diagrama de Paquetes del Sistema.

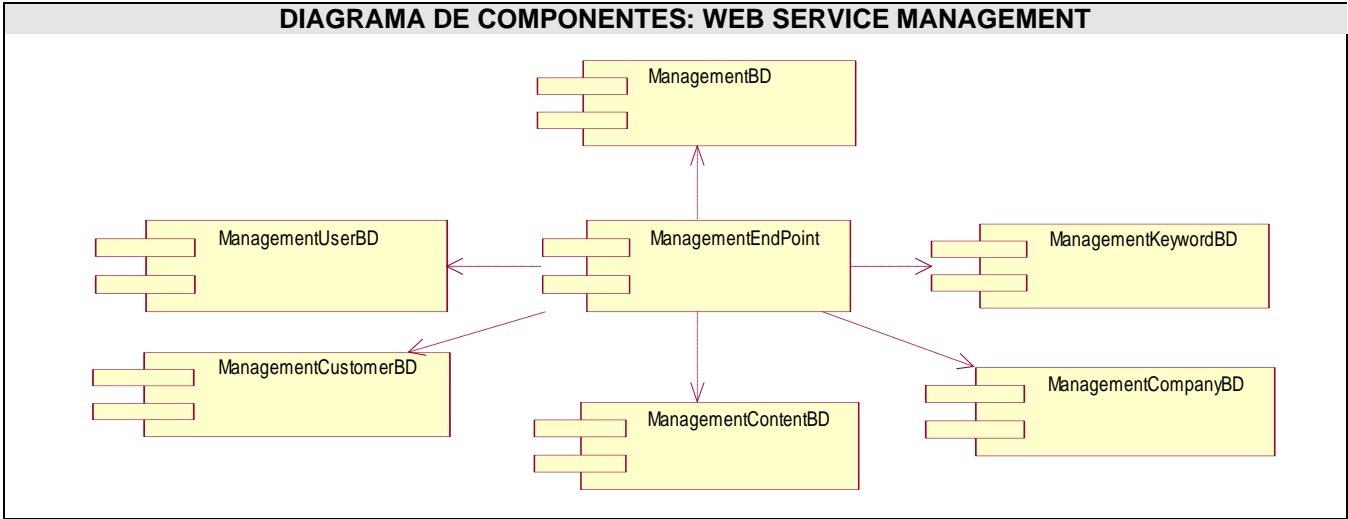


Figura 28. Diagrama de Componentes del Paquete Web Service Management.

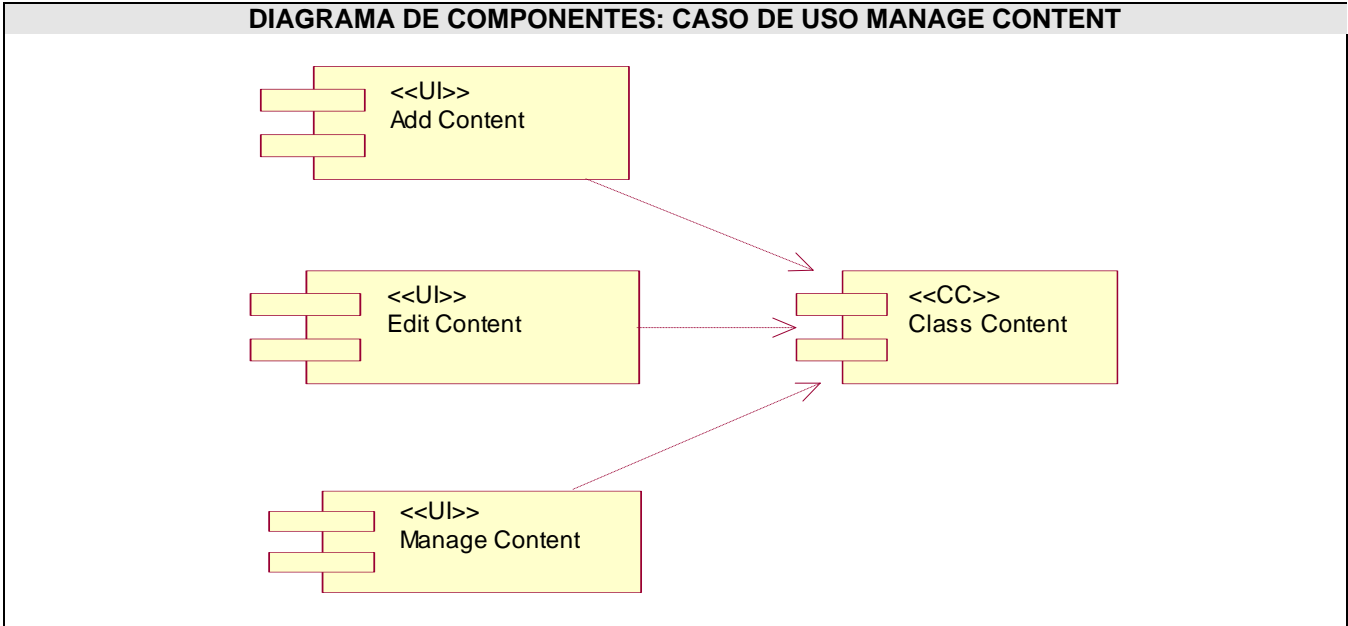


Figura 29. Diagrama de Componentes del Caso de Uso Manage Content.

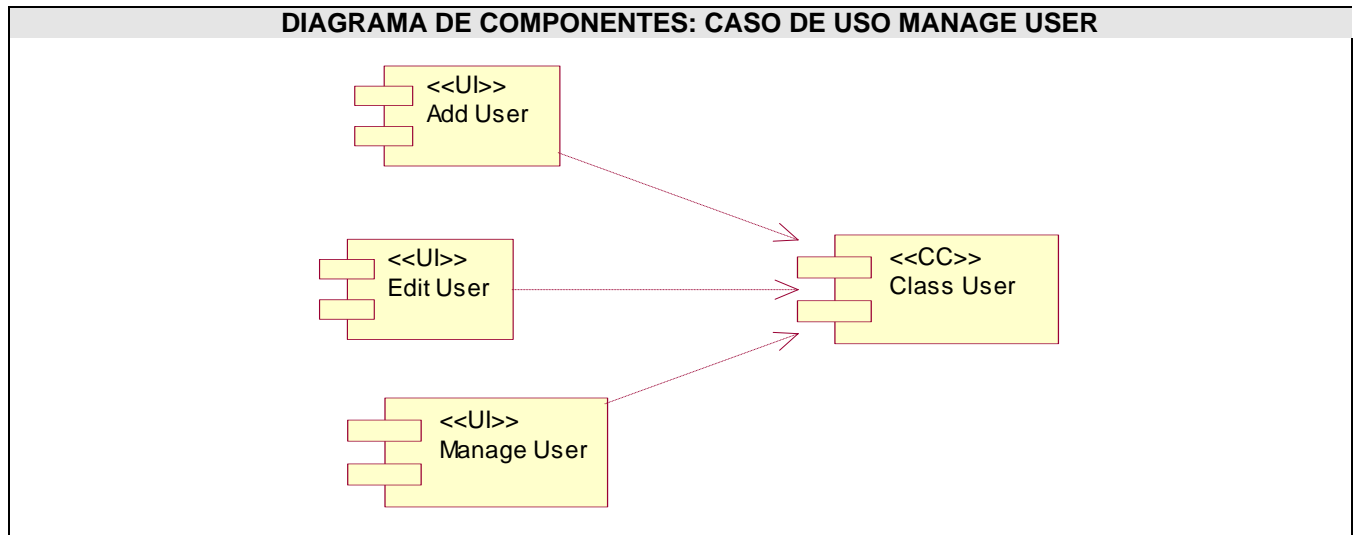


Figura 30. Diagrama de Componentes del Caso de Uso Manage User.



Figura 31. Diagrama de Componentes del Caso de Uso Authenticate.

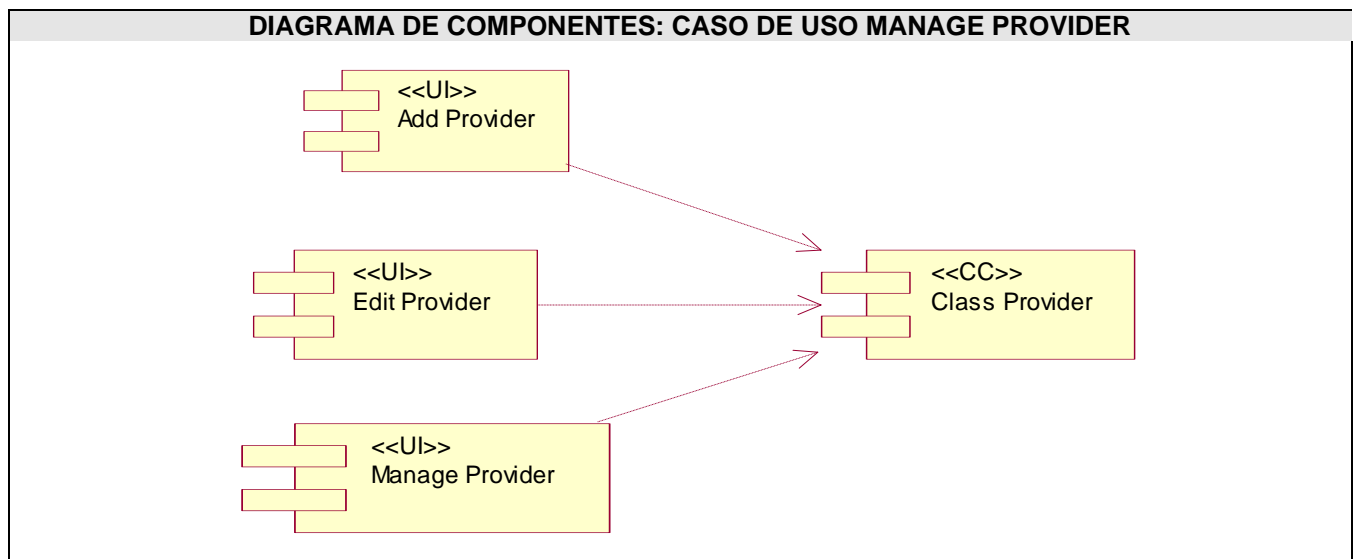


Figura 32. Diagrama de Componentes del Caso de Uso Manage Provider.

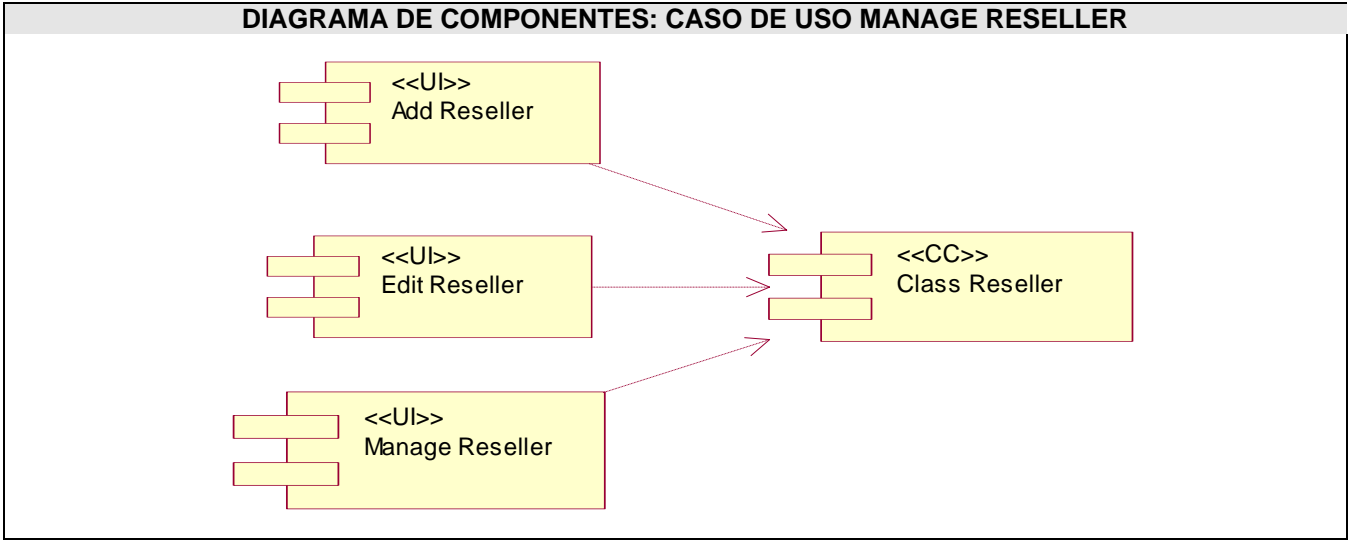


Figura 33. Diagrama de Componentes del Caso de Uso Manage Reseller.

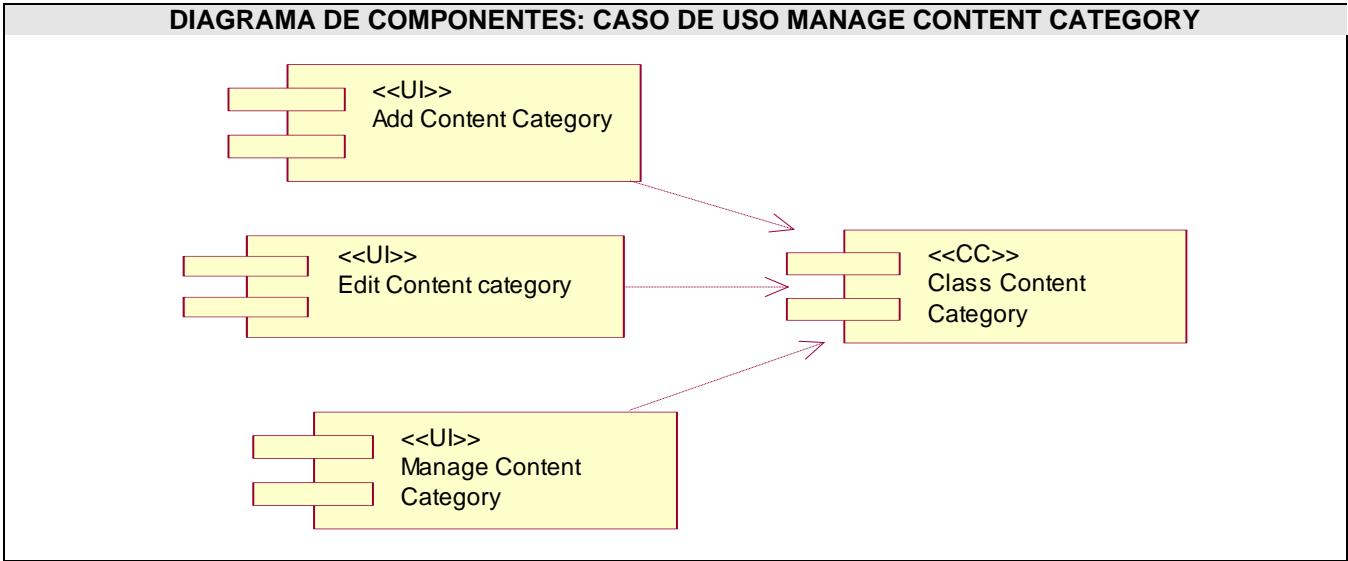


Figura 34. Diagrama de Componentes del Caso de Uso Manage Content Category.

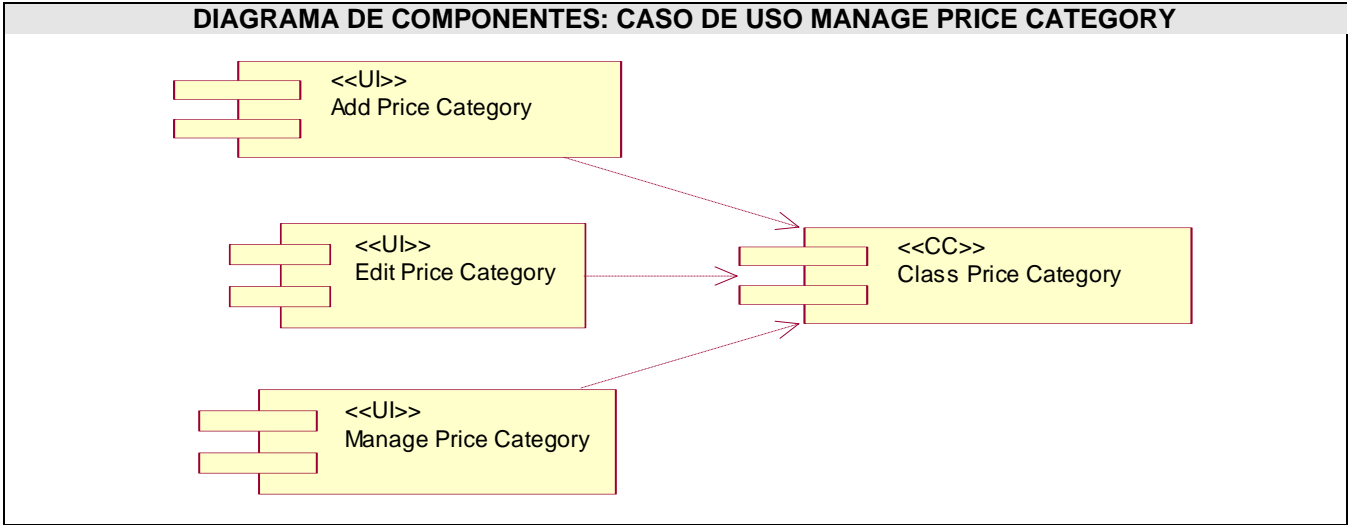


Figura 35. Diagrama de Componentes del Caso de Uso Manage Price Category.

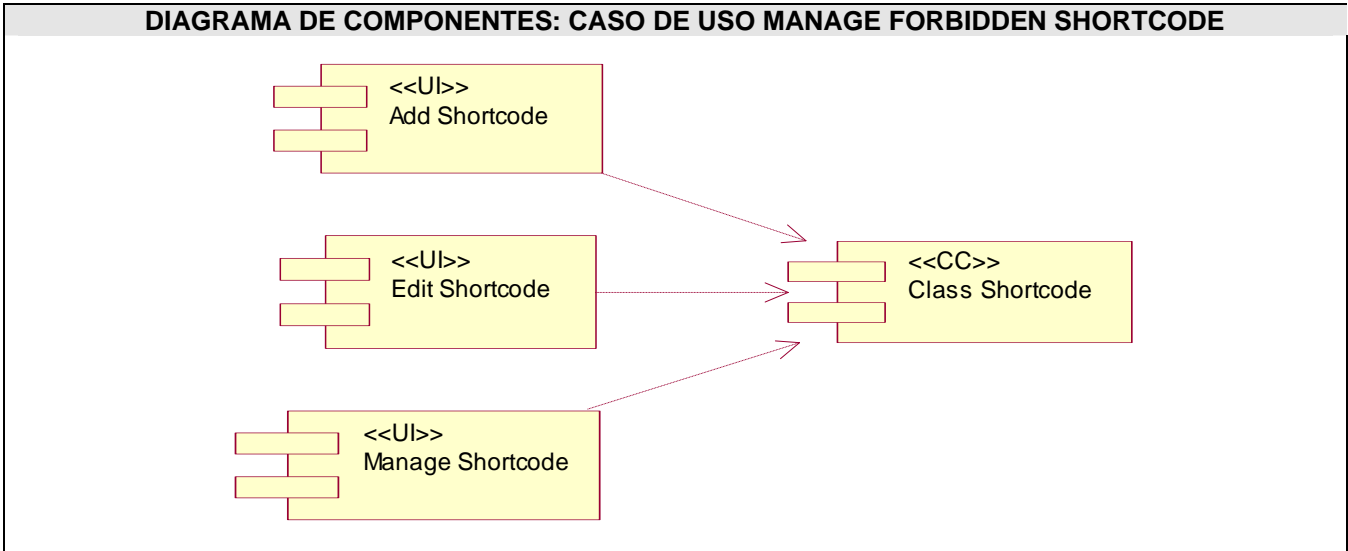


Figura 36. Diagrama de Componentes del Caso de Uso Manage Forbidden Shortcode.

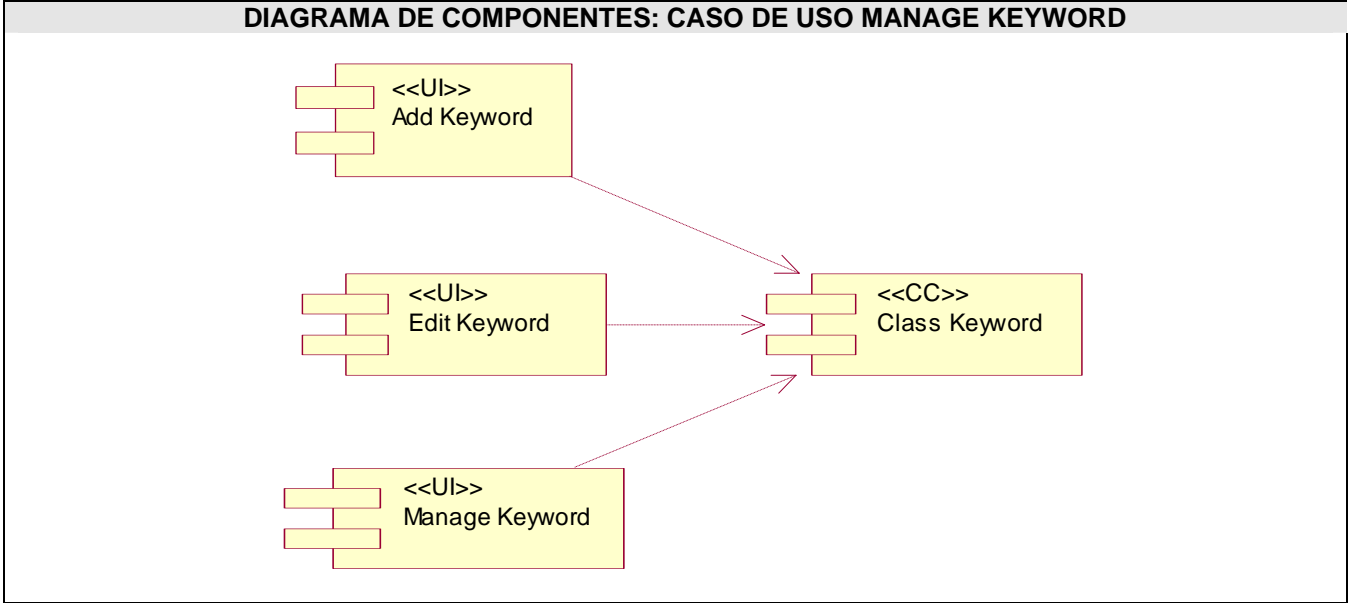


Figura 37. Diagrama de Componentes del Caso de Uso Manage Keyword.

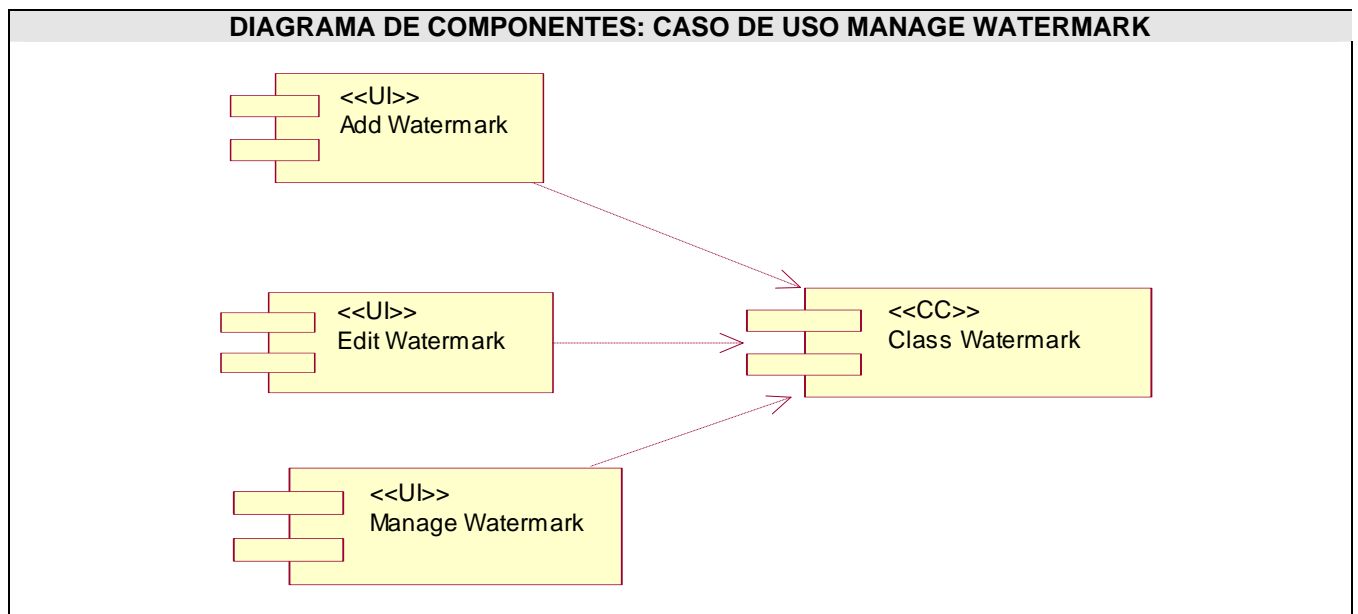


Figura 38. Diagrama de Componentes del Caso de Uso Manage Watermark.

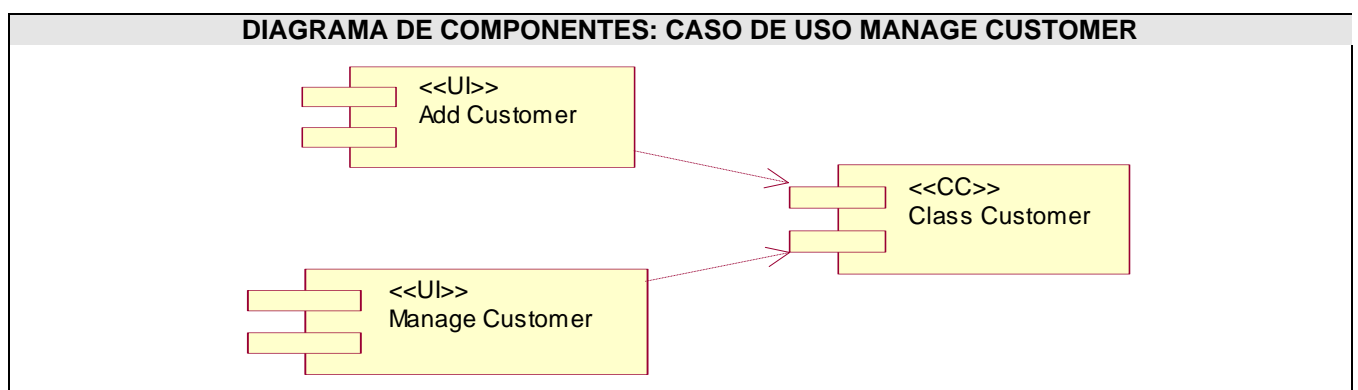


Figura 39. Diagrama de Componentes del Caso de Uso Manage Customer.

Conclusiones

En este capítulo se describió la distribución del sistema en nodos, especificados en el diagrama de despliegue y se muestran los diagramas de componentes organizados por casos de uso. Mediante el modelado de estos diagramas y del estudio llevado a cabo se definieron las capas del sistema y la

comunicación entre ellas permitiendo enfocar la programación a esta arquitectura para lograr que cada módulo se centre en su tarea, es decir, que el sistema provea a todas las capas y componentes de una alta cohesión y un bajo acoplamiento.

CAPÍTULO 5. ESTUDIO DE FACTIBILIDAD.

Introducción

El estudio de factibilidad es una etapa del desarrollo del software que no debe faltar, pues es donde se decide si será factible o no seguir con el desarrollo de este. Existen varios métodos para calcular la factibilidad de un proyecto: COCOMO, COCOMO II, Análisis de Punto de función, Análisis de Puntos de Casos de Uso, entre otros. En este trabajo se decidió usar el método de Puntos de Casos de Uso.

5.1 Estimación por Puntos de Casos de Uso

Existen varios métodos para el cálculo y la estimación, factibles de ser utilizados en proyectos que utilicen orientación a objetos, pero como el método de Puntos de Caso de Uso se definió como extensión del Punto de Función, es un estándar del proceso, de acuerdo a las siguientes consideraciones:

- La alternativa es simple y similar a los Puntos de Función.
- Las otras alternativas no demostraron tener aceptación general ni se han utilizado ampliamente.
- Algunas alternativas son propietarias.

El Método de Puntos de Casos de Uso cumple con al menos tres objetivos fundamentales:

- Es un método similar a lo definido en la Administración de Requerimientos respecto de Puntos de Función para proyectos desarrollados con Análisis Estructurado.
- Permite disponer de una métrica adimensional que con el tiempo podrá ser comparada con las métricas actualmente en uso y se enriquecerán con la historia de los proyectos que se desarrollen bajo la metodología de orientación a objetos.
- Existencia de ejemplos de uso del método de distintas fuentes.

El método de Puntos de Casos de Uso es un método de estimación y cálculo de tamaño del software basado en cuentas hechas sobre los casos de uso para un sistema de software.

Exige la existencia de un modelo de casos de uso, por lo que la labor deberá ser hecha cuando exista algún entendimiento del dominio del problema o cuando se estén realizando las labores de arquitectura y dimensionamiento del tamaño del sistema. Por lo general, estas condiciones están dadas al término de las actividades de Análisis.

En términos simples, el método requiere de casos de uso en modo textual y gráfico sólo en términos de mayor claridad, se revisan en detalle los casos de uso seleccionados en la etapa del proyecto que se defina y se realizan los siguientes pasos: [13]

5.1.1 Cálculo de Puntos de Casos de Uso sin ajustar

$$UUCP = UAW + UUCW$$

$$UUCP = 3 + 105$$

$$UUCP = 108$$

Donde,

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

5.1.1.1 Factor de Peso de los Actores sin ajustar (UAW).

Tabla 13. Factor de Peso de los Actores.

Tipo de actor	Descripción	Factor de peso	Actores	Total
Simple	Sistema con sistema a través de interfaz de programación.	1	0	0
Medio	Sistema con sistema mediante protocolo de interfaz basada en texto.	2	0	0
Complejo	Persona que interactúa con el sistema mediante interfaz gráfica.	3	1	3

$$UAW = S(\text{Factor} * \text{Actores})$$

$$UAW = 3$$

5.1.1.2 Factor de Peso de los Casos de Uso sin ajustar (UUCW).

Tabla 14. Factor de Peso de los Casos de Uso.

Tipo de CU	Descripción	Peso	Cantidad de CU	Total
Simple	El caso de uso tiene de 1 a 3 transacciones.	5	1	5
Medio	El caso de uso tiene de 4 a 7 transacciones.	10	10	100
Complejo	El caso de uso tiene más de 8 transacciones.	15	0	0
UUCW = Sumatoria(Factor * CantCU)			UUCW	105

5.1.2 Cálculo de Puntos de Casos de Uso ajustados.

$$UCP = UUCP \times TCF \times EF$$

$$UCP = 108 \times 1.18 \times 0.665$$

$$UCP = 84.7476$$

Donde,

UCP: Puntos de Casos de Uso ajustados.

UUCP: Puntos de Casos de Uso sin ajustar.

TCF: Factor de complejidad técnica.

EF: Factor de ambiente.

5.1.2.1 Factor de Complejidad Técnica (TCF).

Tabla 15. Factor de Complejidad Técnica.

Factor	Descripción	Peso	Valor asignado	Total
T1	Sistema distribuido	2	5	10
T2	Tiempo de respuesta	1	5	5
T3	Eficiencia del usuario final	1	3	3
T4	Funcionamiento Interno complejo	1	3	3
T5	El código debe ser reutilizable	1	5	5
T6	Facilidad de instalación	0.5	5	2.5

T7	Facilidad de uso	0.5	3	1.5
T8	Portabilidad	2	5	10
T9	Facilidad de cambio	1	5	5
T10	Concurrencia	1	3	3
T11	Incluye objetivos especiales de seguridad	1	4	4
T12	Provee acceso directo a terceras partes	1	5	5
T13	Se requieren facilidades especiales de entrenamiento de usuarios	1	1	1
			Sumatoria	58
TCF = 0.6 + 0.01 * Sumatoria(Peso * Valor)			TCF	1.18

5.1.2.2 Factor de ambiente (EF).

Tabla 16. Factor de Ambiente.

Factor	Descripción	Peso	Valor asignado	Total
E1	Familiaridad con el modelo de proyecto utilizado	1.5	3	4.5
E2	Experiencia en la aplicación	0.5	3	1.5
E3	Experiencia en la orientación a objetos.	1	5	5
E4	Capacidad del analista líder.	0.5	3	1.5
E5	Motivación.	1	5	5
E6	Estabilidad de requerimientos	2	5	10
E7	Personal Part-Time	-1	0	0
E8	Dificultad del lenguaje de programación	-1	3	-3
			Sumatoria	24.5
EF = 1.4 - 0.03 * Sumatoria(Peso * Valor)			EF	0.665

5.1.3 De los Puntos de Casos de Uso a la estimación del esfuerzo.

El esfuerzo en horas-hombre viene dado por:

$$E = UCP \times CF$$

$$E = 84.7476 \times 20$$

$$E = 1694.952$$

Donde,

E: Esfuerzo estimado en horas-hombre.

UCP: Puntos de Casos de Uso ajustados.

CF: Factor de conversión.

Para estimar el factor de conversión:

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 y cuántos están por encima del valor medio (3), para los factores E7 y E8.

- Si el total es 2 o 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.
- 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 este caso el total de la suma de factores que afectan al Factor de ambiente que están por debajo del valor medio (3) para los factores E1 a E6 y los que están por encima del valor medio (3), para los factores E7 y E8, es menor que 2.

Por tanto el CF = 20

UCP = UUCP * TCF * EF = 84.7476

Tabla 18. Distribución de esfuerzo entre las actividades.

Actividad	Porcentaje %	Horas-Hombres
Análisis	20	847.476
Diseño	20	847.476
Implementación	40	1694.952
Pruebas	10	423.738
Sobrecarga (otras actividades)	10	423.738
Total	100	4237.38

Esfuerzo Total (Horas-Hombre)

ET1 = 4237.38.

Esta cifra indica que una persona puede desarrollar el proyecto en 4237.38 horas.

Esfuerzo Total (Mes-Hombre)

$ET2 = 22.0696875$.

Esta cifra indica que una persona puede desarrollar el proyecto en 22 meses y medio aproximadamente.

Para dos personas este número se reduciría a 11 meses de trabajo.

Salario Promedio

$SM = 100$

El salario promedio mensual es \$100 por persona.

Cantidad de Hombres

$CH = 2$ hombres.

Costo Hombre-Mes

$CHM = 200$ \$/mes.

Costo Total

$Costo = CHM * ET / CH$

$Costo = \$ 4413.9375$

Tiempo

$Tiempo = ET / CH$

$Tiempo = 22.0696875 / 2$

$Tiempo = 11.0348438$

Los cálculos realizados arrojan como resultado que con 2 hombres en 11 meses aproximadamente y un monto de \$ 4413.9375 se puede desarrollar el sistema de administración de contenido para móviles satisfactoriamente.

5.2 Beneficios tangibles e intangibles.

El Sistema de Administración de Contenidos para teléfonos celulares es un producto con fines comerciales. Su principal objetivo es gestionar los contenidos para estos dispositivos. Posee funcionalidades adicionales para gestionar proveedores, revendedores, usuarios por roles. Este tipo de aplicaciones es muy cotizada en el mercado internacional. El sistema está actualmente en proceso de contratos de venta por lo que no se tienen beneficios tangibles inmediatos. Los beneficios obtenidos hasta este momento son intangibles:

- Un sistema de administración de contenidos en Procyon.
- La adquisición de conocimientos por los desarrolladores y demás miembros del equipo en cuanto a nuevas tecnologías, tendencias del mercado, etc.
- Un alto nivel de interoperabilidad mediante el uso de servicios web.
- La gestión de funcionalidades agregadas como las categorías de contenido, de precio, palabras claves y marcas de agua.

5.3 Análisis de costo

La factibilidad del sistema depende del nivel de ventas que tenga. Si la aplicación es comprada, existen varias opciones de pago, una de ellas es que puede ser vendido el sistema completo, si el comprador necesita todas las funcionalidades y el sistema satisface sus necesidades, entonces paga la aplicación y puede además pagar el soporte, si cree que lo necesita.

Otra vía de pago sería aplicando revenue sharing, convenio entre compañías que divide los porcentajes de ganancias de cada contenido que se comercialice a través de la aplicación, por ejemplo, la venta de un tono en un dólar, proveería \$0.75 al operador y \$0.25 a la empresa, aumentando las ganancias a medida que aumenta la venta de contenidos.

Como última forma de pago estaría mediar entre las compañías que necesiten acceder a los servicios, o sea, la empresa Procyon sería la encargada de administrar el sistema y ponerlo a disposición de las compañías que necesiten usarlo cobrando así los servicios.

Concluyendo, el sistema es factible, pues está en contrato de venta y usando cualquier vía de pago se obtendrían ganancias económicas que podrían clasificarse como beneficios a largo plazo.

Conclusiones

La factibilidad de un proyecto es un paso muy importante en su desarrollo pues a partir de este punto se decide si se continuará el proyecto o no. En este capítulo se realizó el estudio de factibilidad mediante el método de Punto de Casos de Uso, el cual permite calcular el tamaño del software, la estimación del esfuerzo y costo del proyecto, concluyendo que el sistema es factible.

CONCLUSIONES

En este trabajo se realizaron varios estudios para elegir las tecnologías, lenguajes de programación y metodología de desarrollo a utilizar y se analizaron las tendencias actuales en el desarrollo de sistemas de administración de contenidos. Finalmente se escogieron las que resultaron más adecuadas para el desarrollo del sistema y estaban acordes a los intereses del cliente. Se desarrolló el servicio web sobre la plataforma Java y la aplicación web usando la plataforma .NET, específicamente la tecnología ASP.NET y lenguaje C#.

A través de la metodología RUP se realizó el análisis del sistema, mediante el cual se concibió una primera versión de la arquitectura del sistema, que sirvió de base al posterior diseño de la aplicación, donde se perfeccionó esta arquitectura, enfocada ya a los lenguajes de programación y tecnologías específicas a usar.

Durante el flujo de trabajo de implementación se realizaron los diagramas de componentes y se perfeccionó el diagrama de despliegue. Este modelado facilitó la comprensión de la comunicación entre las capas y la distribución física de estas para la implementación del sistema.

Como resultado de este trabajo se desarrolló un Sistema de Administración de Contenidos que cumple con los requerimientos establecidos inicialmente. Por todo lo anterior se concluye que los objetivos propuestos fueron cumplidos exitosamente.

RECOMENDACIONES

- Se recomienda la implementación de las restantes aplicaciones web que sean clientes de los servicios web Reseller y Provider.
- Se recomienda la implementación de clientes para los servicios web en lenguajes de código abierto con el objetivo de disminuir los costos en términos de licencias.
- Se recomienda someter el sistema a pruebas de calidad.

BIBLIOGRAFÍA

- Walther Stephen. ASP.NET 2.0 Unleashed. Sams 2006. 1992 páginas.
- Spaanjaars Imar, Wilton Paul, Livermore Shawn. ASP.NET 2.0 Instant Results. Wrox 2006. 480 páginas.
- Evjen Bill, Hanselman Scott, Muhammad Farhan, Sivakumar Srinivasa, Rader Devin. Professional ASP.NET 2.0. Wrox 2005. 1253 páginas.
- Graham Steve, Davis Doug, Simeonov Simeon , Daniels Glen, Brittenham Peter, Nakamura Yuichi, Fremantle Paul, Koenig Dieter, Zentner Claudia. Building Web Services with Java: Making Sense of XML, SOAP, WSDL, and UDDI. Sams 2004. 816 páginas.
- Bequet Henry, Kunnumpurath Meeraj, Rhody Sean, Tost Andre. Beginning Java Web Services. Wrox Press 2002. 411 páginas.
- Chappell David A, Jewell Tyler. Java Web Services. O'Reilly Media 2002. 276 páginas.
- Suddeth Jeffery. Programming C# with Visual Studio .NET 2005. 2006. 492 páginas.
- InformaticaMillenium.com.mx. Sistemas para administración de contenido de páginas Web [en línea]. 2007. Disponible en World Wide Web:
<http://www.webtaller.com/maletin/articulos/sistemas_para_administracion_de_contenido_de_paginas_web.php >
- Sistemas para administración de contenido de páginas Web [en línea]. México, Milenium Network® The Internet Marketing Newsletter. Disponible en World Wide Web:
< <http://www.informaticamilenium.com.mx/paginas/mn/articulo98.htm> >

- Vignette pone en el mercado la solución web de nueva generación [en línea]. Madrid. 9 de octubre de 2006. Disponible en World Wide Web:
<<http://www.vignette.com/portal/site/es/menuitem.62215d74e262b2ba32189210180141a0/?vgnnextoid=3012698c0aa2e010VgnVCM1000005610140aRCRD&vgnnext-selectedmenuitem=e4f2973963125a601eb11f10180141a0&gblvcmpguid=3012698c0aa2e010VgnVCM1000005610140aRCRD>>
- Soluciones innovadoras y alta especialización [en línea].2005. Disponible en World Wide Web:
< <http://www.vass.es/actividad/ebusiness.asp?idsol=21> >
- Zamitiz Román, Carlos Alberto. Temas Especiales de Computación [en línea]. UNAM, Facultad de Ingeniería. Disponible en World Wide Web:
- <<http://www.fi-b.unam.mx/pp/profesores/carlos/java/indice.html> >
- Méndez, Justo. Lenguajes de Programación [en línea]. 1997. Disponible en World Wide Web:
<<http://www.monografias.com/trabajos/lengprog/lengprog.shtml> >
- Ruiz Diego. El lenguaje C# y la plataforma .NET [en línea]. Capítulo Gratis. Disponible en World Wide Web: <<http://www.tectimes.com/onweb/seccion.aspx?s=2&numero=lpcu076> >
- Rodríguez Barreira, Noelia. Introducción a servicios web [en línea]. Septiembre, 2002. Disponible en World Wide Web:
<<http://www.elrincondelprogramador.com/default.asp?pag=articulos/leer.asp&id=32>>
- Letelier Patricio, Penadés M^a Carmen. Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP) [en línea]. Universidad Politécnica de Valencia. Disponible en World Wide Web:
< <http://www.willydev.net/descargas/masyxp.pdf> >
- Ayuda de la herramienta Case Rational Rose.

- Booch Grady, Jacobson Ivar, Rumbaugh James. El Proceso Unificado de Desarrollo de Software. Addison Wesley 2000.
- Larman Craig. UML y Patrones. Prentice Hall Hispanoamericana 1999.
- Badani, Sigifredo E. Hernández. Métricas de estimación de tamaño Puntos de Caso de Uso. Septiembre 2002.

REFERENCIAS BIBLIOGRÁFICAS

[1] InformaticaMillenium.com.mx. Sistemas para administración de contenido de páginas Web [en línea]. 2007. Disponible en World Wide Web:

<http://www.webtaller.com/maletin/articulos/sistemas_para_administracion_de_contenido_de_paginas_web.php >

[2] Sistemas para administración de contenido de páginas Web [en línea]. México, Milenium Network® The Internet Marketing Newsletter. Disponible en World Wide Web:

< <http://www.informaticamilenium.com.mx/paginas/mn/articulo98.htm> >

[3] Vignette pone en el mercado la solución web de nueva generación [en línea]. Madrid. Disponible en World Wide Web:

<<http://www.vignette.com/portal/site/es/menuitem.62215d74e262b2ba32189210180141a0/?vgnnextoid=3012698c0aa2e010VgnVCM1000005610140aRCRD&vgnnext-selectedmenuitem=e4f2973963125a601eb11f10180141a0&gblvcmprguid=3012698c0aa2e010VgnVCM1000005610140aRCRD> >

[4] Soluciones innovadoras y alta especialización [en línea].2005. Disponible en World Wide Web:

< <http://www.vass.es/actividad/ebusiness.asp?idsol=21> >

[3] Zamitiz Román, Carlos Alberto. TEMAS ESPECIALES DE COMPUTACIÓN [en línea]. UNAM, Facultad de Ingeniería. Disponible en World Wide Web:

<<http://www.fi-b.unam.mx/pp/profesores/carlos/java/indice.html> >

[4] Méndez, Justo. Lenguajes de Programación [en línea]. 1997. Disponible en World Wide Web:

<<http://www.monografias.com/trabajos/lengprog/lengprog.shtml> >

[5] Ruiz Diego. El lenguaje C# y la plataforma .NET [en línea]. Capitulo Gratis. Disponible en World Wide Web: <<http://www.tectimes.com/onweb/seccion.aspx?s=2&numero=lpacu076> >

[6] Rodríguez Barreira, Noelia. Introducción a servicios web [en línea]. Septiembre, 2002. Disponible en World Wide Web: <<http://www.elrincondelprogramador.com/default.asp?pag=articulos/leer.asp&id=32>>

[7] Letelier Patricio, Penadés M^a Carmen. Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP) [en línea]. Universidad Politécnica de Valencia. Disponible en World Wide Web: < <http://www.willydev.net/descargas/masyxp.pdf> >

[8] Ayuda de la herramienta Case Rational Rose.

[9] Booch Grady, Jacobson Ivar, Rumbaugh James. El Proceso Unificado de Desarrollo de Software. Addison Wesley 2000. Capítulo 1 Páginas 1-12.

[10] Larman Craig. UML y Patrones. Prentice Hall Hispanoamericana 1999. Parte I Introducción. Páginas 17-26.

[11] Booch Grady, Jacobson Ivar, Rumbaugh James. El Proceso Unificado de Desarrollo de Software. Addison Wesley 2000. Capítulo 8 Páginas 165-181, 185-204.

[12] Booch Grady, Jacobson Ivar, Rumbaugh James. El Proceso Unificado de Desarrollo de Software. Addison Wesley 2000. Capítulo 9 Páginas 205-210.

[13] Badani, Sigifredo E. Hernández. Métricas de estimación de tamaño Puntos de Caso de Uso. Septiembre 2002.

GLOSARIO DE TÉRMINOS

ASP.NET: conjunto de tecnologías de desarrollo de aplicaciones web comercializado por Microsoft para construir sitios web, aplicaciones web y servicios XML. Forma parte de la plataforma .NET de Microsoft y es la tecnología sucesora de la tecnología Active Server Pages (ASP).

Content: representa los contenidos que administrará el sistema.

Content Category: representa una categoría de contenido, puede ser amor, deporte, naturaleza, etc.

Customer: representa al cliente final que es un teléfono móvil.

EJB: Enterprise Java Beans es un API que forma parte del estándar de construcción de aplicaciones empresariales J2EE de Sun Microsystems.

Forbidden Shortcode: es un shortcode que no debe ser creado pues pueden formarse palabras desagradables.

J2EE: Java 2 Enterprise Edition es una plataforma de programación para desarrollar y ejecutar software de aplicaciones en lenguaje de programación Java.

JSP: Java Server Page es una tecnología Java que permite a los programadores generar contenido web dinámico, en forma de documentos HTML, XML o de otro tipo. En las páginas JSP se escribe el texto que va a ser devuelto en la salida (normalmente, código HTML) incluyendo código java dentro de él, para poder modificar o generar contenido dinámicamente.

Keyword: representa una palabra clave que está asociada a un tipo de contenido y tiene un conjunto de palabras con las que el cliente final puede referirse a un tipo de contenido, ejemplo: image, img, imag.

Manager: actor que gestiona todos los objetos del sistema.

Portlets: objetos que producen fragmentos de código de marcación que son agregados en una página de un portal. Una página de un portal se visualiza como una colección de ventanas portlet no solapadas, donde cada ventana portlet muestra un portlet. Por lo tanto un portlet (o colección de portlets) se asemeja a una aplicación web que está hospedada en un portal.

Price Category: representa una categoría de precio, que enmarca un rango de precios, ejemplo: barato de cero a tres, medio de tres a cinco, etc.

Procyon: empresa de desarrollo de software creada en el año 2004.

Provider: representa la compañía proveedora de contenidos.

Reseller: representa la compañía que va a vender los contenidos.

RUP: Rational Unified Process es una metodología de desarrollo de software, es un conjunto de procedimientos, técnicas, herramientas y un soporte documental que ayuda a los desarrolladores a realizar un nuevo software.

Shortcode: es un código de cuatro caracteres creado dinámicamente que identifica un contenido cuando se va a descargar para el móvil.

Transaction: datos de una acción que se realiza sobre el sistema.

UML: Unified Modeling Language es un lenguaje de modelado gráfico para visualizar, especificar, construir y documentar los artefactos de un sistema que involucra una gran cantidad de software.

User: representa a los usuarios del sistema, que se especializan en tres tipos, **ManagerUser**, **ProviderUser** y **ResellerUser**.

Watermark: representa una marca de agua, está formado por tres imágenes (media, clara, oscura) que se superponen sobre las imágenes a vender por determinado revendedor.

W3C: El consorcio W3C es una asociación internacional que se dedica al desarrollo de estándares y pautas para la web.