

Universidad de las Ciencias Informáticas

Facultad 2



Título: Devices Manager

Trabajo de Diploma para optar por el título de
Ingeniero Informático

Autor(es): Yeilin Pérez Martínez

Pedro Pablo Peña

Tutor(es): Ing. Damián Ilizastegui

Ing. Lex Karel Zayas

Consultante: Lic. Glicerio Grenot Isac

Ciudad de la Habana, 27 de junio del 2008

DECLARACIÓN DE AUTORÍA

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

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

Yeilin Pérez Martínez

Autora

Pedro Pablo Peña

Autor

Ing. Damian Ilizastegui

Tutor

Ing. Lex Carel Zayas

Tutor

AGRADECIMIENTOS

A mis padres, mi hermano, mi cuñada y mi familia por estar siempre apoyándome en todo lo que necesite.

A mis tutores y mis compañeros de proyecto que siempre me han ayudado de una forma u otra.

A los trabajadores de la empresa Procyon que me han ayudado y enseñado mucho desde que entré a ese proyecto de mi facultad.

A mis amistades entre ellas Daiamna, Rita, Ivette, Maydelis, Alejandro, William, Julio, Arián, Dariena y aquellos que aunque no pueda nombrarlos todos aquí, siempre han estado ahí en el momento que los necesité.

A todas las personas que de una forma u otra me han ayudado y apoyado siempre.

Yeilin Pérez Martínez

A mis tutores y compañeros de estudio y amigos que me ayudaron y me criticaron algunas cosas, especialmente a Rigo que siempre me criticaba cualquier cosa "sucio" o fea que hacía.

A los compañeros de Procyon que siempre estuvieron dispuestos para aclarar cualquier duda, especialmente a Glicerio Isac nuestro consultor.

A la Revolución por haberme dado la oportunidad de estudiar gratuitamente y de formarme como profesional de las ciencias informáticas y a todos los que de alguna forma u otra han hecho posible este sueño.

Pedro Pablo Peña

DEDICATORIA

A mis padres por brindarme siempre tanto amor, confianza, ayuda y protección, a ellos que los quiero tanto.

A mi hermano y mi cuñada por estar siempre ahí preocupándose por mí y que muy pronto me harán tía.

Yeilin Pérez Martínez

Le dedico este trabajo a mi familia que siempre ha confiado en mí, especialmente a mi madre, mis hermanos, mis abuelitos y a mi novia Yadira que siempre me ha dado mucho amor y cariño.

Pedro Pablo Peña

RESUMEN

Los teléfonos celulares han evolucionado a medida que transcurren los años. En la actualidad además de permitir la comunicación de una forma más fácil y agradable para las personas, se les puede incorporar distintas aplicaciones que aumenten el grado de satisfacción de los clientes.

Debido que existe una gran cantidad de dispositivos móviles y cada uno de ellos presenta diferentes características, es necesario conocer las características de cada dispositivo para el desarrollo de estas aplicaciones. Por lo tanto surge la necesidad de tener un software que manipule los datos de estos dispositivos, los cuales serán usados o consultados para desarrollar estas aplicaciones.

La empresa Procyon desarrolla distintas aplicaciones para dispositivos móviles. Teniendo en cuenta el auge de la demanda de contenido para teléfonos celulares que existe actualmente, la empresa decidió desarrollar el proyecto *Devices Manager*, el cual no es más que un sistema de manipulación de datos de dispositivos móviles basado en servicios web.

El sistema cuenta con una aplicación web como cliente del servicio web, que permite interactuar con el mismo y desde el cual se administran todos los procesos. La aplicación cuenta con una interfaz gráfica donde el usuario puede entrar los datos a ser procesados por la aplicación. El sistema permite manipular los datos de los dispositivos móviles, es decir, permite adicionar, actualizar, eliminar y consultar los datos de los dispositivos móviles, sus características y los User Agent que son cadenas que envían los dispositivos para identificarse cuando hace alguna solicitud.

INDICE

INTRODUCCIÓN	1
CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA	5
1.1 Introducción.....	5
1.2 Manipulación de datos de los dispositivos móviles.....	5
1.3 Bases de datos que permiten la manipulación de datos de dispositivos móviles.....	6
1.4 Lenguajes de Programación.....	9
1.5 Tecnologías a utilizar.....	10
1.5.1 J2EE.....	10
1.5.1.1 EJB3.....	11
1.5.2 JSP.....	11
1.6 Metodología de desarrollo.....	12
1.7 Servidor de aplicaciones.....	14
1.7.1 Jboss.....	14
1.8 Herramientas.....	14
1.8.1 Axis.....	15
1.8.2 UML.....	15
1.8.3 MySQL.....	16
1.8.4 Rational Rose.....	16
1.8.5 Eclipse.....	17
1.8.6 Sistemas de control de versiones: Subversion.....	18
1.9 Conclusiones.....	19
CAPÍTULO II: CARACTERÍSTICAS DEL SISTEMA	20
2.1 Introducción.....	20
2.2 Problema y situación problemática.....	20
2.3 Objeto de automatización.....	22
2.4 Propuesta del sistema.....	22
2.5 Modelo de dominio.....	24

2.6	Relación de los Requerimientos	26
2.6.1	Requerimientos funcionales.....	26
2.6.2	Requerimientos no funcionales.....	27
2.7	Definición de los casos de uso	28
2.7.1	Definición de los actores.....	28
2.7.2	Diagrama de casos de uso	29
2.7.3	Casos de uso expandidos	29
2.8	Conclusiones	32

CAPÍTULO III: ANÁLISIS Y DISEÑO DEL SISTEMA33

3.1	Introducción	33
3.2	Análisis.....	33
3.2.1	Diagrama de clases del Análisis	33
3.2.2	Diagramas de colaboración de los CU más significativos.....	37
3.3	Diseño.....	39
3.3.1	Descripción de la Arquitectura utilizada.....	39
3.3.2	Patrones de Diseño.....	40
3.3.3	Diagrama de clases del Diseño	41
3.3.4	Diagramas de secuencia de los CU más significativos.....	47
3.4	Diseño de la base de datos.....	50
3.4.1	Modelo lógico de datos	51
3.4.2	Modelo físico de datos	51
3.5	Conclusiones	52

CAPÍTULO IV: IMPLEMENTACIÓN53

4.1	Introducción	53
4.2	Diagrama de Despliegue	53
4.3	Diagrama de Componentes	54
4.4	Conclusiones	56

CAPÍTULO V. ESTUDIO DE FACTIBILIDAD.....	57
5.1 Introducción	57
5.2 Planificación.....	57
5.2.1 Características del proyecto.	57
5.2.2 Cálculo de las instrucciones fuentes, esfuerzo, tiempo de desarrollo, cantidad de hombres y costo.	59
5.3 Beneficios tangibles e intangibles.....	65
5.4 Análisis de costo.....	65
5.5 Conclusiones	66
CONCLUSIONES.....	67
RECOMENDACIONES	68
BIBLIOGRAFÍA.....	69
REFERENCIAS BIBLIOGRAFICAS	71
ANEXOS	73
GLOSARIO.....	92

INTRODUCCIÓN

Desde sus comienzos en los finales de la década de los 70 los teléfonos celulares han revolucionado las tareas que se realizan cotidianamente y se han convertido en una herramienta primordial para todos, actualmente existen más de 2800 millones de abonados por todo el planeta.

Actualmente se está produciendo una auténtica convulsión en la evolución de los dispositivos móviles. Olvidando aquellos tiempos en que móvil significaba solamente teléfono aparece un nuevo concepto en el que se engloban muy distintos tipos de terminales, con características y capacidades bien diferenciadas. La evolución hacia las nuevas generaciones crea necesidades y proporciona posibilidades que antes sólo estaban al alcance de unos pocos dispositivos.

Con el transcurso de los años los teléfonos celulares han incrementado sus capacidades de hardware y software, lo que ha posibilitado el desarrollo de aplicaciones destinadas a ellos, la creación de plataformas para administración de contenidos se ha convertido en el eslabón fundamental dentro de la industria de software en la telefonía celular.

Los dispositivos móviles son muy diversos y cada uno tiene diferentes características que lo identifican, de ahí la necesidad de tener un software que contenga la información de los diferentes dispositivos móviles y que sea capaz de suministrar dicha información a otra aplicación que se encargue de responder a las solicitudes de estos dispositivos en dependencia de las características del mismo.

En Procyon, entidad que se encuentra enmarcada dentro de la infraestructura productiva (IP) de la Universidad de las Ciencias Informáticas, se fomenta el uso de las nuevas tecnologías de comunicación y la producción de aplicaciones y contenidos para móviles. Aunque la empresa desarrolla juegos y contenidos para móviles, tales como temas, ringtones; la tarea fundamental de la empresa en estos momentos es la producción de una plataforma de administración de contenidos para móviles.

La empresa Procyon implementó una plataforma de administración de contenidos para teléfonos móviles. Dicha plataforma permitía descargar imágenes, ringtones, temas, entre otros contenidos. Teniendo en cuenta el crecimiento de la demanda de software para la administración de contenidos y

la manipulación de los datos de los dispositivos móviles surge la necesidad de migrar el proyecto *Devices Manager* de la plataforma y mejorarlo agregándole nuevas funcionalidades. Esta plataforma era muy abarcadora pero la tecnología usada para la realización de la misma era una tecnología obsoleta. Debido a esto Procyon decidió utilizar nuevas tecnologías para desarrollar sus aplicaciones, las cuales permitirían un mejor mantenimiento del software. A partir de esta situación se lleva a cabo la puesta en marcha de la investigación con el fin de resolver el siguiente problema:

¿ Cómo desarrollar a partir del proyecto Devices Manager, un software para la manipulación de los datos de los dispositivos y la gestión de los mismos, que se acople a la nueva versión de la plataforma que se está desarrollando?

Con vista a la solución del problema anterior se plantea como Objeto de estudio:

Desarrollo de aplicaciones de manipulación de los datos de los dispositivos móviles.

A partir del objeto de estudio se delimita el siguiente Campo de acción:

Desarrollo de un software para la manipulación de los datos de los dispositivos móviles a partir del proyecto Devices Manager.

Con el fin de solucionar el problema planteado anteriormente se define como objetivo general de este trabajo:

Migrar la plataforma de gestión de dispositivos utilizando nuevas tecnologías para que se acople a la nueva versión de la plataforma que se está desarrollando y se logre la correcta manipulación de los datos de los dispositivos.

Para cumplir este objetivo general se trazaron los siguientes objetivos específicos:

- *Diseñar una nueva base de datos a partir de la base de datos existente.*
- *Estudiar metodologías de desarrollo de software.*
- *Analizar y diseñar los nuevos EJB (Enterprise Java Beans) para la interacción con la base de datos de los dispositivos.*
- *Publicar un servicio Web con las funcionalidades que ofrece el sistema.*

- *Implementar una aplicación Web para el despliegue de la información de los dispositivos.*

Además se plantearon las siguientes preguntas investigativas:

- *¿Cómo lograr la correcta manipulación de los datos de los dispositivos móviles?*
- *¿Cómo funcionan los Servicios Web?*
- *¿Cómo desplegar correctamente la información de los dispositivos móviles en la aplicación?*

Se desarrollarán las siguientes tareas para dar cumplimiento a los objetivos trazados:

- *Análisis de las tecnologías existentes a nivel mundial que permitan de forma eficiente la manipulación de los datos de los dispositivos móviles.*
- *Estudio y selección de las principales metodologías de desarrollo establecidas a nivel mundial.*
- *Análisis e implementación de una arquitectura que satisfaga los objetivos que nos planteamos.*
- *Estudio del funcionamiento de los EJB3 para la persistencia de los datos.*
- *Estudio e implementación de patrones de diseño para lograr un software robusto, flexible y acabado.*
- *Diseño e implementación del servicio web y la aplicación web.*

Estructuración del contenido:

Capítulo 1: Fundamentación teórica. En este capítulo se hace un análisis del tema tratado, se investiga acerca de los sistemas informáticos vinculados al campo de acción, se fundamentan las herramientas, las técnicas, el lenguaje y la metodología que se utilizaron para lograr la solución del problema que nos enfrentamos.

Capítulo 2: Características del Sistema. Se describe el modelo del negocio y la descripción de los procesos que serán objeto de automatización. Presenta la propuesta del 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. Incluye el análisis y diseño del sistema, obteniendo como resultado principal, los diagramas de clases del análisis y del diseño, así como los diagramas de interacción para cada realización de los casos de uso. Muestra la descripción de la arquitectura y patrones de diseño empleados. Incluye el diseño de la base de datos.

Capítulo 4: Implementación. 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. Se describe la aplicación de la técnica de estimación del esfuerzo. Se realiza un análisis de los costos y beneficios del proyecto.

CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA

1.1 Introducción.

En este capítulo se hace un análisis sobre la manipulación de los datos de los dispositivos móviles y se investiga acerca de los sistemas informáticos vinculados al campo de acción, se fundamentan las herramientas, las técnicas, las metodologías y el lenguaje que se utilizaron para lograr la solución del problema planteado anteriormente.

1.2 Manipulación de datos de los dispositivos móviles

La telefonía móvil está experimentando una evolución donde el inicial dispositivo de comunicación por voz aporta nuevas posibilidades, convirtiéndose en todo un dispositivo móvil programable capaz de transmitir y recibir datos.

Con la creciente que lleva el mercado celular en los últimos años, no solo los usuarios y los equipos disponibles para ellos crecen, sino que cada vez más empresas y desarrolladores, están creando aplicaciones y contenidos para estos dispositivos en diversas partes del mundo. Entre las compañías líderes en el desarrollo de aplicaciones o plataformas para móviles podemos mencionar a *Sony Ericsson, Nokia, PalmOne y Symbia*.

Pero, a diferencia de lo que ocurre con una computadora, para desarrollar contenido o aplicaciones para celulares, el desarrollo tiene que ser pensando para una cantidad muy grande de diferentes dispositivos. Si bien es cierto que hay similitudes entre los dispositivos móviles, las diferencias en tamaños y disposiciones de pantalla, modelo, marca, los diferentes soportes de sistemas operativos, la capacidad de cada uno en cuanto a soporte para los diferentes formatos de audio y video, cantidad de colores, velocidad de transmisión de datos, capacidad de procesamiento de imágenes, entre otros aspectos; hacen que la prueba final de cada contenido o aplicación sea muy costosa y engorrosa si se debe probar con todos los teléfonos que se pretende lo puedan aplicar.

Debido a esto la manipulación de los datos de los diversos dispositivos móviles es un eslabón fundamental dentro del desarrollo de plataformas de administración de contenidos para móviles. Previa a la descarga e instalación de cualquier contenido en un dispositivo móvil es importante conocer sus características, de manera que se pueda saber de forma rápida qué archivos son compatibles con el mismo y cuáles no.

Normalmente cuando se quiere descargar una imagen, un ringtone, un juego o cualquier otro contenido para el móvil, se hace una petición al servidor y este en dependencia de los privilegios que se tengan dará acceso al contenido y dejará descargarlo, pero detrás de este proceso hay otro proceso que suministra los datos del dispositivo que hace la petición a la plataforma, para que esta pueda dar una respuesta adecuada a dicha petición. Por ejemplo si un usuario quiere descargar una imagen para su móvil, el software que da respuesta a esta petición deberá conocer una serie de características de dicho móvil para que el usuario descargue la imagen con el tamaño adecuado para su teléfono.

Es por ello que es importante tener un sistema de manipulación de datos de los dispositivos móviles, en el cual se encuentren en una base de datos cada una de las características de los mismos. De esta forma se podrá dar una respuesta adecuada a las solicitudes de los dispositivos.

1.3 Bases de datos que permiten la manipulación de datos de dispositivos móviles

En el mundo existen varias bases de datos que permiten la manipulación de los datos de los dispositivos móviles, las más famosas son las bases de datos del UAProf (*User Agent Profile*), del WURFL (*Wireless Universal Resource File*) y la del Polish.

El UAProf se diseñó como un marco común para que todos los fabricantes de teléfonos móviles pudieran describir las características técnicas y la funcionalidad de sus teléfonos mediante el uso de XML y CC/PP (*Composite Capabilities/Preferences Profile*) traducido al español como Capacidades de Composición / Perfiles de Preferencias. CC/PP es un marco más amplio que permite expresar no sólo la funcionalidad de los teléfonos sino también las preferencias del usuario. De hecho, el UAProf es la primera implementación de la norma CC/PP. Mediante el acceso a su sitio www.UAProfile.com se

puede acceder a la información de cada dispositivo móvil, dicha información la da en un xml independiente.

El UAProf ha sido acogido por la mayoría de fabricantes. Sin embargo, no hay homogeneidad en el uso de las reglas CC/PP, es decir, los fabricantes utilizan términos diferentes para referirse al mismo atributo o funcionalidad. Por otra parte, los perfiles UAProf no son 100% fiables, los valores indicados no siempre son precisos y requiere un proceso exhaustivo de prueba para corregir dichos errores. Además, a veces la URL anunciada por el móvil es incorrecta o el servidor que aloja el UAProf no está disponible. Es decir, UAProf no es una fuente fiable de información.

Por último, la información ofrecida por los perfiles UAProf es incompleta para el objetivo que fue diseñado. Para ofrecer una adaptación óptima del contenido en cualquier teléfono hay información clave que no está recogida en UAProf, entre otras:

- * Versión CSS soportada
- * Atributos CSS no soportados
- * Tamaño de las barras de desplazamiento (scroll bars)
- * Dimensiones del área de navegación
- * Tamaño máximo de descarga de video/audio
- * Dimensiones del salvapantallas e imágenes de fondo (wallpapers)

Por otra parte la base de datos del WURFL es dentro del ámbito de desarrollo para celulares, una de las fuentes más completas a la cual recurrir si se quiere lograr aplicaciones que funcionen realmente en todos los teléfonos. Para quienes se encargan de la venta de contenido multimedia, es la herramienta en la cual basarse para saber que contenido entregar a cada aplicación.

El WURFL es una base de datos gratuita en formato xml que contiene la información de las características de la mayoría de los dispositivos móviles existentes. A través del acceso a su sitio <http://wurfl.sourceforge.net/> se puede acceder a su base de datos completa y un usuario puede actualizar su archivo xml. El WURFL contiene la información de los dispositivos pero no cuenta con una interfaz desde la cual se pueda solicitar las características de un dispositivo atendiendo a un criterio determinado y que sea accesible desde otra aplicación; es más bien una forma eficiente de leer

la información contenida en esa base de datos y utilizarla para personalizar las aplicaciones de los usuarios que necesiten la información de las diferentes familias de dispositivos móviles. Mientras que el UAProf da la información de cada dispositivo en un xml independiente, el WURFL da toda la información de todos los dispositivos móviles en un solo xml.

A diferencia de los otros sistemas, la base de datos del WURFL se va completando con datos aportados por usuarios de los dispositivos móviles, es una de las ventajas por ser software libre. Esto tiene el problema de que contiene información no autenticada por los fabricantes. Sin embargo, tiene la ventaja de que se puede recoger información sobre todo tipo de dispositivos (incluso de los que no tienen perfil UAProf). Además, este sistema se complementa con APIs (*Application Programming Interface*) de consulta para su base de datos (que es un fichero XML) en diversos lenguajes de programación (Java, PHP, Perl, Ruby, Python, dotNet,...). Por todo esto, es un sistema mucho más usado y fiable que UAProf.

En el caso del Polish, en su sitio <http://devices.i2mepolish.org/interactivedb/welcome.faces> se puede encontrar su base de datos, la cual mantiene información de más de 300 dispositivos móviles y sus respectivas características tales como el tamaño del canvas, APIs, formatos soportados, entre otras. Estas características están escritas en xml y por tanto pueden ser usadas por los usuarios en sus aplicaciones y ampliarlas fácilmente. Tiene la desventaja de que no cubre toda la amplia gama de características que tienen los dispositivos móviles, sino que se especializa en una parte de ella.

En nuestro país, no abundan los teléfonos celulares, por lo que no hay mucha demanda de contenido para dispositivos móviles. Por lo tanto, no existen sistemas que permitan la manipulación de datos de dispositivos móviles. Recientemente se ha comenzado a dar pasos para aumentar el uso de la telefonía móvil y facilitar el uso de dispositivos móviles a la población. Debido al incremento del uso de los teléfonos celulares que pudiera haber en el país se hace necesario desarrollar aplicaciones para estos dispositivos. Es por esto que la empresa Procyon juega un papel importante debido que es la única empresa que ha incursionado en este campo.

1.4 Lenguajes de Programación

Se decidió utilizar el lenguaje de programación Java debido que es una tecnología madura, extremadamente eficaz y versátil. Este lenguaje permite a los desarrolladores desarrollar software en una plataforma y ejecutarlo prácticamente en cualquier otra y combinar aplicaciones o servicios basados en la tecnología Java para crear servicios o aplicaciones totalmente personalizados. [1]

Es libre y les proporciona a los desarrolladores una mayor flexibilidad, ahorro en los costes de desarrollo, independencia, compatibilidad con otros sistemas, entre otros beneficios. [2]

La tecnología Java está compuesta por 2 partes: el lenguaje de programación Java, y la plataforma Java.

El lenguaje Java es un lenguaje orientado a objetos de alto nivel y sencillo. Brinda un gran nivel de seguridad, capacidad multihilo, gran rendimiento y creación de aplicaciones distribuidas.

Otra ventaja que presenta es que un programa escrito en Java puede ejecutarse independientemente de la plataforma (hardware, software y sistema operativo) en la que se esté actuando. El código fuente se compila a un lenguaje intermedio, llamado *bytecodes*, cercano al lenguaje máquina pero independiente del ordenador y el sistema operativo en que se ejecuta. Ese lenguaje intermedio se interpreta por medio de la máquina virtual de Java (JVM), que sí depende de la plataforma. Los *bytecodes* permiten el ya conocido “*write once, run anywhere*” (compila una sola vez y ejecútalo donde quieras). Esto significa que mientras el ordenador tenga una JVM, podemos compilar nuestros programas a *bytecodes* en cualquier plataforma. [3]

La plataforma Java a diferencia de otras, es sólo de software y se ejecuta sobre otras plataformas de hardware. Tiene dos componentes: la máquina virtual de Java y el Java API. La máquina virtual de Java es la base de la plataforma Java, es el entorno en el que se ejecutan los programas implementados en Java, su misión principal es la de garantizar la portabilidad de las aplicaciones realizadas en Java. El Java API es una gran colección de componentes de software que proporcionan muchas utilidades para el programador, por ejemplo, los API's para las interfaces gráficas. [3]

1.5 Tecnologías a utilizar

Actualmente existen diversas tecnologías de programación que son muy importantes en el desarrollo de aplicaciones. Para la realización del sistema se decidió utilizar J2EE debido a que es la plataforma de Java especialmente pensada para crear aplicaciones Web, el framework EJB3 para el mapeo de los datos en la base de datos y JSP para el desarrollo de la aplicación web.

1.5.1 J2EE

J2EE (*Java 2 Enterprise Edition*) es un conjunto de herramientas que crean un escenario ideal para el desarrollo y despliegue de aplicaciones escalables en la Web.

Este tipo de aplicaciones tiene una arquitectura en capas: una capa de presentación que proporciona la interfaz de usuario, una o más capas intermedias que proporcionan la lógica de negocio de la aplicación y una capa final con los sistemas de información que mantienen aplicaciones y bases de datos. J2EE integra un conjunto de APIs, frameworks y patrones de programación: en la capa de presentación están las páginas JSP (*Java Server Pages*), en la capa de negocio tenemos los componentes *Enterprise JavaBeans*, *servlets* y en la capa de aplicaciones se encuentra el acceso a bases de datos con JDBC (*Java Database Connectivity*). [4]

La plataforma J2EE presenta numerosas ventajas:

- Es portable, la aplicación desarrollada puede utilizarse en cualquier sistema operativo para el que haya disponible una Máquina Virtual Java.
- Es escalable, se puede añadir nuevos componentes J2EE a una aplicación Web para soportar al aumento de clientes, sin tener que reescribir todo el código de nuevo.
- Es altamente soportada, prácticamente cualquier gran empresa de software tiene un servidor de aplicaciones Web compatibles con J2EE.
- Es segura, el entorno de seguridad de la plataforma J2EE permite que se definan restricciones de seguridad en el momento de despliegue de la aplicación, aislando así las aplicaciones de la complejidad de las implementaciones de seguridad.

1.5.1.1 EJB3

EJB (*Enterprise Java Beans*) es un marco de trabajo para el desarrollo de aplicaciones empresariales en Java. Es un framework para la capa de persistencia de datos y además se utiliza para forzar las reglas de negocio, para encapsular la lógica de negocio y para acoplar el modelo de negocio con la aplicación empresarial. Evita la tarea de integrar diferentes frameworks para cada capa distinta de la aplicación. La versión actual de EJB (EJB3.0) incluye el nuevo "API de Persistencia de Java". Es un modelo más sencillo para la implementación de fachadas y por compatibilidad, incluye las APIs del modelo anterior (EJB2.1). [5]

Entre los objetivos buscados y conseguidos por esta especificación se pueden encontrar el simplificar la complejidad en la programación de objetos persistentes, que en versiones anteriores eran objetos realmente pesados para el servidor, muy ineficientes, y complejos de programar.

Los EJB3 se despliegan y ejecutan en un servidor de aplicaciones certificado para soportarlos. Presenta la ventaja de que la curva de aprendizaje es muy corta. También reduce el número de clases y número de interfaces que los programadores deben programar y el número de artefactos (archivos de configuración, descriptores de despliegue) que se requieren para que la aplicación funcione. Otra ventaja que presenta es que evita los anti-patrones y actualiza y elimina los datos en cascada.

1.5.2 JSP

JSP (*Java Server Pages*) es una tecnología Java para crear aplicaciones Web. Es un desarrollo de la compañía *Sun Microsystems*, y su funcionamiento se basa en scripts, que utilizan una variante del lenguaje Java.

Esta tecnología Java permite a los programadores generar contenido dinámico para Web, en forma de documentos HTML, XML, o de otro tipo. Debido que la parte dinámica está escrita en Java, no en VBScript o cualquier lenguaje específico de ASP que es la tecnología competidora de Microsoft, es mejor y más poderosa para desarrollar aplicaciones que requieren componentes reutilizables.

Permite separar la parte dinámica de las páginas Web del HTML estático. También separa el código de presentación y el código de implementación, es decir que cambios realizados en el código HTML relativos a cómo son mostrados los datos, no interfieren en la lógica de programación y viceversa.

La principal ventaja de JSP frente a otros lenguajes es que permite integrarse con clases Java , lo que permite separar en niveles las aplicaciones Web, almacenando en clases Java las partes que consumen más recursos así como las que requieren más seguridad, y dejando la parte encargada de formatear el documento html en el archivo jsp. La idea fundamental es separar la lógica del negocio de la presentación de la información.

Otra ventaja es que JSP hereda la portabilidad de Java, y es posible ejecutar las aplicaciones en múltiples plataformas sin cambios. Es común incluso que los desarrolladores trabajen en una plataforma y que la aplicación termine siendo ejecutada en otra.

1.6 Metodología de desarrollo

Las metodologías de desarrollo de software son un conjunto de procedimientos, técnicas y ayudas a la documentación para el desarrollo de productos de software. Se van indicando paso a paso todas las actividades a realizar para lograr el producto informático deseado, qué personas deben participar en el desarrollo de las actividades y qué papel deben de tener. Además detallan la información que se debe producir como resultado de una actividad y la información necesaria para comenzarla. Las técnicas combinan el empleo de modelos junto con el empleo de procedimientos detallados. Una técnica determinada puede ser utilizada en una o más actividades de la metodología de desarrollo de software.

Los programadores tradicionales argumentan que la aplicación de una metodología supone una gran carga, pero si no se emplea una metodología pueden surgir problemas como: resultados impredecibles, detección tardía de errores, la introducción de nuevas herramientas afectará perjudicialmente al proceso, cambios de organización también afectarán al proceso y resultados distintos con nuevas clases de productos.

Para el desarrollo de este sistema como metodología de desarrollo se decidió utilizar RUP (*Rational Unified Process*) traducido al español como Proceso Unificado de Desarrollo, cuyo objetivo es producir software de alta calidad.

RUP utiliza el paradigma de orientación a objetos para su descripción e implementa las mejores prácticas de desarrollo de software. Está dirigido por casos de uso, estos capturan requerimientos funcionales y representan piezas de funcionalidad que brindan un resultado de valor al usuario. Está centrado en la arquitectura, comprende los aspectos estáticos y dinámicos más importantes del sistema. Es iterativo e incremental, el trabajo se divide en piezas pequeñas o miniproyectos; cada uno proveyendo un subproducto incremental. [6]

Como RUP es un proceso, en su modelación define como sus principales elementos: rol, actividad, artefacto y flujos de trabajo. Un rol es el papel que desempeña una persona en un determinado momento. La actividad es una unidad de trabajo que se asigna a un rol. Los artefactos son elementos de información producidos, modificados o usados por el proceso. Los flujos de trabajo son una lista de actividades, roles y artefactos.

La metodología RUP divide en 4 fases el desarrollo del software: Inicio, Elaboración, Construcción y Transición. Cada fase se completa con la realización de varias iteraciones en las que se desarrollan una serie de actividades que producen un resultado de valor observable que no son más que los flujos de trabajo. RUP está conformado por 9 flujos de trabajo que se dividen en dos grupos, 6 para el desarrollo del sistema: Modelación del Negocio, Requerimientos, Análisis y Diseño, Implementación, Prueba e Instalación y 3 para la gestión del proyecto: Administración del Proyecto, Configuración y Administración de Cambios y Administración del Entorno.

Cabe indicar que el modelo establece que el propio proceso es adaptable a cada caso de desarrollo, puesto que no todos los proyectos requieren del mismo nivel de documentación. El tamaño, la complejidad y el número de participantes, entre otros aspectos, aconsejan definir para el proyecto cómo adaptar el proceso. [7]

Entre las ventajas del RUP tenemos que reduce posibles riesgos altos ya que a través de un predefinido esquema de gestión de riesgos, se pueden reconocer previamente problemas y fallos de forma temprana y prevenirlos o corregirlos. Otra ventaja es que utiliza componentes y eso reduce un poco el tiempo de desarrollo del producto. Ayuda a controlar el desarrollo del software debido que proporciona diferentes elementos de planificación como el plan de desarrollo, plan de iteración, plan de calidad, entre otros.

1.7 Servidor de aplicaciones

1.7.1 Jboss

Jboss es un servidor de aplicaciones de uso libre. Un servidor de aplicaciones es un software encargado de ejecutar las aplicaciones Java en el lado del servidor y devolver las páginas o datos como resultado a las peticiones.

Jboss se ha convertido en uno de los servidores de aplicaciones más usados mundialmente y en el preferido de la comunidad más amplia del mercado. Es un servidor de aplicaciones basado en J2EE, e implementado al 100% en Java. Este servidor de aplicaciones al estar basado en Java, puede ser utilizado en cualquier sistema operativo que lo soporte. Jboss implementa todo el paquete de servicios de J2EE.

Mediante Jboss es posible llevar a cabo un desarrollo con Enterprise Java Bean. Uno de los rasgos más importantes de Jboss es su apoyo a la implementación, por ejemplo si se quiere implementar un nuevo EJB, con solo copiar el archivo correspondiente en el directorio correspondiente y si esto se hace mientras el Bean ya está cargado, Jboss lo descarga automáticamente, y entonces carga la nueva versión.

1.8 Herramientas.

Las herramientas de programación, son aquellas que permiten realizar aplicaciones, programas, rutinas, utilitarios y sistemas de software. Hoy día existen múltiples herramientas de programación en el mercado. Para la realización de este sistema se decidió utilizar herramientas como Axis, UML, MySQL, *Rational Rose*, *Eclipse* y el Sistema de Control de Versiones: *Subversion*.

1.8.1 Axis

Apache Axis es una implementación *OpenSource* de SOAP (*Simple Object Access Protocol*) que proporciona un entorno de ejecución para Servicios Web implementados en Java. Un Servicio Web es una colección de protocolos y estándares que sirven para intercambiar datos, entre distintas aplicaciones de software desarrolladas en lenguajes de programación diferentes, y ejecutadas sobre cualquier plataforma.

Axis proporciona herramientas para crear WSDL (*Web Services Description Language*) desde clases java y clientes Java desde un WSDL. Brinda la posibilidad de desplegar, probar y monitorizar Servicios Web. Además se integra con servidores de aplicaciones y contenedores de Servlets.

WSDL2java es una herramienta de Axis que interpreta ficheros WSDL y emite código Java que encapsula toda la intercomunicación entre los Servicios Web. Facilita la labor de obtener la información necesaria para invocar un Servicio Web (url, nombre, parámetros, tipo de los parámetros, etc), ya que dicha información se detalla en un fichero WSDL. Reduce el esfuerzo de realizar llamadas remotas al hacer el proceso interno más transparente al usuario. [8]

1.8.2 UML

El lenguaje de modelado UML (*Unified Modeling Language*) es el estándar de descripción de esquemas de software y sirve para especificar, visualizar y documentar esquemas de sistemas de software orientado a objetos. UML no es un método de desarrollo, no sirve para determinar qué hacer en primer lugar o cómo diseñar el sistema, sino que simplemente ayuda a visualizar el diseño y a hacerlo más accesible para otros. Divide cada proyecto en un número de diagramas que representan las distintas vistas del proyecto y juntos representan la arquitectura del mismo:

- Diagrama de casos de uso que muestra a los actores, los casos de uso y sus relaciones.
- Diagrama de clases que muestra las clases y la relaciones entre ellas.
- Diagrama de secuencia que muestra los objetos y sus múltiples relaciones entre ellos.
- Diagrama de colaboración que muestra objetos y sus relaciones, destacando los objetos que participan en el intercambio de mensajes.

- Diagrama de estado que muestra estados, cambios de estado y eventos en un objeto o en parte del sistema.
- Diagrama de actividad que muestra actividades, así como los cambios de una a otra actividad junto con los eventos que ocurren en ciertas partes del sistema.
- Diagrama de componentes que muestra los componentes de mayor nivel de la programación.
- Diagrama de implementación que muestra las instancias de los componentes y sus relaciones.

El uso de UML presenta numerosos beneficios en cuanto a: diseño y documentación, descubrimiento de fallas, ahorro de tiempo en el desarrollo del software y las modificaciones son mucho más fáciles de realizar.

1.8.3 MySQL

El software MySQL proporciona un servidor de base de datos SQL (*Structured Query Language*) muy rápido, multi hilo, multi usuario y robusto. El servidor MySQL está diseñado para entornos de producción críticos que presenten alta carga de trabajo así como para integrarse en software para ser distribuido. [9]

Presenta numerosas ventajas, entre las que podemos citar que añade velocidad y flexibilidad, debido que almacena datos en tablas separadas en lugar de poner todos los datos en un gran almacén. Es *Open Source*, esto significa que es posible para cualquiera usar y modificar el software, cualquiera puede bajar el software MySQL desde internet y usarlo sin pagar nada. Además, se puede estudiar el código fuente y cambiarlo para adaptarlo a sus necesidades. Tiene un sistema de privilegios y contraseñas que es muy flexible y seguro, y que permite verificación basada en el host. Las contraseñas son seguras porque todo el tráfico de contraseñas está encriptado cuando se conecta con un servidor. MySQL server tiene soporte para comandos SQL para chequear, optimizar, y reparar tablas.

1.8.4 Rational Rose

Rational Rose es la herramienta líder en el mundo para el modelado de sistemas complejos y de tiempo real. Unifica todos sus equipos de desarrollo a través del modelamiento, el cual está basado en el lenguaje de modelado UML. También permite visualizar, entender, y refinar los requerimientos y

arquitectura antes de enfrentarse al código, esto le permite evitar esfuerzos desperdiciados en el ciclo de desarrollo.

Rational Rose unifica a los analistas de negocios, de sistemas y de datos al permitirles crear y administrar modelos en una herramienta con un solo lenguaje de modelado. Ofrece la capacidad de modelar y visualizar los procesos de negocios. [10]

Es la herramienta con más experiencia para modelar el desarrollo de un software y en este sector una de las más completas de su tipo. Es fácil de usar y la curva de aprendizaje es corta. Utiliza un proceso de desarrollo iterativo controlado, donde el desarrollo se lleva a cabo en una secuencia de iteraciones. Puede haber varias personas trabajando a la vez, cada desarrollador puede operar en un espacio de trabajo privado que contiene el modelo completo y tenga un control exclusivo sobre la propagación de los cambios en ese espacio de trabajo. Se puede generar código en distintos lenguajes de programación a partir de un diseño en UML. Proporciona mecanismos para realizar la denominada Ingeniería Inversa, es decir, a partir del código de un programa, se puede obtener información sobre su diseño. [11]

1.8.5 Eclipse

Eclipse es un Entorno Integrado de Desarrollo (IDE) abierto, extensible y *Open Source*. Eclipse está escrito en su mayor parte en Java y se ejecuta sobre la máquina virtual de ésta. Además permite de una forma muy cómoda, trabajar con varias tecnologías asociadas.

La característica clave de Eclipse es la extensibilidad. Eclipse es una gran estructura formada por un núcleo y muchos *plugins* que van conformando la funcionalidad final. La forma en que los *plugins* interactúan es mediante interfaces o puntos de extensión; así, las nuevas aportaciones se integran sin dificultad ni conflictos. Dispone de una comunidad de desarrolladores así como de la posibilidad de creación e instalación de los *plugins* que se consideren necesarios.

Otra ventaja que presenta Eclipse es que los errores de compilación se muestran en tiempo real subrayando el fragmento de código adecuado con una línea roja. Y además el entorno

automáticamente compila los archivos salvados. Así pues, no será necesario pasar por el lento proceso de compilar, observar los errores y corregirlos después.

Es un IDE basado en archivos: los recursos como clases java, archivos HTML, archivos XML, están almacenados en un sistema de archivo y así se puede obtener fácilmente acceso a ellos.

El marco de trabajo de Eclipse permite a los desarrolladores construir mejores herramientas combinando los complementos de Eclipse en un entorno de herramientas personalizado, además de facilitar la interoperabilidad y la facilidad de uso de una serie de herramientas heterogéneas que se ejecutan bajo un entorno común.

1.8.6 Sistemas de control de versiones: Subversion

Un sistema de control de versiones es un software que administra el acceso a un conjunto de ficheros, y mantiene un historial de cambios realizados. El control de versiones es útil para guardar cualquier documento que cambie con frecuencia, por ejemplo el código fuente de un programa.

Normalmente consiste en una copia maestra en un repositorio central, y un programa cliente con el que cada usuario sincroniza su copia local. Esto permite compartir los cambios sobre un mismo conjunto de ficheros. Además, el repositorio guarda registro de los cambios realizados por cada usuario, y permite volver a un estado anterior en caso de necesidad.

Entre los sistemas de control de versiones más utilizados por los desarrolladores se encuentran *Subversion* y *CVS*. *Subversion* es empleado en la administración de archivos utilizados en el desarrollo de software o contenido. *CVS (Concurrent Versions System)*, considerado su antecesor, es uno de los controladores de versiones más utilizados en proyectos de software libre, sin embargo, a pesar de su amplio uso, el mismo diseño de *CVS* resultó ineficiente para diversos grupos de usuarios. Para el desarrollo de este sistema se decidió a utilizar *Subversion*.

Subversion presenta numerosas ventajas ante su antecesor *CVS*. Para eliminar y cambiar nombres de archivos, *CVS* requería intervención manual en el repositorio para lograrlo, *Subversion* contempló esta deficiencia y la corrigió con éxito. *Subversion* es capaz de mantener un control diferencial sobre

cualquier archivo binario del repositorio, reduciendo así el consumo de espacio, esto contrastado con CVS que requiere archivar copias completas de un archivo binario cada vez que éste cambia. Además maneja eficientemente archivos binarios, mientras que CVS los trata internamente como si fueran de texto. En *Subversion* las modificaciones del repositorio se realizan completas o no se realizan, o se realizan todos los pasos o no se realiza ninguno.

1.9 Conclusiones

En este capítulo se hizo un estudio sobre los sistemas que permiten la manipulación de los datos de los teléfonos móviles. Para el desarrollo de este sistema se decidió utilizar el lenguaje de programación Java por las ventajas y características que este lenguaje presenta y J2EE debido a que es la plataforma Java especialmente pensada para crear aplicaciones Web. El sistema se desarrollará con EJB3 y se utilizarán las páginas JSP para el desarrollo de la aplicación web. La metodología de desarrollo seleccionada es RUP, el servidor de aplicaciones a utilizar será Jboss y se utilizarán herramientas como Axis, UML, MYSQL, *Rational Rose*, *Eclipse* y el controlador de versiones *Subversion*.

CAPÍTULO II: CARACTERÍSTICAS DEL SISTEMA

2.1 Introducción.

En este capítulo se describen los objetivos estratégicos de la organización y procesos de negocio que los soportan. Se realizará un análisis crítico de cómo se ejecutan actualmente esos procesos, las causas que originan la situación problemática y las consecuencias. Se describen los procesos que serán objeto de automatización. Se presenta la propuesta de sistema, la especificación de los requerimientos funcionales y los no funcionales. Se desarrolla el modelo de dominio y se definen los casos de uso.

2.2 Problema y situación problemática

La empresa Procyon tiene entre sus objetivos estratégicos proveer soluciones y servicios en el mundo de las tecnologías de la información, que le permitan generar un alto valor y satisfacción a sus clientes. Desea distinguirse entre otros valores por la rapidez, la calidad, el sentido de innovación, la responsabilidad con que asumen sus compromisos y un personal altamente calificado. Desea crear espacios donde converjan proveedores de información y operadores de telecomunicaciones, posibilitando de esta forma que puedan compartir y satisfacer sus necesidades de negocio, información y comercialización.

Procyon desea prestar una especial atención al empleo de nuevas tecnologías de acceso a contenidos y nuevos modelos de dispositivos móviles así como nuevos formatos multimedia y tipos de contenido. Intenta mantener constantemente su infraestructura tecnológica actualizada con el fin de garantizar a sus clientes una capacidad inmediata de respuesta que satisfaga tanto distintos modelos de negocios, como las nuevas exigencias que puedan presentar operadores, clientes y proveedores de contenido.

Procyon Portal es una de las aplicaciones que desarrolló la empresa. Está diseñada para distribuir contenidos dirigidos a los teléfonos móviles, tales como ringtones monofónicos y polifónicos, imágenes, logos, temas y juegos.

Procyon Portal integra servicios como Procyon MPP (*Mobile Platform Participation*) y Procyon SSP (*Suscripcion Service Provider*) o sistemas de publicidad. Procyon MPP es una plataforma de participación que permite la interacción directa entre los medios informativos (TV, Radio, Prensa, Internet, etc) y sus clientes a través de SMS (*Short Message Service*), traducido al español como Sistema de Mensajes Cortos que es un sistema de mensajes de texto para teléfonos móviles de los operadores de telefonía móvil. Procyon SSP, es una plataforma que da la posibilidad al usuario de suscribirse y recibir en su móvil el pronóstico del tiempo, noticias o su horóscopo a través de un mensaje MMS (*Multimedia Messaging System*) traducido al español como Servicio de Mensajes Multimedia. MMS es un estándar de mensajería que le permite a los teléfonos móviles enviar y recibir contenidos multimedia, incorporando sonido, video, fotos, etc.

El pago se hace a través de SMS Premium, el usuario paga por enviar un mensaje SMS y accede a algún servicio ofrecido por quien le ha invitado a utilizarlo. Posteriormente el contenido es descargado directamente hacia el móvil a través de páginas WAP. WAP (*Wireless Application Protocol*) traducido al español como Protocolo de Aplicaciones Inalámbricas, es un estándar abierto internacional para aplicaciones que utilizan las comunicaciones inalámbricas, p.ej. acceso a servicios de Internet desde un teléfono móvil.

Otro proceso de negocio que soporta a la empresa Procyon es el desarrollo de juegos para móviles, entre estos se puede mencionar a *Holy Goalie*.

Procyon también está desarrollando la plataforma *Mobistore*. *Devices Manager* era uno de los proyectos de esta plataforma, el cual consistía en la confección de una base de datos de dispositivos móviles que se relacionaba con la misma. Este proyecto permitía la manipulación de la información de los dispositivos móviles. Tenía una aplicación que se comunicaba con la plataforma mediante servicios web. En la aplicación se actualizaban los datos desde los diferentes sitios web que ofrecen información de los dispositivos móviles y a su vez esta actualizaba los datos en la plataforma. Permitía adicionar, modificar y eliminar los datos de los dispositivos pero no contaba con una interfaz gráfica mediante la cual se entraran los datos a ser procesados por la aplicación.

La empresa Procyon se caracteriza por adoptar las nuevas tecnologías que van surgiendo y dejar atrás aquellas tecnologías que se convierten en obsoletas. La plataforma *Mobistore* y todos los

proyectos asociados a ella estaban implementados sobre una tecnología que va en desuso debido a las nuevas especificaciones que van surgiendo continuamente. Es por ello que se decidió migrar dichos productos a las nuevas tecnologías que se imponen, logrando así que la implementación de los mismos sea más fácil de comprender y mantener. Además la empresa decidió incorporar nuevas funcionalidades a la plataforma que le permitieran brindar un gran valor de satisfacción a sus clientes.

2.3 Objeto de automatización

Entre los sistemas automatizados que existen en la empresa y que están vinculados con el campo de acción del proyecto se encuentra la plataforma *Mobistore*. Esta plataforma tiene un servicio web que puede ser usado para construir aplicaciones o integrarse fácilmente con sistemas existentes como sistemas de mensajería que existen en un operador. Esta plataforma puede recibir un mensaje SMS ordenando la descarga de contenido móvil.

Los contenidos que son pedidos por los usuarios mediante la plataforma, ya sean juegos, temas, imágenes, animaciones, pistas de música, ringtones monofónicos y polifónicos, les son entregados a los usuarios mediante el envío de SMS, MMS, y WAP Pushes. Un WAP Push es un tipo de SMS que se utiliza para acceder de una manera conocida, sencilla y fácil a un sitio o página WAP sin necesidad de introducir la dirección (URL) de la página en el navegador de nuestro teléfono móvil.

Los contenidos como imágenes, animaciones y archivos de audio pueden ser adaptados para poder dar una respuesta adecuada a cada dispositivo móvil que realiza una petición a la plataforma.

Se desea automatizar un sistema que permita la correcta manipulación de los datos de los diversos dispositivos móviles, es decir que permita adicionar, buscar, actualizar y eliminar cualquier información referente a los dispositivos móviles en la base de datos del sistema. Dicha información puede ser datos de los dispositivos (como el nombre del modelo, nombre completo, imagen), características (tales como el ancho de la pantalla, cantidad de colores que soporta, tipos de imágenes que soporta) o los User Agent.

2.4 Propuesta del sistema

Se propone el desarrollo de un sistema multicapa que permita la correcta manipulación de datos de los diferentes dispositivos móviles. La capa de presentación se desarrollará sobre JSP, tecnología

Java que permite generar contenido dinámico para web y encapsula la lógica de navegación y el control de eventos de la interfaz.

La capa de la lógica del negocio se implementará a través del servicio web, que proporciona interoperabilidad al sistema. Desde el servicio web se administran todos los procesos, la adición, actualización, eliminación y la consulta de los datos de los dispositivos móviles.

La capa de acceso a datos se implementará sobre EJB3 que incluye el nuevo "API de Persistencia de Java". El gestor de base de datos utilizado es MySQL.

El sistema contará con una aplicación web como cliente del servicio web, que permite interactuar con el mismo y desde el cual se pueden ejecutar todas las acciones. La aplicación web se construirá con fines demostrativos, para visualizar las funcionalidades de la lógica del negocio. Se agregarán nuevas funcionalidades que antes no estaban y estarán publicadas en el servicio web, estas permitirán adicionar, buscar y eliminar determinada información en la base de datos de forma más agradable para el usuario.

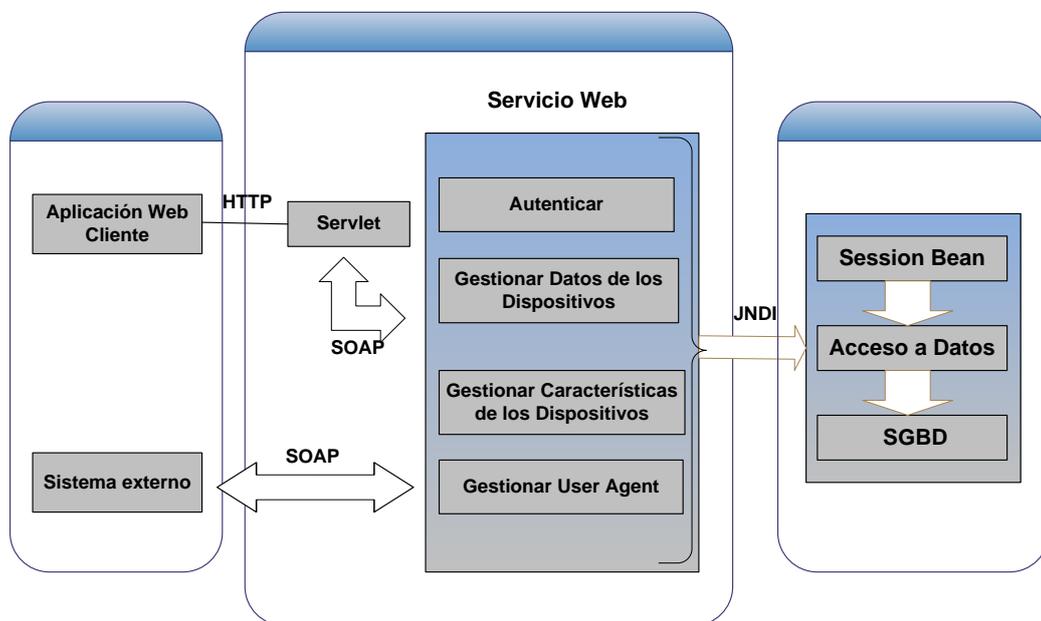


Figura 1. Propuesta del Sistema

Análisis comparativo

Anteriormente en Procyon se había desarrollado el proyecto *Devices Manager* que formaba parte de la plataforma *Mobistore*. La capa de presentación estaba implementada sobre JSP y *Servlets* (código en Java que se ejecuta en un servidor web). La capa de servicios se implementó mediante EJB2.1 y se utilizó MySQL como gestor de almacenamiento de datos.

La propuesta que ahora se está desarrollando está basada en una tecnología más novedosa que la empleada para el sistema anterior, se utilizó el framework EJB3. El sistema existente no contaba con una aplicación visual en la cual un usuario podría introducir, actualizar, eliminar y consultar la información contenida en la base de datos de dispositivos, en cambio esta versión si contempla una aplicación visual. En esta propuesta se agregaron nuevas funcionalidades que antes no existían.

2.5 Modelo de dominio

Un sistema, por pequeño que sea, generalmente es complicado. Por eso se necesita dividirlo en piezas si se pretende comprenderlo y gestionar su complejidad. Esas piezas se pueden representar a través de modelos que permitan abstraer sus características esenciales.

Los procesos del negocio no están bien definidos y las fronteras no están bien establecidas. Sin embargo se puede identificar personas, eventos, transacciones y objetos involucrados. Estos se relacionan en un diagrama que conforma el modelo de dominio.

El modelo de dominio puede ser modelado a través de un diagrama de clases de UML, y nos da una idea de manera general de los objetos que maneja una organización. Los modelos de dominio pueden mostrar: objetos del dominio o clases conceptuales y asociaciones entre estas clases. Una clase conceptual es una idea u objeto. La tarea central es identificar las clases conceptuales relacionadas con el escenario que se está diseñado.

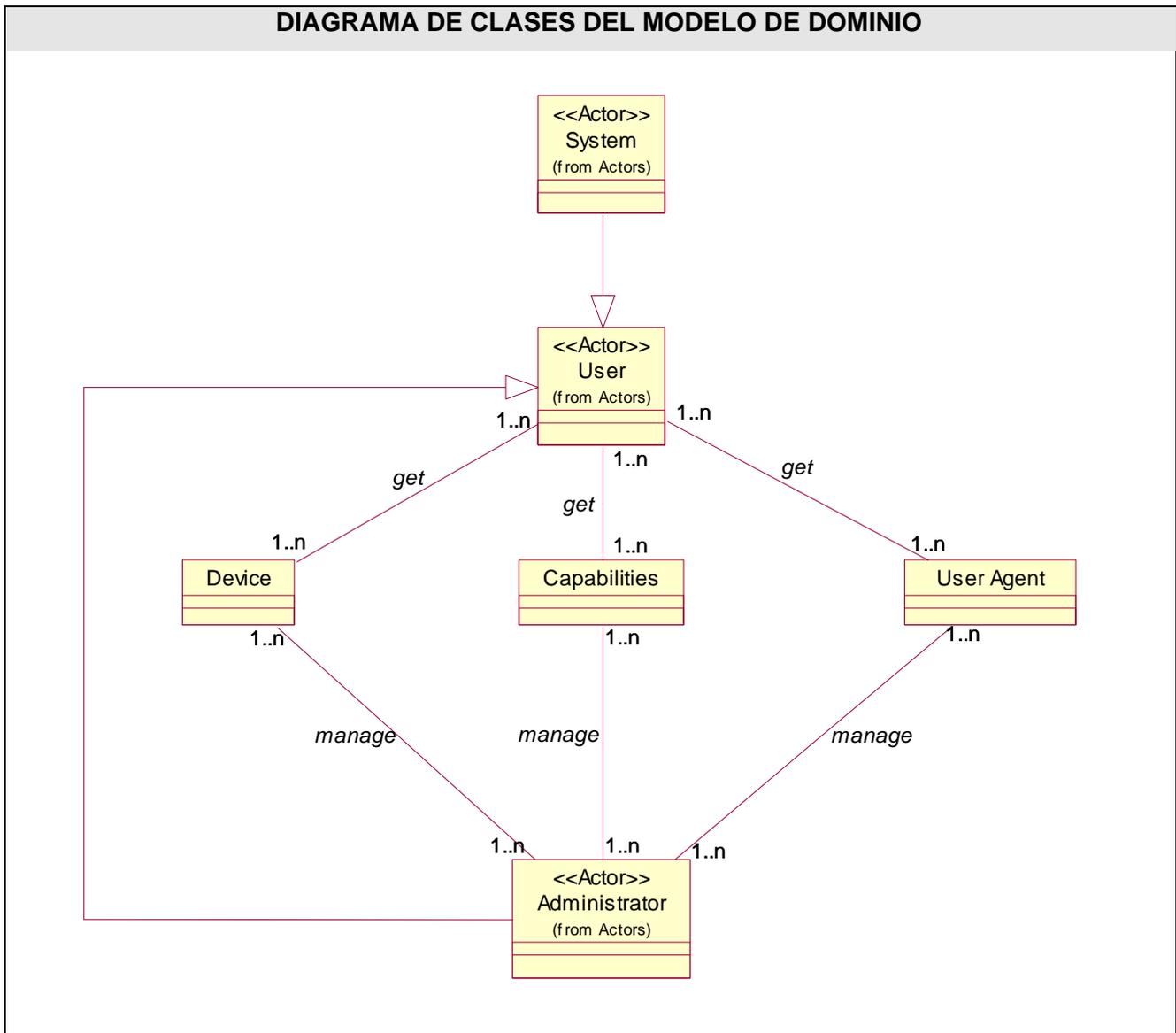


Figura 2. Modelo de Dominio

Descripción de los objetos del dominio

User: actor que lista todos los objetos.

System: actor que lista todos los objetos.

Administrator: actor que gestiona todos los objetos y realiza las acciones del usuario.

Device: objeto que representa los datos de los dispositivos móviles.

Capabilities: objeto que representa las características de los dispositivos móviles.

User Agent: objeto que representa a los User Agent de los dispositivos móviles.

2.6 Relación de los Requerimientos

Un requerimiento es una condición o capacidad que necesita un usuario para resolver un problema o lograr un objetivo. Es una condición o capacidad que tiene que ser alcanzada o poseída por un sistema o componente de un sistema para satisfacer un contrato, estándar, u otro documento impuesto formalmente. Todas las ideas que los clientes, usuarios y miembros del equipo de proyecto tengan acerca de lo que debe hacer el sistema, deben ser analizadas como candidatas a requisitos. Los requisitos se pueden clasificar en: funcionales y no funcionales.

2.6.1 Requerimientos funcionales

Los requerimientos funcionales son capacidades o condiciones que el sistema debe cumplir.

R1. Autenticar Usuario.

1.1 Permitir a los administradores autenticarse.

R2. Administrar datos de un dispositivo.

2.1 Permitir a los administradores adicionar un dispositivo.

2.2 Permitir a los administradores actualizar datos de un dispositivo.

2.3 Permitir a los administradores eliminar un dispositivo.

R3. Listar datos de un dispositivo.

3.1 Permitir a los usuarios obtener datos de un dispositivo.

R4. Administrar características de un dispositivo.

4.1 Permitir a los administradores adicionar características a un dispositivo.

4.2 Permitir a los administradores modificar características de un dispositivo.

R5. Listar características.

5.1 Permitir a los usuarios obtener las características de un dispositivo.

R6. Administrar User Agent.

6.1 Permitir a los administradores adicionar User Agent a un dispositivo.

6.2 Permitir a los administradores actualizar un User Agent de un dispositivo.

6.2 Permitir a los administradores eliminar User Agent de un dispositivo.

R7. Listar User Agent.

7.1 Permitir a los usuarios obtener los User Agent de un dispositivo.

2.6.2 Requerimientos no funcionales

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Son aquellas características que hacen al producto atractivo, usable, rápido o confiable. Normalmente están vinculados a requerimientos funcionales, una vez se conozca lo que el sistema debe hacer se puede determinar cómo ha de comportarse, qué cualidades debe tener o cuán rápido o grande debe ser.

Son importantes para que clientes y usuarios puedan valorar las características no funcionales del producto, pues si se conoce que el mismo cumple con las funcionalidades requeridas y las propiedades no funcionales en cuanto a cuan seguro, usable y agradable es, esto puede marcar la diferencia entre un producto bien aceptado y uno con poca aceptación.

- Apariencia o interfaz externa.

El sistema debe implementar una interfaz web con un diseño sencillo y comunicarse a través del protocolo de comunicación http. Las páginas no deben tener muchas imágenes.

- Usabilidad

La aplicación será fácil de manejar para los usuarios incluso aquellos que no han tenido mucha experiencia en el trabajo con computadoras o con sistemas informáticos.

- Rendimiento

El sistema debe ser eficiente, lograr un tiempo de respuesta mínimo, debido a que de por sí las aplicaciones web son desconectadas y tienen que acceder al servidor en busca de cualquier información.

- Soporte

Cada cierto tiempo la empresa debe brindar soporte al sistema, se debe realizar el mantenimiento del software. Se debe facilitar la posibilidad de actualización y cambios sobre la base de un diseño escalable y robusto.

- Seguridad

El sistema debe proteger la integridad de la información y los contenidos. El acceso a cualquier acción del sistema, debe estar sometido a un proceso de autenticación del usuario donde será especificado el rol, usuario y contraseña. Según el rol que presente el usuario será el nivel de acceso al sistema.

- Confiabilidad

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

2.7 Definición de los casos de uso

A continuación se presenta los actores del sistema, así como la descripción de los casos de uso y el diagrama de casos de uso del sistema.

2.7.1 Definición de los actores.

Tabla 1. Definición de Actores

Actores	Justificación
User	Representa la persona que puede visualizar información relativa a los dispositivos.
System	Representa a un sistema informático que interactúa con el negocio que puede visualizar información relativa a los dispositivos.
Administrator	Representa la persona que va a realizar todas las operaciones en el sistema.

2.7.2 Diagrama de casos de uso

En el siguiente diagrama de casos de uso se presenta las funciones del sistema y los actores que hacen uso de ellas.

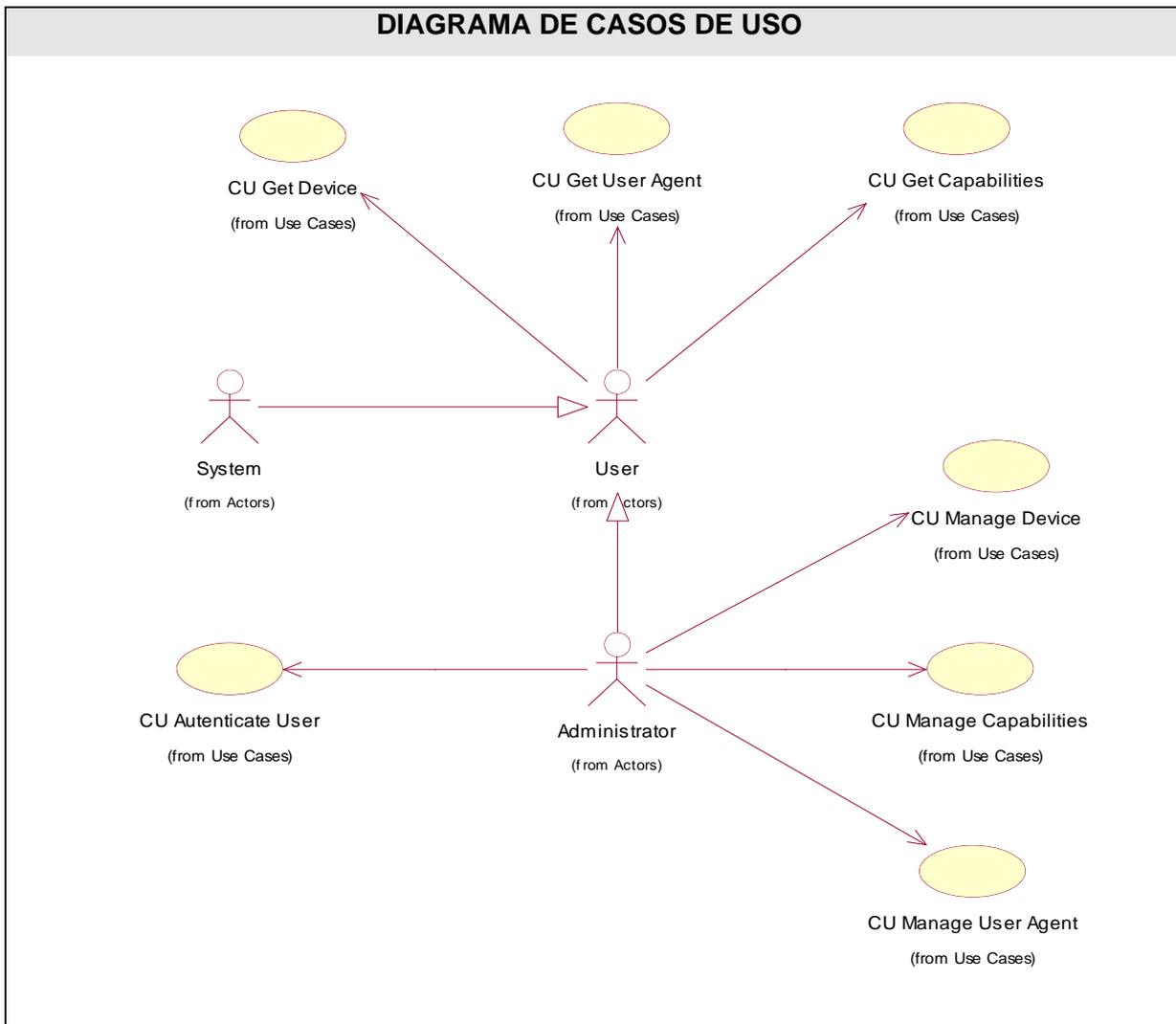


Figura 3. Diagrama de Casos de Uso

2.7.3 Casos de uso expandidos

La descripción detallada de los casos de uso ayuda a entender como debe funcionar el sistema. En esta descripción se especifica qué actor inicializa el caso de uso y las diferentes acciones que se llevan a cabo para la realización del mismo.

A continuación se puede ver la descripción detallada del caso de uso Manage Device, las descripciones de los demás casos de uso se pueden ver en el Anexo 1.

Tabla 2. Descripción Detallada del Caso de Uso Manage Device

Caso de uso	
CU-2	Manage Device
Propósito	Adicionar, actualizar o eliminar un dispositivo móvil.
Actores: Administrator	
Resumen: El caso de uso se inicia cuando el actor solicita adicionar, actualizar o eliminar un dispositivo. En el caso de adicionar llena los datos mostrados y se insertan los datos en la base de datos. En el caso de actualizar o eliminar busca el dispositivo según su identificador, su User Agent o su nombre completo; si desea actualizar modifica los datos que desea y actualiza los datos en la base de datos y si desea eliminar el sistema realiza esta acción.	
Referencias	R2
Precondiciones	Usuario previamente autenticado.
Curso Normal de los Eventos	
Acción del actor	Respuesta del sistema
1. El actor selecciona Administrar Dispositivo.	1.1 El sistema muestra tres opciones: Adicionar Dispositivo, Actualizar Dispositivo y Eliminar Dispositivo.
Escenario 1: Adicionar Dispositivo.	
1. El actor selecciona la opción Adicionar Dispositivo.	1.1 El sistema muestra el formulario a llenar.
2. El actor llena los datos del formulario.	2.1 El sistema verifica que los datos son correctos.
	2.2 Si los datos son correctos el sistema adiciona el dispositivo a la base de datos y le muestra un mensaje al usuario de que adicionó los datos satisfactoriamente.
Flujo Alternativo	
Acción 2.2:	Si los datos no son correctos el sistema le muestra un mensaje de error y solicita que los datos sean

	llenados nuevamente.
Escenario 2: Actualizar Datos del Dispositivo.	
1. El actor selecciona la opción Actualizar Dispositivo.	1.1 El sistema muestra las opciones para buscar el dispositivo: por el identificador, por el User Agent o por el nombre completo del dispositivo.
2. El actor selecciona el criterio de búsqueda para actualizar el dispositivo.	2.1 El sistema muestra los datos resultantes de la búsqueda.
3. El actor selecciona el dispositivo al cual quiere actualizarle la información.	3.1 El sistema muestra los datos del dispositivo seleccionado.
4. El actor modifica los datos que desee actualizar.	4.1 El sistema verifica que los datos son correctos.
	4.2 El sistema actualiza los datos y le muestra un mensaje al usuario de que actualizó los datos satisfactoriamente.
Flujo Alternativo	
Acción 4.2:	Si los datos no son correctos el sistema le muestra un mensaje de error y solicita que los datos sean llenados nuevamente.
Escenario 3: Eliminar Dispositivo.	
1. El actor selecciona la opción Eliminar Dispositivo.	1.1 El sistema muestra las opciones para buscar el dispositivo: por el identificador, por el User Agent o por el nombre completo del dispositivo.
2. El actor selecciona el criterio de búsqueda para eliminar el dispositivo.	2.1 El sistema muestra los datos resultantes de la búsqueda.
3. El actor selecciona el dispositivo a eliminar.	3.1 El sistema muestra un mensaje de confirmación.
4. El actor acepta eliminar el dispositivo.	4.1 El sistema elimina el dispositivo, actualiza la BD y le muestra un mensaje al usuario de que eliminó los datos satisfactoriamente.
Flujo Alternativo	
Acción 3.1:	El actor selecciona la opción cancelar y termina el

	caso de uso sin ejecutar ninguna acción.
--	--

2.8 Conclusiones

En este capítulo se analizaron los objetivos estratégicos de la organización y procesos de negocio que la soportan. Se hizo un análisis de cómo se ejecutan actualmente los procesos involucrados en el campo de acción. Se hizo una descripción de la propuesta del sistema. Se presentó el modelo de dominio. Se especificaron las condiciones y cualidades que el sistema debe tener en los requerimientos de software. Se especificaron los actores y casos de uso.

CAPÍTULO III: ANÁLISIS Y DISEÑO DEL SISTEMA

3.1 Introducción

En este capítulo se describen los procesos que se llevan a cabo en el flujo de trabajo de análisis y diseño. Se presentan los diagramas de clases del análisis y del diseño, así como los diagramas de interacción de ambos flujos y el Modelo de Datos. También se describe la arquitectura utilizada así como los patrones de diseño empleados.

3.2 Análisis

El modelo de análisis proporciona una vista interna del sistema. Se utiliza fundamentalmente para comprender cómo debería darse forma al sistema, es decir, cómo debería ser diseñado e implementado. También sirve como una primera aproximación del diseño. Define realizaciones de casos de uso en las cuales se describen cómo se lleva a cabo y se ejecuta un caso de uso determinado.

3.2.1 Diagrama de clases del Análisis

En un diagrama de clases del análisis se muestran las clases que están asociadas con el dominio del problema y las relaciones entre ellas. A continuación se muestran los diagramas de clases para cada caso de uso.

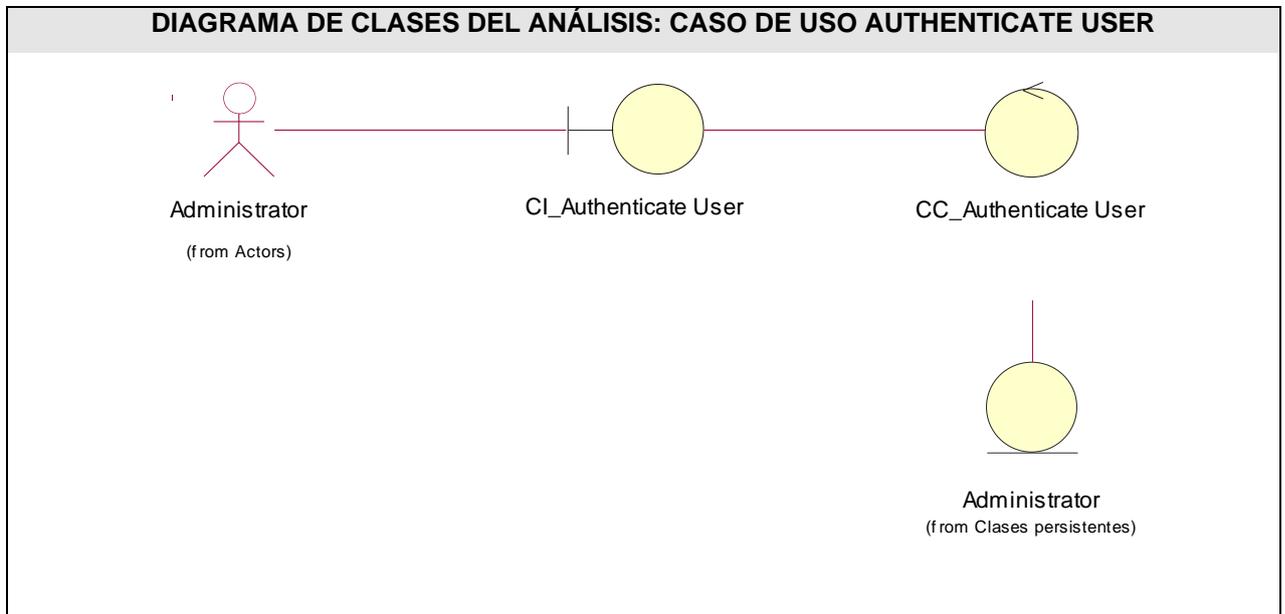


Figura 4. Diagrama de Clases del Análisis. CU Authenticate User

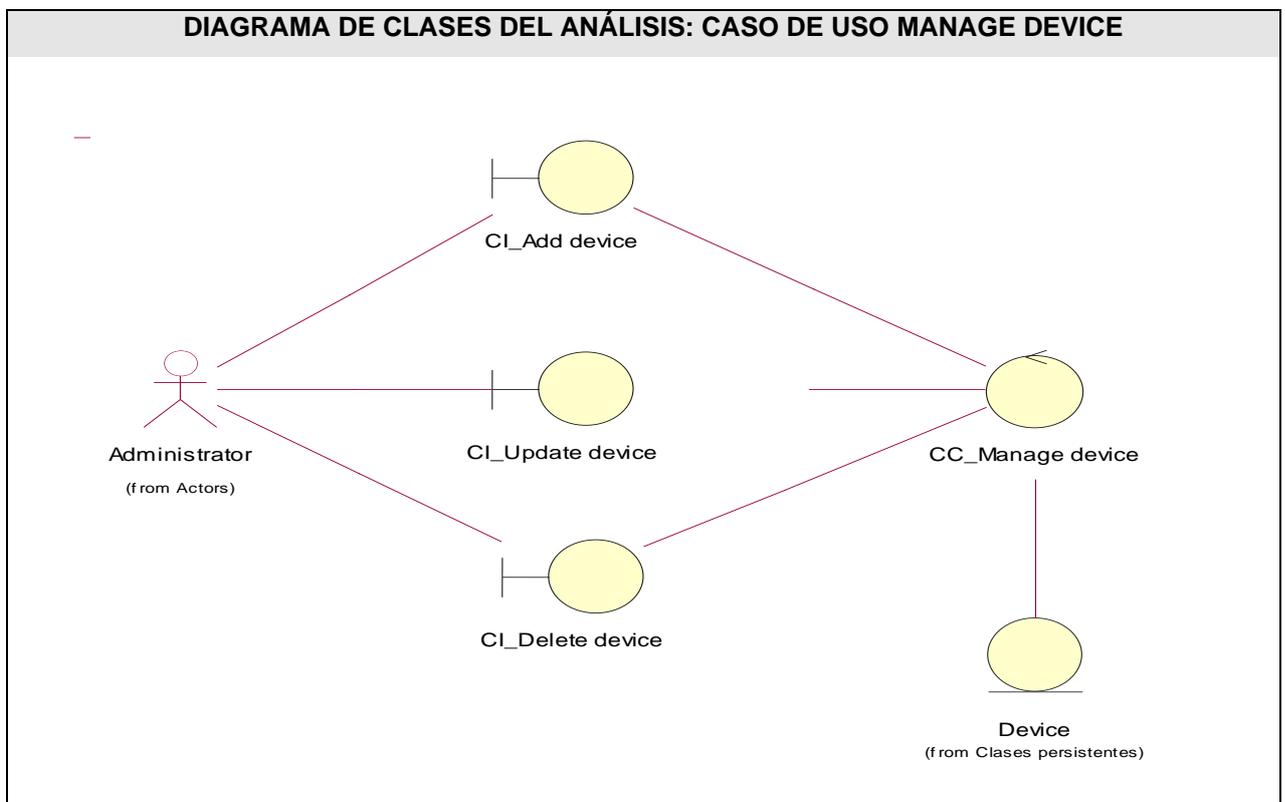


Figura 5. Diagrama de Clases del Análisis. CU Manage Device

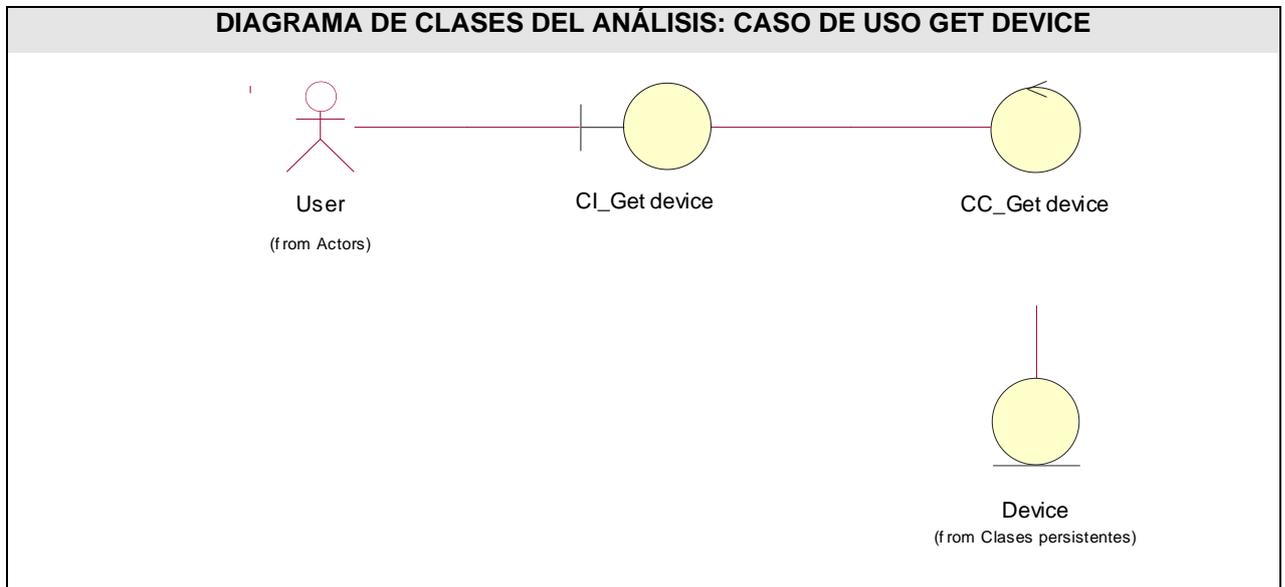


Figura 6. Diagrama de Clases del Análisis. CU Get Device

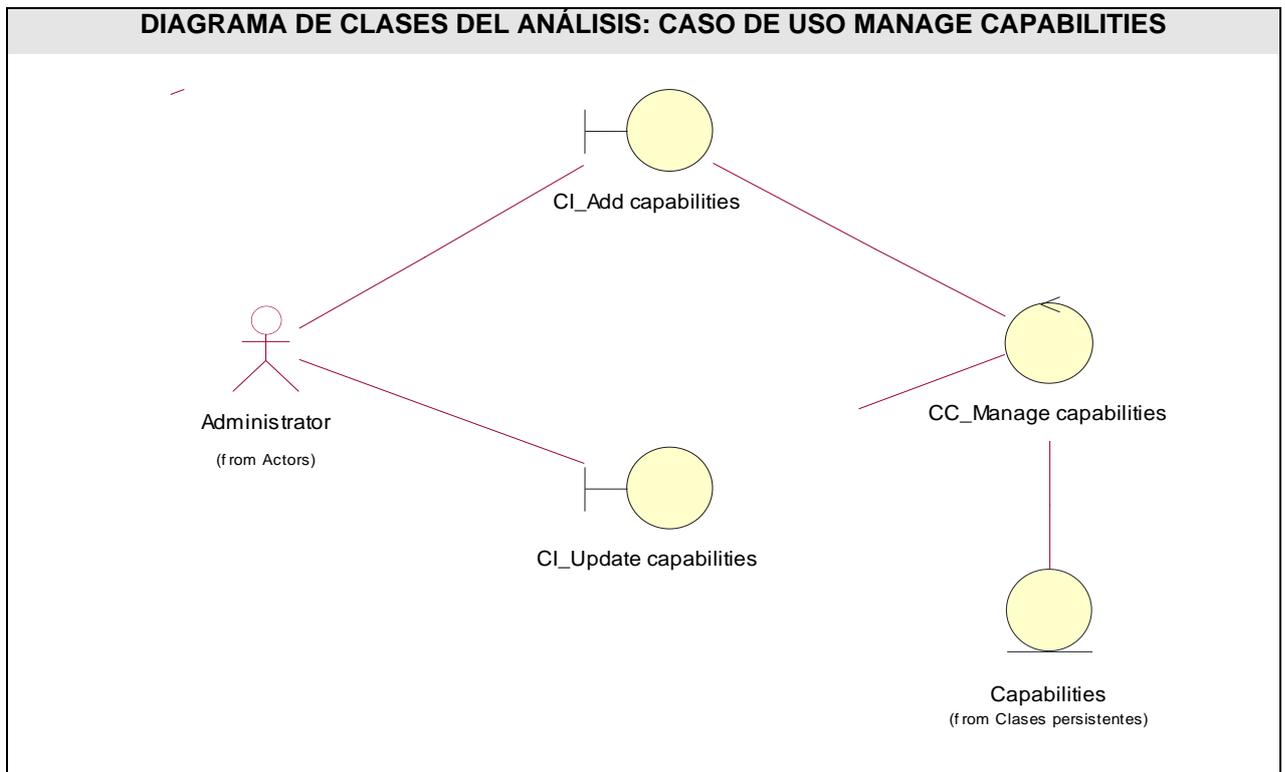


Figura 7. Diagrama de Clases del Análisis. CU Manage Capabilities

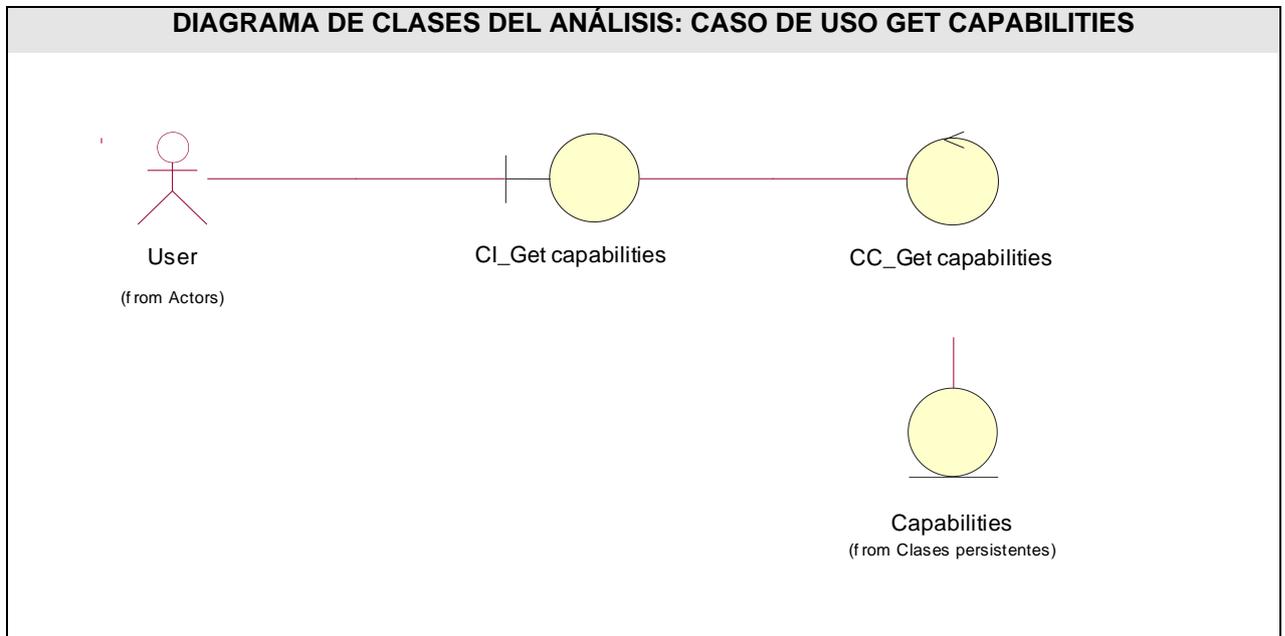


Figura 8. Diagrama de Clases del Análisis. CU Get Capabilities

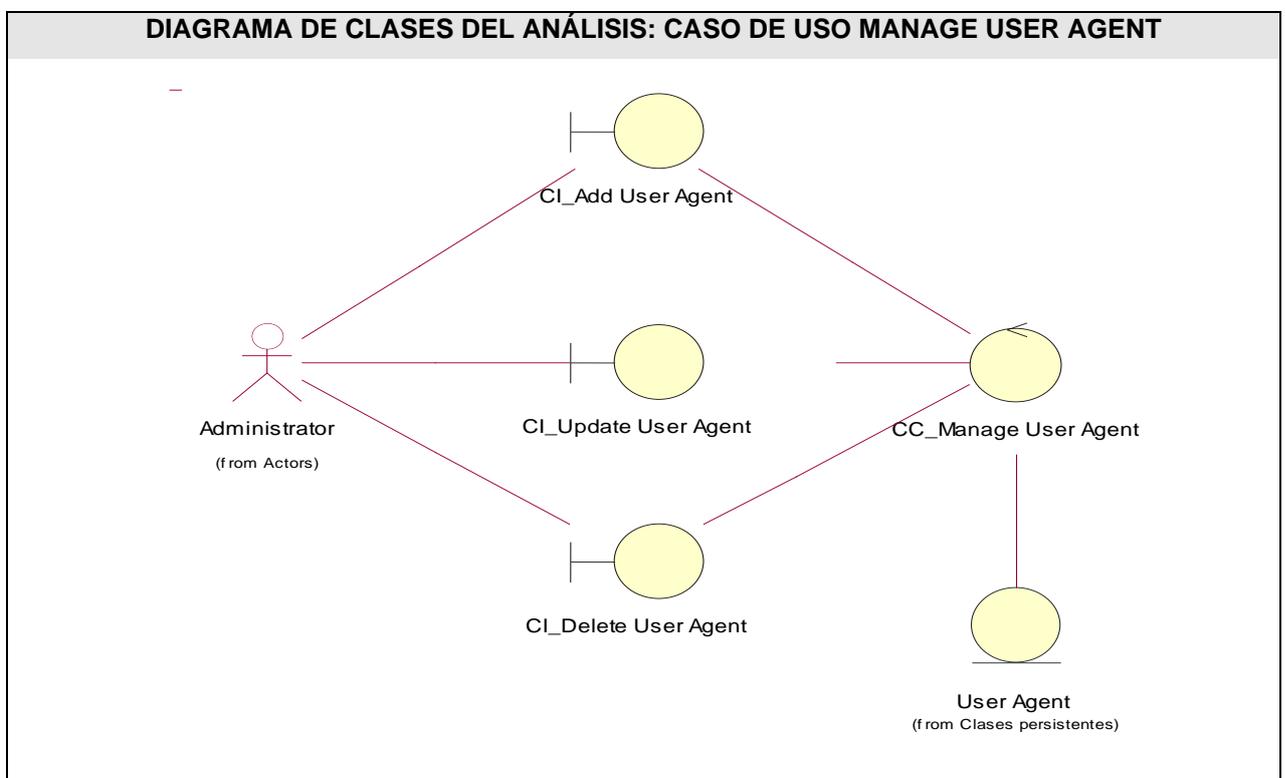


Figura 9. Diagrama de Clases del Análisis. CU Manage User Agent

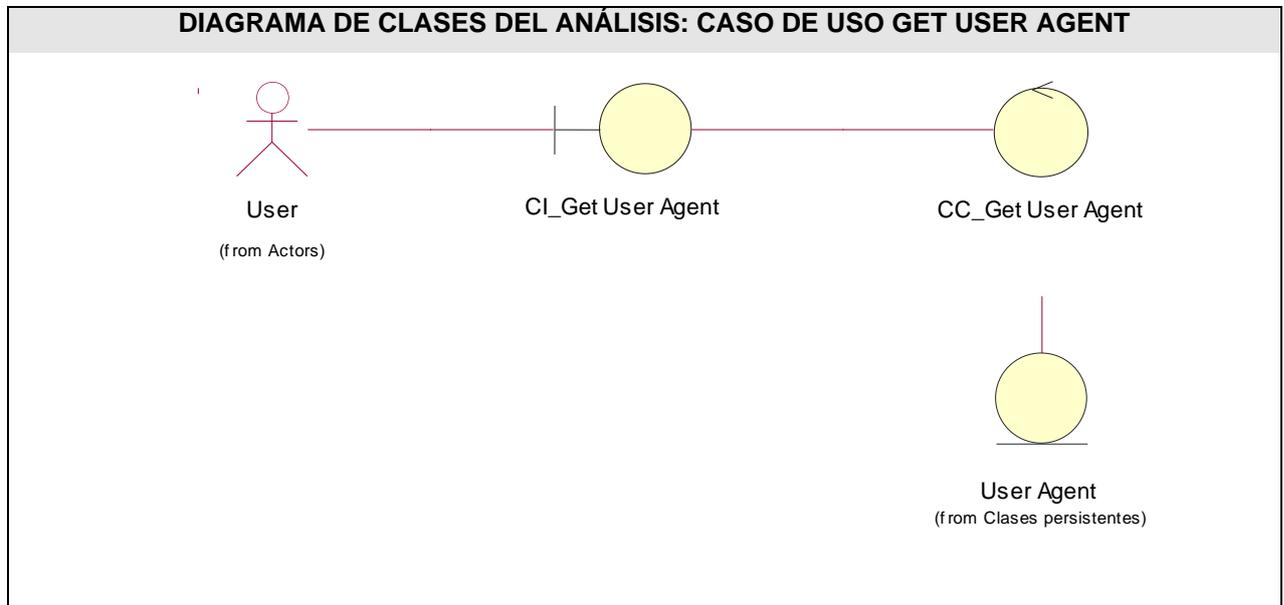


Figura 10. Diagrama de Clases del Análisis. CU Get User Agent

3.2.2 Diagramas de colaboración de los CU más significativos

Los diagramas de colaboración se utilizan para modelar los aspectos dinámicos de un sistema. Constan de un conjunto de objetos y sus relaciones, incluyendo los mensajes que se pueden enviar entre ellos. A continuación se muestra el diagrama de colaboración del caso de uso Manage Device y sus respectivos escenarios, los diagramas de colaboración de los demás casos de uso se pueden ver en el Anexo 2.

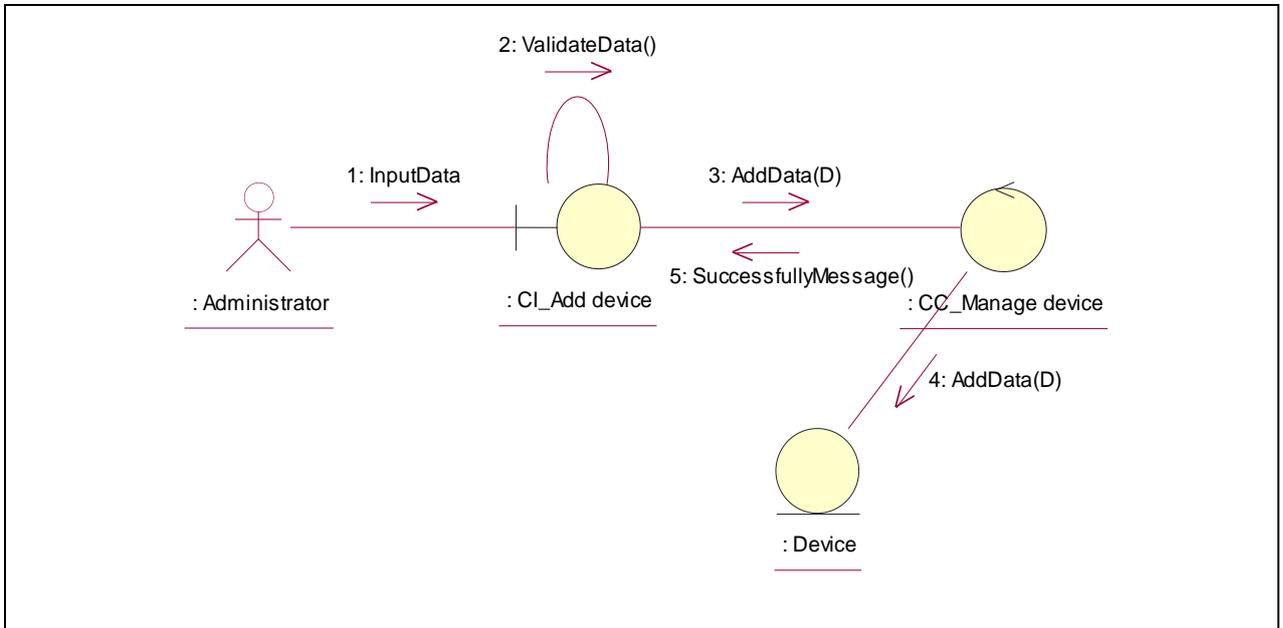


Figura 11. Diagrama de Colaboración del Análisis. CU Manage Device (escenario Add Device)

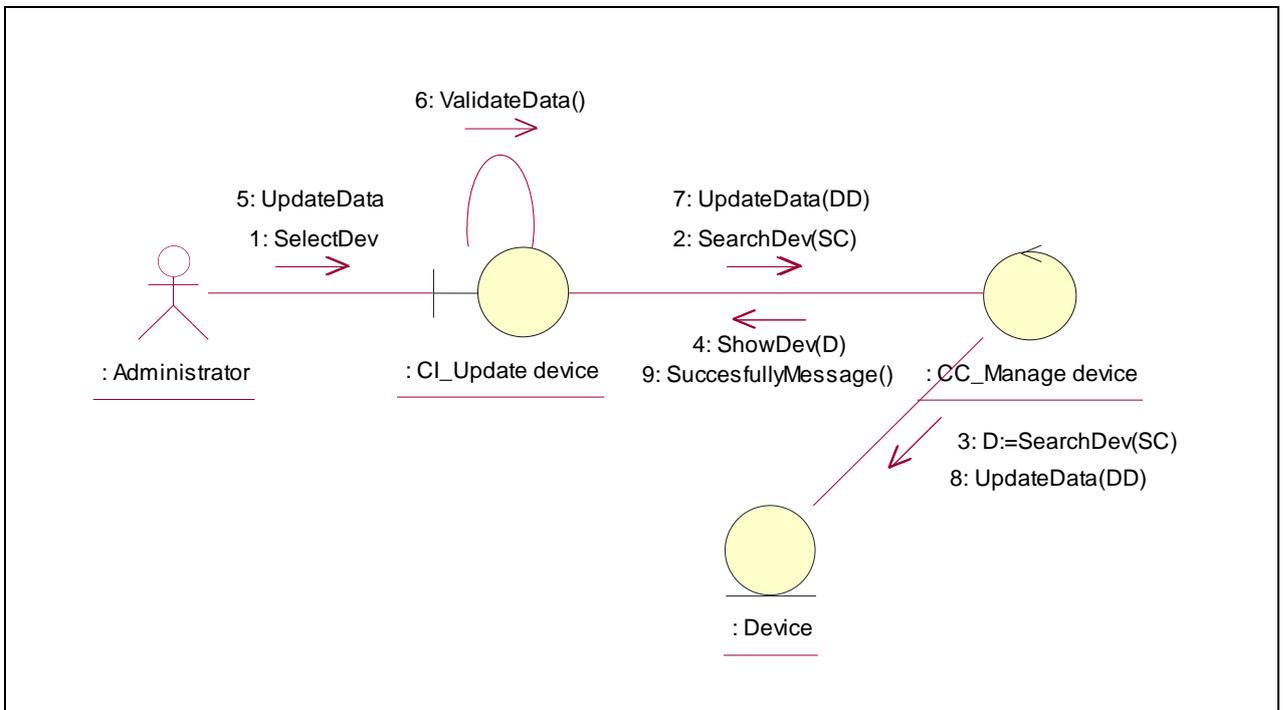


Figura 12. Diagrama de Colaboración del Análisis. CU Manage Device (escenario Update Device)

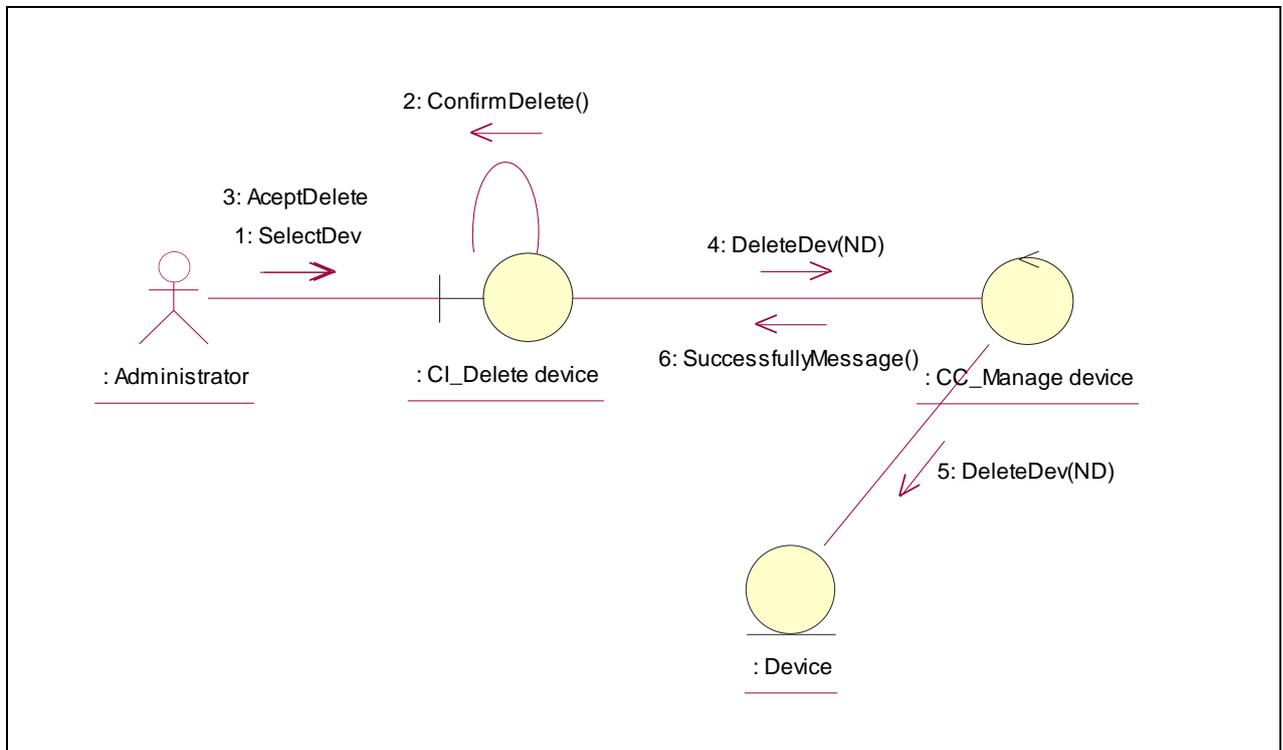


Figura 13. Diagrama de Colaboración del Análisis. CU Manage Device (escenario Delete Device)

3.3 Diseño

El modelo de diseño es un modelo de objetos que describe la realización de los casos de uso, y sirve como una abstracción del modelo de implementación y el código fuente. Se usa frecuentemente como una entrada inicial en las actividades de implementación y prueba. Está compuesto por clases del diseño, subsistemas, paquetes, colaboraciones, y las relaciones entre ellos.

3.3.1 Descripción de la Arquitectura utilizada

La arquitectura cliente/servidor es un modelo para el desarrollo de sistemas de información en el que las transacciones se dividen en procesos independientes que cooperan entre sí para intercambiar información, servicios o recursos. Se denomina cliente al proceso que inicia el diálogo o solicita los recursos y servidor al proceso que responde a las solicitudes. Separa los servicios situando cada uno en su plataforma más adecuada. Separa las funciones: Lógica de presentación, Lógica de negocio y Lógica de datos.

Para la realización de este sistema se ha utilizado esta arquitectura teniendo en cuenta los beneficios que la misma proporciona. Uno de estos beneficios es que el cliente no necesita conocer la lógica del servidor, sólo su interfaz externa; trabaja con distintos datos y aplicaciones, sin importarle dónde están o dónde se ejecutan cada una de ellas. El servidor presenta a todos sus clientes una interfaz única y bien definida. El cliente no depende de la ubicación física del servidor, ni del tipo de equipo físico en el que se encuentra, ni de su sistema operativo. Los cambios en el servidor implican pocos o ningún cambio en el cliente.

3.3.2 Patrones de Diseño

Los patrones de diseño representan soluciones a problemas que surgen cuando se desarrolla un software y facilitan el diseño de un software exitoso. Además capturan la estructura estática y dinámica y la colaboración entre los participantes claves del diseño del software.

Para el desarrollo de nuestro sistema se utilizaron varios patrones de diseño GRASP (*General Responsibility Assignment Software Patterns*), traducido al español como Patrones de Principios Generales para Asignar Responsabilidades:

- **Experto:** Para el desarrollo de un sistema se pueden definir docenas y hasta cientos de clases de software, y una aplicación tal vez requiera el cumplimiento de ciertas responsabilidades. Este patrón se usó para asignar responsabilidades a clases que cuentan con la información necesaria para cumplirlas. El uso de este patrón proporciona que el sistema sea más fácil de mantener y ampliar, y presenta la oportunidad de reutilizar los componentes en futuras aplicaciones.
- **Creador:** La creación de objetos es una de las actividades más frecuentes en un sistema orientado a objetos. Para el desarrollo de este sistema resulta muy conveniente el uso de este patrón ya que para crear una instancia de una clase lo tiene que hacer la clase que tiene la información para ello.
- **Bajo Acoplamiento:** Un bajo acoplamiento significa que una clase no depende de muchas clases. El uso de este patrón permite que los componentes sean fáciles de reutilizar debido a la poca dependencia con otras clases y que no se afecten por cambios de otros componentes.
- **Controlador:** Un evento del sistema es un evento de alto nivel generado por un actor externo; es un evento de entrada externa. Se asocia a operaciones del sistema: las que emite en respuesta a los eventos del sistema. Los controladores manejan esos eventos de entrada. Para

el desarrollo de este sistema se utilizó un controlador de fachada, es una clase que representa de alguna manera al sistema entero.

3.3.3 Diagrama de clases del Diseño

Los diagramas de clases del diseño son diagramas de estructura estática que muestran las clases del diseño y sus interrelaciones. A través de estos se muestra cómo el sistema puede ser construido. A continuación se muestran los diagramas de clases para cada caso de uso.

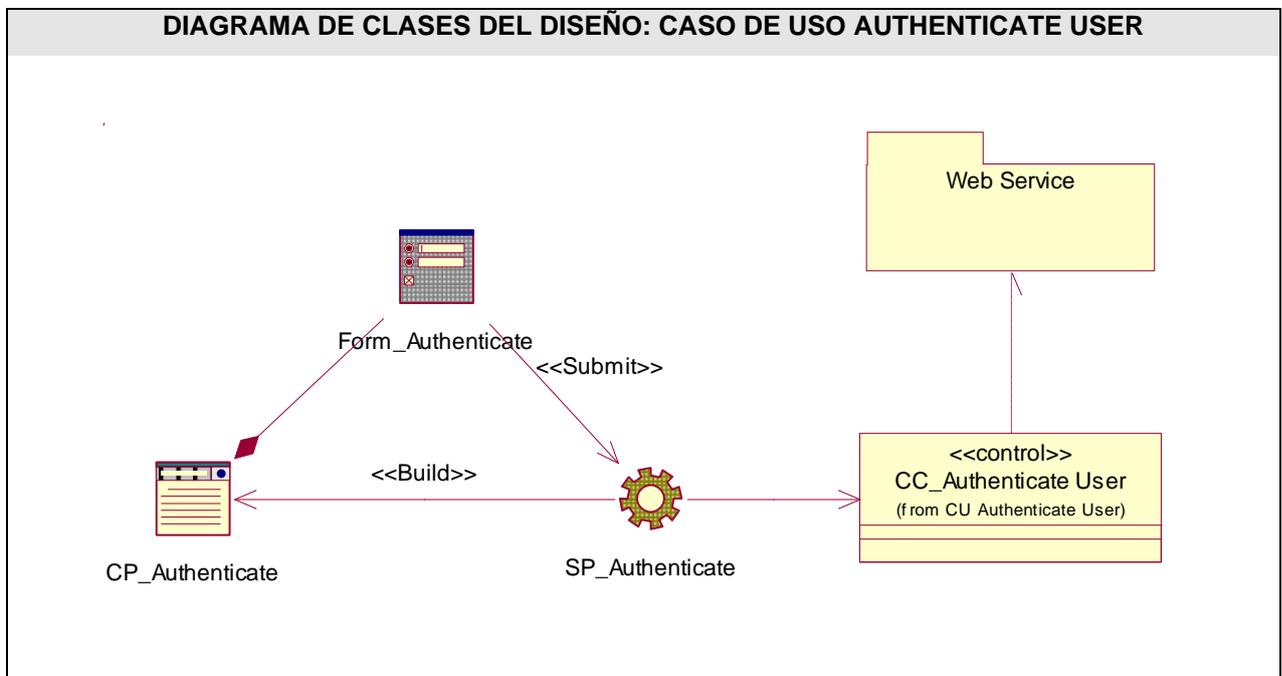


Figura 14. Diagrama de Clases del Diseño. CU Authenticate User

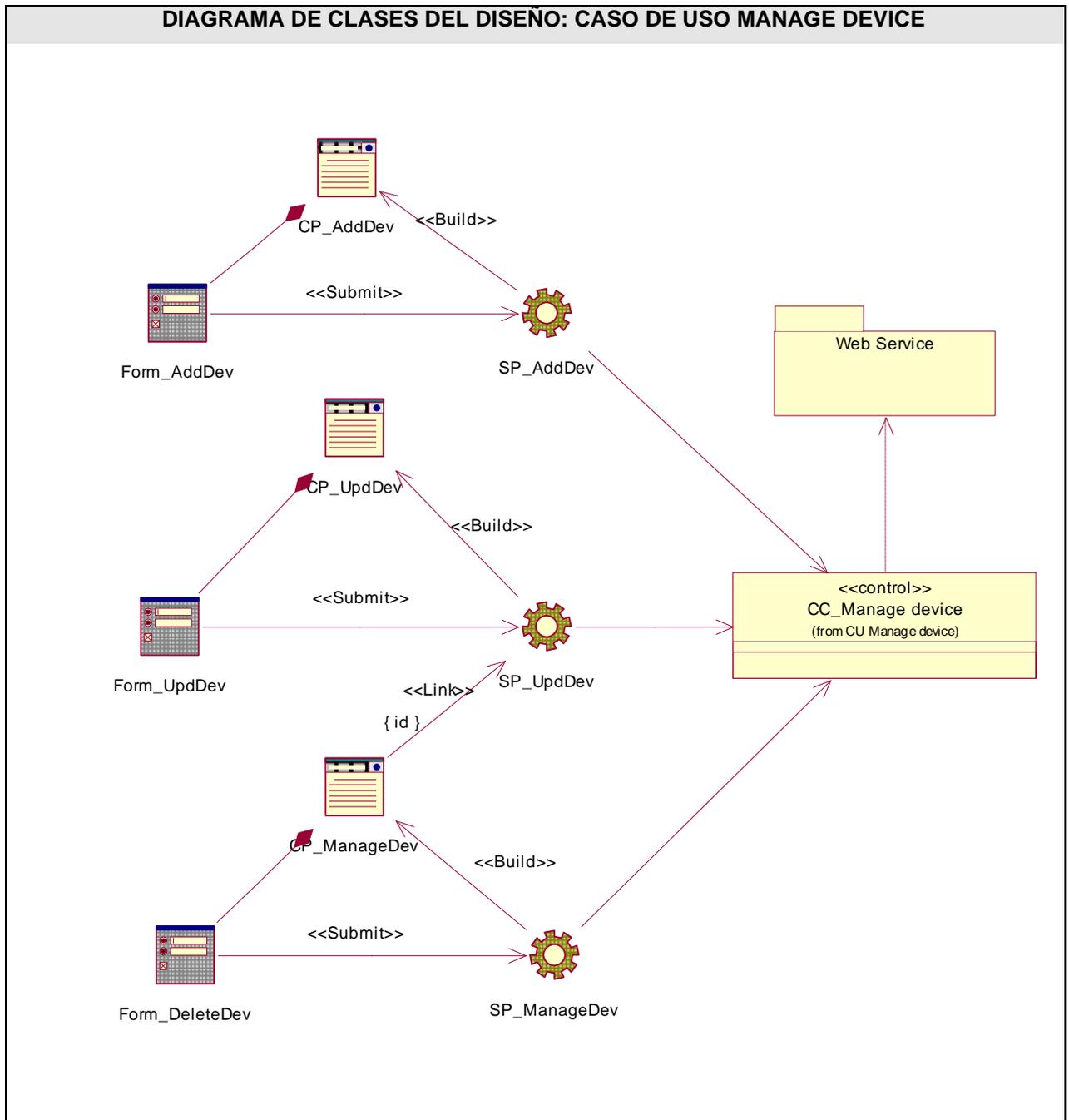


Figura 15. Diagrama de Clases del Diseño. CU Manage Device

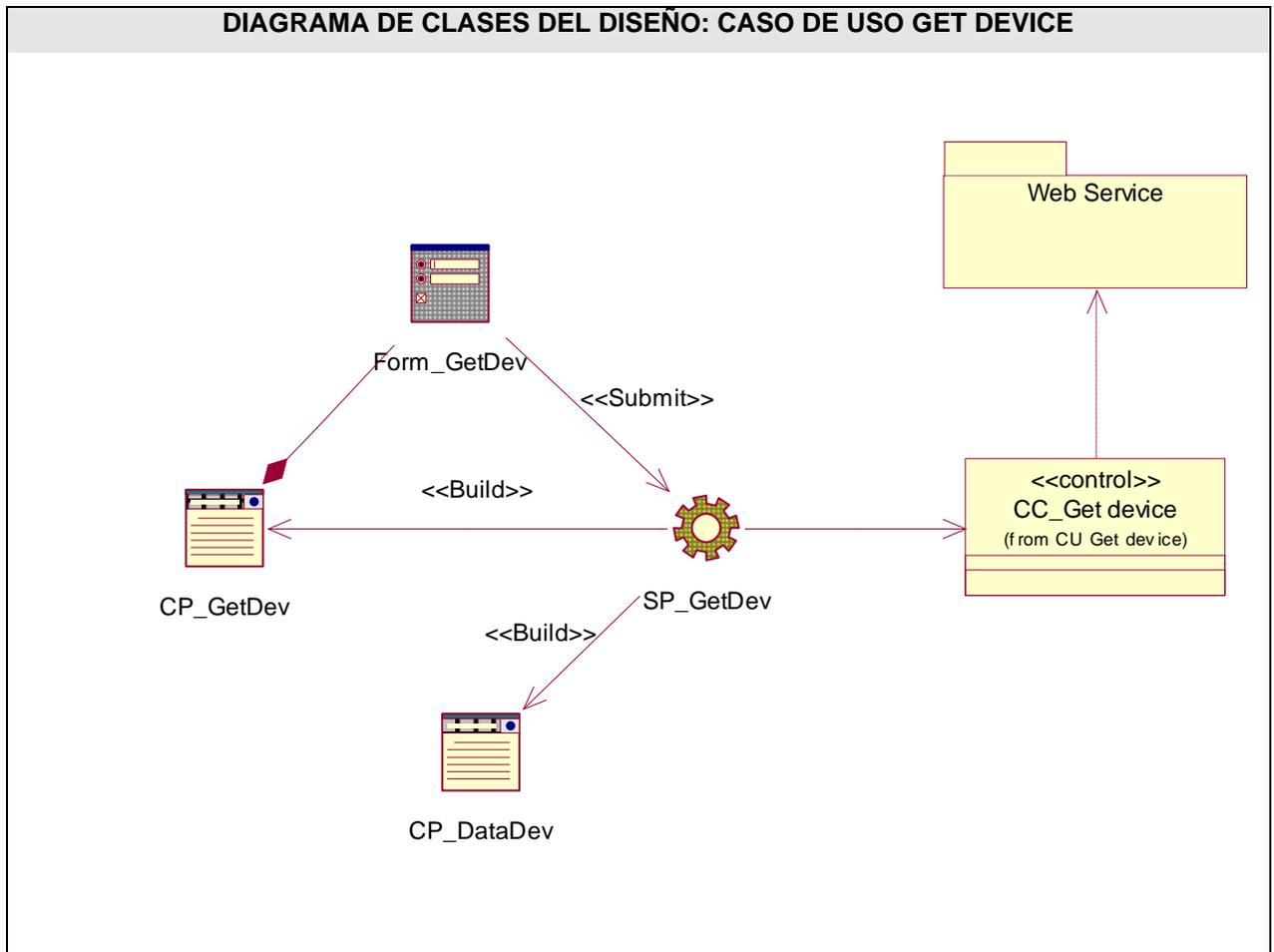


Figura 16. Diagrama de Clases del Diseño. CU Get Device

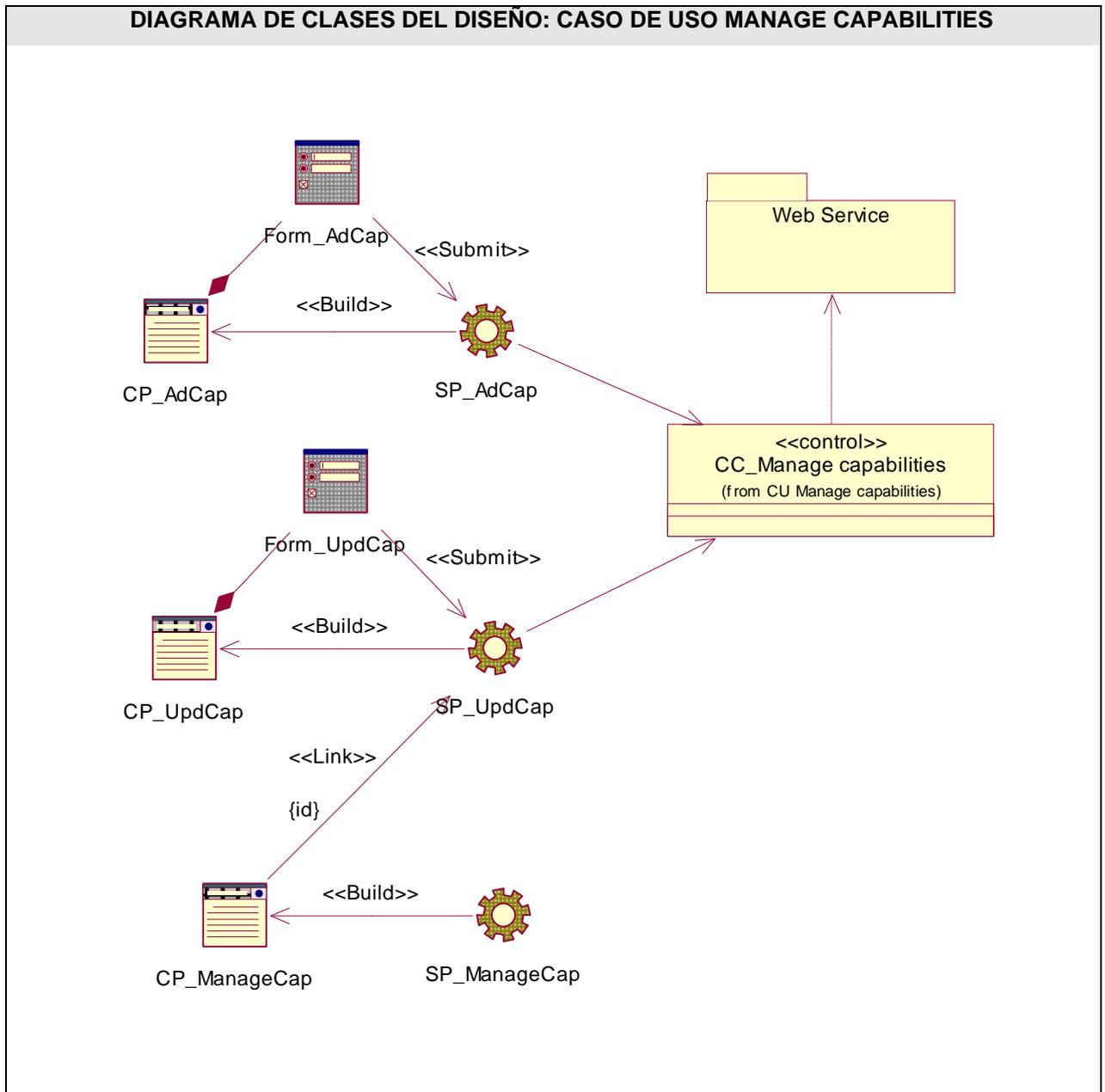


Figura 17. Diagrama de Clases del Diseño. CU Manage Capabilities

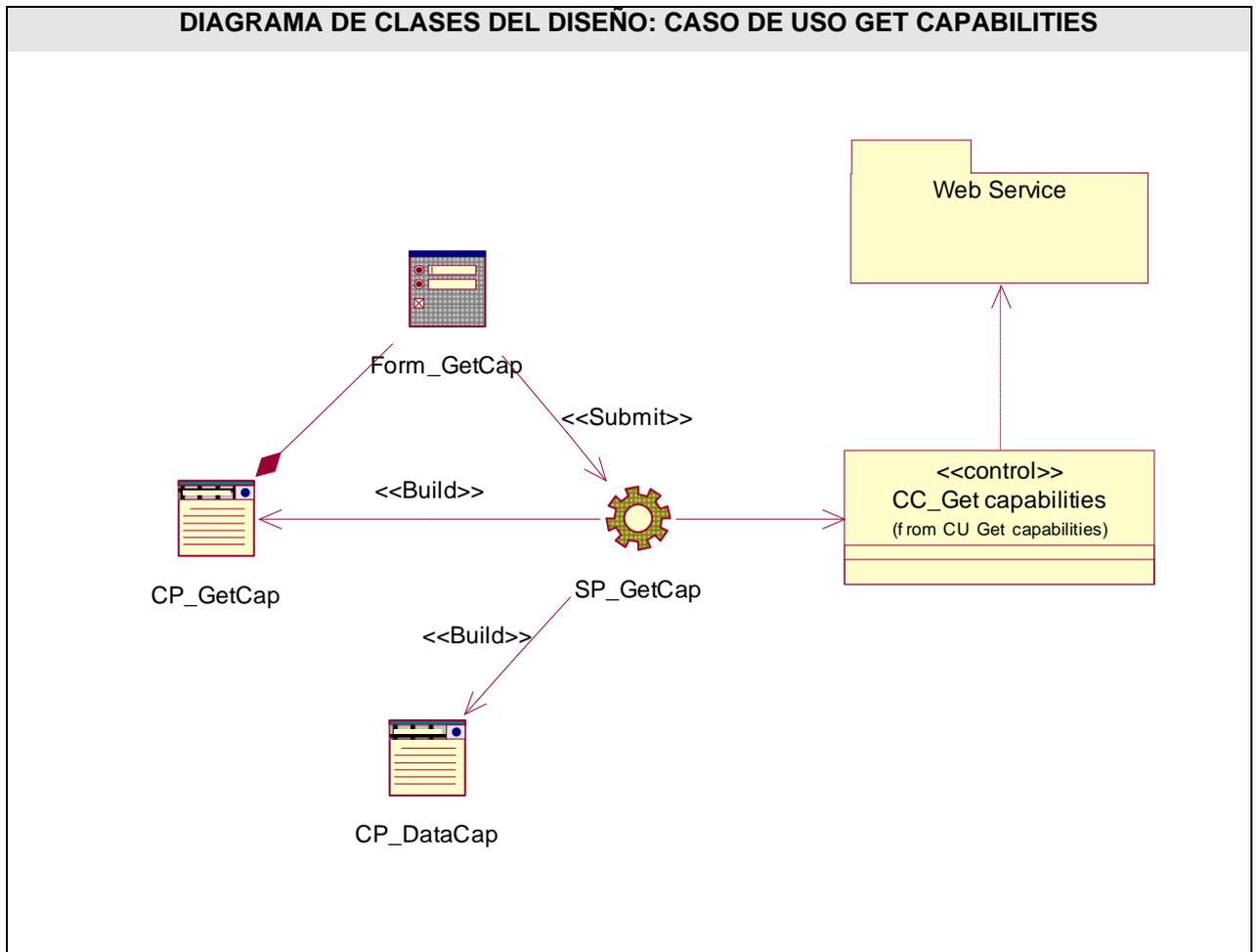


Figura 18. Diagrama de Clases del Diseño. CU Get Capabilities

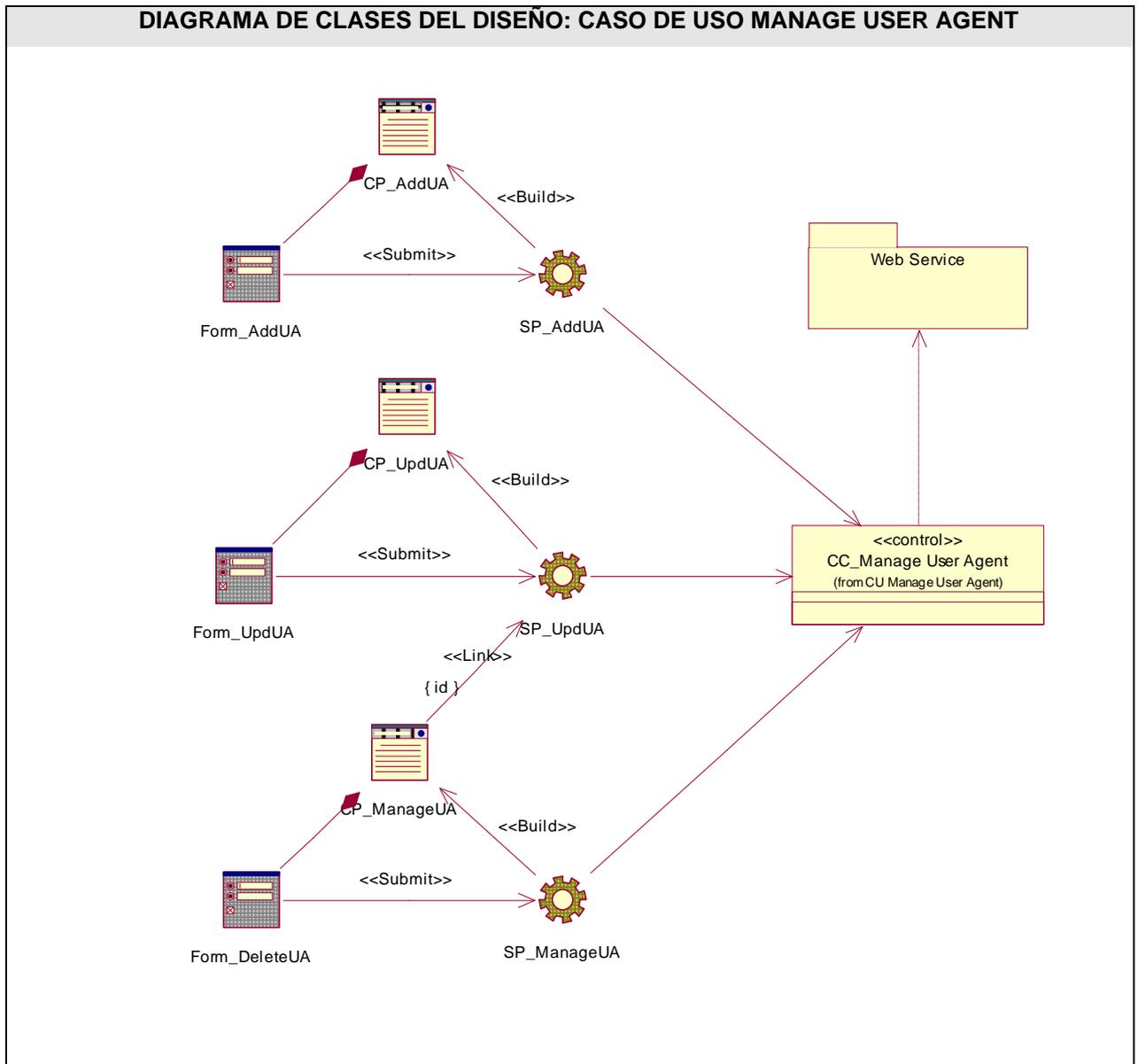


Figura 19. Diagrama de Clases del Diseño. CU Manage User Agent

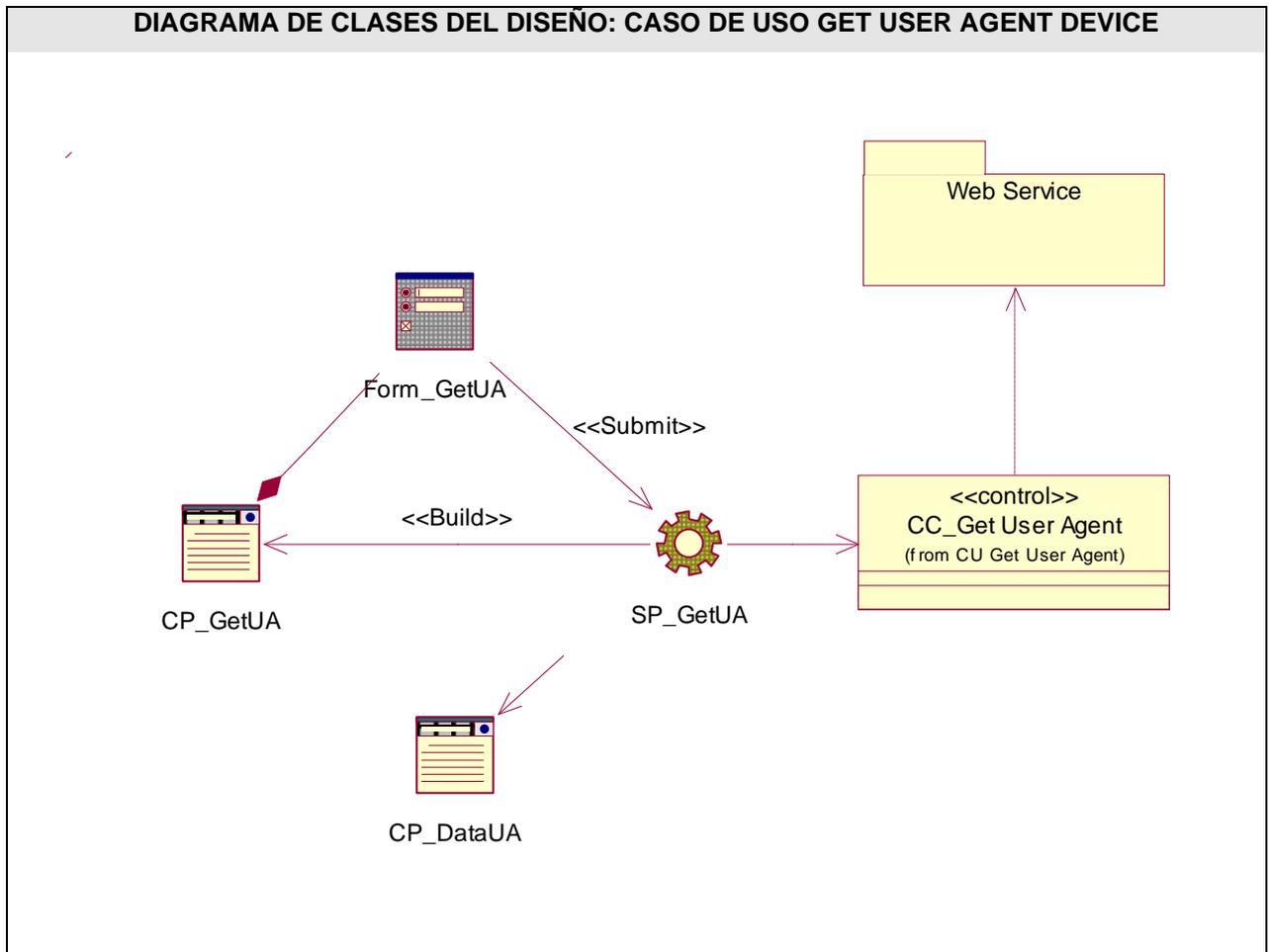


Figura 20. Diagrama de Clases del Diseño. CU Get User Agent

3.3.4 Diagramas de secuencia de los CU más significativos

Un diagrama de secuencia destaca la ordenación temporal de los mensajes. En este diagrama se muestran los objetos, sus relaciones y los mensajes enviados entre ellos. A continuación se muestra el diagrama de secuencia del caso de uso Manage Device y sus respectivos escenarios, los diagramas de secuencia de los demás casos de uso se pueden ver en el Anexo 3.

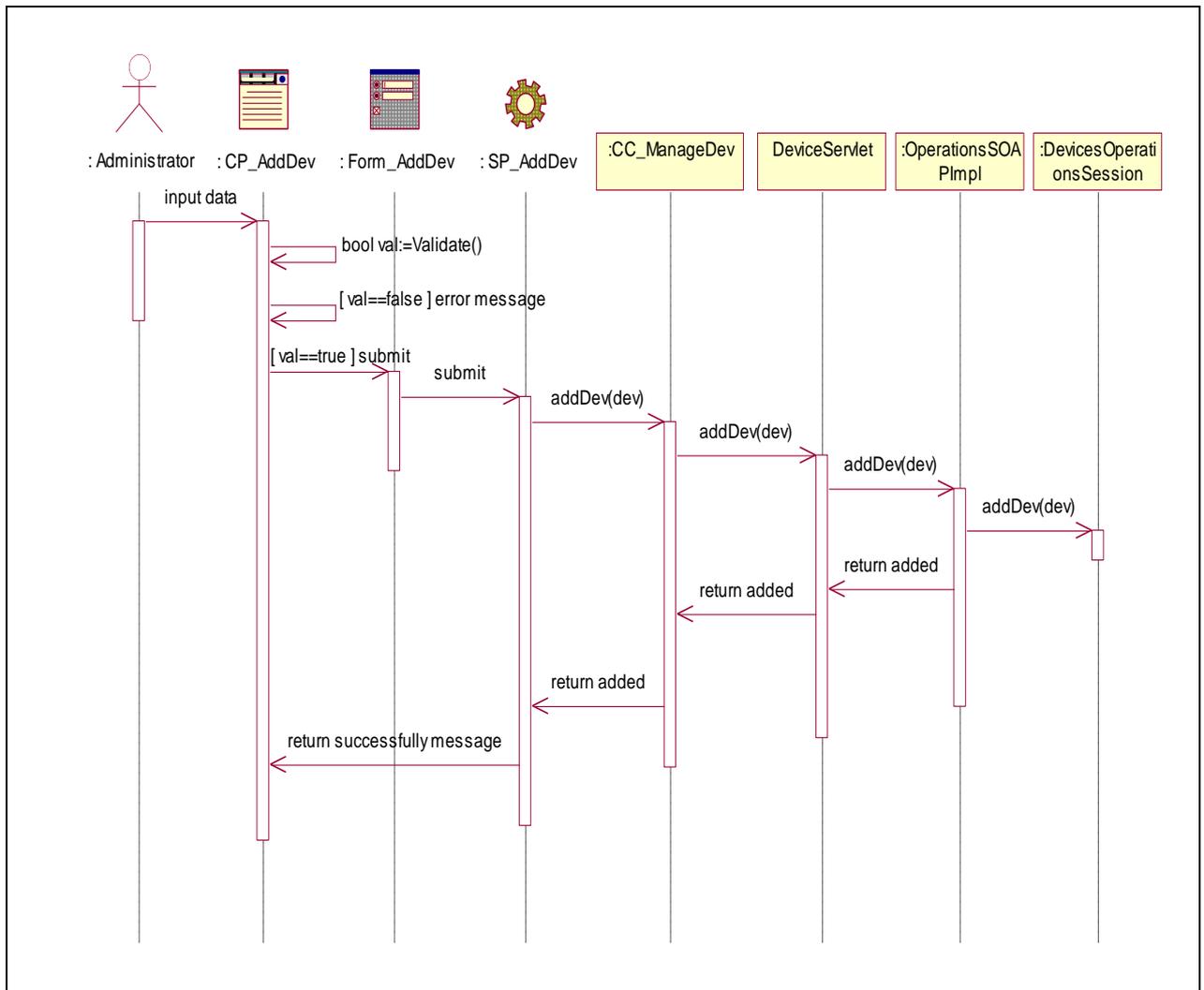


Figura 21. Diagrama de Secuencia del Diseño. CU Manage Device (escenario Add Device)

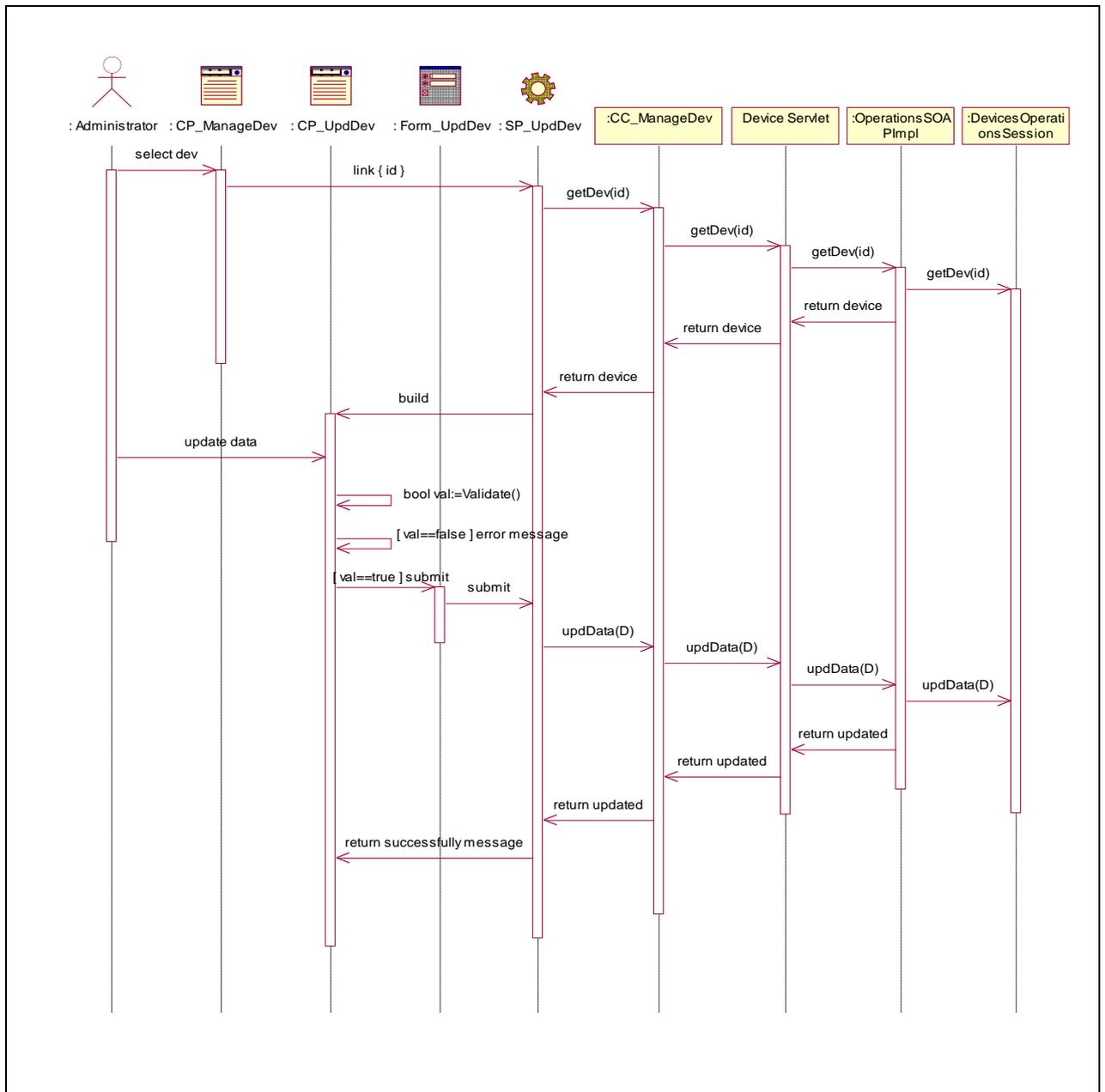


Figura 22. Diagrama de Secuencia del Diseño. CU Manage Device (escenario Update Device)

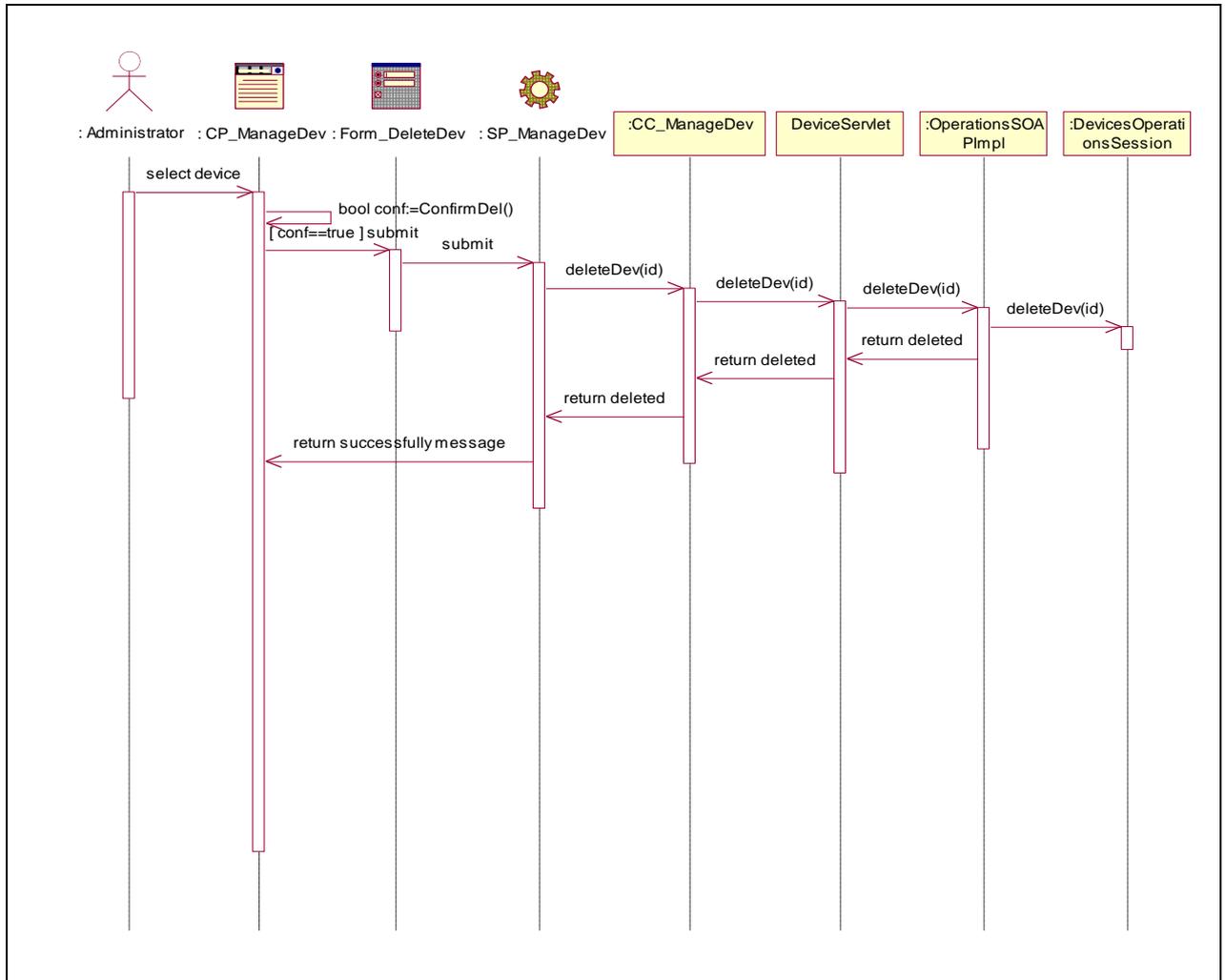


Figura 23. Diagrama de Secuencia del Diseño. CU Manage Device (escenario Delete Device)

3.4 Diseño de la base de datos

El diseño de la base de datos es una de las tareas más importantes en la construcción de un sistema que maneje una base de datos. Un buen diseño de la base de datos garantiza que el software sea eficiente garantizando un acceso eficaz a la información.

3.4.1 Modelo lógico de datos

El modelo lógico de datos representa la información que maneja el sistema. Es una fuente de información para el modelo físico. A continuación se muestra el Modelo lógico de datos que no es más que el diagrama de clases persistentes.

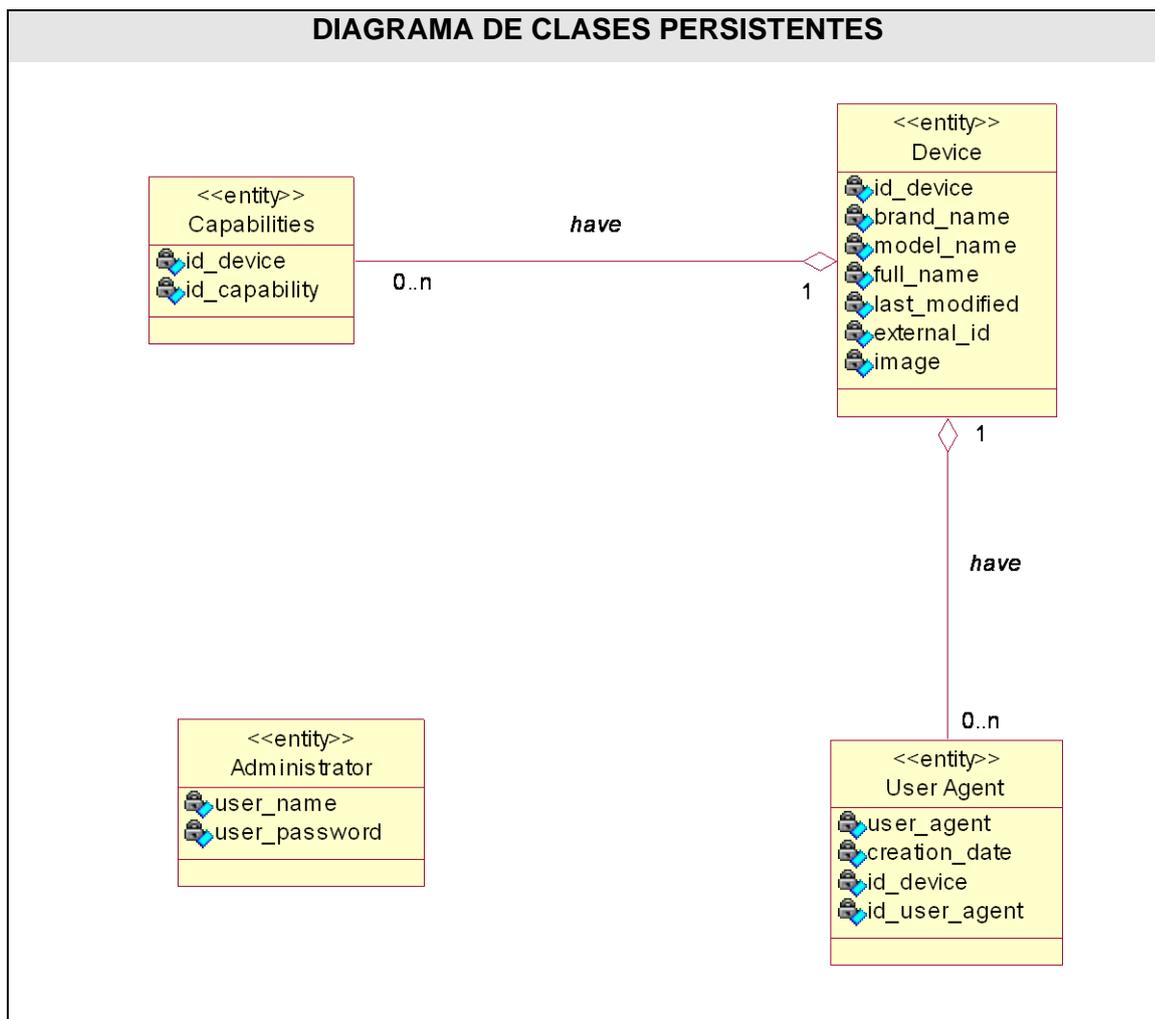


Figura 24. Diagrama de Clases Persistentes

3.4.2 Modelo físico de datos

El Modelo físico de datos es creado a partir de un conjunto de clases persistentes en el modelo de análisis. Es usado para definir el mapeo entre las clases persistentes del análisis y las estructuras de datos persistentes, y para definir las estructuras de datos persistentes mismas.

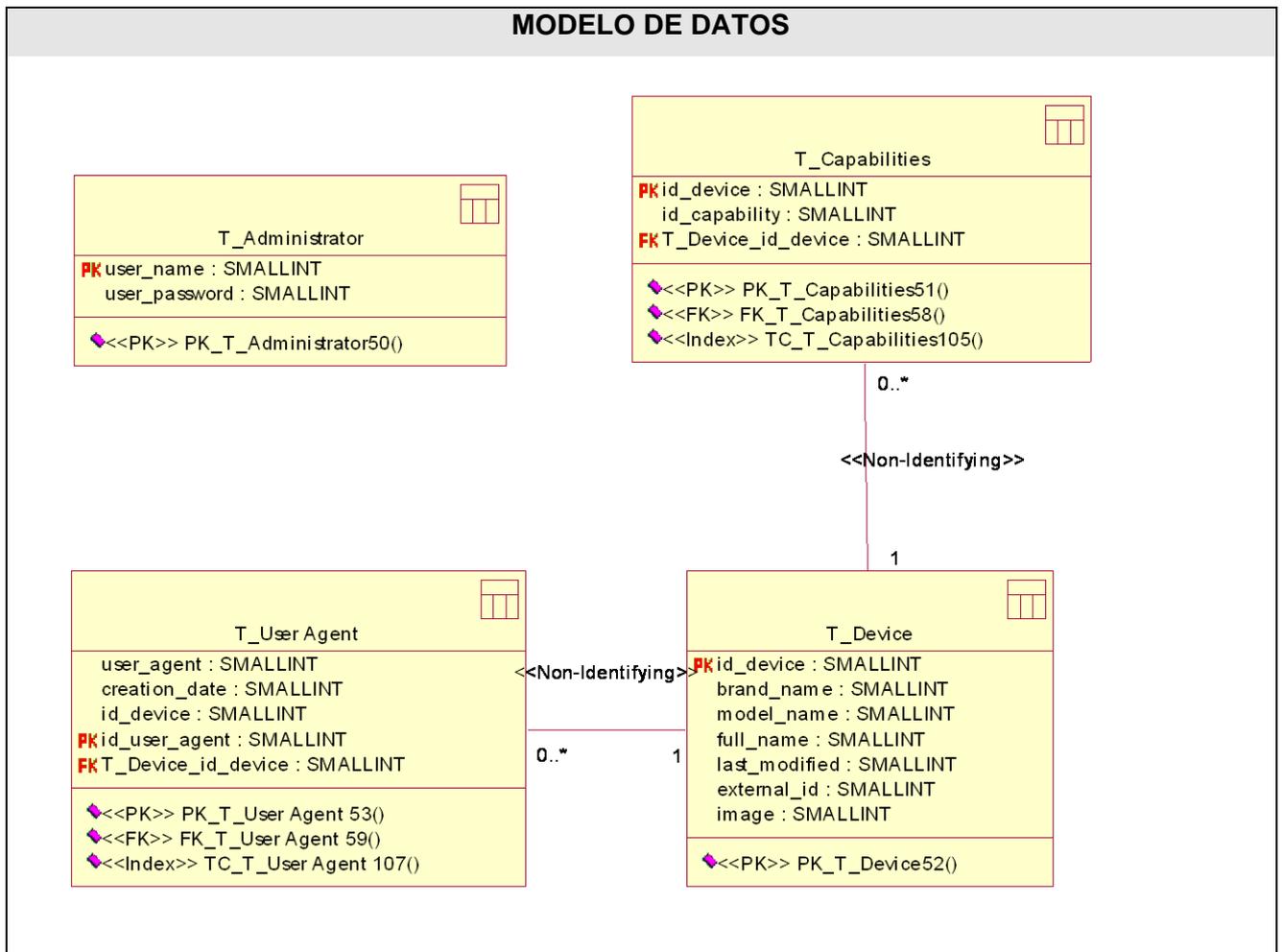


Figura 25. Modelo de Datos

3.5 Conclusiones

En este capítulo se mostraron los diagramas de clases de análisis y diseño, los cuales permitieron obtener una visión del sistema basada en la comprensión de los requisitos funcionales y no funcionales especificados en el capítulo anterior. Se mostraron los diagramas de colaboración del análisis y los diagramas de secuencia del diseño, los cuales permitieron entender el flujo de operaciones que se lleva a cabo en cada acción solicitada. Se describió la arquitectura, así como los patrones de diseño utilizados y se mostró el diseño de la base de datos.

CAPÍTULO IV: IMPLEMENTACIÓN

4.1 Introducción

En este capítulo se mostrará la distribución del sistema en nodos mediante el diagrama de despliegue. Además se mostrará la organización de los componentes del sistema y las relaciones lógicas entre ellos a través del diagrama de componentes.

4.2 Diagrama de Despliegue

Un diagrama de despliegue es un grafo de nodos unidos por conexiones de comunicación que describe la distribución, entrega e instalación de las partes que configuran el sistema físico. Muestra las relaciones físicas de los nodos que participan en la ejecución del sistema describiendo la arquitectura física del sistema en términos de: procesadores, dispositivos y componentes de software.

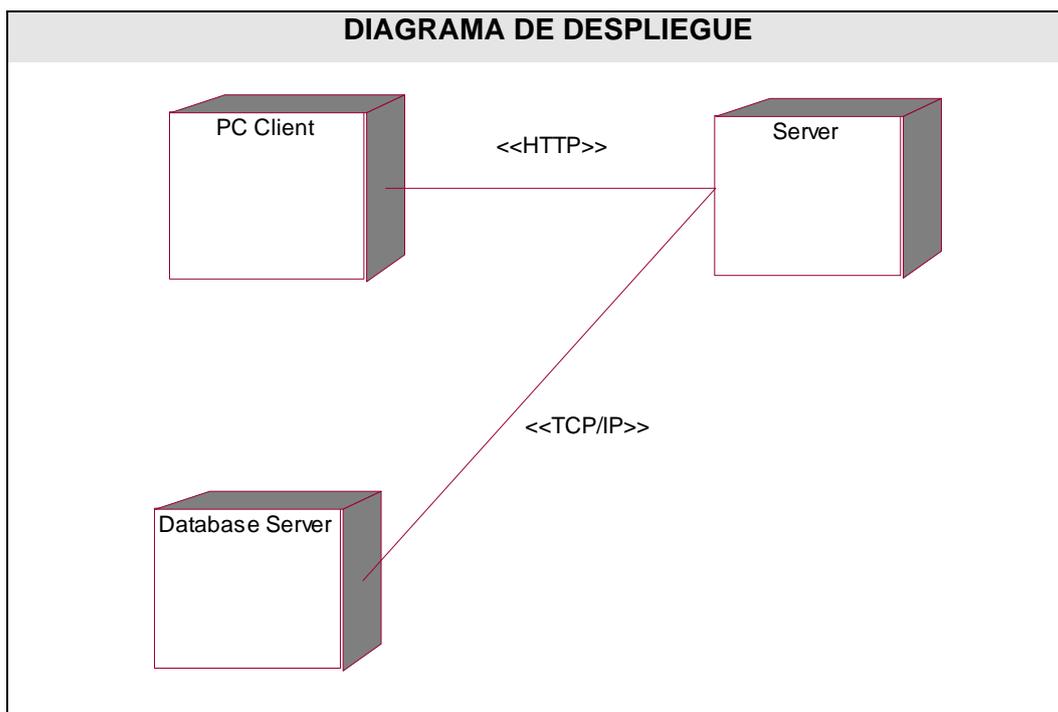


Figura 26. Diagrama de Despliegue

4.3 Diagrama de Componentes

Los diagramas de componentes se utilizan para modelar la vista de implementación estática de un sistema. Muestran tanto los componentes de software (código fuente, binario y ejecutable) como las relaciones lógicas entre ellos en un sistema. Y como todos los diagramas, también puede contener paquetes utilizados para agrupar elementos del modelo.

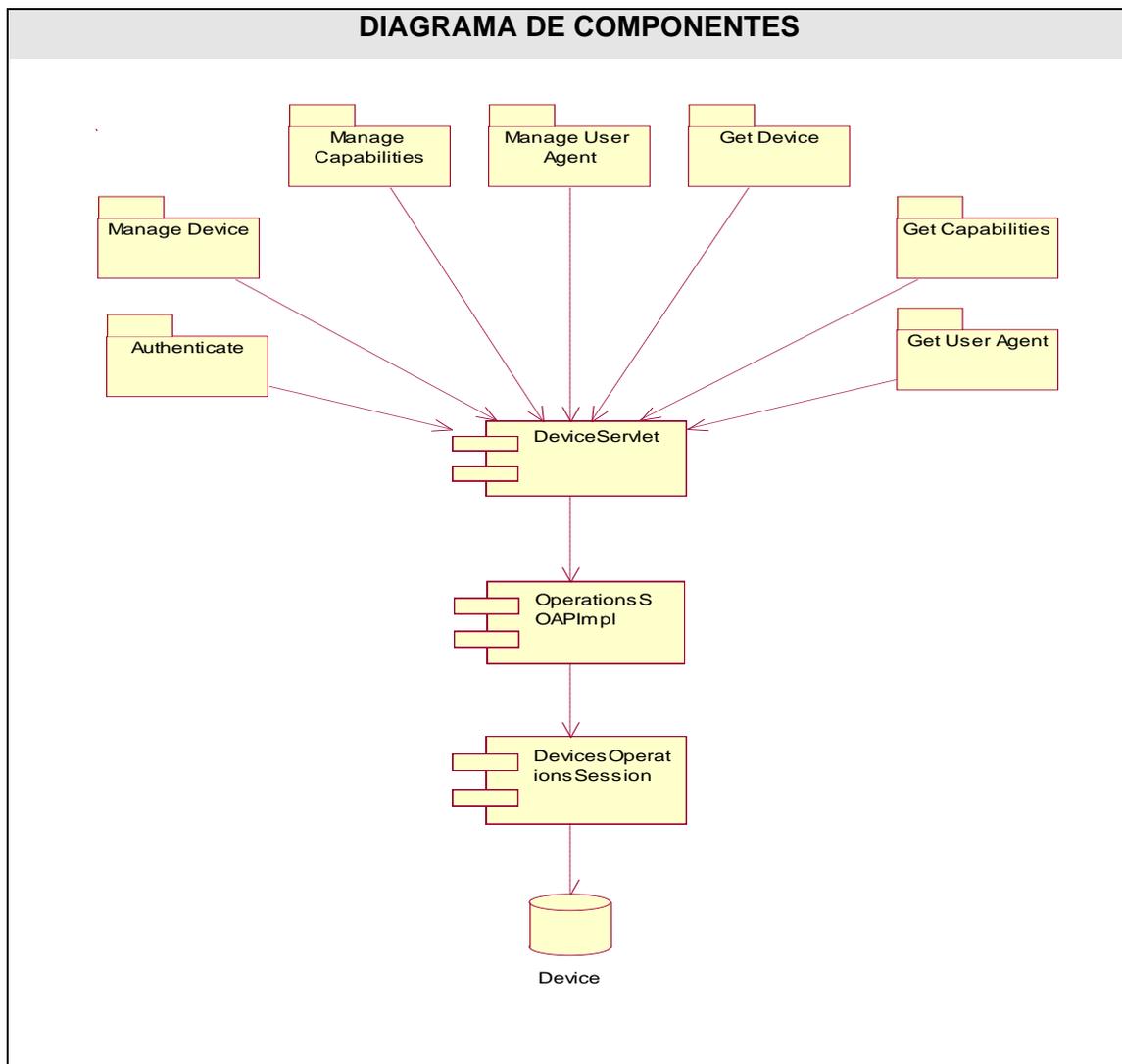


Figura 27. Diagrama de Componentes

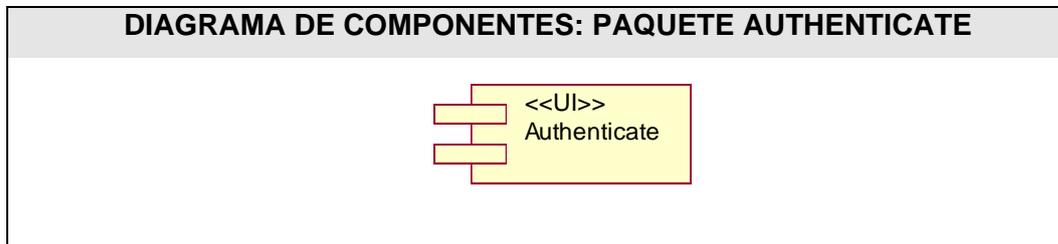


Figura 28. Diagrama de Componentes del Paquete Authenticate

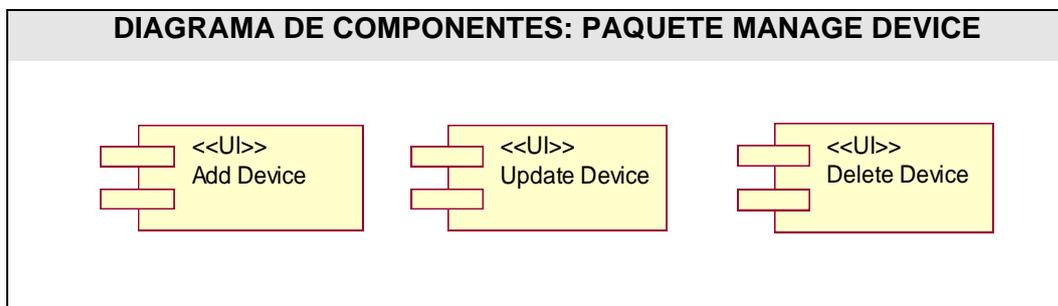


Figura 29. Diagrama de Componentes del Paquete Manage Device

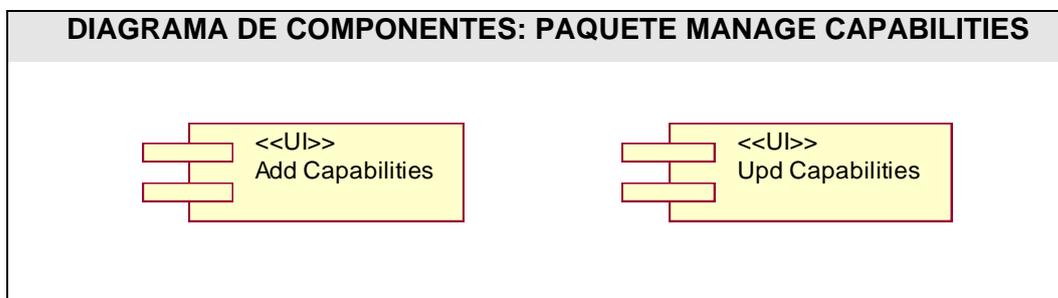


Figura 30. Diagrama de Componentes del Paquete Manage Capabilities

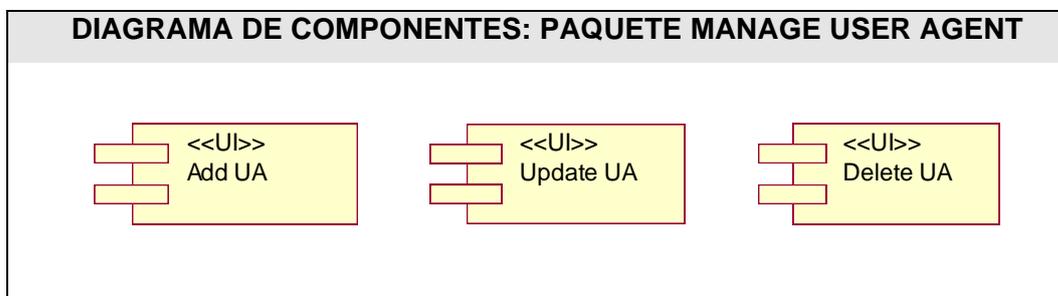


Figura 31. Diagrama de Componentes del Paquete Manage User Agent

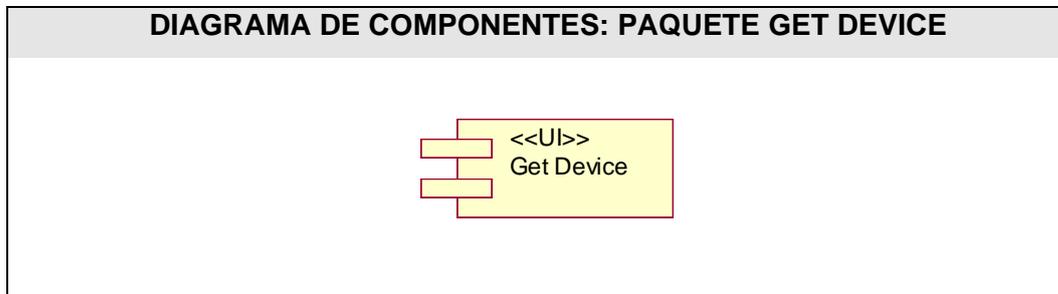


Figura 32. Diagrama de Componentes del Paquete Get Device

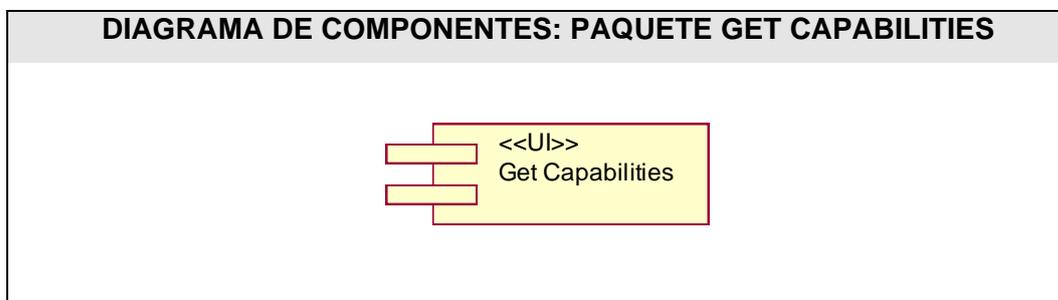


Figura 33. Diagrama de Componentes del Paquete Get Capabilities

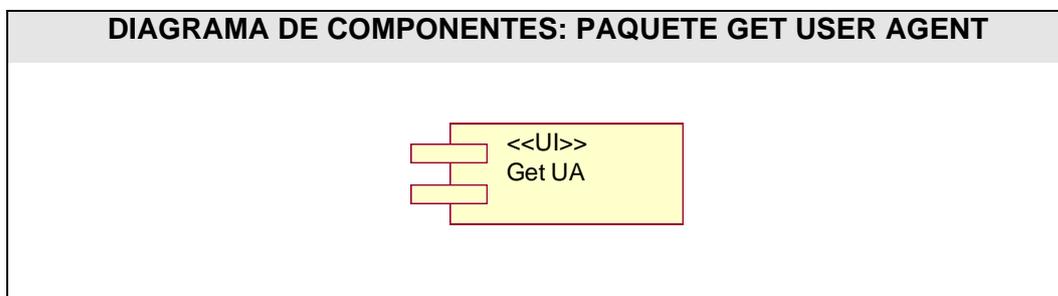


Figura 34. Diagrama de Componentes del Paquete Get User Agent

4.4 Conclusiones

En este capítulo se describió la distribución del sistema en nodos mediante el diagrama de despliegue. Se mostró las organizaciones y las dependencias lógicas entre componentes a través del diagrama de componentes.

CAPÍTULO V. ESTUDIO DE FACTIBILIDAD.

5.1 Introducción

En este capítulo se realizará mediante el método de Puntos de Caso de Uso una estimación del esfuerzo, costo y tiempo de desarrollo del proyecto. Además se hará un estudio de los costos y beneficios del mismo para llegar a la conclusión si el sistema es factible o no.

5.2 Planificación.

5.2.1 Características del proyecto.

- Entradas Externas

Las entradas externas se definen como un proceso mediante el cual ciertos datos cruzan la frontera del sistema desde afuera hacia adentro. En nuestra aplicación tenemos las siguientes entradas externas:

Tabla 3. Entradas Externas

Nombre de la entrada externa	Cantidad de ficheros	Cantidad de Elementos de datos	Clasificación (Simple, Media y Compleja)
Adicionar un dispositivo	1	5	Simple
Actualizar un dispositivo	1	5	Simple
Eliminar un dispositivo	3	1	Media
Adicionar características de un dispositivo	1	156	Media
Actualizar características de un dispositivo	1	156	Media
Adicionar User Agent	1	2	Simple
Actualizar User Agent	1	2	Simple
Eliminar User Agent	1	1	Simple

- Peticiones

Las peticiones se definen como un proceso con componentes de entrada y de salida donde un actor del sistema rescata datos de uno o más Ficheros Internos o Archivos de Interfaz Externos. Los datos de entrada no actualizan ni mantienen ningún archivo y los datos de salida no contienen datos derivados (es decir, los datos de salida son básicamente los mismos que se obtienen de los archivos). En nuestra aplicación tenemos las siguientes peticiones:

Tabla 4. Peticiones

Nombre de la petición	Cantidad de ficheros	Cantidad de Elementos de datos	Clasificación (Simple, Media y Compleja)
Listar datos de un dispositivo	1	5	Simple
Listar características de un dispositivo	1	157	Media
Listar User Agent	1	1-19	Simple

- Ficheros Internos

Los Ficheros Internos son un grupo de datos relacionados lógicamente e identificables por el usuario, que residen dentro de los límites del sistema y se mantienen a través de las Entradas Externas. En nuestra aplicación tenemos los siguientes Ficheros Internos:

Tabla 5. Ficheros Internos

Nombre del Fichero Interno	Cantidad de records	Cantidad de Elementos de datos	Clasificación (Simple, Media y Compleja)
Dispositivo	1	7	Simple
Características del dispositivo	2	156	Complejo
User Agent	2	4	Simple

Tabla 6. Puntos de Función Desajustados

Elementos	Simple	X Peso	Medios	X Peso	Complejos	X Peso	Subtotal de puntos de función
Ficheros lógicos internos	2	14			1	15	29
Entradas externas	5	15	3	12			27
Peticiones	2	6	1	4			10
Total							66

5.2.2 Cálculo de las instrucciones fuentes, esfuerzo, tiempo de desarrollo, cantidad de hombres y costo.

Tabla 7. Características

Características	Valor
Puntos de función desajustados	66
Lenguaje	53
Instrucciones fuentes por puntos de función	3498
Instrucciones fuentes	3.498

Tabla 8. Cálculos

Cálculo de	Valor
Esfuerzo	8.3369(mes-hombre)
Tiempo de desarrollo	4
Cantidad de hombres	2
Salario medio	100
Costo	\$ 833.69

Para calcular el esfuerzo realizaremos la estimación mediante el análisis de Puntos de Casos de Uso, el cual es un método propuesto originalmente por Gustav Karner de Objectory AB, y posteriormente refinado por muchos otros autores. Realizaremos esta estimación a través de una serie de pasos.

- Paso 1. Cálculo de Puntos de Casos de Uso sin ajustar

$$UUCP = UAW + UUCW$$

Donde

UUCP: Puntos de Casos de Uso sin ajustar

UAW: Factor de Peso de los Actores sin ajustar

UUCW: Factor de Peso de los Casos de Uso sin ajustar

Tabla 9. Factor de peso de los casos de uso sin ajustar.

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

$$UAW = \sum \text{cant actores} * \text{Peso}$$

$$UAW = 1 * 2 + 2 * 3$$

$$UAW = 8$$

Tabla 10. Factor de peso de los casos de uso sin ajustar.

Tipo de CU	Descripción	Cantidad de CU	Peso
Simple	El caso de uso tiene de 1 a 3 transacciones.	5	5

Medio	El caso de uso tiene de 4 a 7 transacciones.	2	10
Complejo	El caso de uso tiene más de 8 transacciones.	0	15

$$UUCW = \sum \text{cant CU} * \text{Peso}$$

$$UUCW = 5 * 5 + 2 * 10$$

$$UUCW = 45$$

$$UUCP = UAW + UUCW$$

$$UUCP = 8 + 45$$

$$UUCP = 53$$

- Paso 2. Cálculo de Puntos de Casos de Uso ajustados

$$UCP = UUCP * TCF * EF$$

Donde

UCP: Puntos de Casos de Uso ajustados

UUCP: Puntos de Casos de Uso sin ajustar

TCF: Factor de complejidad técnica

EF: Factor de ambiente

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

Tabla 11. Factor de complejidad técnica.

Factor	Descripción	Peso	Valor asignado
T1	Sistema distribuido	2	3
T2	Tiempo de respuesta	1	4
T3	Eficiencia del usuario final	1	2
T4	Funcionamiento Interno complejo	1	5

T5	El código debe ser reutilizable	1	4
T6	Facilidad de instalación	0,5	3
T7	Facilidad de uso	0,5	3
T8	Portabilidad	2	5
T9	Facilidad de cambio	1	2
T10	Concurrencia	1	4
T11	Incluye objetivos especiales de seguridad	1	2
T12	Provee acceso directo a terceras partes	1	1
T13	Se requieren facilidades especiales de entrenamiento de usuarios	1	1

$$TCF = 0.6 + 0.01 \times \sum (\text{Peso}_i \times \text{Valor asignado}_i)$$

$$TCF = 0.6 + 0.01 \times (6 + 4 + 2 + 5 + 4 + 1.5 + 1.5 + 10 + 2 + 4 + 2 + 1 + 1)$$

$$TCF = 0.6 + 0.01 \times 44$$

$$TCF = 1.04$$

El Factor de ambiente (EF) se calcula mediante la cuantificación de un conjunto de factores que determinan las habilidades y el entrenamiento del grupo involucrado en el desarrollo. Cada uno de los factores se cuantifica con un valor de 0 a 5, donde 0 significa un aporte irrelevante y 5 un aporte muy importante.

Tabla 12. Factor de ambiente

Factor	Descripción	Peso	Valor asignado
E1	Familiaridad con el modelo de proyecto utilizado	1,5	4
E2	Experiencia en la aplicación	0,5	3
E3	Experiencia en la orientación a objetivos.	1	5
E4	Capacidad del analista líder.	0,5	4
E5	Motivación.	1	5
E6	Estabilidad de requerimientos	2	5

E7	Personal Part-Time	-1	0
E8	Dificultad del lenguaje de programación	-1	3

$$EF = 1.4 - 0.03 \times \sum (\text{Peso}_i \times \text{Valor asignado}_i)$$

$$EF = 1.4 - 0.03 \times (6 + 1.5 + 5 + 2 + 5 + 10 - 3)$$

$$EF = 1.4 - 0.03 \times 26.5$$

$$EF = 0.605$$

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

$$UCP = 53 \times 1.04 \times 0.605$$

$$UCP = 33.3476$$

- Paso 3. Estimación del esfuerzo a través de los puntos de casos de uso

$$E = UCP \times CF$$

Donde:

E: Esfuerzo estimado en horas hombres.

UCP: Punto de casos de usos ajustados.

CF: Factor de conversión.

Para obtener el Factor de conversión se cuentan 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 o 4, se utiliza el factor de conversión 28 horas-hombre/Punto de Casos de Uso. 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 utilizaremos el Factor de conversión 20 horas-hombre/Punto de Casos de Uso.

$$E = UCP \times CF$$

$$E = 33.3476 \times 20$$

$$E = 666.952$$

- Paso 4. Calcular esfuerzo de todo el proyecto

Tabla 13. Esfuerzo del proyecto

Actividad	Porcentaje %	Horas-Hombres
Análisis	10	166.738
Diseño	20	333.476
Implementación	40	666.952
Pruebas	15	250.107
Sobrecarga (otras actividades)	15	250.107
Total	100	1667.38

El Esfuerzo Total sería 1667.38 horas-hombre, si estimamos teniendo en cuenta las condiciones que presentan los desarrolladores de este software, que un mes tiene 200 horas laborables entonces el Esfuerzo Total en mes-hombre sería 8.3369 mes-hombre.

- Paso 4. Calcular el costo de todo el proyecto

$$\text{Costo} = \text{CHM} * \text{ET} / \text{CH}$$

Donde

CH: Cantidad de hombres.

CHM: Costo Hombre - Mes

ET = Esfuerzo Total

Se asume como Cantidad de hombres 2 hombres y como Salario Promedio mensual \$100.00.

$$\text{CHM} = \text{CH} * \text{Salario Promedio}$$

$$\text{CHM} = 2 * 100$$

$$\text{CHM} = 200.00$$

$$\text{Costo} = \text{CHM} * \text{ET} / \text{CH}$$

$$\text{Costo} = 200.00 * 8.3369 / 2$$

Costo = \$ 833.69

- Paso 5. Calcular el tiempo de desarrollo de todo el proyecto

Tiempo = E_T / CH

Tiempo = 8.3369 / 2

Tiempo = 4.16845 \approx 4 meses

5.3 Beneficios tangibles e intangibles.

El software se está realizando con fines de integrarse a la nueva versión de la plataforma Mobistore que la empresa Procyon está desarrollando. Esta plataforma presenta fines comerciales, tiene entre otros objetivos la correcta administración de contenidos y manipulación de los datos de los dispositivos móviles. *Devices Manager* es un software para la manipulación de datos de diferentes dispositivos móviles y ofrece diversos beneficios, entre estos podemos identificar los siguientes:

- La aplicación cuenta con una interfaz gráfica agradable y sencilla, mediante la cual el usuario puede entrar los datos a ser procesados por la aplicación. Esto posibilita aumentar el grado de satisfacción del usuario puesto que el proyecto anterior *Devices Manager* no contaba con esta posibilidad.
- Se incorporaron nuevas funcionalidades al sistema que permiten brindar un gran valor de satisfacción a sus clientes. Estas funcionalidades facilitan la navegación del usuario en la aplicación además de las operaciones que se realizan en la misma.
- Posteriormente la aplicación se vinculará con la plataforma Mobistore posibilitando que esta pueda pedir los datos de los diferentes dispositivos móviles sin la intervención de un usuario.

5.4 Análisis de costo.

Para el desarrollo del sistema no se incurrió en grandes gastos en el pago de licencias de software, debido que la mayoría de las herramientas que se usaron eran libres. Las implementaciones se planean y automatizan, por lo que reducen los costos al reducir el tiempo de implementación.

El costo en el que se incurre al capacitar al usuario es mínimo debido a que nuestro sistema es fácil de entender y consta de una interfaz visual amigable para el usuario. No se incurrieron en demasiados gastos en cuanto al pago de los desarrolladores.

Teniendo en cuenta los costos y los beneficios planteados anteriormente llegamos a la conclusión de que es factible desarrollar un sistema con estas características.

5.5 Conclusiones

El estudio de factibilidad realizado en este capítulo mediante el método de Puntos de Caso de Uso junto con el cálculo del esfuerzo y tiempo de desarrollo del proyecto llevan a la conclusión de que el sistema es factible.

CONCLUSIONES

Para la realización de este sistema se hizo un estudio y un análisis acerca de los sistemas existentes que manipulan la información de los dispositivos móviles, las tecnologías, el lenguaje de programación y las herramientas a utilizar. Se decidió utilizar el lenguaje de código abierto Java y la plataforma J2EE.

Se utilizó el framework EJB3 para la persistencia de datos, Axis para el desarrollo del servicio web y Jboss como servidor de aplicaciones. La metodología de desarrollo utilizada fue RUP, esta permitió desarrollar un software de alta calidad y ayudó a controlar el desarrollo del software a lo largo de todo el proceso.

Como resultado de este trabajo se publicó un servicio web que permite la correcta manipulación de los datos de los dispositivos móviles. Se desarrolló una aplicación web mediante la cual se adicionan, actualizan y eliminan correctamente los distintos datos relacionados a los dispositivos móviles. Este sistema cumple con los requerimientos funcionales y no funcionales planteados inicialmente. Por lo tanto podemos decir que los objetivos propuestos se cumplieron exitosamente.

RECOMENDACIONES

- Se recomienda a la empresa Procyon revisar los datos provenientes de las fuentes de información utilizadas, debido que en el desarrollo de este sistema se encontraron algunos datos erróneos que pudieran arrojar resultados no esperados o información no válida.
- Mantener constantemente actualizada la base de datos de los dispositivos móviles del sistema, debido que los teléfonos celulares están en constante evolución y siempre surgen nuevos modelos o se les agregan nuevas características a modelos existentes.

BIBLIOGRAFÍA

- Bequet Henry, Kunumpurath Meeraj, Rhody Sean, Tost Andre. Beginning Java Web Services. Wrox Press 2002. 411 páginas.ç
- 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.
- Ayuda de la herramienta Case Rational Rose.
- Badani, Sigifredo E. Hernández. Métricas de estimación de tamaño Puntos de Caso de Uso.Septiembre 2002.
- Mario Peralta. Estimación del esfuerzo basada en casos de uso [en línea]. Buenos Aires, 2004. Disponible en World Wide Web:
<<http://www.itba.edu.ar/capis/rtis/rtis-6-1/estimaci%F3n-del-esfuerzo-basada-en-casos-de-usos.pdf> >
- Breve historia de la tecnología Java [en línea]. Disponible en World Wide Web:
< <http://www.java.com/es/about/> >
- Cruz Martínez-Lacaci, Fernando. El código abierto en el ámbito empresarial [en línea].2007. Disponible en World Wide Web:
< <http://www.mkm-pi.com/mkmpi.php?article56> >
- La tecnología Java: Programación Orientada a Objetos, Máquina Virtual de Java, Clases en Java, Herencia en Java, J2EE, J2SE [en línea].2006. Madrid, España. Disponible en World Wide Web:
< http://java.ciberaula.com/articulo/tecnologia_java/ >
- Presentación [en línea]. Disponible en World Wide Web:
< <http://www.jtech.ua.es/j2ee/2004-2005/index.htm> >

- Herrera Basurto, Cristhian Kirs. Comparativa entre Hibernate y EJB3 en la Capa de Persistencia[en línea]. Septiembre, 2007. Disponible en World Wide Web:
<<http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=hibernateVSEJB3>>
- Pérez Benedi, Jennifer. Web Services [en línea]. Valencia, España. Disponible en World Wide Web:
<http://72.14.205.104/search?q=cache:b3ixrFBv_a4J:www.gvpontis.gva.es/fileadmin/conselleria/images/Documentacion/migracionSwAbierto/Anexos/Anexo_C9.pdf+web+service%2Baxis&hl=es&ct=clnk&cd=21&gl=cu&lr=lang_es&client=firefox-a>
- Metodología para el desarrollo de las herramientas informáticas de la REyE[en línea]. 2006. Costa Rica. Disponible en World Wide Web:
<http://72.14.205.104/search?q=cache:F8VHuskoSUMJ:old.iabin.net/binary_docs/meetings_workshops/SSTN_2006_08_Costa_Rica/Metodologia_desarrollo_herramientas_red.pdf+%22metodolog%C3%ADa+de+desarrollo%22&hl=es&ct=clnk&cd=25&gl=cu&lr=lang_es>
- [8] RUP - Rational Unified Process[en línea]. Disponible en World Wide Web:
<<http://www.kynetia.es/calidad/rup-rational-unified-process.html>>
- MySQL 5.0 Reference Manual[en línea]. Noviembre, 2007. Disponible en World Wide Web:
<<http://dev.mysql.com/doc/refman/5.0/es/index.html>>
- Productos Rational[en línea]. Disponible en World Wide Web:
<http://www.infosgroup.com/paginas/v4/publico/soluciones/soluciones_producto/rational/productos_rational.asp?referal=/paginas/v4/publico/soluciones/soluciones_producto/rational.asp#rose>
- UML - Unified Modeling Lenguaje [en línea]. Disponible en World Wide Web:
<<http://gidis.ing.unlpam.edu.ar/personas/glafuente/uml/uml.html>>

REFERENCIAS BIBLIOGRAFICAS

[1] Breve historia de la tecnología Java [en línea]. Disponible en World Wide Web:

< <http://www.java.com/es/about/> >

[2] Cruz Martínez-Lacaci, Fernando. El código abierto en el ámbito empresarial [en línea].2007. Disponible en World Wide Web:

< <http://www.mkm-pi.com/mkmpi.php?article56> >

[3] La tecnología Java: Programación Orientada a Objetos, Máquina Virtual de Java, Clases en Java, Herencia en Java, J2EE, J2SE [en línea].2006. Madrid, España. Disponible en World Wide Web:

< http://java.ciberaula.com/articulo/tecnologia_java/ >

[4] Presentación [en línea]. Disponible en World Wide Web:

< <http://www.jtech.ua.es/j2ee/2004-2005/index.htm> >

[5] Herrera Basurto, Cristhian Kirs. Comparativa entre Hibernate y EJB3 en la Capa de Persistencia[en línea]. Septiembre, 2007. Disponible en World Wide Web:

< <http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=hibernateVSEJB3> >

[6] Metodología para el desarrollo de las herramientas informáticas de la REyE[en línea]. 2006. Costa Rica. Disponible en World Wide Web:

<http://72.14.205.104/search?q=cache:F8VHuskoSUMJ:old.iabin.net/binary_docs/meetings_workshops/SSTN_2006_08_Costa_Rica/Metodologia_desarrollo_herramientas_red.pdf+%22metodolog%C3%ADa+de+desarrollo%22&hl=es&ct=clnk&cd=25&gl=cu&lr=lang_es >

[7] RUP - Rational Unified Process[en línea]. Disponible en World Wide Web:

< <http://www.kynetia.es/calidad/rup-rational-unified-process.html> >

[8] Pérez Benedi, Jennifer. Web Services [en línea]. Valencia, España. Disponible en World Wide Web:

<http://72.14.205.104/search?q=cache:b3ixrFBv_a4J:www.gvpontis.qva.es/fileadmin/conselleria/images/Documentacion/migracionSwAbierto/Anexos/Anexo_C9.pdf+web+service%2Baxis&hl=es&ct=clnk&cd=21&gl=cu&lr=lang_es&client=firefox-a>

[9] MySQL 5.0 Reference Manual[en línea]. Noviembre, 2007. Disponible en World Wide Web:

< <http://dev.mysql.com/doc/refman/5.0/es/index.html> >

[10] Productos Rational[en línea]. Disponible en World Wide Web:

<http://www.infosgroup.com/paginas/v4/publico/soluciones/soluciones_producto/rational/productosrational.asp?referal=/paginas/v4/publico/soluciones/soluciones_producto/rational.asp#rose>

[11] UML - Unified Modeling Lenguaje[en línea]. Disponible en World Wide Web:

< <http://gidis.ing.unlpam.edu.ar/personas/glafuente/uml/uml.html> >

ANEXOS

Anexo 1: Descripción detallada de los casos de uso

Tabla 14. Descripción Detallada del Caso de Uso Authenticate User

Caso de uso	
CU-1	Authenticate User
Propósito	Autenticar un usuario para que el mismo pueda realizar determinadas acciones en el sistema.
Actores: Administrator	
Resumen: El caso de uso se inicia cuando el actor solicita autenticarse en el sistema. El actor introduce su nombre y contraseña, y el sistema realiza la acción seleccionada.	
Referencias	R1
Curso Normal de los Eventos	
Acción del actor	Respuesta del sistema
1. El actor selecciona autenticarse en el sistema.	1.1 El sistema muestra el formulario a llenar con su nombre y contraseña.
2. El actor llena los datos del formulario.	2.1 El sistema verifica que los datos son correctos.
	2.2 Si los datos son correctos el sistema autentica al usuario.
Flujo Alternativo	
Acción 2.2:	Si los datos no son correctos el sistema le muestra un mensaje de error y solicita que los datos sean llenados nuevamente.

Tabla 15. Descripción Detallada del Caso de Uso Get Device

Caso de uso	
CU-3	Get Device
Propósito	Listar los datos principales de un dispositivo móvil
Actores: User	
Resumen: El caso de uso se inicia cuando el actor solicita ver los datos de un dispositivo. El actor selecciona el criterio de búsqueda que puede ser por: el nombre completo del dispositivo, el	

identificador, el User Agent o el último que fue modificado. Posteriormente el sistema lista los datos del dispositivo.	
Referencias	R3
Curso Normal de los Eventos:	
Acción del actor	Respuesta del sistema
1. El actor selecciona Listar datos de un dispositivo.	1.1 El sistema muestra las opciones para buscar el dispositivo: el nombre completo del dispositivo, el identificador, el User Agent o el último que fue modificado.
2. El actor selecciona el criterio de búsqueda para buscar el dispositivo.	2.1 El sistema muestra los datos resultantes de la búsqueda.
3. El actor selecciona el dispositivo a buscar.	3.1 El sistema muestra los datos del dispositivo seleccionado.

Tabla 16. Descripción Detallada del Caso de Uso Manage Capabilities

Caso de uso	
CU-4	Manage Capabilities
Propósito	Adicionar o actualizar las características de un dispositivo móvil.
Actores: Administrator	
Resumen: El caso de uso se inicia cuando el actor solicita adicionar o actualizar las características de un dispositivo. En el caso de adicionar selecciona el dispositivo que se le van a adicionar las características mediante su nombre completo, llena los datos mostrados y adiciona los datos en la base de datos. En el caso de actualizar selecciona el dispositivo a modificar mediante su nombre completo y actualiza las características del mismo.	
Referencias	R4
Precondiciones	Usuario previamente autenticado.
Curso Normal de los Eventos	
Acción del actor	Respuesta del sistema
1. El actor selecciona Administrar Características.	1.1 El sistema muestra dos opciones: Adicionar y Actualizar Características.
Escenario 1: Adicionar Características.	
1. El actor selecciona la opción Adicionar	1.1 El sistema muestra el formulario a llenar.

Características.	
2. El actor llena los datos del formulario.	2.1 El sistema verifica que los datos son correctos.
	2.2 Si los datos son correctos el sistema adiciona las características del dispositivo a la base de datos y le muestra un mensaje al usuario de que adicionó los datos satisfactoriamente.
Flujo Alternativo	
Acción 2.2:	Si los datos no son correctos el sistema le muestra un mensaje de error y solicita que los datos sean llenados nuevamente.
Escenario 2: Actualizar Características.	
1. El actor selecciona la opción Actualizar Características.	1.1 El sistema muestra la opción de buscar el dispositivo por su nombre completo.
2. El actor selecciona un dispositivo.	2.1 El sistema muestra las características del dispositivo seleccionado.
3. El actor actualiza las características.	3.1 El sistema verifica que los datos son correctos.
	3.2 Si los datos son correctos el sistema actualiza las características del dispositivo en la base de datos y le muestra un mensaje al usuario de que actualizó los datos satisfactoriamente.
Flujo Alternativo	
Acción 3.2:	Si los datos no son correctos el sistema le muestra un mensaje de error y solicita que los datos sean llenados nuevamente.

Tabla 17. Descripción Detallada del Caso de Uso Get Capabilities

Caso de uso	
CU-5	Get Capabilities
Propósito	Listar las características de un dispositivo móvil
Actores: User	
Resumen: El caso de uso se inicia cuando el actor solicita ver las características de un dispositivo. El actor selecciona el criterio de búsqueda que puede ser por: el nombre completo del	

dispositivo o el User Agent. Posteriormente el sistema lista las características del dispositivo.	
Referencias	R5
Curso Normal de los Eventos:	
Acción del actor	Respuesta del sistema
1. El actor selecciona Listar las características de un dispositivo.	1.1 El sistema muestra las opciones para buscar el dispositivo: el nombre completo del dispositivo o el User Agent.
2. El actor selecciona el criterio de búsqueda para eliminar el dispositivo.	2.1 El sistema muestra los datos resultantes de la búsqueda.
3. El actor selecciona el dispositivo a buscar.	3.1 El sistema muestra las características del dispositivo seleccionado.

Tabla 18. Descripción Detallada del Caso de Uso Manage User Agent

Caso de uso	
CU-6	Manage User Agent
Propósito	Adicionar, actualizar o eliminar un User Agent de un dispositivo móvil.
Actores: Administrator	
Resumen: El caso de uso se inicia cuando el actor solicita adicionar, actualizar o eliminar un User Agent. En el caso de adicionar llena los datos mostrados y adiciona los datos en la base de datos. En el caso de actualizar o eliminar selecciona el User Agent, si desea actualizar modifica el nombre del User Agent y actualiza los datos en la base de datos y si desea eliminar el sistema elimina el User Agent.	
Referencias	R6
Precondiciones	Usuario previamente autenticado.
Curso Normal de los Eventos	
Acción del actor	Respuesta del sistema
1. El actor selecciona Administrar User Agent.	1.1 El sistema muestra tres opciones: Adicionar, Actualizar y Eliminar User Agent.
Escenario 1: Adicionar User Agent.	
1. El actor selecciona la opción Adicionar User Agent.	1.1 El sistema muestra el formulario a llenar.
2. El actor llena los datos del formulario.	2.1 El sistema verifica que los datos son correctos.

	2.2 Si los datos son correctos el sistema adiciona el User Agent a la base de datos y le muestra un mensaje al usuario de que adicionó los datos satisfactoriamente.
Flujo Alternativo	
Acción 2.2:	Si los datos no son correctos el sistema le muestra un mensaje de error y solicita que los datos sean llenados nuevamente.
Escenario 2: Actualizar User Agent.	
1. El actor selecciona la opción Actualizar User Agent.	1.1 El sistema muestra todos los User Agent.
2. El actor selecciona un User Agent.	2.1 El sistema muestra los datos del User Agent seleccionado.
3. El actor modifica el nombre del User Agent	3.1 El sistema verifica que la información esté correcta.
	3.2 Si la información es correcta el sistema actualiza los datos en la base de datos y le muestra un mensaje al usuario de que actualizó los datos satisfactoriamente.
Flujo Alternativo	
Acción 3.2:	Si la información no es correcta el sistema le muestra un mensaje de error y solicita que los datos sean llenados nuevamente.
Escenario 3: Eliminar User Agent.	
1. El actor selecciona la opción Eliminar User Agent.	1.1 El sistema muestra todos los User Agent.
3. El actor selecciona el User Agent a eliminar.	3.1 El sistema muestra un mensaje de confirmación.
4 El actor acepta eliminar el User Agent.	4.1 El sistema elimina el User Agent, actualiza la BD y le muestra un mensaje al usuario de que eliminó el User Agent satisfactoriamente.
Flujo Alternativo	

Acción 3.1:	El actor selecciona la opción cancelar y termina el caso de uso sin ejecutar ninguna acción.
-------------	--

Tabla 19. Descripción Detallada del Caso de Uso Get User Agent

Caso de uso	
CU-7	Get User Agent
Propósito	Listar los User Agent de un dispositivo móvil
Actores: User	
Resumen: El caso de uso se inicia cuando el actor solicita ver los User Agent de un dispositivo. El actor selecciona el nombre completo del dispositivo para ver sus User Agent. Posteriormente el sistema lista todos los User Agent.	
Referencias	R7
Curso Normal de los Eventos:	
Acción del actor	Respuesta del sistema
1. El actor selecciona Listar los User Agent de un dispositivo.	1.1 El sistema muestra los nombres completos de los dispositivos.
2. El actor selecciona el dispositivo a buscar.	2.1 El sistema muestra los User Agent del dispositivo seleccionado.

Anexo 2. Diagrama de Colaboración del Análisis

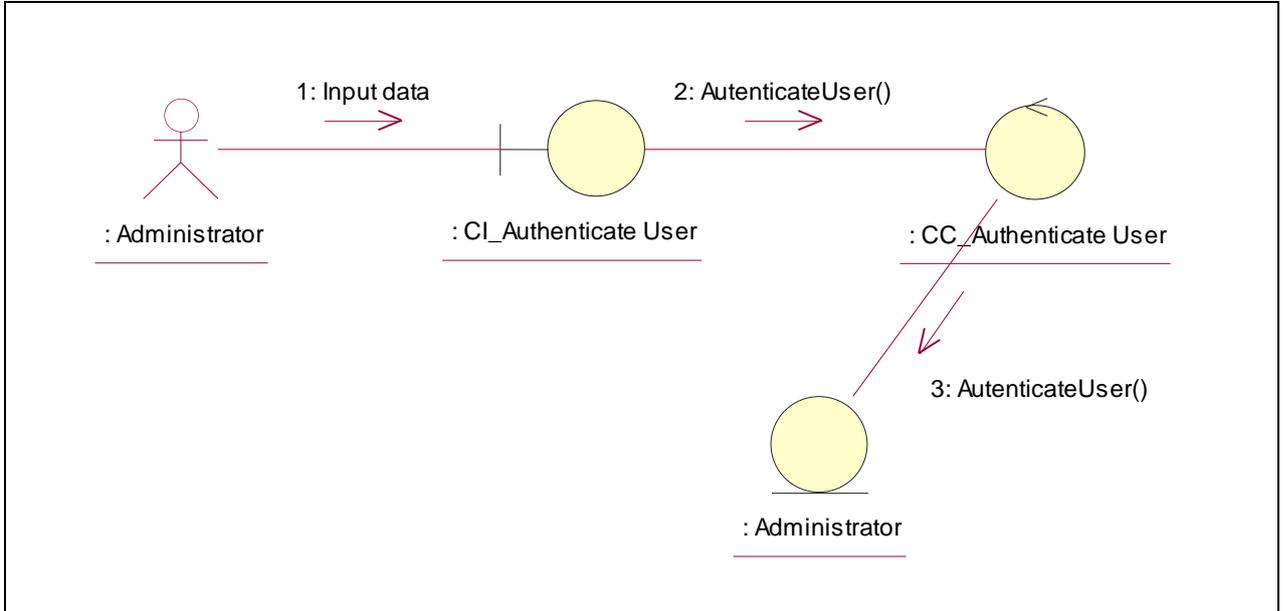


Figura 35. Diagrama de Colaboración del Análisis. CU Authenticate User

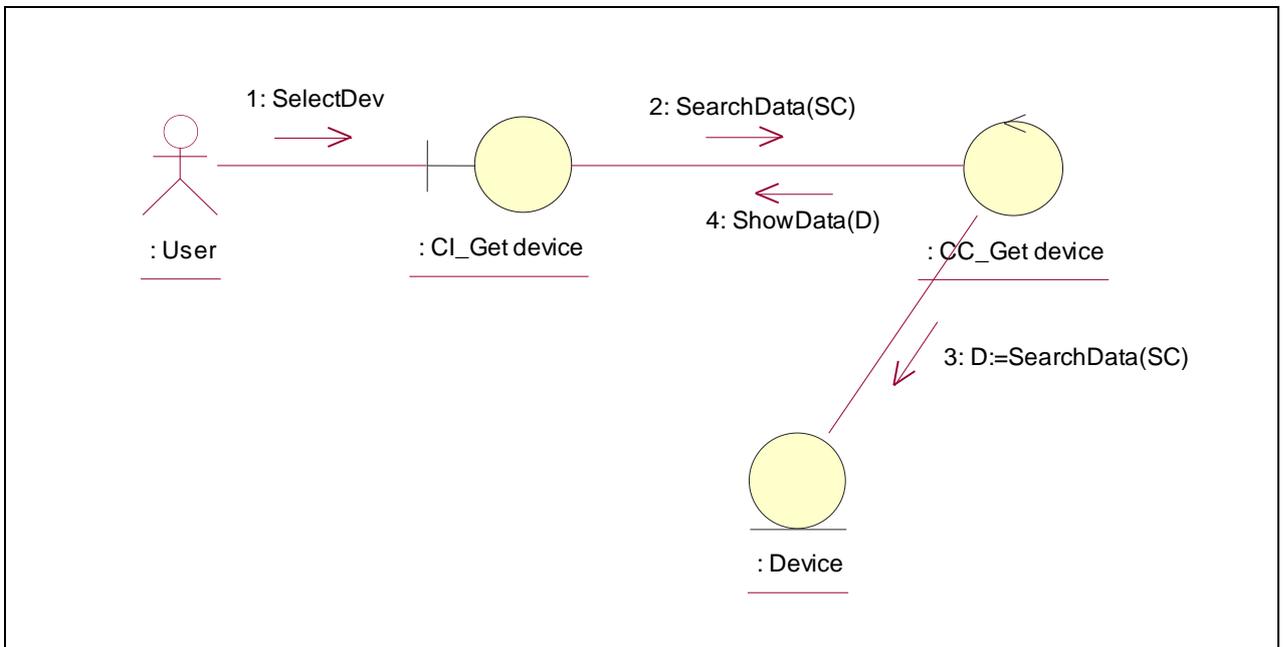


Figura 36. Diagrama de Colaboración del Análisis. CU Get Device

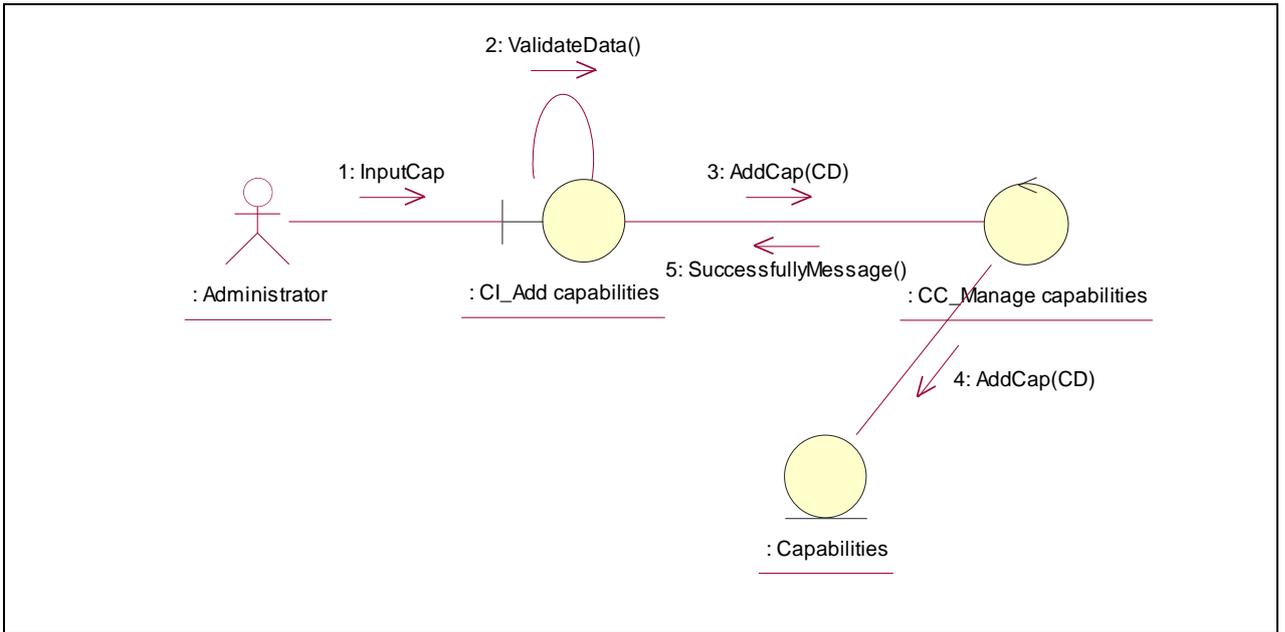


Figura 37. Diagrama de Colaboración del Análisis. CU Manage Capabilities (escenario Add Capabilities)

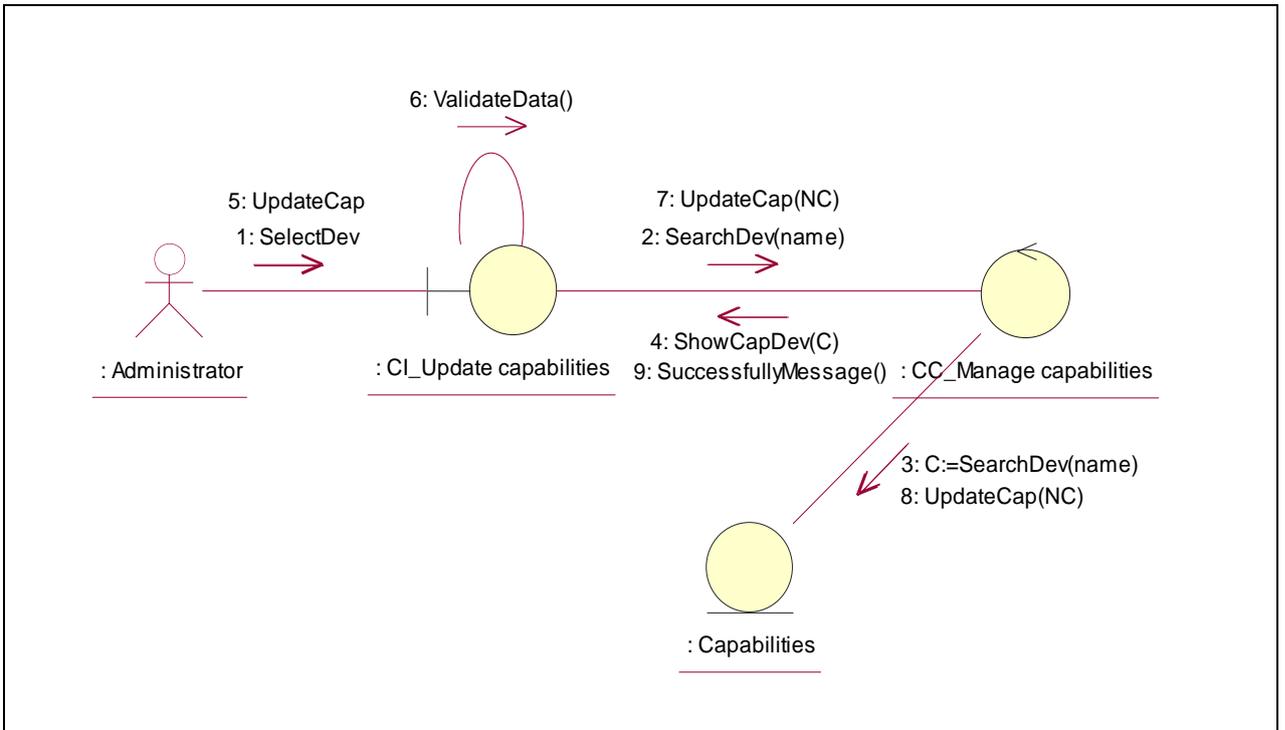


Figura 38. Diagrama de Colaboración del Análisis. CU Manage Capabilities (escenario Update Capabilities)

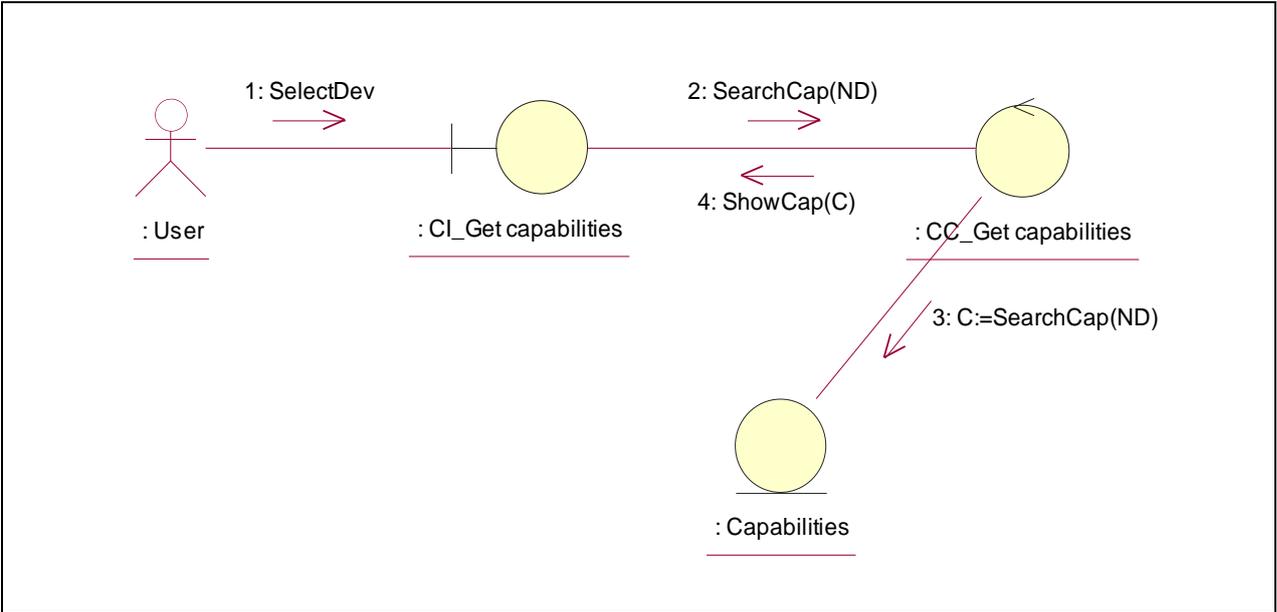


Figura 39. Diagrama de Colaboración del Análisis. CU Get Capabilities

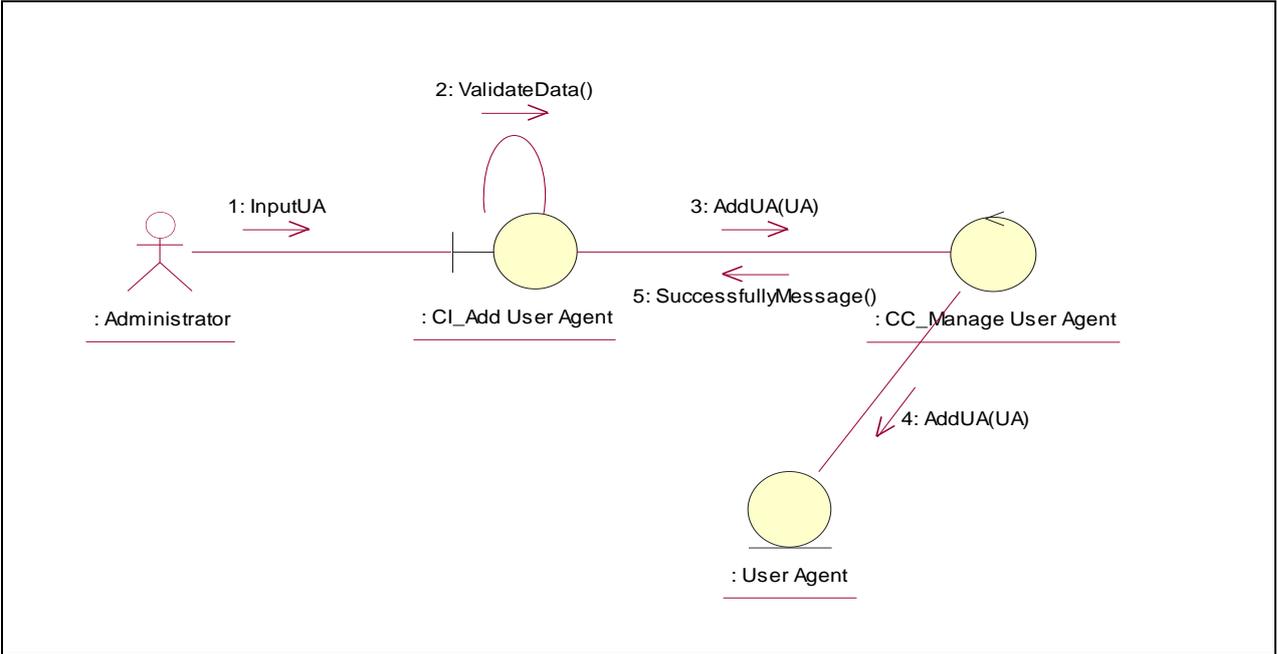


Figura 40. Diagrama de Colaboración del Análisis. CU Manage User Agent (escenario Add User Agent)

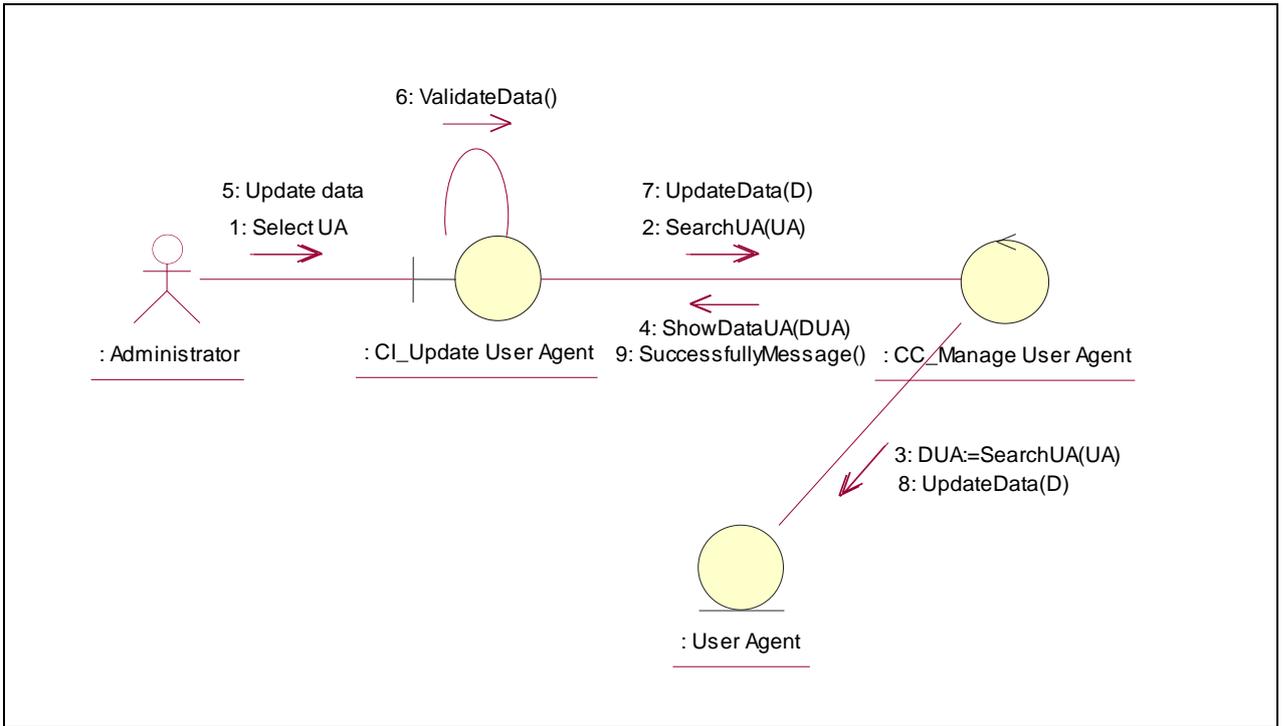


Figura 41. Diagrama de Colaboración del Análisis. CU Manage User Agent (escenario Update User Agent)

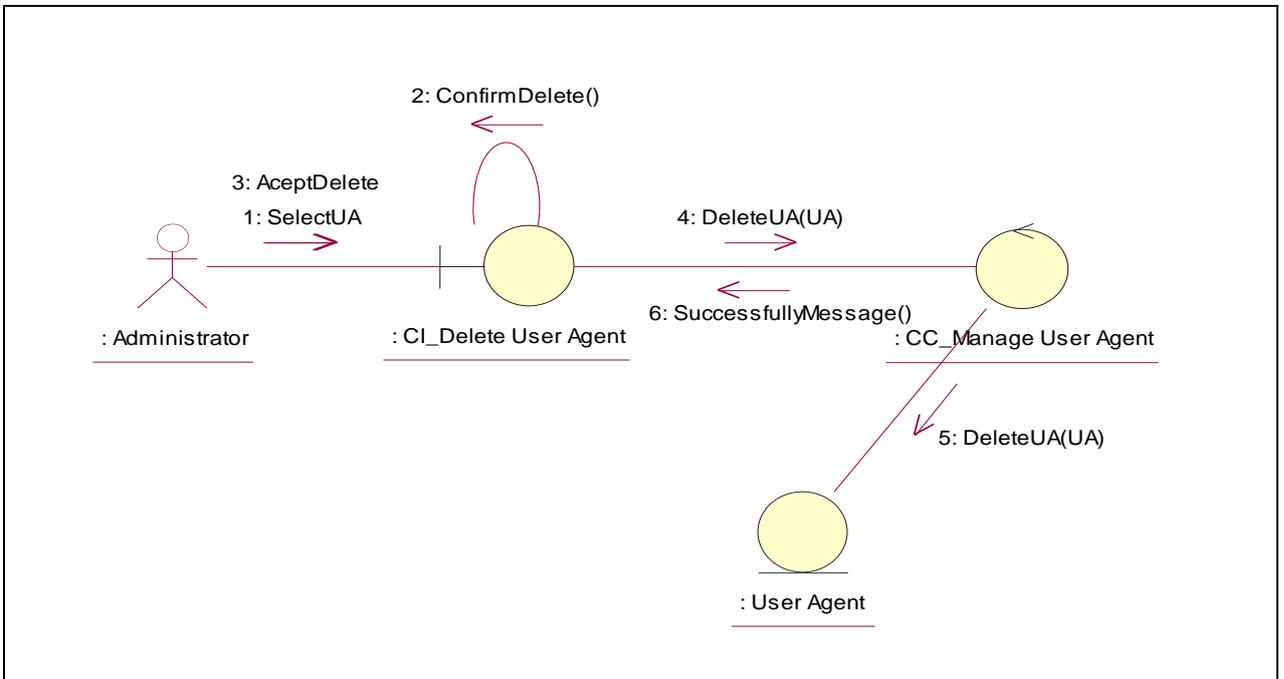


Figura 42. Diagrama de Colaboración del Análisis. CU Manage User Agent (escenario Delete User Agent)

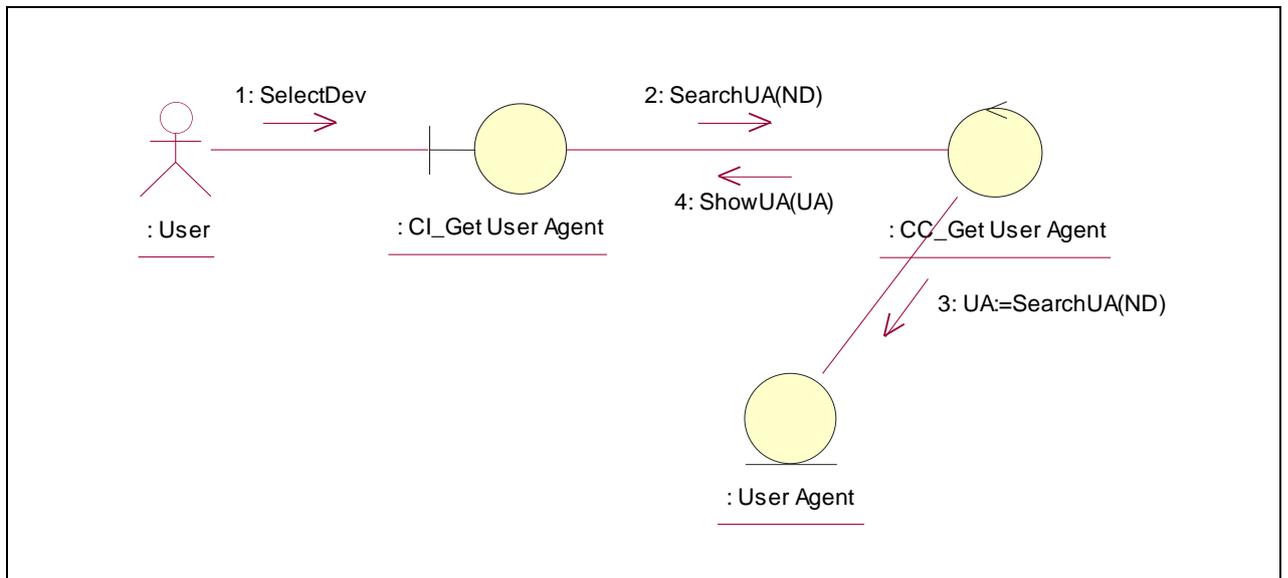


Figura 43. Diagrama de Colaboración del Análisis. CU Get User Agent

Anexo 3. Diagrama de Secuencia del Diseño

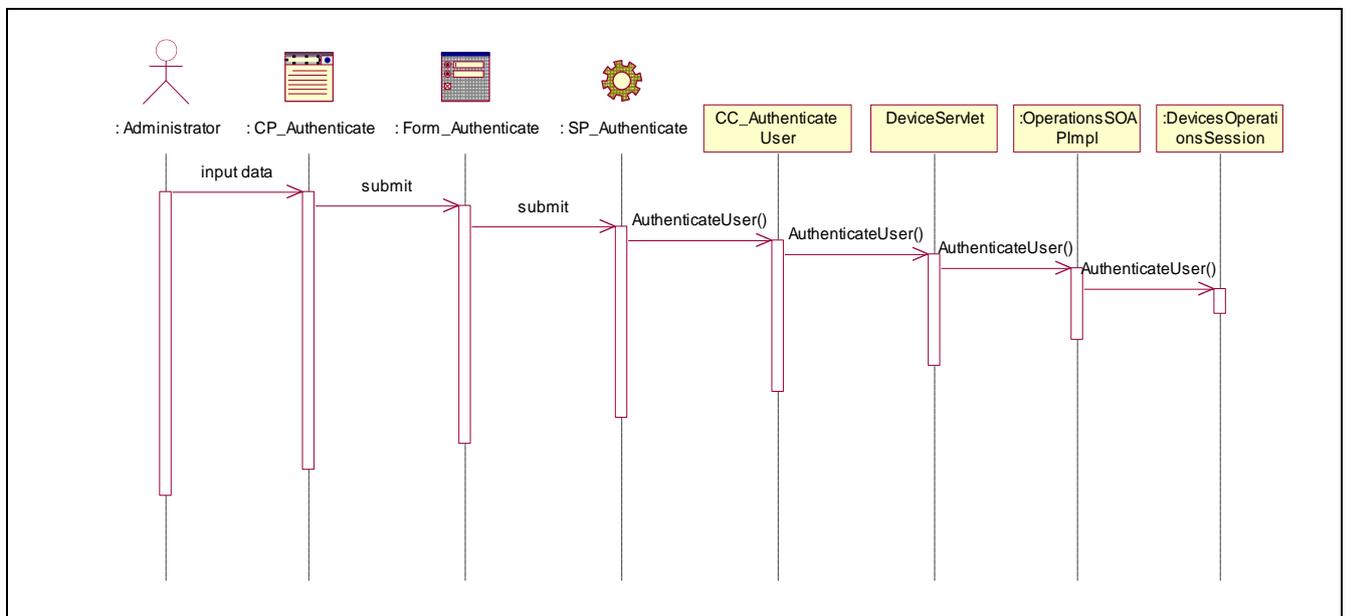


Figura 44. Diagrama de Secuencia del Diseño. CU Authenticate User

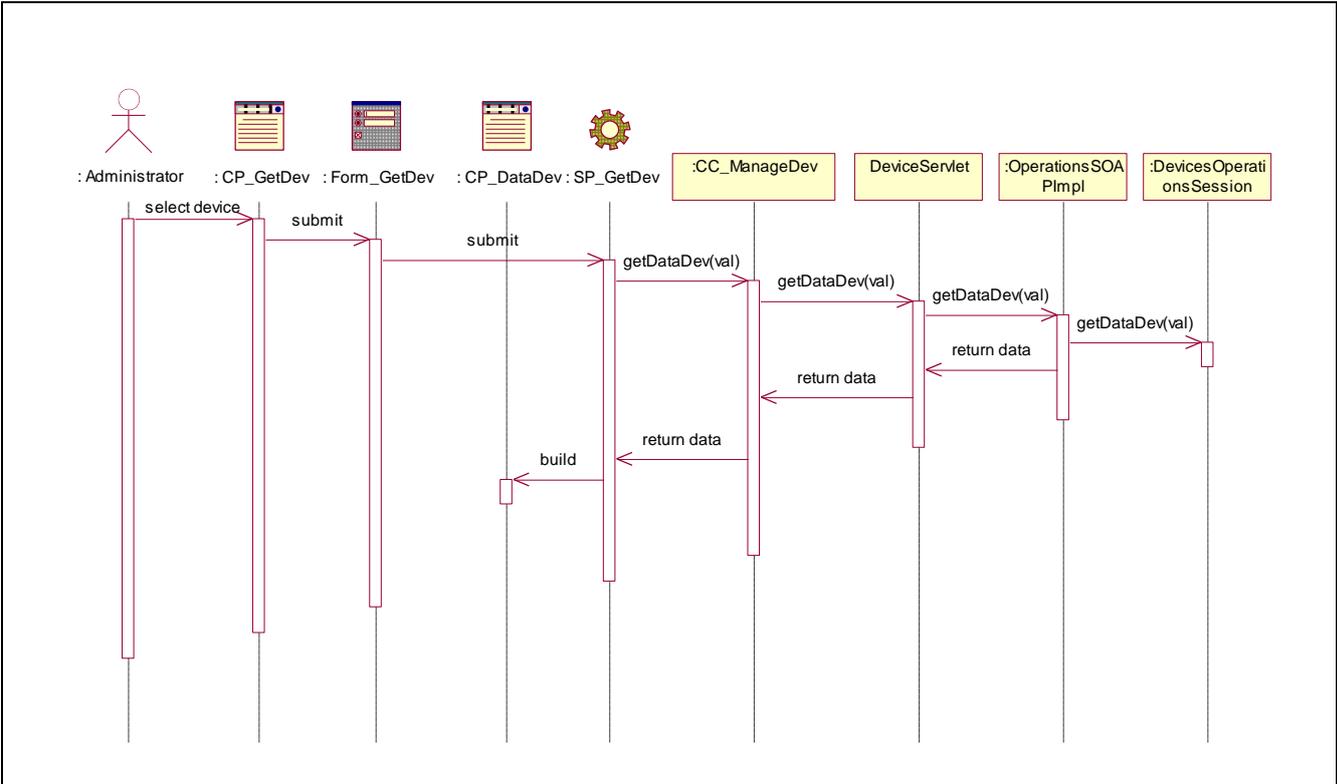


Figura 45. Diagrama de Secuencia del Diseño. CU Get Device

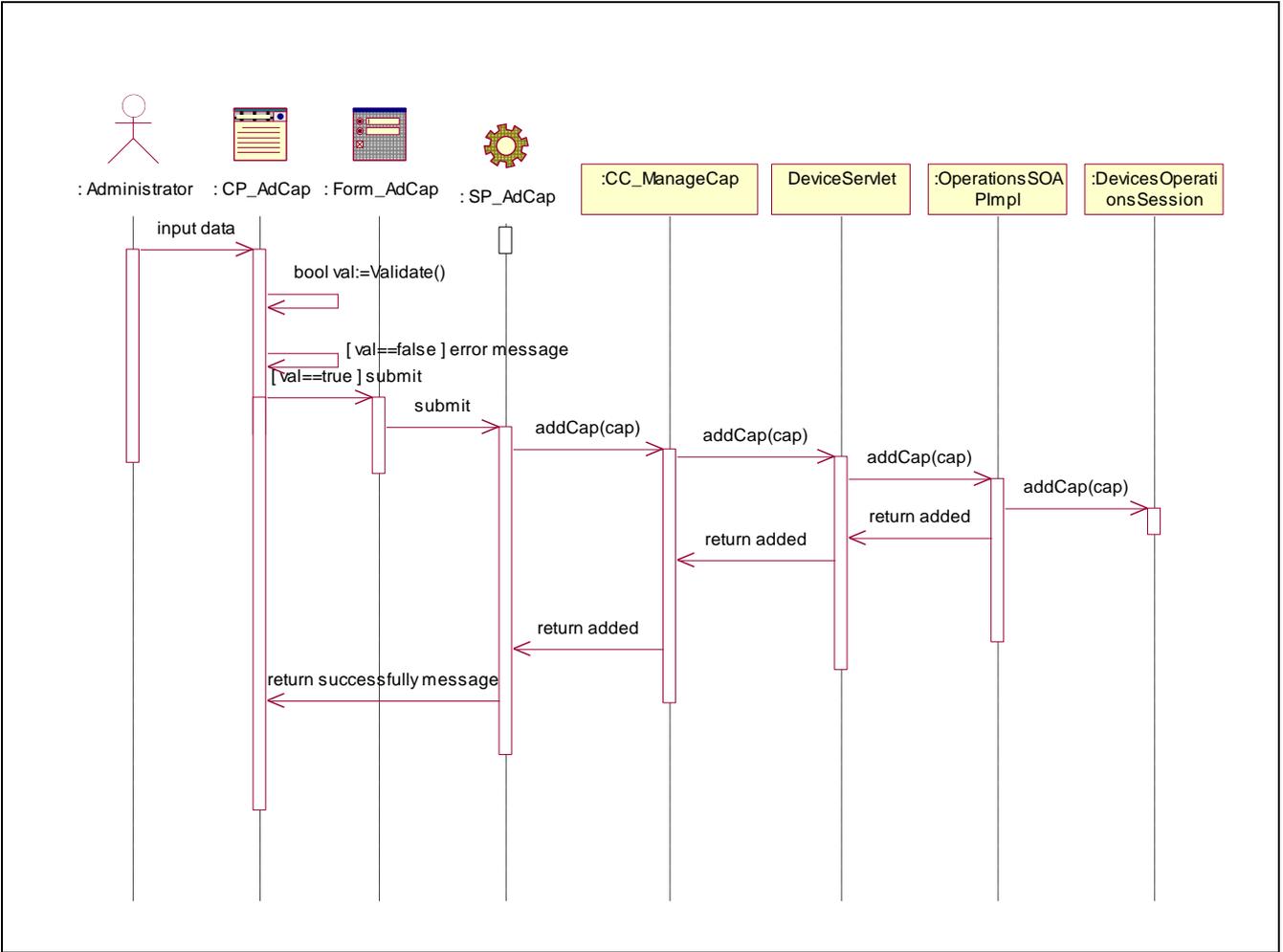


Figura 46. Diagrama de Secuencia del Diseño. CU Manage Capabilities (escenario Add Capabilities)

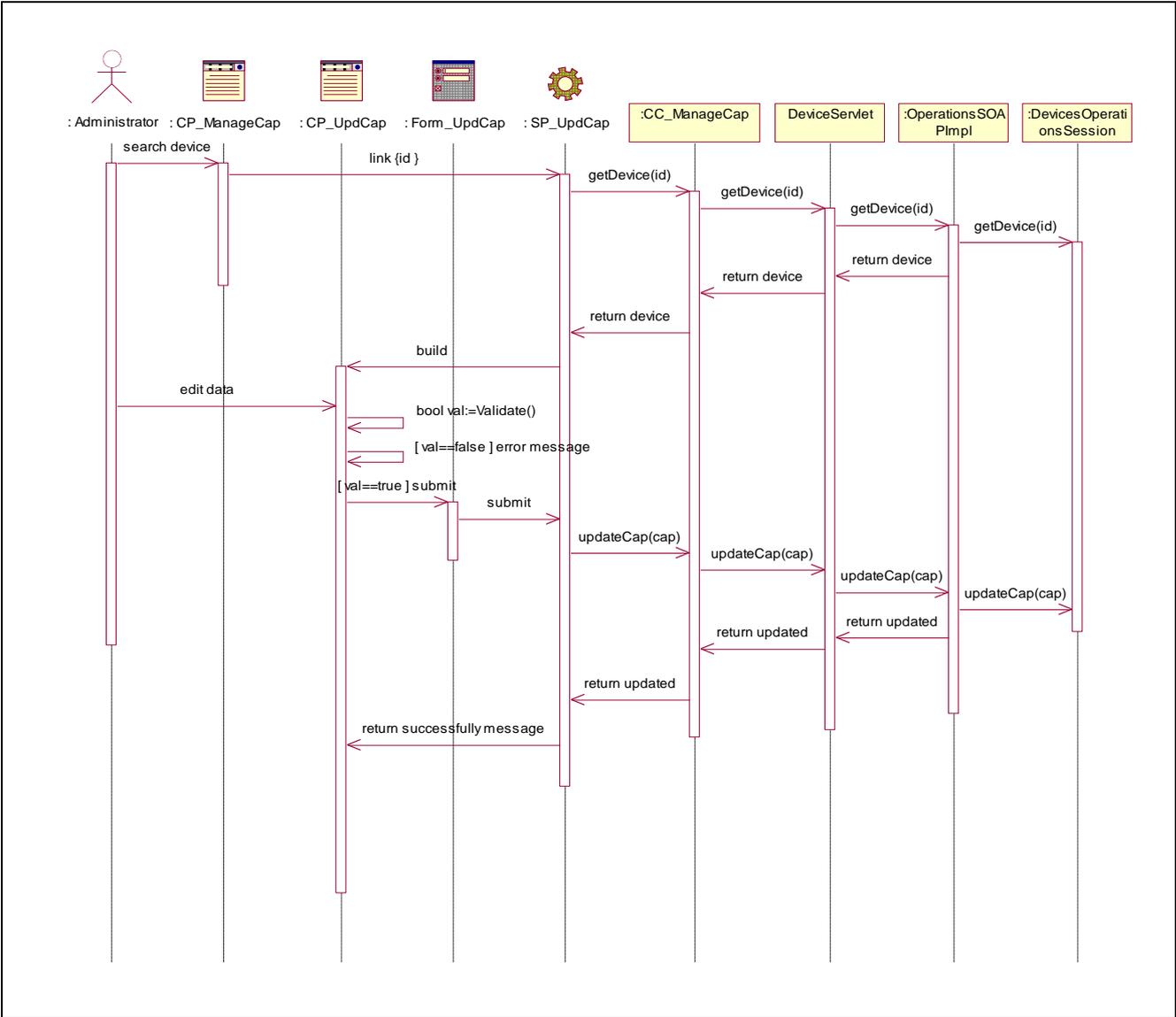


Figura 47. Diagrama de Secuencia del Diseño. CU Manage Capabilities (escenario Update Capabilities)

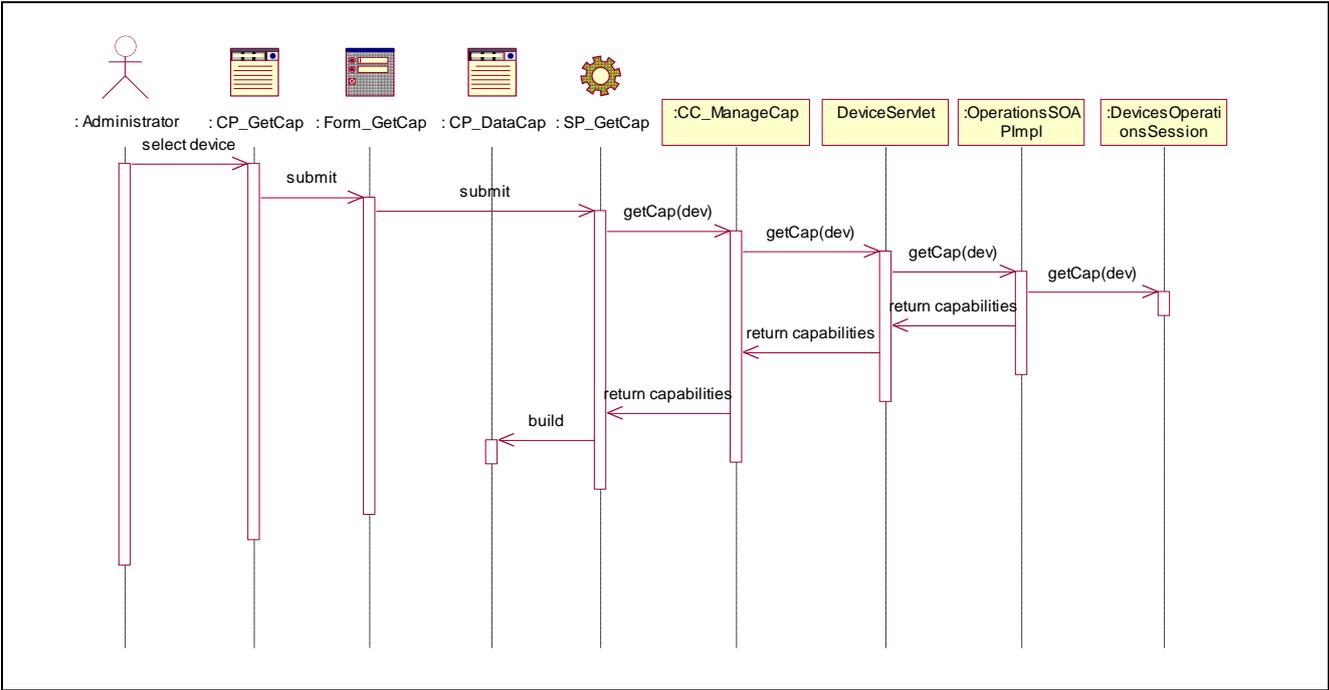


Figura 48. Diagrama de Secuencia del Diseño. CU Get Capabilities

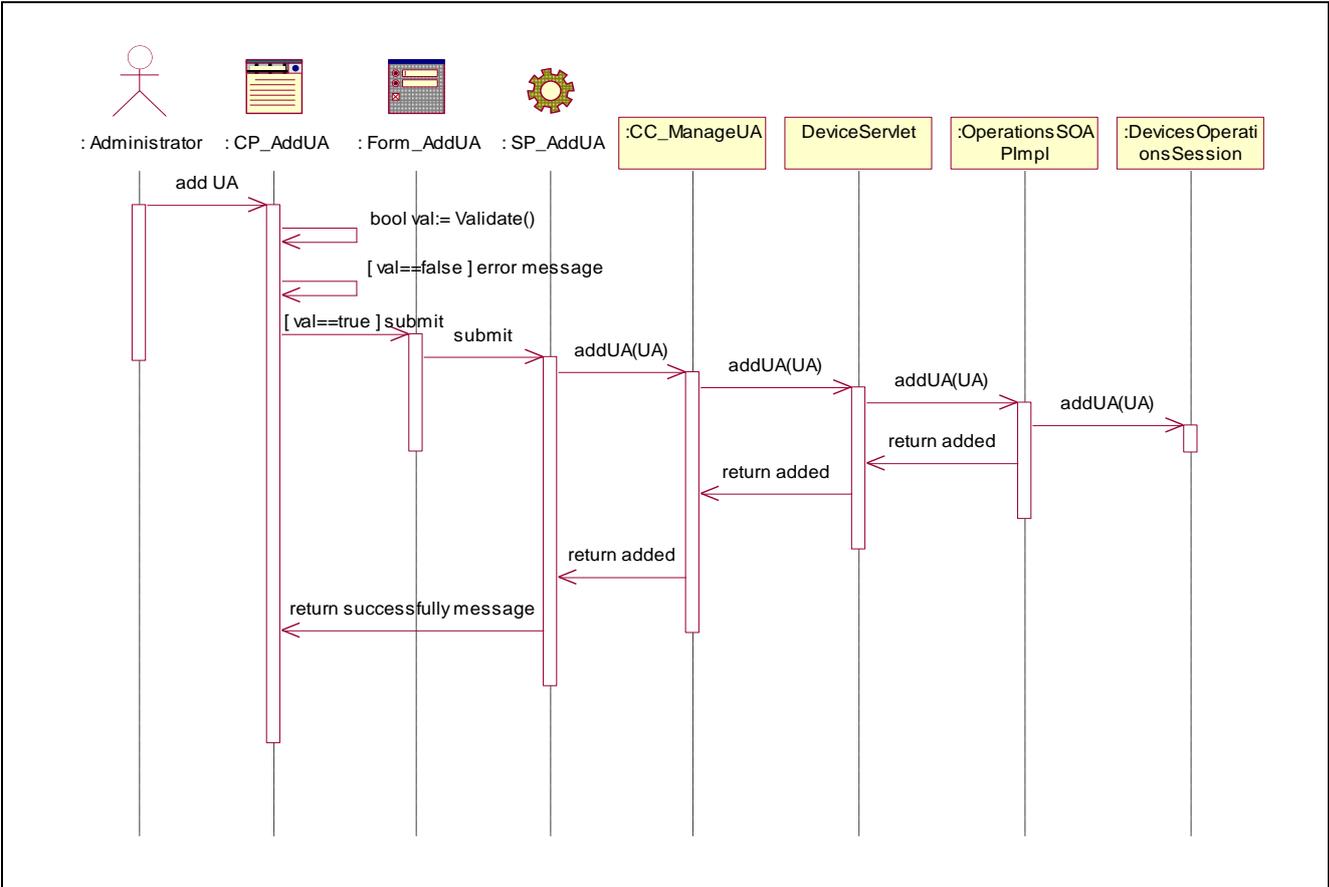


Figura 49. Diagrama de Secuencia del Diseño. CU Manage User Agent (escenario Add User Agent)

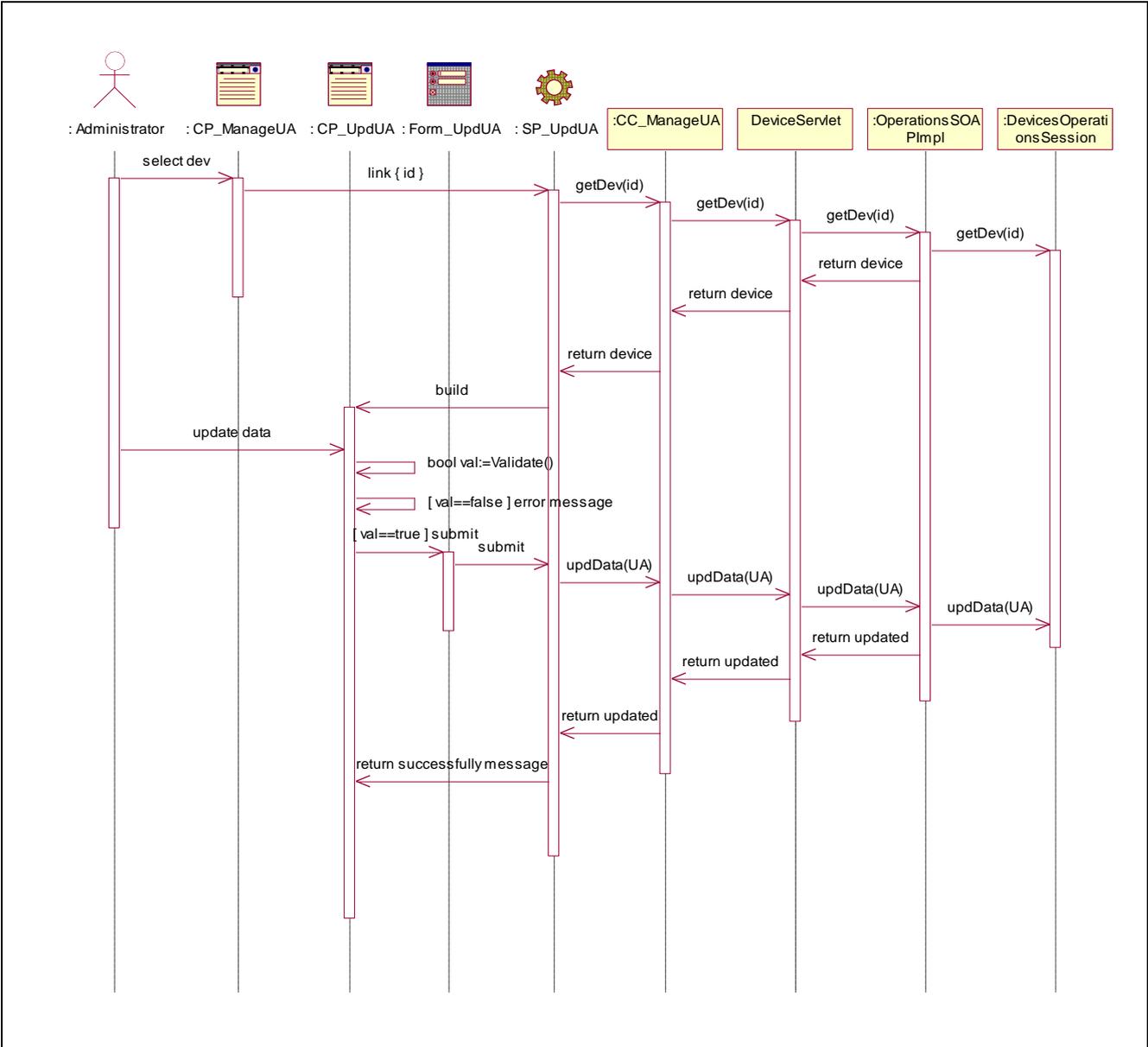


Figura 50. Diagrama de Secuencia del Diseño. CU Manage User Agent (escenario Update User Agent)

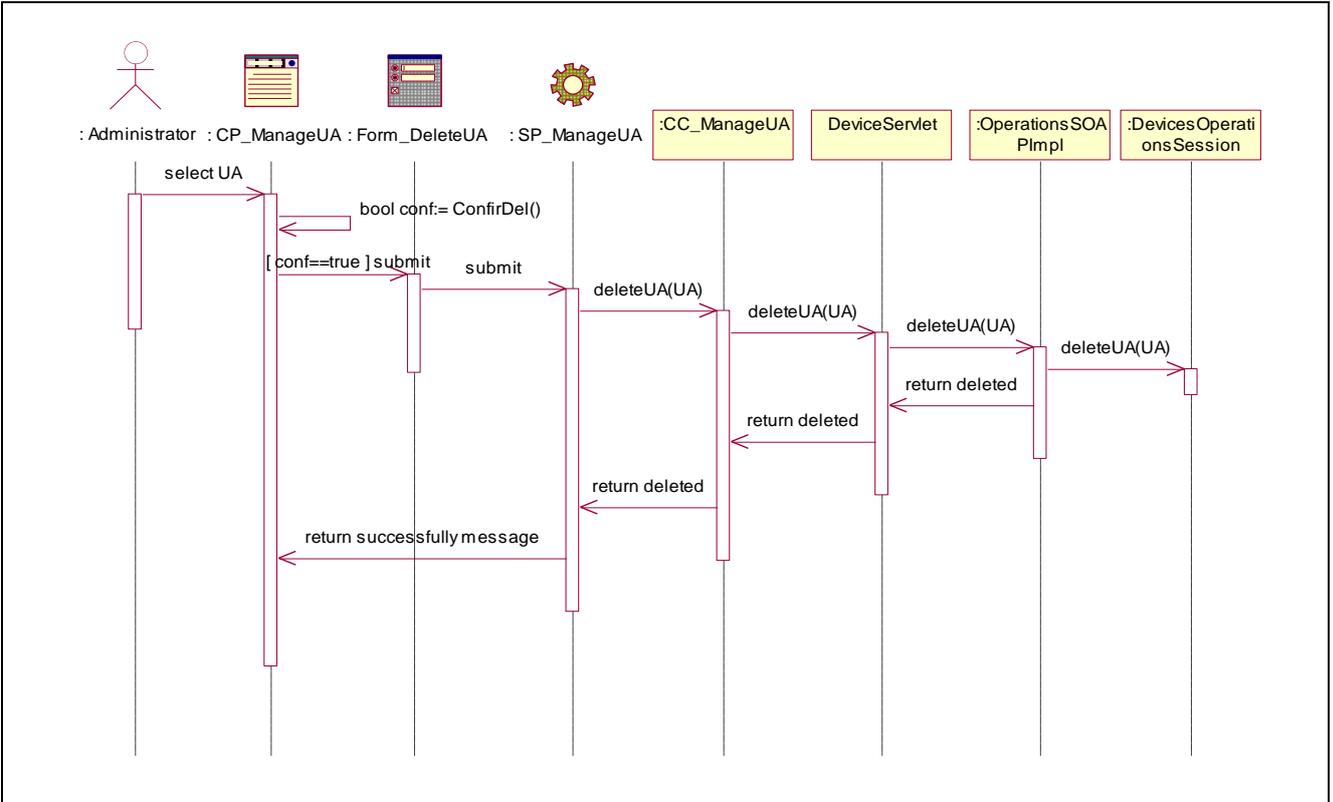


Figura 51. Diagrama de Secuencia del Diseño. CU Manage User Agent (escenario Delete User Agent)

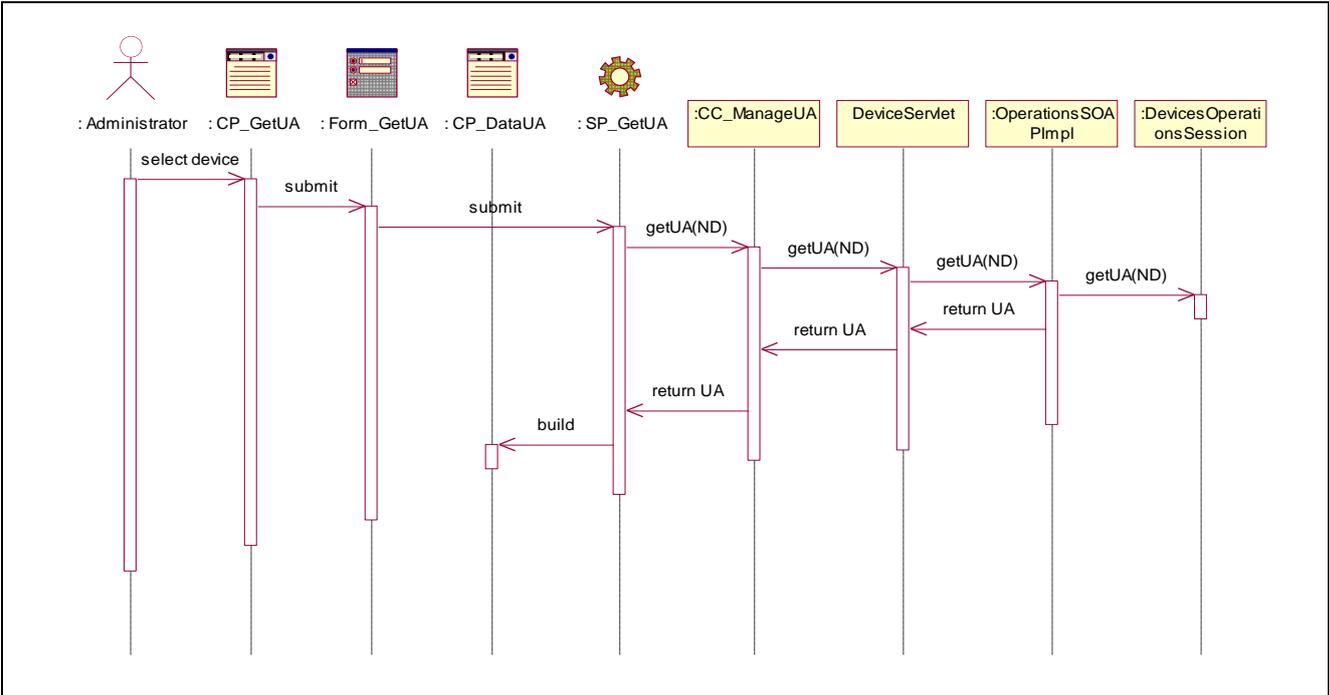


Figura 52. Diagrama de Secuencia del Diseño. CU Get User Agent

GLOSARIO

API (*Application Programming Interface*): Es un conjunto de funciones que facilitan el intercambio de mensajes o datos entre dos aplicaciones.

Capability: Características que posee un dispositivo móvil.

CC/PP (*Composite Capabilities/Preferences Profile*): Es una descripción de las capacidades de los dispositivos y las preferencias de los usuarios.

Devices Manager: Proyecto que pertenece a la plataforma que se encuentra desarrollando la empresa Procyon.

EJB (*Enterprise Java Beans*): Es un API que forma parte del estándar de construcción de aplicaciones empresariales J2EE de Sun Microsystems que proporciona un modelo de componentes distribuido estándar del lado del servidor.

IDE (*Integrated Development Environment*): Es un programa compuesto por un conjunto de herramientas para un programador. Puede dedicarse en exclusiva a un sólo lenguaje de programación o bien, poder utilizarse para varios.

J2EE (*Java 2 Enterprise Edition*): Es una plataforma de programación para desarrollar y ejecutar software de aplicaciones en lenguaje de programación Java.

JSP (*JavaServer Pages*): Es una tecnología Java que permite generar contenido dinámico para web, en forma de documentos HTML, XML o de otro tipo.

Polish: Base de datos que contiene información de diversos dispositivos móviles que puede ser consultada en <http://devices.i2mepolish.org/interactivedb/welcome.faces> .

Procyon: Empresa de desarrollo de software creada en el año 2004, cuyo sitio web es www.procyonsoluciones.com .

Ringtone: Melodía en el celular para personalizar las llamadas entrantes.

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.

SMS (*Short Message Service*): Es un sistema de mensajes de texto para teléfonos móviles de los operadores de telefonía móvil.

UAProf: Base de datos que contiene información de los dispositivos móviles que puede ser consultada en www.UAProfile.com .

UML (*Unified Modeling Language*): Es un lenguaje de modelado gráfico que tiene un conjunto de herramientas, que permite modelar, analizar y diseñar sistemas orientados a objetos.

User Agent: Es una cadena que envía el dispositivo móvil para identificarse cuando hace alguna solicitud.

WSDL2java: Herramienta muy útil de Axis que ayuda a simplificar el acceso a los servicios web.

WURFL: Base de datos que contiene información de los dispositivos móviles que puede ser consultada en <http://wurfl.sourceforge.net/> .

WAP (*Wireless Application Protocol*): Es un estándar seguro que permite que los usuarios accedan a información de forma instantánea a través de dispositivos inalámbricos.

Mobistore: Plataforma que se encuentra desarrollando la empresa Procyon.