

# UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

## Facultad 9



*Título: Informe del rol de analista del subsistema de modelos y propiedades de un simulador para la industria química.*

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

**Autor:** Adalberto Velázquez Batista

**Tutor:** Ing. Juan Carlos Suárez López.

***Ciudad de la Habana, Mayo de 2009***

***“Año del 50 Aniversario del Triunfo de la Revolución Cubana”***

## *Agradecimientos*

*A mi mamá y mi papá porque a ellos les debo todo lo que soy*

*A mi hermano y mi hermana por ayudarme y preocuparse por mí durante todo este tiempo*

*A lisy.....*

*A la fea de Sandra por su apoyo, ayuda y su cariño que tanto aprecio (que nadie piense que puse lo de fea de cariño, de verdad que es fea jajajaja)*

*A mis hijos Raúl y Bárbaro y a Keilan por su amistad*

*A la revolución por darme la oportunidad de estudiar esta carrera*

*A todos los que aportaron un grano de arena en mi formación*

*A mi tutor por su ayuda y colaboración en el desarrollo de este trabajo*

## *Dedicatoria*

*A mi abuelo que la vida no le dejo verme ingeniero*

*A mi mamá y mi papá por ser lo más grande que tengo*

*A mi hermano y mi hermana que tanto quiero*

*A Lisy por ser lo que más he querido*

*A Maribel y Richar por ser como mis padres*

*A Dele y Andito que aunque ellos no sepan nada de esto, los quiero mucho*

*A toda mi familia, mis amistades y todas las personas que quiero*

# Resumen

El presente trabajo muestra la realización del análisis y diseño de un subsistema de modelos matemáticos, componentes químicos y propiedades físico-químicas para un Simulador de Procesos Químicos. En él se presentan los pasos seguidos para garantizar la obtención de un diseño que cumpla con los requerimientos planteados para el subsistema.

Para diseñar el subsistema, se utiliza la metodología de desarrollo RUP, empleando como lenguaje de modelado UML y como herramienta CASE la Suite de Rational. La elaboración del diseño se realiza teniendo en cuenta un conjunto de patrones que permiten obtener un producto altamente reusable. Se definen las clases fundamentales que permitirán que el subsistema propuesto cumpla las funcionalidades requeridas, y se describen los atributos y métodos que deben tener estas clases, para brindar a los desarrolladores la información necesaria para que desarrollen un buen trabajo.

# Índice de Contenido

<b>INTRODUCCIÓN .....</b>	<b>1</b>
<b>CAPÍTULO 1 FUNDAMENTACIÓN TEÓRICA.....</b>	<b>5</b>
1.1 INTRODUCCIÓN .....	5
1.2 SIMULACIÓN DE PROCESOS QUÍMICOS.....	5
1.2.1 <i>Definición de Procesos Químicos.....</i>	<i>5</i>
1.2.2 <i>Definición de Simulación.....</i>	<i>5</i>
1.2.3 <i>Simulación de Procesos Químicos.....</i>	<i>6</i>
1.3 MODELOS MATEMÁTICOS, COMPONENTES QUÍMICOS Y PROPIEDADES FÍSICO-QUÍMICAS .....	6
1.4 SIMULADORES DE PROCESOS QUÍMICOS.....	8
1.4.1 <i>Simuladores de Procesos Químicos Internacionales.....</i>	<i>9</i>
1.4.2 <i>Simuladores de Procesos Químicos en Cuba.....</i>	<i>13</i>
1.5 METODOLOGÍAS DE DESARROLLO DE SOFTWARE.....	14
1.5.1 <i>Proceso Unificado de Desarrollo de Software(RUP).....</i>	<i>15</i>
1.5.2 <i>Programación Extrema (XP).....</i>	<i>16</i>
1.5.3 <i>Selección.....</i>	<i>17</i>
1.6 HERRAMIENTAS CASE.....	17
1.6.1 <i>Rational Rose Enterprise.....</i>	<i>18</i>
1.6.2 <i>Visual Paradigm.....</i>	<i>19</i>
1.6.3 <i>Selección.....</i>	<i>20</i>
1.7 CONCLUSIONES .....	20
<b>CAPÍTULO 2 ANÁLISIS DEL SISTEMA.....</b>	<b>21</b>
2.1 INTRODUCCIÓN .....	21
2.2 SITUACIÓN PROBLEMÁTICA.....	21
2.3 MODELO DE DOMINIO.....	21
2.3.1 <i>¿Por qué modelo de dominio?.....</i>	<i>22</i>
2.3.2 <i>Conceptos del Modelo del Dominio.....</i>	<i>22</i>
2.3.3 <i>Diagrama del Modelo del Dominio.....</i>	<i>23</i>
2.3.4 <i>Descripción del Modelo del Dominio.....</i>	<i>24</i>
2.4 DESCRIPCIÓN DEL SUBSISTEMA PROPUESTO .....	25
2.5 REQUERIMIENTOS DEL SUBSISTEMA.....	25
2.5.1 <i>Requerimientos funcionales.....</i>	<i>25</i>
2.5.2 <i>Requerimientos no funcionales.....</i>	<i>27</i>
2.5.3 <i>Modelo de Casos de Uso del Subsistema.....</i>	<i>29</i>

2.5.3.1	Actores.....	29
2.5.3.2	Casos de Uso del Subsistema.....	30
2.5.4	<i>Diagrama de Casos de Uso del Subsistema</i> .....	33
2.6	DESCRIPCIÓN DE LOS CU .....	34
2.7	CONCLUSIONES .....	45
<b>CAPÍTULO 3 ANÁLISIS Y DISEÑO.....</b>		<b>46</b>
3.1	INTRODUCCIÓN .....	46
3.2	ANÁLISIS .....	46
3.2.1	DIAGRAMA DE CLASES DEL ANÁLISIS .....	46
3.3	DISEÑO.....	52
3.3.1	<i>Patrones de diseño</i> .....	53
3.3.2	<i>Diagramas de clases del diseño</i> .....	55
3.3.3	<i>Diagramas de iteración del diseño</i> .....	65
3.4	CONCLUSIONES .....	65
<b>CONCLUSIONES GENERALES .....</b>		<b>66</b>
<b>RECOMENDACIONES .....</b>		<b>67</b>
<b>TRABAJOS CITADOS .....</b>		<b>68</b>
<b>BIBLIOGRAFÍA.....</b>		<b>70</b>
<b>ANEXO 1. DESCRIPCIONES TEXTUALES DE CASOS DE USOS.....</b>		<b>73</b>
<b>ANEXO 2. DIAGRAMAS DE CLASES DEL ANÁLISIS .....</b>		<b>82</b>
<b>ANEXO 3. DIAGRAMAS DE CLASES DEL DISEÑO .....</b>		<b>85</b>
<b>ANEXO 4. DIAGRAMAS DE INTERACCIÓN (SECUENCIA).....</b>		<b>90</b>

# Índice de Figuras

FIGURA 1 DIAGRAMA 1 MODELO DE DOMINIO .....	24
FIGURA 2 MODELO DE CASOS DE USO DEL SISTEMA .....	33
FIGURA 3 DIAGRAMA DE CLASES DEL ANÁLISIS CASO DE USO: BUSCAR COMPONENTE .....	47
FIGURA 4 DIAGRAMA DE CLASES DEL ANÁLISIS CASO DE USO: GESTIONAR PAQUETES DE PROPIEDADES .....	47
FIGURA 5 DIAGRAMA DE CLASES DEL ANÁLISIS CASO DE USO: GESTIONAR UNIDADES DE OPERACIÓN.....	48
FIGURA 6 DIAGRAMA DE CLASES DEL ANÁLISIS CASO DE USO: GESTIONAR CORRIENTE.....	49
FIGURA 7 DIAGRAMA DE CLASES DEL ANÁLISIS CASO DE USO: GESTIONAR CORRIENTE DE ENTRADA.....	50
FIGURA 8 DIAGRAMA DE CLASES DEL ANÁLISIS CASO DE USO: CALCULAR PROPIEDADES.....	50
FIGURA 9 DIAGRAMA DE CLASES DEL ANÁLISIS CASO DE USO: GUARDAR RESULTADOS .....	51
FIGURA 10 DIAGRAMA DE CLASES DEL ANÁLISIS CASO DE USO: SIMULAR.....	51
FIGURA 11 DIAGRAMA DE CLASES DEL ANÁLISIS CASO DE USO: GESTIONAR COMPONENTES DEL PROCESO .....	52
FIGURA 12 DIAGRAMA DE CLASES DEL ANÁLISIS CASO DE USO: GESTIONAR RESULTADO.....	52
FIGURA 13 DIAGRAMA DE CLASES DEL DISEÑO CASO DE USO: BUSCAR COMPONENTE .....	56
FIGURA 14 DIAGRAMA DE CLASES DEL DISEÑO CASO DE USO: GESTIONAR PAQUETE PROPIEDADES.....	57
FIGURA 15 DIAGRAMA DE CLASES DEL DISEÑO CASO DE USO: GESTIONAR UNIDADES OPERACIÓN.....	58
FIGURA 16 DIAGRAMA DE CLASES DEL DISEÑO CASO DE USO: GESTIONAR CORRIENTE.....	59
FIGURA 17 DIAGRAMA DE CLASES DEL DISEÑO CASO DE USO: GESTIONAR CORRIENTE ENTRADA.....	60
FIGURA 18 DIAGRAMA DE CLASES DEL DISEÑO CASO DE USO: CALCULAR PROPIEDADES.....	61
FIGURA 19 DIAGRAMA DE CLASES DEL DISEÑO CASO DE USO: GUARDAR RESULTADOS .....	62
FIGURA 20 DIAGRAMA DE CLASES DEL DISEÑO CASO DE USO: SIMULAR.....	63
FIGURA 21 DIAGRAMA DE CLASES DEL DISEÑO CASO DE USO: GESTIONAR COMPONENTES DEL PROCESO.....	64
FIGURA 22 DIAGRAMA DE CLASES DEL DISEÑO CASO DE USO: GESTIONAR RESULTADO.....	64
FIGURA 23 DIAGRAMA DE CLASES DEL ANÁLISIS CASO DE USO: GESTIONAR COMPONENTE .....	82
FIGURA 24 DIAGRAMA DE CLASES DEL ANÁLISIS CASO DE USO: GESTIONAR BIBLIOTECA .....	82
FIGURA 25 DIAGRAMA DE CLASES DEL ANÁLISIS CASO DE USO: GESTIONAR PROCESO .....	83
FIGURA 26 DIAGRAMA DE CLASES DEL ANÁLISIS CASO DE USO: GESTIONAR PROPIEDADES .....	83
FIGURA 27 DIAGRAMA DE CLASES DEL ANÁLISIS CASO DE USO: MANIPULAR UNIDADES DE MEDIDA .....	84
FIGURA 28 DIAGRAMA DE CLASES DEL DISEÑO CASO DE USO: GESTIONAR COMPONENTE .....	85
FIGURA 29 DIAGRAMA DE CLASES DEL DISEÑO CASO DE USO: GESTIONAR BIBLIOTECA .....	86
FIGURA 30 DIAGRAMA DE CLASES DEL DISEÑO CASO DE USO: GESTIONAR PROCESO .....	87
FIGURA 31 DIAGRAMA DE CLASES DEL DISEÑO CASO DE USO: GESTIONAR PROPIEDADES .....	88
FIGURA 32 DIAGRAMA DE CLASES DEL DISEÑO CASO DE USO: MANIPULAR UNIDADES MEDIDA .....	89
FIGURA 33 DIAGRAMA DE SECUENCIA CASO DE USO: BUSCAR COMPONENTE .....	90
FIGURA 34 DIAGRAMA DE SECUENCIA CASO DE USO: GESTIONAR PAQUETE PROPIEDADES (ESCENARIO ASIGNAR PAQUETE) .....	90
FIGURA 35 DIAGRAMA DE SECUENCIA CASO DE USO: GESTIONAR PAQUETE PROPIEDADES (ESCENARIO LISTAR PAQUETES).....	91

FIGURA 36 DIAGRAMA DE SECUENCIA CASO DE USO: GESTIONAR UNIDAD DE OPERACIÓN (ESCENARIO ADICIONAR UNIDAD AL PROCESO) .....91

FIGURA 37 DIAGRAMA DE SECUENCIA CASO DE USO: GESTIONAR UNIDADES DE OPERACIÓN (ESCENARIO ELIMINAR UNIDAD DEL PROCESO).....92

FIGURA 38 DIAGRAMA DE SECUENCIA CASO DE USO: GESTIONAR UNIDADES DE OPERACIÓN (ESCENARIO LISTAR UNIDADES).....92

FIGURA 39 DIAGRAMA DE SECUENCIA CASO DE USO: GESTIONAR UNIDADES DE OPERACIÓN (ESCENARIO MODIFICAR PROPIEDADES).....93

FIGURA 40 DIAGRAMA DE SECUENCIA CASO DE USO: GESTIONAR UNIDADES DE OPERACIÓN (ESCENARIO OBTENER ESTADO) .....93

FIGURA 41 DIAGRAMA DE SECUENCIA CASO DE USO: GESTIONAR CORRIENTE (ESCENARIO CONECTAR CORRIENTE) .....94

FIGURA 42 DIAGRAMA DE SECUENCIA CASO DE USO: GESTIONAR CORRIENTE (ESCENARIO CREAR CORRIENTE) .....95

FIGURA 43 DIAGRAMA DE SECUENCIA CASO DE USO: GESTIONAR CORRIENTE (ESCENARIO ELIMINAR CORRIENTE).....95

FIGURA 44 DIAGRAMA DE SECUENCIA CASO DE USO: GESTIONAR CORRIENTE (ESCENARIO LISTAR CORRIENTES) .....95

FIGURA 45 DIAGRAMA DE SECUENCIA CASO DE USO: GESTIONAR CORRIENTE (ESCENARIO OBTENER PROPIEDADES DE CORRIENTE).....96

FIGURA 46 DIAGRAMA DE SECUENCIA CASO DE USO: GESTIONAR CORRIENTE DE ENTRADA (ESCENARIO MODIFICAR COMPOSICIÓN).....96

FIGURA 47 DIAGRAMA DE SECUENCIA CASO DE USO: GESTIONAR CORRIENTE DE ENTRADA (ESCENARIO MODIFICAR PROPIEDADES).....97

FIGURA 48 DIAGRAMA DE SECUENCIA CASO DE USO: CALCULAR PROPIEDADES.....97

FIGURA 49 DIAGRAMA DE SECUENCIA CASO DE USO: GUARDAR RESULTADOS .....98

FIGURA 50 DIAGRAMA DE SECUENCIA CASO DE USO: SIMULAR.....98

FIGURA 51 DIAGRAMA DE SECUENCIA CASO DE USO: GESTIONAR COMPONENTES DEL PROCESO (ESCENARIO ADICIONAR COMPONENTE AL PROCESO) ..99

FIGURA 52 DIAGRAMA DE SECUENCIA CASO DE USO: GESTIONAR COMPONENTES DEL PROCESO (ESCENARIO ELIMINAR COMPONENTE DEL PROCESO)...99

FIGURA 53 DIAGRAMA DE SECUENCIA CASO DE USO: GESTIONAR RESULTADO (ESCENARIO CARGAR RESULTADOS) .....100

FIGURA 54 DIAGRAMA DE SECUENCIA CASO DE USO: GESTIONAR RESULTADO (ESCENARIO ELIMINAR RESULTADO) .....100

FIGURA 55 DIAGRAMA DE SECUENCIA CASO DE USO: GESTIONAR RESULTADO (ESCENARIO LISTAR RESULTADOS).....101

FIGURA 56 DIAGRAMA DE SECUENCIA CASO DE USO: GESTIONAR RESULTADO (ESCENARIO OBTENER RESULTADO) .....101

FIGURA 57 DIAGRAMA DE SECUENCIA CASO DE USO: GESTIONAR COMPONENTE (ESCENARIO CARGAR COMPONENTE) .....102

FIGURA 58 DIAGRAMA DE SECUENCIA CASO DE USO: GESTIONAR COMPONENTE (ESCENARIO CLONAR COMPONENTE) .....102

FIGURA 59 DIAGRAMA DE SECUENCIA CASO DE USO: GESTIONAR COMPONENTE (ESCENARIO ELIMINAR COMPONENTE PERSONALIZADO) .....103

FIGURA 60 DIAGRAMA DE SECUENCIA CASO DE USO: GESTIONAR COMPONENTE (ESCENARIO NUEVO COMPONENTE) .....103

FIGURA 61 DIAGRAMA DE SECUENCIA CASO DE USO: GESTIONAR BIBLIOTECA (ESCENARIO ADICIONAR COMPONENTE A LA BIBLIOTECA) .....104

FIGURA 62 DIAGRAMA DE SECUENCIA CASO DE USO: GESTIONAR BIBLIOTECA (ESCENARIO CARGAR BIBLIOTECAS) .....104

FIGURA 63 DIAGRAMA DE SECUENCIA CASO DE USO: GESTIONAR BIBLIOTECA (ESCENARIO CREAR BIBLIOTECA) .....105

FIGURA 64 DIAGRAMA DE SECUENCIA CASO DE USO: GESTIONAR BIBLIOTECA (ESCENARIO ELIMINAR BIBLIOTECA).....105

FIGURA 65 DIAGRAMA DE SECUENCIA CASO DE USO: GESTIONAR BIBLIOTECA (ESCENARIO ELIMINAR COMPONENTE DE LA BIBLIOTECA).....106

FIGURA 66 DIAGRAMA DE SECUENCIA CASO DE USO: GESTIONAR BIBLIOTECA (ESCENARIO LISTAR BIBLIOTECAS).....106

FIGURA 67 DIAGRAMA DE SECUENCIA CASO DE USO: GESTIONAR BIBLIOTECA (ESCENARIO LISTAR COMPONENTES DE UNA BIBLIOTECA).....107

FIGURA 68 DIAGRAMA DE SECUENCIA CASO DE USO: GESTIONAR PROCESO (ESCENARIO ABRIR PROCESO).....107

FIGURA 69 DIAGRAMA DE SECUENCIA CASO DE USO: GESTIONAR PROCESO (ESCENARIO NUEVO PROCESO).....108

FIGURA 70 DIAGRAMA DE SECUENCIA CASO DE USO: GESTIONAR PROCESO (ESCENARIO SALVAR PROCESO) .....108

FIGURA 71 DIAGRAMA DE SECUENCIA CASO DE USO: GESTIONAR PROPIEDADES (ESCENARIO ESTIMAR PROPIEDADES) .....109

FIGURA 72 DIAGRAMA DE SECUENCIA CASO DE USO: GESTIONAR PROPIEDADES (ESCENARIO MODIFICAR PROPIEDADES) .....109

FIGURA 73 DIAGRAMA DE SECUENCIA CASO DE USO: MANIPULAR UNIDADES DE MEDIDA (ESCENARIO MANIPULAR UM COMPONENTE).....110

FIGURA 74 DIAGRAMA DE SECUENCIA CASO DE USO: MANIPULAR UNIDADES DE MEDIDA (ESCENARIO MANIPULAR UM CORRIENTE) .....110

FIGURA 75 DIAGRAMA DE SECUENCIA CASO DE USO: MANIPULAR UNIDADES DE MEDIDA (ESCENARIO MANIPULAR UM UNIDAD ASOCIAD .....111

# Índice de Tablas

TABLA 1 ACTORES.....	29
TABLA 2 CASOS DE USO DEL SUBSISTEMA.....	30
TABLA 3 CASO DE USO: BUSCAR COMPONENTE.....	34
TABLA 4 CASO DE USO: GESTIONAR PAQUETES DE PROPIEDADES.....	34
TABLA 5 CASO DE USO: GESTIONAR UNIDAD DE OPERACIÓN .....	35
TABLA 6 CASO DE USO: GESTIONAR CORRIENTE .....	37
TABLA 7 CASO DE USO: GESTIONAR CORRIENTE DE ENTRADA .....	38
TABLA 8 CASO DE USO: CALCULAR PROPIEDADES .....	40
TABLA 9 CASO DE USO: GUARDAR RESULTADOS.....	40
TABLA 10 CASO DE USO: SIMULAR .....	41
TABLA 11 CASO DE USO: GESTIONAR COMPONENTES DEL PROCESO .....	42
TABLA 12 CASO DE USO: GESTIONAR RESULTADO .....	43
TABLA 134 CASO DE USO: GESTIONAR COMPONENTE.....	73
TABLA 145 CASO DE USO: GESTIONAR BIBLIOTECA.....	75
TABLA 156 CASO DE USO: GESTIONAR PROCESO.....	77
TABLA 167 CASO DE USO: GESTIONAR PROPIEDADES.....	79
TABLA 178 CASO DE USO: MANIPULAR UNIDADES DE MEDIDA.....	80

# Introducción

Los simuladores han surgido como una herramienta efectiva para el entrenamiento y prueba de las más diversas ramas de la actividad humana [1]. El análisis de procesos a través de las herramientas de simulación juega un papel importante en el incremento de la eficiencia técnico-económica de las industrias. Debido a las potencialidades que brindan los simuladores para el estudio de diversas situaciones problemáticas, como pueden ser la toma de decisiones, solución de problemas y el ahorro en el campo económico de posibles inversiones fallidas o innecesarias, estos han experimentado un creciente auge en las últimas décadas, revolucionando los antiguos métodos y paradigmas seguidos hasta el momento por los diversos centros industriales, entre estos se encuentran los centros de procesamiento de productos químicos, los cuales no han quedado desvinculados al desarrollo de estos software.

Conocidas las posibilidades y ventajas del uso de estas aplicaciones para la simulación de procesos, en especial su incidencia en la industria química, desde la década de los 60 algunas empresas han venido desarrollando aplicaciones informáticas con este objetivo, dentro de las cuales se encuentran la Compañía Petrolera Shell con el CHEOPS, la Compañía Petrolera MOBIL con el PEDLAN, la Dow Chemical con el POWERFACTS, y junto a ellas otras muchas compañías y corporaciones, que dieron lugar a un sin número de simuladores de procesos químicos como son ASPEN PLUS, CHEMCAD, COCO simulator, HYSYS, GEMCS, SUGARS, OpenModelica, Dymola entre otros.

Cuba no quedó atrás en el desarrollo de estos sistemas, enfocándose principalmente en la simulación de procesos químicos para las industrias azucareras y alcoholeras, por lo cual se han desarrollado las más diversas bases para la simulación de procesos, entre los que se encuentran TERMOAZUCAR, ACOPLA, SIMFAD, SIDEL y AGE, pero estos software se caracterizan por desarrollar una simulación poco flexible y carecer de facilidades para su uso y manejo, así como no posibilitar un análisis detallado de los resultados del proceso de simulación, lo que los ubica en un escalón inferior a los simuladores que se utilizan actualmente en esta rama a nivel internacional. Por estas razones en la Universidad de Ciencias Informáticas (UCI) se desarrolló un simulador con el fin de obtener un producto que no presentara las limitaciones de sus similares desarrollados en Cuba, pero el mismo presenta como características

desfavorables el estar desarrollado para propósitos específicos y no posibilitar un soporte general a la simulación de procesos, por lo que se hace necesario realizar el análisis y diseño para un simulador que permita un soporte general a la simulación de procesos en la industria química y pueda ubicarse a la altura de sus similares extranjeros.

Como **problema científico** en el presente trabajo se define, la inexistencia del análisis y diseño de los componentes que garanticen una adecuada implementación del subsistema de modelos matemáticos, componentes químicos y propiedades físico-químicas en el desarrollo de un simulador para la industria química.

En correspondencia con el problema científico planteado, se define como **objeto de estudio** el análisis y diseño de un simulador de procesos para la industria química.

Delimitando el **campo de acción** al análisis y diseño de los paquetes de modelos matemáticos, componentes químicos y propiedades físico-químicos de un simulador para la industria química.

Por tanto la investigación se plantea como **objetivo general** desarrollar un correcto análisis y diseño del subsistema de modelos matemáticos, componentes químicos y propiedades físico-químico, que garantice la adecuada implementación de este subsistema para un simulador de procesos, para la industria química.

Para lograr el cumplimiento del objetivo general se proponen los siguientes **objetivos específicos**:

- Realizar una correcta captura de los requerimientos funcionales y no funcionales del subsistema a desarrollar.
- Desarrollar un correcto análisis del subsistema de modelos matemáticos, componentes químicos y propiedades físico-químico.
- Desarrollar un correcto diseño del subsistema de modelos matemáticos, componentes químicos y propiedades físico-químico.

Como **idea a defender** se presenta que una realización correcta del Análisis y Diseño del subsistema de modelos matemáticos, componentes químicos y propiedades físico-químico, garantizará la adecuada

implementación de estos subsistemas para el desarrollo de un simulador de procesos en la industria química.

Para cumplir con los objetivos y resolver el problema planteado se proponen las siguientes **tareas de investigación**:

- Caracterizar los principales simuladores existentes respecto a su concepción y métodos del proceso de simulación.
- Caracterizar la metodología y herramientas a utilizar en el desarrollo Análisis y Diseño del subsistema de modelos matemáticos, componentes químicos y propiedades físico-químico
- Determinación de los requerimientos funcionales y no funcionales del subsistema.
- Realizar el análisis del subsistema de modelos matemáticos, componentes químicos y propiedades físico-químico.
- Realizar el diseño del subsistema de modelos matemáticos, componentes químicos y propiedades físico-químico.

Estructuración del contenido por capítulos:

Capítulo 1: Fundamentación teórica, se definen los principales conceptos relacionados con el objeto de estudio así como se describen las principales herramientas y metodologías utilizadas durante la investigación.

Capítulo 2: Análisis del Sistema, se determina y describe un modelo para simular los procesos en la industria química, así como el modelo de dominio, los requerimientos funcionales, no funcionales y se presenta el diagrama de casos de usos del sistema.

Capítulo 3: Análisis y diseño, se describe el análisis y diseño de la aplicación, se presentan los diagramas que lo caracteriza así como los patrones de diseño que se utilizan.

En el presente trabajo se utilizan un conjunto de **Métodos Científicos de Investigación** los cuales representan los procedimientos que posibilitan estudiar la realidad, la naturaleza, la sociedad y el

pensamiento, con el propósito de descubrir su esencia y sus relaciones. Los métodos científicos se pueden clasificar en **teóricos** y **empíricos**. De estos métodos se pusieron en práctica los que se muestran a continuación:

Teóricos:

- **Analítico Sintético:** Este método se utiliza para realizar una investigación de todo lo relacionado con la simulación de procesos químicos, permitiendo analizar documentación existente relacionada al campo de acción, así como en la extracción de los elementos más importantes.
- **Histórico Lógico:** Permite analizar la evolución y desarrollo de los simuladores de procesos químicos desde su creación hasta el día de hoy, posibilitando ubicar este trabajo en el espacio y marco actual de las técnicas que se utilizan para la simulación de procesos químicos.
- **Modelación:** Permite la creación de modelos, los cuales representan una vista simplificada de los procesos y posibilitan descubrir nuevas relaciones y cualidades del objeto de estudio.

Empíricos:

- **Entrevista:** Este método es utilizado fundamentalmente para obtener información acerca de los requerimientos que debe cumplir el software, así como información necesaria para el desarrollo del trabajo.

# Capítulo

# 1

## Fundamentación Teórica

### 1.1 Introducción

En este capítulo se explican los principales conceptos y definiciones relacionados con el tema de los simuladores de procesos químicos, tales como son Proceso Químico, Simulación y Simulador de Procesos Químicos de forma general. También se realiza una caracterización de los principales simuladores de procesos químicos desarrollados en Cuba y el mundo, así como un análisis de las principales metodologías de desarrollo de software y herramientas CASE (Computer Aided Software Engineering, Ingeniería de Software Asistida por Ordenador) existentes en la actualidad.

### 1.2 Simulación de Procesos Químicos

#### 1.2.1 Definición de Procesos Químicos

Un proceso químico es un conjunto de operaciones químicas y/o físicas ordenadas a la transformación de unas materias iniciales en productos finales diferentes [2]. Esto no es más que dado un conjunto de productos o sustancias iniciales obtener después de llevar a cabo diversas reacciones químicas y/o cambios físicos un conjunto de productos o sustancias que se diferencian del conjunto inicial en su composición o se encuentren en un estado que no sea en el que se encontraban antes del comienzo de las reacciones químicas y/o cambios físicos.

#### 1.2.2 Definición de Simulación

Simulación es el proceso de diseñar y desarrollar un modelo computarizado de un sistema o proceso y conducir experimentos con este modelo con el propósito de entender el comportamiento del sistema o

evaluar varias estrategias con las cuales se puede operar el sistema [3], expresado de forma más sencillas, la simulación es la representación de un proceso o fenómeno mediante otro más simple, lo cual posibilita analizar sus características.

De aquí se puede definir al proceso o tarea de simular como aquella en la cual proponemos ciertos valores de entrada al simulador o programa de simulación para obtener ciertos resultados o valores de salida, tales que estiman el comportamiento del sistema real bajo esas condiciones. [4]

### **1.2.3 Simulación de Procesos Químicos**

La simulación de procesos se ha convertido en una de las más potentes herramientas de la ingeniería química, logrando aportar un mayor control de los procesos existentes y una constante mejora de los mismos, basada su esencia en modelos de propiedades físicas, modelos de dimensiones y operación de equipos, la simulación de procesos resuelve las variables desconocidas a partir de variables ya conocidas o parámetros de diseños especificados.

### **1.3 Modelos matemáticos, componentes químicos y propiedades físico-químicas**

Una de las partes fundamentales de un simulador de la industria química lo constituyen los modelos matemáticos, los componentes químicos y las propiedades físico-químicas, de los cuales, los modelos matemáticos constituyen la columna vertebral de los simuladores de procesos. La riqueza y efectividad de un simulador viene dada, entre otros aspectos, por la variedad, validez y alcance de su biblioteca de modelos, incluyendo las funciones de cálculo de las propiedades físicas de las corrientes [7]. Teniendo en cuenta que el proceso de simular no es más que la realización del diseño de un modelo matemático para un sistema, el cual debe ser capaz de representar al proceso real con la mayor exactitud posible, obtenemos que el resultado de la simulación será más valioso y más verídico, en cuanto el modelo matemático se adecue en mayor grado al proceso real en cuestión.

Los modelos matemáticos de procesos de forma integral están conformados por la unión de los modelos matemáticos correspondientes a los equipos presentes en el proceso a simular, que no son más que una representación del comportamiento del equipo, estos modelos pueden ser resueltos mediante diversos algoritmos de solución, entre los principales algoritmos disponibles para dar solución a estos modelos o sistemas de ecuaciones de simuladores de procesos se encuentran los siguientes:

- Método Modular Secuencial.
- Método Orientado a Ecuaciones.
- Método Modular Simultáneo.

**Secuencial Modular:**

Los cálculos son realizados unidad por unidad, en forma secuencial, los procesos con reciclo son descompuestos en varias iteraciones de cálculo hasta lograr convergencia. El elemento básico de este método es el modelo de operación unitaria, el cual es construido hasta obtener un conjunto de ecuaciones algebraicas no lineales, el cual debe ser compatible y determinado. Este método es utilizado por simuladores como: Aspen Plus, Chemcad, Hysys y PROII.

**Orientado a Ecuaciones:**

En este método todas las ecuaciones del modelo matemático general, algebraicas no lineales y diferenciales, son integradas en un único conjunto y son resueltas simultáneamente. El método Orientado a Ecuaciones es más flexible que el Secuencial-Modular aunque menos robusto, requiere mayor esfuerzo de programación y más recursos de almacenamiento. Esta arquitectura está más orientada a la solución de modelos dinámicos, este método es utilizado por simuladores como Aspen Dynamics y Simulink.

**Método Modular Simultáneo:**

El método Modular Simultáneo o Híbrido combina los métodos Secuencial Modular y Orientado a Ecuaciones. En este método los modelos rigurosos de las operaciones unitarias son resueltos secuencialmente, mientras que los modelos lineales son resueltos globalmente para interconectar los resultados de cada módulo.

Entre las características de los simuladores químicos se encuentran, el poseer un banco de modelos matemáticos de los equipos presentes en la simulación, con objeto de brindar facilidades entorno a escoger el modelo más adecuado para realizar la simulación, otras características son tener una lista de componentes químicos, ya sean puros o compuestos con sus respectivas propiedades físico-químicas, así como poseer un conjunto de modelos para la estimación de las propiedades físico-químicas en caso de

ser necesario para los componentes compuestos. En caso de realizar la simulación de un proceso, se realiza el diseño del mismo, lo que para el simulador no es más que el diseño de un modelo matemático, se seleccionan los componentes involucrados en el proceso, se evalúa el modelo matemático obtenido teniendo en cuenta las propiedades físico-químicas y los valores de inicialización asignados, obteniendo luego los resultados de la simulación.

#### 1.4 Simuladores de Procesos Químicos

Un simulador de procesos químicos es un programa con la capacidad de anticipar el estado y condiciones en que se encontrará un sistema en un determinado momento, teniendo en cuenta una serie de datos suministrados con anterioridad. Los principales simuladores comerciales como son ASPEN-PLUS, HYSYS, CHEMCAD, PRO/II son herramientas que proporcionan innumerables facilidades a la industria química en general, sin embargo éstos solo están disponibles a un alto costo.

Los simuladores son utilizados fundamentalmente para los siguientes objetivos:

- Estimar y realizar análisis de regresión de propiedades físicas.
- Analizar procesos donde los componentes deben ser mezclados, separados, calentados, enfriados y convertidos mediante operaciones básicas.
- Predecir el comportamiento de un proceso utilizando principios de la ingeniería química.
- Calcular balances de materia, energía, equilibrio químico y entre fases.
- Analizar el comportamiento de un sistema y realizar estudios de optimización y sensibilidad.
- Cambiar las condiciones de operación y analizar alternativas.
- Estimar costes.
- Generar salida gráficas o bien tablas de resultados.
- Entrenar operarios.
- Apoyar la docencia en las universidades.

Ventajas de la Simulación de Procesos

- Minimizar el tiempo necesario para realizar cálculos.
- Minimizar las fuentes de error humano, generando de este modelo resultados más precisos, que permiten al ingeniero ocuparse de otras tareas como el análisis de los datos, y modificación de

condiciones de operación que le permitan tener un conocimiento más profundo de lo que sucede en el proceso.

- Incrementar la eficiencia y eficacia del proceso.
- Organizar y ejecutar de forma más eficiente las evaluaciones de equipos, subprocesos y del proceso en su totalidad.
- Evaluar propuestas de inversiones ahorrando el dinero para financiarlas y a la vez crear mayores ganancias.
- Posibilidad de examinar rápidamente varias configuraciones para determinada planta.
- Analizar la factibilidad de los proyectos de investigación e innovación.
- Aumentar de forma significativa el conocimiento de los ingenieros de fábrica sobre el proceso, aspecto especialmente importante cuando estos son jóvenes.
- Reducción en el tiempo de diseño de una planta.
- Encontrar respuestas a situaciones determinadas y establecer las condiciones óptimas para el desarrollo del proceso a prueba.
- Observar el comportamiento de equipos bajo condiciones que podrían ser peligrosas.
- Ajustar puntos de control más cercanos a los límites reales del proceso.

#### 1.4.1 Simuladores de Procesos Químicos Internacionales



CHEMCAD nace en 1984 cuando un profesor universitario formó un equipo para desarrollar un simulador de procesos para computadoras personales (PC). El simulador fue vendido a la sección de software de McGraw Hill (COADE) y luego siguió siendo desarrollado y distribuido por Chemstations Inc.

CHEMCAD ha venido evolucionando durante estos años para convertirse en un paquete de módulos que abarca cálculo y diseño de intercambiadores de calor (CC-THERM), simulación de destilaciones dinámicas (CC-DCOLUMN), simulación de reactores por lotes (CC-ReACS), simulación de destilaciones por lotes (CC-BATCH), simulación de redes de tuberías (CC-SAFETY NET). Paquete de módulos que abarca:

- Cálculo y diseño de intercambiadores de calor.
- Simulación de destilaciones dinámicas.
- Simulación de reactores por lotes.
- Simulación de destilaciones por lotes.
- Simulación de redes de tuberías.

Características generales de CHEMCAD:

- Incluye bases de datos de componentes químicos, métodos termodinámicos y unidades de operación que permiten la simulación en estado estacionario de procesos químicos continuos desde escala laboratorio a escala industrial.
- Interfaz gráfica.
- Funcionamiento bajo Windows.
- Diagramas de Flujo y Reportes configurables, en MS Word o MS Excel.
- Poderosas capacidades de gráfico: curvas de destilación/absorción, diagramas de fases, diagramas presión-temperatura Vs. concentración, gráficas de predicción propiedades físico-químicas Vs. temperatura de corrientes de proceso y de sustancias puras, curvas de calor, ejes logarítmicos, exportación de datos a Excel.
- Simulación en régimen dinámico y de procesos discontinuos.
- Poderosa herramienta de Optimización de Procesos.
- Herramienta de Análisis de Sensibilidad que le permite probar escenarios sin la necesidad de modificar la simulación base e identificar situaciones críticas y óptimas.
- Predicción de hidratos.
- Reportes de Impacto Ambiental de corrientes de descarga según parámetros de la EPA, Carbono orgánico total, y Demanda Química de Oxígeno (DQO).
- Conexión con Visual Basic/ Excel que le permite programar sus propias operaciones unitarias dentro del diagrama de proceso, utilizando funciones termodinámicas y la base de datos de sustancias puras de CHEMCAD desde la programación en Excel.
- Convergencia de operaciones unitarias independientes del diagrama de proceso. Esta característica es excelente para un manejo más rápido y flexible de la convergencia de simulaciones grandes y complejas.



ASPEN PLUS, líder en el mercado del modelado de procesos químicos, Sistema Avanzado para Ingeniería de Procesos del inglés Advanced System for Process Engineering (ASPEN), fue desarrollado en los años 1970 por investigadores del Massachusetts Institute of Technology (MIT) y comercializado desde 1980 por la compañía AspenTech.

#### Características generales de ASPEN PLUS:

- Contiene la base de datos más amplia entre los simuladores de procesos comerciales, e incluye comportamiento de iones y de electrolitos.
- Flexible y fácil de usar.
- Regresión de datos experimentales.
- Diseño preliminar de los diagramas de flujo usando modelos de equipos simplificados.
- Balance de materia y energía riguroso, usando modelos de equipos detallados.
- Dimensionar piezas clave de los equipos.
- Optimización on-line de unidades de proceso completas o plantas.
- Optimizador incorporado.
- Enorme bases de datos de propiedades de sustancias, datos de modelos termodinámicos, incluyendo parámetros para muchas mezclas.
- Enorme cantidad de módulos adiciones disponibles (Polymer, Batch, Zyquad, Water, Plant).
- Integración progresiva en el estándar CAPE.
- Inclusión de resolución EO (Equation Oriented) en sus métodos de cálculo.
- Análisis de convergencia: rotura de ciclos, secuencia de soluciones.
- Incluye Fortran y Excel. Acceso a Visual Basic (Aspen Plus 2004.1).
- Análisis de sensibilidad: variaciones debidas a cambios introducidos.
- Caso de estudio para simular con varias entradas.
- Especificaciones de diseño para el cálculo automático de condiciones de operación y parámetros de equipos.
- Ajuste de datos.

- Optimización para maximizar rendimientos, consumo de energía, pureza.
- Línea de despliegue de modelos.



HYSYS es un software especializado para la industria petroquímica, fue adquirido por AspenTech en 2004 por lo que es desarrollado en la actualidad por Aspen Technology, considerado uno de los mejores programas de simulación de procesos químicos.

#### Características generales de HYSYS:

- Desarrollado para su uso en PCs.
- Integración progresiva en el estándar CAPE.
- Interfaz gráfica de fácil uso (orientado a objeto).
- Gran cantidad de operaciones disponibles. Numerosos parámetros disponibles.
- Paquetes de propiedades especiales.
- Base de datos extensa (superada solo por la de Aspen Plus).
- Utiliza datos experimentales para sus correlaciones.
- Enorme base de datos de modelos termodinámicos, incluyendo parámetros para muchas mezclas.
- Optimizador incorporado.
- Unidades de operación en estado estacionario o dinámico.
- Estructura para interpretar los comandos en forma interactiva.
- Capacidad de ejecutar todos los cálculos relacionados con las modificaciones de cualquier punto del proceso en forma bidireccional.
- Especifica expresiones para reacciones cinéticas.
- Posibilidad de adicionar código de programación con VISUAL BASIC.
- Extensión de la funcionalidad de la simulación en respuesta a posibles cambios.



Desarrollado por Simulation Sciences, es uno de los principales simuladores usados en la industria química actual.

#### Características generales de PRO/II:

- Entorno gráfico de simulación interactivo.
- Funciona bajo entorno Windows.
- Posibilidad de incluir código de programación en VISUAL BASIC.
- Posibilidad de realizar simulación en régimen dinámico (paquete PROTISS).
- Puede ser utilizado en: petróleo, refinería, petroquímica, polímeros, química fina y farmacéutica, construcciones, ingeniería, etc.
- Diseño de nuevos procesos.
- Evaluación de configuraciones alternas.
- Modernizar y renovar las instalaciones existentes.
- Evaluación del cumplimiento con las reglamentaciones ambientales.
- Solución de problemas y debottleneck planta de procesos.
- Monitoreo, optimización, mejora de los rendimientos y la rentabilidad de la planta
- Combina los recursos de datos de una gran biblioteca de componentes químicos y extensas propiedades termodinámicas.
- Transferencia bidireccional de variables aplicables.
- Creación de interfaces de operador personalizadas.
- Diseño sencillo de los procesos de destilación por lotes.

#### 1.4.2 Simuladores de Procesos Químicos en Cuba

En Cuba se han desarrollado diversos simuladores especialmente vinculados a la industria azucarera y alcohólica, entre los que se encuentran TERMOAZUCAR, ACOPLA, SIMFAD y SIDEL. De estos programas desarrollados el TERMOAZUCAR es el único de naturaleza modular-secuencial, aunque fue creado para procesos específicos, utiliza un sistema experto para el análisis de los resultados [5]. El simulador TERMOAZUCAR puede enlazarse coherentemente con un Programa Experto computarizado (el ANSTE), cuenta con un solo módulo de cálculo para la simulación de destilerías de alcohol, lo cual limita sus posibilidades de evaluación en distintas tecnologías.

Según un estudio realizado en la Universidad de las Ciencias Informáticas (UCI) por el Grupo para la informatización de la industria cubana, el TERMOAZUCAR y otros Simuladores cubanos están limitados en muchos aspectos que deben ser superados; tanto desde el punto de vista de la programación como de la posibilidad del empleo coherente de un conjunto de técnicas, empleadas internacionalmente, para la síntesis de procesos y la optimización [6]. Teniendo en cuenta estos aspectos, se desarrolla en la UCI un simulador con el fin de obtener un producto que no presentara las limitaciones de sus similares desarrollados anteriormente en Cuba y pueda ubicarse a la altura de sus similares extranjeros, pero el mismo presenta como características desfavorables estar desarrollado para propósitos específicos, no soporte general a la simulación de procesos y no genericidad de los flujos, por lo cual se hace necesario realizar el análisis y diseño para un simulador que permita un soporte general a la simulación de procesos en la industria química.

### **1.5 Metodologías de Desarrollo de Software**

El análisis y diseño conforman elementos esenciales en el proceso de desarrollo de un producto software, obligando a que se le preste una atención especial, por tal motivo se han ido creando metodologías que sirven de base para tomar decisiones que afectarán el producto final.

Las Metodologías de Desarrollo de Software comienzan su proceso de evolución con el advenimiento de la disciplina de la ingeniería del software, sus bases son proveer a los desarrolladores de software un conjunto de procedimientos, técnicas, herramientas y soporte documental que apoyen la construcción de un producto software final.

Una metodología de desarrollo de software define Quién debe hacer Qué, Cuándo y Cómo debe hacerlo, lo que permite producir organizada y económicamente software de alta calidad, siguiendo una serie de pasos donde se utilizan un conjunto de técnicas, notación y normas de documentación preestablecidas en dependencia de la metodología usada, las metodologías presentan una forma de modelar el mundo real o entorno a automatizar, lo que posibilita obtener una visión global del sistema, para facilitar la especificación de los requerimientos, las restricciones del sistema, la solución del problema, y con ello la obtención de un producto de mejor calidad, las tendencias actuales exigen un proceso bien definido en vista a lograr los objetivos propuestos.

Un proceso de desarrollo de software es el conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema software [8]. En la actualidad las metodologías de desarrollo para el control del proceso se agrupan en dos clasificaciones diferentes, las metodologías rígidas o pesadas, caracterizadas por exponer procesos basados en planeación exhaustiva y entre las que se encuentran el Proceso Unificado (RUP), Microsoft Solution Framework (MSF), Cubika Development Process (CDP), y las metodologías ágiles o ligeras, las cuales presentan como característica fundamental la habilidad de responder al cambio aún en etapas tardías del desarrollo, entre las que se pueden mencionar Extreme Programming (XP), Crystal Clear, SCRUM y Dynamic System Development Method (DSDM).

### 1.5.1 Proceso Unificado de Desarrollo de Software(RUP)

El Proceso Unificado de Desarrollo de Software o simplemente Proceso Unificado es una de las metodologías llamadas tradicionales, robustas o pesadas, descrito por sus autores Ivar Jacobson, Grady Booch y James Rumbaugh como el producto final de tres décadas de desarrollo y uso práctico, presenta entre sus características ser un proceso de desarrollo orientado a objeto, utilizar el Lenguaje Unificado de Modelado (Unified Modeling Language, UML) como lenguaje de representación visual, unificar los mejores elementos de metodologías anteriores y estar preparado para desarrollar grandes y complejos proyectos.

El Proceso Unificado se repite a lo largo de una serie de ciclos que constituyen la vida de un sistema. Cada ciclo consta de cuatro fases: concepción o inicio, elaboración, construcción y transición, siempre que concluye un ciclo se produce una nueva versión del sistema y cada versión es un producto preparado para su entrega [9].

El Proceso Unificado de Desarrollo de Software se caracteriza fundamentalmente por ser:

- Dirigido por Casos de Uso.
- Centrado en la Arquitectura.
- Iterativo e Incremental.
- Dirigido por casos de uso: Los casos de uso representan los requisitos de software capturados durante el flujo de trabajo de requisitos, la planificación del proyecto se hace en términos de casos de uso, los desarrolladores crean realizaciones de casos de uso en términos de clases y subsistemas, los componentes se incorporan en los incrementos y cada uno realiza un conjunto de

casos de uso, y por último se verifica que el sistema implementa los casos de uso correctos para el usuario. En otras palabras los casos de uso guían la arquitectura del sistema, enlazan todas las actividades del desarrollo y dirigen el proceso de desarrollo.

- **Centrado en la arquitectura:** La arquitectura representa la forma del futuro sistema en términos de vistas arquitectónicas, sobre la cual equipo de desarrollo y usuarios deben estar de acuerdo, ya que estas describen los elementos del modelo más importantes para su desarrollo, la arquitectura va madurando en las interacciones comenzando con los casos de uso relevantes desde el punto de vista arquitectónico.
- **Iterativo e incremental:** El Proceso Unificado propone que cada fase se desarrolle en iteraciones, ya que el incremento en la complejidad de los sistemas actuales hace que sea factible dividir el trabajo en partes más pequeñas o mini-proyectos. Cada mini-proyecto es una iteración que resulta en un incremento. Una iteración involucra actividades de todos los flujos de trabajo, aunque desarrolla fundamentalmente algunos más que otros.

### 1.5.2 Programación Extrema (XP)

Programación Extrema o Extreme Programming (XP) es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, ideada por Kent Beck, como proceso de creación de software diferente al convencional. En palabras de Beck: "XP es una metodología ligera, eficiente, con bajo riesgo, flexible, predecible y divertida para desarrollar software" [10], sus principales objetivos son lograr la satisfacción del cliente y potenciar al máximo el trabajo en grupo. La metodología XP se basa en la realimentación continua entre el cliente y el equipo de desarrollo, la comunicación fluida entre todos los participantes, la simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. La principal diferencia con las metodologías pesadas es que pone más énfasis en la adaptabilidad que en la previsibilidad.

Las características fundamentales del método son:

- Apropiado para entornos volátiles.
- Desarrollo iterativo e incremental: pequeñas mejoras continuas.

- Programación en parejas: dos personas por máquina, esto hace el desarrollo más llevadero y el surgimiento de ideas, y la estandarización del código.
- Pruebas unitarias continuas: Orientado a las pruebas, se realizan pruebas de unidad de los módulos frecuentemente repetidas y automatizadas.
- 40 horas de trabajo semanal: las horas extras mitigan y agotan los ánimos de los desarrolladores.
- Cliente en el equipo: Se recomienda que un representante del cliente trabaje junto al equipo de desarrollo.
- Corrección de todos los errores antes de añadir nueva funcionalidad. Hacer entregas frecuentes.
- Refactorización del código: refactorizar siempre que se pueda.
- Propiedad colectiva del código: en lugar de dividir responsabilidades en el desarrollo, todo desarrollador puede corregir y extender cualquier parte del proyecto.
- Simplicidad en el código: es la mejor manera de que las cosas funcionen.
- Centrado en el resultado: cumplir con lo que se pide y nada más.

### 1.5.3 Selección

Para el presente trabajo se selecciona como metodología de desarrollo de software el Proceso Unificado de Desarrollo (RUP), debido a que es una metodología probada para el desarrollo de proyectos complejos, cubre todo el ciclo de vida del producto guiado por los casos de uso, propone un desarrollo iterativo e incremental y las primeras iteraciones se enmarcan en obtener una arquitectura de software que garantice la viabilidad del proyecto.

### 1.6 Herramientas CASE

A medida que los sistemas que hoy se construyen se tornan más y más complejos, las herramientas de modelado con UML ofrecen muchos beneficios para todos los involucrados en un proyecto, por ejemplo, administrador del proyecto, analistas, arquitectos, desarrolladores y otros. Las herramientas CASE de modelado con UML permiten aplicar la metodología de análisis y diseño orientados a objetos y abstraerse del código fuente, en un nivel donde la arquitectura y el diseño se tornan más obvios y más fáciles de entender y modificar. Cuanto más grande es un proyecto, es más importante utilizar una herramienta CASE. [11]

### 1.6.1 Rational Rose Enterprise

Rational Rose Enterprise desarrollada por la Rational Corporation, es una herramienta CASE de desarrollo, basada en modelos que se integra con las bases de datos y los IDE de las principales plataformas del sector, utiliza UML para el modelado visual, es una herramienta de diseño orientado a objetos que permite, especificar, analizar, diseñar el sistema antes de Codificarlo, además de hacer énfasis en los detalles más importantes del desarrollo, lo que posibilita obtener un software de mayor calidad, propone la utilización de cuatro tipos de modelo para realizar un diseño del sistema, utilizando una vista estática y otra dinámica de los modelos del sistema, uno lógico y otro físico. Permite crear y refinar estas vistas creando de esta forma un modelo completo que representa el dominio del problema y el sistema de software, además incluye información entrelazada de diversos artefactos y descripciones de las diversas actividades. En general, Rational Rose Enterprise ofrece un lenguaje de modelado común que agiliza la creación del software.

Características:

- Desarrollo Iterativo.
- Trabajo en Grupo.
- Generador de Código.
- Ingeniería Inversa.
- Mantiene la consistencia de los modelos del sistema software.
- Chequeo de la sintaxis UML.
- Generación Documentación automáticamente.
- Generación de Código a partir de los Modelos.
- Ingeniería Inversa (crear modelo a partir código).
- Soporte para análisis de patrones ANSI C++, Rose J y Visual C++ basado en "Design Patterns: Elements of Reusable Object-Oriented Software".
- Característica de control por separado de componentes modelo que permite una administración más granular y el uso de modelos.
- Soporte de ingeniería Forward y/o reversa para algunos de los conceptos más comunes de Java 1.5.

- La generación de código Ada, ANSI C ++, C++, CORBA, Java y Visual Basic, con capacidad de sincronización modelo- código configurables.
- Capacidad de análisis de calidad de código.
- **Modelado UML** para trabajar en diseños de base de datos, con capacidad de representar la integración de los datos y los requerimientos de aplicación a través de diseños lógicos y físicos.
- Integración con otras herramientas de desarrollo de Rational.
- Capacidad para integrarse con cualquier sistema de control de versiones SCC-compliant.

### 1.6.2 Visual Paradigm

Visual Paradigm para UML es una herramienta CASE profesional, potente y de fácil uso para el modelado en UML, soporta el ciclo de vida completo del desarrollo de software, dígame análisis y diseño, construcción, pruebas y despliegue. Esta herramienta permite construir aplicaciones de calidad a un menor costo. Entre sus rasgos distintivos está el de posibilitar crear todos los tipos de diagramas UML, revertir código fuente a modelos UML, generar código desde diagramas y generar documentación.

Características:

- Diagramas de Procesos de Negocio - Proceso, Decisión, Actor de negocio, Documento.
- Modelado colaborativo con CVS y Subversion.
- Ingeniería de ida y vuelta.
- Ingeniería inversa - Código a modelo, código a diagrama.
- Ingeniería inversa Java, C++, Esquemas XML, XML, NET exe/dll, CORBA IDL.
- Generación de código - Modelo a código, diagrama a código.
- Editor de Detalles de Casos de Uso - Entorno todo-en-uno para la especificación de los detalles de los casos de uso, incluyendo la especificación del modelo general y de las descripciones de los casos de uso.
- Diagramas de flujo de datos.
- Generación de bases de datos - Transformación de diagramas de Entidad-Relación en tablas de base de datos.

- Ingeniería inversa de bases de datos - Desde Sistemas Gestores de Bases de Datos (DBMS) existentes a diagramas de Entidad-Relación.
- Generador de informes para generación de documentación.
- Distribución automática de diagramas - Reorganización de las figuras y conectores de los diagramas UML.
- Importación y exportación de ficheros XMI.
- Integración con Visio - Dibujo de diagramas UML con plantillas (stencils) de MS Visio.
- Editor de figuras.

### **1.6.3 Selección**

Como se ha visto en las descripciones anteriores ambas herramientas son potentes software y ofrecen muchas facilidades de forma general para el desarrollo de aplicaciones, para el presente trabajo se selecciona Rational Rose Enterprise entre otras razones por estar vinculado directamente a la metodología de desarrollo seleccionada anteriormente, además de que el hardware en el laboratorio donde estará enmarcado el desarrollo del producto no cuenta con las potencialidades necesarias para trabajar con la herramienta Visual Paradigm.

## **1.7 Conclusiones**

En este capítulo se definieron los conceptos fundamentales para el desarrollo del trabajo, así como se realizó una caracterización de los principales simuladores de procesos químicos desarrollados en Cuba y el mundo, se desarrolló un estudio de las metodologías y herramientas posibles a utilizar en el desarrollo del presente trabajo, seleccionando como metodología de desarrollo de software el Proceso Unificado de Desarrollo (RUP) y como herramienta CASE al Rational Rose Enterprise.

# Capítulo

# 2

## Análisis del Sistema

### 2.1 Introducción

En este capítulo, se realiza el modelado de un Subsistema de modelos matemáticos, componentes químicos y propiedades físico-químicas, que de solución al problema planteado, se describen los principales conceptos y se presenta el modelo de dominio como alternativa al modelo de negocio con objetivo de entender el contexto en que se ubica el Subsistema. Se enumeran los requisitos funcionales y no funcionales que guían el desarrollo del trabajo, así como se determinan y describen los actores y casos de uso, y se propone el subsistema a desarrollar.

### 2.2 Situación Problemática

El simulador desarrollado en la Universidad de Ciencias Informáticas (UCI) con el objetivo de obtener un producto que no presentara las limitaciones de sus similares desarrollados en Cuba, presenta como características desfavorables las que se mencionan a continuación:

- Desarrollado para propósitos específicos.
- No posibilita un soporte general a la simulación de procesos.
- No genericidad de los flujos de corrientes.
- No presenta una base de componentes químicos.

### 2.3 Modelo de Dominio

El modelo de dominio es una de las vías que brinda el Proceso Unificado para la comprensión del contexto del sistema, capturando los tipos más importantes de objetos o eventos dentro del entorno donde este se encontrará. Los objetos del dominio representan “cosas” que existen o los eventos que suceden en el entorno en el que trabaja el sistema. [12]

El modelo de dominio se describe mediante diagramas de UML, en los cuales se muestran a los clientes, usuarios, revisores y a otros desarrolladores las clases del dominio y cómo se relacionan unas con otras mediante asociaciones. [12]

El objetivo fundamental del modelo de dominio es comprender y describir las principales clases dentro de lo que abarca el sistema, posibilitando, a los involucrados en el desarrollo del producto ya sean clientes, desarrolladores e interesados, utilizar un vocabulario común el cual permita, capturar correctamente los requisitos y por ende obtener el producto deseado.

### **2.3.1 ¿Por qué modelo de dominio?**

Se realiza el modelo de dominio debido primeramente a que no existe un negocio definido, por lo cual, no se pueden determinar los procesos y roles del proceso de negocio, haciéndose engorroso y poco exacto la descripción de los mismos. Teniendo en cuenta además de lo anteriormente expuesto que no están definidas las actividades y procesos a automatizar, que el proceso en desarrollo está altamente centrado en herramientas y tecnologías informáticas y que no se cuenta con un cliente, debido a tratarse del desarrollo de un producto para el mercado, se determina que no es viable llevar a cabo un modelo de negocio y se hace necesario realizar la descripción de la aplicación mediante un modelo de dominio o modelo de conceptos del dominio, el cual agrupa los conceptos fundamentales de objetos relacionados con el dominio, sus atributos y sus relaciones.

### **2.3.2 Conceptos del Modelo del Dominio**

Proceso: Conjunto de operaciones químicas y/o físicas en las cuales están involucrados determinados equipos o dispositivos, así como componentes químicos. Un proceso representa las mezclas, separación, calentamiento y enfriado de estos componentes a través de los equipos presentes. El proceso tiene como objetivo obtener un conjunto de productos o sustancias que se diferencian del conjunto inicial en su composición o estado.

Diagrama de Flujo de Proceso (DFP): Contiene la representación gráfica de un proceso en cuestión.

Diagrama de Flujo de Información (DFI): Contiene el orden en que fluye la información en el diagrama de procesos (DFP), el DFI es quien rige el orden en que se deben evaluar las operaciones unitarias presentes en el proceso.

Subsistema de Interfaz Gráfica: Es a través del cual se realiza el diagrama de flujo de información, así como todas las operaciones sobre la BD de componentes, interactúan con la mayoría de las funcionalidades del subsistema.

Subsistema de Análisis de los Resultados: Se encarga de analizar los resultados de la simulación.

Modelo Matemático: Sistema de ecuaciones que simula el funcionamiento real de un equipo determinado, tiene asociado un conjunto de variables de entrada, que deben ser especificadas, esto es datos conocidos a través de los cuales por medio de la resolución del modelo matemático se obtienen un conjunto de variables de salidas o variables desconocidas.

Componente químico: Sustancia química la cual tiene asociado un conjunto de propiedades físico-químicas.

Flujos: Representan las corrientes de entrada o salida a los equipos.

Propiedades físico-químicas: Conjunto de propiedades ya sean físicas o químicas que representan las características de un componente químico determinado.

Unidad de operación, equipo o módulo: Representan a los equipos que son usados en la realidad en los distintos procesos. Tienen asociado un conjunto de modelos matemáticos, en dependencia de la exactitud de la representación deseada para simular su funcionamiento o de las características específicas del proceso.

### **2.3.3 Diagrama del Modelo del Dominio**

A continuación se muestra cómo interactúan los conceptos manipulados en Modelo de Dominio.

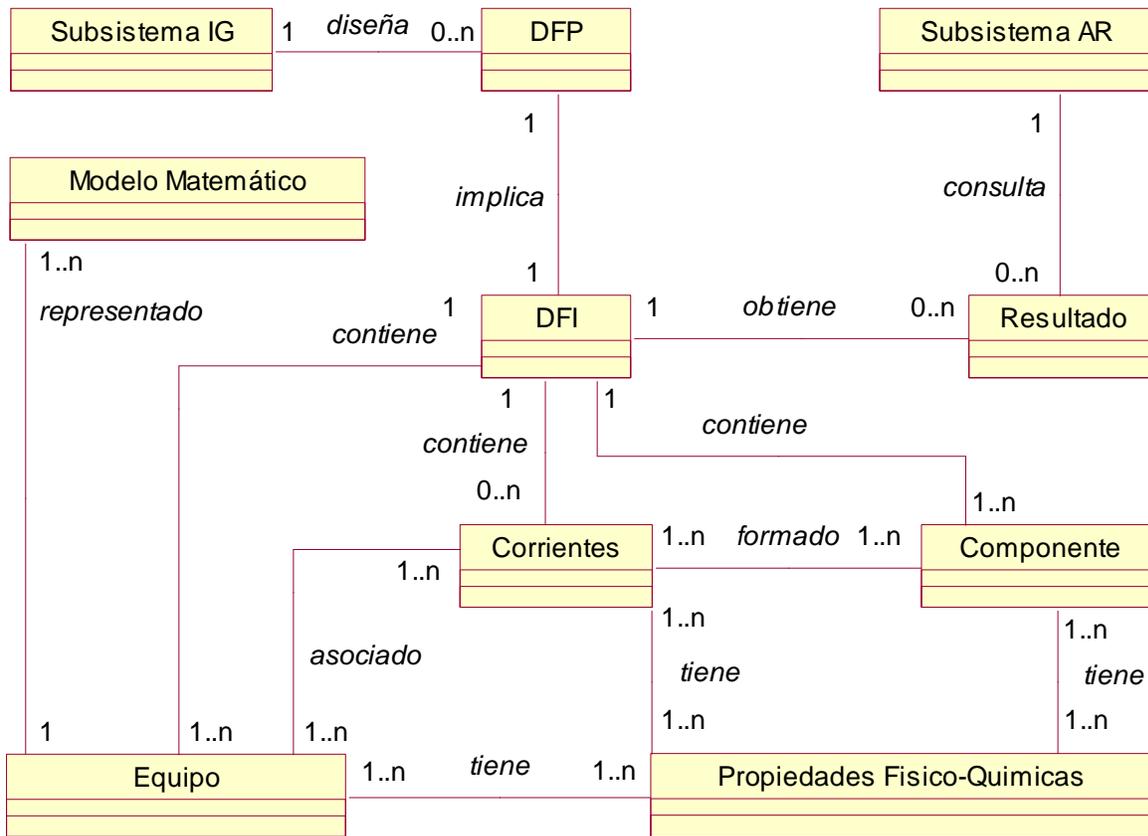


Figura 1 Diagrama 1 Modelo de Dominio

### 2.3.4 Descripción del Modelo del Dominio

El subsistema de Interfaz Gráfica es el encargado de realizar el diseño del proceso a simular a través del diagrama de flujo del proceso (DFP), especificando las unidades que participan en el proceso así como los flujos o corrientes de energía o materia entre estas unidades. El DFP es transmitido al subsistema de modelos matemáticos componentes químicos y propiedades físico-químicos que aquí se diseña en forma de diagrama de flujo de información (DFI). El subsistema de Interfaz Gráfica especifica los componentes químicos que participan en el proceso, selecciona el paquete de propiedades que se utilizará en la simulación del proceso en cuestión, y realizar ajustes adicionales al proceso. Existiendo la información necesaria se procede a la simulación y obtención de los resultados, los cuales son consultados por el subsistema de Análisis de Resultados para su procesamiento.

## 2.4 Descripción del Subsistema propuesto

El subsistema de modelos matemáticos, componentes químicos y propiedades físico-químicas permite la integración con los subsistemas de Interfaz Gráfica y Análisis de los resultados brindándole a los mismos interfaces donde están disponibles las funcionalidades que brinda este subsistema para ser usadas y llevar a cabo la simulación de procesos químicos y análisis de los resultados obtenidos, las cuales responden a los requisitos que se muestran a continuación.

## 2.5 Requerimientos del Subsistema

Los requerimientos son condiciones o capacidades que necesita un usuario sean satisfechas para resolver un problema o lograr un objetivo específico, o sea los requerimientos son las condiciones o capacidades que tienen que ser alcanzadas o que debe cumplir un sistema o componente de un sistema para satisfacer al usuario o cliente. [13]

### 2.5.1 Requerimientos funcionales

Los requerimientos funcionales son capacidades o condiciones que el sistema debe cumplir, lo cual los convierte en el punto de partida para identificar qué debe hacer el sistema. [13]

- R.1. Gestionar componentes químicos de la Base de Datos de componentes.
  - R.1.1. Cargar componentes de la Base de Datos.
  - R.1.2. Insertar nuevo componente químico.
    - R.1.2.1. Insertar componente químico puro.
    - R.1.2.2. Insertar componente químico compuesto.
  - R.1.3. Gestionar propiedades de un componente personalizado.
    - R.1.3.1. Modificar propiedades de un componente personalizado.
    - R.1.3.2. Estimar propiedades de componente químicos compuesto.
  - R.1.4. Realizar búsqueda de componente.
  - R.1.5. Clonar componente.
  - R.1.6. Eliminar componente personalizado.
- R.2. Gestionar biblioteca de componentes.
  - R.2.1. Cargar bibliotecas de componentes.

- R.2.2. Listar bibliotecas de componentes.
- R.2.3. Crear biblioteca de componentes personalizada.
- R.2.4. Gestionar componentes de una biblioteca personalizada.
  - R.2.4.1. Adicionar componente a una biblioteca.
  - R.2.4.2. Eliminar componente de una biblioteca.
  - R.2.4.3. Listar componentes de una biblioteca.
- R.2.5. Eliminar biblioteca de componentes personalizada.
- R.3. Gestionar proceso de simulación.
  - R.3.1. Crear nuevo proceso.
  - R.3.2. Salvar fichero de proceso.
  - R.3.3. Abrir fichero de proceso.
- R.4. Simular proceso químico.
  - R.4.1. Gestionar componentes químicos presentes en un proceso.
    - R.4.1.1. Adicionar componente químico a un proceso.
    - R.4.1.2. Eliminar componente químico de un proceso.
  - R.4.2. Gestionar paquetes de propiedades.
    - R.4.2.1. Listar paquetes de propiedades.
    - R.4.2.2. Asignar paquetes de propiedades al proceso.
  - R.4.3. Gestionar unidad de operación.
    - R.4.3.1. Listar unidades de operación.
    - R.4.3.2. Adicionar unidad de operación o módulo.
    - R.4.3.3. Modificar propiedades específicas de la unidad.
    - R.4.3.4. Eliminar unidad de operación.
    - R.4.3.5. Obtener estado de una unidad de operación.
  - R.4.4. Gestionar corriente.
    - R.4.4.1. Listar corrientes.
    - R.4.4.2. Crear corriente.
    - R.4.4.3. Gestionar corriente de entrada o información.
      - R.4.4.3.1. Modificar composición de la corriente.

R.4.4.3.2. Modificar propiedades de la corriente.

R.4.4.4. Obtener propiedades de la corriente.

R.4.4.5. Conectar corriente a unidad de operación.

R.4.4.6. Eliminar corriente.

R.4.4.7. Calcular propiedades.

R.5. Guardar resultados de la simulación.

R.6. Gestionar resultados de una simulación.

R.6.1. Cargar resultado de la simulación de procesos.

R.6.2. Listar resultados de la simulación de procesos.

R.6.3. Obtener resultados de la simulación de un proceso.

R.6.4. Eliminar resultado de la simulación de un proceso.

R.7. Manipular unidades de medida.

R.7.1. Manipular unidades de medida de las propiedades de un componente.

R.7.2. Manipular unidades de medida de las propiedades de una unidad de operación.

R.7.3. Manipular unidades de medida de las propiedades de una corriente.

### 2.5.2 Requerimientos no funcionales

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener, Los requerimientos no funcionales forman una parte significativa de la especificación. Son importantes para que clientes y usuarios puedan valorar las características no funcionales del producto. [13]

➤ Interfaces de usuario.

Este subsistema de modelos matemáticos, componentes químicos y propiedades físico-químicas no interactúa directamente con el usuario de la aplicación, siendo transparente para este, por lo cual no tiene requisitos de apariencia o interfaz externa.

➤ Interfaces Software.

El subsistema de modelos matemáticos, componentes químicos y propiedades físico-químicas debe brindar interfaces a los subsistemas de Interfaz Gráfica y Análisis de los Resultados donde se presenten las funcionalidades disponibles.

➤ Usabilidad.

La aplicación podrá ser usada por cualquier personal capacitado, en especial por ingenieros químicos. Contendrá un manual de usuario con las descripciones básicas sobre cómo trabajar con la aplicación incluyendo las especificaciones o características de los procesos. No tiene restricciones de usuarios.

➤ Rendimiento.

Las operaciones unitarias se efectuarán inmediatamente se disponga de los datos necesarios para realizar los cálculos.

➤ Seguridad.

- Confiabilidad: No existen restricciones de acceso a la información manejada.
- Integridad: Solo se puede modificar o eliminar aquella información que haya sido introducida por el usuario.
- Disponibilidad: El subsistema brinda información de los procesos, componentes, propiedades, flujos y corrientes cada vez que le sean solicitadas.

➤ Soporte

- Se le debe dar un mantenimiento periódico al subsistema y a la Base de Datos de componentes.
- Se debe brindar una guía sobre las funcionalidades que brinda el subsistema, donde se expongan y explique en qué consisten.

➤ De Software

Para poder instalar este sistema se debe disponer con un sistema operativo superior o igual a Windows XP.

➤ De Hardware

Para instalar la aplicación se necesitan 1 GB de disco duro disponibles, máquinas con 256 MB de RAM y Procesador Pentium 3 o superior.

➤ Restricciones en el Diseño y la Implementación

Se usará como lenguaje de programación C#.

Se utiliza UML como lenguaje de modelado.

### 2.5.3 Modelo de Casos de Uso del Subsistema

El modelo de casos de uso captura todos los requisitos funcionales del subsistema, permitiendo que los desarrolladores de software, usuarios e interesados lleguen a un acuerdo sobre los mismos, es decir, acordar las condiciones y posibilidades que debe cumplir el subsistema. De forma general el modelo de casos de uso describe las funcionalidades propuestas para el nuevo subsistema y ayuda a los clientes, usuarios, desarrolladores e interesados a llegar a un acuerdo sobre cómo utilizar el mismo.

Un modelo de casos de uso es un modelo del subsistema el cual contiene a los actores y casos de uso identificados, así como sus relaciones. Estos casos de uso, actores y relaciones son representados en un diagrama de casos de uso el cual modela el comportamiento del sistema en términos de asociación entre cada par actor/caso de uso que interactúan.

Los actores son el entorno del subsistema, comunicándose con este mediante el envío y recepción de mensajes. Por su parte los casos de uso especifican el subsistema y están diseñados para cumplir los deseos del usuario cuando utiliza el subsistema [14].

#### 2.5.3.1 Actores

Los actores representan personas, sistemas o hardware externo que interactúan con el sistema, o sea, los actores son agentes externos al subsistema los cuales tienen relación con el mismo. La identificación de los actores del sistema, determina el entorno externo del sistema [14]. A continuación se muestra una tabla con la descripción de los actores determinados para el subsistema de modelos matemáticos, componentes químicos y propiedades físico-químicas.

Tabla 1 Actores

Actor	Descripción
Subsistema Interfaz Gráfica	Es a través del cual se realiza el diagrama de flujo de información, así como la todas las operaciones sobre la BD de componentes.
Subsistema Análisis de los Resultados	Se encarga de analizar los resultados de un

	proceso de simulación.
--	------------------------

### 2.5.3.2 Casos de Uso del Subsistema

Cada forma que los actores usan el subsistema se representa con un caso de uso, los mismos son fragmentos de funcionalidad del subsistema y especifican una secuencia de acciones, incluyendo alternativas dentro de la secuencia, que este puede llevar a cabo, y que producen un resultado observable de valor para un actor concreto [14].

Tabla 2 Casos de Uso del Subsistema

#	Nombre CU	Actor	Descripción	Referencia	Prioridad
1	Gestionar Componente	Subsistema IG	Posibilita gestionar componentes de la BD de componentes del simulador permitiendo cargar los componentes, adicionar, clonar, modificar y eliminar componentes personalizados.	R.1, R.1.1, R.1.2, R.1.3, R.1.4, R.1.5, R.1.6 CU incluidos: Gestionar Propiedades	Secundario
2	Gestionar Proceso	Subsistema IG	Permite gestionar procesos de simulación, crear un nuevo proceso, salvar y abrir procesos existentes.	R.3, R.3.1, R.3.2, R.3.3	Secundario
3	Gestionar Biblioteca Personalizada	Subsistema IG	Permite gestionar bibliotecas de componentes personalizados, cargar, crear biblioteca, adicionar o eliminar componentes de la biblioteca y eliminar una biblioteca	R.2, R.2.1, R.2.2, R.2.3, R.2.4, R.2.4.1, R.2.4.2, R.2.4.3, R.2.5	Secundario

4	Gestionar Unidad de Operación	Subsistema IG	Permite gestionar todo lo relacionado con las unidades de operación unitarias o módulos, listar unidades de operación, adicionar y eliminar unidades del proceso de simulación, modificar propiedades de una unidad y obtener estado de la unidad.	R.4.3, R.4.3.1, R.4.3.2, R.4.3.3, R.4.3.4, R.4.3.5	Critico
5	Gestionar Propiedades	Subsistema IG	Permite gestionar las propiedades de componentes químicos personalizados, modificar las propiedades y para el caso de componentes compuestos estimar las mismas.	R.1.3, R.1.3.1, R.1.3.2	Secundario
6	Guardar Resultado	Subsistema IG	Permite guardar los resultados de la simulación de un proceso.	R.5	Critico
7	Gestionar Corriente	Subsistema IG	Permite gestionar las corrientes de los procesos de simulación, listar corrientes, crear corriente, obtener propiedades de una corriente, conectar corriente a unidad de operación o eliminar una corriente.	R.4.4, R.4.4.1, R.4.4.2, R.4.4.4, R.4.4.5, R.4.4.6	Critico
8	Gestionar Corriente de Entrada	Subsistema IG	Permite gestionar las corrientes de entrada de información, modificar composición o modificar propiedades de una corriente de entrada.	R.4.4.3, R.4.4.3.1, R.4.4.3.2 CU incluidos: Calcular	Critico

				Propiedades	
9	Gestionar Paquetes de Propiedades	Subsistema IG	Permite listar los paquetes de propiedades o asignarle un nuevo paquete de propiedades al proceso de simulación en cuestión.	R.4.2, R.4.2.1, R.4.2.2	Critico
10	Gestionar Componente Proceso	Subsistema IG	Permite gestionar los componentes que estarán presentes en un proceso de simulación, ya sea adicionar componentes al proceso o eliminarlos.	R.4.1, R.4.1.1, R.4.1.2	Critico
11	Buscar Componentes	Subsistema IG	Permite realizar una búsqueda de componente en la BD del simulador.	R.1.4	Critico
12	Calcular Propiedades	Subsistema IG	Permite luego que se haya modificado las propiedades de una corriente, calcular las propiedades dependientes de esa corriente.	R.4.4.7	Critico
13	Manipular Unidades de Medida	Subsistema IG	Permite expresar los datos en diferentes unidades de medida	R.7, R.7.1, R.7.2, R.7.3	Secundario
14	Gestionar Resultados	Subsistema AR	Permite gestionar los resultados de la simulación de procesos, cargar, listar, obtener y eliminar resultado.	R.6, R.6.1, R.6.2, R.6.3, R.6.4	Critico
15	Simular	Subsistema IG	Permite realizar la simulación de un proceso hasta donde los	R.4	Critico

			datos sean suficientes, teniendo en cuenta el DFI.		
--	--	--	--	--	--

### 2.5.4 Diagrama de Casos de Uso del Subsistema

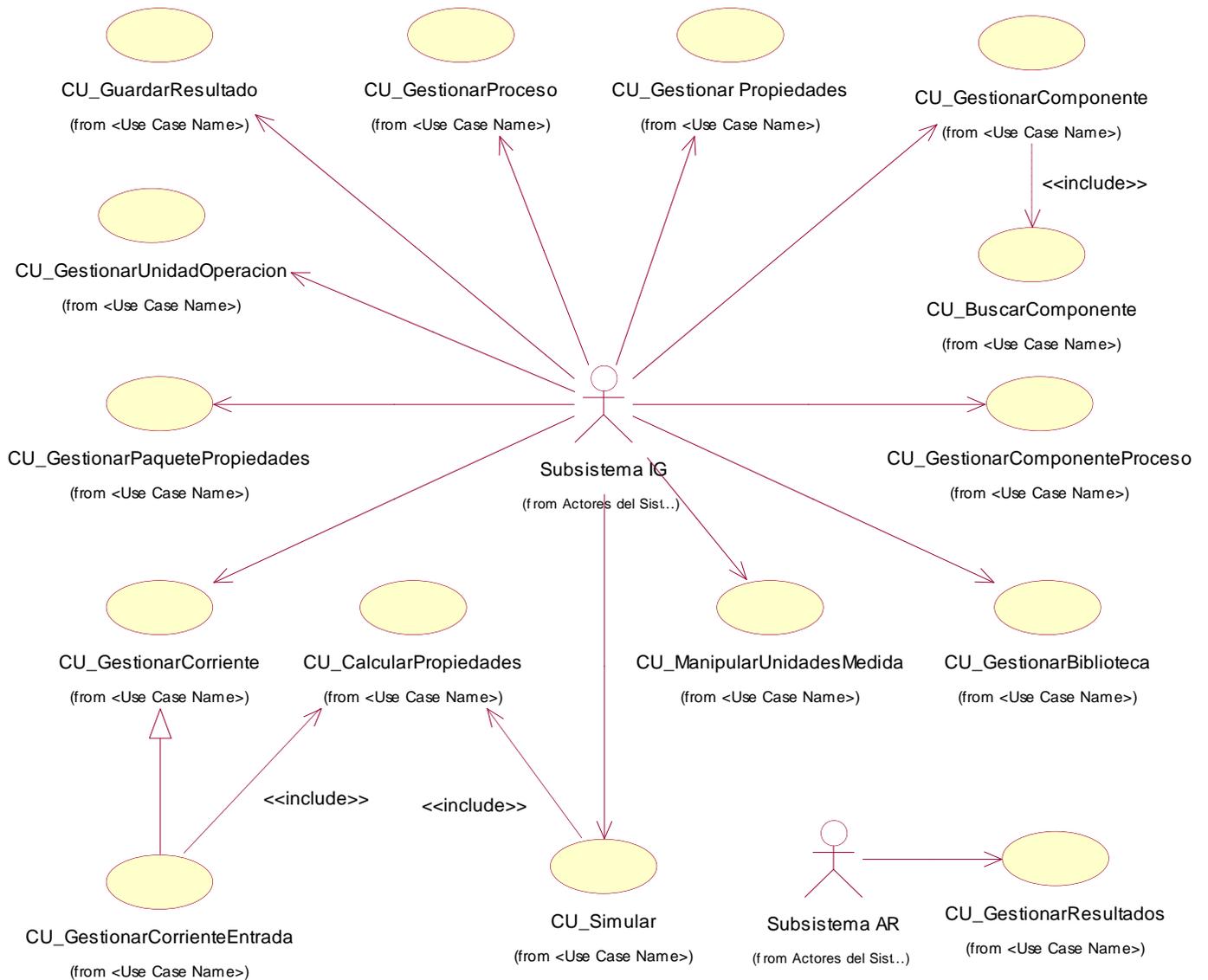


Figura 2 Modelo de Casos de Uso del Sistema

## 2.6 Descripción de los CU

A continuación se muestran las descripciones de los Casos de Uso críticos, para ver las descripciones de los restantes Casos de Uso, consultar el Anexo 1.

Tabla 3 Caso de Uso: Buscar Componente

<b>Caso de Uso:</b>	Buscar Componente	
<b>Actores:</b>	Subsistema Interfaz Gráfica	
<b>Caso de Uso:</b>	Buscar Componente	
<b>Precondiciones:</b>		
<b>Referencias</b>	R.1.4	
<b>Prioridad</b>	Critico	
<b>Flujo Normal de Eventos</b>		
<b>Acción del Actor</b>		<b>Respuesta del Subsistema</b>
1. El subsistema Interfaz Gráfica invoca la función buscar componente e introduce los criterios de búsqueda.		2. El subsistema realiza la búsqueda de componentes. 3. El subsistema devuelve los componentes que cumplen los criterios de búsqueda.
<b>Flujos Alternos</b>		
Línea 3: El subsistema lanza un mensaje de error porque no se encontraron componentes que cumplan los criterios.		
<b>Poscondiciones</b>	La BD de componentes no sufre cambios.	

Tabla 4 Caso de Uso: Gestionar Paquetes de Propiedades

<b>Caso de Uso:</b>	Gestionar Paquetes de Propiedades
<b>Actores:</b>	Subsistema Interfaz Gráfica
<b>Resumen:</b>	El Caso de Uso se inicia cuando el subsistema Interfaz Grafica invoca algunas de las funciones para gestionar paquetes de propiedades ya sea listar los paquetes de propiedades o asignarle un nuevo paquete de propiedades al un proceso de simulación en cuestión. El subsistema ejecuta la función invocada y

	termina el CU.
<b>Precondiciones:</b>	
<b>Referencias</b>	R.4.2, R.4.2.1, R.4.2.2
<b>Prioridad</b>	Critico
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Subsistema</b>
1. El subsistema Interfaz Gráfica invoca alguna de las funciones para gestionar paquetes de propiedades.	2. El subsistema procesa la solicitud.
<b>Sección "Listar paquetes"</b>	
<b>Acción del Actor</b>	<b>Respuesta del Subsistema</b>
3. El subsistema Interfaz Gráfica invoca la función listar paquetes de propiedades.	4. El subsistema devuelve la lista de los paquetes de propiedades.
<b>Sección "Asignar paquete"</b>	
<b>Acción del Actor</b>	<b>Respuesta del Subsistema</b>
5. El subsistema Interfaz Gráfica invoca la función asignar paquete e introduce los datos necesarios.	6. El subsistema valida los datos. 7. El subsistema asigna el paquete de propiedades al proceso de simulación.
<b>Flujos Alternos</b>	
Línea 6: El subsistema lanza un mensaje de error porque no se entraron los datos correctamente.	
<b>Poscondiciones</b>	

Tabla 5 Caso de Uso: Gestionar Unidad de Operación

<b>Caso de Uso:</b>	Gestionar Unidad de Operación
<b>Actores:</b>	Subsistema Interfaz Gráfica
<b>Resumen:</b>	El Caso de Uso se inicia cuando el subsistema Interfaz Gráfica invoca algunas de las funciones para gestionar unidades de operaciones del proceso, ya sea listar las unidades de operación, agregar o eliminar unidad de operación al proceso, modificar propiedades de la unidad u obtener estado de una unidad. El

	subsistema ejecuta la función invocada y termina el CU.	
<b>Precondiciones:</b>		
<b>Referencias</b>	R.4.3, R.4.3.1, R.4.3.2, R.4.3.3, R.4.3.4, R.4.3.5	
<b>Prioridad</b>	Critico	
<b>Flujo Normal de Eventos</b>		
<b>Acción del Actor</b>		<b>Respuesta del Subsistema</b>
1.	El subsistema Interfaz Gráfica invoca alguna de las funciones para gestionar unidades de operación.	2. El subsistema procesa la solicitud.
<b>Sección “Listar unidades”</b>		
<b>Acción del Actor</b>		<b>Respuesta del Subsistema</b>
3.	El subsistema Interfaz Gráfica invoca la función listar unidades de operación.	4. El subsistema devuelve la lista de las unidades de operación.
<b>Sección “Adicionar unidad al proceso”</b>		
<b>Acción del Actor</b>		<b>Respuesta del Subsistema</b>
5.	El subsistema Interfaz Gráfica invoca la función adicionar nueva unidad al proceso e introduce los datos necesarios.	6. El subsistema valida los datos. 7. El subsistema agrega la nueva unidad de operación.
<b>Sección “Modificar propiedades”</b>		
<b>Acción del Actor</b>		<b>Respuesta del Subsistema</b>
8.	El subsistema Interfaz Gráfica invoca la función modificar propiedades de una unidad de operación e introduce los datos necesarios.	9. El subsistema valida los datos. 10. El subsistema modifica las propiedades de la unidad de operación.
<b>Sección “Eliminar unidad del proceso”</b>		
<b>Acción del Actor</b>		<b>Respuesta del Subsistema</b>
11.	El subsistema Interfaz Gráfica invoca la función eliminar unidad del proceso e introduce los datos necesarios.	12. El subsistema valida los datos. 13. El subsistema elimina la unidad de operación del proceso.
<b>Sección “Obtener estado”</b>		

Acción del Actor		Respuesta del Subsistema
14. El subsistema Interfaz Gráfica invoca la función obtener estado de una unidad de operación e introduce los datos necesarios.		15. El subsistema valida los datos. 16. El subsistema devuelve el estado de la unidad de operación.
Flujos Alternos		
Línea 6: El subsistema lanza un mensaje de error porque no se entraron los datos correctamente.		
Poscondiciones		

Tabla 6 Caso de Uso: Gestionar Corriente

<b>Caso de Uso:</b>	Gestionar Corriente	
<b>Actores:</b>	Subsistema Interfaz Gráfica	
<b>Resumen:</b>	El Caso de Uso se inicia cuando el subsistema Interfaz Grafica invoca algunas de las funciones para gestionar corrientes ya sea listar corrientes, crear corriente, obtener propiedades de una corriente, conectar corriente a unidad de operación o eliminar una corriente. El subsistema ejecuta la función invocada y termina el CU.	
<b>Precondiciones:</b>		
<b>Referencias</b>	R.4.4, R.4.4.1, R.4.4.2, R.4.4.4, R.4.4.5, R.4.4.6	
<b>Prioridad</b>	Critico	
Flujo Normal de Eventos		
Acción del Actor		Respuesta del Subsistema
1. El subsistema Interfaz Gráfica invoca alguna de las funciones para gestionar corrientes.		2. El subsistema procesa la solicitud.
Sección "Listar corrientes"		
Acción del Actor		Respuesta del Subsistema
3. El subsistema Interfaz Gráfica invoca la función listar corrientes de un proceso e introduce los datos.		4. El subsistema comprueba los datos. 5. El subsistema devuelve la lista de corrientes del proceso.
Sección "Crear corriente"		

<b>Acción del Actor</b>		<b>Respuesta del Subsistema</b>	
6. El subsistema Interfaz Gráfica invoca la función crear nueva corriente e introduce los datos necesarios.		7. El subsistema comprueba los datos. 8. El subsistema crea la nueva corriente.	
<b>Sección “Obtener propiedades de corriente”</b>			
<b>Acción del Actor</b>		<b>Respuesta del Subsistema</b>	
9. El subsistema Interfaz Gráfica invoca la función obtener propiedades de corriente e introduce los datos necesarios.		10. El subsistema comprueba los datos. 11. El subsistema devuelve las propiedades de la corriente.	
<b>Sección “Conectar corriente”</b>			
<b>Acción del Actor</b>		<b>Respuesta del Subsistema</b>	
12. El subsistema Interfaz Gráfica invoca la función conectar corriente e introduce los datos necesarios.		13. El subsistema comprueba los datos y si se puede realizar la conexión. 14. El subsistema conecta la corriente a una unidad de operación y actualiza el DFI.	
<b>Sección “Eliminar corriente”</b>			
<b>Acción del Actor</b>		<b>Respuesta del Subsistema</b>	
15. El subsistema Interfaz Gráfica invoca la función eliminar corriente e introduce los datos necesarios.		16. El subsistema comprueba los datos. 17. El subsistema elimina la corriente y actualiza el DFI.	
<b>Flujos Alternos</b>			
<p><b>Línea 4:</b> El subsistema lanza un mensaje de error porque los datos no son válidos.</p> <p><b>Línea 7:</b> El subsistema lanza un mensaje de error porque los datos no son válidos.</p> <p><b>Línea 10:</b> El subsistema lanza un mensaje de error porque los datos no son válidos.</p> <p><b>Línea 13:</b> El subsistema lanza un mensaje de error porque los datos no son válidos o porque no se puede conectar la corriente a la unidad especificada.</p> <p><b>Línea 16:</b> El subsistema lanza un mensaje de error porque los datos no son válidos.</p>			
<b>Poscondiciones</b>	Se actualiza el DFI.		

Tabla 7 Caso de Uso: Gestionar Corriente de Entrada

<b>Caso de Uso:</b>	Gestionar Corriente de Entrada	
<b>Actores:</b>	Subsistema Interfaz Gráfica	
<b>Resumen:</b>	El Caso de Uso se inicia cuando el subsistema Interfaz Gráfica invoca algunas de las funciones para gestionar corrientes de entrada ya sea modificar composición de una corriente de entrada o modificar propiedades de una corriente de entrada, El subsistema ejecuta la función invocada y termina el CU.	
<b>Precondiciones:</b>		
<b>Referencias</b>	R.4.4.3, R.4.4.3.1, R.4.4.3.2, CU incluidos: Calcular Propiedades	
<b>Prioridad</b>	Critico	
<b>Flujo Normal de Eventos</b>		
<b>Acción del Actor</b>		<b>Respuesta del Subsistema</b>
1. El subsistema Interfaz Gráfica invoca alguna de las funciones para gestionar corrientes de entrada.		2. El subsistema procesa la solicitud.
<b>Sección “Modificar Composición”</b>		
<b>Acción del Actor</b>		<b>Respuesta del Subsistema</b>
3. El subsistema Interfaz Gráfica invoca la función modificar composición e introduce los datos necesarios.		4. El subsistema valida los datos. 5. El subsistema modifica la composición de la corriente. 6. El subsistema ejecuta el CU Calcular Propiedades
<b>Sección “Modificar Propiedades”</b>		
<b>Acción del Actor</b>		<b>Respuesta del Subsistema</b>
7. El subsistema Interfaz Gráfica invoca la función modificar propiedades de la corriente e introduce los datos necesarios.		8. El subsistema valida los datos. 9. El subsistema modifica las propiedades de la corriente. 10. El subsistema ejecuta el CU Calcular Propiedades
<b>Flujos Alternos</b>		

<b>Línea 4:</b> El subsistema lanza un mensaje de error porque los datos no son válidos.
<b>Línea 6, 10:</b> El subsistema lanza un mensaje de error porque no se pueden calcular las propiedades.
<b>Línea 8:</b> El subsistema lanza un mensaje de error porque los datos no son válidos.
<b>Poscondiciones</b> Se actualiza el DFI.

Tabla 8 Caso de Uso: Calcular Propiedades

<b>Caso de Uso:</b>	Calcular Propiedades	
<b>Actores:</b>	Subsistema Interfaz Gráfica	
<b>Resumen:</b>	El Caso de Uso se inicia cuando el subsistema Interfaz Gráfica a modificado las propiedades de una corriente, como consecuencia se calculan las propiedades dependientes. El subsistema ejecuta y termina el CU.	
<b>Precondiciones:</b>		
<b>Referencias</b>	R.4.4.7	
<b>Prioridad</b>	Critico	
<b>Flujo Normal de Eventos</b>		
<b>Acción del Actor</b>	<b>Respuesta del Subsistema</b>	
1. El subsistema Interfaz Grafica ejecuto la sección modificar propiedades del caso de uso Gestionar Corriente de Entrada.	2. El subsistema obtiene las propiedades de la corriente que ha sido modificada. 3. El subsistema calcula las propiedades dependientes. 4. El subsistema modifica las propiedades de la corriente sustituyéndolas por las calculadas.	
<b>Flujos Alternos</b>		
<b>Línea 3:</b> El subsistema lanza un mensaje de error porque existen datos suficientes.		
<b>Poscondiciones</b>	Se modifican las propiedades dependientes de la corriente.	

Tabla 9 Caso de Uso: Guardar Resultados

<b>Caso de Uso:</b>	Guardar Resultados
<b>Actores:</b>	Subsistema Interfaz Gráfica

<b>Resumen:</b>	El Caso de Uso se inicia cuando el subsistema Interfaz Gráfica invoca la función guardar resultado. El subsistema ejecuta y termina el CU.	
<b>Precondiciones:</b>		
<b>Referencias</b>	R.5	
<b>Prioridad</b>	Secundario	
<b>Flujo Normal de Eventos</b>		
<b>Acción del Actor</b>		<b>Respuesta del Subsistema</b>
1.	El subsistema Interfaz Gráfica invoca la función guardar resultados.	2. El subsistema procesa la solicitud.
<b>Sección “Guardar Resultado”</b>		
<b>Acción del Actor</b>		<b>Respuesta del Subsistema</b>
3.	El subsistema Interfaz Gráfica invoca la función guardar resultado de la simulación.	4. El subsistema verifica que se pueda realizar la operación. 5. El subsistema guarda los resultados de la simulación del proceso.
<b>Flujos Alternos</b>		
<b>Línea 4:</b> El subsistema lanza un mensaje de error porque no se pueden guardar los resultados.		
<b>Poscondiciones</b>		

Tabla 10 Caso de Uso: Simular

<b>Caso de Uso:</b>	Simular	
<b>Actores:</b>	Subsistema Análisis de Resultado	
<b>Resumen:</b>	El Caso de Uso se inicia cuando el subsistema Interfaz Gráfica invoca la función simular proceso. El subsistema ejecuta y termina el CU.	
<b>Precondiciones:</b>		
<b>Referencias</b>	R.4	
<b>Prioridad</b>	Critico	
<b>Flujo Normal de Eventos</b>		
<b>Acción del Actor</b>		<b>Respuesta del Subsistema</b>

1. El subsistema Interfaz Gráfica invoca la función simular proceso.	2. El subsistema procesa la solicitud.
<b>Sección “Simular Proceso”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Subsistema</b>
3. El subsistema Interfaz Gráfica invoca la función simular proceso.	4. El subsistema verifica si se puede realizar la simulación basándose en el DFI. 5. El subsistema resuelve las operaciones unitarias e invoca el CU Calcular Propiedades según el DFI. 6. El subsistema actualiza las propiedades de las unidades de operación y las corrientes del proceso.
<b>Flujos Alternos</b>	
<b>Línea 4:</b> El subsistema lanza un mensaje de error porque no se pueden realizar la simulación y los motivos.	
<b>Poscondiciones</b>	

Tabla 11 Caso de Uso: Gestionar Componentes del Proceso

<b>Caso de Uso:</b>	Gestionar Componentes del Proceso
<b>Actores:</b>	Subsistema Análisis de Resultado
<b>Resumen:</b>	El Caso de Uso se inicia cuando el subsistema Interfaz Gráfica invoca algunas de las funciones para gestionar los componentes presentes en un proceso determinado. El subsistema ejecuta y termina el CU.
<b>Precondiciones:</b>	
<b>Referencias</b>	R.4.1, R.4.1.1, R.4.1.2
<b>Prioridad</b>	Critico
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Subsistema</b>
1. El subsistema Interfaz Gráfica invoca algunas	2. El subsistema procesa la solicitud.

de las funciones para gestionar los componentes presentes en un proceso.	
<b>Sección “Adicionar Componente al Proceso”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Subsistema</b>
3. El subsistema Interfaz Gráfica invoca la función adicionar componente al proceso e introduce los datos necesarios.	4. El subsistema valida los datos. 5. El subsistema adiciona un nuevo componente al proceso.
<b>Sección “Eliminar Componente del Proceso”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Subsistema</b>
6. El subsistema Interfaz Gráfica invoca la función eliminar componente del proceso e introduce los datos necesarios.	7. El subsistema valida los datos y si se puede eliminar el componente sin errores. 8. El subsistema elimina el componente del proceso.
<b>Flujos Alternos</b>	
<p><b>Línea 4:</b> El subsistema lanza un mensaje de error porque los datos no son correctos.</p> <p><b>Línea 7:</b> El subsistema lanza un mensaje de error porque los datos no son correctos o no se puede eliminar el componente.</p>	
<b>Poscondiciones</b>	

Tabla 12 Caso de Uso: Gestionar Resultado

<b>Caso de Uso:</b>	Gestionar Resultado
<b>Actores:</b>	Subsistema Análisis de Resultado
<b>Resumen:</b>	El Caso de Uso se inicia cuando el subsistema Análisis de Resultado invoca alguna de las funciones para gestionar resultados. El subsistema ejecuta y termina el CU.
<b>Precondiciones:</b>	
<b>Referencias</b>	R.6, R.6.1, R.6.2, R.6.3, R.6.4
<b>Prioridad</b>	Secundario

<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Subsistema</b>
1. El subsistema Análisis de Resultado invoca algunas de las funciones para gestionar resultados.	2. El subsistema procesa la solicitud.
<b>Sección “Cargar Resultado”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Subsistema</b>
3. El subsistema Análisis de Resultado crea una instancia del subsistema.	4. El subsistema carga los resultados de la BD de resultados.
<b>Sección “Listar Resultado”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Subsistema</b>
5. El subsistema Análisis de Resultado invoca la función listar resultados de simulación.	6. El subsistema devuelve un listado con los resultados de la simulación de procesos.
<b>Sección “Obtener Resultado”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Subsistema</b>
7. El subsistema Análisis de Resultado invoca la función obtener resultado de simulación e introduce los datos necesarios.	8. El subsistema verifica los datos suministrados. 9. El subsistema devuelve los resultados de la simulación del proceso.
<b>Sección “Eliminar Resultado”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Subsistema</b>
10. El subsistema Análisis de Resultado invoca la función eliminar resultado de simulación e introduce los datos necesarios.	11. El subsistema verifica los datos suministrados. 12. El subsistema elimina el resultado de la simulación de un proceso.
<b>Flujos Alternos</b>	
<b>Línea 8:</b> El subsistema lanza un mensaje de error porque no son válidos los datos suministrados.	
<b>Línea 11:</b> El subsistema lanza un mensaje de error porque no son válidos los datos suministrados.	
<b>Poscondiciones</b>	

## **2.7 Conclusiones**

En este capítulo se describió el modelo de dominio para comprender el contexto donde se ubica el subsistema a desarrollar, se determinaron las funcionalidades que debe brindar el mismo representadas en requisitos funcionales y no funcionales, a partir de los cuales se identificaron un total de 15 casos de uso. Así como también se elaboró el diagrama de casos de uso del sistema y se realizó la expansión o caracterización de los CU.

# Capítulo

# 3

## Análisis y Diseño

### 3.1 Introducción

En este capítulo se realiza el análisis y diseño de la propuesta de solución, definiendo como se desarrolla el Subsistema de modelos matemáticos, componentes químicos y propiedades físico-químicas, a través de la realización de los Casos de Uso presentes en el proceso, se obtienen los diagramas de clases del análisis y diseño, así como los diagramas de interacción correspondientes.

### 3.2 Análisis

El modelo de análisis es un modelo de objetos conceptual, descrito con el lenguaje de los desarrolladores. Ayuda a refinar los requisitos y permite razonar sobre los aspectos internos del sistema, su objetivo fundamental es analizar con mayor profundidad los requerimientos, ofreciendo una especificación más precisa de los mismos, permitiendo a los desarrolladores comprender cómo debería darse forma al sistema. Un modelo de análisis estructura los requisitos de un modo que facilita su comprensión, su preparación, su modificación, y de forma general, su mantenimiento. Puede considerarse como una primera aproximación al modelo de diseño, por lo cual constituye una entrada fundamental cuando se da forma al sistema en el diseño y la implementación. [15]

#### 3.2.1 Diagrama de Clases del Análisis.

Los diagramas de clases de análisis representan los conceptos fundamentales del dominio del problema, y las relaciones entre las clases. Las clases del análisis se clasifican en Interfaz, de Control o Entidad.

- Clase Interfaz: Modela la interacción entre el sistema y sus actores.
- Clase Entidad: Modela información que posee una larga vida y es a menudo persistente.

- Clase Control: Representa coordinación, secuencia, transacciones, y control de otros objetos.

Los diagramas de clases del análisis que se representan a continuación corresponden con los Casos de Uso críticos, para consultar los diagramas de clases del análisis de los Casos de Uso restantes, ver el Anexo 2

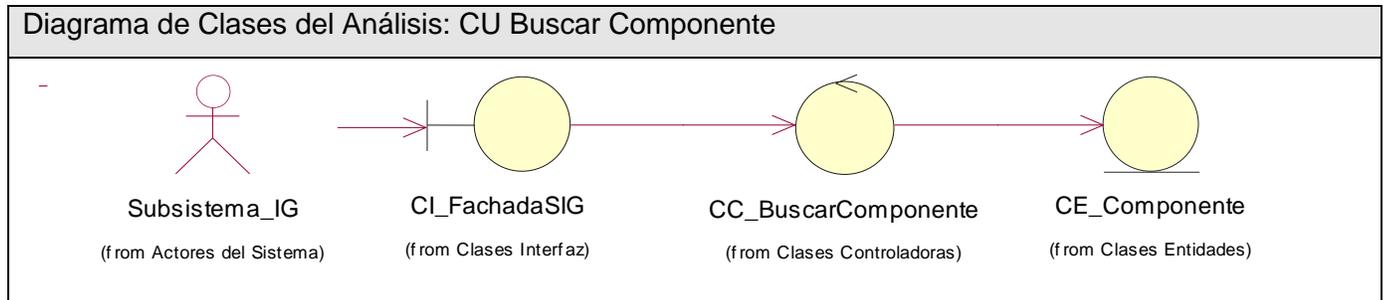


Figura 3 Diagrama de Clases del Análisis Caso de Uso: Buscar Componente

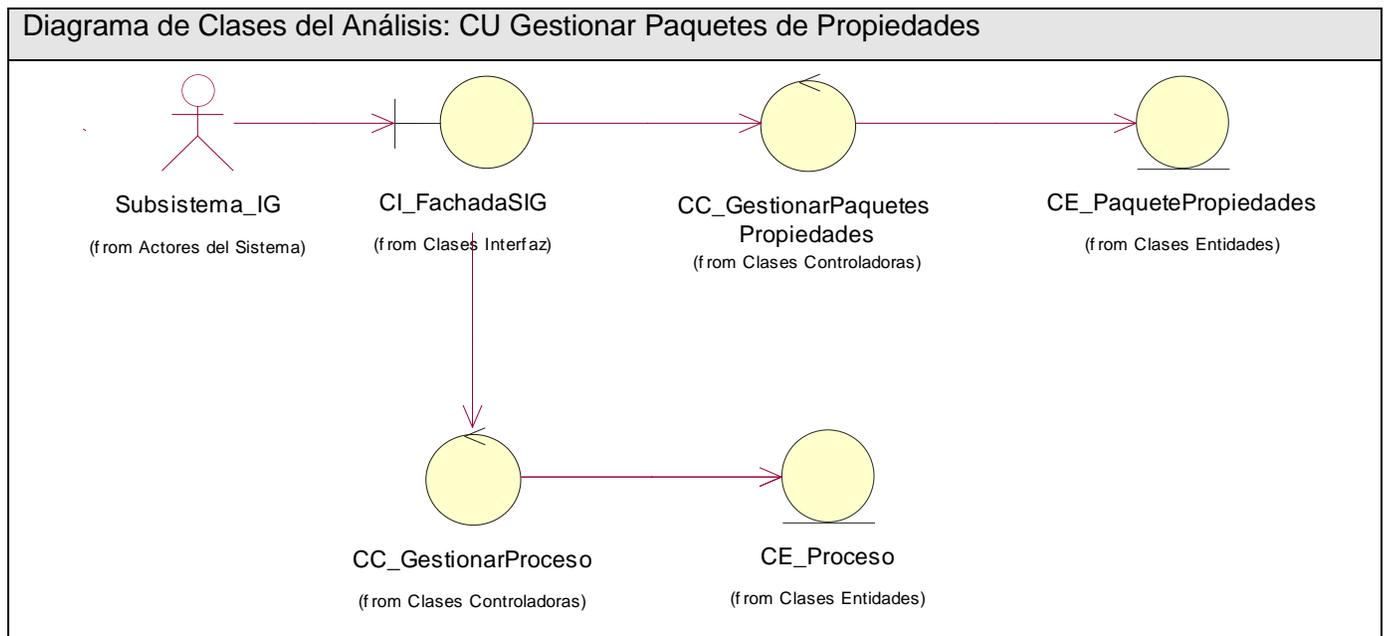


Figura 4 Diagrama de Clases del Análisis Caso de Uso: Gestionar Paquetes de Propiedades

Diagrama de Clases del Análisis: CU Gestionar Unidades de Operación

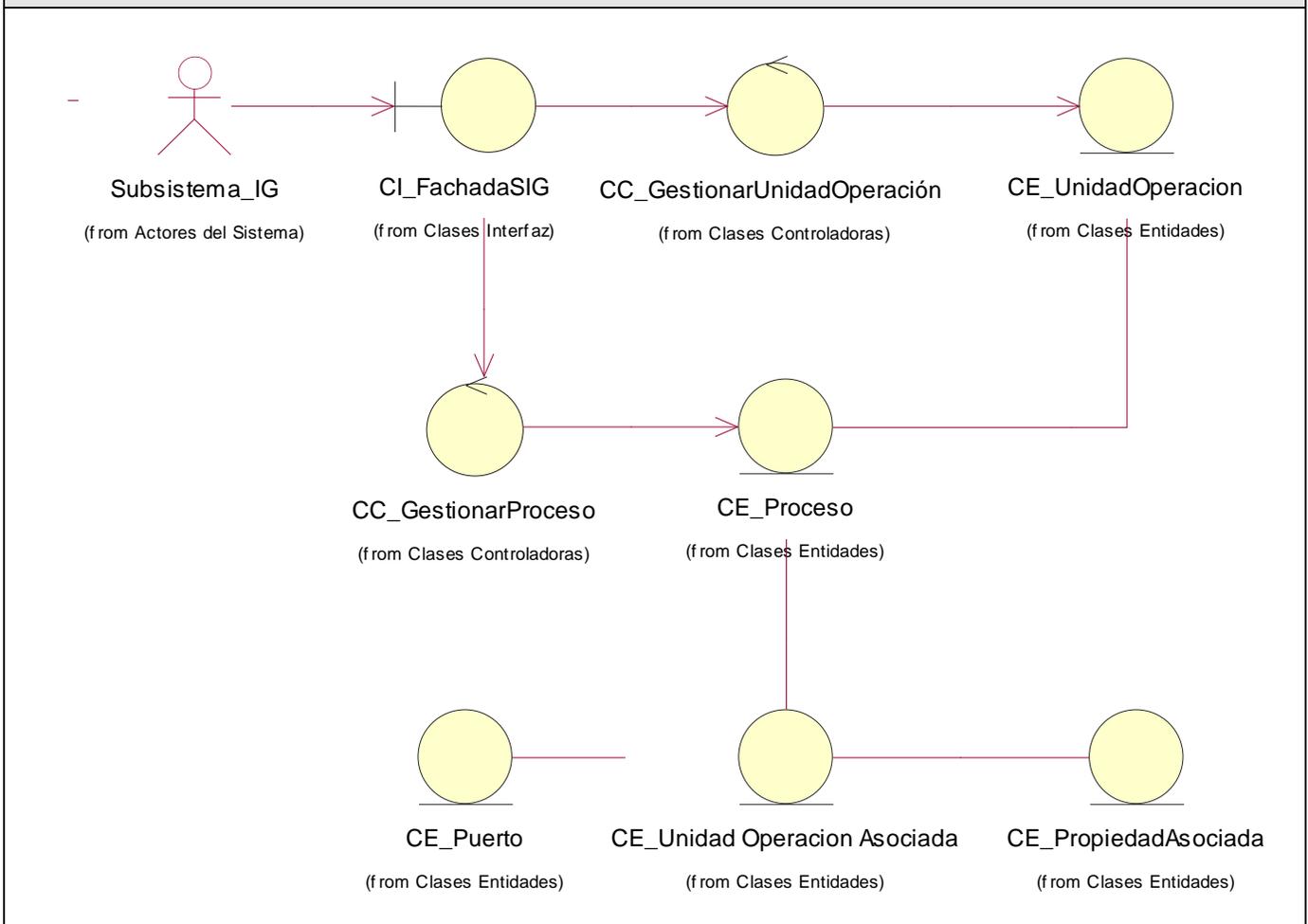


Figura 5 Diagrama de Clases del Análisis Caso de Uso: Gestionar Unidades de Operación

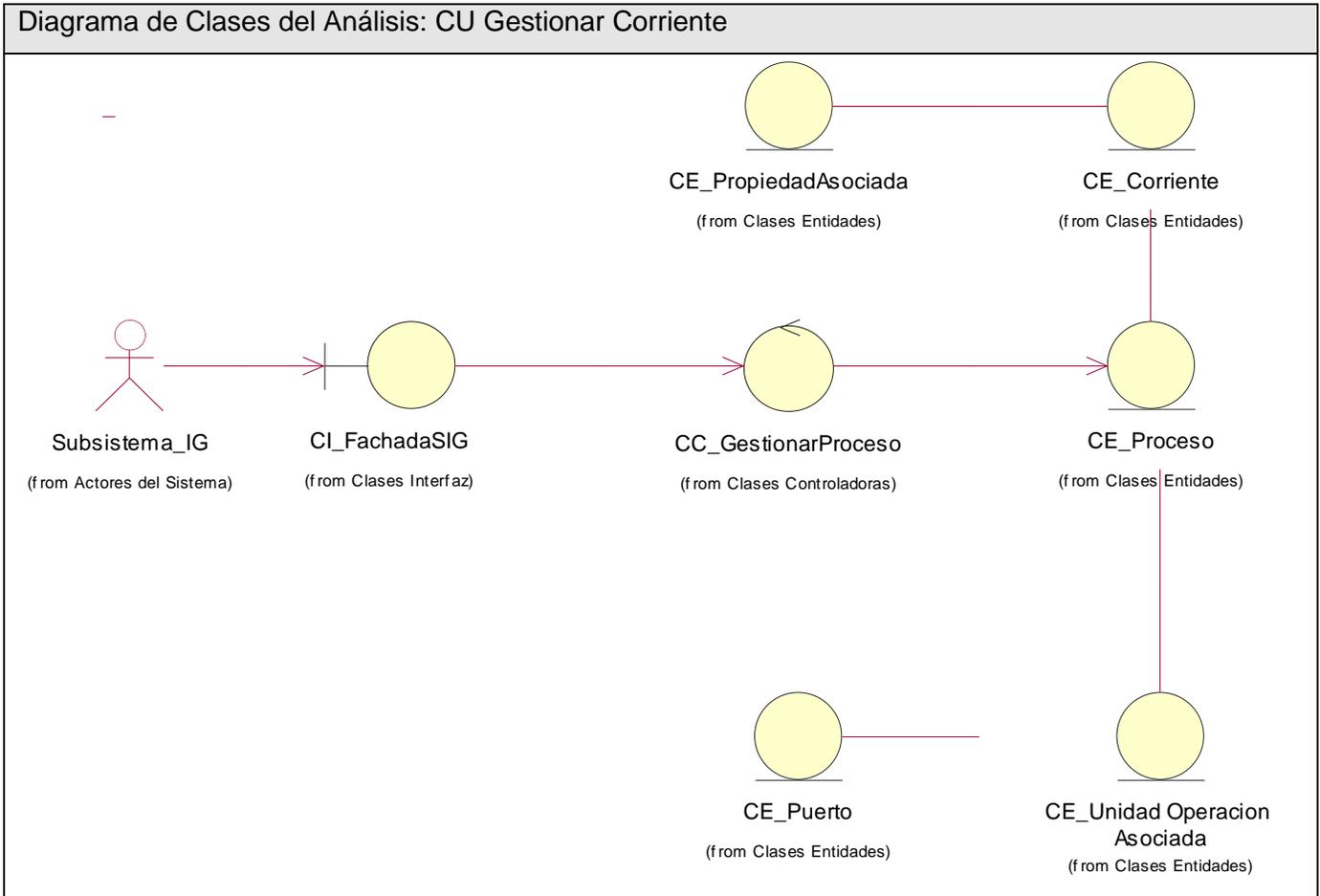


Figura 6 Diagrama de Clases del Análisis Caso de Uso: Gestionar Corriente

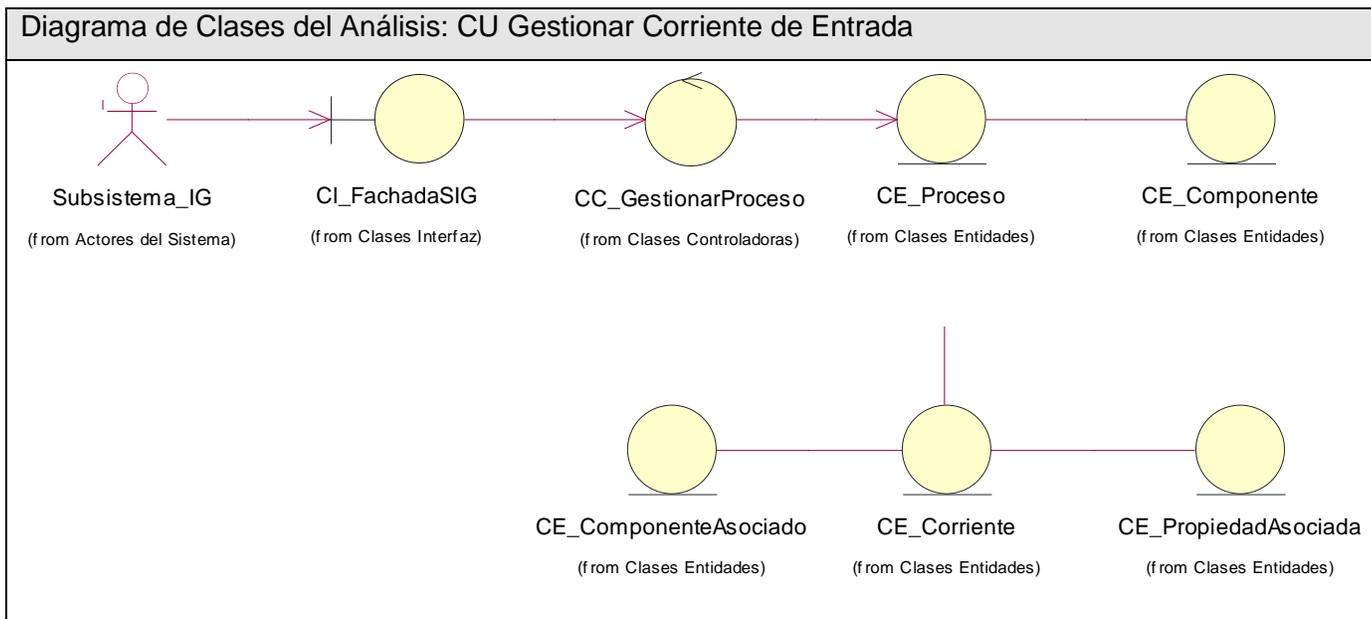


Figura 7 Diagrama de Clases del Análisis Caso de Uso: Gestionar Corriente de Entrada

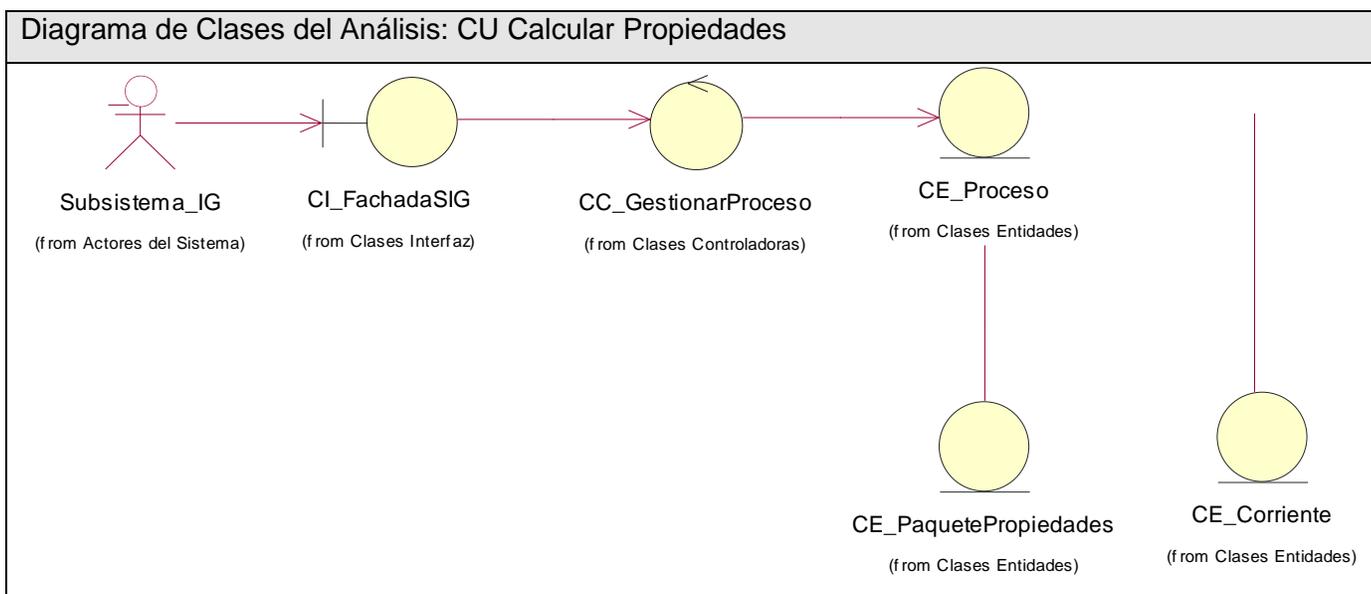


Figura 8 Diagrama de Clases del Análisis Caso de Uso: Calcular Propiedades

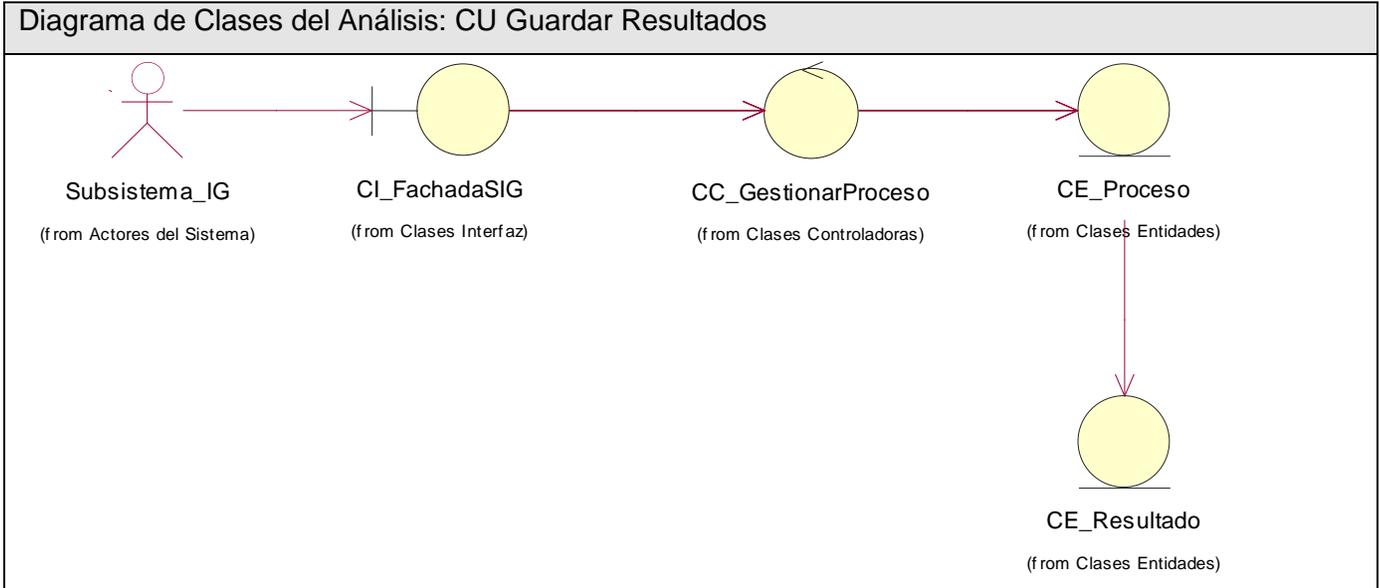


Figura 9 Diagrama de Clases del Análisis Caso de Uso: Guardar Resultados

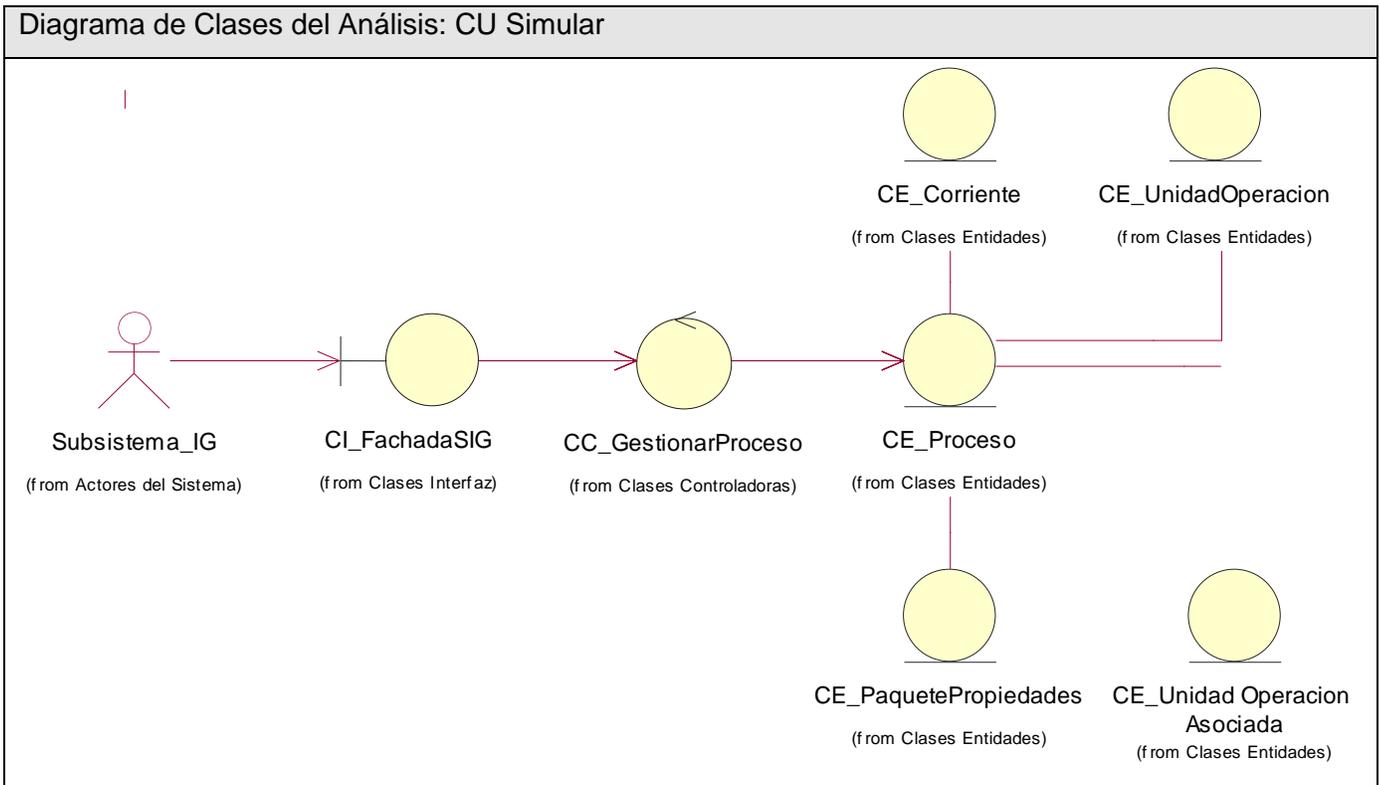


Figura 10 Diagrama de Clases del Análisis Caso de Uso: Simular

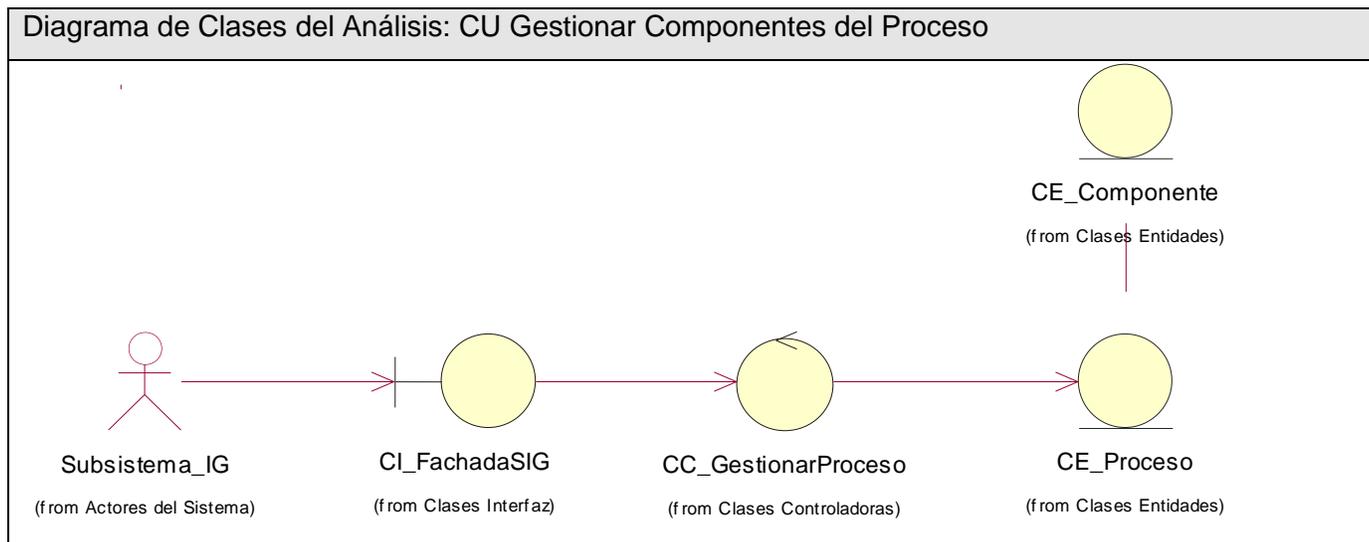


Figura 11 Diagrama de Clases del Análisis Caso de Uso: Gestionar Componentes del Proceso

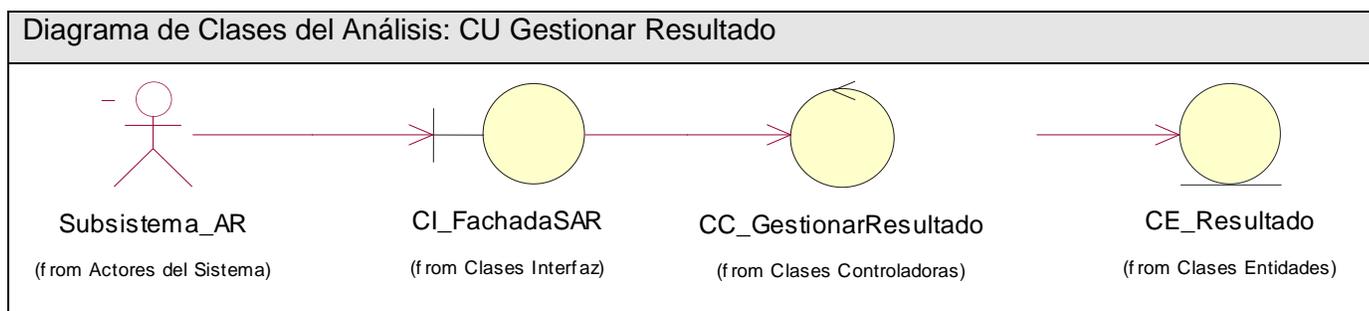


Figura 12 Diagrama de Clases del Análisis Caso de Uso: Gestionar Resultado

### 3.3 Diseño

El modelo de diseño al igual que el modelo de análisis es un modelo de objetos, pero en este caso que describe la realización física de los casos de uso centrándose en cómo los requisitos funcionales y no funcionales, junto con otras entradas relacionadas con el entorno de implementación, tienen impacto en el sistema a considerar. Los subsistemas de diseño y clases del diseño representan abstracciones del subsistema y componentes de la implementación del sistema, siendo estas abstracciones directas, y representando una sencilla correspondencia entre el diseño y la implementación. El modelo de diseño en sí constituye una entrada fundamental de las actividades de implementación. [15]

### 3.3.1 Patrones de diseño

Un patrón de diseño es una descripción de un problema y su solución, estos patrones de forma general constituyen parejas problema/solución a los cuales se le asigna un nombre y pueden ser empleados en otros contextos, con sugerencias sobre la manera de usarlo en situaciones nuevas. [16] De aquí que estos patrones, constituyan la base para la búsqueda de soluciones a problemas de diseño en el desarrollo de software.

#### Patrones GRASP

Los patrones GRASP (General Responsibility Assignment Software Patterns), patrones generales de software para asignar responsabilidades, describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones. Estos patrones constituyen el fundamento de cómo se diseñará el sistema. [16]

- **Experto:** Asignar una responsabilidad al experto en información: la clase que cuenta con la información necesaria para cumplir la responsabilidad. Este patrón se usa más que cualquier otro al asignar responsabilidades; es un principio básico que suele utilizarse en el diseño orientado a objetos; expresa simplemente la "intuición" de que los objetos hacen cosas relacionadas con la información que poseen.
- **Creador:** Asignar a la clase B la responsabilidad de crear una instancia de clase A en alguno de los siguientes casos (B agrega los objetos de A; B contiene a los objetos de A; B registra las instancias de los objetos de A; B utiliza específicamente los objetos de A; B tiene los datos de inicialización que serán transmitidos a A cuando este objeto sea creado). El patrón Creador guía la asignación de responsabilidades relacionadas con la creación de objetos, tarea muy frecuente en los sistemas orientados a objetos. El propósito fundamental de este patrón es encontrar un creador que debemos conectar con el objeto producido en cualquier evento. Al escogerlo como creador, se da soporte al bajo acoplamiento.
- **Alta Cohesión:** Asignar una responsabilidad de modo que la cohesión siga siendo alta. Una clase tiene responsabilidades moderadas en un área funcional y colabora con las otras para llevar a

cabo las tareas, la alta cohesión simplifica el mantenimiento y los mejoramientos. La ventaja que significa una gran funcionalidad también soporta un aumento de la capacidad de reutilización.

- **Bajo Acoplamiento:** Asignar una responsabilidad para mantener bajo acoplamiento. El Bajo Acoplamiento estimula asignar una responsabilidad de modo que su colocación no incremente el acoplamiento tanto que produzca los resultados negativos propios de un alto acoplamiento. Además soporta el diseño de clases más independientes, que reducen el impacto de los cambios, y también más reutilizables, que acrecienta la oportunidad de una mayor productividad.
- **Controlador:** Asignar la responsabilidad del manejo de un mensaje de los eventos de un sistema a una clase que represente por ejemplo el “sistema” global (controlador de fachada). La mayor parte de los sistemas reciben eventos de entrada externa, por lo cual, si se recurre a un diseño orientado a objetos, hay que elegir los controladores que manejen esos eventos de entrada.
- **Fabricación Pura:** Asigna un conjunto altamente cohesivo de responsabilidades a una clase artificial que no representa nada en el dominio del problema: una cosa inventada para dar soporte a una alta cohesión, un bajo acoplamiento y reutilización.

## Patrones GoF

Los patrones GoF, descritos en el libro Design Patterns y conocidos como “Gang of Four” (Pandilla de los Cuatro), son un conjunto de patrones de diseños de gran utilidad agrupados en tres tipos fundamentales; patrones de creación, patrones estructurales y patrones de comportamiento. [17] Durante el desarrollo del presente trabajo se usan los que se describe a continuación.

- **Fachada:** Es un patrón de tipo estructural sobre el ámbito de los objetos. Se le da el nombre de Fachada a la clase definida que ofrece una interfaz común a un conjunto de interfaces de un subsistema, este patrón define una interfaz de más alto nivel que facilita el uso de un subsistema. Entre sus beneficios se pueden destacar el hecho de facilitar a los clientes el uso de un subsistema, así como proporciona un acoplamiento débil entre un subsistema y los clientes.
- **Singleton:** Es un patrón de creación sobre el ámbito de los objetos. Asegura que una clase tenga una única instancia, un “singleton” (solitario), la entidad solo tendrá un punto de acceso global.

- **Método de Plantilla:** Es un patrón de comportamiento sobre el ámbito de las clases. Define el esqueleto (esquema, patrón) de un algoritmo en una operación, difiriendo algunos pasos a las subclases. Permite a las subclases redefinir ciertos pasos de un algoritmo sin cambiar la estructura del mismo.
- **Método Fábrica:** Es un patrón de creación sobre el ámbito de las clases. Un caso especial del patrón Método de Plantilla. Define una interfaz para crear un objeto, el método fábrica se encarga de crear una instancia pero permite a las subclases decidir la clase a instanciar.
- **Agente Virtual:** Es un patrón de creación sobre el ámbito de los objetos. Un Agente Virtual es un agente que hace referencia inteligente a otro objeto (el sujeto real) que materializa este último al referenciarlo por primera vez, implementando una materialización por demanda. Es un objeto ligero que representa a un objeto real, el cual puede materializarse o no.

#### Otros patrones

- **Intermediario de base de datos:** Propone construir una clase que se encargue de materializar, desmaterializar y guardar un objeto en un objeto caché. Un intermediario de base de datos es una clase de Fabricación Pura que soporta los patrones de Alta Cohesión y Bajo Acoplamiento.
- **Administración de Caché:** Propone asignar a los intermediarios de bases de datos la responsabilidad de dar mantenimiento a su caché. En caso de utilizar un intermediario diferente con cada objeto persistente, el intermediario habrá de darle mantenimiento a su propia caché.

#### 3.3.2 Diagramas de clases del diseño

Los diagramas de clases muestran un conjunto de interfaces, clases, colaboraciones y relaciones. A continuación se muestran los diagramas de clases del diseño de los casos de uso críticos, para ver los diagramas de los casos de uso restantes consultar el Anexo 3.

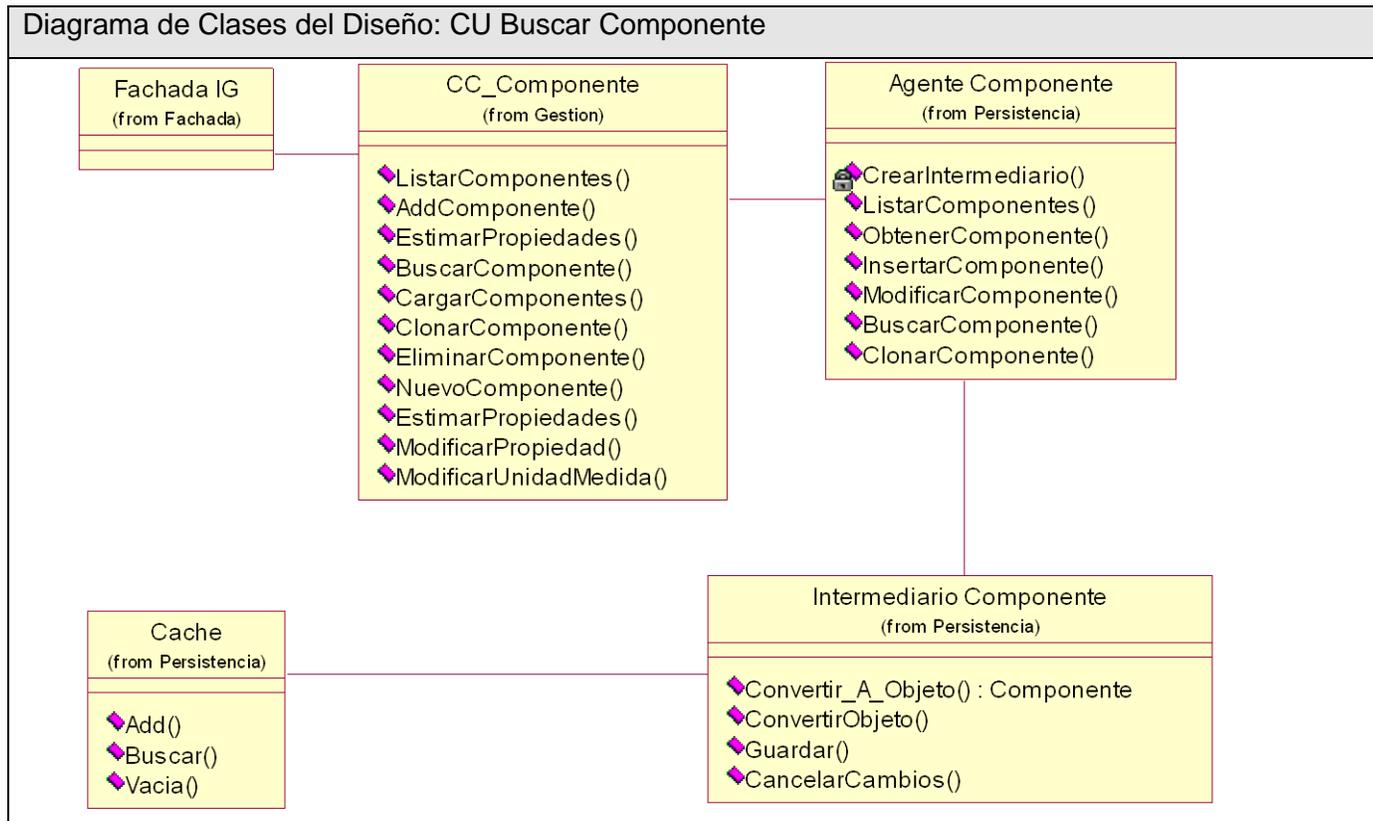


Figura 13 Diagrama de Clases del Diseño Caso de Uso: Buscar Componente

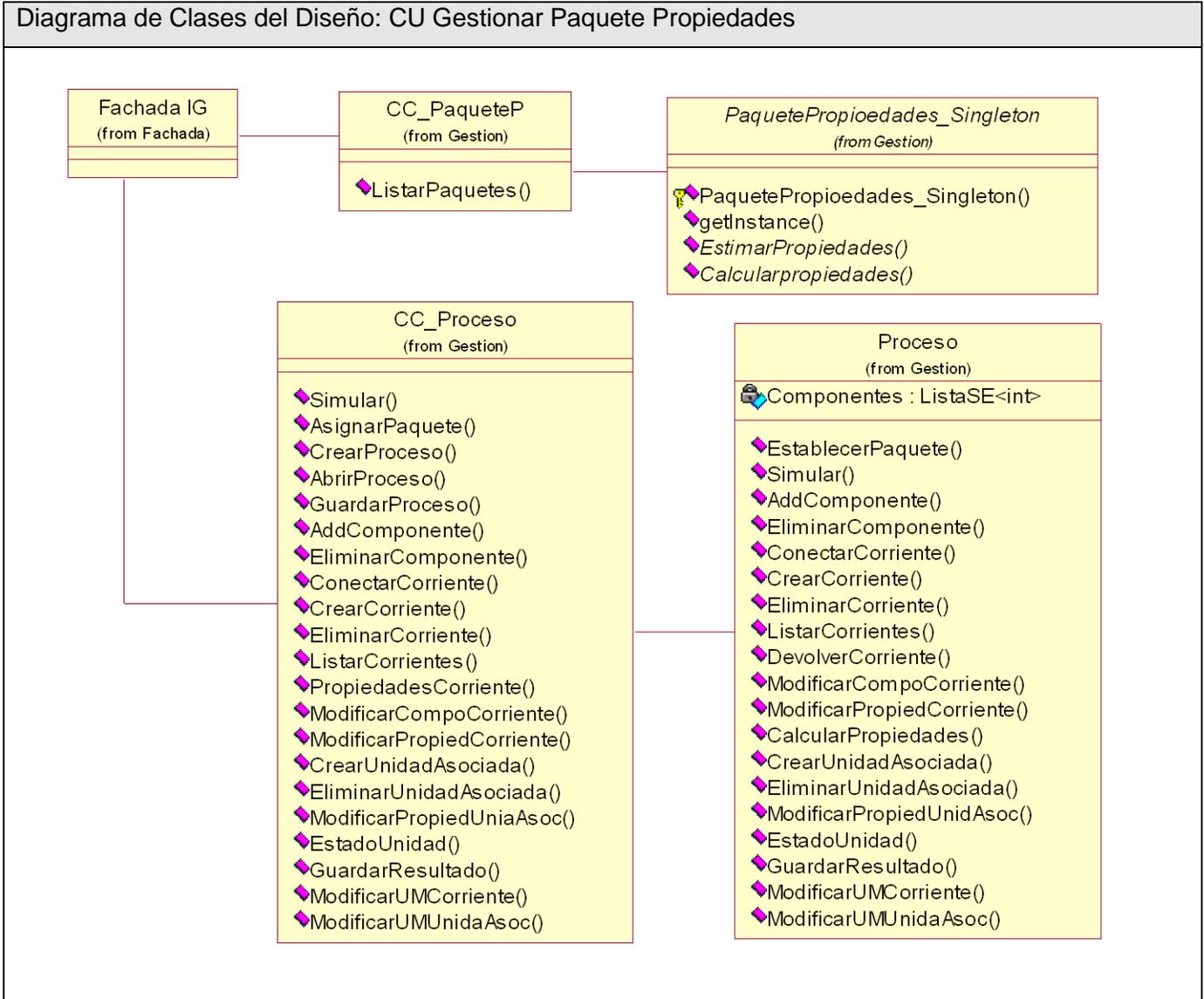


Figura 14 Diagrama de Clases del Diseño Caso de Uso: Gestionar Paquete Propiedades

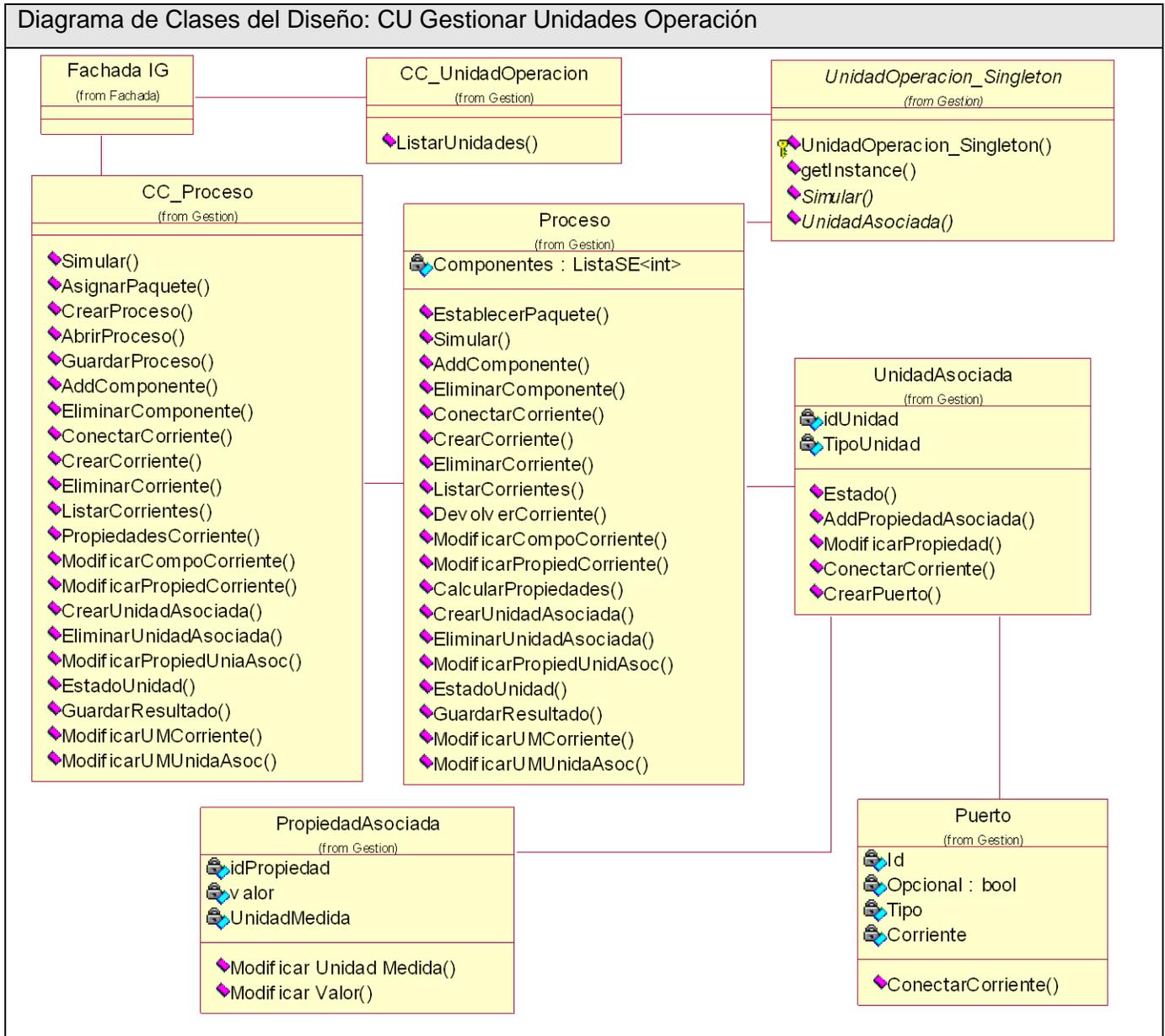


Figura 15 Diagrama de Clases del Diseño Caso de Uso: Gestionar Unidades Operación

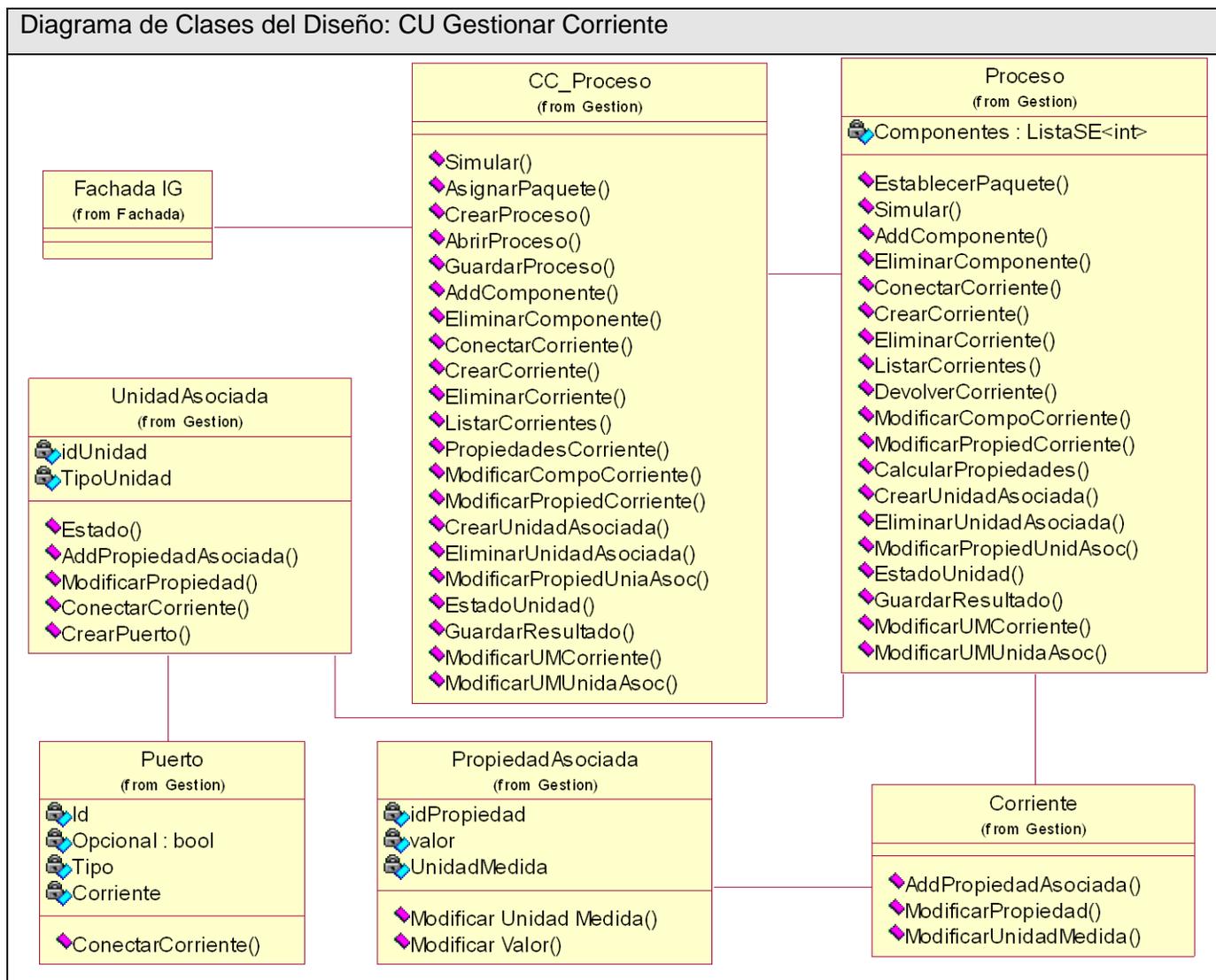


Figura 16 Diagrama de Clases del Diseño Caso de Uso: Gestionar Corriente

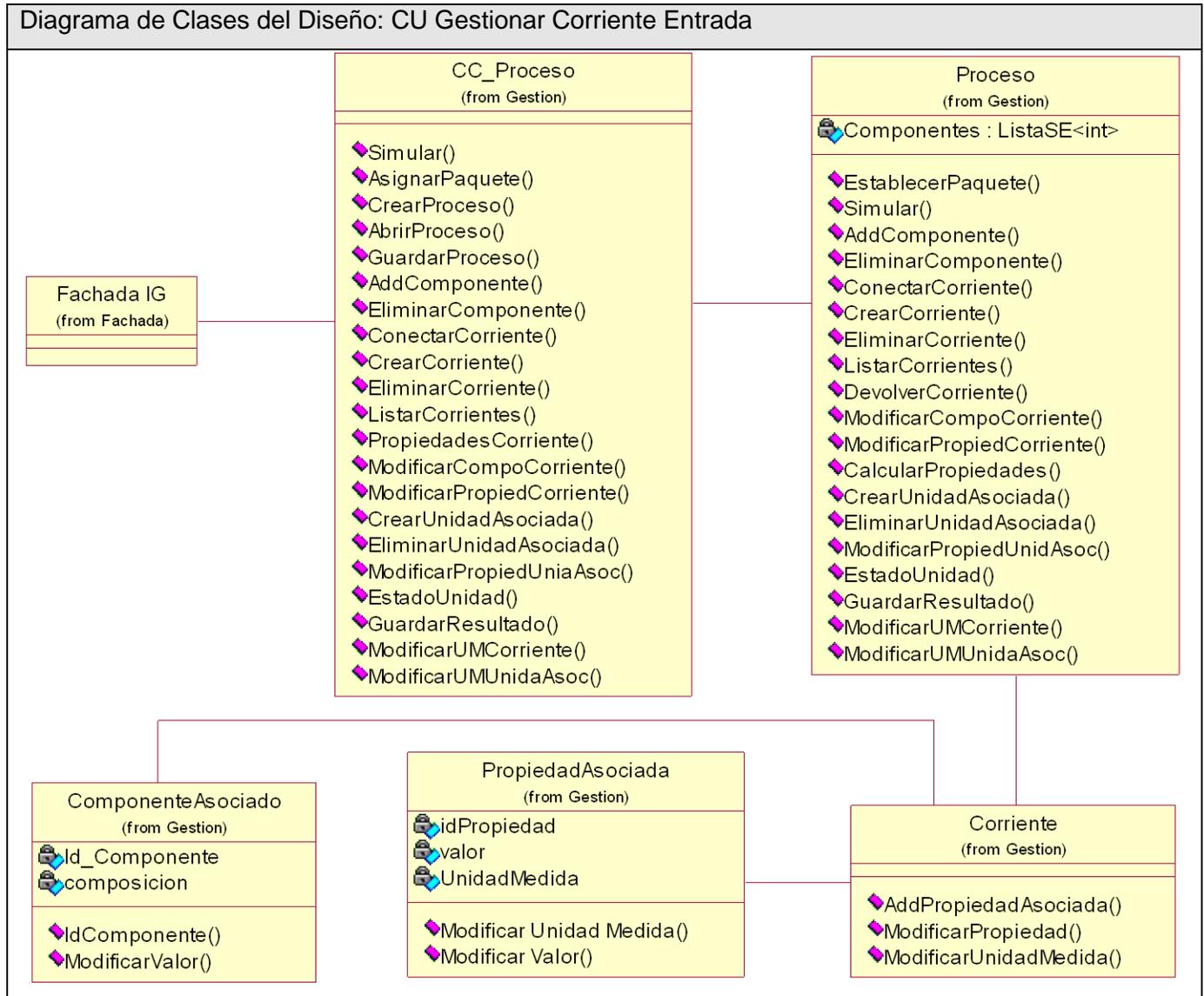


Figura 17 Diagrama de Clases del Diseño Caso de Uso: Gestionar Corriente Entrada

Diagrama de Clases del Diseño: CU Calcular Propiedades



Figura 18 Diagrama de Clases del Diseño Caso de Uso: Calcular Propiedades

Diagrama de Clases del Diseño: CU Guardar Resultados

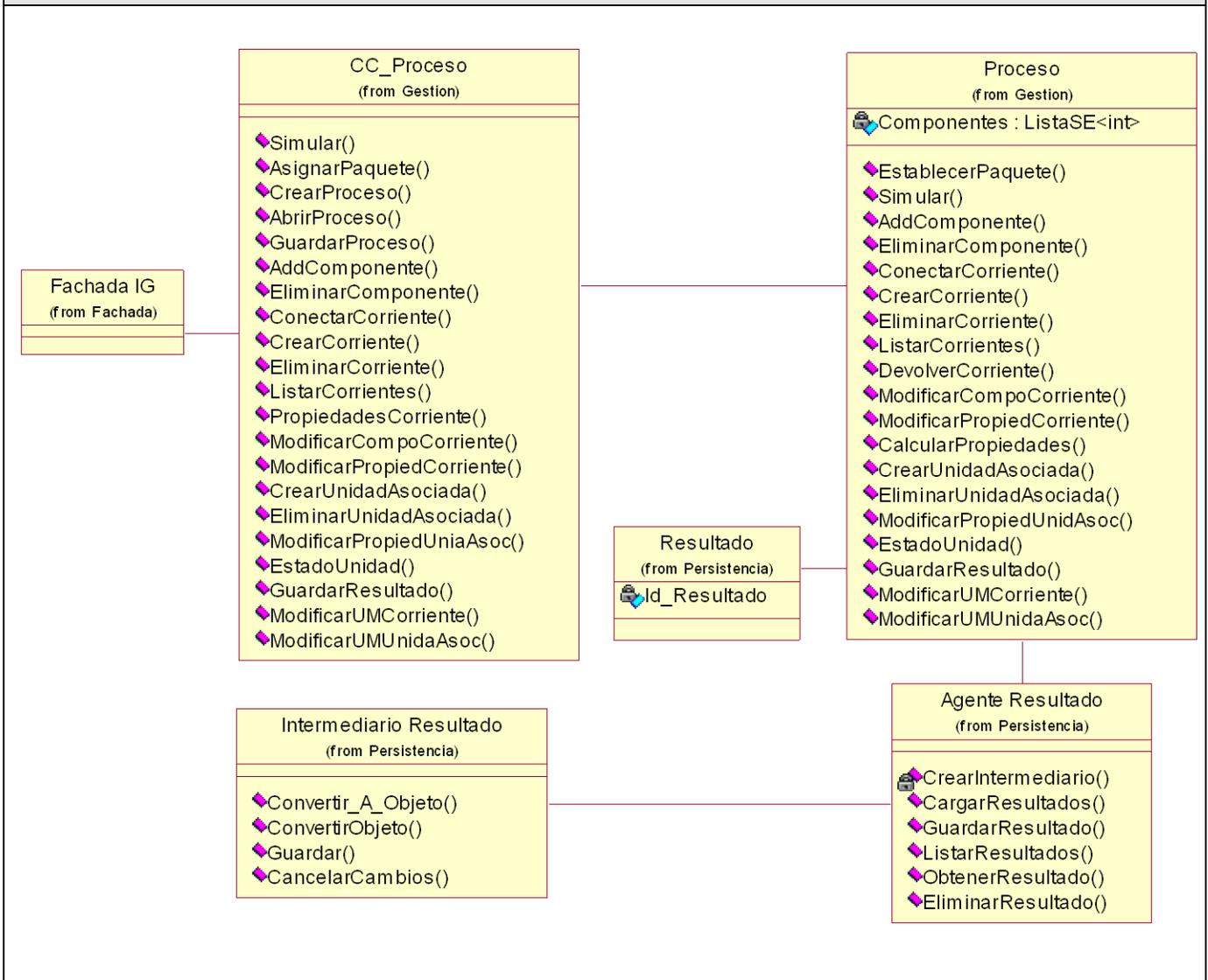


Figura 19 Diagrama de Clases del Diseño Caso de Uso: Guardar Resultados

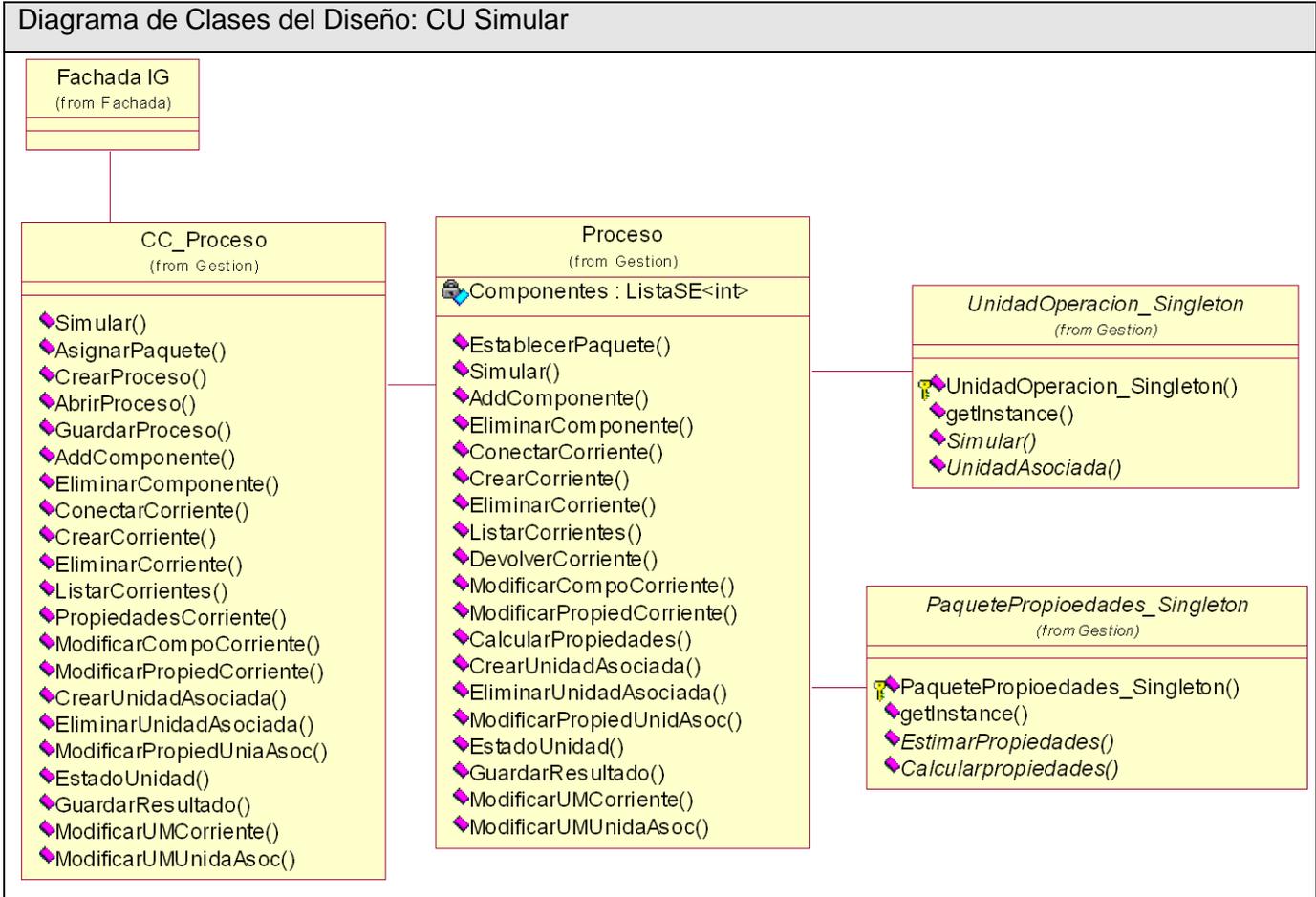


Figura 20 Diagrama de Clases del Diseño Caso de Uso: Simular

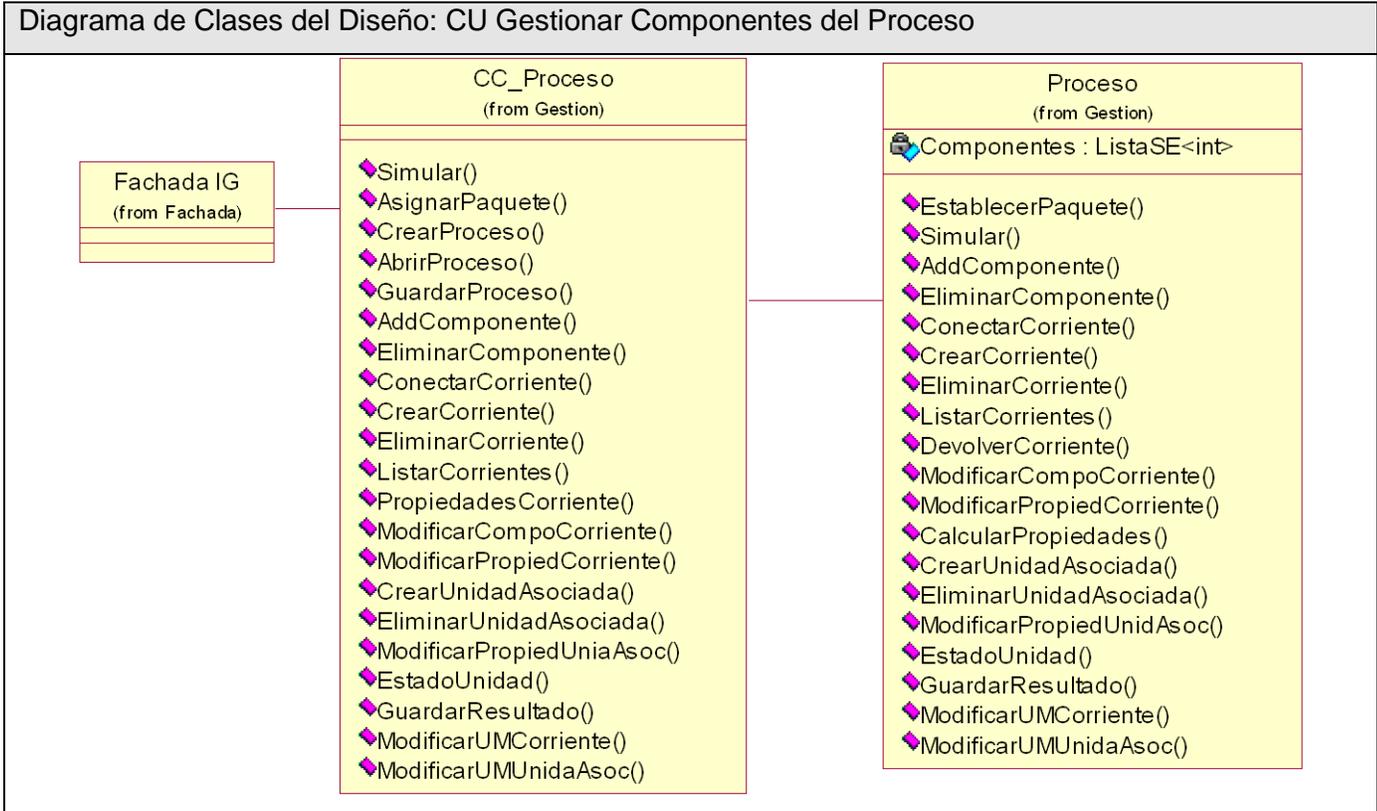


Figura 21 Diagrama de Clases del Diseño Caso de Uso: Gestionar Componentes del Proceso

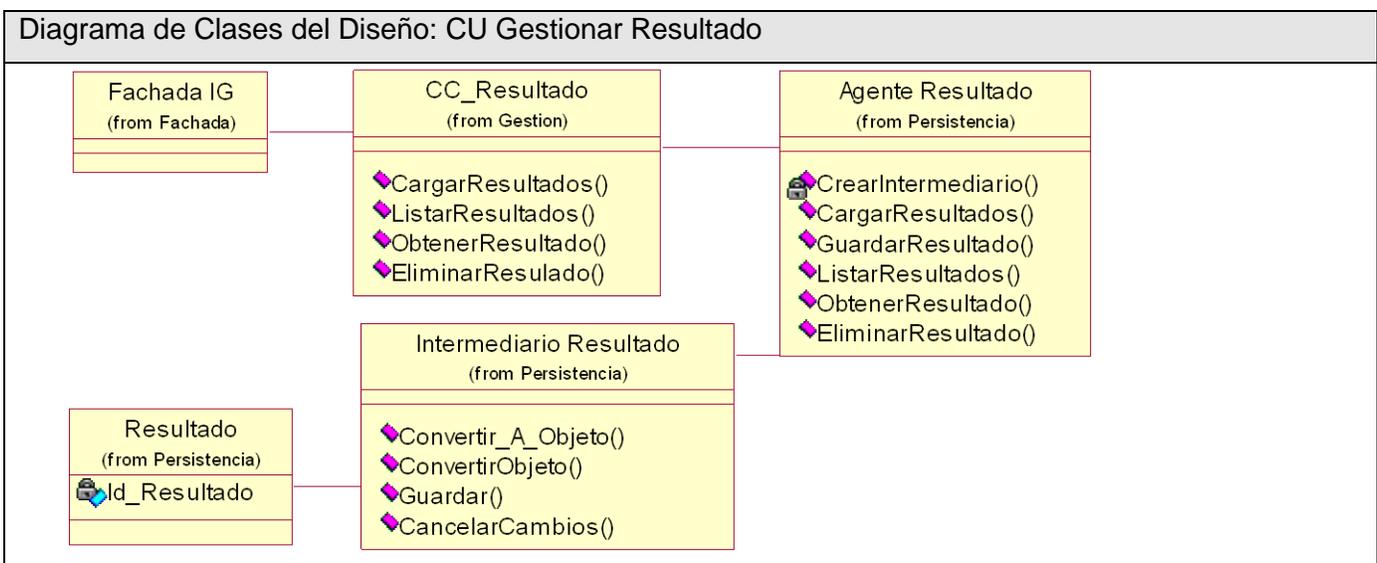


Figura 22 Diagrama de Clases del Diseño Caso de Uso: Gestionar Resultado

### 3.3.3 Diagramas de iteración del diseño

Los diagramas de interacción modelan cómo grupos de objetos colaboran para conseguir algún fin determinado. Entre estos diagramas podemos encontrar los diagramas de secuencia y los diagramas de colaboración. Por ser los diagramas de secuencia uno de los diagramas más efectivos para modelar interacción entre objetos en un sistema, el presente trabajo hace referencia a ellos, para ver los diagramas correspondientes a los escenarios por casos de uso, consultar el Anexo 4.

### 3.4 Conclusiones

En este capítulo se realizó el análisis y diseño del Subsistema de modelos matemáticos, componentes químicos y propiedades físico-químicas, presentándose los diagramas de clases obtenidos de la realización de los Casos de Uso del análisis, así como de la realización de los Casos de Uso del diseño para la solución propuesta, también se describieron un conjunto de patrones utilizados para resolver problemas de diseño en el desarrollo de software.

# Conclusiones Generales

Después de realizar un estudio del proceso a informatizar y haber pasado por los distintos flujos de trabajo, Negocio, Requerimientos y Análisis & Diseño que propone la metodología de desarrollo utilizada (RUP) se obtuvieron todos los artefactos pertenecientes a los mismos, a modo de conclusión para el presente trabajo se puede destacar que:

- Se realizó un estudio detallado para la simulación de procesos de la industria química.
- Se determinaron las principales funcionalidades del subsistema a diseñar.
- Se realizó el análisis y diseño del sistema aplicando patrones de diseño GRASP y GOF, además de utilizar otros patrones.
- Se desarrolló un correcto análisis y diseño del subsistema de modelos matemáticos, componentes químicos y propiedades físico-químico, el cual garantiza una adecuada implementación de este subsistema para un simulador de procesos de la industria química.

De esta forma se da cumplimiento a los objetivos generales trazados para la elaboración del trabajo de diploma, obteniendo como resultado final el análisis y diseño del subsistema de modelos matemáticos, componentes químicos y propiedades físico-químico.

# Recomendaciones

Los objetivos generales de este trabajo fueron alcanzados, pero se considera válido realizar las siguientes recomendaciones:

- Realizar los restantes flujos de trabajo que propone la metodología utilizada (RUP), llegando a implementar las funcionalidades propuestas para el subsistema de modelos matemáticos, componentes químicos y propiedades físico-químico.
- Continuar investigando sobre posibles funcionalidades a incorporar que amplíen las prestaciones del subsistema.
- Integrar el subsistema obtenido con los subsistemas Interfaz Gráfica y Análisis de los Resultados.

# Trabajos Citados

1. **Quintana Ramírez, Idelkys, Bretau Camejo, Osley y Ramírez Orozco, Raissel.** [www.informaticahabana.com](http://www.informaticahabana.com).  
[www.informaticahabana.com](http://www.informaticahabana.com). [En línea] 2006. [http://www.informaticahabana.com/evento\\_virtual/files/MUL036.pdf](http://www.informaticahabana.com/evento_virtual/files/MUL036.pdf).
2. **José Costa López, J. Costa, S. Cervera March, F. Cunill García, S. Esplugas Vidal, C. Mans Teixido, J. Mata Alvarez.** *Curso de Ingeniería Química*. 2002.
3. **Shannon, Robert E.** [www.mitecnologico.com](http://www.mitecnologico.com). [www.mitecnologico.com](http://www.mitecnologico.com). [Online]  
<http://www.mitecnologico.com/Main/DefinicionElImportanciaSimulacionEnIngenieria>.
4. **SCENNA, Dr. Nicolás José, y otros.** *MODELADO, SIMULACIÓN Y OPTIMIZACIÓN DE PROCESOS QUÍMICOS*.
5. **Pérez Ones, Osney y Gozá León, Osvaldo.** *Simulación Modular Secuencial de destilerías de alcohol con fines energéticos*. Ciudad de la Habana : s.n., 2005.
6. **León, DrC. Rolando Alfredo Hernández and Vitoria, DrC. Hector Eugenio Pérez de Alejo.** *Informatización de la ingeniería de proceso en la industria química cubana*. La Habana : s.n., 2005.
7. **Pérez Ones, Osney y Gozá León, Osvaldo.** *Simulación Modular Secuencial de destilerías de alcohol con fines energéticos*. Ciudad de la Habana : s.n., 2005.
8. **Ivar Jacobson, Grady Booch, James Rumbaugh.** *El proceso unificado de desarrollo de software, Volumen I*. La Habana, Editorial Félix Varela, 2004.
9. **Ivar Jacobson, Grady Booch, James Rumbaugh.** *El proceso unificado de desarrollo de software, Volumen I*. La Habana, Editorial Félix Varela, 2004. pp 4-12
10. <http://teleformacion.uci.cu>. <http://teleformacion.uci.cu>. [Online] [http://teleformacion.uci.cu/file.php/102/Curso\\_2008-2009/Materiales\\_Complementarios/Materiales\\_Complementarios\\_Conf\\_1/Metodologias\\_de\\_desarrollo\\_de\\_software.rar](http://teleformacion.uci.cu/file.php/102/Curso_2008-2009/Materiales_Complementarios/Materiales_Complementarios_Conf_1/Metodologias_de_desarrollo_de_software.rar).
11. **Apexnet.** <http://www.apexnet.com.ar>. <http://www.apexnet.com.ar>. [Online]  
<http://www.apexnet.com.ar/index.php/news/main/38/event=view>.

12. **Ivar Jacobson, Grady Booch, James Rumbaugh.** *El proceso unificado de desarrollo de software, Volumen I. La Habana, Editorial Félix Varela, 2004.* pp. 112-115.
13. **UCI.** <http://teleformacion.uci.cu>. <http://teleformacion.uci.cu>. [Online] [http://teleformacion.uci.cu/file.php/102/Curso\\_2008-2009/Materiales\\_Basicos/Materiales\\_Basicos\\_Conf\\_4/Conf\\_4\\_FT\\_Requerimiento.doc](http://teleformacion.uci.cu/file.php/102/Curso_2008-2009/Materiales_Basicos/Materiales_Basicos_Conf_4/Conf_4_FT_Requerimiento.doc).
14. **Ivar Jacobson, Grady Booch, James Rumbaugh.** *El proceso unificado de desarrollo de software, Volumen I. La Habana, Editorial Félix Varela, 2004.* pp. 35-40, 126-133.
15. **Ivar Jacobson, Grady Booch, James Rumbaugh.** *El proceso unificado de desarrollo de software, Volumen I. La Habana, Editorial Félix Varela, 2004.* pp. 165-253.
16. **Larman, Craig.** *UML y Patrones. La Habana : Félix Varela, 2004.* pp. 189-215. Vol. 1.
17. **Larman, Craig.** *UML y Patrones. La Habana : Félix Varela, 2004.* pp. 405-419. Vol. 2.

# Bibliografía

1. **Pérez Ones, Osney y Gozá León, Osvaldo.** *Simulación Modular Secuencial de destilerías de alcohol con fines energéticos.* Ciudad de la Habana : s.n., 2005.
2. **Scenna, Dr. Nicolás José.** [www.modeloingenieria.edu.ar](http://www.modeloingenieria.edu.ar). [www.modeloingenieria.edu.ar](http://www.modeloingenieria.edu.ar). [En línea] [http://www.modeloingenieria.edu.ar/libros/modeinge/modinge\\_f.htm](http://www.modeloingenieria.edu.ar/libros/modeinge/modinge_f.htm).
3. **Blanco, Luis Javier Lozano.** [www.upct.es](http://www.upct.es). [www.upct.es](http://www.upct.es). [En línea] 2007. [http://www.upct.es/contenido/estudios\\_postgrado/programas\\_posgrado/Asignaturas/SIMULACION\\_DE\\_PROCESOS\\_QUIMICO\\_S.pdf](http://www.upct.es/contenido/estudios_postgrado/programas_posgrado/Asignaturas/SIMULACION_DE_PROCESOS_QUIMICO_S.pdf).
1. **UCI.** <http://teleformacion.uci.cu>. <http://teleformacion.uci.cu>. [Online] [http://teleformacion.uci.cu/file.php/63/Materiales\\_de\\_Consulta\\_y\\_Conferencias\\_Tema\\_No\\_3/Power\\_UCI\\_Conferencia\\_No\\_6\\_MIC.ppt](http://teleformacion.uci.cu/file.php/63/Materiales_de_Consulta_y_Conferencias_Tema_No_3/Power_UCI_Conferencia_No_6_MIC.ppt).
4. **Vera, Alan Mestanza.** <http://simulacioniq.blogspot.com>. <http://simulacioniq.blogspot.com>. [Online] 5 9, 2008. <http://simulacioniq.blogspot.com/2008/05/chemcad-chemstations.html>.
5. **José Costa López, J. Costa, S. Cervera March, F. Cunill García, S. Esplugas Vidal, C. Mans Teixido, J. Mata Alvarez.** *Curso de Ingeniería Química.* 2002.
6. **León, DrC. Rolando Alfredo Hernández and Vitoria, DrC. Hector Eugenio Pérez de Alejo.** *Informatización de la ingeniería de proceso en la industria química cubana.* La Habana : s.n., 2005.
7. **Dr. Nicolás José SCENNA, Dr. Pío Antonio Aguirre, Dra. Sonia Judith Benz, Dr. Omar Juan Chiotti, Dr. H. José Espinosa, Ing. Marta Beatriz Ferrero, Dr. Jorge Marcelo Montagna, Ing. Miguel C. Mussati, Ing. Gustavo Alberto Pérez, Ing. Jorge Rodríguez, Dr. Héctor Enrique Salomone, Dr. Alejandro S. M. Santa Cruz, Dr. Enrique Eduardo Tarifa, Dr. Jorge Vega.** *MODELADO, SIMULACIÓN Y OPTIMIZACIÓN DE PROCESOS QUÍMICOS.*
8. **Camacho, Isaac Camacho.** *Desarrollo de software para predicción de propiedades termodinámicas de compuestos puros, mezclas binarias y multicomponentes en fase homogénea y heterogénea.* Puebla, México : s.n., 2003.

9. **Dra. María Jesús Ramos Marcos.** [www.uclm.es](http://www.uclm.es). [www.uclm.es](http://www.uclm.es). [Online] 2008.  
[www.uclm.es/PROFESORADO/mjramos/Simulaci3n/simuladores.ppt](http://www.uclm.es/PROFESORADO/mjramos/Simulaci3n/simuladores.ppt) .
10. <http://webdelprofesor.ula.ve>. <http://webdelprofesor.ula.ve>. [Online]  
<http://webdelprofesor.ula.ve/ingenieria/iedinzo/INTRODUCCION%20A%20LA%20SIMULACION%20DE%20PROCESOS.ppt>.
11. **Rodríguez, Néstor Hugo.** Informática Aplicada a la Ingeniería de Procesos I. [Online]  
<http://www.modeloingenieria.edu.ar/utnfrro/orientacionl/prog.pdf>.
12. **Ivar Jacobson, Grady Booch, James Rumbaugh.** El Proceso Unificado de desarrollo de Software. 2000.  
<http://bibliodoc.uci.cu/pdf/reg00060.pdf>
13. **Roger S. Pressman.** Ingeniería del Software. Un enfoque práctico.  
<http://bibliodoc.uci.cu/pdf/reg02689.pdf>
14. **Welicki, León and García, Fernando Cano.** <http://www26.brinkster.com>. <http://www26.brinkster.com>. [Online]  
<http://www26.brinkster.com/welicki/articles/PersonasMetodologiaISO4.pdf>.
15. <http://teleformacion.uci.cu>. <http://teleformacion.uci.cu>. [Online] [http://teleformacion.uci.cu/file.php/102/Curso\\_2008-2009/Materiales\\_Complementarios/Materiales\\_Complementarios\\_Conf\\_1/Metodologias\\_de\\_desarrollo\\_de\\_software.rar](http://teleformacion.uci.cu/file.php/102/Curso_2008-2009/Materiales_Complementarios/Materiales_Complementarios_Conf_1/Metodologias_de_desarrollo_de_software.rar).
16. <http://univafuadoo.files.wordpress.com>. <http://univafuadoo.files.wordpress.com>. [Online]  
<http://univafuadoo.files.wordpress.com/2008/12/metodologias-agiles-en-el-desarrollo-de-software.doc>.
17. <http://oness.sourceforge.net>. <http://oness.sourceforge.net>. [Online]  
<http://oness.sourceforge.net/proyecto/html/ch05s03.html>.
18. **MSc. Hugo F. Arboleda Jiménez.** <http://www.acis.org.co>. <http://www.acis.org.co>. [Online]  
<http://www.acis.org.co/index.php?id=551>.
19. <http://www.rational.com.ar>. <http://www.rational.com.ar>. [Online]  
<http://www.rational.com.ar/herramientas/roseenterprise.html>.
20. **Blanco, Rubén González and Tobalina, Sergio Pérez.** [www.lsi.upc.edu](http://www.lsi.upc.edu). [www.lsi.upc.edu](http://www.lsi.upc.edu). [Online] [www.lsi.upc.edu/~ese/web/documents/lab/0304Q2/lessons/lese-2/LESE-2%20-%20Introduccion%20a%20Rational%20Rose.ppt](http://www.lsi.upc.edu/~ese/web/documents/lab/0304Q2/lessons/lese-2/LESE-2%20-%20Introduccion%20a%20Rational%20Rose.ppt).
21. <http://www.rational.com.ar>. <http://www.rational.com.ar>. [Online]  
<http://www.rational.com.ar/herramientas/roseenterprise.html>.
22. **IBM.** <http://www-142.ibm.com>. <http://www-142.ibm.com>. [Online] [http://www-142.ibm.com/software/dre/ecatalog/Detail.wss?locale=es\\_ES&synkey=M221280M46834Z27&&S\\_TACT=none&S\\_CMP=none](http://www-142.ibm.com/software/dre/ecatalog/Detail.wss?locale=es_ES&synkey=M221280M46834Z27&&S_TACT=none&S_CMP=none).
23. <http://www.freedownloadmanager.org>. <http://www.freedownloadmanager.org>. [Online]  
[http://www.freedownloadmanager.org/es/downloads/Paradigma\\_Visual\\_para\\_UML\\_\(M%c3%8d\)\\_14720\\_p/](http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_(M%c3%8d)_14720_p/).

24. **Larman, Craig.** *UML y Patrones. La Habana : Félix Varela, 2004. Vol. 1.*
25. **Larman, Craig.** *UML y Patrones. La Habana : Félix Varela, 2004. pp. 405-419. Vol. 2.*

# Anexos

## Anexo 1. Descripciones textuales de Casos de Usos

Tabla 134 Caso de Uso: Gestionar Componente

<b>Caso de Uso:</b>	Gestionar Componente	
<b>Actores:</b>	Subsistema Interfaz Gráfica	
<b>Resumen:</b>	El Caso de Uso se inicia cuando el subsistema Interfaz Gráfica invoca algunas de las funciones para gestionar componentes ya sea cargar los componentes, insertar un nuevo componente, clonar uno existente, buscar, gestionar las propiedades o eliminar un componente personalizado, el subsistema ejecuta la función invocada y termina el CU.	
<b>Precondiciones:</b>		
<b>Referencias</b>	R.1, R.1.1, R.1.2, R.1.3, R.1.4, R.1.5, R.1.6, CU incluidos: Gestionar Propiedades	
<b>Prioridad</b>	Secundario	
<b>Flujo Normal de Eventos</b>		
<b>Acción del Actor</b>	<b>Respuesta del Subsistema</b>	
1. El subsistema Interfaz Gráfica invoca alguna de las funciones para gestionar componentes de la BD de componentes.	2. El subsistema procesa la solicitud.	
<b>Sección “Cargar componentes”</b>		
<b>Acción del Actor</b>	<b>Respuesta del Subsistema</b>	
3. El subsistema Interfaz Grafica crea una instancia del subsistema.	4. El subsistema carga los componentes de la BD de componentes.	
<b>Sección “Nuevo componente”</b>		
<b>Acción del Actor</b>	<b>Respuesta del Subsistema</b>	
5. El subsistema Interfaz Gráfica invoca la función nuevo componente e introduce los datos del	6. EL subsistema valida los datos 7. El subsistema crea el nuevo componente.	

nuevo componente.	8. El subsistema adiciona el nuevo componente a la BD de componentes.
<b>Sección “Gestionar propiedades”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Subsistema</b>
9. El subsistema Interfaz Gráfica invoca alguna de las funciones para gestionar propiedades de un componente químico.	10. Se ejecuta el CU gestionar propiedades.
<b>Sección “Buscar componente”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Subsistema</b>
11. El subsistema Interfaz Gráfica invoca la funciones buscar componente e introduce los parámetros de búsqueda.	12. Se ejecuta el CU Buscar componente. 13. El subsistema devuelve los componentes encontrados.
<b>Sección “Clonar componente”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Subsistema</b>
14. El subsistema Interfaz Gráfica invoca la funciones clonar componente e introduce el identificador del componente a clonar.	15. Se ejecuta el CU Buscar Componente. 16. El subsistema hace una copia del componente y la guarda en la BD de componentes con un nuevo identificador.
<b>Sección “Eliminar componente personalizado”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Subsistema</b>
17. El subsistema Interfaz Gráfica invoca la funciones eliminar componente e introduce el identificador del componente.	18. El subsistema elimina el componente de BD de componentes.
<b>Flujos Alternos</b>	
<b>Sección “Nuevo componente”</b>	
Línea 6: El subsistema lanza un mensaje de error porque los parámetros para crear el nuevo componente no son suficientes.	
<b>Sección “Buscar componente”</b>	
Línea 12: El subsistema lanza un mensaje de error porque los parámetros de búsqueda no dieron	

ningún resultado.	
<b>Poscondiciones</b>	La BD de componentes queda actualizada si se realiza algún cambio.

Tabla 145 Caso de Uso: Gestionar Biblioteca

<b>Caso de Uso:</b>	Gestionar Biblioteca	
<b>Actores:</b>	Subsistema Interfaz Gráfica	
<b>Resumen:</b>	El Caso de Uso se inicia cuando el subsistema Interfaz Gráfica invoca algunas de las funciones para gestionar bibliotecas ya sea cargar bibliotecas, listar las bibliotecas, crear o eliminar una biblioteca personalizada, listar, adicionar o eliminar componentes de una biblioteca, el subsistema ejecuta la función invocada y termina el CU. Los cambios efectuados en las bibliotecas no afectaran la BD de componentes.	
<b>Precondiciones:</b>		
<b>Referencias</b>	R.2, R.2.1, R.2.2, R.2.3, R.2.4, R.2.4.1, R.2.4.2, R.2.4.3, R.2.5	
<b>Prioridad</b>	Secundario	
<b>Flujo Normal de Eventos</b>		
<b>Acción del Actor</b>		<b>Respuesta del Subsistema</b>
1. El subsistema Interfaz Gráfica invoca alguna de las funciones para gestionar bibliotecas personalizadas.		2. El subsistema procesa la solicitud.
<b>Sección "Cargar bibliotecas"</b>		
<b>Acción del Actor</b>		<b>Respuesta del Subsistema</b>
3. El subsistema Interfaz Grafica invoca la función cargar bibliotecas.		4. El subsistema carga las bibliotecas de componentes.
<b>Sección "Listar biblioteca"</b>		
<b>Acción del Actor</b>		<b>Respuesta del Subsistema</b>
5. El subsistema Interfaz Gráfica invoca la función listar bibliotecas.		6. El subsistema devuelve una lista de las bibliotecas.

<b>Sección “Crear biblioteca”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Subsistema</b>
<b>7.</b> El subsistema Interfaz Gráfica invoca la función crear nueva biblioteca e introduce los datos necesarios.	<b>8.</b> El subsistema verifica que los datos sean validos. <b>9.</b> El subsistema crea una nueva biblioteca de componentes personalizada.
<b>Sección “Adicionar componente a la biblioteca”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Subsistema</b>
<b>10.</b> El subsistema Interfaz Gráfica invoca la función adicionar componente a biblioteca e introduce los datos necesarios.	<b>11.</b> El subsistema valida los datos suministrados. <b>12.</b> El subsistema adiciona el componente a la biblioteca.
<b>Sección “Eliminar componente de la biblioteca”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Subsistema</b>
<b>13.</b> El subsistema Interfaz Gráfica invoca la función eliminar componente de la biblioteca e introduce los datos necesarios.	<b>14.</b> El subsistema valida los datos suministrados. <b>15.</b> El subsistema elimina el componente de la biblioteca.
<b>Sección “Listar componentes una biblioteca”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Subsistema</b>
<b>16.</b> El subsistema Interfaz Gráfica invoca la función listar componentes de una biblioteca e introduce los datos necesarios.	<b>17.</b> El subsistema valida los datos suministrados. <b>18.</b> El subsistema devuelve una lista de los componentes de una biblioteca.
<b>Sección “Eliminar biblioteca”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Subsistema</b>
<b>19.</b> El subsistema Interfaz Gráfica invoca la función eliminar biblioteca e introduce el identificador	<b>20.</b> El subsistema verifica que existe la biblioteca.

de la biblioteca a eliminar.	21. El subsistema elimina la biblioteca.
<b>Flujos Alternos</b>	
<b>Sección “Crear biblioteca”</b>	
<b>Línea 6:</b> El subsistema lanza un error con la razón por la cual no se puede crear la biblioteca.	
<b>Sección “Adicionar componente a una biblioteca”</b>	
<b>Línea 9:</b> El subsistema lanza un error con la razón por la cual no se puede adicionar el componente a la biblioteca.	
<b>Sección “Eliminar componente de una biblioteca”</b>	
<b>Línea 12:</b> El subsistema lanza un error con la razón por la cual no se puede eliminar el componente de la biblioteca.	
<b>Sección “Listar componentes de una biblioteca”</b>	
<b>Línea 15:</b> El subsistema lanza un error con la razón por la cual no se puede listar los componentes de la biblioteca.	
<b>Sección “Eliminar biblioteca”</b>	
<b>Línea 18:</b> El subsistema lanza un error con la razón por la cual no se puede eliminar la biblioteca.	
<b>Poscondiciones</b>	La BD de componentes no sufre cambios, se actualiza la BD de bibliotecas.

Tabla 156 Caso de Uso: Gestionar Proceso

<b>Caso de Uso:</b>	Gestionar Proceso
<b>Actores:</b>	Subsistema Interfaz Gráfica
<b>Resumen:</b>	El Caso de Uso se inicia cuando el subsistema Interfaz Grafica invoca algunas de las funciones para gestionar procesos de simulación ya sea crear un nuevo proceso, salvar o abrir un proceso, el subsistema ejecuta la función invocada y termina el CU.
<b>Precondiciones:</b>	
<b>Referencias</b>	R.3, R.3.1, R.3.2, R.3.3
<b>Prioridad</b>	Secundario

<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Subsistema</b>
1. El subsistema Interfaz Grafica invoca alguna de las funciones para gestionar procesos de simulación.	2. El subsistema procesa la solicitud.
<b>Sección “Nuevo proceso ”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Subsistema</b>
3. El subsistema Interfaz Grafica invoca la función nuevo proceso e introduce los datos necesarios.	4. El subsistema crea el nuevo proceso.
<b>Sección “Salvar proceso”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Subsistema</b>
5. El subsistema Interfaz Gráfica invoca la función salvar proceso e introduce la localización donde desea guardar el proceso.	6. Chequea que la localización sea válida. 7. El subsistema guarda el proceso.
<b>Sección “Abrir proceso”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Subsistema</b>
8. El subsistema Interfaz Gráfica invoca la función abrir proyecto e introduce la ruta de acceso al proyecto.	9. Chequea que la localización sea válida 10. El subsistema abre el proyecto.
<b>Flujos Alternos</b>	
<b>Sección “Salvar proceso”</b>	
<b>Línea 6:</b> Si la localización no es válida lanza un mensaje de error diciendo que no se puede acceder a esa dirección.	
<b>Sección “Abrir proceso”</b>	
<b>Línea 9:</b> Si la localización no es válida lanza un mensaje de error diciendo que no se puede acceder a esa dirección.	

<b>Línea 10:</b> El subsistema lanza un error porque el fichero especificado no es compatible con el subsistema.	
<b>Poscondiciones</b>	Si se ejecuta la sección salvar proceso se habrá creado un fichero con la información referente al proceso en cuestión, en caso de la sección abrir proceso, se carga un proceso de simulación.

Tabla 167 Caso de Uso: Gestionar Propiedades

<b>Caso de Uso:</b>	Gestionar Propiedades	
<b>Actores:</b>	Subsistema Interfaz Gráfica	
<b>Resumen:</b>	El Caso de Uso se inicia cuando el subsistema Interfaz Grafica invoca algunas de las funciones para gestionar propiedades ya sea listar o cargar propiedades, modificar las propiedades de un componente personalizado o estimar las propiedades de un compuesto químico. El subsistema ejecuta la función invocada y termina el CU.	
<b>Precondiciones:</b>		
<b>Referencias</b>	R.1.3, R.1.3.1, R.1.3.2	
<b>Prioridad</b>	Secundario	
<b>Flujo Normal de Eventos</b>		
<b>Acción del Actor</b>		<b>Respuesta del Subsistema</b>
1. El subsistema Interfaz Gráfica invoca alguna de las funciones para gestionar propiedades.		2. El subsistema procesa la solicitud.
<b>Sección “Cargar propiedades”</b>		
<b>Acción del Actor</b>		<b>Respuesta del Subsistema</b>
3. El subsistema Interfaz Grafica crea una instancia del subsistema.		4. El subsistema carga las propiedades de los componentes.
<b>Sección “Listar propiedades”</b>		
<b>Acción del Actor</b>		<b>Respuesta del Subsistema</b>
5. El subsistema Interfaz Grafica invoca la función listar propiedades físico- químicas e introduce		6. El subsistema muestra un listado con las propiedades solicitadas.

los datos necesarios.	
<b>Sección “Modificar propiedades”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Subsistema</b>
<b>7.</b> El subsistema Interfaz Gráfica invoca la función modificar propiedades e introduce los datos necesarios (componente y propiedades).	<b>8.</b> El subsistema valida los datos. <b>9.</b> El subsistema modifica de propiedades del componente.
<b>Sección “Estimar propiedades”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Subsistema</b>
<b>10.</b> El subsistema Interfaz Gráfica invoca la función estimar propiedades e introduce los datos necesarios (componente y correlaciones).	<b>11.</b> El subsistema valida los datos. <b>12.</b> El subsistema estima las propiedades del compuesto y devuelve las propiedades estimadas.
<b>Flujos Alternos</b>	
<b>Línea 8:</b> El subsistema lanza un mensaje de error porque no se entraron los datos correctamente.	
<b>Línea 11:</b> El subsistema lanza un mensaje de error porque no se entraron los datos correctamente.	
<b>Poscondiciones</b>	La BD de componentes queda actualizada con los cambios realizados.

Tabla 178 Caso de Uso: Manipular Unidades de Medida

<b>Caso de Uso:</b>	Manipular Unidades de Medida
<b>Actores:</b>	Subsistema Interfaz Gráfica
<b>Resumen:</b>	El Caso de Uso se inicia cuando el subsistema Interfaz Gráfica invoca alguna de las funciones para manipular unidades de medida. El subsistema ejecuta y termina el CU.
<b>Precondiciones:</b>	
<b>Referencias</b>	R.7, R.7.1, R.7.2, R.7.3
<b>Prioridad</b>	Secundario
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Subsistema</b>

1. El subsistema Interfaz Gráfica invoca alguna de las funciones para manipular unidades de medida.	2. El subsistema procesa la solicitud.
<b>Sección “Modificar unidad de medida Componente”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Subsistema</b>
3. El subsistema Interfaz Grafica invoca la función cambiar unidad de medida de propiedad de un componente e introduce los datos necesarios.	4. El subsistema valida los datos. 5. El subsistema realiza las transformaciones correspondientes en las unidades de medida. 6. El subsistema modifica los valores y las unidades de medida.
<b>Sección “Modificar unidad de medida Unidad Asociada”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Subsistema</b>
7. El subsistema Interfaz Grafica invoca la función cambiar unidad de medida de una unidad asociada e introduce los datos necesarios.	8. El subsistema valida los datos. 9. El subsistema realiza las transformaciones correspondientes en las unidades de medida. 10. El subsistema modifica los valores y las unidades de medida.
<b>Sección “Modificar unidad de medida Corriente”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Subsistema</b>
11. El subsistema Interfaz Grafica invoca la función cambiar unidad de medida de un corriente e introduce los datos necesarios.	12. El subsistema valida los datos. 13. El subsistema realiza las transformaciones correspondientes en las unidades de medida. 14. El subsistema modifica los valores y las unidades de medida.
<b>Flujos Alternos</b>	
<b>Línea 4:</b> El subsistema lanza un mensaje de error porque los datos no están correctos. <b>Línea 8:</b> El subsistema lanza un mensaje de error porque los datos no están correctos. <b>Línea 12:</b> El subsistema lanza un mensaje de error porque los datos no están correctos.	
<b>Poscondiciones</b>	

## Anexo 2. Diagramas de Clases del Análisis

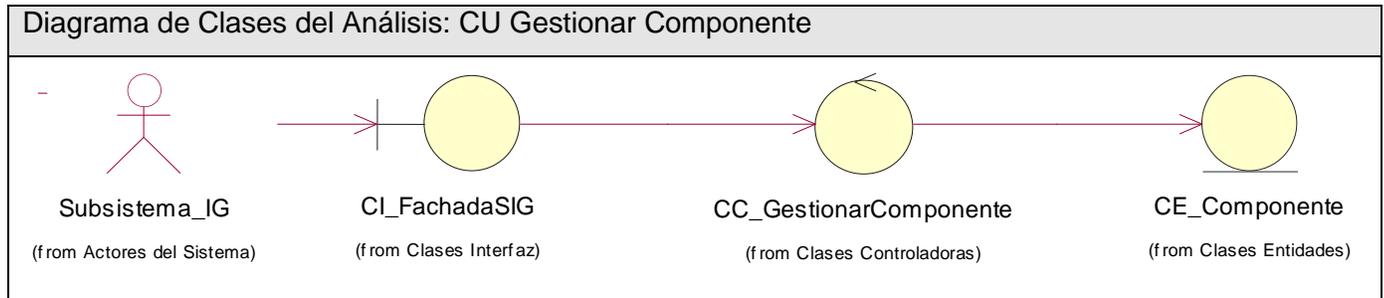


Figura 23 Diagrama de Clases del Análisis Caso de Uso: Gestionar Componente

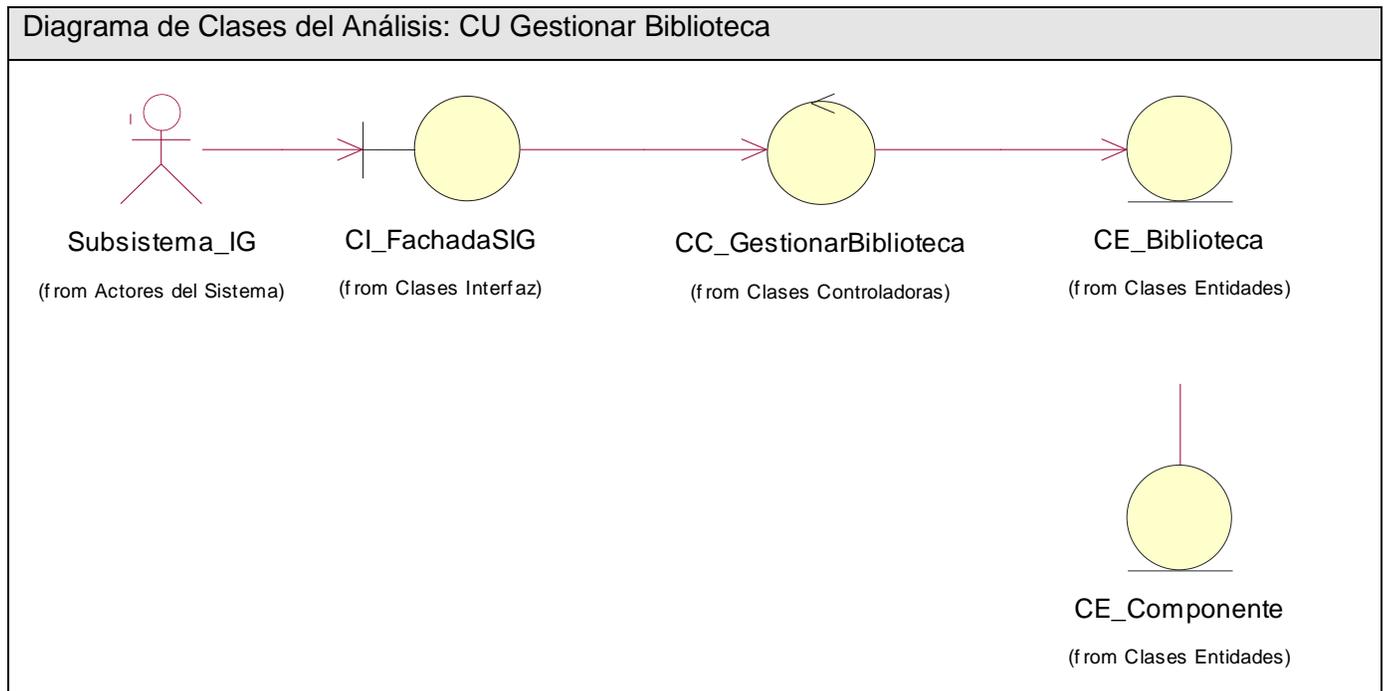


Figura 24 Diagrama de Clases del Análisis Caso de Uso: Gestionar Biblioteca

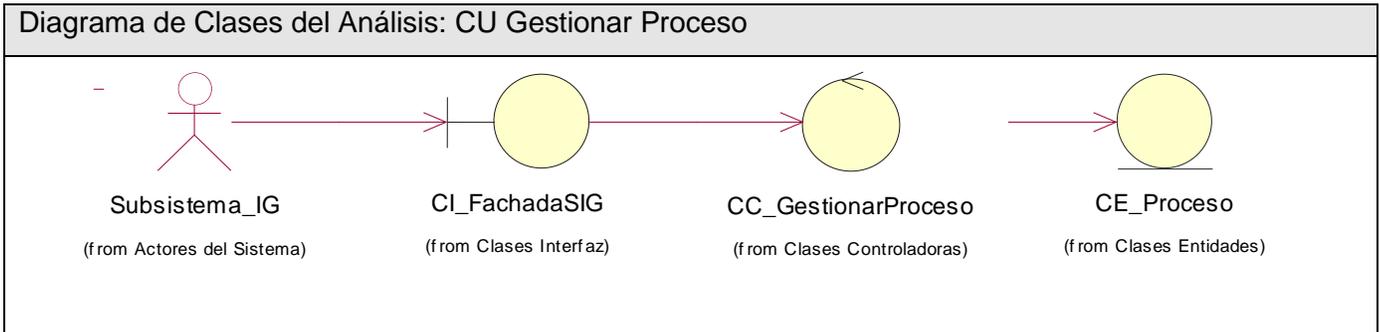


Figura 25 Diagrama de Clases del Análisis Caso de Uso: Gestionar Proceso

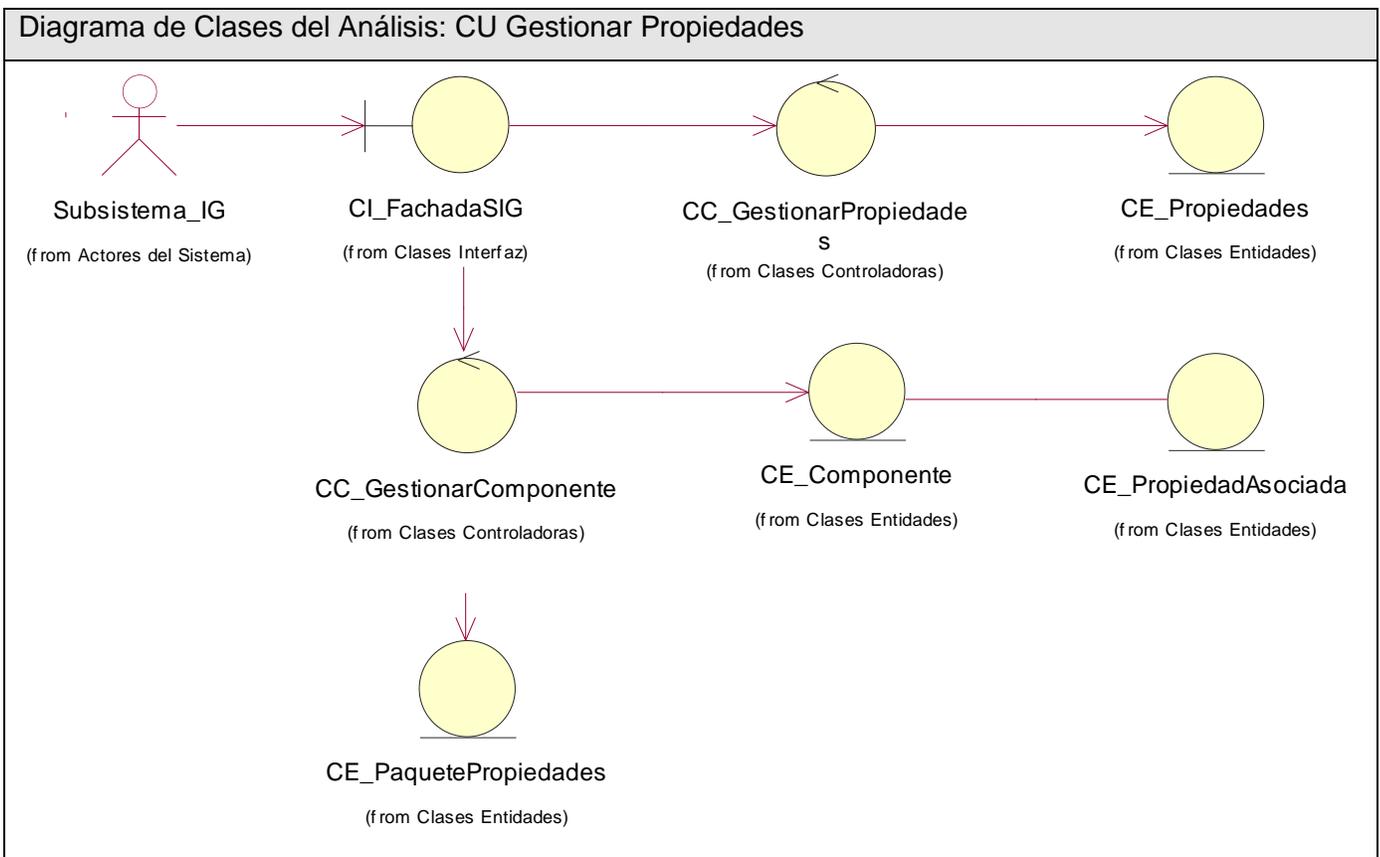


Figura 26 Diagrama de Clases del Análisis Caso de Uso: Gestionar Propiedades

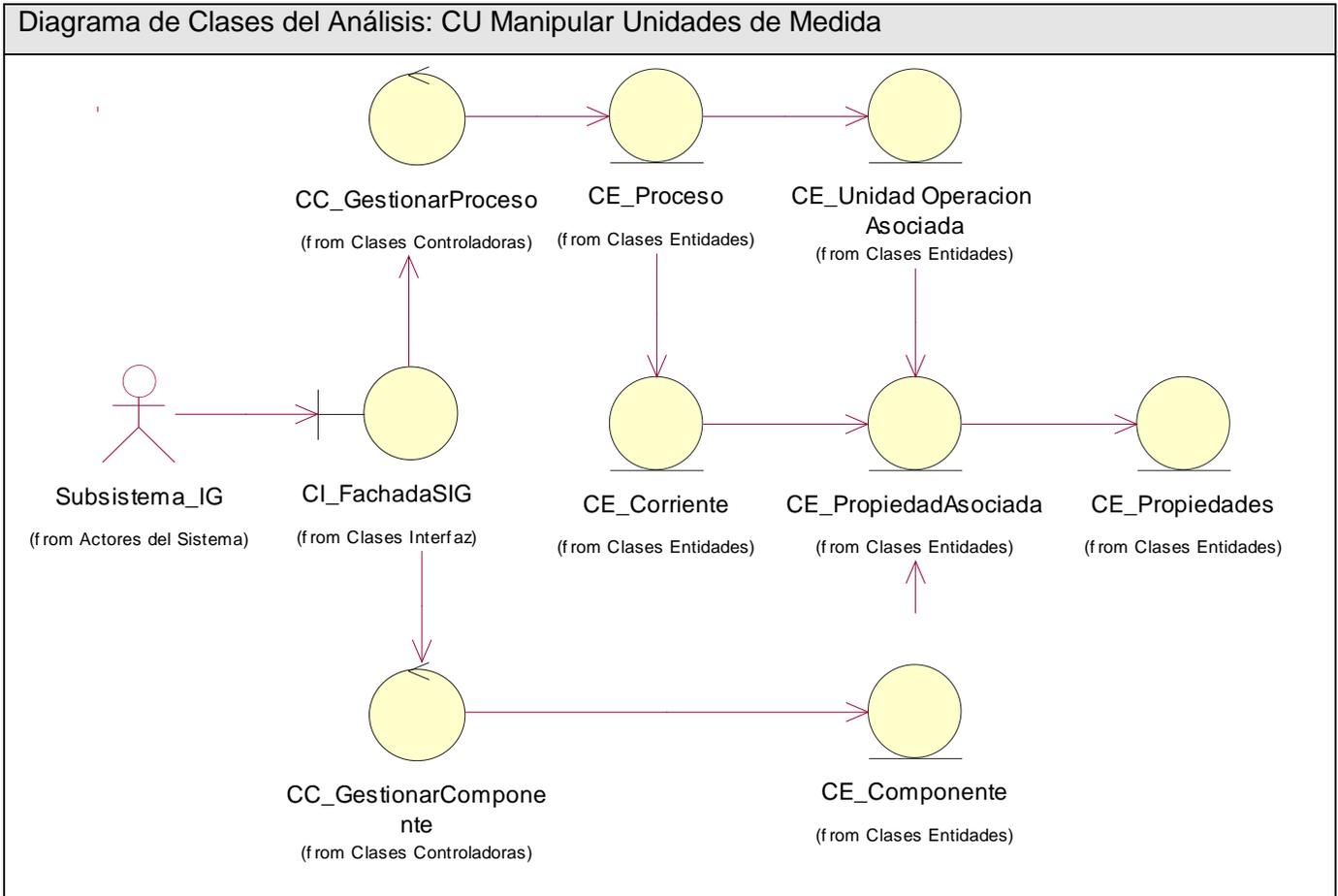


Figura 27 Diagrama de Clases del Análisis Caso de Uso: Manipular Unidades de Medida

### Anexo 3. Diagramas de Clases del Diseño

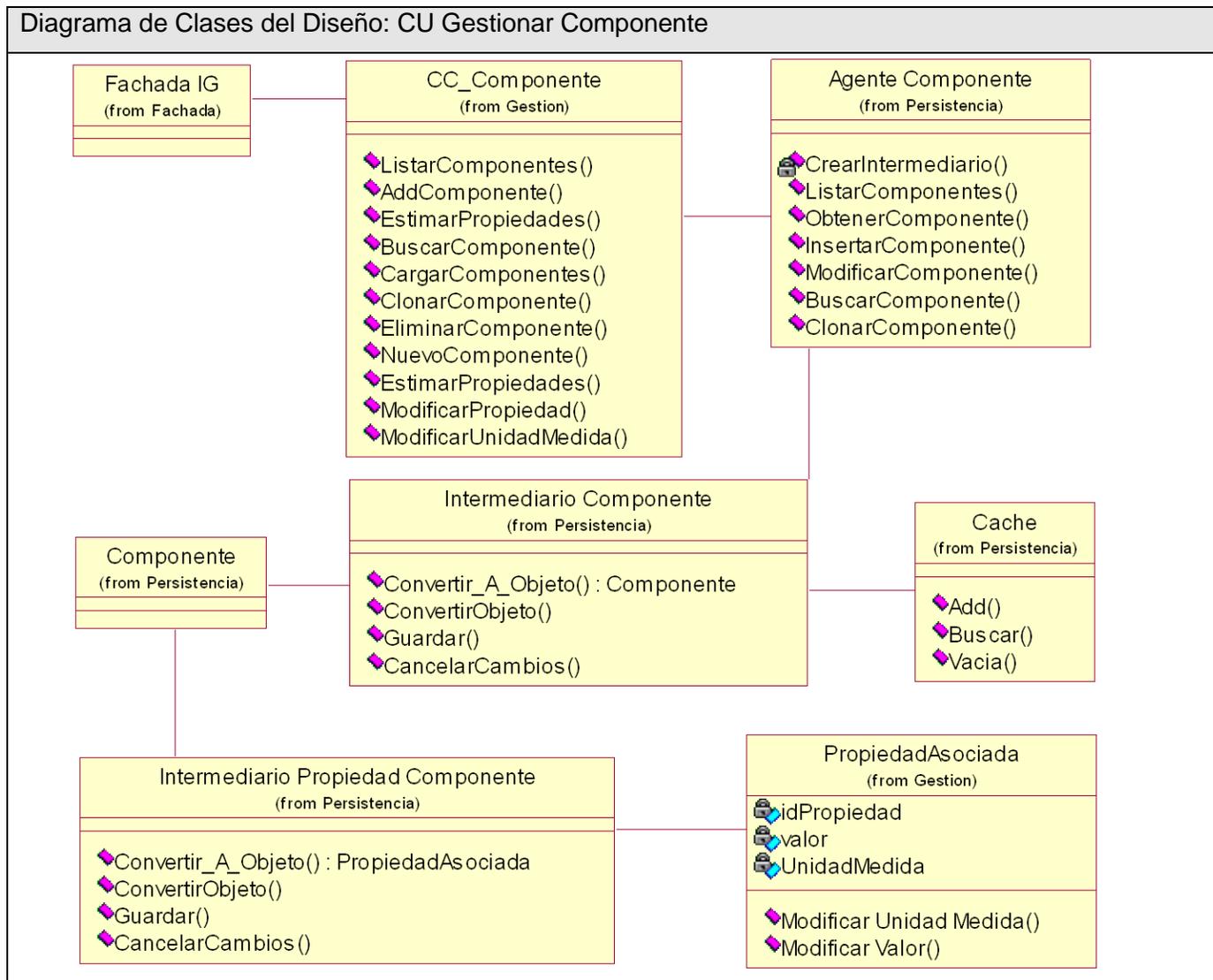


Figura 28 Diagrama de Clases del Diseño Caso de Uso: Gestionar Componente

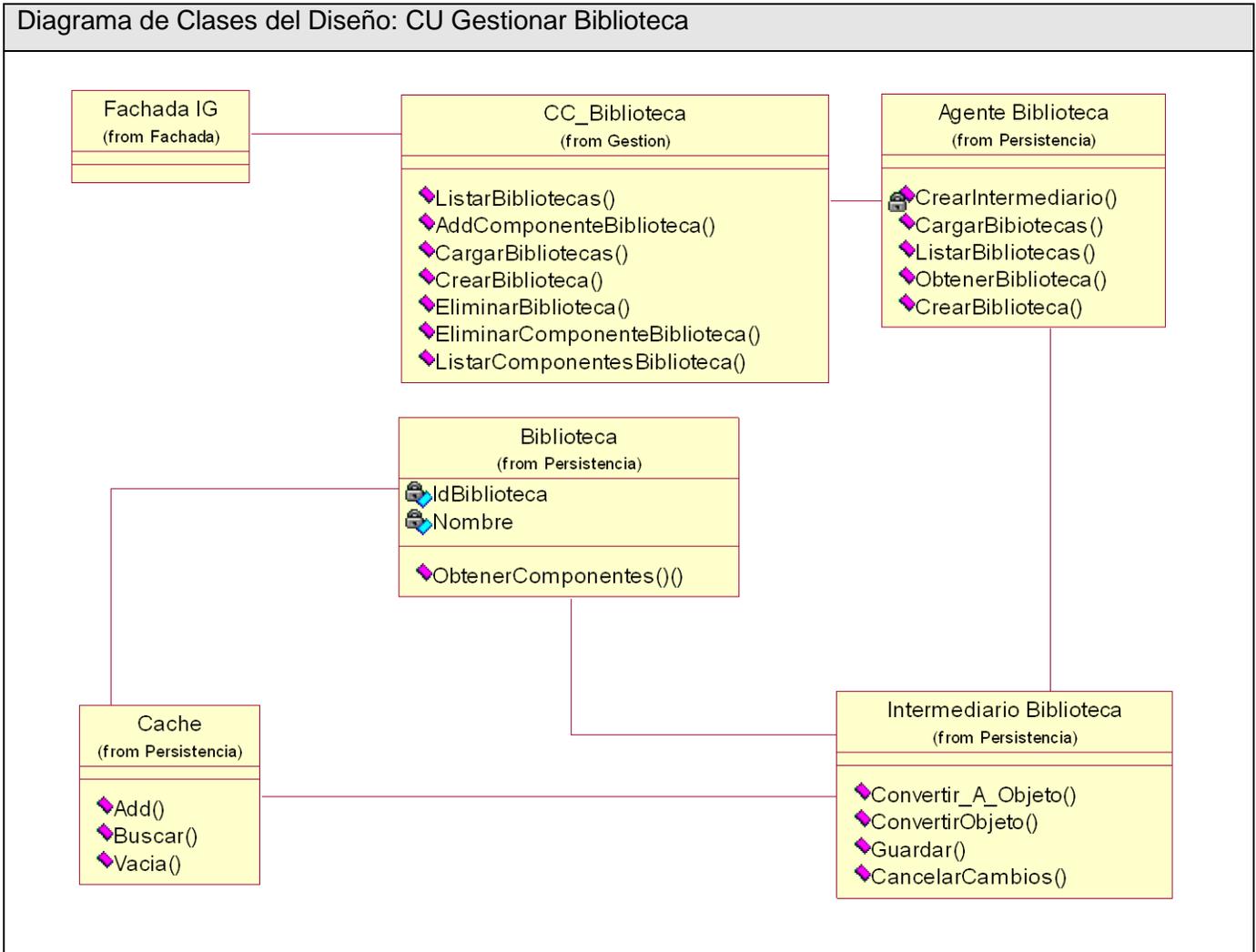


Figura 29 Diagrama de Clases del Diseño Caso de Uso: Gestionar Biblioteca

Diagrama de Clases del Diseño: CU Gestionar Proceso

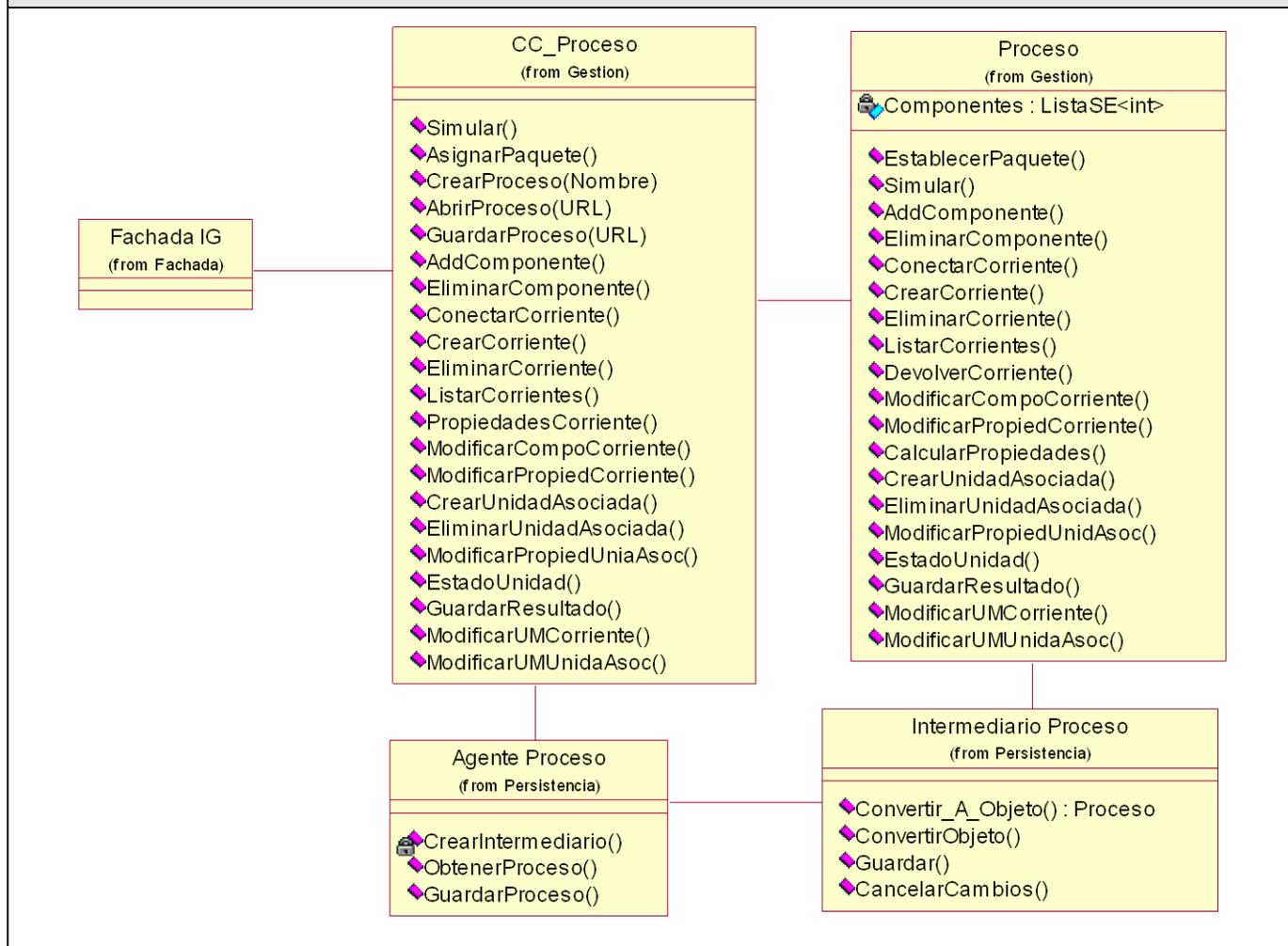


Figura 30 Diagrama de Clases del Diseño Caso de Uso: Gestionar Proceso

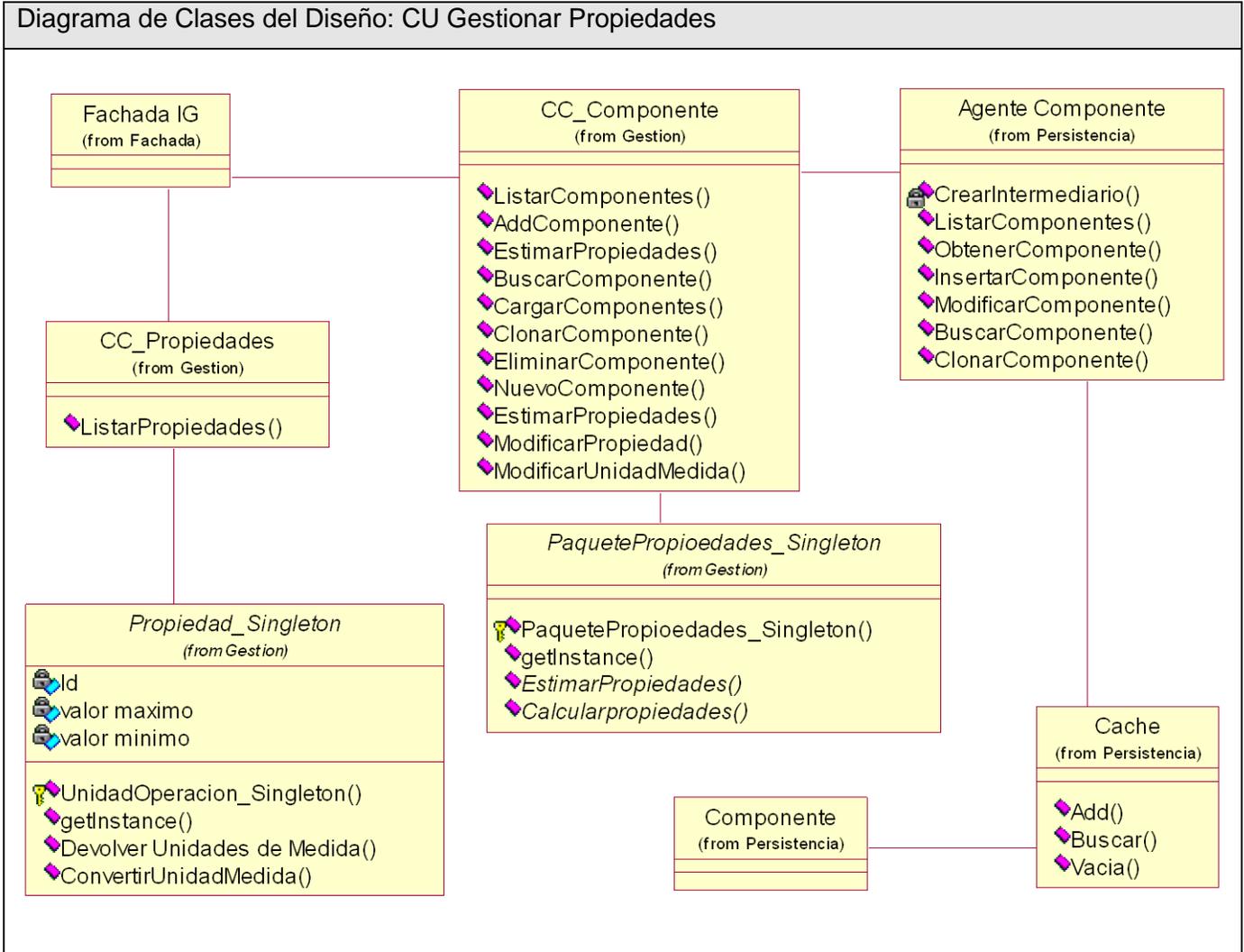


Figura 31 Diagrama de Clases del Diseño Caso de Uso: Gestionar Propiedades

Diagrama de Clases del Diseño: CU Manipular Unidades de Medida

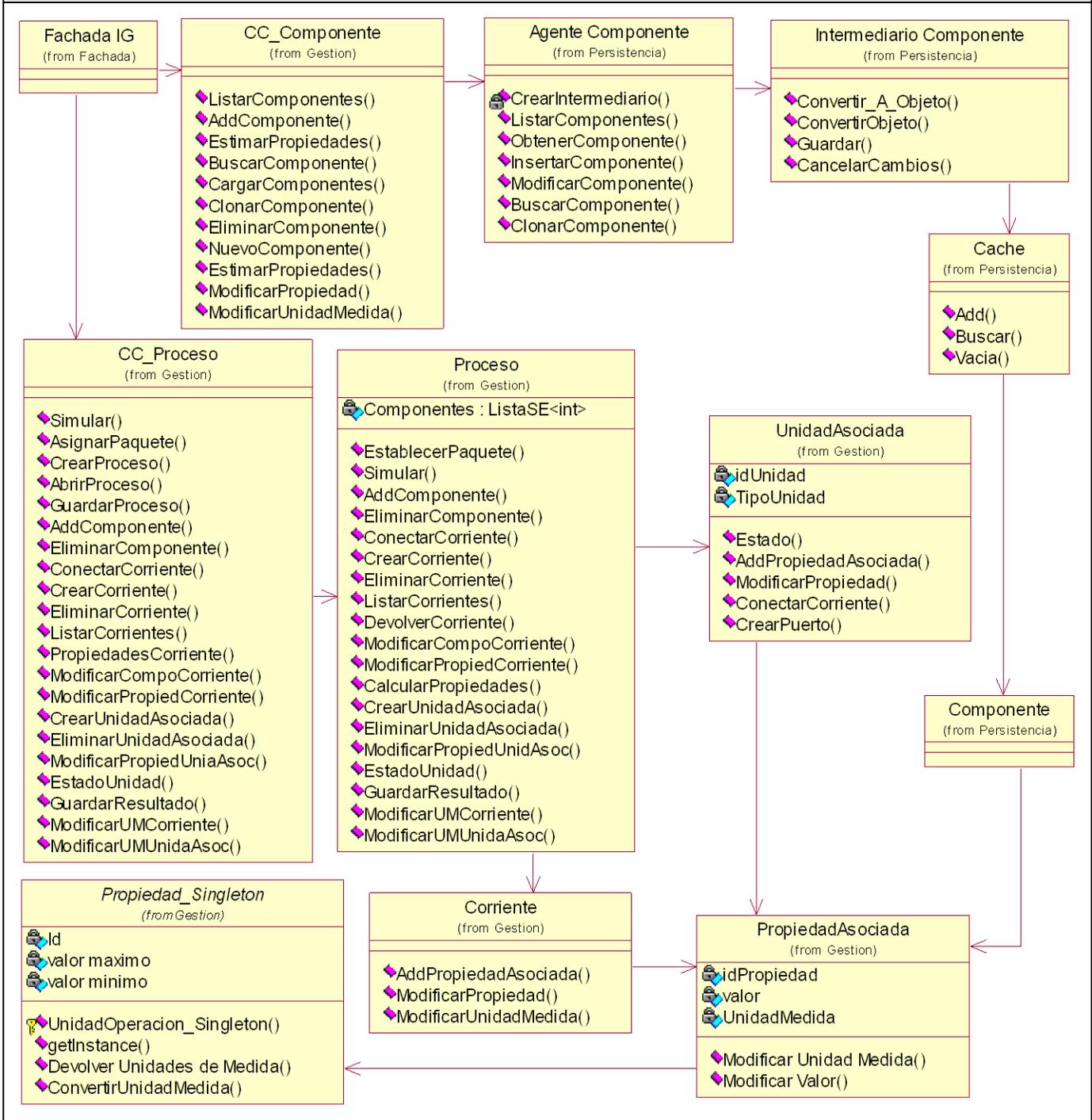


Figura 32 Diagrama de Clases del Diseño Caso de Uso: Manipular Unidades Medida

### Anexo 4. Diagramas de Interacción (Secuencia)

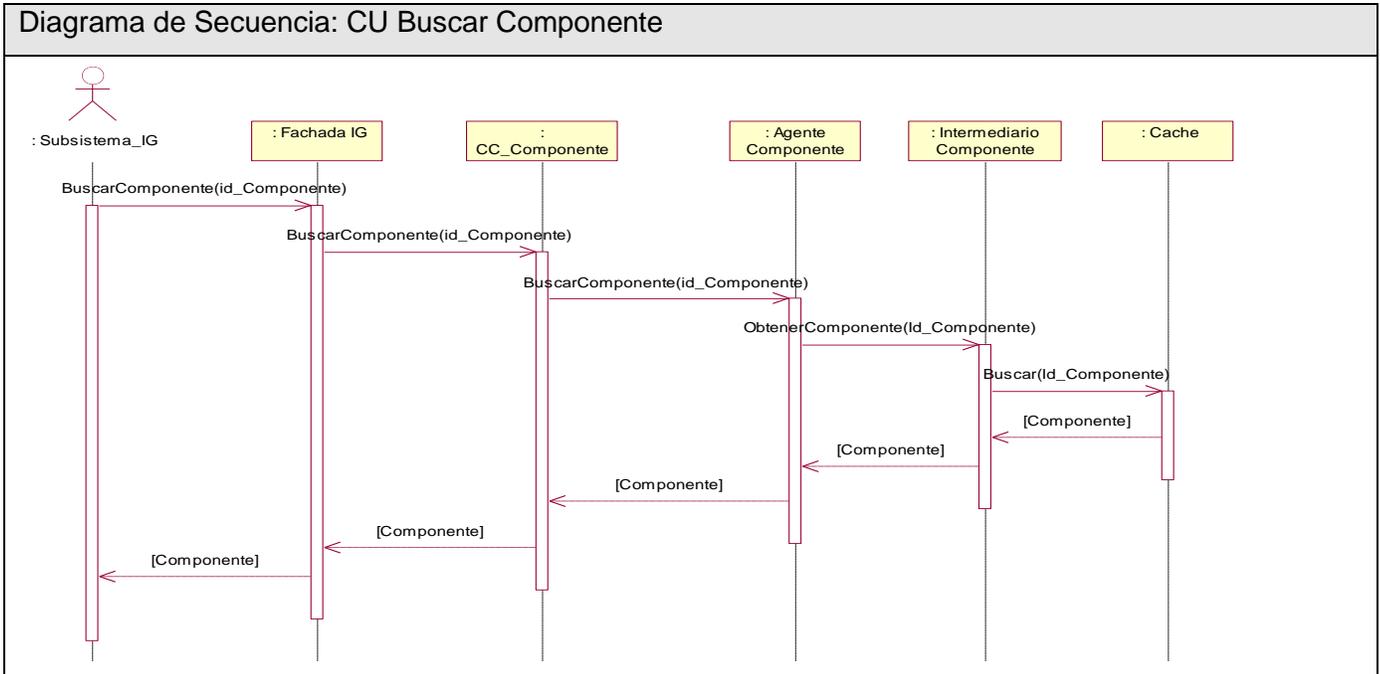


Figura 33 Diagrama de Secuencia Caso de Uso: Buscar Componente

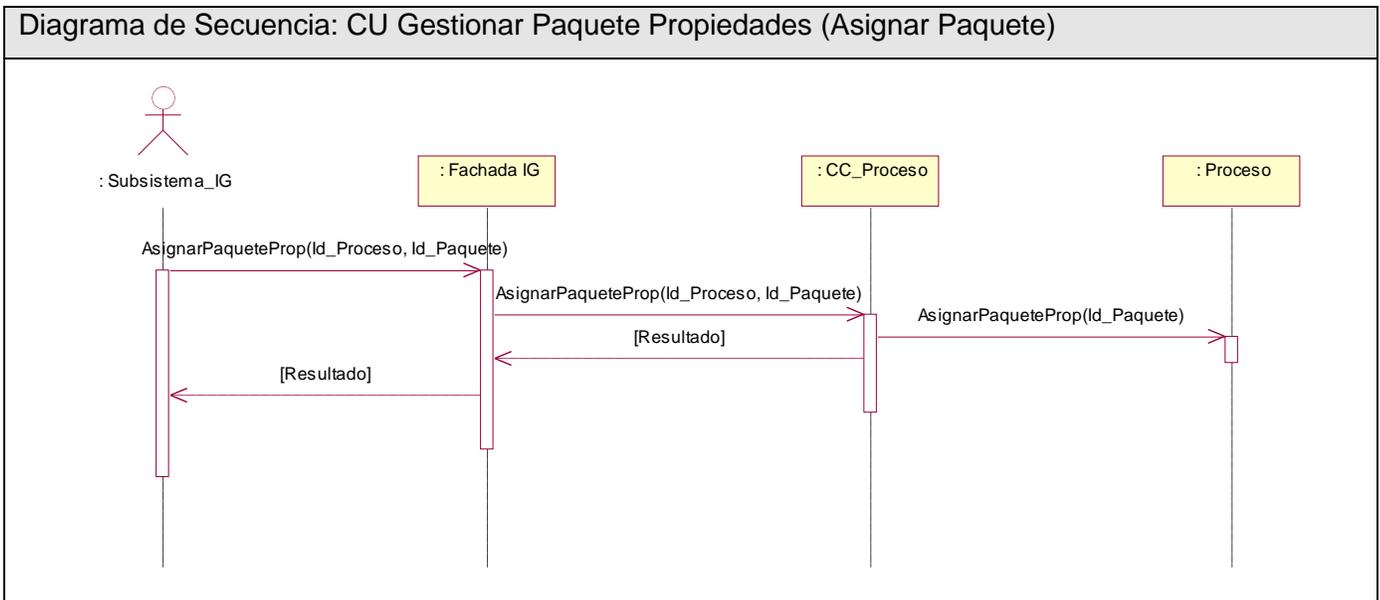


Figura 34 Diagrama de Secuencia Caso de Uso: Gestionar Paquete Propiedades (Escenario Asignar Paquete)

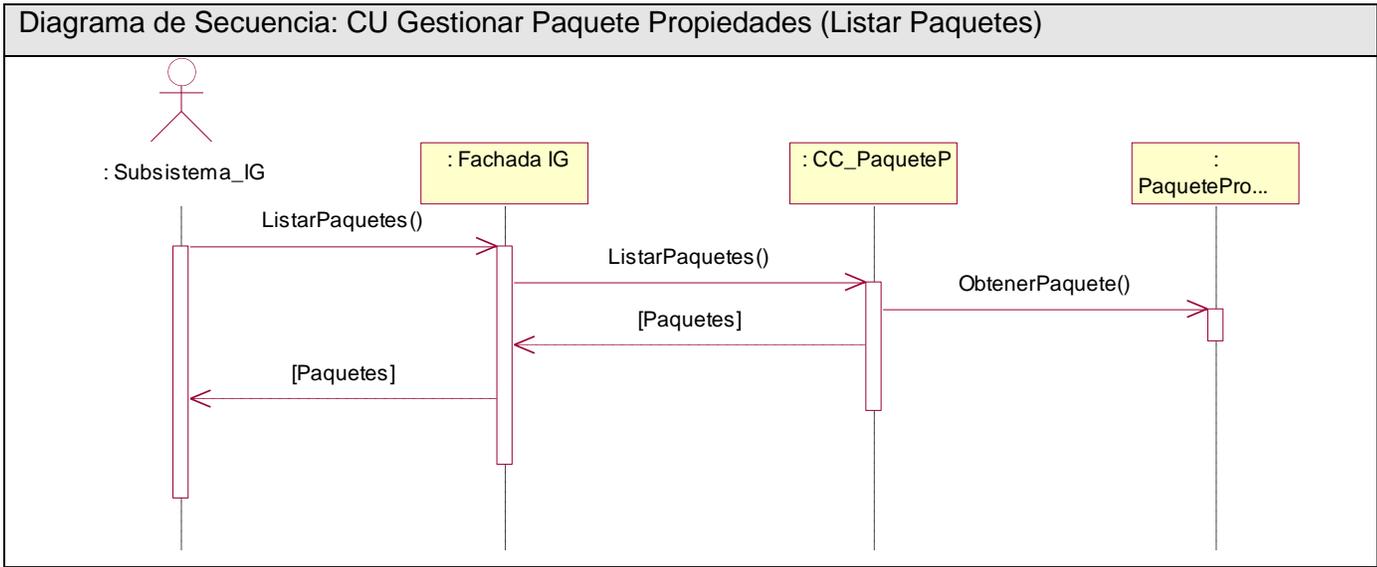


Figura 35 Diagrama de Secuencia Caso de Uso: Gestionar Paquete Propiedades (Escenario Listar Paquetes)

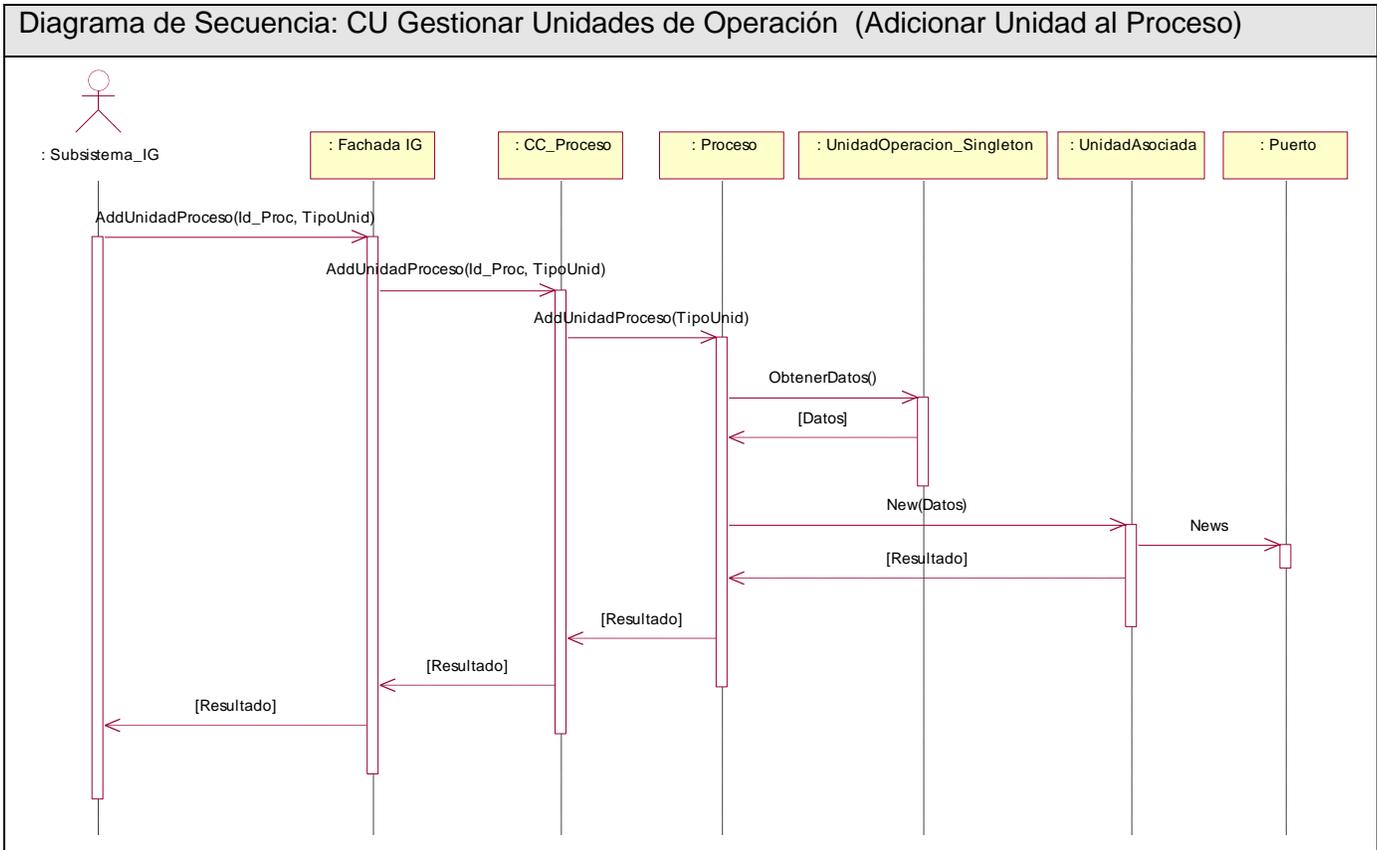


Figura 36 Diagrama de Secuencia Caso de Uso: Gestionar Unidad de Operación (Escenario Adicionar Unidad al Proceso)

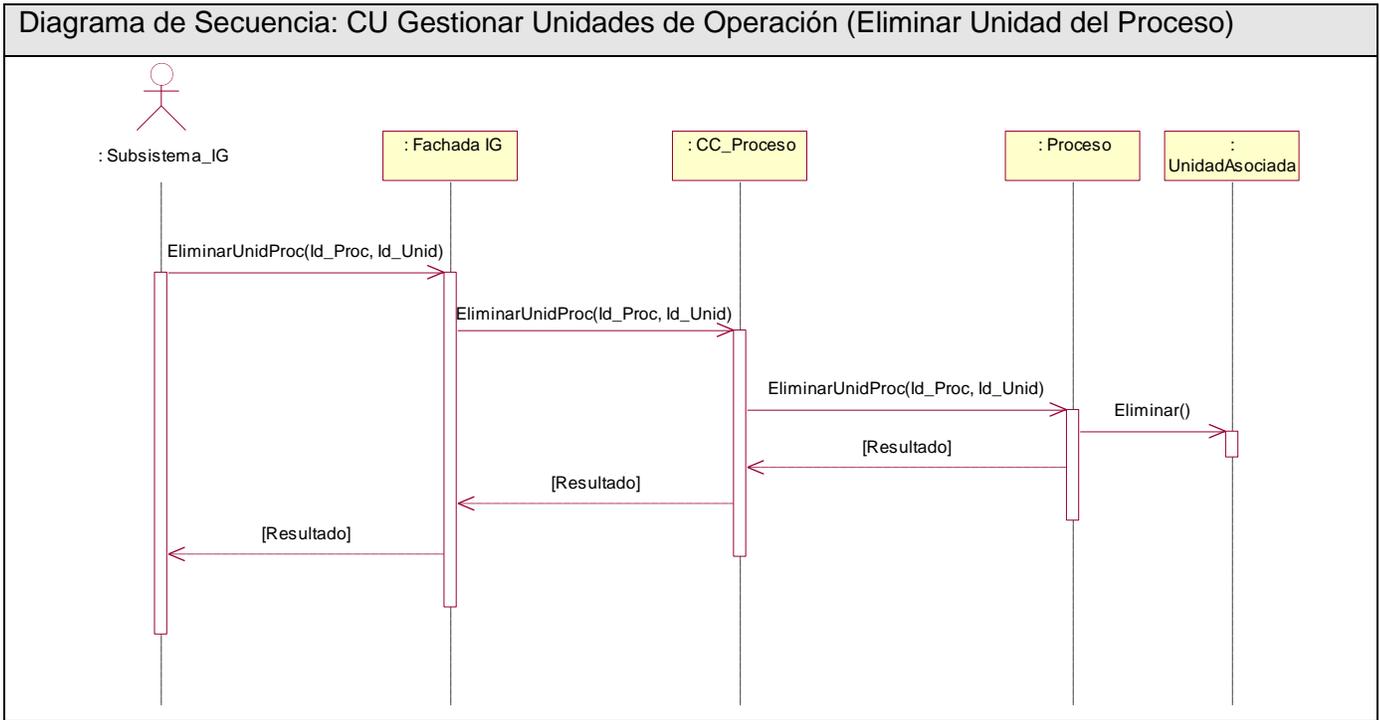


Figura 37 Diagrama de Secuencia Caso de Uso: Gestionar Unidades de Operación (Escenario Eliminar Unidad del Proceso)

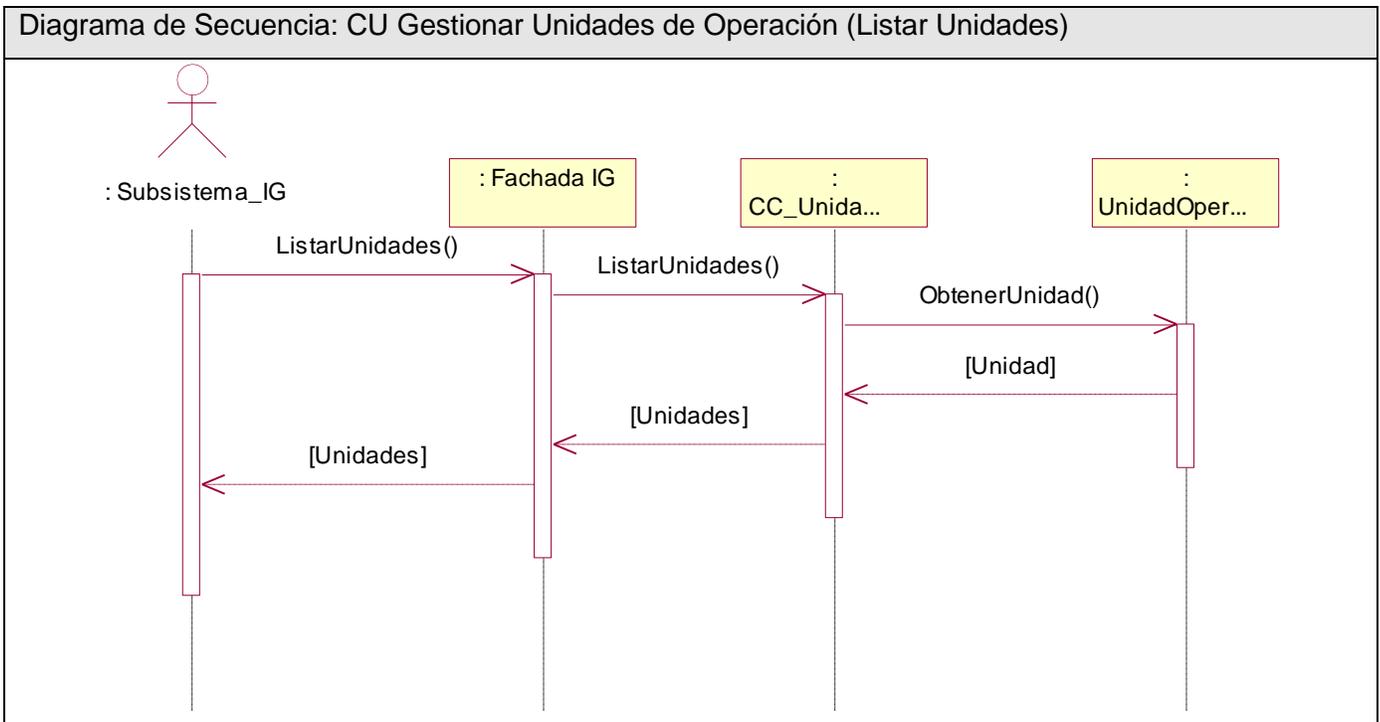


Figura 38 Diagrama de Secuencia Caso de Uso: Gestionar Unidades de Operación (Escenario Listar Unidades)

Diagrama de Secuencia: CU Gestionar Unidades de Operación (Modificar Propiedades)

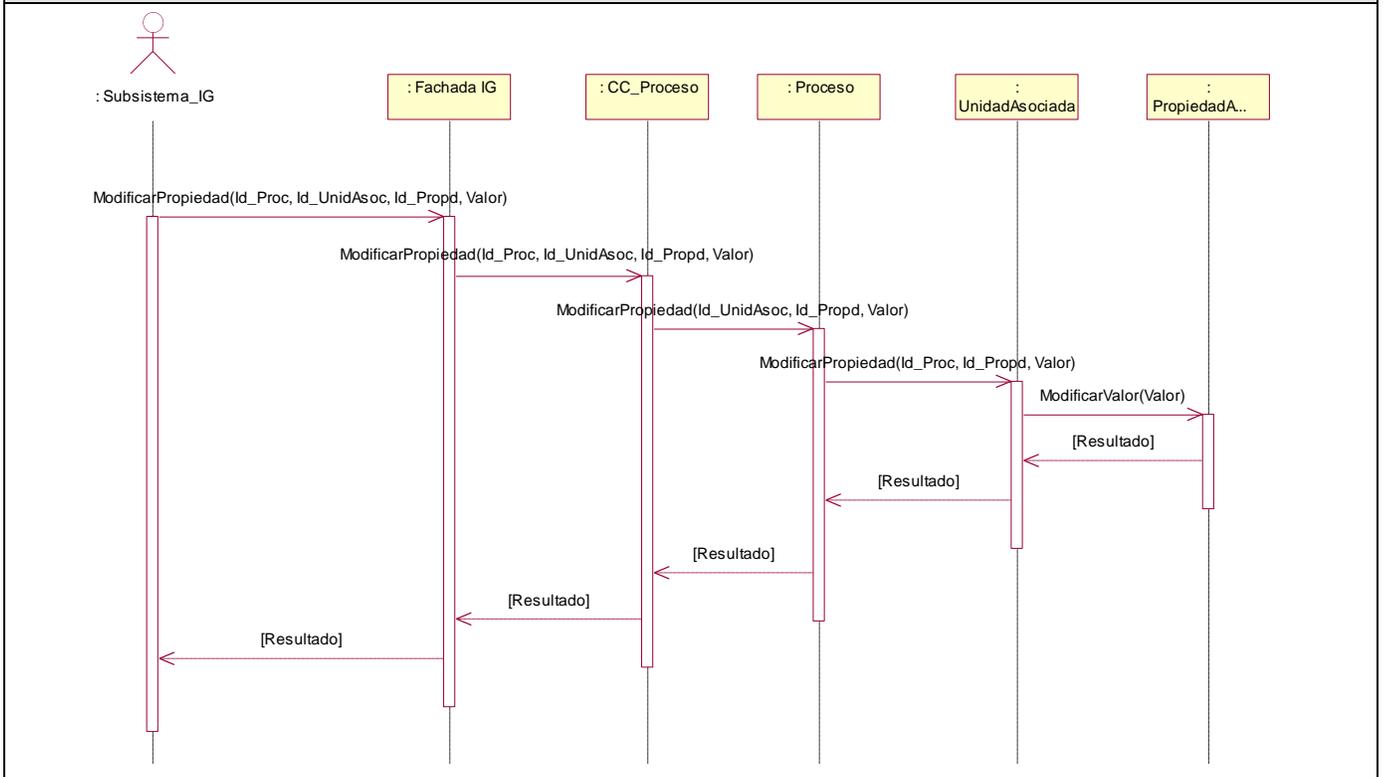


Figura 39 Diagrama de Secuencia Caso de Uso: Gestionar Unidades de Operación (Escenario Modificar Propiedades)

Diagrama de Secuencia: CU Gestionar Unidades de Operación (Obtener Estado)

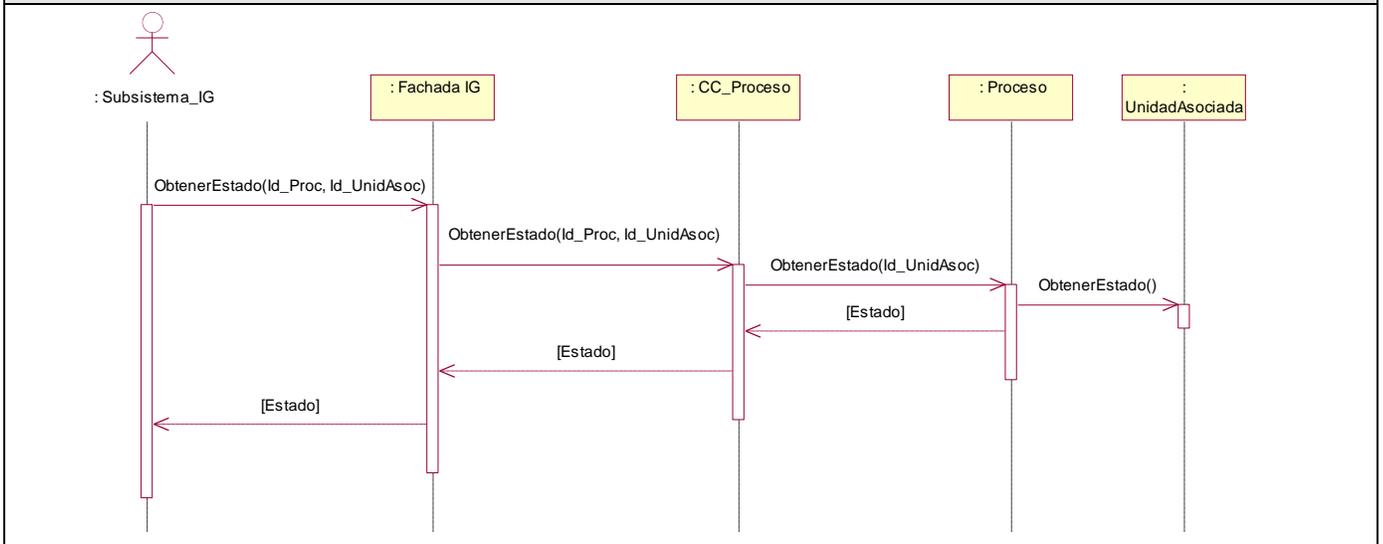


Figura 40 Diagrama de Secuencia Caso de Uso: Gestionar Unidades de Operación (Escenario Obtener Estado)

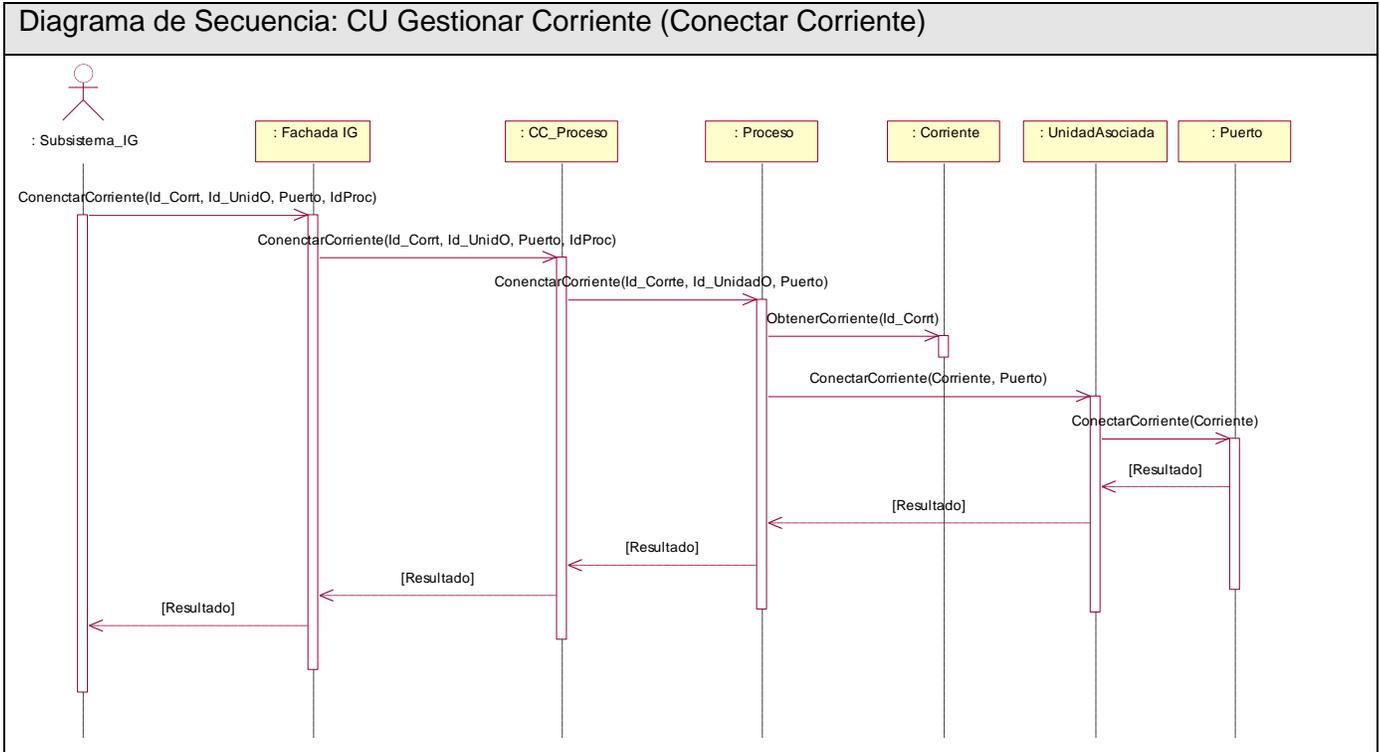


Figura 41 Diagrama de Secuencia Caso de Uso: Gestionar Corriente (Escenario Conectar Corriente)

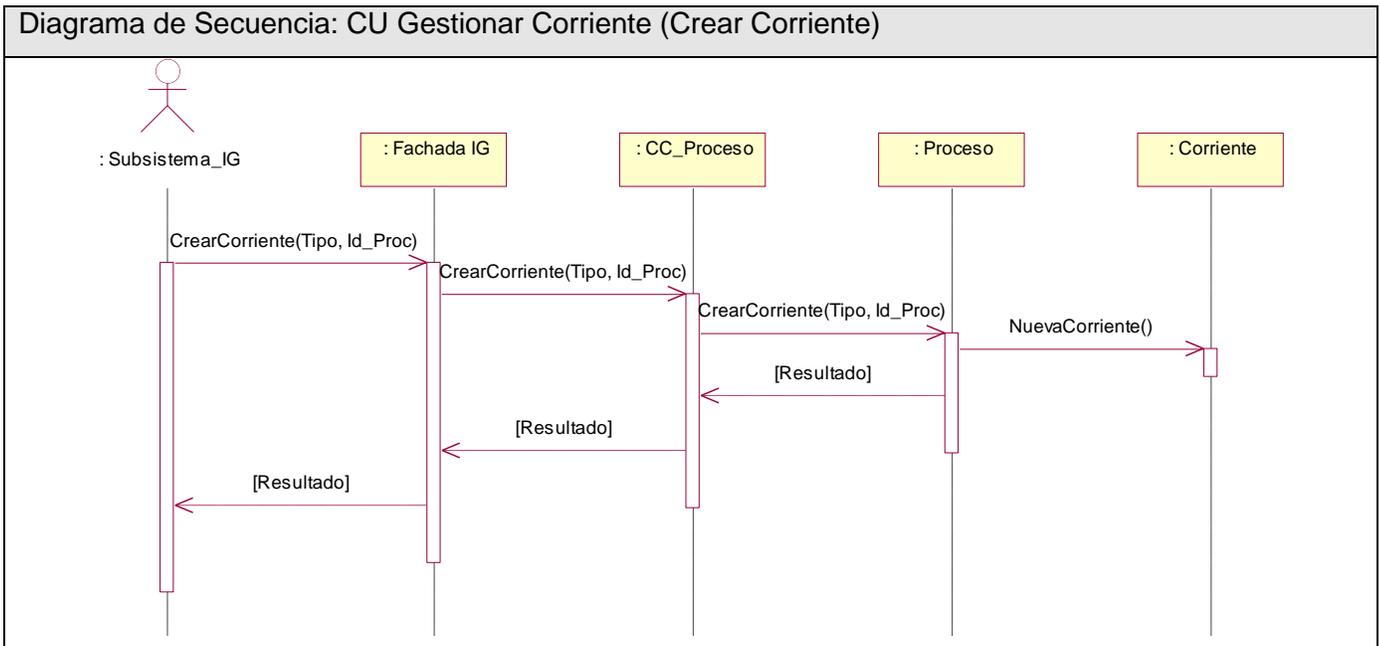


Figura 42 Diagrama de Secuencia Caso de Uso: Gestionar Corriente (Escenario Crear Corriente)

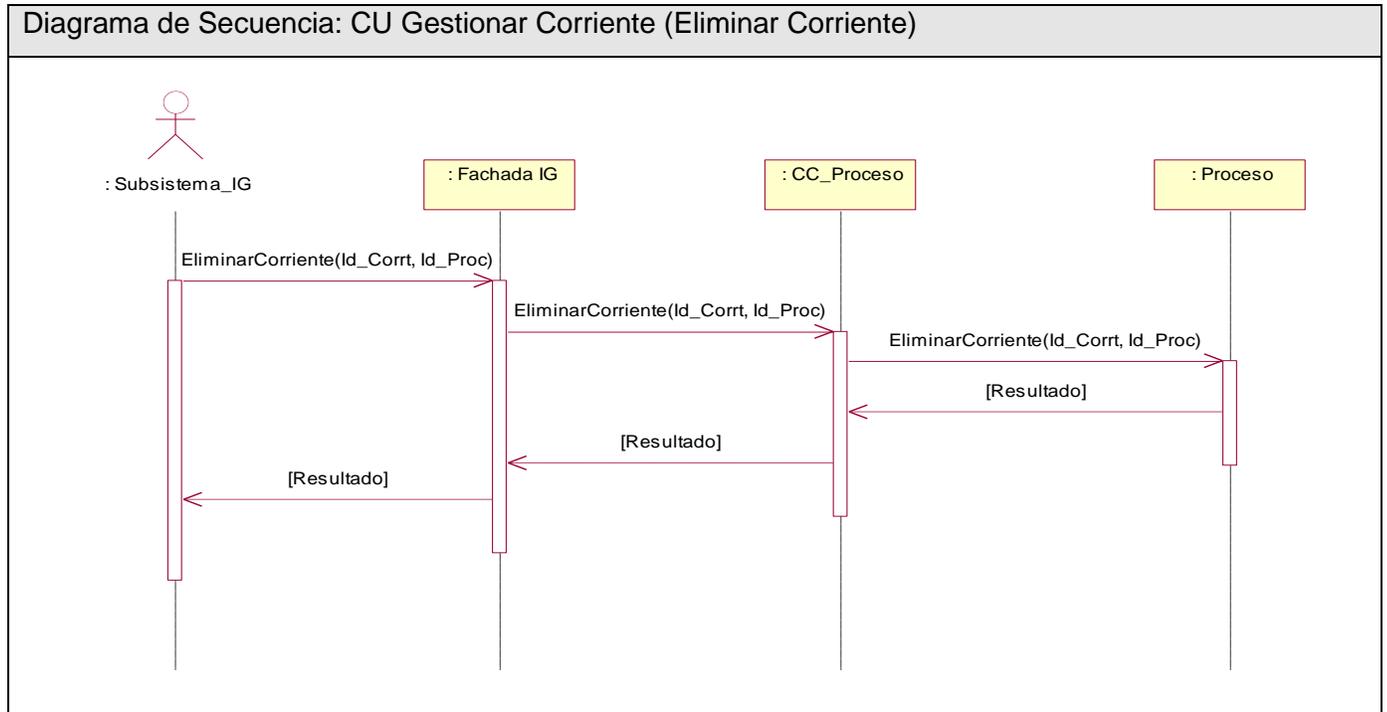


Figura 43 Diagrama de Secuencia Caso de Uso: Gestionar Corriente (Escenario Eliminar Corriente)

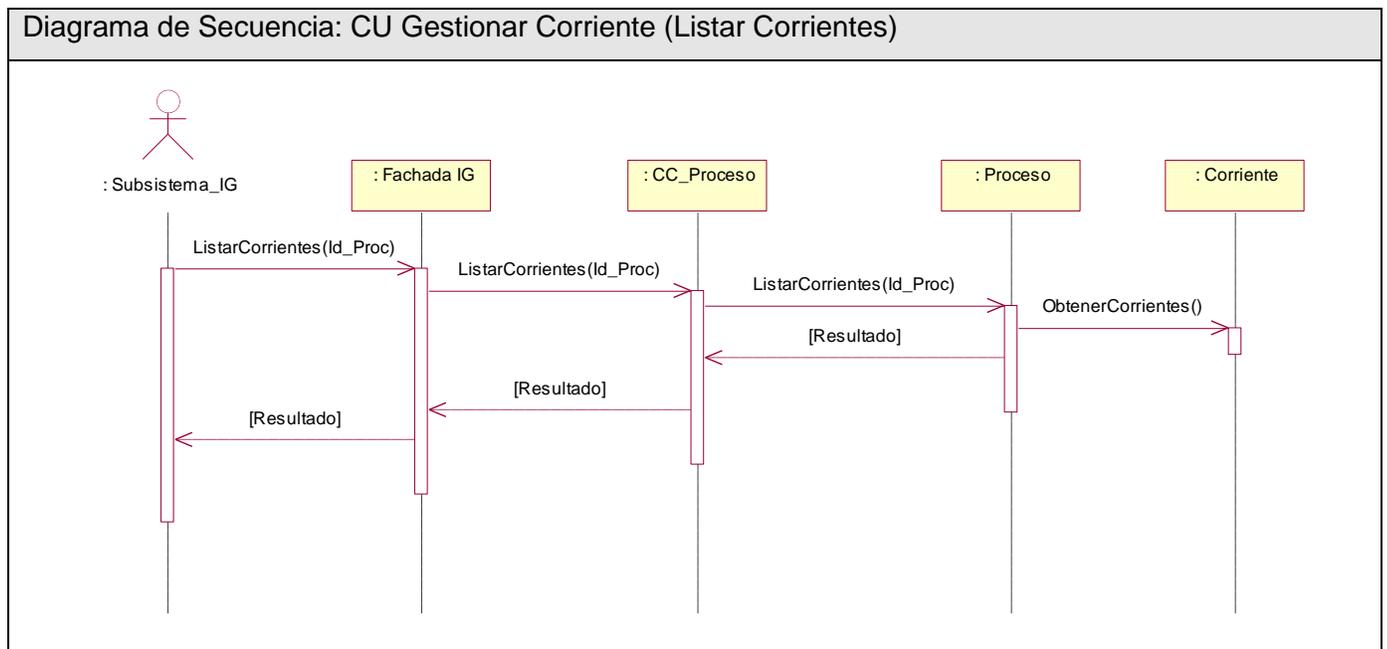


Figura 44 Diagrama de Secuencia Caso de Uso: Gestionar Corriente (Escenario Listar Corrientes)

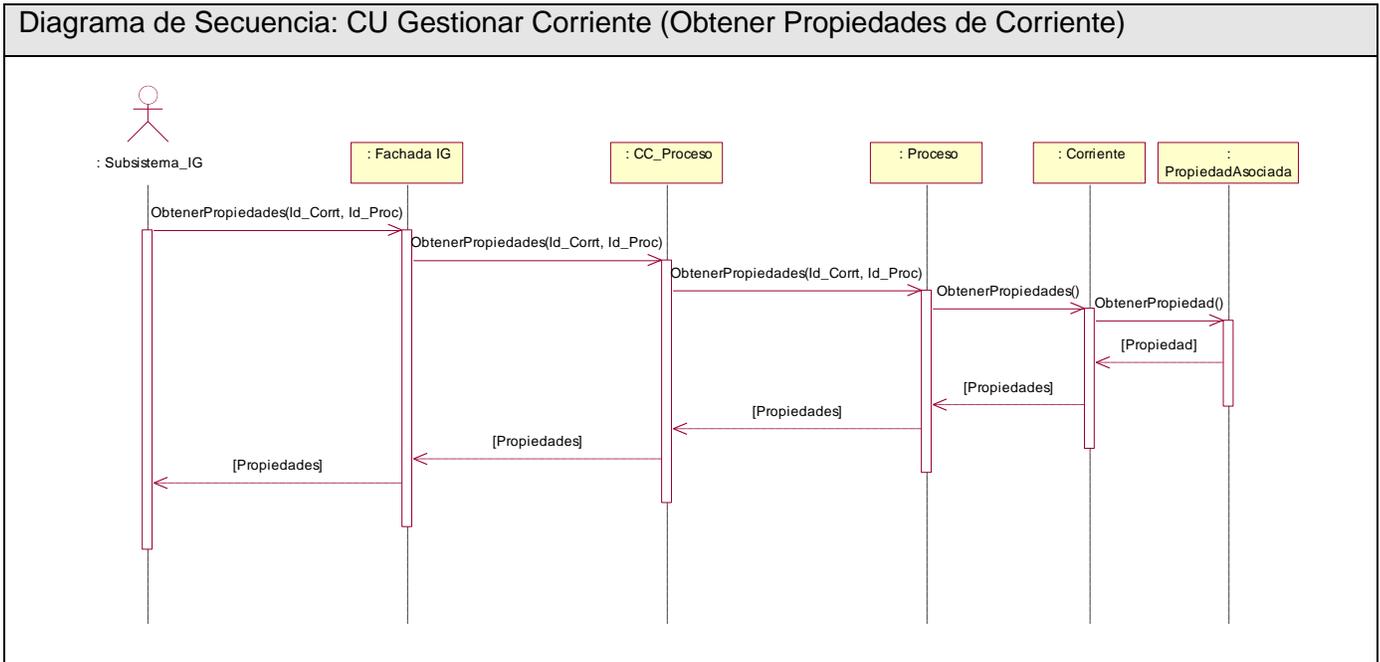


Figura 45 Diagrama de Secuencia Caso de Uso: Gestionar Corriente (Escenario Obtener Propiedades de Corriente)

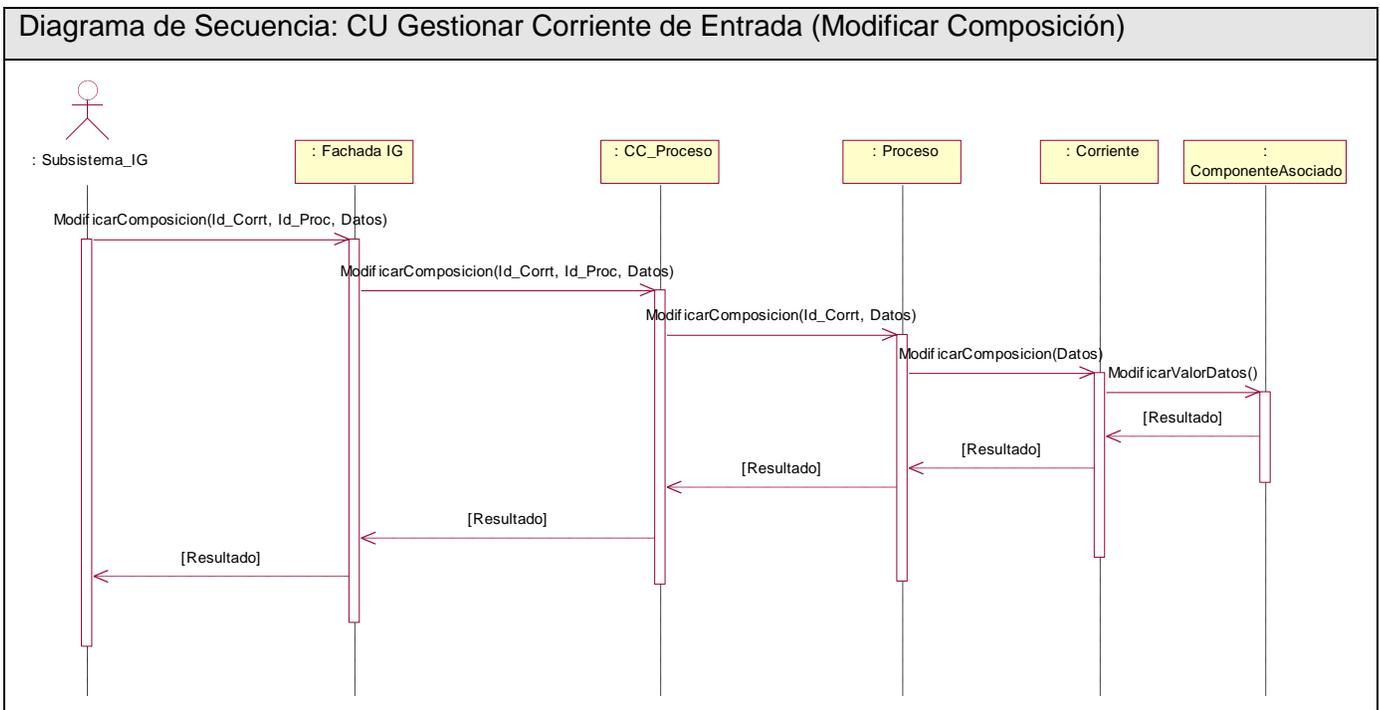


Figura 46 Diagrama de Secuencia Caso de Uso: Gestionar Corriente de Entrada (Escenario Modificar Composición)

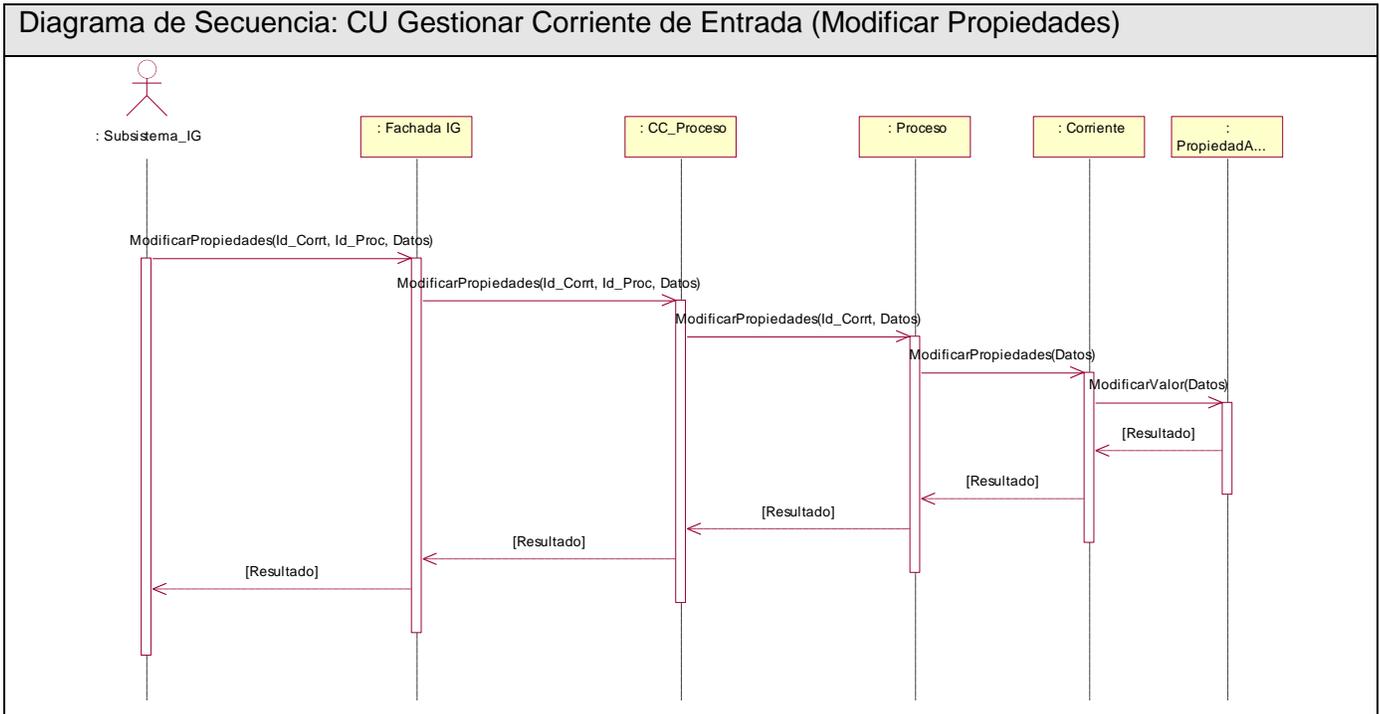


Figura 47 Diagrama de Secuencia Caso de Uso: Gestionar Corriente de Entrada (Escenario Modificar Propiedades)

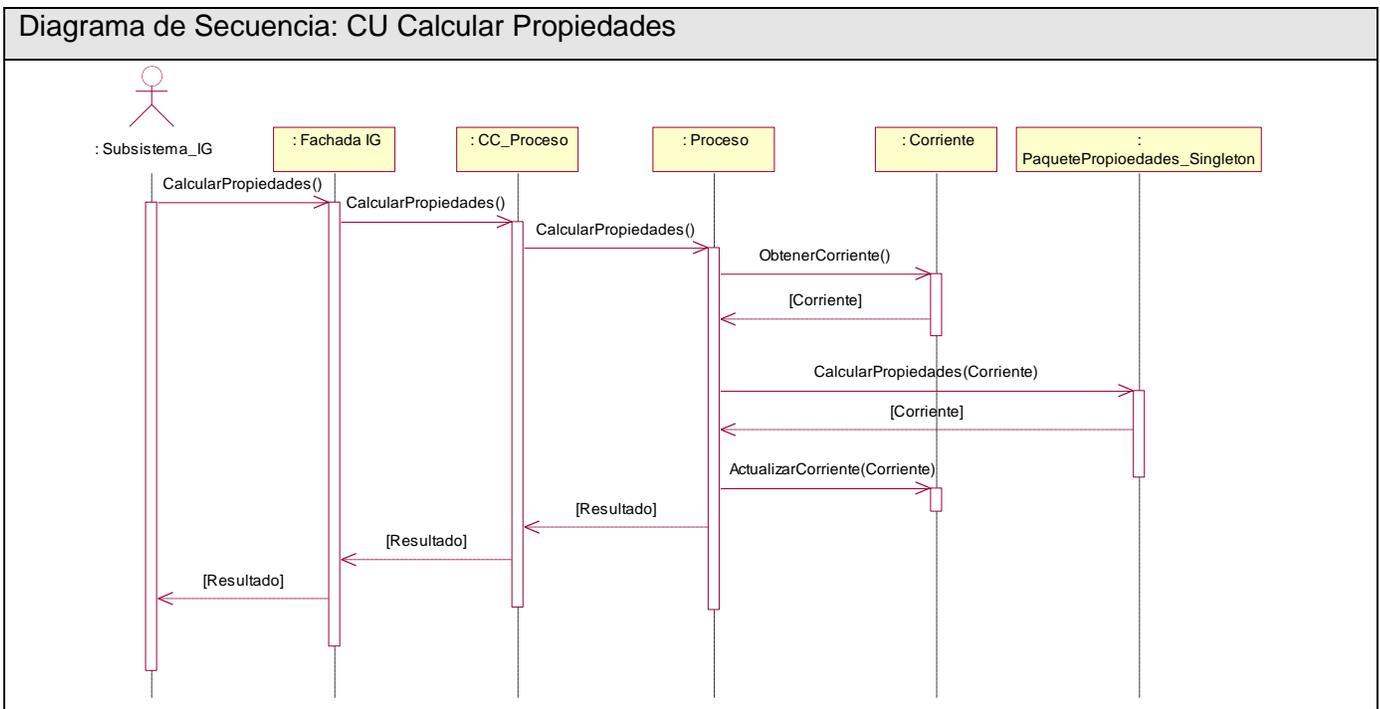


Figura 48 Diagrama de Secuencia Caso de Uso: Calcular Propiedades

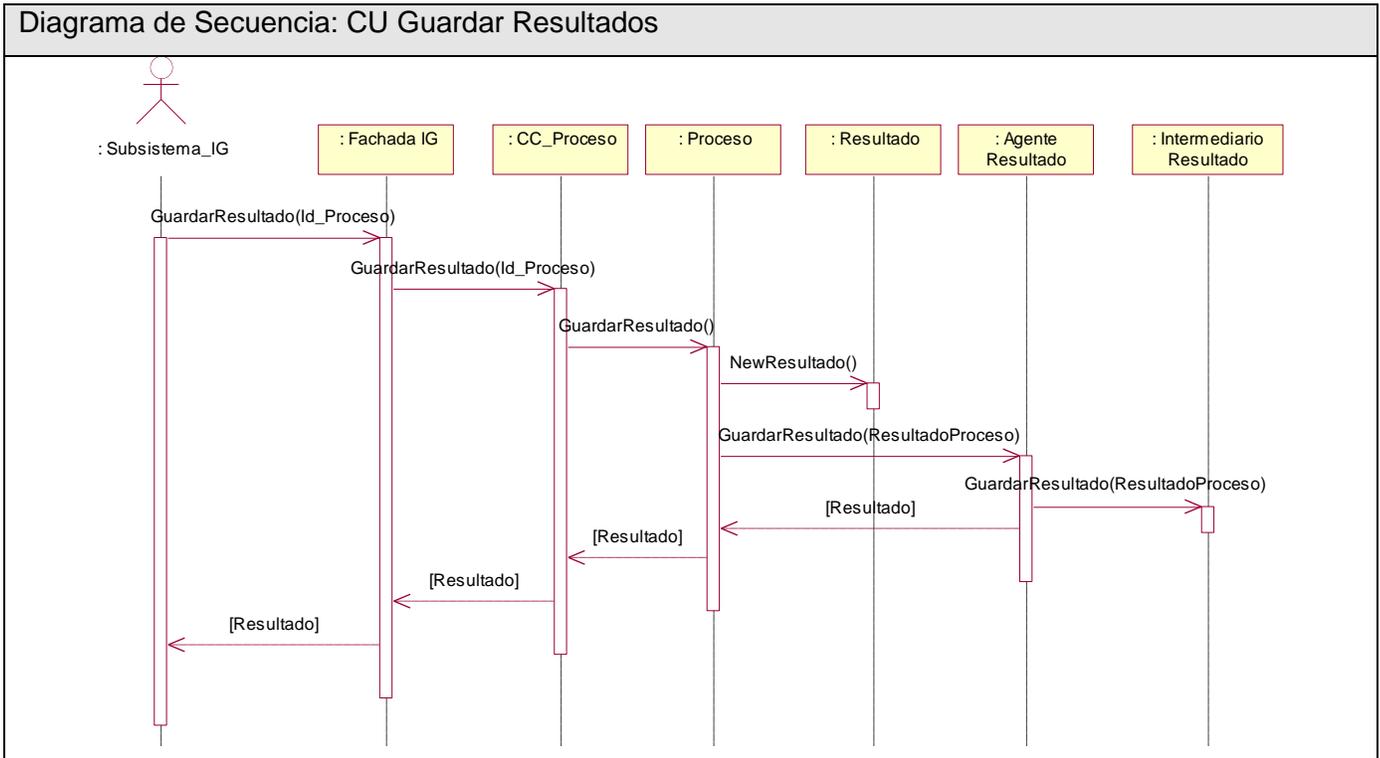


Figura 49 Diagrama de Secuencia Caso de Uso: Guardar Resultados

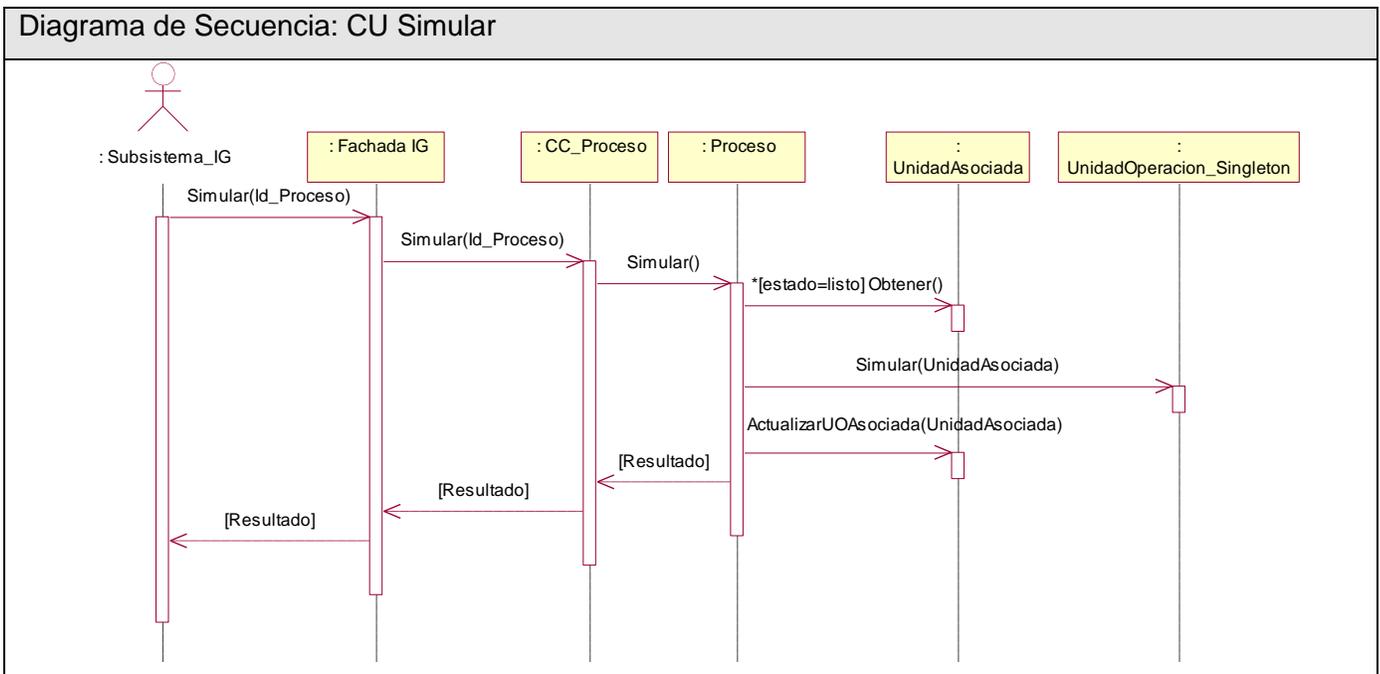


Figura 50 Diagrama de Secuencia Caso de Uso: Simular

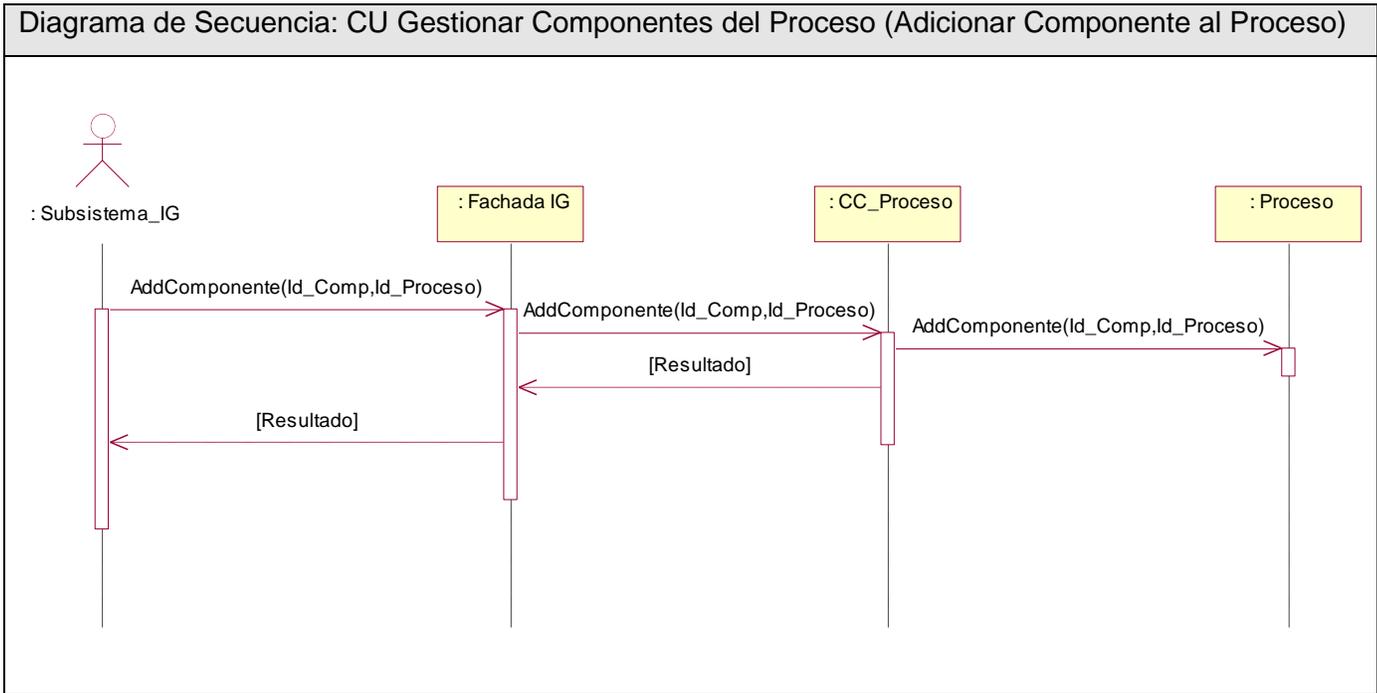


Figura 51 Diagrama de Secuencia Caso de Uso: Gestionar Componentes del Proceso (Escenario Adicionar Componente al Proceso)

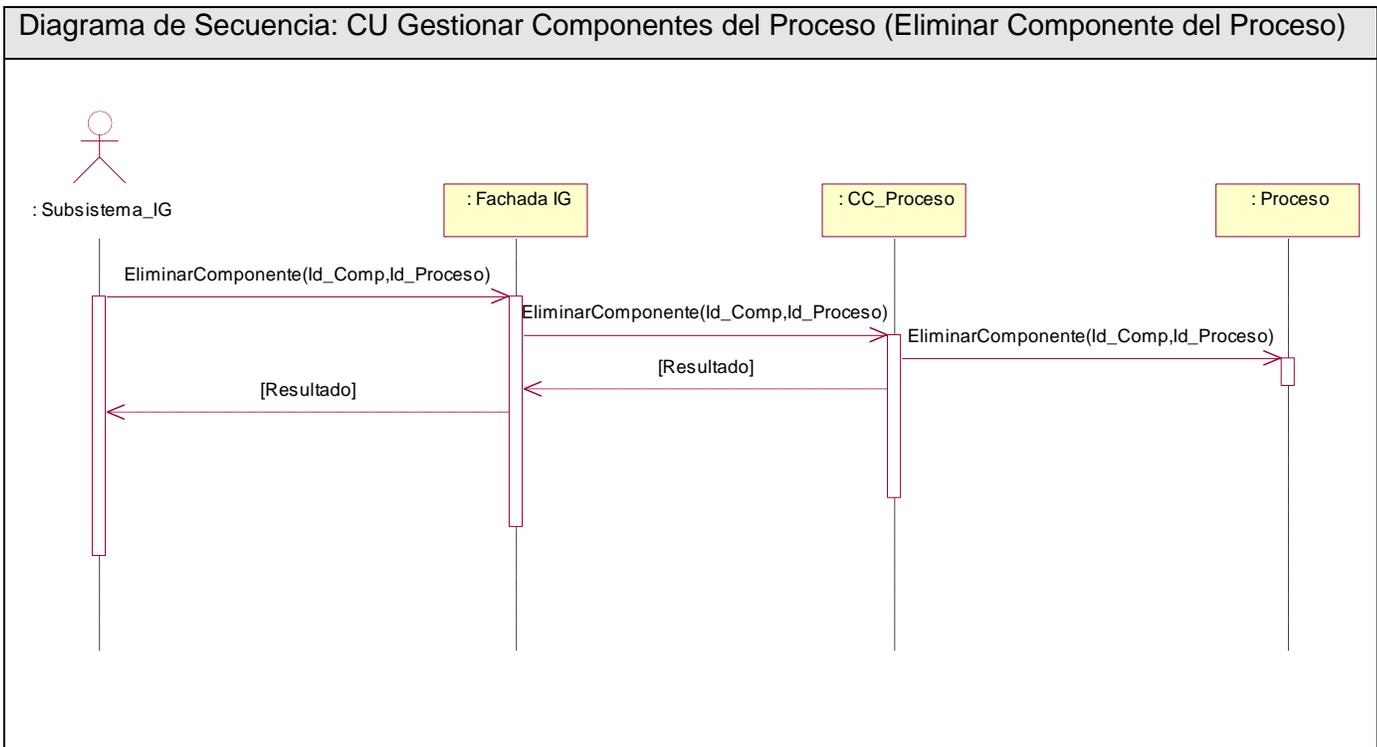


Figura 52 Diagrama de Secuencia Caso de Uso: Gestionar Componentes del Proceso (Escenario Eliminar Componente del Proceso)

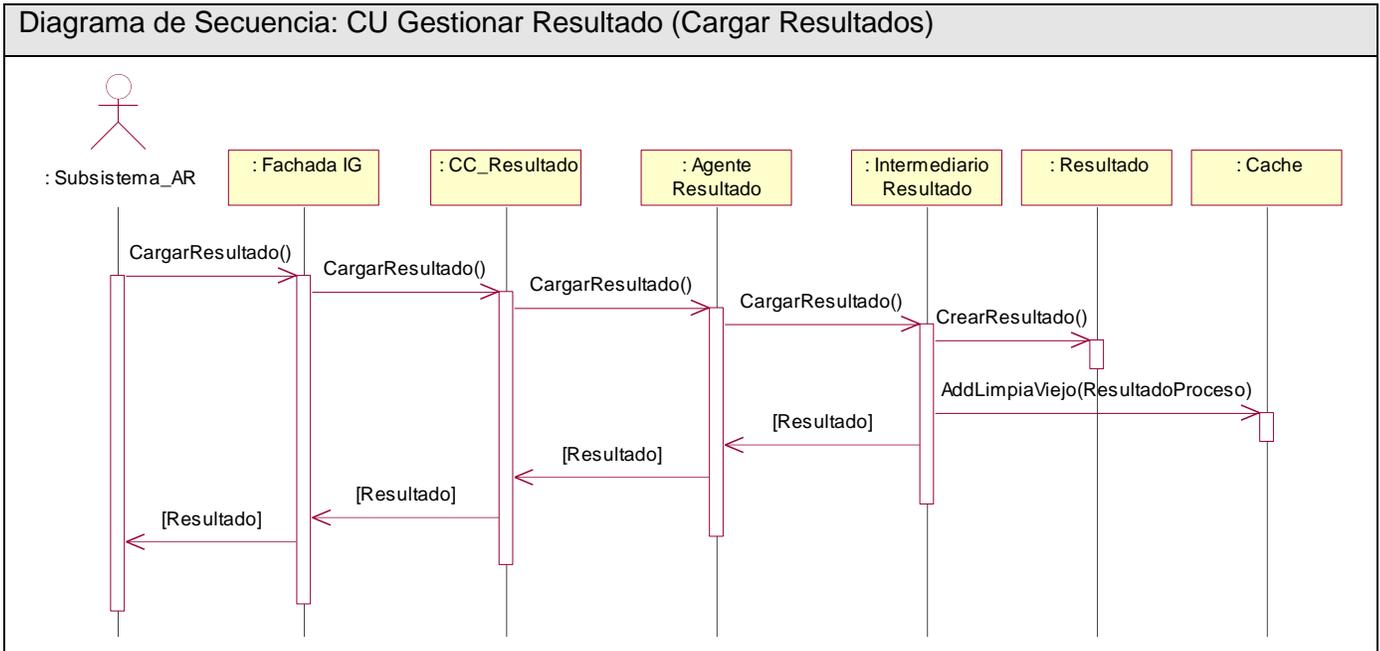


Figura 53 Diagrama de Secuencia Caso de Uso: Gestionar Resultado (Escenario Cargar Resultados)

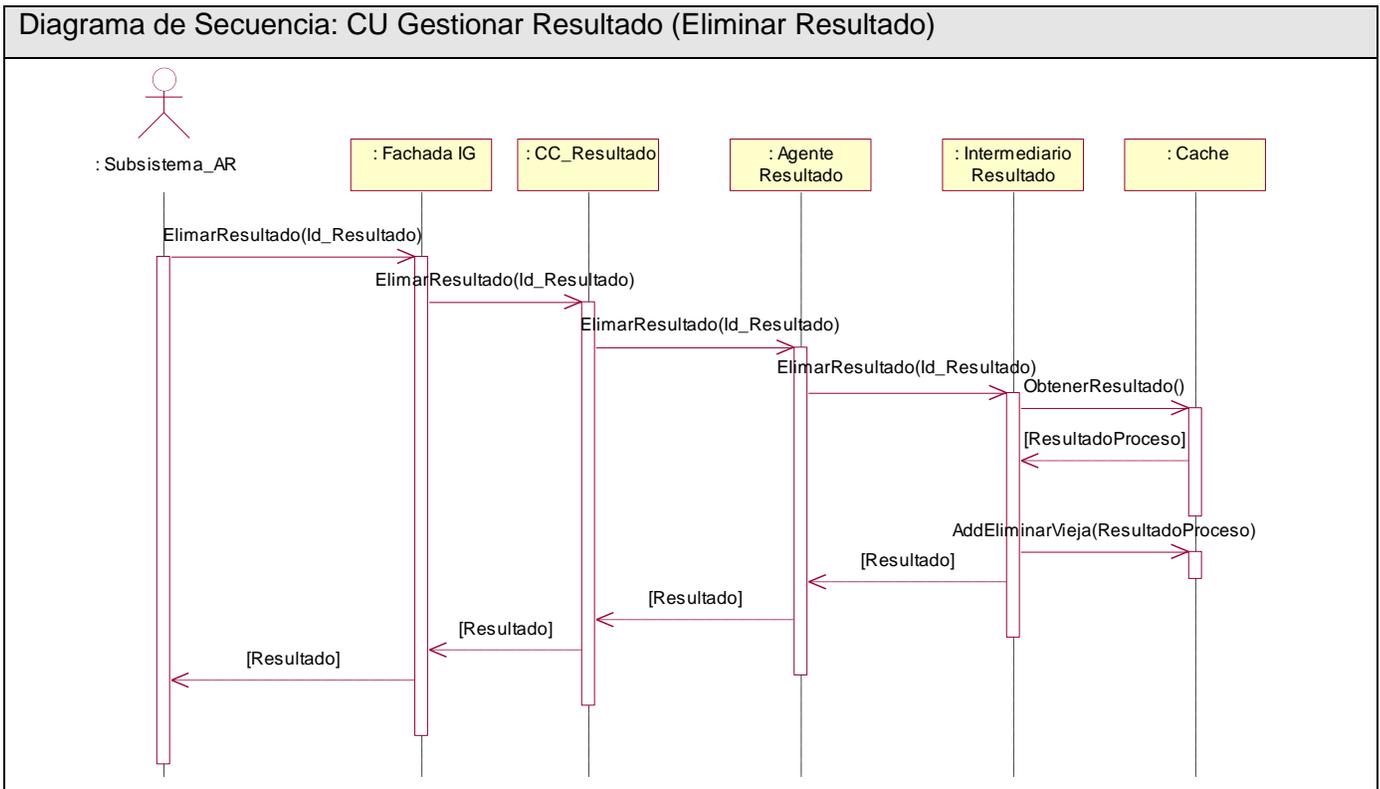


Figura 54 Diagrama de Secuencia Caso de Uso: Gestionar Resultado (Escenario Eliminar Resultado)

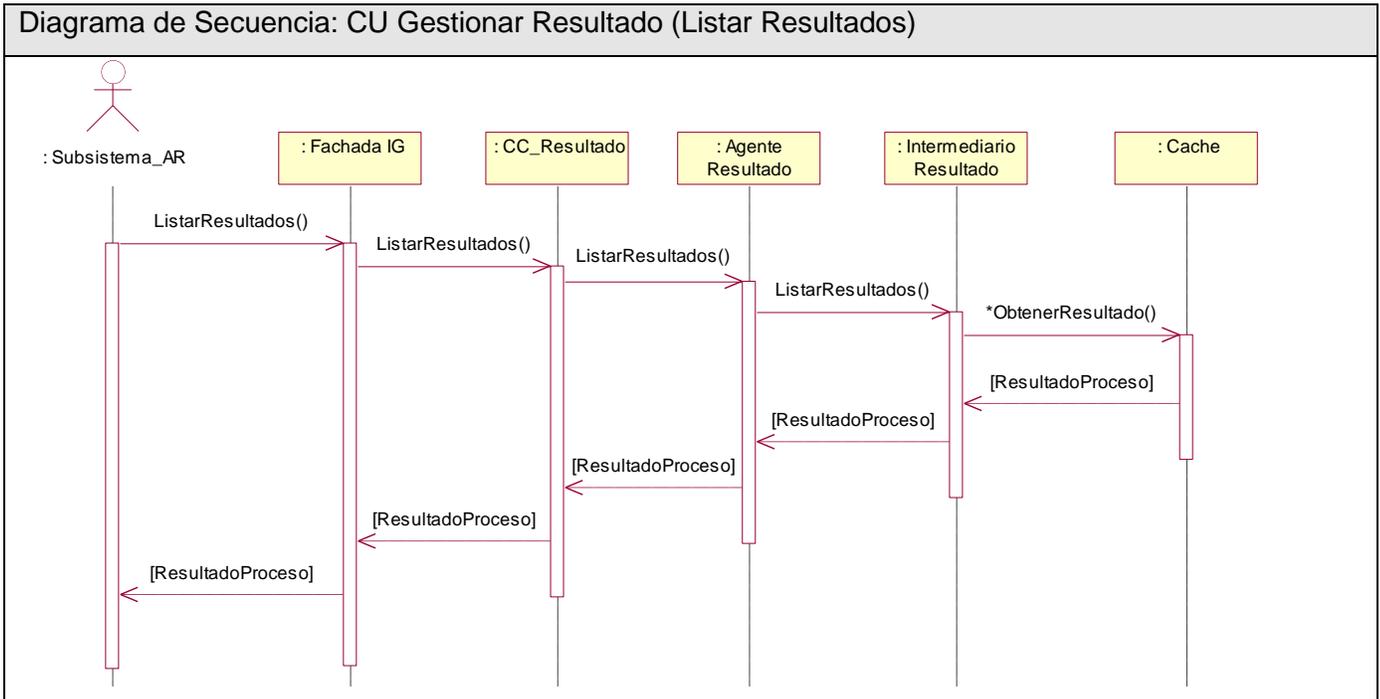


Figura 55 Diagrama de Secuencia Caso de Uso: Gestionar Resultado (Escenario Listar Resultados)

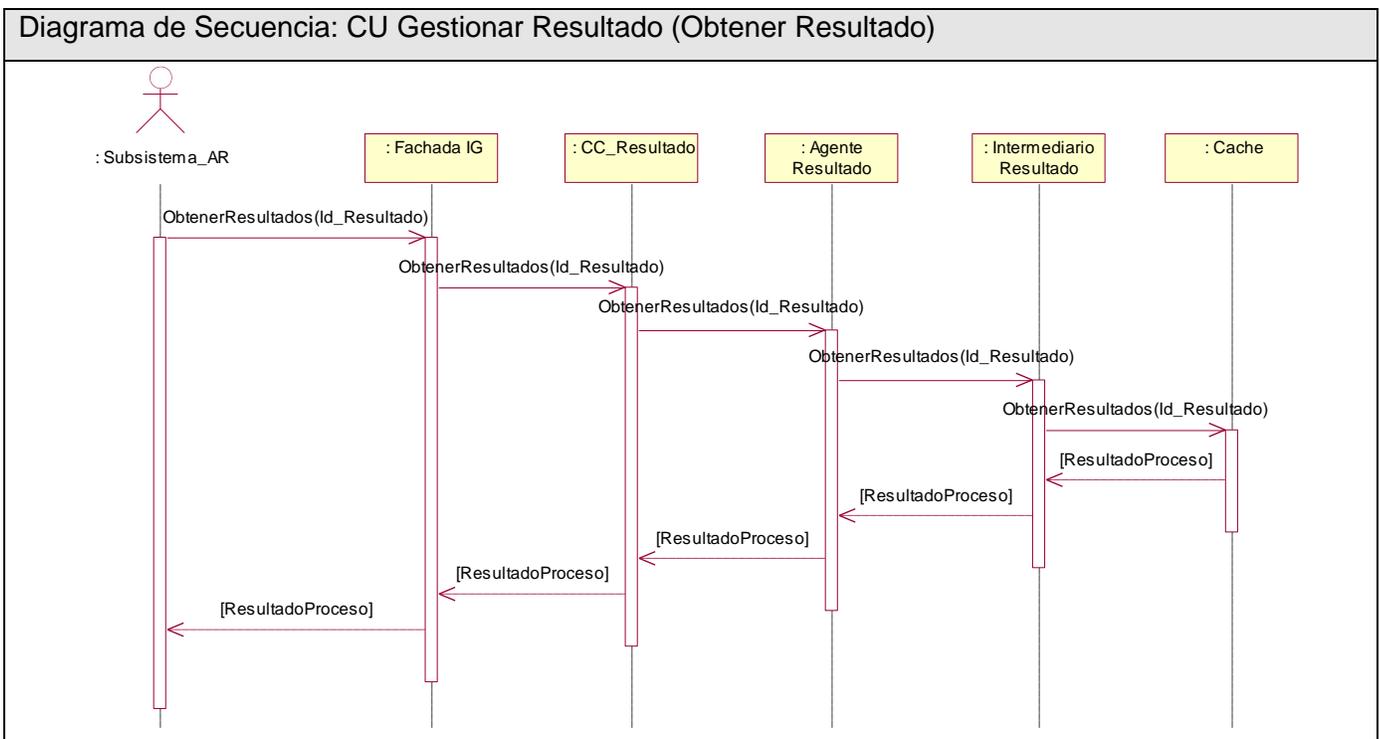


Figura 56 Diagrama de Secuencia Caso de Uso: Gestionar Resultado (Escenario Obtener Resultado)

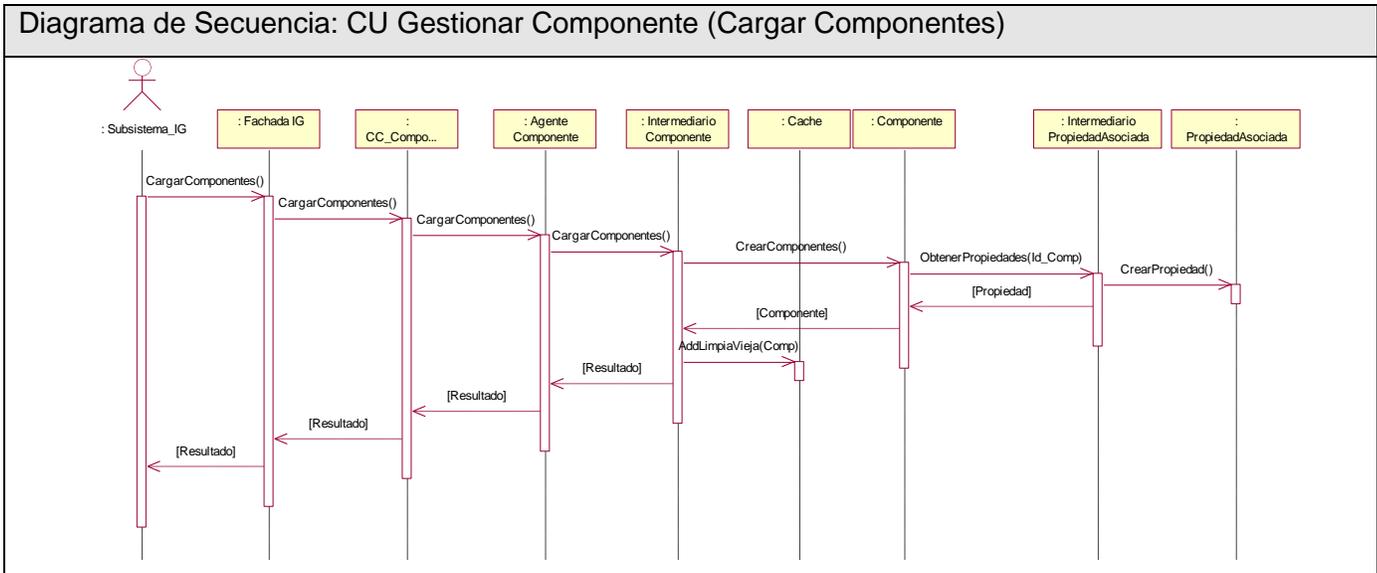


Figura 57 Diagrama de Secuencia Caso de Uso: Gestionar Componente (Escenario Cargar Componente)

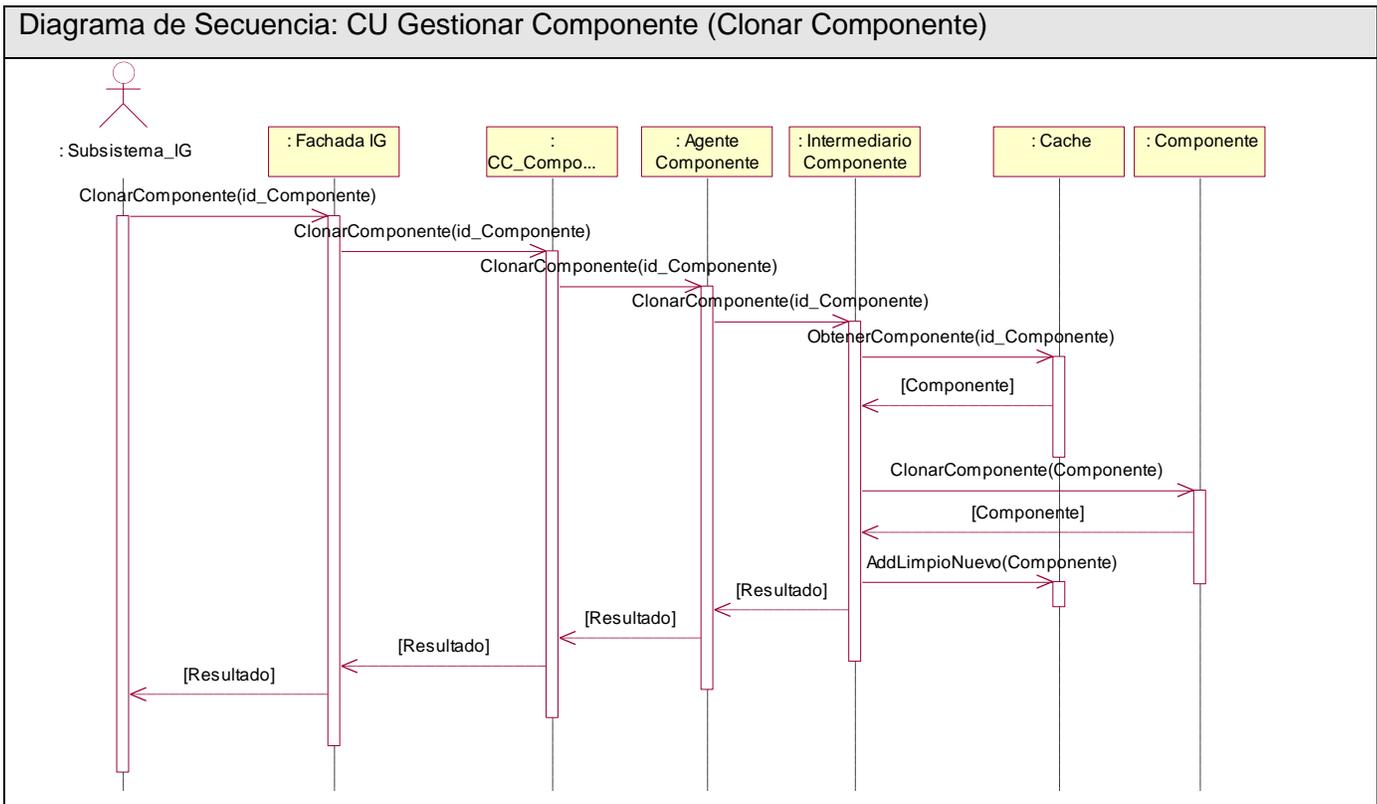


Figura 58 Diagrama de Secuencia Caso de Uso: Gestionar Componente (Escenario Clonar Componente)

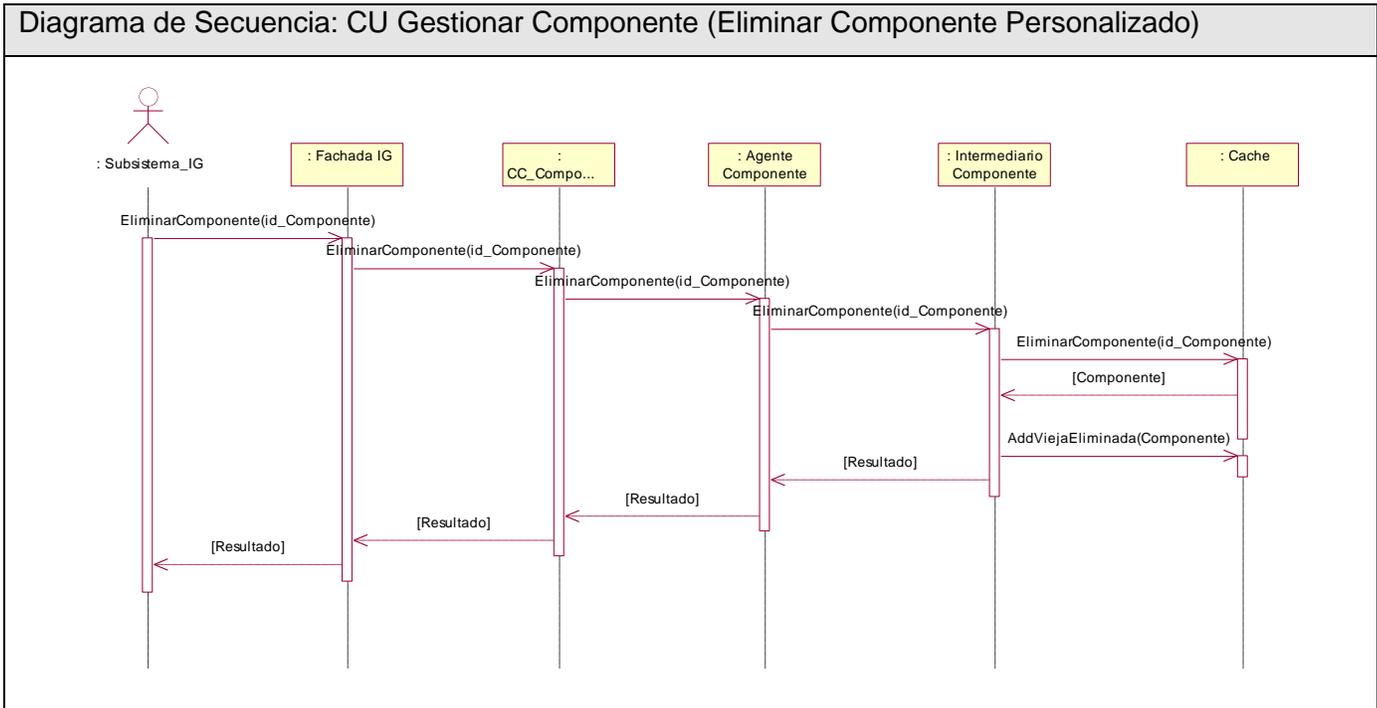


Figura 59 Diagrama de Secuencia Caso de Uso: Gestionar Componente (Escenario Eliminar Componente Personalizado)

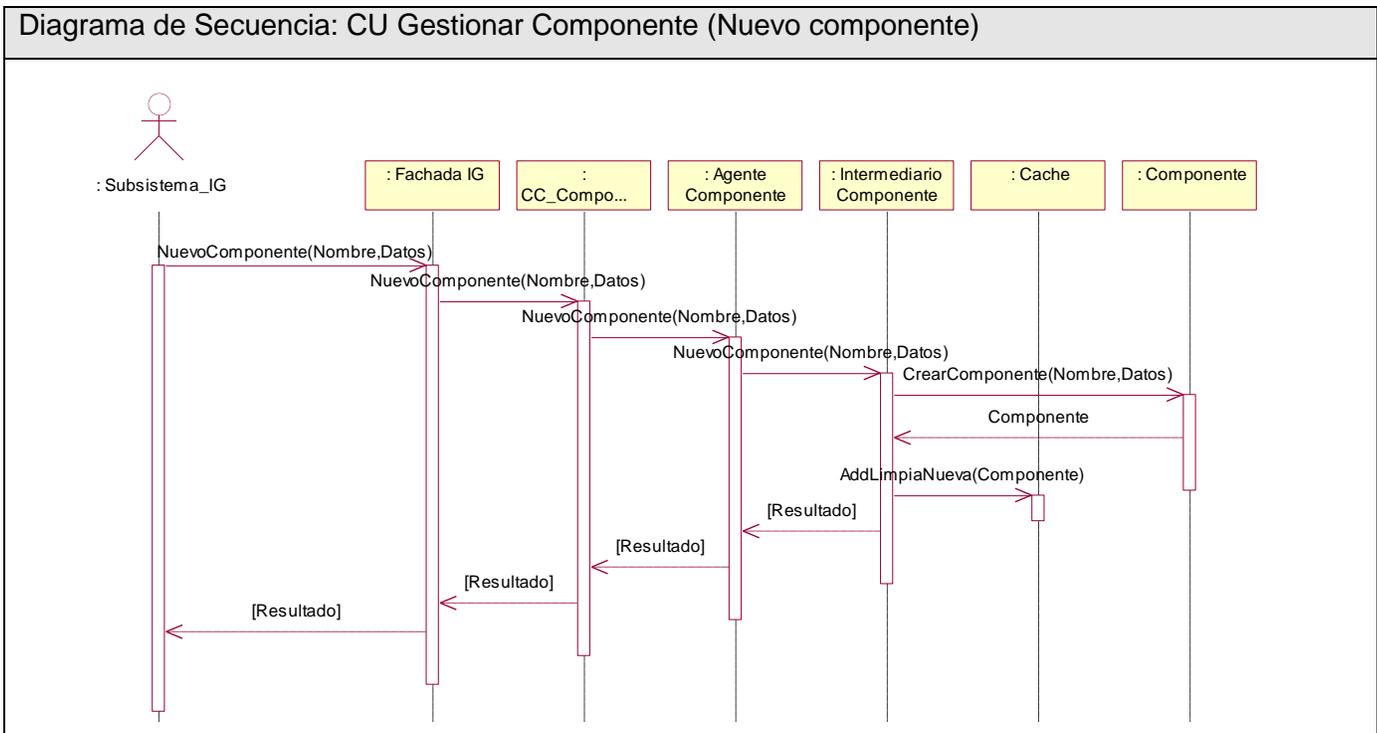


Figura 60 Diagrama de Secuencia Caso de Uso: Gestionar Componente (Escenario Nuevo Componente)

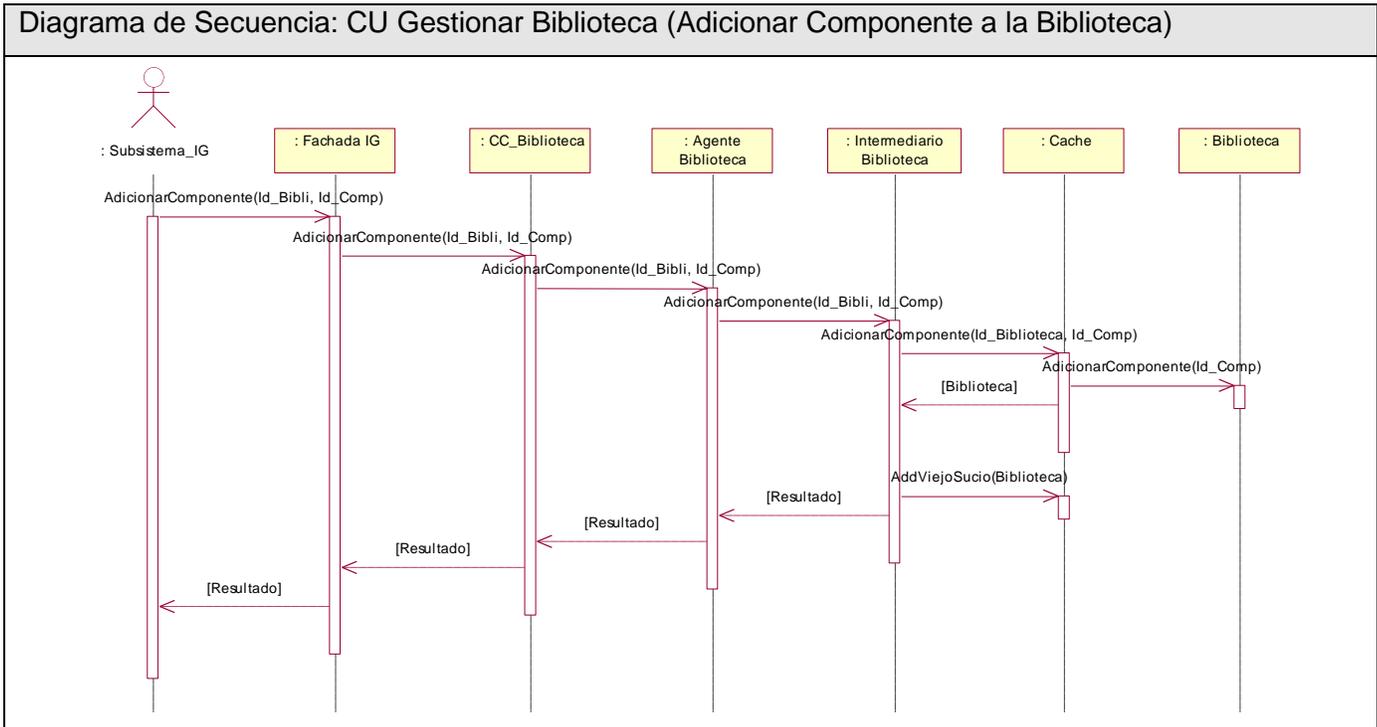


Figura 61 Diagrama de Secuencia Caso de Uso: Gestionar Biblioteca (Escenario Adicionar Componente a la Biblioteca)

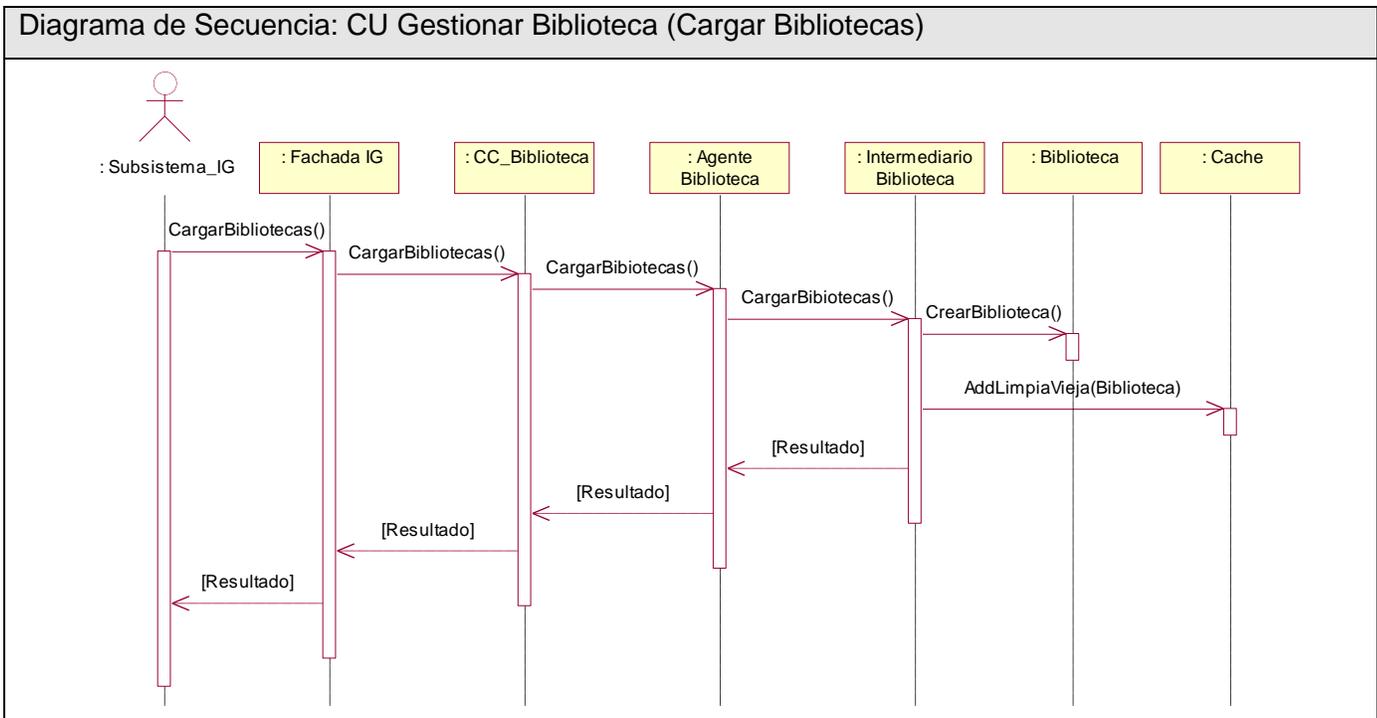


Figura 62 Diagrama de Secuencia Caso de Uso: Gestionar Biblioteca (Escenario Cargar Bibliotecas)

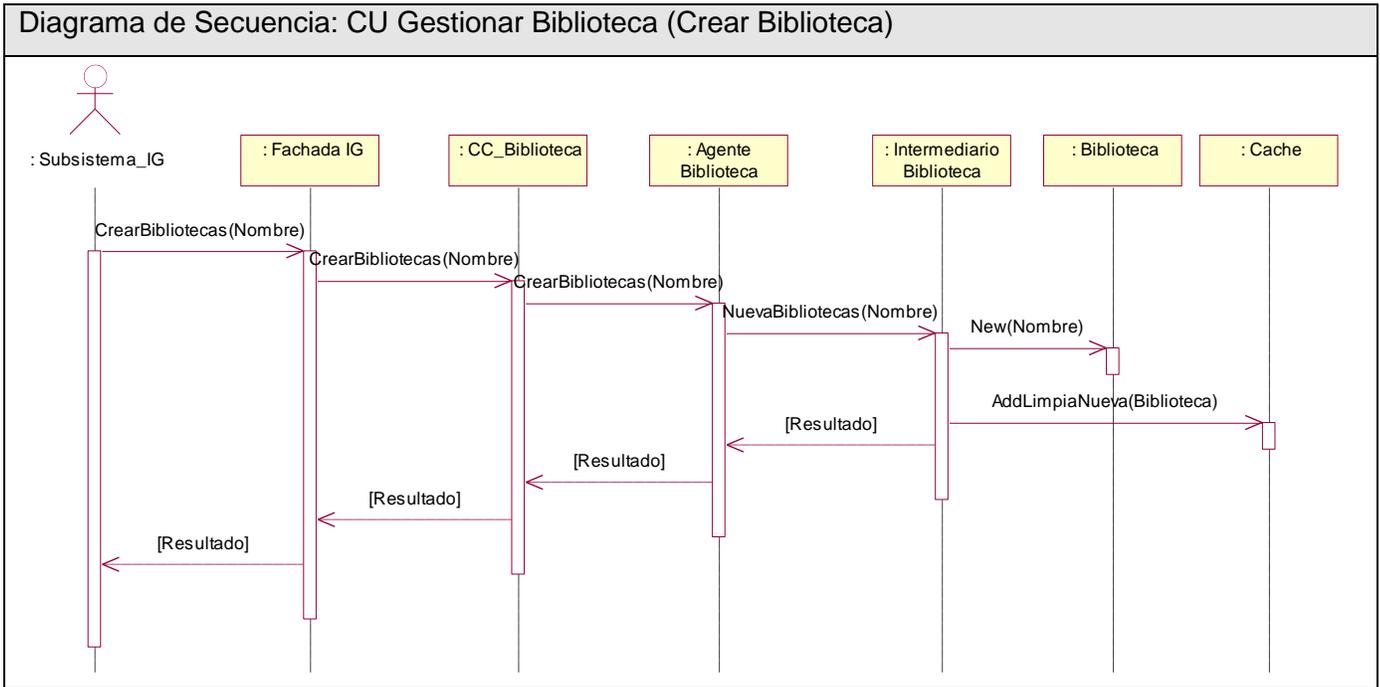


Figura 63 Diagrama de Secuencia Caso de Uso: Gestionar Biblioteca (Escenario Crear Biblioteca)

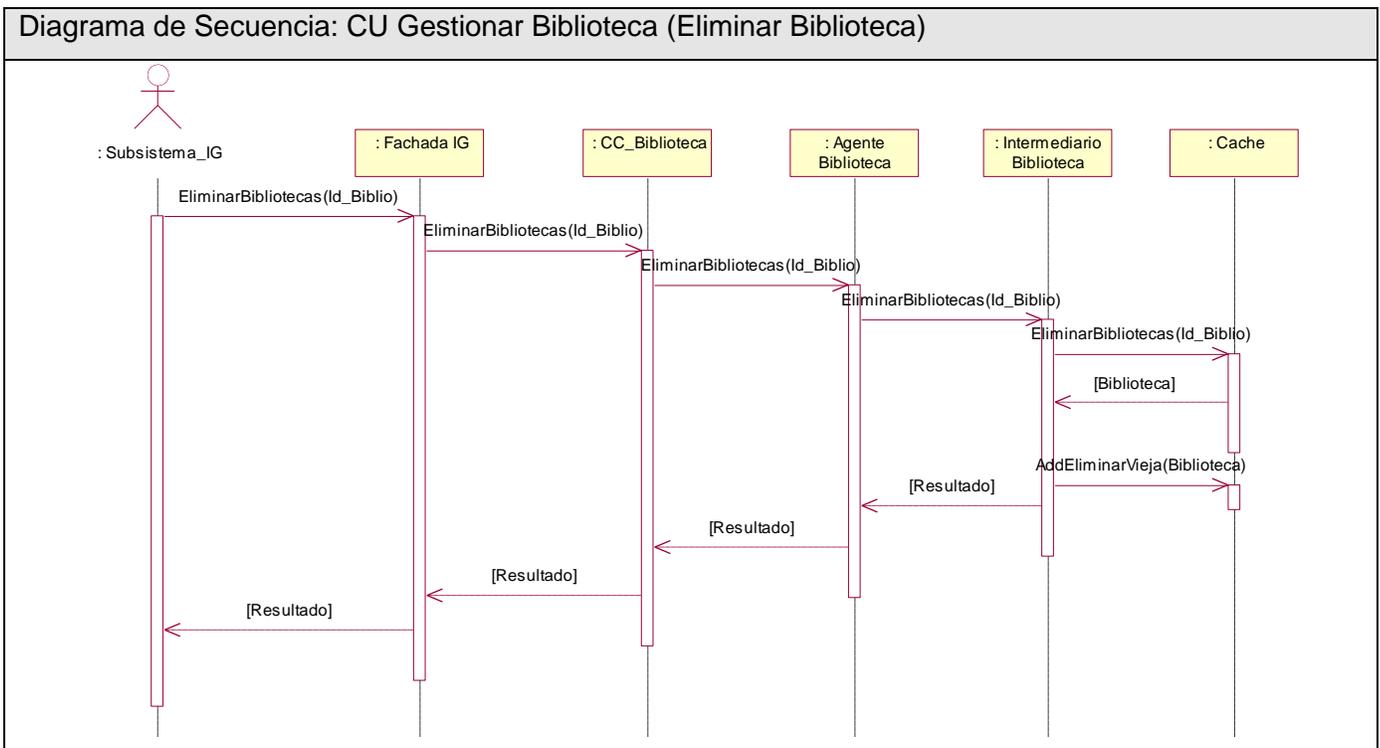


Figura 64 Diagrama de Secuencia Caso de Uso: Gestionar Biblioteca (Escenario Eliminar Biblioteca)

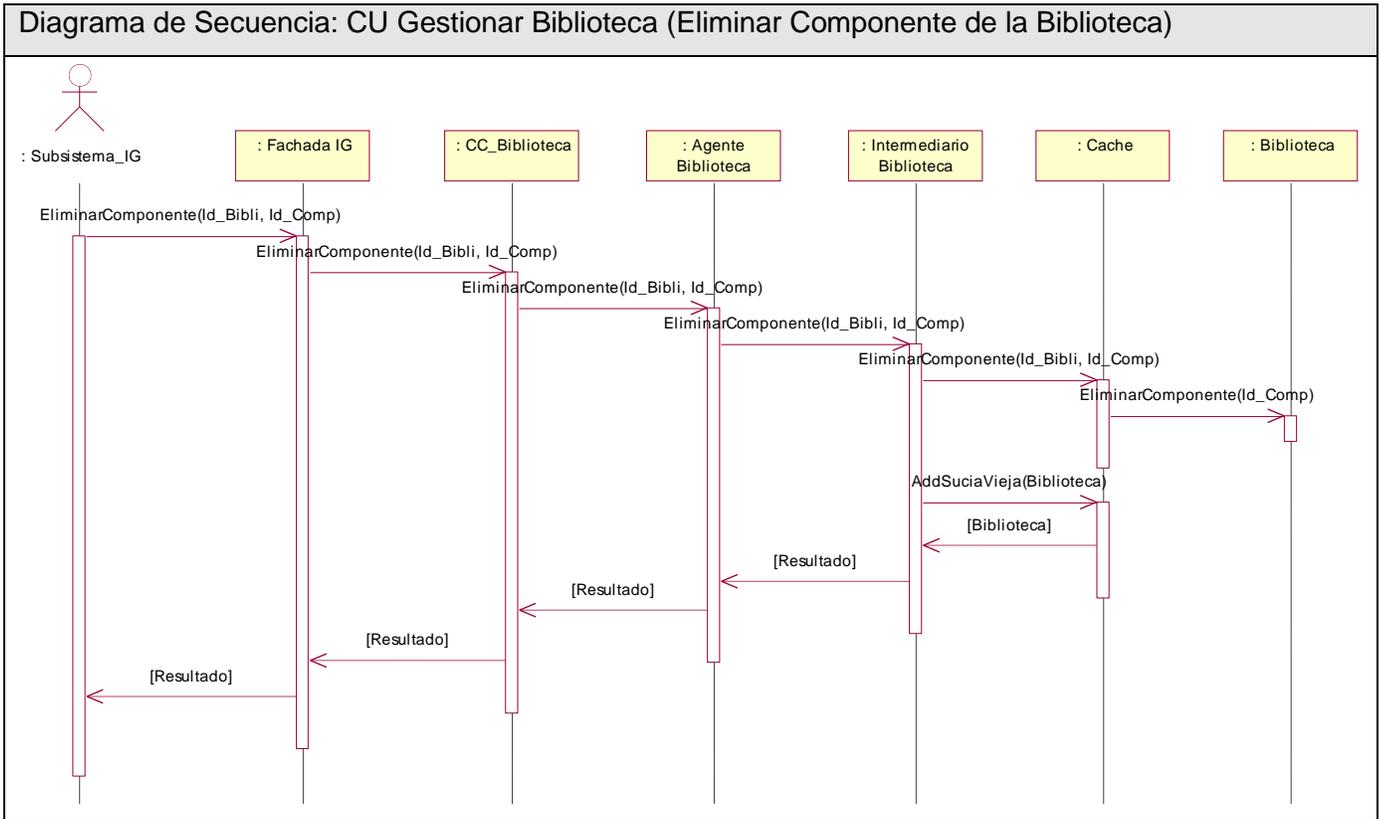


Figura 65 Diagrama de Secuencia Caso de Uso: Gestionar Biblioteca (Escenario Eliminar Componente de la Biblioteca)

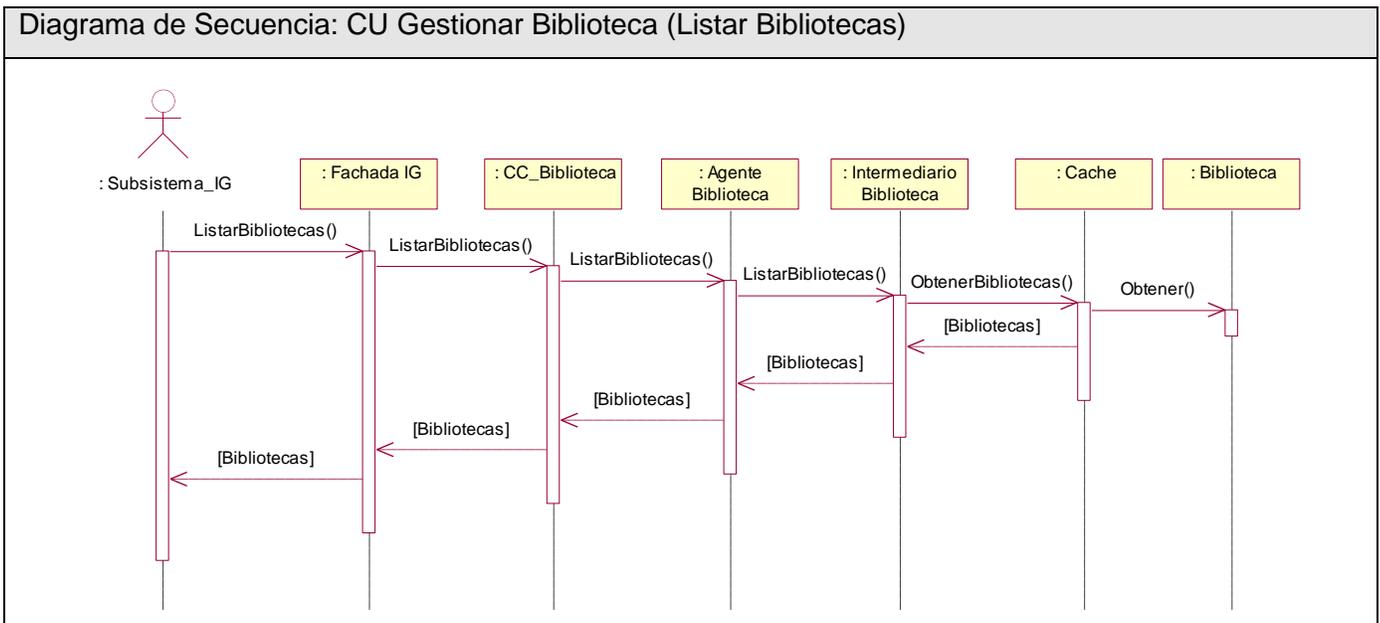


Figura 66 Diagrama de Secuencia Caso de Uso: Gestionar Biblioteca (Escenario Listar Bibliotecas)

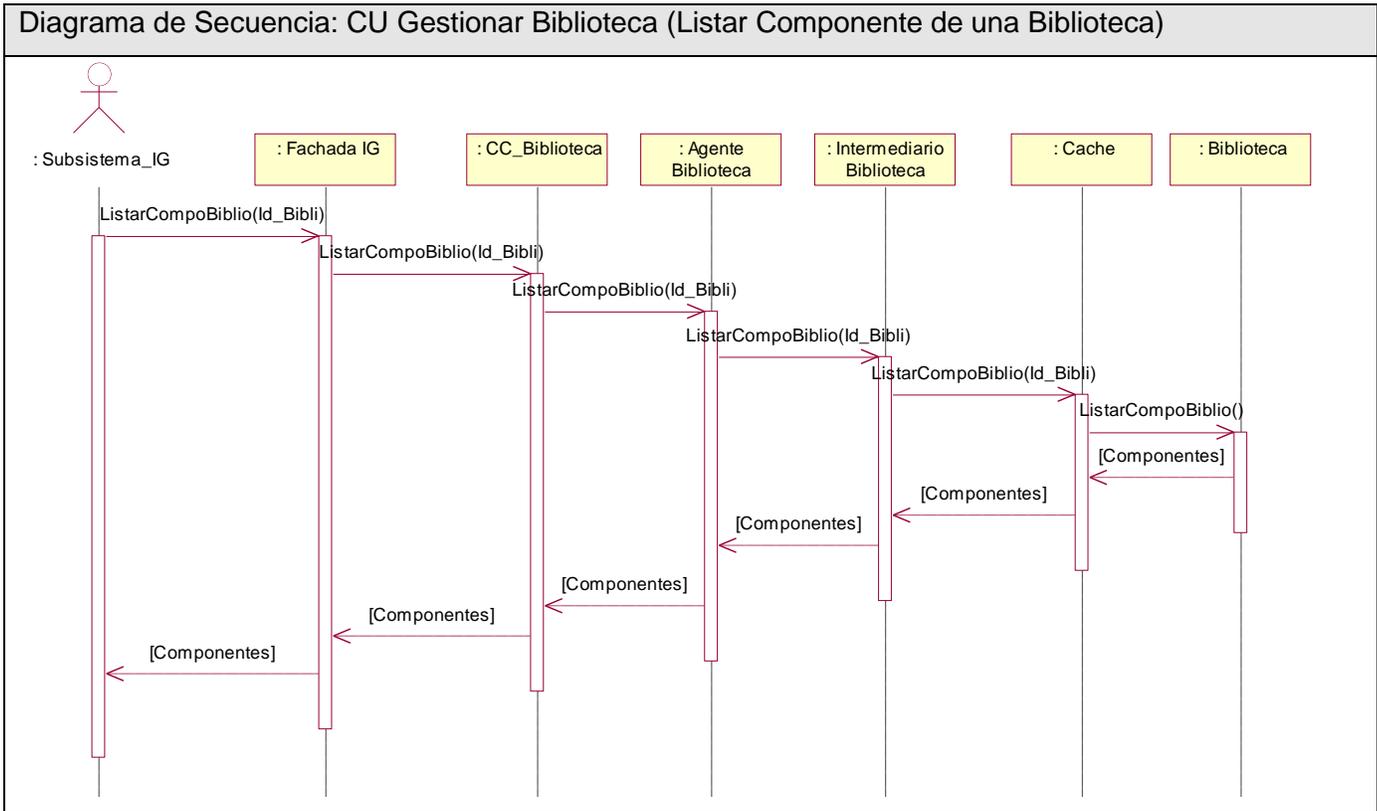


Figura 67 Diagrama de Secuencia Caso de Uso: Gestionar Biblioteca (Escenario Listar Componentes de una Biblioteca)

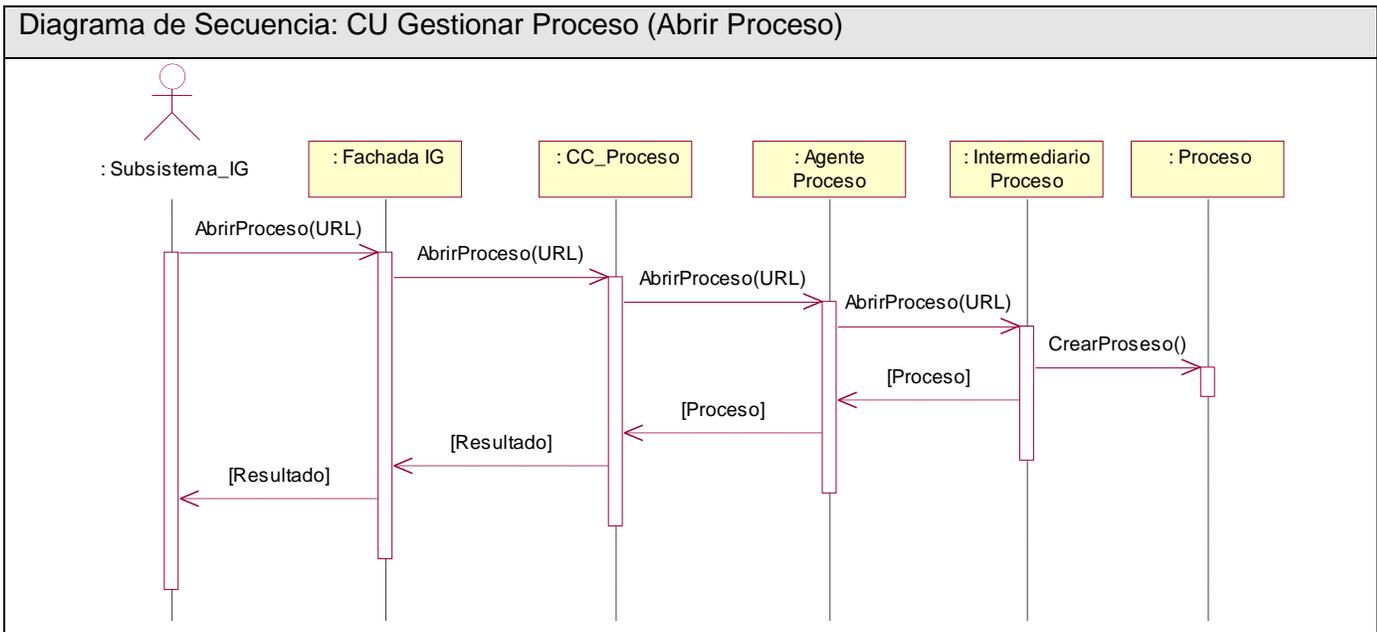


Figura 68 Diagrama de Secuencia Caso de Uso: Gestionar Proceso (Escenario Abrir Proceso)

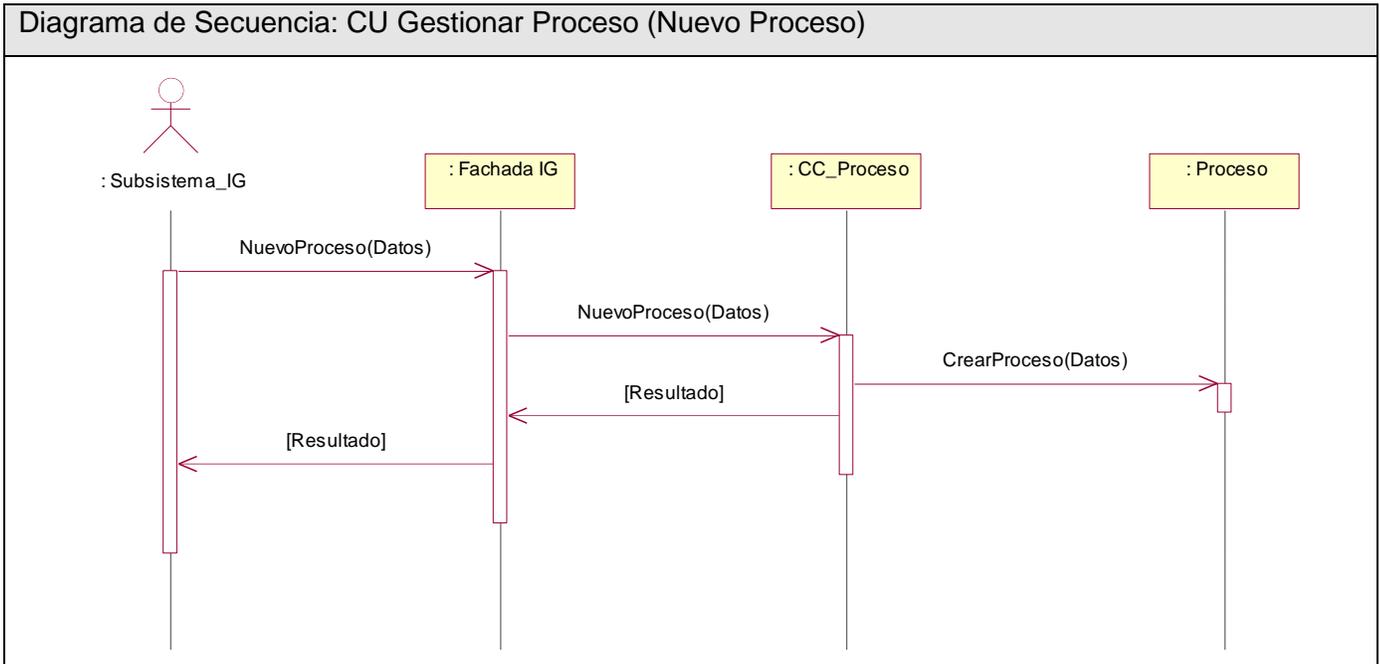


Figura 69 Diagrama de Secuencia Caso de Uso: Gestionar Proceso (Escenario Nuevo Proceso)

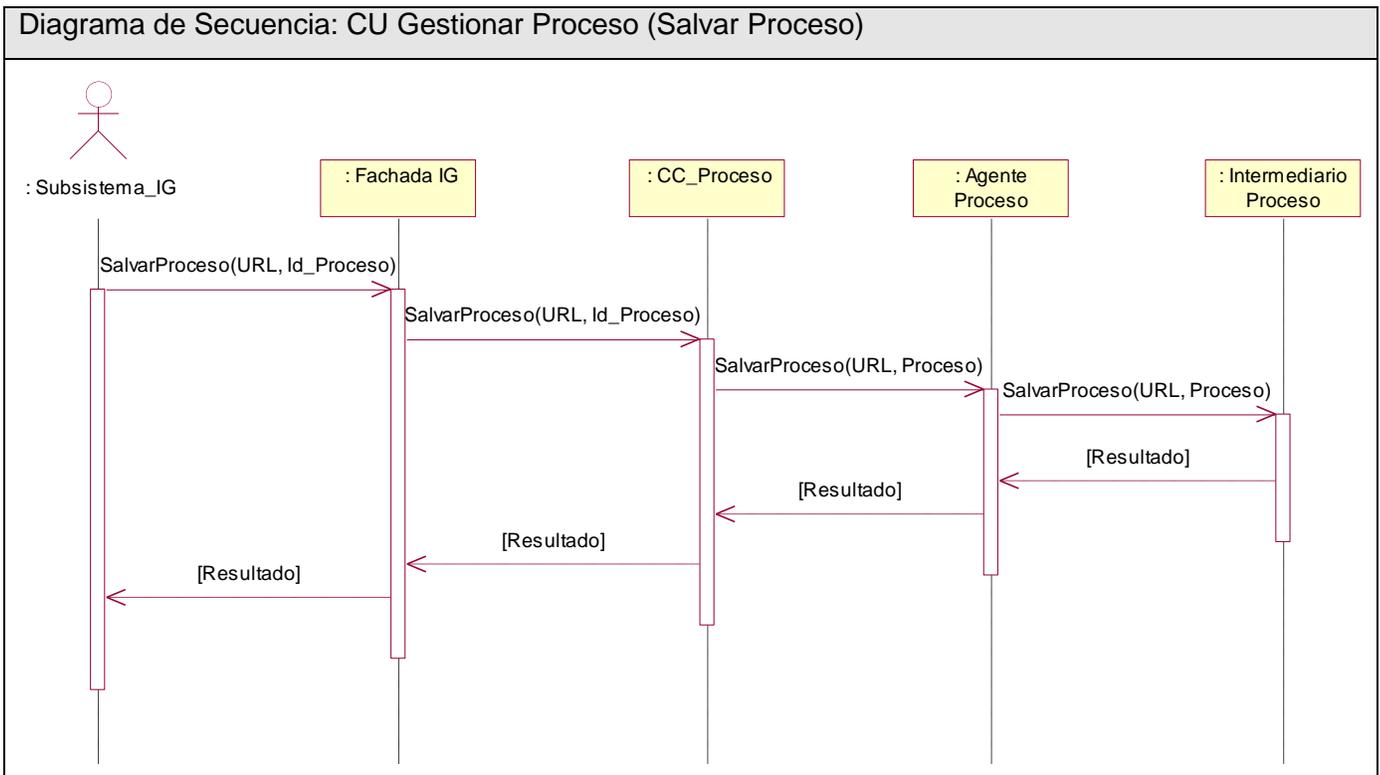


Figura 70 Diagrama de Secuencia Caso de Uso: Gestionar Proceso (Escenario Salvar Proceso)

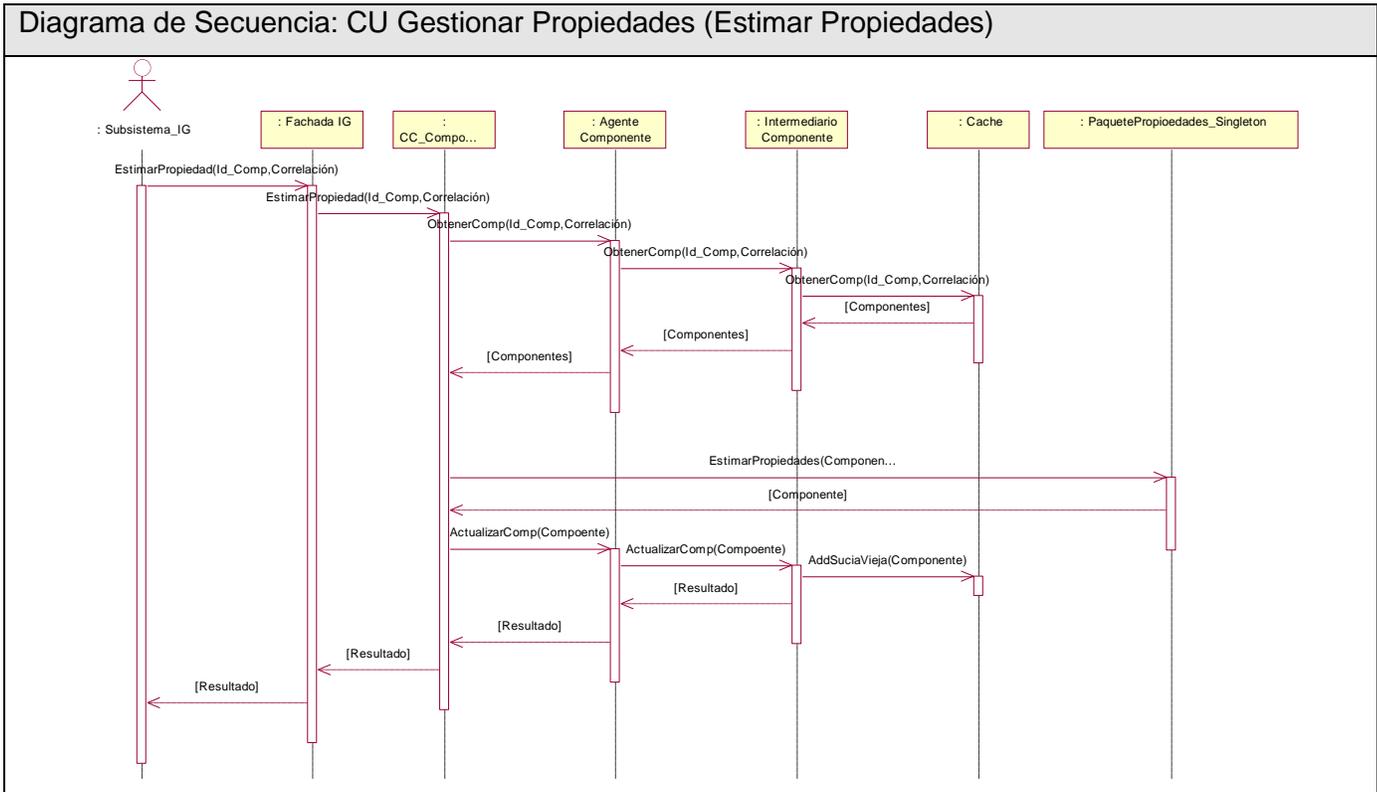


Figura 71 Diagrama de Secuencia Caso de Uso: Gestionar Propiedades (Escenario Estimar Propiedades)

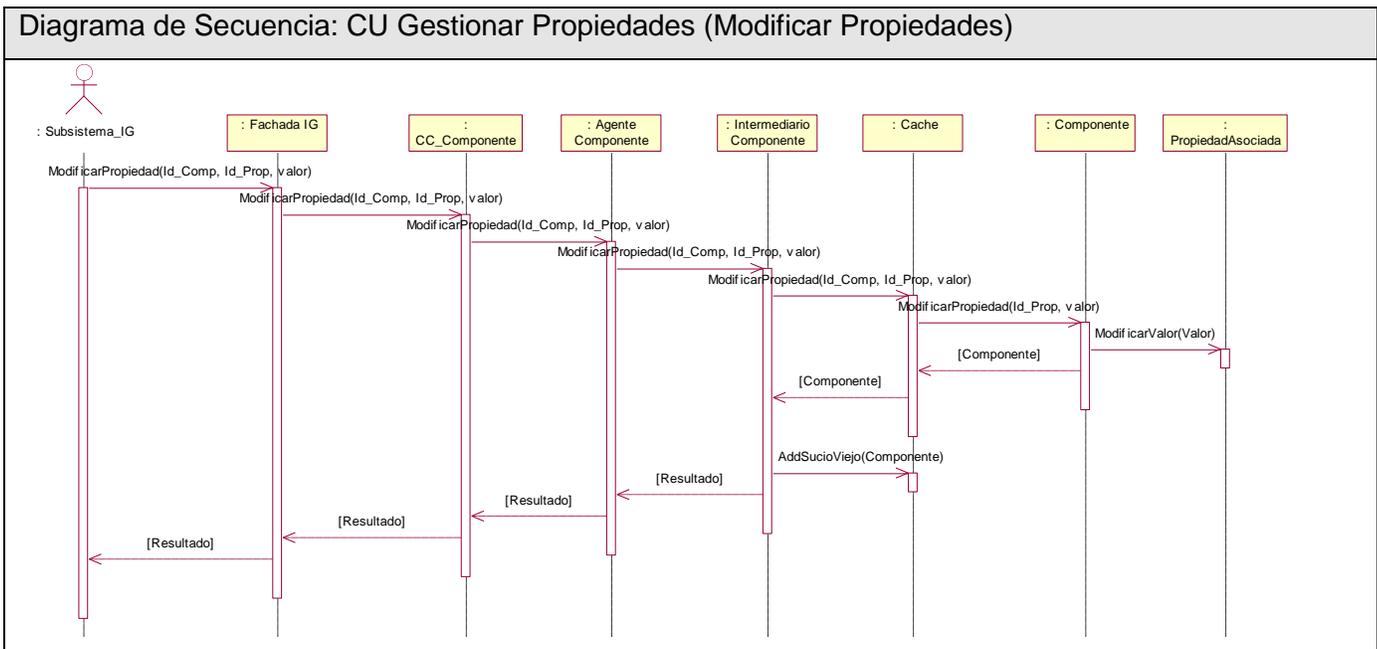


Figura 72 Diagrama de Secuencia Caso de Uso: Gestionar Propiedades (Escenario Modificar Propiedades)

Diagrama de Secuencia: CU Manipular Unidades de Medida (Manipular UM Componente)

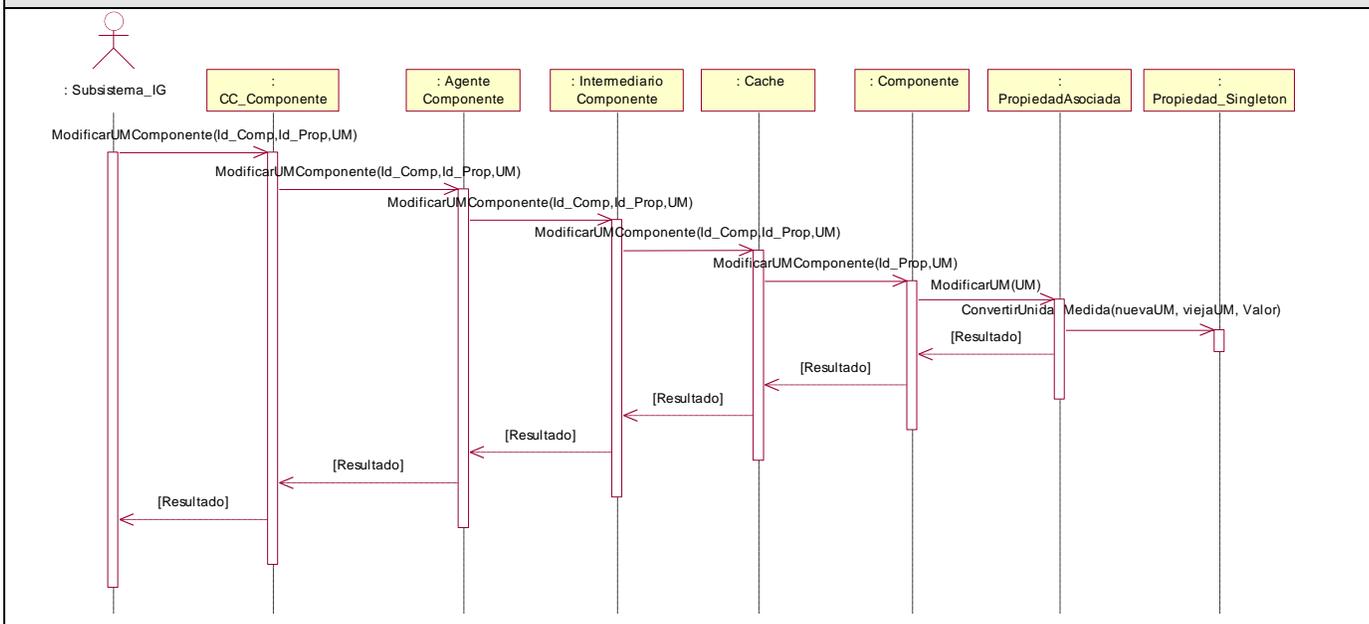


Figura 73 Diagrama de Secuencia Caso de Uso: Manipular Unidades de Medida (Escenario Manipular UM Componente)

Diagrama de Secuencia: CU Manipular Unidades de Medida (Manipular UM Corriente)

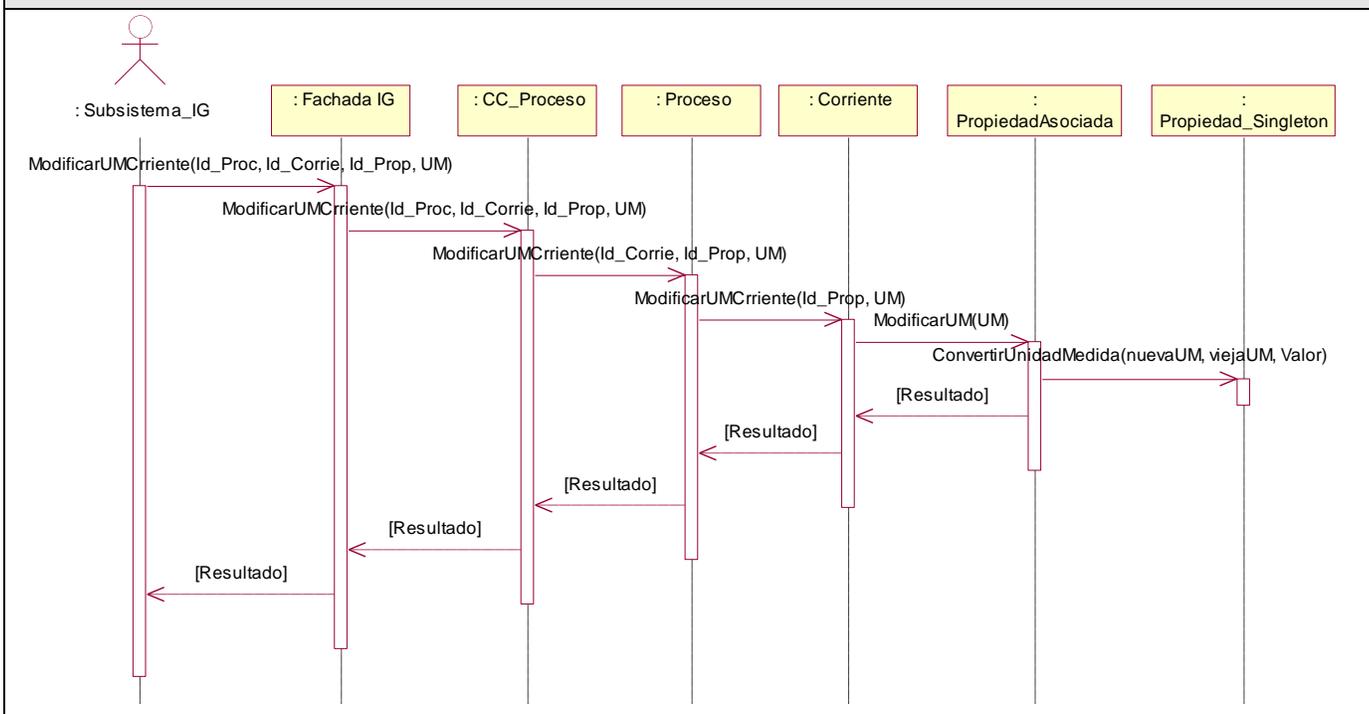


Figura 74 Diagrama de Secuencia Caso de Uso: Manipular Unidades de Medida (Escenario Manipular UM Corriente)

Diagrama de Secuencia: CU Manipular Unidades de Medida (Manipular UM Unidad Asociada)

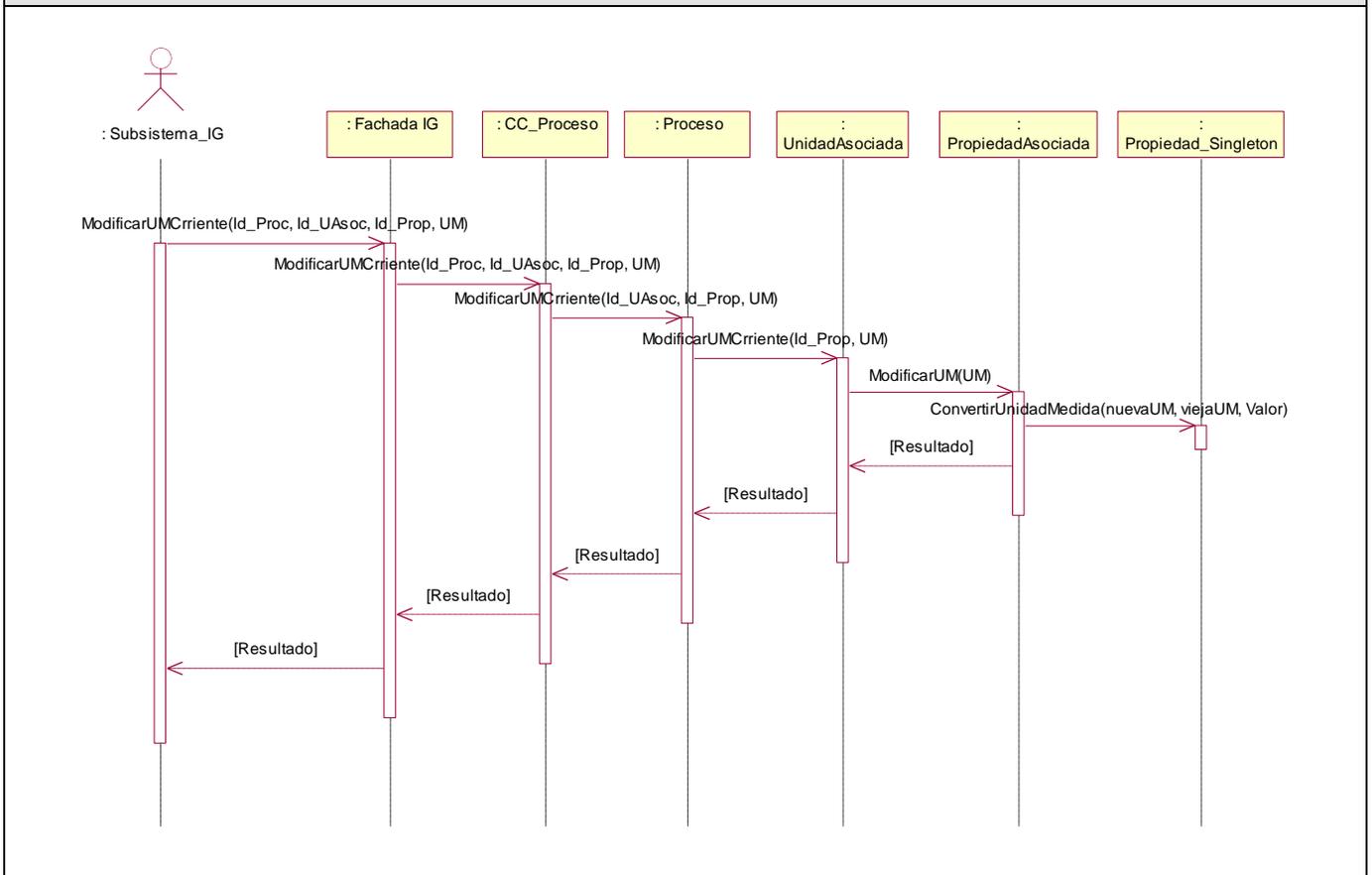


Figura 75 Diagrama de Secuencia Caso de Uso: Manipular Unidades de Medida (Escenario Manipular UM Unidad Asociad