

**Universidad de las Ciencias Informáticas**

**Facultad 3**



**Título: Análisis y Diseño de una herramienta interactiva de simulación de procesos: Nodo Virtual de Procesos. Segunda Iteración.**

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

**Autores:** Eliober Cleger Despaigne

Annarella María Tornés Montes de Oca

**Tutor:** Ing. Yalice Gámez Batista

**Co-tutor:** Ing. Mailen Edith Escobar Pompa

**Asesor:** Ing. Ameirys Betancourt Vázquez

Ciudad de la Habana

Abril 2009

*“Lo importante no es hacer cosas extraordinarias  
sino las cosas ordinarias extraordinariamente bien”*

*Ernesto Che Guevara*

## DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor de este trabajo y autorizo a la Facultad 3 de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmo la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

Eliober Cleger Despaigne

Annarella María Tornés Montes de Oca

---

Firma del Autor

---

Firma del Autor

Yalice Gámez Batista

---

Firma del Tutor

## *Agradecimientos*

*A Mamá y a Papá, por los regaños y por estar siempre ahí cuando los necesito.*

*A Rey, por incitarme a estudiar más y por ser mi súper hermano.*

*A Yalíce, por ayudarnos siempre con la tesis.*

*A Ameirys y a Mailén, por su apoyo incondicional.*

*A mis amistades.*

*Annarella*

*Agradecer primero a mis padres y a mi tía por su cariño y devoción, los quiero mucho a los tres.*

*A mi hermana y amiga Yadira por estar siempre ahí, cuidarme y quererme.*

*A mis vecinos Norma, Evin, Irina, Yamila, Nereyda, Gelenis, por ayudarme siempre que lo necesito.*

*A Eva que tanto me quiere y me ayuda, al loco de Alfonso mi amigo de siempre, y a Bárbara que tanto espera de mí.*

*A todos mis tíos que siempre me han tenido como su otro hijo.*

*A Yalíce y a Yoan por su apoyo incondicional en estos 5 años.*

*A Mailén y Ameirys por su aporte a nuestro trabajo.*

*A mis amigos de la vieja guardia (Baraguá), a mis amigos de ahora y a los de siempre, a mi motivo de placer y disgusto.*

*A todos los que me ayudaron a salir adelante.*

*Eliober*

## *Dedicatoria*

*A Mamá y a Papá, por todo lo que han sacrificado por mí.*

*A Rey, por adentrarme en la informática.*

*A Elaine, por su atención y cariño.*

*A Pedro, por aceptarme como soy.*

*A Abuela.*

*Annarella*

*Al compañero Fidel, motor impulsor de todas las obras que han hecho posible mi formación.*

*A tres personas que de conjunto han hecho de mí un hombre de bien, mis padres y mi tía Minerva.*

*A mi prima Yadira por su preocupación y su apoyo incondicional.*

*A mis hermanitos Elioner y Frey que alegran cada momento de mi vida.*

*A mis familiares, vecinos y amigos, quienes han dado muestra en todo momento de preocupación y entrega a mi porvenir.*

*A todo aquel que aunque no esté, haya contribuido a mi buen desempeño.*

*Eliober*

## RESUMEN

En la actualidad las Tecnologías de la Información y las Comunicaciones (TICs) forman parte de la cultura tecnológica que rodea el mundo y con la que se debe convivir. La gama de servicios que brindan posibilita la creación de modelos, procesos y programas; lo que permite controlar y simular actividades reales o virtuales. Para poder ejecutar estas acciones en el curso 2007-2008 se propuso modelar desde el punto de vista informático un Nodo Virtual de Procesos (NVP) que permitiera: crear modelos de procesos industriales, realizar simulaciones, administrar usuarios, límites y controladores, y consultar reportes; con el objetivo de proveer a los estudiantes de la carrera de Ingeniería Automática de una herramienta que les permitiera consolidar los contenidos asociados a procesos. Partiendo de esta propuesta inicial, en el presente Trabajo de Diploma se realizó una segunda iteración que consistió en el refinamiento de los requisitos y en completar la arquitectura para lograr una implementación eficiente del sistema.

Para ello se realizó un estudio sobre: conceptos asociados a nodos virtuales, el comportamiento de las funcionalidades previstas inicialmente para el NVP en otros sistemas similares en el mundo, metodologías de desarrollo de software, herramientas de modelado, lenguajes de programación, sistemas gestores de base de datos, además de otros aspectos asociados a la Ingeniería de Requisitos (IR) y al flujo de trabajo Análisis y Diseño definido en el Proceso Unificado de Desarrollo (RUP). Finalmente se aplicaron métricas dirigidas a evaluar la calidad de los artefactos generados, obteniéndose resultados satisfactorios.

## PALABRAS CLAVE

“Nodo Virtual, Simulación de Procesos, Procesos Industriales, Análisis, Diseño”

## ÍNDICE

<b>INTRODUCCIÓN</b> .....	<b>1</b>
<b>CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA</b> .....	<b>7</b>
INTRODUCCIÓN .....	7
1.1 NODO VIRTUAL DE SIMULACIÓN DE PROCESOS. ....	7
1.2 ANÁLISIS DEL COMPORTAMIENTO DE LAS FUNCIONALIDADES A REALIZAR POR EL NVP EN OTROS SISTEMAS SIMILARES. ....	9
1.3 CONTROLADORES. ....	12
1.3.1 <i>Protocolos de comunicación asociados</i> .....	14
1.4 METODOLOGÍAS DE DESARROLLO DE SOFTWARE.....	17
1.4.1 <i>Programación Extrema</i> . ....	18
1.4.2 <i>SCRUM</i> . ....	20
1.4.3 <i>Proceso Unificado de Desarrollo (RUP)</i> . ....	21
1.5 HERRAMIENTAS DE MODELADO.....	22
1.5.1 <i>ArgoUML</i> .....	23
1.5.2 <i>Rational Rose</i> .....	25
1.5.3 <i>Visual Paradigm</i> .....	26
1.6 LENGUAJES DE PROGRAMACIÓN.....	28
1.6.1 <i>Lenguaje C++</i> . ....	28
1.6.2 <i>Lenguaje Java</i> . ....	28
1.6.3 <i>Lenguaje C#</i> . ....	28
1.7 ANÁLISIS. ....	29
1.7.1 <i>Principios del Análisis</i> . ....	29
1.7.2 <i>Rol de analista de sistemas</i> . ....	30
1.7.3 <i>Actividades de la Ingeniería de Requisitos</i> . ....	31
1.7.4 <i>Técnicas para el levantamiento de requisitos</i> . ....	32
1.8 DISEÑO.....	34
1.8.1 <i>Calidad del diseño</i> .....	34
1.8.2 <i>Principios del diseño</i> . ....	35
1.9 SISTEMAS DE GESTIÓN DE BASES DE DATOS. ....	36
1.9.1 <i>MySQL</i> .....	36

1.9.2 PostgreSQL.....	37
CONCLUSIONES .....	39
<b>CAPÍTULO 2: REFINAMIENTO DE REQUISITOS.....</b>	<b>41</b>
INTRODUCCIÓN .....	41
2.1 CARACTERÍSTICAS DEL SISTEMA.....	41
2.1.1 Clases del Dominio.....	41
2.2 REDEFINICIÓN DE REQUERIMIENTOS.....	43
2.2.1 Requisitos funcionales.....	44
2.2.2 Requisitos no funcionales.....	56
2.3 REESTRUCTURACIÓN DEL DIAGRAMA DE CASOS DE USO DEL SISTEMA.....	57
2.3.1 Especificación textual de los casos de uso.....	58
CONCLUSIONES .....	75
<b>CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL NVP .....</b>	<b>76</b>
INTRODUCCIÓN .....	76
3.1 ANÁLISIS.....	76
3.1.1 Diagrama de Clases del Análisis .....	77
3.1.2 Realización de casos de uso del análisis.....	80
3.2 DISEÑO.....	83
3.2.1 Arquitectura definida para el sistema.....	83
3.2.2 Justificación de los Patrones de Diseño utilizados.....	84
3.2.3 Modelo de Diseño.....	85
3.2.4 Realización de Casos de Uso.....	94
CONCLUSIONES .....	100
<b>CAPÍTULO 4: VALIDACIÓN DE LOS RESULTADOS.....</b>	<b>101</b>
INTRODUCCIÓN .....	101
4.1 MÉTRICA DE LA CALIDAD DE ESPECIFICACIÓN DE LOS REQUISITOS.....	101
4.2 COMPLEJIDAD DE LOS SUBSISTEMAS.....	103
4.3 MÉTRICAS ORIENTADAS A CLASES.....	106
4.3.1 Tamaño de Clase (TC).....	106

4.3.2 Relaciones entre Clases (RC).....	109
CONCLUSIONES .....	113
<b>CONCLUSIONES .....</b>	<b>114</b>
<b>RECOMENDACIONES.....</b>	<b>115</b>
<b>BIBLIOGRAFÍA.....</b>	<b>116</b>
<b>GLOSARIO DE TÉRMINOS.....</b>	<b>120</b>
<b>ANEXOS .....</b>	<b>121</b>
<i>Anexo 1: Control manual: un lazo de control que incluye el proceso, un sensor, el operador y un elemento final de control (válvula).....</i>	<i>121</i>
<i>Anexo 2: Control centralizado por ordenador.....</i>	<i>121</i>
<i>Anexo 3: Estructura de un controlador difuso. ....</i>	<i>122</i>
<i>Anexo 4: Fases de XP.....</i>	<i>122</i>
<i>Anexo 5: Objetivos del Sistema.....</i>	<i>122</i>
<i>Anexo 6: Descripción detallada del CUS Gestionar_Conexión_Sistema. ....</i>	<i>129</i>
<i>Anexo 7: Descripción detallada del CUS Gestionar_Conexión_Proceso. ....</i>	<i>131</i>
<i>Anexo 8: Descripción detallada del CUS Establecer_Límites.....</i>	<i>135</i>
<i>Anexo 9: Descripción detallada del CUS Gestionar_Tipo_Proceso.....</i>	<i>138</i>
<i>Anexo 10: Descripción detallada del CUS Gestionar_Modelo_Proceso.....</i>	<i>141</i>
<i>Anexo 11: Descripción detallada del CUS Configurar_Proceso. ....</i>	<i>145</i>
<i>Anexo 12: Descripción detallada del CUS Simular_Proceso.....</i>	<i>147</i>
<i>Anexo 13: Descripción detallada del CUS Probar_Aplicación. ....</i>	<i>149</i>
<i>Anexo 14: Descripción detallada del CUS Gestionar_Controlador.....</i>	<i>151</i>
<i>Anexo 15: Diagrama de Clases del Análisis. CU Gestionar_Conexión_Sistema. ....</i>	<i>154</i>
<i>Anexo 16: Diagrama de Clases del Análisis. CU Gestionar_Conexión_Proceso.....</i>	<i>154</i>
<i>Anexo 17: Diagrama de Clases del Análisis. CU Establecer_Límites.....</i>	<i>155</i>
<i>Anexo 18: Diagrama de Clases del Análisis. CU Gestionar_Tipo_Proceso.....</i>	<i>155</i>
<i>Anexo 19: Diagrama de Clases del Análisis. CU Gestionar_Modelo_Proceso.....</i>	<i>156</i>
<i>Anexo 20: Diagrama de Clases del Análisis. CU Configurar_Proceso.....</i>	<i>156</i>
<i>Anexo 21: Diagrama de Clases del Análisis. CU Simular_Proceso, CU Probar_Aplicación.....</i>	<i>157</i>

<i>Anexo 22: Diagrama de Clases del Análisis. CU Gestionar_Controlador.</i>	157
<i>Anexo 23: Diagrama de Colaboración del Análisis. CU Graficar.</i>	158
<i>Anexo 24: Diagrama de Colaboración del Análisis. CU Mostrar_Registro_Variables.</i>	158
<i>Anexo 25: Diagrama de Colaboración del Análisis. CU Activar_Automáticamente_Procesos.</i>	159
<i>Anexo 26: Diagrama de Colaboración del Análisis. CU Priorizar_Número_Procesos.</i>	159
<i>Anexo 27: Diagrama de Colaboración del Análisis. CU Mostrar_Registro_Usuario.</i>	160
<i>Anexo 28: Diagrama de Colaboración del Análisis. CU Mostrar_Procesos_Activos.</i>	160
<i>Anexo 29: Diagrama de Colaboración del Análisis. CU Mostrar_Usuarios_Conectados.</i>	161
<i>Anexo 30: Diagrama de Colaboración del Análisis. CU Mostrar_Usuarios_Proceso.</i>	161
<i>Anexo 31: Diagrama de Colaboración del Análisis. CU Mostrar_Procesos_Activados_Usuario.</i>	162
<i>Anexo 32: Diagrama de Colaboración del Análisis. CU Gestionar_Conexión_Sistema.</i>	162
<i>Anexo 33: Diagrama de Colaboración del Análisis. CU Gestionar_Conexión_Proceso.</i>	163
<i>Anexo 34: Diagrama de Colaboración del Análisis. CU Establecer_Límites.</i>	164
<i>Anexo 35: Diagrama de Colaboración del Análisis. CU Gestionar_Tipo_Proceso.</i>	165
<i>Anexo 36: Diagrama de Colaboración del Análisis. CU Gestionar_Modelo_Proceso.</i>	167
<i>Anexo 37: Diagrama de Colaboración del Análisis. CU Configurar_Proceso.</i>	168
<i>Anexo 38: Diagrama de Colaboración del Análisis. CU Simular_Proceso.</i>	169
<i>Anexo 39: Diagrama de Colaboración del Análisis. CU Gestionar_Controlador.</i>	169
<i>Anexo 40: Interfaz de Usuario. IU_Inicio.</i>	171
<i>Anexo 41: Interfaz de Usuario. IU_Registrar_Usuario, IU_Autenticar.</i>	171
<i>Anexo 42: Interfaz de Usuario. IU_Principal_Usuario, IU_Principal_Administrador.</i>	172
<i>Anexo 43: Interfaz de Usuario. IU_Gestionar_Conexión_Proceso.</i>	172
<i>Anexo 44: Interfaz de Usuario. IU_Configurar_Proceso.</i>	173
<i>Anexo 45: Interfaz de Usuario. IU_Mostrar_Registro_Variables.</i>	174
<i>Anexo 46: Interfaz de Usuario. IU_Seleccionar_Variables, IU_Graficar.</i>	174
<i>Anexo 47: Interfaz de Usuario. IU_Desconectar_Usuario.</i>	175
<i>Anexo 48: Interfaz de Usuario. IU_Gestionar_Rol_Administrador.</i>	175
<i>Anexo 49: Interfaz de Usuario. IU_Establecer_Límites.</i>	176
<i>Anexo 50: Interfaz de Usuario. IU_Gestionar_Tipo_Proceso.</i>	176
<i>Anexo 51: Interfaz de Usuario. IU_Gestionar_Modelo_Proceso.</i>	177
<i>Anexo 52: Interfaz de Usuario. IU_Gestionar_Controlador.</i>	177

<i>Anexo 53: Interfaz de Usuario. IU_Activar_Automáticamente</i> .....	178
<i>Anexo 54: Interfaz de Usuario. IU_Priorizar_Tipos_Procesos</i> .....	178
<i>Anexo 55: Interfaz de Usuario. IU_Reportes</i> .....	179
<i>Anexo 56: Diagrama de clases del Diseño. CU Gestionar_Conexión_Sistema</i> .....	179
<i>Anexo 57: Diagrama de clases del Diseño. CU Gestionar_Conexión_Proceso</i> .....	180
<i>Anexo 58: Diagrama de clases del Diseño. CU Establecer_Límites</i> .....	180
<i>Anexo 59: Diagrama de clases del Diseño. CU Gestionar_Tipo_Proceso</i> .....	181
<i>Anexo 60: Diagrama de clases del Diseño. CU Gestionar_Modelo_Proceso</i> .....	181
<i>Anexo 61: Diagrama de clases del Diseño. CU Configurar_Proceso</i> .....	182
<i>Anexo 62: Diagrama de clases del Diseño. CU Gestionar_Controlador</i> .....	182
<i>Anexo 63: Diagrama de Secuencia. CU Desconectar_Usuario_Proceso</i> .....	183
<i>Anexo 64: Diagrama de Secuencia. CU Gestionar_Rol_Administrador</i> .....	183
<i>Anexo 65: Diagrama de Secuencia. CU Graficar</i> .....	185
<i>Anexo 66: Diagrama de Secuencia. CU Mostrar_Procesos_Activos</i> .....	185
<i>Anexo 67: Diagrama de Secuencia. CU Mostrar_Usuarios_Conectados</i> .....	186
<i>Anexo 68: Diagrama de Secuencia. CU Mostrar_Usuarios_Proceso</i> .....	186
<i>Anexo 69: Diagrama de Secuencia. CU Mostrar_Procesos_Activados_Usuario</i> .....	187
<i>Anexo 70: Diagrama de Secuencia. CU Gestionar_Conexión_Sistema</i> .....	187
<i>Anexo 71: Diagrama de Secuencia. CU Gestionar_Conexión_Proceso</i> .....	188
<i>Anexo 72: Diagrama de Secuencia. CU Establecer_Límites</i> .....	189
<i>Anexo 73: Diagrama de Secuencia. CU Gestionar_Tipo_Proceso</i> .....	190
<i>Anexo 74: Diagrama de Secuencia. CU Gestionar_Modelo_Proceso</i> .....	192
<i>Anexo 75: Diagrama de Secuencia. CU Configurar_Proceso</i> .....	193
<i>Anexo 76: Diagrama de Secuencia. CU Gestionar_Controlador</i> .....	194

## INTRODUCCIÓN

Las Tecnologías de la Información y las Comunicaciones (TICs) forman parte de la cultura tecnológica que rodea el mundo y con la que se debe convivir. Amplían las capacidades físicas y mentales y las posibilidades de desarrollo social. Su incidencia es perceptible no sólo en la informática y sus tecnologías asociadas: telemática y multimedia, sino también en los medios de comunicación de todo tipo: comunicación social y comunicación interpersonales tradicionales con soporte tecnológico como el teléfono, fax y los otros tantos que se conocen.

La Modelación y Simulación Computacional (MCS) es un ejemplo de servicio que constituye un eslabón fundamental para llevar a cabo proyectos que requieran de su uso. La MCS se ha venido desarrollando desde la década de los 40, un ejemplo de ello es la investigación realizada para la creación y construcción de las primeras bombas atómicas; pero adquieren auge e inusitado impulso a partir de los años 80, donde se ha visto un notable aumento en el desarrollo de la industria. (1)

Hoy en día, la simulación va más allá del mero modelado de sistemas de alto nivel y se centra más en la creación de prototipos de software, consiguiendo un ahorro de tiempo y dinero a las empresas en sus desarrollos. Según De Perthuis<sup>1</sup> (2), una de las áreas de la simulación virtual en la que más se ha avanzado es, precisamente, en el análisis de elementos finitos (FEA). “Esta capacidad permite a los ingenieros de diseño evaluar cómo afectan los cambios en el diseño al rendimiento del producto, acelerando el tiempo de desarrollo y mejorando el conocimiento del comportamiento físico de diseño. Se obtiene así una mayor fiabilidad respecto a la calidad del producto durante la fase de desarrollo”. Y es que, en el ámbito de la gestión del ciclo de vida de los productos, el uso de la simulación se ha convertido en un proceso cada vez más vital para el rápido desarrollo de productos innovadores. “Acelera el ciclo de desarrollo de productos al proporcionar un acceso oportuno a la información apropiada, a través de seguras funcionalidades de almacenamiento, búsqueda y recuperación, específicas para los datos y procesos de simulación”, apunta De Perthuis.

Aunque los sectores que más utilizan este tipo de tecnologías son la automoción, el transporte y la industria aeroespacial y aeronáutica, otros mercados como los fabricantes de bienes de consumo, la

---

1 Eric de Perthuis, director general para España y Portugal de Dassault Systèmes.

industria farmacéutica y biotecnología o el sector energético y petroquímico también han descubierto los beneficios que pueden aportar a la hora de desarrollar nuevos productos y soluciones innovadoras.

En la actualidad existen herramientas informáticas que permiten realizar simulaciones de procesos, pero todas ellas tienen limitaciones y restricciones que las hacen parcialmente ineficientes, por lo que para solucionar estos problemas se han implementado otras aplicaciones que hacen uso de nodos virtuales en los cuales se simulan los procesos.

Estados Unidos, Inglaterra, Alemania, Austria y Holanda son los países que mayor progreso han logrado en el desarrollo de este tipo de software: en EUA, la Universidad de Pennsylvania creó DETER, un simulador de redes que proporciona una plataforma de simulación y evaluación, con altos niveles de flexibilidad y fidelidad en términos de hardware y configuración de redes, compatibilidad de código y métricas de redes.

Por otra parte, en Inglaterra, la Universidad de Nottingham Trent desarrolló una aplicación para la simulación de sistemas de agua, y el laboratorio Nacional de Lawrence Berkeley en EUA desarrolló un simulador para pozos de petróleo todas basadas en nodos virtuales. En Austria, la Universidad de Viena diseñó NetSim, esta herramienta como principal tarea simula el bus<sup>2</sup> (2) y además tiene funciones para analizar el tráfico, estas funciones pueden ser usadas para optimizar el uso de hardware en una instalación. Y por último en Holanda, la Universidad de Ámsterdam implementó Virtual Vascular Surgery, un sistema prototipo para cirugía vascular virtual que puede usarse durante la planificación del pre-operatorio o como una valiosa herramienta en el entrenamiento de cirujanos vasculares principiantes. El prototipo utiliza simulaciones y visualizaciones avanzadas y distribuidas para apoyar a la toma de decisiones en el pre-operatorio.

Se conoce del desarrollo de Sistemas de Control Industrial (ICS) en Cuba. Por ejemplo en el año 1991 fue implementado por ingenieros de la Unión del Níquel en la provincia de Holguín un ICS conocido como EROS que se usó en 27 plantas de diferentes tipos; y en la actualidad se está desarrollando en la Universidad de las Ciencias Informáticas (UCI) un sistema SCADA (Supervisory Control and Data Acquisition) para el trabajo con imágenes utilizando una plataforma virtual de procesamiento. También se

---

2 El bus, representa básicamente una serie de cables mediante los cuales pueden cargarse datos en la memoria y desde allí transportarse a la CPU. El bus se controla y maneja desde la CPU.

conoce de la existencia en dicha universidad del Proyecto SIMPRO el cual se dedica al desarrollo de simuladores.

Por todo lo anteriormente expresado se puede concluir que tanto a nivel mundial como en Cuba no se tienen grandes experiencias en el desarrollo de nodos virtuales para la simulación de procesos. A esto se une el hecho de que la carencia de una aplicación que realice estas actividades, trae consigo problemas para el desarrollo y comprensión de los contenidos docentes en la carrera de Ingeniería Automática; pues en ésta se imparten una serie de contenidos que proveen a los estudiantes de toda la teoría necesaria para ejercer como profesionales de la rama. Sin embargo no siempre estos conocimientos son fundamentados desde el punto de vista práctico con herramientas que fortalezcan la comprensión de los mismos.

En el curso 2007-2008 se propuso realizar el análisis y diseño de un NVP, que respondía a la necesidad de una herramienta adecuada para fortalecer los conocimientos adquiridos por los estudiantes de la carrera de Ingeniería Automática desde el punto de vista práctico. Del problema planteado inicialmente sólo se materializó el diseño de los casos de uso críticos y surgieron ambigüedades cuando se definieron las funcionalidades que debe tener el NVP.

Tomando todo lo anterior se concluye que la **situación problémica** consiste en que el análisis y diseño de la aplicación quedó incompleto, sólo se realizaron los casos de uso críticos para la arquitectura, dificultando su implementación, por lo que es necesario refinar los requerimientos y completar el análisis y diseño de la misma.

Según lo descrito anteriormente se puede llegar al siguiente **problema**:

¿Cómo transformar las necesidades de los estudiantes de la carrera de Ingeniería Automática a un lenguaje entendible por los desarrolladores, que permita la posterior implementación de un NVP?

Por consiguiente el **objeto de la investigación** sería, el Proceso de Desarrollo de Software, centrando la atención en el flujo de trabajo Análisis y Diseño definido en RUP, identificado como campo de acción de la investigación.

Lo que conlleva a la siguiente **hipótesis**:

Si se realiza un correcto análisis y diseño para un NVP, se podrá implementar el mismo de forma más eficiente, de modo que los estudiantes de la carrera de Ingeniería Automática puedan adquirir y fortalecer los contenidos asociados con procesos.

A partir de ésta se formulan los **objetivos** de este trabajo, los cuales se precisan a continuación:

### **Objetivo General**

- Realizar el análisis y diseño de un Nodo Virtual que simule procesos industriales.

### **Objetivos Específicos**

- Analizar el comportamiento de las funcionalidades previstas para el NVP en otros sistemas similares en el mundo.

Tarea 1: Estudio de la bibliografía con el fin de caracterizar cómo se implementan las funcionalidades previstas para el NVP en otros sistemas.

- Refinar los requerimientos propuestos inicialmente para el NVP.

Tarea 2: Revisión de los requerimientos con el fin de eliminar ambigüedades y añadir nuevas funcionalidades.

Tarea 3: Reestructuración del diagrama de casos de uso del sistema.

- Obtener artefactos del análisis.

Tarea 4: Confección de los diagramas de clases del análisis con los diagramas de colaboración correspondientes.

- Obtener artefactos del diseño.

Tarea 5: Confección de los diagramas de clases del diseño con los diagramas de secuencia correspondientes.

- Validar los resultados obtenidos durante el análisis y el diseño del NVP.

Tarea 6: Selección y aplicación de métricas para la validación de los resultados.

Con el propósito de desarrollar las tareas planteadas, se utilizaron los **métodos** de investigación siguientes:

### **Teóricos:**

La presente investigación se fundamenta en el **método dialéctico materialista** como método basado en el conocimiento de las leyes más generales del desarrollo. Se potencia la autorregulación de la actividad cognoscitiva de cada usuario lo que posibilita el logro de nuevos resultados a través de las conexiones y contradicciones manifestadas en los procesos. La aplicación, Nodo Virtual de Procesos, que se propone en unidad, interrelación e independencia con todos sus módulos facilitará desplegar un proceso de interacción de los usuarios con los diferentes procesos.

**Histórico lógico:** Se realiza un análisis de cómo se han implementado las funcionalidades previstas para el NVP en diferentes herramientas de simulación con el uso de nodos virtuales. Obteniendo una tendencia de cómo se debe comportar en la actualidad.

**Sistémico:** Se tienen en cuenta todas y cada una de las relaciones que se establecen entre los módulos dentro de la aplicación y las contradicciones que generan los mismos.

**Hipotético-deductivo:** A partir de la interpretación de la realidad se establecen posibles situaciones o resultados para llegar a conclusiones.

### **Empíricos**

**Consulta a especialistas:** Se empleó para comprobar la necesidad y la funcionalidad práctica de la aplicación, Nodo Virtual de Procesos que se propone en la presente investigación.

**La observación:** Mediante guías de observación se le dará seguimiento al desarrollo de la aplicación.

### **Métodos Particulares:**

#### **Las entrevistas y la toma de criterios de expertos:**

Propiciaron recoger las opiniones que sobre el tema de investigación poseen distintos especialistas concedores de la rama tratada.

### **La tormenta de ideas:**

Herramienta utilizada para posibilitar la generación de un elevado número de ideas, por parte de un grupo, y la presentación ordenada de éstas.

### **Como aporte esperado de esta investigación:**

**Aporte Teórico:** Refinamiento de los requerimientos del Nodo Virtual de Procesos, obtención y realización de los casos de uso para el flujo de trabajo definido por RUP como Análisis y Diseño.

La presente investigación consta de tres capítulos estructurados de la siguiente forma:

En el capítulo uno se realizará una breve reseña de los nodos virtuales, los controladores y los diferentes protocolos de comunicación asociados. Se llevará a cabo un estudio sobre las metodologías y herramientas a utilizar, para establecer una comparación que permita escoger las más adecuadas. Además se estudiarán los principales conceptos y características de los flujos de análisis y diseño.

En el capítulo dos se refinarán los requisitos previstos para el NVP, además de las descripciones de los casos de uso para una mayor comprensión de las acciones en cada uno y el esclarecimiento del diseño del sistema a construir.

En el capítulo tres se continúa con el flujo de Análisis y Diseño del sistema. Aquí se van a representar los artefactos correspondientes a este flujo como son: los diagramas de clases del análisis, los diagramas de colaboración correspondientes, el modelo de diseño, los diagramas de clases del diseño por subsistemas de diseño, los diagramas de secuencia. Todos estos artefactos servirán para lograr un mejor entendimiento del funcionamiento del sistema a construir y complementarán un diseño estable para su implementación.

En los capítulos dos y tres se reflejan en el cuerpo del trabajo de diploma los artefactos correspondientes a los casos de uso no críticos. Esto se debe a que en la primera iteración se generaron los artefactos asociados a los casos de uso críticos y en la presente iteración estos últimos solo fueron refinados.

Finalmente en el capítulo cuatro se realizará la validación de los resultados, mediante la aplicación de métricas.

## Capítulo 1: Fundamentación Teórica

### Introducción

Como una de las propuestas de solución a la problemática planteada en la introducción de este trabajo se pensó en un Nodo Virtual, ya que permite correr diferentes instancias del software en un único nodo (nodo físico) y cada instancia del software trabaja en un entorno de ejecución independiente (nodo virtual). Por tanto en este capítulo se pretende abordar los aspectos y conceptos relacionados con el Nodo Virtual. Se hará un análisis del comportamiento de las funcionalidades a realizar por el NVP en otros sistemas similares. Se realizará un estudio de las metodologías de desarrollo existentes y se explicarán las herramientas escogidas para dar solución a la problemática propuesta, pretendiendo dejar sentadas las bases teóricas para un correcto análisis y diseño del software.

#### 1.1 Nodo Virtual de Simulación de Procesos.

La simulación es reproducir el ambiente, las variables (rasgos, apariencia, características, contexto) de un sistema real. Es imitar una situación del mundo real en forma matemática.

La simulación constituye una técnica económica que permite ofrecer varios escenarios posibles de una situación y permite equivocarse sin provocar efectos sobre el mundo real (por ejemplo un simulador de vuelo o conducción). (3)

Definición:

Simulación es una técnica numérica para realizar experimentos en una computadora digital. Estos experimentos involucran ciertos tipos de modelos matemáticos y lógicos que describen el comportamiento de sistemas de negocios, económicos, sociales, biológicos, físicos o químicos a través de largos periodos de tiempo. (4)

Otra Definición:

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. (5)

En la actualidad existen muchas herramientas informáticas (6), (7), (8), (9), (10), que permiten realizar simulaciones de procesos industriales (en tiempo real o no), pero todas éstas tienen limitaciones en cuanto al número de recursos que necesitan para su implementación o en cuanto al número de procesos simultáneos que se pueden simular. Incluso, en su mayoría, o simulan los procesos o permiten probar aplicaciones reales, nunca las dos prestaciones.

Para dar solución a esta problemática se pensó en un Nodo Virtual ya que permite ejecutar diferentes instancias del software en un único nodo (nodo físico) y cada instancia del software trabaja en un entorno de ejecución independiente (nodo virtual).

Esta filosofía de trabajo permite la simulación simultánea de diferentes procesos concurrentes sin que interfieran uno con otros. La virtualización de nodos provee una vía de regular el acceso a recursos de hardware exclusivos de un determinado número de consumidores. En este caso los consumidores son los entornos de ejecución para cada proceso, los cuales están sujetos a las propiedades de la simulación. (6)

De ahí se derivan los siguientes requerimientos para la virtualización del nodo: (11)

- El parámetro más importante es minimizar los gastos de virtualización para preservar los recursos para el proceso en ejecución.
- Cada entorno de ejecución introducido por la virtualización del nodo debe ser tan transparente como sea posible para los restantes. Esto es importante para que la medición de la implementación no sufra modificaciones en comparación con la real.

Hoy en día se conoce como nodo virtual de procesos a aquel software que permita implementar modelos de distintos procesos ya sea para su simulación o la prueba de aplicaciones en tiempo real, por lo que será necesario que varios procesos estén activos simultáneamente para requerir de la cantidad de nodos y computadoras. (12)

Para ello se establece como nodo a aquella estructura a la cual se interconectan varios elementos. No hay que pensar en un nodo como un elemento constituido solamente por una parte física, sino más bien considerarlo como una unidad funcional en donde tiene que haber tanto hardware como software.

Por otra parte, al ser el punto de conexión de dos o más elementos, el nodo por lo general tiene la capacidad de recibir información, procesarla y enrutarla a otro u otros nodos. De esta manera, un nodo

puede ser el punto de conexión para transmitir los datos, el punto desde el cual se distribuye los datos hacia otros nodos y el punto final al que se transmiten los datos.

Para el establecimiento de pruebas se necesita de una fracción de recursos de un nodo de prueba y un número de aplicaciones dirigidas a dispositivos de pocos recursos de forma tal que se puedan ejecutar varios procesos en este nodo de prueba más conocido como nodo físico o pnodo y que cada proceso provea un entorno de ejecución del mismo de manera separada, a esto se le conoce como nodo virtual o vnodo. (11)

### **1.2 Análisis del comportamiento de las funcionalidades a realizar por el NVP en otros sistemas similares.**

En la actualidad existen muchas aplicaciones que utilizan los nodos virtuales por las grandes ventajas que brindan en el aprovechamiento máximo de los recursos de las computadoras.

Para la simulación de redes, en la Universidad de Stuttgart en Alemania, instituto especializado en sistemas distribuidos y paralelos, se creó la aplicación “Network Emulation Testbed” (6). Esta aplicación va dirigida a simular un entorno de redes configurable que permita reproducir un escenario real de cientos de nodos en comunicación. De esta manera posibilita medir de manera comparativa el comportamiento de una aplicación en diferentes entornos de redes o de varias aplicaciones en un mismo entorno de red. Esta aplicación es utilizada tanto en redes tradicionales como en un entorno de redes ad hoc inalámbrica. Permite a partir de una red de 64 computadoras traducirlo a un escenario de más de 1920 nodos.

La principal limitante de esta aplicación está en que es diseñada para simular redes por lo que no es posible utilizarla para la simulación de procesos que tienen una dinámica un tanto más compleja y variada.

En la Universidad de Nottingham Trent, se creó un software para la simulación de sistemas de agua (7). Surge por la necesidad que tenían en la industria del agua de adquirir y almacenar datos de estaciones remotas para la inspección ingenieril. A medida que fue creciendo el sistema, fue además incrementándose la acumulación de grandes volúmenes de datos que debían ser procesados por los ingenieros. Para ayudar a reducir la carga de trabajo y aumentar la eficiencia y la efectividad del control del sistema se introdujo el software de simulación. Este software tiene como tareas analizar los parámetros del sistema y arribar a decisiones que pueden ser aceptadas o modificadas por el ingeniero responsable de estas operaciones. La integración del sistema de supervisión con el software de

simulación, para lograr un sistema capaz de tomar decisiones en tiempo real, conllevó a un incremento de los requerimientos computacionales para los algoritmos de simulación con un respectivo aumento de tamaño de la red física. Una solución clásica fue dividir la red en subredes para resolver las subredes de manera aislada y de esta manera coordinar las soluciones de los subsistemas para encontrar la solución del sistema completo.

Esta aplicación no es la solución pues no está concebida para simular procesos independientes, sino que simula subprocesos de un sistema específico, para luego integrarlo en el proceso general como un todo.

En el laboratorio Nacional de Lawrence Berkeley en EUA, se enfrentaban al problema de tratar las condiciones límites en los pozos de petróleo en el momento de formular y codificar un simulador numérico multifase de la reserva. Esto se debe a la complejidad de las ecuaciones diferenciales que gobiernan el flujo de la superficie que son una mezcla de tipo hiperbólica-parabólica que provocan problemas de convergencia computacional (8). El método convencional de tratamiento de pozos geotermales o de reservas de petróleo no es lo suficientemente riguroso y puede dar lugar a soluciones físicamente incorrectas para las diferentes capas del pozo. Por esta razón se utilizó el método de nodos virtuales donde a cada capa se le asignó un nodo virtual cuyo tratamiento está dado en dependencia de las características de la capa a la que represente para los cálculos del flujo. La solución en el pozo se obtendrá de resolver las ecuaciones del balance de masa para el nodo del pozo. De esta manera se provee de un procedimiento numérico eficiente y consistente de manera física para el manejo de los problemas del flujo en los pozos.

De la misma manera que el anterior este simulador fue diseñado para el estudio de las diferentes capas de un pozo petrolífero para luego integrarlo en el modelo del pozo.

En la Universidad de Stanford se creó un algoritmo de nodos virtuales para el trabajo con imágenes en tres dimensiones (9). Un elemento es fragmentado para crear varias réplicas del elemento y se le asigna una porción real del material a cada réplica. De esto resultan elementos que contienen material real y regiones vacías. El material faltante está contenido en otra u otras copias. El algoritmo de nodo virtual determina automáticamente el número de réplicas así como la asignación de material para cada una. Provee además los grados de libertad requeridos para simular el material parcial o completamente fragmentado en una imagen consistente con la geometría. Aprovecha las posibilidades de la simulación de una geometría compleja con una simple mezcla.

Este simulador tampoco satisface las necesidades planteadas.

En Japón se creó una herramienta “StarBED” para dar solución al vacío que existe entre internet y los entornos para experimentación, atendiendo a los aspectos de escala, complejidad y realidad (10). Es una aplicación de pruebas basada en lotes de nodos que tiene como objetivos construir entornos de experimentaciones reales, complejos y de gran escala. Está diseñado para 512 computadoras en las cuales se ejecutan diez computadoras virtuales. Es capaz de soportar una topología de experimentación por encima de los 512 pero resulta difícil manejar y controlar todos esos nodos. Además se implementó “SpringOS” para soportar las simulaciones de acuerdo con la configuración de los usuarios. Ya ha sido probada con efectividad en diferentes experimentos como son:

- Test de rendimiento de switches L2 con tráfico multiemitido.
- Observación del comportamiento del TCP.
- Comparación entre el comportamiento entre routers de hardware manufacturado por vendedores populares y routers de software en código abierto ejecutándose en una PC.
- Simulación de redes inalámbricas en redes cableadas.
- Simulación de redes de teléfonos celulares G4.

Estas herramientas fueron creadas para evitar las influencias de los servicios críticos de internet en las aplicaciones de simulación. Se utilizan para la simulación de diferentes tipologías de redes y no están concebidas para la simulación de procesos. Por esta razón no satisface los requerimientos de la aplicación planteada.

De manera general se puede concluir que en la actualidad los nodos virtuales son potencialmente usados para trabajar en aplicaciones diseñadas para la simulación de redes, para la simulación de procesos que por su complejidad no pueden ser realizadas por las herramientas tradicionales y para el tratamiento de imágenes. Por estas razones es necesario que el Nodo Virtual de Procesos incorpore muchas características que no poseen dichas aplicaciones. (11)

### 1.3 Controladores.

Un controlador en un NVP se puede entender como un operador modelado por ecuaciones matemáticas encargadas de controlar<sup>3</sup> (13) los valores de entrada que van a ser usados durante la simulación de un proceso en función de la salida deseada. (Ver Anexo 1 y 2). Si un controlador es configurado correctamente los resultados de la simulación tienen mayor aproximación a la realidad. En resumen, los controladores tienen el objetivo de:

- Operar el proceso en forma segura y estable.
- Evitar cambios considerables y rápidos en variables manipuladas que podrían incumplir restricciones de operación.
- Operar el proceso en forma congruente con los objetivos de calidad de cada producto; además de controlar las cualidades del producto en valores que maximicen su utilidad cuando se consideren índices y valores de productos y además, minimicen el consumo de energía.

Entre los controladores que se usan mundialmente en el sector industrial, se encuentran el PID y el Fuzzy:

**PID:** Si bien existen muchos tipos de control, el control proporcional, derivativo e integral (PID), es el que mayor implantación tiene en la industria de procesos. Dicho control consiste esencialmente en obtener la acción de control como la suma de tres términos: término proporcional, término derivativo y término integral. Se pueden obtener variaciones a este esquema consistentes en la no introducción de los términos derivativo e integral; en dicho caso el control coincide con el llamado control proporcional. Si solamente se prescinde de la acción integral, el control se denomina control proporcional derivativo (PD). Si, por el contrario, el único término que desaparece es el derivativo, el control se denomina proporcional integral (PI). La amplia implantación del control PID en la industria, se debe fundamentalmente a los siguientes factores:

- El término derivativo proporciona cierta anticipación sobre la respuesta al sistema.
- El término integral permite eliminar el error en régimen permanente.
- El control PID obtiene resultados satisfactorios para una amplia gama de procesos.

---

3 “Controlar” significa mantener la medición dentro de límites aceptables.

- Existen sencillas reglas heurísticas que permiten obtener los parámetros del controlador PID. Dichas reglas hacen posible el ajuste del controlador, sin presuponer un gran conocimiento en teoría de control automático por parte del operador.
- El controlador PID se puede adquirir como un módulo compacto, donde los distintos parámetros del controlador se pueden ajustar manualmente. Actualmente muchos de los PIDs industriales proporcionan ciertas opciones de autosintonía.

El controlador PID, si bien tiene una amplia implantación en la industria, no es utilizado convenientemente en muchas ocasiones. Generalmente las causas de mal funcionamiento se deben al ajuste inadecuado de los parámetros del controlador. Un número elevado de los PIDs en la industria han sido sintonizados manualmente, sin la realización de un estudio previo de las características del proceso a controlar. Este tipo de sintonización “manual”, puede proporcionar buenos resultados en función de la experiencia del operador, sobre todo si el control es PI. En caso de requerir el ajuste de tres o más parámetros del controlador, la obtención de un ajuste manual satisfactorio puede resultar una tarea extremadamente difícil que requiere de una gran experiencia por parte del operador. En algunas ocasiones, la falta de conocimientos y experiencia sobre el control PID, se traduce en que el controlador PID se implante con los parámetros originales de fábrica. (14)

**FUZZY:** El control difuso es la primera aplicación exitosa industrial de la lógica difusa (15). Permite mediante el conocimiento experto de una o varias personas, generar una base de conocimientos que dará al sistema la capacidad de tomar decisiones sobre ciertas acciones que se presenten en su funcionamiento. El control difuso puede aplicarse en innumerables sistemas, en los cuales los modelos matemáticos son muy complejos, así, empleando técnicas de razonamiento aproximado es posible controlar sistemas superiores cuando el entorno no se conoce de forma precisa. Dicha característica permite mayor flexibilidad que el control básico, en el que para la realización de un controlador se requiere de un alto grado de cálculo matemático. Así, al desarrollar un controlador difuso es posible prescindir de la rigidez matemática y transmitir el conocimiento de un experto humano hacia un sistema.

Este controlador consta de tres bloques, en el primer bloque llamado fuzzificador, los datos de entrada son calculados; posteriormente se tiene el dispositivo de inferencia que junto con la base de conocimientos realiza la toma de decisiones que dictará la forma en que actuará el sistema; la última etapa que se tiene dentro del controlador es el defuzzificador, quien realiza el procesado final adecuando los valores difusos

obtenidos de la inferencia en valores no difusos útiles para el proceso que se ha de controlar (16). (Ver Anexo 3).

El control difuso tiene tres características principales: (17)

- Es un control lógico. El calificativo de lógico significa que el algoritmo de control usa expresiones IF-THEN, en las que se puede describir una amplia variedad de condiciones, combinando expresiones lógicas con IF y AND.
- Es un control disperso. Esta característica, que diferencia esencialmente los sistemas de control difuso de los sistemas de control basados en una sola ecuación, permite la coexistencia de controladores con lógicas distintas y su ejecución en paralelo.
- Es un control lingüístico. En efecto permite el uso de variables lingüísticas imprecisas, en particular en los antecedentes de las reglas. Este lenguaje cualitativo es fácil de entender, permite realizar el control mediante un diálogo con los operadores, utilizando sus ojos experimentados en la observación del proceso como entradas externas, e introducir cosas como las condiciones del proceso como información útil para el control.
- Además el algoritmo de control puede incluir los procedimientos inusuales que acompañan siempre las operaciones de un proceso real.

### 1.3.1 Protocolos de comunicación asociados.

Los protocolos son reglas de comunicación que permiten el flujo de información entre computadoras distintas que manejan lenguajes distintos, por ejemplo, dos computadores conectados en la misma red pero con protocolos diferentes no podrían comunicarse jamás, para ello, es necesario que ambas "hablen" el mismo idioma (18). En un NVP ellos van a permitir la comunicación con los controladores físicos para probar las aplicaciones en tiempo real.

**OPC:** El OPC (*OLE for Process Control*) es un estándar de comunicación en el campo del control y supervisión de procesos. Este estándar permite que diferentes fuentes de datos envíen datos a un mismo servidor OPC, al que a su vez podrán conectarse diferentes programas compatibles con dicho estándar.

De este modo se elimina la necesidad de que todos los programas cuenten con drivers<sup>4</sup> (19) para dialogar con múltiples fuentes de datos, basta que tengan un driver OPC.

En realidad OPC es un conjunto de protocolos entre los que se pueden destacar los siguientes:

- OPC-DA (Data Access): El original, sirve para el intercambio de datos a tiempo real entre servidores y clientes.
- OPC-AE (Alarms & Events): Proporciona alarmas y notificaciones de eventos.
- OPC B (Batch): Útil en procesos discontinuos.
- OPC DX (Data eXchange): Proporciona interoperabilidad entre varios servidores.
- OPC HDA (Historical Data Access): Acceso histórico a datos OPC.
- OPC S (Security).- Especifica cómo controlar el acceso de los clientes a los servidores.
- OPC XML-DA (XML Data Access): Sirve para el intercambio de datos entre servidores y clientes como OPC-DA pero en vez de utilizar tecnología COM/DCOM utiliza mensajes SOAP (sobre HTTP) con documentos en XML.
- OPC CD (Complex Data): Permite a los servidores exponer y describir tipos de datos más complicados en forma de estructuras binarias y documentos XML. (20)

**TCP/IP:** Los protocolos TCP/IP son fundamentales para el desarrollo de internet tal como hoy se conoce. Su misión es complementaria y tiene como objetivo el que la información llegue a su destino de la manera más eficiente posible.

Suponiendo que un mensaje es un puzzle, el protocolo TCP es el encargado de desmontar cada una de las piezas y memorizar el orden para poder reconstruirlo, cada una de las piezas pueden viajar incluso por caminos diferentes, sin embargo al llegar a su destino el mismo protocolo TCP será el responsable de hacer coincidir otra vez el rompecabezas, incluso si detecta que por el camino alguna pieza se ha estropeado, es capaz de volver a pedir un recambio original para reconstruir la información.

---

4 Un driver o controlador posibilita que el sistema operativo de una computadora pueda entenderse con un equipamiento periférico.

Por otra parte, el protocolo IP es el encargado de hacer llegar a su destino cada una de las piezas, él memoriza de dónde vienen y cuál es su período de caducidad. El trabajo conjunto de los dos protocolos hace que la información llegue al ordenador desde cualquier parte del mundo y en muy poco tiempo, o no. (21)

Además, el protocolo TCP/IP de Ethernet, apto para Internet, es ideal para acoger la proliferación de dispositivos industriales inteligentes de bajo nivel, que cada día exigen más de la anchura de banda disponible. (22)

**DCOM:** Microsoft Distributed COM (DCOM) extiende COM (Component Object Model) para soportar comunicación entre objetos en ordenadores distintos, en una LAN, WAN, o incluso en Internet. DCOM maneja detalles muy bajos de protocolos de red, por lo que uno se puede centrar en la realidad de los negocios: proporcionar soluciones a clientes. Actualmente DCOM viene con los sistemas operativos Windows 2000, NT, 98 y también está disponible una versión para Windows 95 en la página de Microsoft. También hay una implementación de DCOM para Apple Macintosh y se está trabajando en implementaciones para plataformas UNIX como Solaris. Éste define cómo los componentes y sus clientes interactúan entre sí. Esta interacción es definida de tal manera que el cliente y el componente se pueden conectar sin la necesidad de un sistema intermedio. El cliente llama a los métodos del componente sin tener que preocuparse de niveles más complejos.

Cuando el cliente y el componente residen en distintas máquinas, DCOM simplemente reemplaza la comunicación entre procesos locales por un protocolo de red. Ni el cliente ni el componente se enteran de que la unión que los conecta es ahora un poco más grande.

DCOM es completamente independiente del lenguaje. Virtualmente cualquier lenguaje puede ser utilizado para crear componentes COM, y estos componentes pueden ser utilizados por muchos más lenguajes y herramientas. Java, Microsoft Visual C++, Microsoft Visual Basic, Delphi, PowerBuilder, y Micro Focus COBOL interactúan perfectamente con DCOM.

DCOM puede utilizar cualquier protocolo de transporte, como TCP/IP, UDP, IPX/SPX y NetBIOS. DCOM proporciona un marco de seguridad a todos estos protocolos.

Los desarrolladores pueden simplemente utilizar las características proporcionadas por DCOM y asegurar que sus aplicaciones son completamente independientes del protocolo. (23)

**DDE:** El protocolo de intercambio dinámico de datos DDE (Dynamic Data Exchange) es uno de los métodos de comunicación entre procesos que permite intercambiar datos entre aplicaciones de Windows. El protocolo DDE está basado en el sistema de mensajería construido por Windows (24). Así, dos programas de aplicación bajo Windows realizan una “conversación DDE” enviándose mensajes entre ellos. Estos dos programas se conocen como el servidor y el cliente. Un servidor DDE es el programa que tiene acceso a los datos que pueden ser útiles a otros programas. El cliente DDE es el programa que obtiene estos datos desde el servidor.

En comunicaciones de TCP/IP, las aplicaciones abren una línea de comunicación y entonces transfieren datos crudos. DDE trabaja a un nivel más alto, donde las aplicaciones envían mensajes para intercambiar información. Un mensaje simple es enviar un comando a otra aplicación. La mayoría de los mensajes se transfieren como datos, donde los datos son referenciados por nombre.

#### **1.4 Metodologías de desarrollo de software.**

El término Metodología se define como un conjunto de métodos eficientes orientados a conseguir un objetivo propuesto. Son un conjunto de procesos que organizados dan una secuencia de pasos a seguir para obtener los hitos propuestos y finalmente el producto final. (25)

Las metodologías de desarrollo de software constituyen un factor importante para lograr productos con calidad en el tiempo requerido. Actualmente existen múltiples metodologías, todas con muchas características comunes, pero que comparten también significativas diferencias, por lo que se recomienda hacer un estudio de éstas para realizar una correcta selección, garantizando así el éxito del proyecto de software.

Las ventajas de tener una metodología son: (26)

Mejora de los procesos de desarrollo.

- Todas las personas del proyecto trabajan bajo un marco común.
- Estandarización de conceptos, actividades y nomenclaturas.
- Actividades de desarrollo apoyadas por procedimientos y guías.
- Resultados del desarrollo predecibles.

- Uso de herramientas de Ingeniería Software.
- Planificación de actividades en base a un conjunto de tareas definidas y a la experiencia en otros proyectos.
- Recopilación de mejores prácticas para proyectos futuros.

Mejora de los productos.

- Se asegura que los productos cumplen con los objetivos de calidad propuestos.
- Detección temprana de errores.
- Se garantiza la trazabilidad de los productos a lo largo del desarrollo.

Mejora de las relaciones con el cliente.

- El cliente percibe el orden en los procesos.
- Facilita al cliente el seguimiento de la evolución del Proyecto.
- Se establecen mecanismos para asegurar que los productos desarrollados cumplen con las expectativas del cliente.

A continuación se realizará un pequeño estudio de posibles metodologías a usar en el desarrollo del sistema a construir.

#### 1.4.1 Programación Extrema.

La Programación Extrema (XP), es una de las metodologías de desarrollo de software con más éxito en la actualidad. Es una metodología ligera de desarrollo de software que se basa en la simplicidad, la comunicación y la realimentación o reutilización del código desarrollado. Se utiliza en proyectos con equipos de desarrollo pequeños y con plazos de entrega corto. La metodología consiste en una programación rápida o extrema. Una particularidad es tener como miembro del equipo al usuario final.

Esta metodología tiene las siguientes características:

- **Pruebas Unitarias:** Se basa en las pruebas realizadas a los principales procesos, de tal manera que se adelante en algo hacia el futuro, se puede hacer pruebas de las fallas que pudieran ocurrir. Es como si se adelantara a obtener los posibles errores.

- **Refabricación:** Se basa en la reutilización de código, para lo cual se crean patrones o modelos estándares, siendo más flexible al cambio.
- **Programación en pares:** Una particularidad de esta metodología es que propone la programación en pares, la cual consiste en que dos desarrolladores participen en un proyecto en una misma estación de trabajo. Cada miembro lleva a cabo la acción que el otro no está haciendo en ese momento.

Lo fundamental de XP es:

- **La comunicación:** Entre los usuarios y los desarrolladores.
- **La simplicidad:** Al desarrollar y codificar los módulos del sistema.
- **La retroalimentación:** Concreta y frecuente del equipo de desarrollo, el cliente y los usuarios finales.

Fases de XP: (Ver Anexo 4)

El desarrollo bajo XP tiene características que lo distinguen claramente de otras metodologías:

- Los diseñadores y programadores se comunican efectivamente con el cliente y entre ellos mismos.
- Los diseños del software se mantienen sencillos y libres de complejidad o pretensiones excesivas.
- Se obtiene retroalimentación de usuarios y clientes desde el primer día gracias a las baterías de pruebas.
- El software es liberado en entregas frecuentes tan pronto como sea posible.
- Los cambios se implementan rápidamente tal y como fueron sugeridos.
- Las metas en características, tiempos y costos son reajustadas permanentemente en función del avance real obtenido.

Cuándo usar XP: (27)

- Cuando los clientes no tienen idea clara de los requerimientos y los van cambiando.
- Para proyectos de riesgo: fecha fija de entrega, algo nunca hecho por el grupo, algo nunca hecho por la comunidad de desarrolladores.

- Entre dos y diez programadores. No es apto para proyectos con mucho personal.
- Integra gerentes y clientes a la formulación de preguntas, la negociación de cronograma y alcances, la creación de las pruebas.
- Automatiza las pruebas; es posible en casi todos los dominios. Es lícito repensar el diseño para facilitar el ensayo.
- El objetivo es entregar el software tal cual se necesita y en el momento en que se necesita. Incidentalmente, los proyectos XP muestran mayor productividad.

Esta metodología ha sido diseñada para solucionar el eterno problema del desarrollo de software por encargo: entregar el resultado que el cliente necesita a tiempo.

#### **1.4.2 SCRUM.**

El Scrum es un proceso de desarrollo iterativo e incremental enfocado a la gestión de procesos de desarrollo de software, aunque también puede ser utilizado en equipos de mantenimiento de software, que sirve para administrar y controlar el desarrollo de sistemas.

Esta metodología, inicialmente documentada por dos japoneses (Takeuchi y Nonaka) en 1986 y con aportes de especialistas (Sutherland y Schwaber) a lo largo de la última década, tiende a hacer que los procesos que influyen en el desarrollo de tecnología se incrementen en rapidez y flexibilidad, siempre tomando en cuenta los tiempos y reglas de negocio que dan motivo al desarrollo del proyecto. Se emplea en entornos que trabajan con requisitos inestables y que requieren rapidez y flexibilidad; situaciones frecuentes en el desarrollo de determinados sistemas de software, usualmente con equipos chicos y multidisciplinarios.

¿De qué se trata esta metodología?

Cada iteración termina con una pieza de software ejecutable que incorpora nueva funcionalidad. Estas iteraciones tienen en general una duración entre dos y cuatro semanas.

Scrum se focaliza en priorizar el trabajo en función del valor que tenga para el negocio, maximizando la utilidad de lo que se construye y el retorno de inversión. Está diseñado especialmente para adaptarse a los cambios en los requerimientos. Los requerimientos y las prioridades se revisan y ajustan durante el proyecto en intervalos muy cortos y regulares. De esta forma se puede adaptar en tiempo real el producto

que se está construyendo a las necesidades del cliente o, dicho de otra manera, incorporar cambios con rapidez y en cualquier fase del proyecto.

En Scrum se construye primero la funcionalidad que resulte en el mayor beneficio para el negocio y se evita en todo momento desarrollar cosas que no serán de utilidad para el cliente. De esta manera se puede desarrollar un sistema en menos tiempo evitando el trabajo innecesario. (28)

Actores de SCRUM: (29)

- Product Owner
- Scrum Master
- Scrum Team
- Usuarios o Clientes

Las acciones fundamentales de SCRUM son:

- Product Backlog
- Sprint Backlog
- Daily Scrum Meeting

Una ventaja de esta metodología es que es muy fácil de aprender y requiere muy poco esfuerzo para comenzar a utilizarse. Scrum se utiliza como marco para otras prácticas de Ingeniería de Software como RUP o Extreme Programming.

### **1.4.3 Proceso Unificado de Desarrollo (RUP).**

El Proceso Unificado de Desarrollo (RUP), propuesto por Jacobson, Booch y Rumbaugh se publicó en 1999.

Es un proceso de desarrollo de software, definido como un conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema software (30). En el curso 2007-2008 luego de un estudio realizado a tres metodologías de desarrollo (RUP, XP y SCRUM) se decidió utilizar RUP (12). A continuación se exponen las razones por las que fue escogida esta metodología.

RUP identifica claramente a los profesionales (actores) involucrados en el desarrollo del software y sus responsabilidades en cada una de las actividades. Además, explícitamente indica qué actor es responsable de qué artefacto en cada actividad.

RUP es una metodología de desarrollo pensada para proyectos, el NVP no es un sistema de compleja construcción, pero requiere que su realización sea a largo plazo y con un equipo de desarrollo numeroso. Una particularidad de esta metodología es que, en cada ciclo de iteración, se hace exigente el uso de artefactos, siendo por este motivo, una de las metodologías más importantes para alcanzar un grado de certificación en el desarrollo del software. Durante el proceso de desarrollo del NVP no se va a interactuar con el cliente por lo que dichos artefactos formarán parte del producto final y serán la clave para obtener un software robusto.

### **1.5 Herramientas de modelado.**

En el mundo informático se han creado varias herramientas para el desarrollo de la Ingeniería de Software con el objetivo de desarrollar programas. Las herramientas CASE (Computer Aided Software Engineering), Ingeniería de Software Asistida por Ordenador son un conjunto de ayudas para el desarrollo de programas informáticos, desde la planificación pasando por el análisis y diseño, hasta la generación del código fuente de los programas y la documentación (31).

Es conocido que desde los años 70 se comenzaron los estudios para la creación de una herramienta para crear proyectos. Fue en 1984 que aparece la primera herramienta CASE con el nombre de Excelerator y, es entonces que en la década de los 90 que comienza a tener auge entre la comunidad.

Las herramientas CASE tienen como objetivo primordial:

- Mejorar la productividad en el desarrollo y mantenimiento del software.
- Aumentar la calidad del software.
- Mejorar el tiempo y coste de desarrollo y mantenimiento de los sistemas informáticos.
- Mejorar la planificación de un proyecto.
- Aumentar la biblioteca de conocimiento informático de una empresa ayudando a la búsqueda de soluciones para los requisitos.

- Automatizar, desarrollo del software, documentación, generación de código, pruebas de errores y gestión del proyecto.
- Ayuda a la reutilización del software, portabilidad y estandarización de la documentación.
- Gestión global en todas las fases de desarrollo de software con una misma herramienta.
- Facilitar el uso de las distintas metodologías propias de la Ingeniería de Software.

Las herramientas CASE-UML se basan en las fases del ciclo de desarrollo que cubren y se clasifican en:

- Upper Case: planificación y requerimientos.
- Middle Case: análisis y diseño.
- Lower Case: generación de código, pruebas e implementación. (31)

Entre las herramientas CASE orientadas a UML se pueden encontrar: ArgoUML, Rational Rose y Visual Paradigm.

### 1.5.1 ArgoUML.

ArgoUML surge a partir de un proyecto de Jason Robbins en 1999. Es una aplicación de diagramado de UML escrita en Java y publicada bajo la Licencia BSD (Berkeley Software Distribution) open source. Dado que es una aplicación Java, está disponible en cualquier plataforma soportada por Java.

Sin embargo, desde la versión 0.20, ArgoUML está incompleto. No es conforme completamente a los estándares UML y carece de soporte completo para algunos tipos de diagramas incluyendo los diagramas de secuencia y los de colaboración. (32)

#### **Características (33)**

- Independiente de plataforma.
- Basado en los estándares abiertos: XMI, SVG y PGML.
- Compatibilidad con AndroMDA<sup>5</sup>. (34)

---

<sup>5</sup> AndroMDA (pronunciado "Andrómeda") es un programa informático de tipo marco de trabajo, de generación extensible de código, que se adhiere al paradigma de la arquitectura dirigida por modelos.

- Soporte cognitivo.
- Herramienta dirigida al apoyo de fases de planificación, análisis de requisitos y estrategia de desarrollo apoyándose en diagramas UML.
- La mayoría de las funciones ahora soportan la selección múltiple de los elementos del modelo.
- Arrastrar y soltar desde el árbol de exploración al diagrama y dentro del árbol de exploración.
- Construido en diseños críticos suministra una revisión no obstructiva del diseño y sugerencias para mejoras.
- Interfaz de módulos extensible.
- Soporte de Internacionalización para inglés, alemán, francés, español y ruso.
- Restricciones OCL (Object Constraint Language) para Clases.
- Soporte para el lenguaje de generación de Código: Java, PHP, Python, C++ y Csharp (C#).
- Realiza Ingeniería Inversa.
- Disposición automática del diagrama de clases.
- Generación de ficheros PNG, GIF, JPG, SVG, EPS desde diagramas.
- Soporte para comentarios para múltiples elementos.
- Todos los diagramas 1.4 están soportados.

**Debilidades (33)**

- No tiene botón "deshacer".
- Los modelos a veces no pueden ser re-abiertos.
- No hay llamadas-reflexivas en los diagramas de secuencia.
- Debes seleccionar una clase para crear un diagrama de secuencia.

### 1.5.2 Rational Rose.

Es una de las más poderosas del modelado visual para el análisis y diseño de sistemas basados en objetos. Se utiliza para modelar un sistema antes de proceder a construirlo. Cubre todo el ciclo de vida de un proyecto.

**Características:** (35)

- 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 Inversa para algunos de los conceptos más comunes de Java 1.5.
- Generación de código Ada, ANSI C ++, C++, CORBA, Java y Visual Basic, con capacidad de sincronización modelo- código configurables.
- Soporte Enterprise Java Beans™ 2.0.
- Capacidad de análisis de calidad de código.
- El Add-In para modelado Web provee visualización, modelado y las herramientas para desarrollar aplicaciones de Web.
- 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.
- Capacidad de crear definiciones de tipo de documento XML (DTD) para el uso en la aplicación.
- Integración con otras herramientas de desarrollo de Rational.
- Capacidad para integrarse con cualquier sistema de control de versiones SCC-compliant, incluyendo a Rational ClearCase.
- Publicación web y generación de informes para optimizar la comunicación dentro del equipo.

### 1.5.3 Visual Paradigm.

Visual Paradigm para UML es una herramienta profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El software de modelado ayuda a una construcción más rápida de aplicaciones de calidad, mejores y a un menor coste, además de permitir el dibujo de todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación, así como una serie de tutoriales con demostraciones interactivas y proyectos (36). Es importante destacar que el Visual Paradigm como herramienta de modelado no es gratuito, pero sí posee una licencia universitaria que se puede utilizar sin fines de lucro.

Visual Paradigm para UML (VP-UML) es amistoso y con alternativa económica junto a la Borland IBM Rational Rose. Además de su Smart Development Environment (SDE) está disponible en varias versiones las cuales se integran en un IDE.

A continuación aparecen diez razones por las cuales una organización debe elegir VP-UML y SDE: (37)

- Permite incorporar dibujos de Visio en cualquier diagrama UML: A diferencia de otras herramientas CASE-UML, VP-UML y SDE prolongan esta limitación permitiendo incorporar dibujos de Visio en cualquier diagrama de UML, pudiendo modelar un dominio específico de hardware, software, redes, componentes, etc.
- Integración perfecta con los principales IDEs: SDE está disponible en varios IDEs incluyendo Microsoft Visual Studio, Borland JBuilder, Eclipse/IBM WSAD, NetBeans/Sun ONE, IntelliJ IDEA y JDeveloper, lo que puede proporcionar un entorno que integra todo el modelo de código desplegando el proceso de desarrollo de software.
- Completa integración con Microsoft Office: VP-UML y SDE soporta la copia de diagramas como objetos OLE, para poder pegar el diagrama a cualquier documento Word, Excel, Power Point. El contenido del diagrama se puede editar directamente, sin embargo no hay que preocuparse de perder el diagrama original de la fuente.
- Herramientas efectivas y costeables: Son las más asequibles con las actuales funciones de las herramientas CASE-UML. Cuestan la octava parte del precio de otras herramientas de modelado

con funcionalidades similares. Sus diversas ediciones se encuentran disponibles y se puede elegir de acuerdo a su presupuesto y necesidades.

- Herramientas UML más fáciles de usar: Ofrece una edición de líneas para diagramas UML. Se puede crear y especificar un modelo de elementos sin cuadro de diálogos. Los recursos centrados en la interfaz ayuda, en las tareas y botones de acceso directo se muestran cuando un elemento del diagrama está seleccionado. Se realizan operaciones como mostrar/ocultar compartimientos, ajuste de tamaño y creación de elementos con un sólo clic del ratón.
- Soporta múltiples plataformas: No importa el Sistema Operativo que esté en uso, VP-UML y SDE se puede ejecutar en equipo y están disponibles en muchas plataformas como Windows, Linux, Mac OS X y Java Desktop.
- Análisis textual para la identificación de clases candidatas: El análisis textual es una técnica útil y práctica para la captura de los requisitos del sistema y la identificación de las clases candidatas. Éstas son las únicas herramientas CASE que apoyan el análisis textual. Sólo se rellena la declaración del problema y las herramientas pueden ayudarle a identificar el sistema de clases con sólo arrastrar y soltar.
- Soporte para las tarjetas CRC: Las tarjetas CRC identifican clases, responsabilidades y colaboraciones en un sistema OO. Son las únicas herramientas de modelado que apoyan las tarjetas CRC.
- Conversión instantánea de código fuente, archivos ejecutables y binarios en modelos: Permite invertir programas de código fuente, archivos ejecutables y binarios y modelos UML de inmediato. Además de idiomas y formatos que incluyen XML, esquema XML, .NET dll o exe, Java de fuente/class/jar, origen de archivo C++ y archivos fuente CORBA IDL.
- Diseño automático del diagrama: Con su diseño automático se pueden ordenar los diagramas desordenados con tan sólo unos pocos clics con el ratón. Se puede elegir el trazado ortogonal, la disposición jerárquica o el árbol del diseño de acuerdo a la naturaleza del diagrama. Cada estilo del diseño puede afinarse con un conjunto de parámetros configurables.

El VP-UML es una herramienta de modelado multiplataforma que ofrece una interfaz amigable, permite la integración con varios IDE de desarrollo, permite Ingeniería Inversa lo que posibilita entregar al cliente un

producto mejor documentado, y por último incluye un panel para el diseño de la interfaz de usuario por lo que no es necesaria la integración con Visio.

## **1.6 Lenguajes de Programación.**

Para el desarrollo del NVP se compararon numerosas alternativas de desarrollo. Desde usar un único lenguaje de alta eficiencia como C++ y basarse en librerías intermedias para lograr portabilidad, hasta lenguajes que no dependieran en absoluto de la máquina de ejecución como puede ser Java. Pero sin perder de vista que para el objetivo fundamental de las operaciones también es necesario altas capacidades de cálculo en punto flotante y una eficiente gestión de la memoria. (38)

Dentro de los lenguajes de programación los más populares hoy en día son Java, C++ y C#. (11)

### **1.6.1 Lenguaje C++.**

Desde sus inicios, C++ intentó ser un lenguaje que incluyera completamente al lenguaje C (quizá el 99% del código escrito en C es válido en C++) pero al mismo tiempo incorpora muchas características sofisticadas no incluidas en aquel, tales como: programación orientada a objetos, excepciones, sobrecarga de operadores, templates o plantillas. (11)

### **1.6.2 Lenguaje Java.**

La sintaxis del lenguaje Java heredó características de C y C++, explícitamente eliminando aquellas que para muchos programadores (según los diseñadores) resultan excesivamente complejas e inseguras. En la actualidad su uso es promovido para el desarrollo de aplicaciones empresariales del lado del servidor, especialmente a través del estándar J2EE, así como en dispositivos móviles (a través del estándar J2ME.)

En realidad, Java hace referencia a un conjunto de tecnologías entre las cuales el lenguaje Java es sólo una de ellas. Por tal motivo muchas veces se habla de la "plataforma Java", la cual es indesligable del lenguaje. (11)

### **1.6.3 Lenguaje C#.**

El lenguaje C# es una versión avanzada de C y C++ y se ha diseñado especialmente para el entorno .NET. Es un nuevo lenguaje orientado a objetos empleado por programadores de todo el mundo para desarrollar aplicaciones que se ejecuten en la plataforma .NET. C# es un paso muy importante en la

evolución de los lenguajes de programación, y es una solución ideal para las aplicaciones de alto nivel. Con C# se puede desarrollar todo tipo de proyectos de aplicaciones cliente/servidor. (11)

C# amplía las capacidades de C, C++, Visual Basic y Java para proporcionar un completo entorno de desarrollo en el que crear aplicaciones. C# mezcla la potencia de C, las capacidades de orientación a objetos de C++ y la interfaz gráfica de Visual Basic. (39)

En la Tesis de Maestría “Herramienta interactiva para la enseñanza en la carrera de Ingeniería Automática: Nodo Virtual de Procesos.” de la Ingeniera Automática Yalice Gámez Batista, a la cual tributa el presente Trabajo de Diploma, se realizó un estudio comparativo sobre los lenguajes antes mencionados en el que se tuvieron en cuenta factores como: portabilidad, capacidades 2d/3d, matemática de precisión compleja, gestión de memoria, velocidad de ejecución, licencia, eficiencia y modularidad; asignando un peso a cada factor obteniéndose como resultado la selección del lenguaje C++ para implementar el NVP.

### **1.7 Análisis.**

El modelado de análisis es independiente del paradigma de ingeniería a usar. Es realmente un conjunto de modelos que muestran las primeras presentaciones técnicas del sistema. Este modelo debe lograr tres objetivos primarios:

- Describir lo que requiere el cliente.
- Establecer una base para la creación de un diseño de software.
- Definir un conjunto de requisitos que se pueda validar una vez que se construye el software. (40)

#### **1.7.1 Principios del Análisis.**

Son conocidos los estudios realizados por investigadores donde alegan haber identificado los problemas del análisis y sus causas, además de las heurísticas para dar solución a los mismos. Cada método de análisis tiene su punto de vista y cada uno de ellos tiene relación con un conjunto de principios operativos:

- Debe representarse y entenderse el dominio de información de un problema.
- Deben definirse las funciones que debe realizar el software.
- Debe representarse el comportamiento del software (como consecuencia de acontecimientos externos).

- Deben dividirse los modelos que representan información, función y comportamiento de manera que se descubran los detalles por capas o jerárquicamente.

El proceso de análisis debería ir desde la información esencial hasta el detalle de la implementación. (40)

Además de los principios operativos de análisis, se sugieren principios directrices, los cuales si se aplican correctamente, se puede desarrollar una excelente especificación del software de forma tal que se proporcione un buen fundamento para el diseño. Los principios directrices se basan en:

- Entender el problema antes de empezar a crear el modelo de análisis.
- Desarrollar prototipos que permitan al usuario entender cómo será la interacción hombre-máquina.
- Registrar el origen y la razón de cada requisito.
- Usar múltiples planteamientos de requisitos.
- Dar prioridad a los requisitos.
- Trabajar para eliminar la ambigüedad.

### **1.7.2 Rol de analista de sistemas.**

El analista de sistemas surge de la necesidad de analizar, identificar y separar en procesos toda la información referente al software que se desee construir o mejorar, es el encargado de proponer soluciones y seleccionar la idea más idónea para el problema en cuestión.

Varias de las actividades en el desempeño del rol del analista de sistemas son:

- Consultores externos para negocios.
- Experto de soporte dentro de un negocio.
- Agente de cambio en situaciones tanto internas como externas.

La labor del analista de sistemas comienza desde el inicio de la creación del software, en la identificación de los procesos del negocio valorando la manera en que funcionan los mismos y examinando las entradas, el procesamiento de datos y la salida de información, con el objetivo de automatizar los procesos identificando las necesidades del cliente.

En cualquier proceso de desarrollo de software es indispensable el analista de sistemas. Su labor es lograr que tanto el cliente como los desarrolladores hablen el mismo idioma. Selecciona la metodología y define la estrategia de captura de requisitos con el propósito de lograr los objetivos que se proponen.

### 1.7.3 Actividades de la Ingeniería de Requisitos.

Varios autores definen diferentes actividades para la realización de la IR. Éstas según Pressman (40) se centran en los siguientes procesos:

- **Elicitación:** Consiste en extraer de varias fuentes y formas, información que necesita el equipo de desarrollo de software para entender el problema en cuestión y saber qué es lo que el sistema necesita. Permite conocer los procesos que se desarrollan en el negocio y comenzar la interacción con los clientes. Este proceso no es sencillo, pues el equipo de trabajo comienza a interactuar en un entorno y terminología desconocidos para él.
- **Análisis de Requisitos y Negociación:** Cuando se tienen los requisitos, se agrupan y organizan por categorías y subconjuntos, se analizan, se examinan buscando problemas en su consistencia, completitud, ambigüedad, compatibilidad con los demás requisitos, y se clasifican según las necesidades del cliente. Muchas veces los clientes proponen requisitos contradictorios. El desarrollador, en este caso el analista, debe llevar a cabo entonces un proceso de negociación con el cliente, donde se clasifiquen los requisitos, se vean los conflictos presentes según su prioridad, los riesgos que puede presentar cada requisito y se realicen estimaciones de esfuerzo para valorar el impacto de cada uno de los requisitos en el plazo de entrega y coste del proyecto.
- **Especificación de Requisitos:** Se describen las restricciones, funcionalidades y características que deberá tener el sistema a desarrollar. La Especificación del Sistema es el producto final sobre los requisitos del sistema obtenido por el ingeniero. Sirve como fundamento para la ingeniería del hardware, ingeniería del software, la ingeniería de bases de datos y la ingeniería humana. Describe la función y características de un sistema de computación y las restricciones que gobiernan su desarrollo. La especificación delimita cada elemento del sistema. La Especificación del Sistema describe la información (datos y control) que entra y sale del sistema. La Especificación de los Requisitos es el documento que permite negociar el acuerdo de lo que el sistema debe hacer y cumplir, y de esta forma evitar retrasos y mayores costos por errores a

solucionar. Es el punto de partida de la estimación de costo, tiempo y esfuerzo del proyecto e indudablemente, influye en la calidad del producto final.

- **Validación de Requisitos:** Se evalúa la calidad del trabajo realizado hasta el momento. Examina las especificaciones con el objetivo de asegurarse de que los requisitos no son ambiguos, inconsistentes y que los errores detectados anteriormente fueron corregidos. Se basa también en demostrar que los requisitos definidos para el sistema constituyen lo que el cliente desea. La validación puede guiarse para descubrir errores. Algo que resulta muy útil es chequear cada requisito con un cuestionario. Esto le da más seguridad al equipo a la hora de revisar cada requisito. La revisión técnica formal es un mecanismo de validación. Se reúne un equipo de clientes, usuarios, desarrolladores, y revisan la documentación en busca de errores, ambigüedades, inconsistencias, redacciones inconclusas.
- **Gestión de Requisitos:** Es un conjunto de actividades paralelas a las anteriores, que se llevan a cabo durante todo el período de desarrollo que le proporcionan al equipo de trabajo la posibilidad de identificar, controlar y seguir los cambios en los requisitos.

#### **1.7.4 Técnicas para el levantamiento de requisitos.**

A través de los años se ha optado por desarrollar algunas técnicas con el objetivo de ayudar a mejorar el levantamiento de información. Estas son manejadas por un equipo de clientes y desarrolladores que trabajan juntos para comprender el problema, proponer soluciones, negociar diferentes perspectivas o puntos de vista y especificar un conjunto básico de requisitos de la solución.

- **Entrevista:** Es usada con frecuencia para acercar al cliente y al desarrollador, pues logra una muy buena comunicación entre ambas partes. Es muy aceptada y ampliamente extendida. Permite obtener información sobre el problema en cuestión y comprender los objetivos para su solución. Para realizarla es necesario seleccionar correctamente a los entrevistados, definir previamente las preguntas, pues la forma de redacción puede influir en las respuestas y por último, analizar los resultados. (41)
- **Cuestionarios:** Consisten en una serie de preguntas a responder que pueden ser abiertas o cerradas. Las abiertas son más reveladoras, pues permiten que los encuestados respondan con su propia terminología y no los limitan en sus respuestas. Este tipo de preguntas son útiles cuando no

se conoce mucho del tema, y por ende no es posible predeterminar posibles respuestas, o cuando se desea profundizar en criterios personales, opiniones, ideas y, por tanto, necesitan de mayor esfuerzo para su elaboración, respuesta y análisis. Las cerradas poseen las posibles respuestas a elegir, requieren menos esfuerzo por parte del encuestado, son más fáciles de preparar para su análisis, requieren de menos tiempo de realización, pero limitan muchas veces al encuestado a la hora de responder, pues a veces su respuesta no se ajusta exactamente con las posibles variantes que se le presentan. (42)

- Tormenta de ideas (Brainstorming): Es una técnica de reuniones que se usa para generar ideas. Su objetivo principal es concebir la mayor cantidad de requisitos para el sistema. Reúne integrantes de varias disciplinas, mientras mayor experiencia y preparación tengan, mayor posibilidad se presenta de generar buenas ideas. En este tipo de reunión no se debe socavar ideas que parezcan locas, ni evaluar o criticar las ideas de los demás, porque puede producir un ambiente hostil que perjudique el dinamismo de la reunión, además de que esa idea puede constituir un gran aporte luego de ser madurada y perfeccionada o relacionada con otras. (43)
- Desarrollo conjunto de aplicaciones (Joint Application Development, JAD): Es una práctica de grupo donde participan analistas, clientes, usuarios, administradores. Se basa en cuatro principios: dinámica en grupo, el uso de ayudas visuales para mejorar la comunicación, mantener un proceso organizado y racional y una filosofía de documentación (44). Esta técnica se realiza durante varios días y favorece la dinámica de grupo, el trabajo en equipo, la mantención de un proceso organizado. En cada encuentro se analizan los problemas y se exploran soluciones que se presentan de manera que se concrete y documente todo lo acordado y concluido. De esta forma se analizan opiniones de varias personas a la vez, a diferencia de la entrevista, donde se ve todo por separado. Esta técnica constituye una alternativa a la entrevista individual que ahorra tiempo al evitar que se analice la opinión de cada cliente por separado. Además, la documentación es revisada por todo el grupo presente. (45)
- Mapas conceptuales: Es una técnica muy eficiente para obtener una idea del negocio, las relaciones que presenta y el vocabulario, incluso puede ayudar a aportar términos al glosario. Consiste en la representación de conceptos sobre el problema donde se muestran sus asociaciones y atributos mediante un grafo, donde los conceptos son los vértices, y las relaciones

son las aristas. Esta técnica es muy usada en la IR, pues es fácil de entender por parte del usuario o cliente, aunque puede llegar a ser ambigua si no se usa correctamente. (46)

- Arqueología de documentos: Se inspecciona la documentación de la empresa con el fin de familiarizarse más con el negocio y poder identificar posibles requisitos. (42)

## **1.8 Diseño.**

En el contexto de la Ingeniería de Software, el diseño se centra en cuatro áreas importantes de interés: datos, arquitectura, interfaces y componentes. El diseño de software es uno de los pilares fundamentales de la Ingeniería de Software. Es la única forma de convertir exactamente los requisitos de un cliente en un producto o sistema de software finalizado (40). Entre las tareas fundamentales del diseño están: producir un diseño de datos, un diseño arquitectónico, un diseño de interfaz y un diseño de componentes.

### **1.8.1 Calidad del diseño.**

La importancia del diseño de software se puede describir con una sola palabra “calidad”. El diseño es el lugar en donde se fomentará la calidad en la Ingeniería de Software (40). Existen tres características fundamentales que se deben tener en cuenta a la hora de realizar un buen diseño.

Características:

- El diseño deberá implementar todos los requisitos explícitos del modelo de análisis, y deberá ajustarse a todos los requisitos implícitos que desea el cliente.
- El diseño deberá ser una guía legible y comprensible para aquellos que generan código y para aquellos que comprueban y consecuentemente, dan soporte al software.
- El diseño deberá proporcionar una imagen completa del software, enfrentándose a los dominios de comportamiento, funcionales y de datos desde una perspectiva de implementación.

Si se tienen en cuenta estas características cuando se realice el diseño se está garantizando que éste tenga calidad. Además existen otros aspectos que se deben tener en cuenta a la hora de diseñar.

Existen varios criterios por los cuales se puede evaluar la calidad del diseño:

- Un diseño deberá presentar una estructura jerárquica que facilite el trabajo con los componentes del software.

- Un diseño deberá ser modular, es decir que deberá dividirse de forma lógica en elementos que realicen funciones específicas.
- Un diseño deberá contener distintas representaciones de datos y procedimientos.
- Un diseño deberá conducir a módulos que tengan características funcionales independientes.
- Deberá conducir a interfaces que reduzcan la complejidad de las conexiones entre los módulos y el entorno externo.
- Deberá producir un diseño mediante un método que pudiera repetir la información obtenida durante el análisis de los requisitos del software.

### 1.8.2 Principios del diseño.

Roger Pressman en su libro: “Ingeniería del Software. Un enfoque práctico” plantea una serie de principios básicos del diseño que hacen posible que el ingeniero de software navegue por el proceso de diseño, y que hay que tener en cuenta a la hora de diseñar ya que ellos garantizan el correcto funcionamiento del sistema.

- En el proceso de diseño no deberá utilizarse “orejeras”<sup>6</sup> (40).
- El diseño deberá poderse rastrear hasta el modelo de análisis.
- El diseño no deberá inventar nada que ya esté inventado.
- El diseño deberá minimizar la distancia intelectual entre el software y el problema como si de la vida real se tratara.
- El diseño deberá presentar uniformidad e integración.
- El diseño deberá estructurarse para admitir cambios.
- El diseño deberá estructurarse para degradarse poco a poco, incluso cuando se enfrenta con datos, sucesos o condiciones de operación aberrantes.
- El diseño no es escribir código y escribir código no es diseñar.

---

6 Orejeras: no escuchar alternativas en función de los requisitos del problema y los recursos disponibles.

- El diseño deberá evaluarse en función de la calidad mientras se va creando, no después de terminado.
- El diseño deberá revisarse para minimizar los errores conceptuales (semánticos).

### **1.9 Sistemas de Gestión de Bases de Datos.**

El software que permite la utilización y/o actualización de los datos almacenados en una o varias bases de datos por uno o varios usuarios desde diferentes puntos de vista y a la vez, se denomina sistema de gestión de bases de datos (SGBD).

El objetivo fundamental de un SGBD consiste en suministrar al usuario las herramientas que le permitan manipular, en términos abstractos, los datos, o sea, de forma que no le sea necesario conocer el modo de almacenamiento de los datos en la computadora, ni el método de acceso empleado.

Los programas de aplicación operan sobre los datos almacenados en la base utilizando las facilidades que brindan los SGBD, los que, en la mayoría de los casos, poseen lenguajes especiales de manipulación de la información que facilitan el trabajo de los usuarios.

#### **1.9.1 MySQL.**

MySQL Database Server es el gestor de base de datos de código fuente abierto más usado del mundo. Su ingeniosa arquitectura lo hace extremadamente rápido y fácil de personalizar.

Su principal objetivo de diseño fue la velocidad, y como otra de sus características importante a considerar es que consume pocos recursos, tanto de CPU como de memoria.

Entre sus principales ventajas se encuentran:

- Gran rendimiento y velocidad tanto al conectar con el servidor como al servir selects<sup>7</sup> (47) y demás.
- Las herramientas de MySQL son potentes y flexibles, sin sacrificar su capacidad de uso. Ofrece utilidades de administración (backup, recuperación de errores, etc.). Resulta fácil de utilizar y de administrar.

---

7 Palabra clave que indica que la sentencia de SQL que queremos ejecutar es de selección.

- Dispone de muchas de las funciones que exigen los desarrolladores profesionales, como compatibilidad completa con ACID, compatibilidad para la mayor parte de SQL, volcados online, duplicación, funciones SSL e integración con la mayor parte de los entornos de programación.
- Se ejecuta en la inmensa mayoría de sistemas operativos y, en la mayor parte de los casos, los datos se pueden transferir de un sistema a otro sin dificultad.
- Aunque se cuelgue, no suele perder información ni corromper los datos.
- No hay límites en el tamaño de los registros.
- Eficiente control de acceso, en el sentido de que usuarios tienen acceso a que tablas y con qué permisos.

MySQL también presenta sus inconvenientes entre los que tenemos:

- No soporta transacciones, "roll-backs" ni subselects.
- No considera las claves ajenas. Ignora la integridad referencial, dejándola en manos del programador de la aplicación.
- Los índices son una desventaja en aquellas tablas las que se utiliza frecuentemente operaciones de escritura (Insert, Delete, Update), esto es porque los índices se actualizan cada vez que se modifica una columna.
- No soporta comprobación de limitación (Check constraint).
- Soporta subconsultas, pero en algunas formas puede ser un gran impacto en el rendimiento.
- Es propiedad y está patrocinado por Sun Microsystems.

### **1.9.2 PostgreSQL.**

PostgreSQL es un gestor de base de datos objeto relacional libre, liberado bajo la licencia BSD. Como muchos otros proyectos open source, el desarrollo de PostgreSQL no es manejado por una sola compañía sino que es dirigido por una comunidad de desarrolladores y organizaciones comerciales las cuales trabajan en su desarrollo, dicha comunidad es denominada el PGDG (PostgreSQL Global Development Group). (48)

PostgreSQL está diseñado para ser un sistema de bases de datos de mayor nivel que MySQL, a la altura de Oracle, Sybase o Interbase.

Entre sus principales ventajas se encuentran:

- Por su arquitectura de diseño, escala muy bien al aumentar el número de CPUs y la cantidad de RAM.
- PostgreSQL puede comprimir y descomprimir sus datos sobre la marcha con un rápido sistema de compresión para encajar más datos en un espacio de disco asignado, lo que permite además ahorrar espacio en disco, por lo que lee datos con mayor rapidez.
- Soporta transacciones y desde la versión 7.0, claves ajenas (con comprobaciones de integridad referencial).
- Tiene mejor soporte para triggers y procedimientos en el servidor.
- Proporciona características que pueden conducir a un rendimiento más rápido en algunas queries: indexación parcial, una mayor asignación de memoria de forma predeterminada en las últimas versiones de sistemas capaces y mejora de la gestión de caché en las versiones 8.1 y 8.2.
- Soporta las “reglas”, que permiten operar en el árbol de sintaxis de la consulta, y puede hacer algunas operaciones más simplificadas que tradicionalmente son realizadas por disparadores.
- No tiene un tipo de dato entero sin signo, pero tiene mucho más soporte de tipos de datos en varios aspectos: el cumplimiento de las normas, la lógica fundamental del tipo de datos boolean, mecanismos de tipo de datos definidos por el usuario, tipos nativos y contribuidos.

PostgreSQL también presenta sus inconvenientes entre los que tenemos:

- Límite del tamaño de cada fila de las tablas de 8k (se puede ampliar a 32k recompilando, pero con un coste añadido en el rendimiento).
- Carece de un conjunto de herramientas que permitan una fácil gestión de los usuarios y de las bases de datos que contenga el sistema.

- La velocidad de respuesta que ofrece con bases de datos relativamente pequeñas puede parecer un poco deficiente, aunque esta misma velocidad la mantiene al gestionar bases de datos realmente grandes.

A pesar de los inconvenientes que presenta PostgreSQL como gestor de base de datos provee ventajas en cuanto al rendimiento, compresión de datos, concurrencia y velocidad que van a permitir un mejor funcionamiento del NVP en tiempo de ejecución. Teniendo en cuenta las funcionalidades previstas para el NVP, va a ser necesario almacenar, modificar, eliminar y recuperar grandes volúmenes de información constantemente y en el menor tiempo posible, así como el uso de la futura aplicación por varios usuarios a la vez.

### **Conclusiones**

En este capítulo se hace un análisis de los principales aspectos y conceptos relacionados con el Nodo Virtual, de cómo las aplicaciones existentes implementan las funcionalidades propuestas para el NVP. También se analizan temas y conceptos de importancia dentro del análisis y diseño de software, pretendiendo dejar sentadas las bases teóricas del trabajo a desarrollar. Se realiza un estudio de las posibles herramientas a utilizar.

Como resultado de este análisis se llegó a las siguientes conclusiones:

- Es necesaria la construcción de un Nodo Virtual de Procesos, ya que esta herramienta será de valor incalculable para que estudiantes desarrollen las habilidades y los conocimientos adquiridos.
- La aplicación a construir tiene objetivos bien definidos, requerimientos estables cuya dinámica se mantendrá a lo largo del proceso de desarrollo de la misma. Se encuentra en proceso de documentación sin elemento alguno de desarrollo, no existe interacción con el cliente y todavía no se define un equipo de trabajo para su realización. Por lo expuesto anteriormente la metodología a utilizar en el desarrollo del proyecto será Rational Unified Process (RUP), metodología robusta que resulta más adaptable para los proyectos a largo plazo que necesiten un desarrollo iterativo capaz de mantener un buen control sobre los cambios.
- Existe la necesidad de desarrollar un producto multiplataforma, teniendo en cuenta que el sistema es para la simulación de procesos (actividad que genera mucha información). Una de las características principales con que debe contar es la velocidad con las que se realicen las

operaciones, lo cual incidirá en la calidad del trabajo del sistema. Debido a los anteriores requerimientos se utilizará PostgreSQL en su versión 8.0 como gestor de base de datos, se empleará como herramienta de modelado Visual Paradigm en su versión 6.1, C++ como lenguaje de programación y el IDE a utilizar será QtCreator, el mismo fue seleccionado luego de un estudio realizado en el Trabajo de Diploma “Implementación de la Solución Informática “Nodo Virtual de Procesos” para la carrera Ingeniería Automática” de los autores Rolando Moreno Martínez y Daniel Hernández Garrigó.

## Capítulo 2: Refinamiento de Requisitos

### Introducción

En el presente capítulo se explicará de forma general el funcionamiento del NVP y algunos aspectos que ayudarán a una mejor comprensión de las características del sistema. Se centrará la atención en la revisión de los requerimientos con el fin de obtener un refinamiento adecuado y la consecuente modificación de los artefactos correspondientes.

#### 2.1 Características del Sistema.

Entre las características principales del NVP se encuentran:

- Permitir la simulación concurrente de procesos industriales.
- Permitir la conexión con dispositivos físicos para probar estrategias de control, mediante la implementación de protocolos de comunicación industriales.
- Permitir la supervisión y control por parte del profesor sobre los procesos que están corriendo simultáneamente y las acciones de los usuarios (alumnos).

Toda la información de los procesos industriales que han sido simulados va a encontrarse tabulada y gráfica para facilitar un mejor entendimiento del mismo. Es importante garantizar una interfaz que facilite la visualización de la información y permita que ésta sea consultada desde cualquier equipo de cómputo. (12)

Se utilizará PostgreSQL como gestor de base de datos que contendrá toda la información necesaria para el funcionamiento del sistema. La aplicación mostrará un registro de variables como resultado de la simulación de un proceso, al que podrán acceder los usuarios que estén conectados a dicho proceso. Además gestionará una serie de reportes que ayudarán al administrador del sistema a controlar el buen funcionamiento éste, entre ellos se encuentra el Registro de un Usuario, que permitirá al administrador conocer las acciones realizadas por los usuarios del sistema.

##### 2.1.1 Clases del Dominio.

A continuación se presenta la descripción de las clases del dominio. (Ver Figura 1) en (12).

**Usuario:** Persona que va a interactuar con el sistema.

**Nodo Virtual:** Servidor de aplicación para la simulación concurrente de procesos industriales.

**Controlador:** Dispositivo físico para regular procesos industriales.

**Tipo de Proceso:** Clasificación en la que se agrupan los procesos.

**Modelo de Proceso:** Conjunto de datos de un proceso definidos por el administrador, como son: modelo matemático, controlador y variables de entrada y salida.

**Proceso:** Modelo de Proceso configurado para realizar la simulación.

**Proceso Activado:** Proceso al que se ha conectado un usuario para su posterior simulación.

**Proceso Inactivo:** Proceso sin usuarios conectados.

El usuario se conecta al Nodo Virtual, servidor donde se encuentran los procesos industriales con los que puede interactuar. Para conectarse a un proceso, el usuario debe seleccionar un proceso perteneciente a un tipo determinado. Los procesos se pueden encontrar de dos formas dentro de la aplicación: activos e inactivos; el estado estará visible en forma tabular para el usuario.

Si el usuario decide conectarse a un proceso activo, el sistema, luego de verificar disponibilidad de conexión a ese proceso, le mostrará una interfaz con el resultado de la simulación del mismo. Si el usuario decide conectarse a un proceso inactivo, el sistema le da la posibilidad de convertirse en usuario maestro de ese proceso, permitiéndole activarlo e interactuar completamente con el mismo.

Para activar el proceso primeramente el usuario maestro debe de configurarlo. El sistema inicialmente mostrará los datos del modelo de ese proceso. El usuario debe introducir los valores de las variables de entrada, además de seleccionar y configurar un controlador. Después de realizada la configuración el usuario maestro puede pulsar la opción Simular.

Si el usuario maestro al configurar el proceso seleccionó un controlador físico la simulación se convierte en una prueba del proceso en tiempo real.

El usuario maestro puede detener la simulación para graficar o imprimir el registro de variables obtenido hasta ese instante, además el sistema le va a permitir reconfigurar el proceso. Cuando el usuario lo desee, puede terminar la simulación.

De esta forma quedarán automatizados los servicios de simulación de procesos que van ayudar a mejorar el sistema de aprendizaje de los estudiantes de la carrera de Ingeniería Automática.

Partiendo de la descripción anterior, el modelo de dominio se estructuró de la siguiente forma:

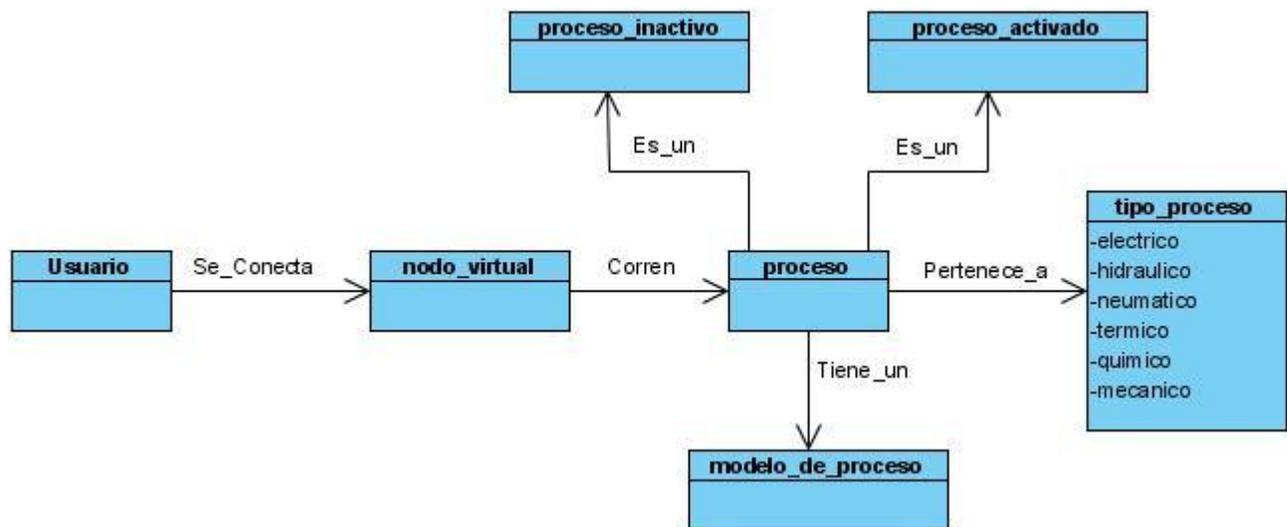


Figura 1: Modelo de Dominio

## 2.2 Redefinición de requerimientos.

En el curso 2007-2008 en el Trabajo de Diploma “Análisis y Diseño de un Nodo Virtual de Procesos” (12) se realizó un levantamiento de requisitos para la construcción de la herramienta NVP. En el presente Trabajo de Diploma se revisaron dichos requisitos debido al surgimiento de ambigüedades y la omisión de algunos aspectos necesarios para el funcionamiento eficiente del sistema. Entre los aspectos omitidos se encuentran: el registro de acciones de los usuarios, las notificaciones generadas durante los diferentes procesos de gestión, la selección del requerimiento de tiempo durante la configuración de procesos y la programación del tiempo de duración de los procesos activados automáticamente. Para implementar todas las funcionalidades previstas, se incluyeron cambios en cuanto al sistema operativo del servidor y el gestor de base de datos, además se definieron los protocolos de comunicación entre los Dispositivos Físicos y el Nodo Virtual y se propuso la utilización de los recursos hardware de la PC cliente para la graficación.

Para llevar a cabo el proceso de revisión se realizaron algunos de los pasos contenidos en las actividades de la Ingeniería de Requisitos. Se comenzó por la Elicitación, debido a la necesidad de volver a estudiar la documentación referente al NVP. Las técnicas utilizadas fueron: la tormenta de ideas entre desarrolladores, para capturar la mayor cantidad de requerimientos, la entrevista con la parte interesada, con el fin de captar información lo más exacta, clara y abarcadora posible; y la arqueología de documento, de forma tal que se lograra un refinamiento de los requisitos capturados hasta el momento. Como parte del Análisis, éstos se agruparon y organizaron por objetivos del sistema; durante la Especificación se describieron las restricciones, funcionalidades y características que tendrá el sistema a desarrollar y se identificaron los datos que entrarán y saldrán del sistema. La calidad del refinamiento realizado en las anteriores actividades se evaluó mediante la Validación, revisándose las especificaciones y construyéndose los prototipos de interfaz correspondientes para asegurarse de que los requisitos no son ambiguos, inconsistentes y que los errores detectados anteriormente fueron corregidos.

### **2.2.1 Requisitos funcionales.**

Teniendo en cuenta que el NVP es un sistema que entre otras funciones se encarga de gestionar información sobre procesos y usuarios, se incluyó el concepto Requisito de Información (RI) que describe qué información debe almacenar el sistema para satisfacer las necesidades de clientes y usuarios, además identifica los conceptos relevantes sobre los que se debe almacenar información y los datos específicos que son de interés. (49)

Los Objetivos del Sistema son:

- Conexión a Sistema.
- Conexión a Proceso.
- Desconectar a un usuario de un proceso.
- Gestionar rol de administrador.
- Establecer límites.
- Gestionar Tipos de Procesos.
- Gestionar Modelos de Procesos.

- Configurar Proceso.
- Simular Proceso.
- Probar Aplicaciones.
- Gestionar Controladores.
- Activar automáticamente uno o varios procesos.
- Priorizar procesos activos por tipo de proceso.
- Gestionar registro de usuarios.
- Gestionar reportes.

(Ver Anexo 5).

Los Requisitos de Información definidos para el sistema se describen a continuación:

<b>RI-1</b>	Información sobre Administradores.
<b>Objetivos asociados</b>	OBJ-4 Gestionar rol administrador
<b>Requisitos Asociados</b>	RF 06. Mostrar listado de los administradores. RF 07. Adicionar administrador. RF 08. Modificar datos de un administrador. RF 09. Eliminar un administrador.
<b>Descripción</b>	El sistema deberá almacenar los datos correspondientes a los administradores en dependencia de la acción realizada.
<b>Datos específicos</b>	<ul style="list-style-type: none"> <li>➤ Datos del listado: Usuario, nombre y apellidos.</li> <li>➤ Datos del administrador:</li> </ul>

	Usuario, nombre, apellidos y contraseña.
--	--

<b>RI-2</b>	Información sobre Tipos de Procesos.
<b>Objetivos asociados</b>	OBJ-6. Gestionar Tipos de Procesos
<b>Requisitos Asociados</b>	<p>RF 13. Mostrar listado de los tipos de procesos existentes.</p> <p>RF 14. Adicionar un tipo de proceso.</p> <p>RF 15. Modificar un tipo de proceso.</p> <p>RF 16. Eliminar un tipo de proceso.</p> <p>RF 17. Notificar resultado de la gestión de tipos de procesos.</p>
<b>Descripción</b>	El sistema deberá almacenar los datos correspondientes a los tipos de procesos en dependencia de la acción seleccionada.
<b>Datos específicos</b>	<ul style="list-style-type: none"> <li>➤ Datos del listado: Nombre y descripción.</li> <li>➤ Datos del tipo de proceso: Nombre y descripción.</li> </ul>

<b>RI-3</b>	Información sobre los Modelos de Procesos.
<b>Objetivos asociados</b>	OBJ-7. Gestionar Modelos de Procesos

<b>Requisitos Asociados</b>	<p>RF 18. Adicionar un modelo de un proceso.</p> <p>RF 19. Mostrar un listado de los modelos de procesos existentes.</p> <p>RF 20. Modificar un modelo de un proceso.</p> <p>RF 21. Eliminar un modelo de un proceso.</p> <p>RF 38. Actualizar registro.</p> <p>RF 22. Notificar resultado de la gestión de modelos de procesos.</p>
<b>Descripción</b>	<p>El sistema deberá almacenar los datos correspondientes a los modelos de procesos en dependencia de la acción seleccionada.</p>
<b>Datos específicos</b>	<ul style="list-style-type: none"> <li>➤ Datos del listado: Nombre, cantidad de usuarios conectados y estado.</li> <li>➤ Datos del modelo de proceso: Nombre, tipo, nombre de variables de entrada, nombre de variables de salida, valores de variables de entrada, valores de variables de salida, modelo matemático.</li> </ul>

<b>RI-4</b>	<p>Información sobre los controladores.</p>
<b>Objetivos asociados</b>	<p>OBJ-11. Gestionar controladores</p>
<b>Requisitos Asociados</b>	<p>RF 28. Mostrar listado de los controladores.</p> <p>RF 29. Adicionar controladores.</p> <p>RF 30. Mostrar datos de controladores.</p>

	RF 31. Eliminar controladores.
<b>Descripción</b>	El sistema deberá almacenar los datos correspondientes a los controladores en dependencia de la acción seleccionada.
<b>Datos específicos</b>	<ul style="list-style-type: none"> <li>➤ Datos del listado: Id de controlador, nombre y descripción.</li> <li>➤ Datos del controlador: -Controlador Predefinido Nombre, tipo, nombre de variables de entrada, nombre de variables de salida, valores de variables de entrada, valores de variables de salida, modelo matemático. -Controlador Físico Nombre, puerto, dirección física, variables de entrada, variables de salida.</li> </ul>

<b>RI-5</b>	Información sobre los usuarios.
<b>Objetivos asociados</b>	OBJ-14. Gestionar registro de usuarios
<b>Requisitos Asociados</b>	<p>RF 37. Crear registro del usuario.</p> <p>RF 38. Actualizar registro del usuario.</p> <p>RF 39. Mostrar registro del usuario.</p> <p>RF 40. Mostrar listado de usuarios existentes.</p>
<b>Descripción</b>	El sistema deberá almacenar los datos correspondientes a los usuarios en dependencia de la acción seleccionada.

<b>Datos específicos</b>	<ul style="list-style-type: none"> <li>➤ Datos del listado: Usuario, nombre y apellidos.</li> <li>➤ Datos del registro acciones del usuario: Fecha y hora de las acciones: conexión y desconexión del sistema, conexión y desconexión de un proceso, gestión de un modelo de proceso, configuración de un proceso, activación automática de procesos.</li> </ul>
--------------------------	--

<b>RI-6</b>	Información sobre los Reportes.
<b>Objetivos asociados</b>	OBJ-15. Gestionar Reportes
<b>Requisitos Asociados</b>	<p>RF 41. Mostrar listado de procesos activos.</p> <p>RF 42. Mostrar listado de usuarios conectados.</p> <p>RF 43. Mostrar listado de usuarios conectados a un proceso.</p> <p>RF 44. Mostrar listado de los procesos activados por un usuario.</p> <p>RF 45. Imprimir informes.</p>
<b>Descripción</b>	El sistema deberá mostrar los datos correspondientes a los reportes en dependencia de la acción seleccionada.
<b>Datos específicos</b>	<ul style="list-style-type: none"> <li>➤ Datos del listado de procesos activos: Nombre, tipo, cantidad de usuarios conectados y estado.</li> <li>➤ Datos del listado de usuarios conectados: Usuario, nombre, apellidos, cantidad de procesos a los que se ha conectado.</li> </ul>

	<ul style="list-style-type: none"> <li>➤ Datos del listado de usuarios conectados a un proceso : Usuario, nombre, apellidos.</li> <li>➤ Datos del listado de los procesos activados por un usuario: Nombre, tipo, cantidad de usuarios conectados y estado.</li> </ul>
--	--

Se tomó como alternativa para reflejar los requisitos funcionales, una tabla que contiene el requisito con su nomenclador de la forma RF- #, la descripción del requisito con todos los datos implicados y el caso de uso con el que se le da seguimiento. Los que se presentan a continuación no están contenidos dentro de los requisitos de información descritos anteriormente.

<b>RF-01</b>	Conectarse al sistema.
<b>Descripción</b>	<p>El sistema deberá definir dos roles: usuario y administrador, que permitirán registrarse y autenticarse respectivamente. Cuando pueda establecerse la conexión, el sistema mostrará un panel personalizado en dependencia del rol y se creará el registro de acciones.</p> <ul style="list-style-type: none"> <li>➤ Datos requeridos del usuario: Nombre, apellidos y usuario.</li> <li>➤ Datos requeridos del administrador: Usuario y contraseña.</li> </ul>
<b>Seguimiento</b>	CUS Gestionar_Conexión_Sistema

<b>RF-02</b>	Desconectarse del sistema.
<b>Descripción</b>	El sistema permitirá la desconexión a petición del usuario. El usuario

	sólo podrá desconectarse del sistema si no se encuentra conectado a ningún proceso. Cuando se realice la desconexión el registro de acciones del usuario es actualizado.
<b>Seguimiento</b>	CUS Gestionar_Conexión_Sistema

<b>RF-03</b>	Conectarse a un proceso.
<b>Descripción</b>	El sistema permitirá al usuario conectarse a un proceso de un tipo determinado y de estado activo o inactivo. Cuando se realice la conexión a un proceso, el registro de acciones del usuario es actualizado. Si el usuario cuando se conecta a un proceso inactivo decide ser el usuario maestro, el sistema le permitirá configurarlo.
<b>Seguimiento</b>	CUS Gestionar_Conexión_Proceso

<b>RF-04</b>	Desconectarse de un proceso.
<b>Descripción</b>	El sistema permitirá el usuario desconectarse de un proceso. Cuando se realice la desconexión se reasigna un nuevo usuario maestro y se le envía una notificación, en caso de que existan otros usuarios conectados a ese proceso o, se desactiva el proceso si no existen otros usuarios conectados a él. El registro de acciones del usuario es actualizado.
<b>Seguimiento</b>	CUS Gestionar_Conexión_Proceso

<b>RF-05</b>	Desconectar usuario de un proceso.
--------------	------------------------------------

<b>Descripción</b>	El sistema permitirá al administrador desconectar a un usuario de un proceso. Cuando se realice la desconexión se reasigna un nuevo usuario maestro y se le envía una notificación, en caso de que existan otros usuarios conectados a ese proceso o, se desactiva el proceso si no existen otros usuarios conectados a él. El registro de acciones del usuario es actualizado. Se le envía una notificación al usuario desconectado.
<b>Seguimiento</b>	CUS Desconectar_Usuario_Proceso

<b>RF-10</b>	Limitar el número de usuarios conectados al sistema.
<b>Descripción</b>	El sistema permitirá al administrador restringir el número de usuarios que pueden estar conectados al sistema, que debe ser mayor o igual a la cantidad de usuarios conectados al mismo en ese instante.
<b>Seguimiento</b>	CUS Establecer_Límites

<b>RF-11</b>	Limitar el número de usuarios conectados por procesos.
<b>Descripción</b>	El sistema permitirá al administrador restringir el número de usuarios de un proceso, que debe ser mayor o igual a la cantidad de usuarios conectados al mismo en ese instante.
<b>Seguimiento</b>	CUS Establecer_Límites

<b>RF-12</b>	Limitar el número de procesos activos.
--------------	--

<b>Descripción</b>	El sistema permitirá al administrador restringir el número de procesos activos, que debe ser mayor o igual a la cantidad de usuarios conectados al sistema en ese instante.
<b>Seguimiento</b>	CUS Establecer_Límites

<b>RF-23</b>	Configurar un modelo de proceso.
<b>Descripción</b>	<p>El sistema permitirá al usuario maestro configurar varios parámetros de un modelo de proceso.</p> <p>➤ Datos del editables del modelo de proceso:</p> <p>Nombre de variables de entrada, valores de variables de entrada, método numérico, controlador, tiempo de simulación.</p> <p>-Controlador Predefinido</p> <p>Nombre de variables de entrada, valores de variables de entrada.</p> <p>-Controlador Físico</p> <p>Nombre, protocolo, dirección física.</p>
<b>Seguimiento</b>	CUS Configurar_Proceso

<b>RF-24</b>	Notificar resultado de la configuración.
<b>Descripción</b>	El sistema envía una notificación al resto de los usuarios conectados al proceso configurado, mostrando el resultado de la configuración.
<b>Seguimiento</b>	CUS Configurar_Proceso

<b>RF-25</b>	Mostrar el resultado final de la simulación.
<b>Descripción</b>	El sistema mostrará los valores de las variables de salida que se obtuvieron durante la simulación.
<b>Seguimiento</b>	CUS Simular_Proceso, Probar_Aplicación

<b>RF-26</b>	Mostrar el resultado de la simulación de forma gráfica.
<b>Descripción</b>	El sistema permitirá al usuario seleccionar las variables de salida a graficar y mostrará la simulación de forma gráfica.
<b>Seguimiento</b>	CUS Graficar

<b>RF-27</b>	Establecer conexión con el controlador.
<b>Descripción</b>	El sistema verificará los parámetros para establecer la conexión entre el Nodo Virtual y el controlador físico.
<b>Seguimiento</b>	CUS Probar_Aplicación

<b>RF-32</b>	Mostrar un listado de los modelos de procesos inactivos por tipo de proceso.
<b>Descripción</b>	El sistema permitirá al administrador ver los modelos de procesos inactivos de un tipo determinado.

	<p>➤ Datos del listado:</p> <p>Nombre, cantidad de usuarios conectados y estado.</p>
<b>Seguimiento</b>	CUS Activar_Automáticamente _Procesos

<b>RF-33</b>	Programar el tiempo de duración de la activación.
<b>Descripción</b>	El sistema permitirá al administrador definir el tiempo en que van a permanecer los procesos activados automáticamente.
<b>Seguimiento</b>	CUS Activar_Automáticamente _Procesos

<b>RF-34</b>	Activar procesos automáticamente.
<b>Descripción</b>	El sistema permitirá al administrador activar varios procesos automáticamente durante un tiempo programado.
<b>Seguimiento</b>	CUS Activar_Automáticamente _Procesos

<b>RF-35</b>	Definir prioridades por tipo de proceso.
<b>Descripción</b>	El sistema permitirá al administrador restringir el número de procesos a activar por tipo.
<b>Seguimiento</b>	CUS Priorizar_Número_Procesos

<b>RF-36</b>	Actualizar configuración del NVP.
<b>Descripción</b>	El sistema actualizará los datos introducidos por el administrador en la Base de Datos.
<b>Seguimiento</b>	CUS Priorizar_Número_Procesos, Establecer_Límites

### 2.2.2 Requisitos no funcionales.

#### Hardware

- Servidor
  1. Discos duros: SCCSI.
  2. Microprocesador: 3 GHz como mínimo.
  3. Memoria RAM: 1Gb.
  4. PC PENTIUM IV o superior.
  5. 3 GB de espacio libre en el disco duro.
- PC clientes
  1. Microprocesador: 2 GHz como mínimo.
  2. Memoria RAM: 512 MB.
  3. PC PENTIUM IV o superior.
  4. Impresora.

#### Software

- Servidor
  1. Sistema Operativo: Ubuntu Server.
  2. Gestor de Base de datos: PostgreSQL.
- PC clientes

1. Sistema Operativo: Windows 2000 o superior, Linux.

### **Seguridad**

1. Cada usuario accederá sólo a la información que le corresponda según su nivel de acceso.
2. Se llevará a cabo tratamiento de excepciones.
3. Programación de disparadores (Triggers) en la base de datos para no permitir la manipulación de los datos directamente en el SGBD.
4. El tipo de tecnología para la configuración que tendrán los discos duros en el servidor será Raid.

### **Redes**

1. La red existente en las instalaciones debe de soportar la tecnología multicast.
2. Las transacciones y recuperación de los datos en la comunicación servidor - PC cliente, se realizara a través del protocolo TCP/ IP.
3. Protocolos de comunicación con controladores: TCP/IP, DCOM, DDE y OPC.

### **Portabilidad**

1. La aplicación deberá correr sobre cualquier Sistema Operativo.

### **Restricciones de diseño e implementación**

1. Uso de plataforma libre.
2. Se utilizará el lenguaje de programación C++.
3. El IDE a usar será QTCreator.
4. Empleo de la filosofía de la Programación Orientada a Objetos (POO).
5. Se implementará la graficación con los recursos hardware de la PC cliente.

### **2.3 Reestructuración del Diagrama de Casos de Uso del Sistema.**

Durante el refinamiento realizado a los requisitos planteados inicialmente para el NVP surgieron nuevas funcionalidades como los reportes Mostrar Registro de Usuarios y Mostrar Usuarios Conectados. Además

se incorporaron nuevas actividades a los casos de uso que ya existían (12), las cuales se describen en la especificación de cada caso de uso.

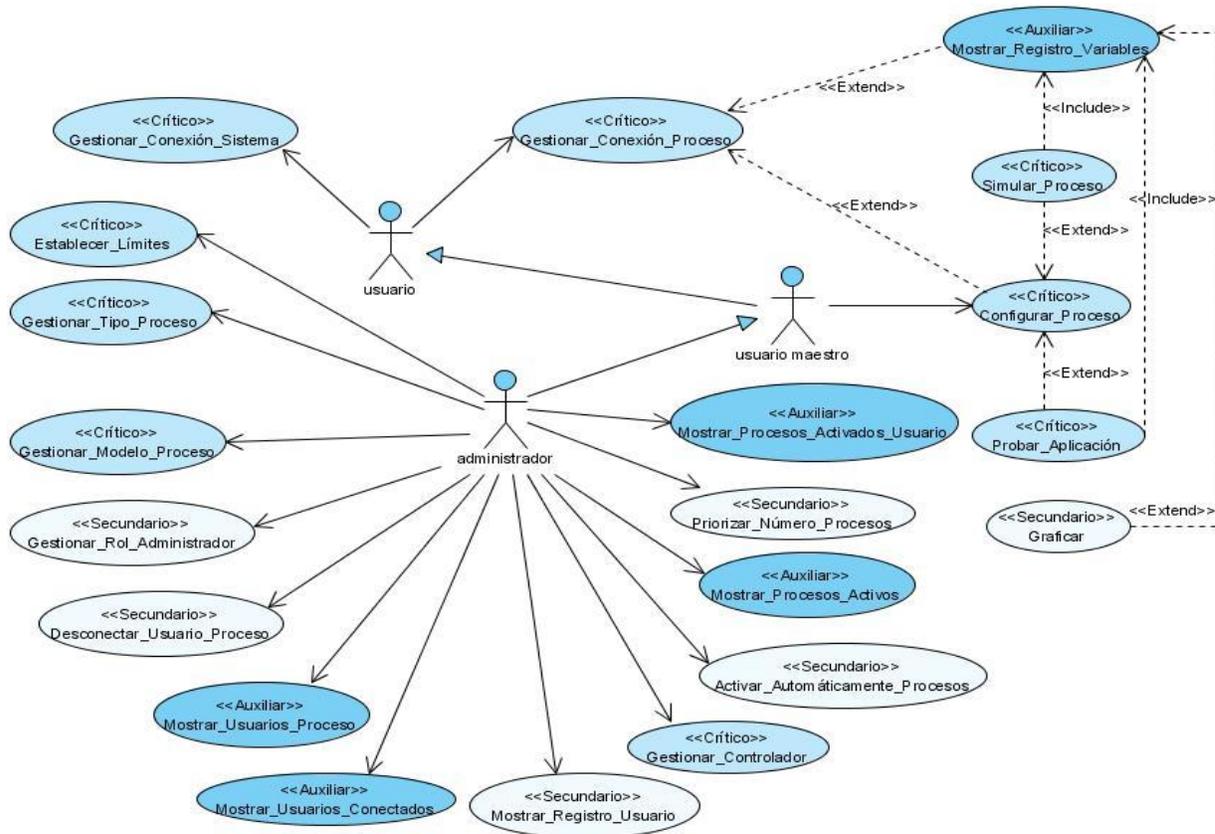


Figura 2: Diagrama de CU del Sistema

### 2.3.1 Especificación textual de los casos de uso.

La especificación de los casos de uso contiene las propiedades textuales de los casos de uso. Es utilizada junto a una herramienta de gestión de requisitos para especificar y marcar los requerimientos dentro de las propiedades de los casos de uso. Las descripciones que se muestran a continuación corresponden a los casos de uso no críticos del NVP. Cada caso de uso hace referencia al prototipo de interfaz correspondiente.

**Descripción detallada del CUS Desconectar\_Usuario\_Proceso:** (Ver Anexo 47)

<b>Caso de Uso – 3</b>	Desconectar_Usuario_Proceso	
<b>Actores</b>	Administrador	
<b>Propósito</b>	Permite que el administrador lleve a cabo la acción de desconectar a un usuario de un proceso.	
<b>Resumen</b>	El CUS se inicia cuando el administrador de la aplicación selecciona desconectar a un usuario de un proceso, el sistema termina el acceso del usuario a un proceso del Nodo Virtual y finaliza el caso de uso.	
<b>Pos-condiciones</b>	Usuario desconectado del proceso.	
<b>Flujo Normal de Eventos</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	
<ol style="list-style-type: none"> <li>1. El Administrador solicita la desconexión de un usuario de un proceso.</li> <li>2. El Administrador selecciona el proceso del cual va a desconectar al usuario.</li> <li>3. El Administrador selecciona el usuario que quiere desconectar.</li> <li>4. El Administrador confirma o no desconectar al usuario.</li> </ol>	<ol style="list-style-type: none"> <li>1.1. El sistema muestra el listado de los procesos activos en ese momento.</li> <li>2.1. El sistema muestra el listado de los usuarios conectados a ese proceso.</li> <li>3.1. El sistema muestra un mensaje de advertencia para confirmar la acción a realizar.</li> <li>4.1. Si el administrador confirma la desconexión del usuario, el sistema comprueba si es usuario maestro del proceso.</li> <li>4.2. Si el usuario es maestro, verifica la existencia de otros usuarios conectados al proceso.</li> <li>4.3. Si no existen otros usuarios conectados al proceso, éste se desactiva.</li> </ol>	

	<p>4.4. El sistema elimina al usuario de la lista de usuarios conectados al proceso.</p> <p>4.5. El sistema actualiza el registro de acciones del usuario, envía una notificación a éste y finaliza el caso de uso.</p>
<b>Cursos Alternos</b>	
	<p>4.1. Si el administrador no confirma la desconexión del usuario, el sistema no realiza ninguna acción, lo envía a la acción 2 y finaliza el caso de uso.</p> <p>4.2. Si el usuario es común, el sistema pasa a la acción 4.4.</p> <p>4.3. Si existen otros usuarios conectados al proceso, mediante un sistema de notificaciones gestiona un nuevo maestro y pasa a la acción 4.4.</p>
<b>Referencia:</b>	RF 5, RF 38, RF 41, RF 43
<b>Prioridad:</b>	Secundario

**Descripción detallada del CUS Gestionar\_Rol\_Administrador:** (Ver Anexo 48)

<b>Caso de Uso – 4</b>	Gestionar_Rol_Administrador
<b>Actores</b>	Administrador
<b>Propósito</b>	Permite al administrador gestionar (adicionar, modificar o eliminar) a un administrador.
<b>Resumen</b>	El CUS se inicia cuando el administrador de la aplicación selecciona la opción de Gestionar Rol de Administrador, luego selecciona el tipo de

	gestión, el sistema realiza la acción seleccionada por el administrador y finaliza el caso de uso.
<b>Descripción</b>	
<b>Pos-condiciones</b>	<ol style="list-style-type: none"> <li>1. Información del rol de administrador adicionada a la Base de Datos.</li> <li>2. Información del rol de administrador modificada en la Base de Datos.</li> <li>3. Información del rol de administrador eliminada de la Base de Datos.</li> </ol>
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. El Administrador selecciona la opción de Gestionar Rol de Administrador.	1.1. El sistema muestra el listado de los administradores existentes en la Base de Datos y las opciones de Adicionar Administrador, Modificar Administrador y Eliminar Administrador.
<b>Escenario 1: Adicionar Administrador.</b>	
<ol style="list-style-type: none"> <li>1. El Administrador selecciona la opción de Adicionar Administrador.</li> <li>2. El Administrador introduce los datos requeridos.</li> </ol>	<ol style="list-style-type: none"> <li>1.1. El sistema muestra el formulario a completar para la adición de un nuevo administrador.</li> <li>2.1. El sistema verifica los datos introducidos por el administrador.</li> <li>2.2. Si los datos introducidos son correctos, el sistema verifica si el administrador a adicionar existe.</li> <li>2.3. Si no existe, el sistema adiciona dicha información en la Base de Datos y finaliza el caso de uso.</li> </ol>
<b>Cursos Alternos</b>	
	2.2. Si los datos introducidos por el administrador

	<p>son incorrectos, el sistema muestra un mensaje de error indicándole donde está el dato erróneo y lo envía a la acción 2.</p> <p>2.3. Si el administrador a adicionar existe, el sistema muestra un mensaje indicándolo y lo envía a la acción 2.</p>
<b>Escenario 2: Modificar Administrador.</b>	
<p>1. El Administrador selecciona el administrador a modificar.</p> <p>2. El Administrador selecciona la opción de Modificar Administrador.</p> <p>3. El Administrador realiza los cambios necesarios a los datos.</p>	<p>2.1. El sistema localiza los datos del administrador y los muestra, listos para modificar.</p> <p>3.1. El sistema verifica los datos modificados por el administrador.</p> <p>3.2. Si los datos introducidos por el administrador son correctos, el sistema actualiza la Base de Datos y finaliza el caso de uso.</p>
<b>Cursos Alternos</b>	
	<p>3.2. Si los datos introducidos por el administrador son incorrectos, el sistema muestra un mensaje de error indicándole donde está el dato erróneo y lo envía a la acción 3.</p>
<b>Escenario 3: Eliminar Administrador.</b>	
<p>1. El Administrador selecciona el administrador a eliminar.</p> <p>2. El Administrador selecciona la opción de Eliminar Administrador.</p> <p>3. El Administrador confirma o no la eliminación.</p>	<p>2.1. El sistema muestra un mensaje de advertencia para confirmar la acción a realizar.</p> <p>3.1. Si el administrador confirma la eliminación, el sistema elimina el administrador seleccionado y finaliza el caso de uso.</p>
<b>Cursos Alternos</b>	

	3.1. Si el administrador no confirma la eliminación, finaliza el caso de uso sin ejecutar ninguna acción.
<b>Referencia:</b>	RF 6, RF 7, RF 8, RF 9
<b>Prioridad:</b>	Secundario

**Descripción detallada del CUS Graficar:** (Ver Anexo 46)

<b>Caso de Uso – 10</b>	Graficar (extendido de Mostrar_Registro_Variables)	
<b>Actores</b>		
<b>Propósito</b>	Permite mostrar gráficamente un proceso.	
<b>Resumen</b>	El CUS se inicia dentro del flujo de acciones del caso de uso Mostrar_Registro_Variables, cuando el usuario maestro confirma seleccionar variables para graficar, el sistema realiza la acción seleccionada y finaliza el caso de uso.	
<b>Descripción</b>		
<b>Pos-condiciones</b>	Resultado de la simulación o prueba de un proceso mostrado gráficamente.	
<b>Flujo Normal de Eventos</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	
<p>2.1. El Usuario Maestro selecciona las variables de salida.</p> <p>3.1. El Usuario Maestro confirma o no el inicio de la graficación.</p>	<p>1. Cuando se inicia el caso de uso Mostrar_Registro_Variables y el usuario selecciona la opción de graficar el sistema activa el caso de uso Graficar, para mostrar gráficamente el resultado de la simulación o la prueba del proceso.</p> <p>2. El sistema muestra una interfaz para que el</p>	

	<p>usuario escoja las variables a graficar.</p> <p>3. El sistema muestra un mensaje de advertencia para confirmar el inicio de la graficación.</p> <p>4. Si el usuario confirma la acción, el sistema muestra la graficación y finaliza el caso de uso.</p>
<b>Cursos Alternos</b>	
	<p>4. Si el usuario no confirma la graficación, el sistema finaliza el caso de uso sin realizar ninguna acción.</p>
<b>Referencia:</b>	RF 26
<b>Prioridad:</b>	Secundario

**Descripción detallada del CUS Mostrar\_Registro\_Variables:** (Ver Anexo 45)

<b>Caso de Uso – 12</b>	Mostrar_Registro_Variables (incluido de Simular_Proceso y Probar_Aplicación)
<b>Actores</b>	
<b>Propósito</b>	Permite que el usuario maestro obtenga el reporte de la simulación de un proceso.
<b>Resumen</b>	El CUS se inserta en el flujo de acciones de los casos de uso Simular_Proceso y Probar_Aplicación para mostrar el resultado de la simulación o la prueba. El sistema da la posibilidad de imprimir el reporte.
<b>Descripción</b>	
<b>Pos-condiciones</b>	Reporte de simulación o prueba.

<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
<p>2.1. El Usuario Maestro solicita o no detener la simulación.</p> <p>3.1. El Usuario Maestro solicita o no imprimir el historial.</p> <p>4.1. El Usuario Maestro solicita o no graficar el comportamiento de las variables de salida.</p> <p>5.1. El Usuario Maestro confirma o no la selección.</p> <p>7.1. El Usuario Maestro selecciona o no la opción de Cancelar.</p>	<p>1. Durante la ejecución de los casos de uso Simular_Proceso y Probar_Aplicación el sistema activa el caso de uso Mostrar_Registro_Variables para mostrar el resultado de la simulación o prueba.</p> <p>2. El sistema muestra el historial de las variables de salida de la simulación o prueba.</p> <p>3. Si el usuario maestro solicita detener la simulación o prueba, el sistema habilita las opciones imprimir y graficar.</p> <p>4. Si el usuario maestro solicita imprimir, el sistema realiza la impresión.</p> <p>5. Si el usuario maestro solicita graficar, el sistema muestra un mensaje de advertencia para la confirmación de la selección de variables.</p> <p>6. Si el usuario maestro confirma la selección, el sistema muestra la interfaz correspondiente al CUS Graficar.</p> <p>7. Si el usuario maestro selecciona la opción de Cancelar, finaliza el caso de uso.</p>
<b>Cursos Alternos</b>	
	<p>6. Si el usuario maestro no confirma la selección, el sistema muestra la representación gráfica de todas las variables de salida.</p>

<b>Referencia:</b>	RF 26, RF 45
<b>Prioridad:</b>	Auxiliar

**Descripción detallada del CUS Activar\_Automáticamente\_Procesos:** (Ver Anexo 53)

<b>Caso de Uso – 14</b>	Activar_Automáticamente_Procesos	
<b>Actores</b>	Administrador	
<b>Propósito</b>	Permite que el administrador active uno o varios procesos del sistema.	
<b>Resumen</b>	El CUS se inicia cuando el administrador selecciona la opción de activar automáticamente uno o varios procesos, introduce los datos necesarios, el sistema realiza la acción seleccionada y finaliza el caso de uso.	
<b>Descripción</b>		
<b>Pos-condiciones</b>	Procesos Activados.	
<b>Flujo Normal de Eventos</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	
<ol style="list-style-type: none"> <li>1. El Administrador selecciona la opción de Activar Automáticamente Procesos.</li> <li>2. El Administrador selecciona un tipo de proceso.</li> <li>3. El Administrador selecciona el proceso a activar.</li> <li>4. El Administrador programa el tiempo de activación.</li> <li>5. El Administrador selecciona la opción de activar proceso por defecto o por parámetros.</li> </ol>	<ol style="list-style-type: none"> <li>1.1. El sistema muestra un listado con los tipos de procesos existentes en la aplicación.</li> <li>2.1. El sistema muestra los procesos del tipo seleccionado que están sin activar.</li> <li>3.1. El sistema comprueba la capacidad disponible para activar el proceso.</li> <li>3.2. Si existe capacidad disponible para activar el proceso, el sistema muestra una interfaz para programar el tiempo de activación.</li> <li>5.1. Si el administrador decide activar proceso por defecto, el sistema realiza la acción, lo asigna</li> </ol>	

	como usuario maestro del proceso, actualiza su registro de acciones y finaliza el caso de uso.
<b>Cursos Alternos</b>	
	<p>3.2. Si no existe capacidad disponible, el sistema muestra un mensaje indicándoselo al administrador y finaliza el caso de uso sin realizar ninguna acción.</p> <p>5.1. Si el administrador selecciona activar procesos por parámetros, el sistema realiza la acción, lo asigna como usuario maestro del proceso, el sistema muestra la interfaz correspondiente al CUS Configurar_Proceso y finaliza el caso de uso.</p>
<b>Referencia:</b>	RF 13, RF 32, RF 33, RF 34, RF 38
<b>Prioridad:</b>	Secundario

**Descripción detallada del CUS Priorizar\_Número\_Procesos:** (Ver Anexo 54)

<b>Caso de Uso – 15</b>	Priorizar_Número_Procesos
<b>Actores</b>	Administrador
<b>Propósito</b>	Permite al administrador que priorice los procesos.
<b>Resumen</b>	El CUS inicia cuando el administrador selecciona la opción de priorizar el número de procesos por tipos de procesos, introduce los datos necesarios, el sistema realiza la acción seleccionada por el administrador y finaliza el caso de uso.
<b>Descripción</b>	

<b>Pos-condiciones</b>	Procesos priorizados.	
<b>Flujo Normal de Eventos</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	
<p>1. El Administrador selecciona la opción de priorizar el número de procesos por tipo de proceso.</p> <p>2. El Administrador selecciona el tipo de proceso al cual quiere priorizar el número de procesos activos.</p> <p>3. El Administrador introduce los datos requeridos.</p>	<p>1.1. El sistema muestra un listado de los tipos de procesos existentes en la Base de Datos.</p> <p>2.1. El sistema muestra un formulario a llenar para priorizar los procesos.</p> <p>3.1. El sistema verifica los datos introducidos por el administrador.</p> <p>3.2. Si los datos introducidos son correctos, el sistema verifica que la cantidad introducida sea mayor o igual que la cantidad de procesos activos del tipo seleccionado en ese instante.</p> <p>3.3. Si la cantidad introducida es mayor o igual, el sistema actualiza la configuración del Nodo Virtual y finaliza el caso de uso.</p>	
<b>Cursos Alternos</b>		
	<p>3.2. Si los datos introducidos por el administrador son incorrectos, el sistema muestra un mensaje de error indicándole donde está el dato erróneo y lo envía a la acción 3.</p> <p>3.3. Si la cantidad introducida es menor, el sistema muestra un mensaje indicándolo y lo envía a la acción 3.</p>	
<b>Referencia:</b>	RF 13, RF 35, RF 36	

<b>Prioridad:</b>	Secundario
-------------------	------------

**Descripción detallada del CUS Mostrar\_Registro\_Usuario: (Ver Anexo 55)**

<b>Caso de Uso – 16</b>	Mostrar_Registro_Usuario	
<b>Actores</b>	Administrador	
<b>Propósito</b>	Permite al administrador obtener un reporte de las acciones realizadas por los usuarios del sistema.	
<b>Resumen</b>	El CUS se inicia cuando el administrador de la aplicación selecciona la opción de Mostrar Registro de Usuario, el sistema realiza la acción seleccionada por el administrador y finaliza el caso de uso.	
<b>Descripción</b>		
<b>Pos-condiciones</b>	Registro de un usuario mostrado al administrador.	
<b>Flujo Normal de Eventos</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	
1. El Administrador selecciona la opción de Mostrar Registro de Usuario.	1.1. El sistema muestra un listado de todos los usuarios.	
2. El Administrador selecciona el usuario del que desea conocer su registro de acciones.	2.1. El sistema localiza el registro correspondiente al usuario y se los muestra al administrador.	
3. El Administrador solicita o no imprimir el registro.	3.1. Si el administrador solicita imprimir, el sistema realiza la impresión.	
4. El Administrador selecciona o no la opción de Cerrar.	4.1. Si el administrador selecciona la opción de Cerrar, finaliza el caso de uso.	

Cursos Alternos	
Referencia:	RF 39
Prioridad:	Secundario

**Descripción detallada del CUS Mostrar\_Procesos\_Activos: (Ver Anexo 55)**

<b>Caso de Uso – 17</b>	Mostrar_Procesos_Activos
<b>Actores</b>	Administrador
<b>Propósito</b>	Permite que el administrador obtenga un listado de todos los procesos activos existentes en la aplicación.
<b>Resumen</b>	El CUS se inicia cuando el administrador selecciona la opción de Mostrar los Procesos Activos en ese instante en la aplicación. El sistema da la posibilidad de imprimir el listado.
<b>Descripción</b>	
<b>Pos-condiciones</b>	Listado de procesos activos.
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. El Administrador selecciona la opción Mostrar los Procesos Activos en la aplicación. 2. El Administrador confirma o no imprimir el reporte.	1.1. El sistema muestra todos los procesos activos en ese instante y un mensaje de advertencia para confirmar imprimir el reporte.  2.1. Si el administrador confirma la impresión, el sistema imprime el reporte y finaliza el caso de uso.
<b>Cursos Alternos</b>	

	2.1. Si el administrador no confirma la impresión, finaliza el caso de uso.
<b>Referencia:</b>	RF 41, RF 45
<b>Prioridad:</b>	Auxiliar

**Descripción detallada del CUS Mostrar\_Usuarios\_Conectados:** (Ver Anexo 55)

<b>Caso de Uso – 18</b>	Mostrar_Usuarios_Conectados
<b>Actores</b>	Administrador
<b>Propósito</b>	Permite que el administrador obtenga un listado de todos los usuarios conectados a la aplicación en ese instante.
<b>Resumen</b>	El CUS se inicia cuando el administrador selecciona la opción de Mostrar los Usuarios Conectados en ese instante en la aplicación. El sistema da la posibilidad de imprimir el listado.
<b>Descripción</b>	
<b>Pos-condiciones</b>	Listado de usuarios conectados.
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>

<p>1. El Administrador selecciona la opción Mostrar los Usuarios Conectados a la aplicación.</p> <p>2. El Administrador confirma o no imprimir el reporte.</p>	<p>1.1. El sistema muestra todos los usuarios conectados en ese momento y un mensaje de advertencia para confirmar imprimir el reporte.</p> <p>2.1. Si el administrador confirma la impresión, el sistema imprime el reporte y finaliza el caso de uso.</p>
<b>Cursos Alternos</b>	
	<p>2.1. Si el administrador no confirma la impresión, finaliza el caso de uso.</p>
<b>Referencia:</b>	RF 42, RF 45
<b>Prioridad:</b>	Auxiliar

**Descripción detallada del CUS Mostrar\_Usuarios\_Proceso:** (Ver Anexo 55)

<b>Caso de Uso – 19</b>	Mostrar_Usuarios_Proceso
<b>Actores</b>	Administrador
<b>Propósito</b>	Permite que el administrador obtenga un listado de todos los usuarios conectados a un proceso.
<b>Resumen</b>	El CUS se inicia cuando el administrador selecciona la opción de Mostrar los Usuarios Conectados a un Proceso. El sistema da la posibilidad de imprimir el listado.
<b>Descripción</b>	
<b>Pos-condiciones</b>	Listado de usuarios conectados a un proceso.

Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
<p>1. El Administrador selecciona la opción Mostrar los Usuarios Conectados a un Proceso.</p> <p>2. El Administrador selecciona un modelo de proceso.</p> <p>3. El Administrador confirma o no imprimir el reporte.</p>	<p>1.1. El sistema muestra una lista con todos los modelos de procesos activos.</p> <p>2.1. El sistema muestra todos los usuarios conectados en ese momento al modelo de proceso seleccionado y un mensaje para confirmar imprimir el reporte.</p> <p>3.1. Si el administrador confirma la impresión, el sistema imprime el reporte y finaliza el caso de uso.</p>
Cursos Alternos	
	<p>3.1. Si el administrador no confirma la impresión, finaliza el caso de uso.</p>
<b>Referencia:</b>	RF 43, RF 45
<b>Prioridad:</b>	Auxiliar

**Descripción detallada del CUS Mostrar\_Procesos\_Activados\_Usuario: (Ver Anexo 55)**

<b>Caso de Uso – 20</b>	Mostrar_Procesos_Activados_Usuario
<b>Actores</b>	Administrador
<b>Propósito</b>	Permite que el administrador obtenga un listado de todos los procesos que han sido activados por un usuario.
<b>Resumen</b>	El CUS se inicia cuando el administrador selecciona la opción de Mostrar los Procesos Activados por un Usuario. El sistema da la posibilidad de

	imprimir el listado.	
<b>Descripción</b>		
<b>Pos-condiciones</b>	Listado de procesos activados por un usuario.	
<b>Flujo Normal de Eventos</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	
1. El Administrador selecciona la opción Mostrar los Procesos Activados por un Usuario. 2. El Administrador selecciona un usuario. 3. El Administrador confirma o no imprimir el reporte.	1.1. El sistema muestra una lista con todos los usuarios conectados al sistema. 2.1. El sistema muestra los procesos que han sido activados por el usuario seleccionado y un mensaje de advertencia para confirmar imprimir el reporte. 3.1. Si el administrador confirma la impresión, el sistema imprime el reporte y finaliza el caso de uso.	
<b>Cursos Alternos</b>		
	3.1 Si el administrador no confirma la impresión, finaliza el caso de uso.	
<b>Referencia:</b>	RF 44, RF 45	
<b>Prioridad:</b>	Auxiliar	

Consultar las descripciones de los casos de uso críticos:

CUS Gestionar\_Conexión\_Sistema: (Ver Anexo 6 y 41).

CUS Gestionar\_Conexión\_Proceso: (Ver Anexo 7 y 43).

CUS Establecer\_Límites: (Ver Anexo 8 y 49).

CUS Gestionar\_Tipo\_Proceso: (Ver Anexo 9 y 50).

CUS Gestionar\_Modelo\_Proceso: (Ver Anexo 10 y 51).

CUS Configurar\_Proceso: (Ver Anexo 11 y 44).

CUS Simular\_Proceso: (Ver Anexo 12 y 44).

CUS Probar\_Aplicación: (Ver Anexo 13 y 44).

CUS Gestionar\_Controlador: (Ver Anexo 14).

### **Conclusiones**

Después de la revisión de los requerimientos, la reestructuración del diagrama de CUS y de la incorporación de nuevas actividades a los casos de uso, resultado de las actividades de la Ingeniería de Requisitos se concluye que:

- La Elicitación con el uso de técnicas de levantamiento de requisitos: como entrevistas, arqueología de documentos y tormenta de ideas, fue el elemento principal para captar todas las necesidades de la parte interesada. El Análisis permitió estructurarlos por objetivos del sistema. La Especificación facilitó el entendimiento de las funcionalidades y restricciones del sistema. Por último la Validación posibilitó evaluar la calidad del refinamiento realizado.
- Se logró un mayor nivel de detalle en cuanto a la descripción de los casos de uso, que permitirá concluir el análisis y diseño del NVP.

## Capítulo 3: Análisis y Diseño del NVP

### Introducción

En el presente capítulo se obtendrán los principales artefactos para la eficiente implementación de un Nodo Virtual que simule procesos industriales. En el análisis definido por la metodología de desarrollo escogida se tuvieron en cuenta los requisitos funcionales durante la confección de los diagramas de clases y los diagramas de colaboración correspondientes a cada caso de uso. El diseño consolidará el análisis propuesto con los diagramas de clases, la distribución en subsistemas de los casos de uso, los diagramas de secuencia correspondientes a cada escenario y la aplicación de patrones de arquitectura y de diseño como una manera más práctica de describir ciertos aspectos; teniendo en cuenta las cualidades o propiedades con las que el NVP debe contar.

### 3.1 Análisis.

Las actividades del análisis son desarrolladas con el objetivo de facilitar la entrada al diseño, por lo que son un paso inicial y una primera aproximación conceptual para aumentar el nivel de especificidad en aras de garantizar el cubrimiento de los requisitos funcionales y obtener una visión de qué hace el sistema.

El modelo de análisis está compuesto por artefactos como: clases del análisis y realizaciones de casos de uso del análisis.

Las clases del análisis van a representar abstracciones de conceptos, en las cuales deben incluirse atributos y operaciones a un nivel alto.

La realización de casos de uso del análisis describe cómo se lleva a cabo y se ejecuta un caso de uso determinado en términos de las clases del análisis y de sus objetos en interacción.

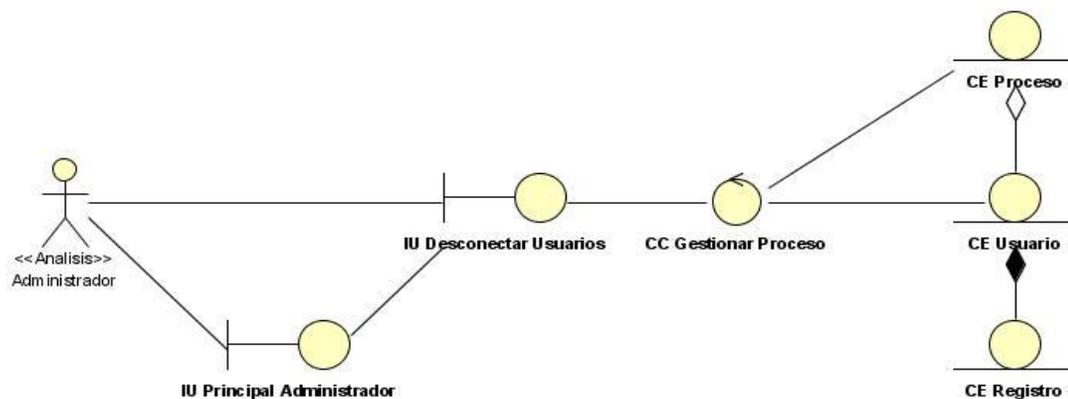
Para ello RUP propone que para cada caso de uso:

- Se construya un diagrama de clases que muestre las clases del análisis participantes.
- Se construya alguno de los tipos de diagrama de interacción para representar el flujo en términos de objetos del análisis. (30)

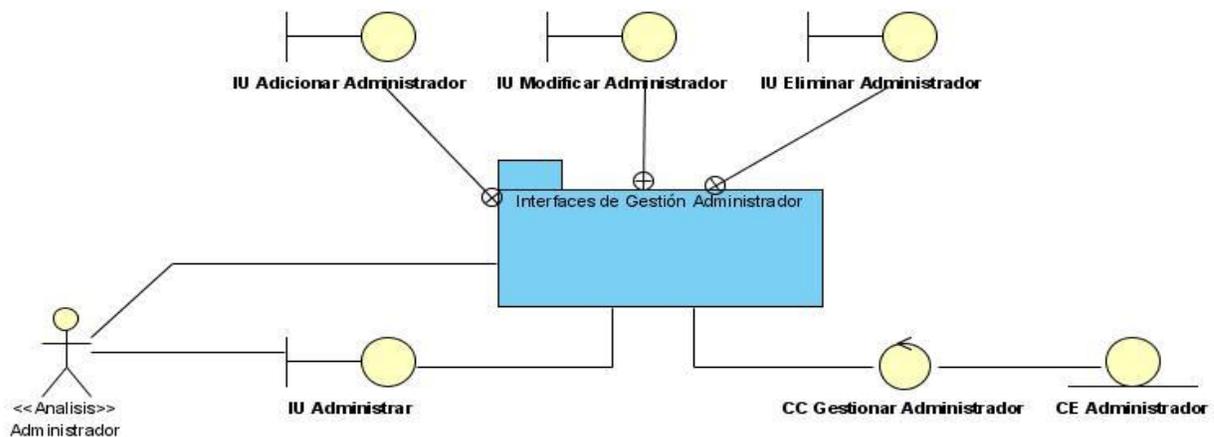
**3.1.1 Diagrama de Clases del Análisis.**

Las clases del análisis se centran en los requisitos funcionales y son evidentes en el dominio del problema porque representan conceptos y relaciones del dominio. Tienen atributos y entre ellas se establecen relaciones de asociación, agregación-composición, generalización-especialización y tipos asociativos. RUP propone clasificar a las clases en: interfaz, control y entidad. (30)

A continuación se muestran los diagramas de clases del análisis correspondientes a los casos de uso no críticos.



**Figura 3: Diagrama de Clases del Análisis. CU Desconectar\_Usuario\_Proceso**



**Figura 4: Diagrama de Clases del Análisis. CU Gestionar\_Rol\_Administrador**

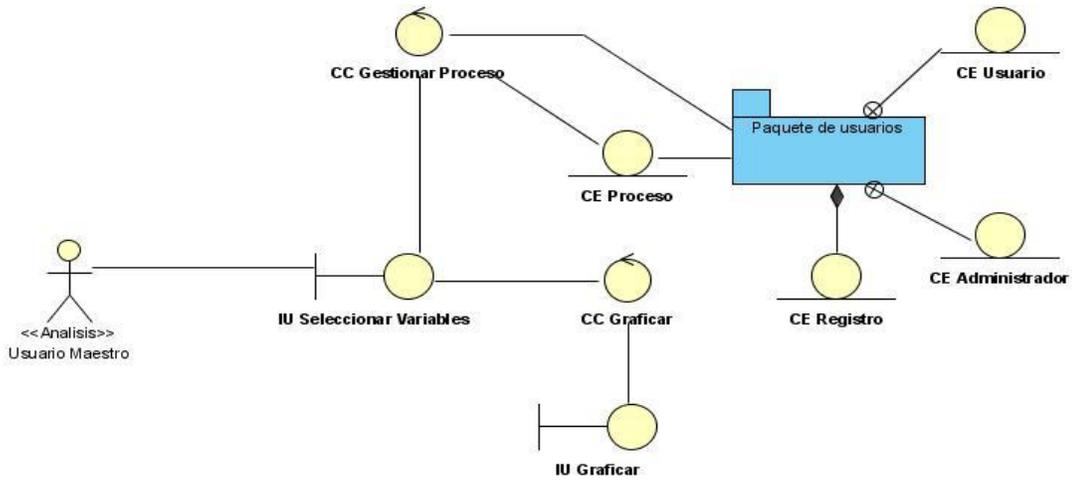


Figura 5: Diagrama de Clases del Análisis. CU Graficar

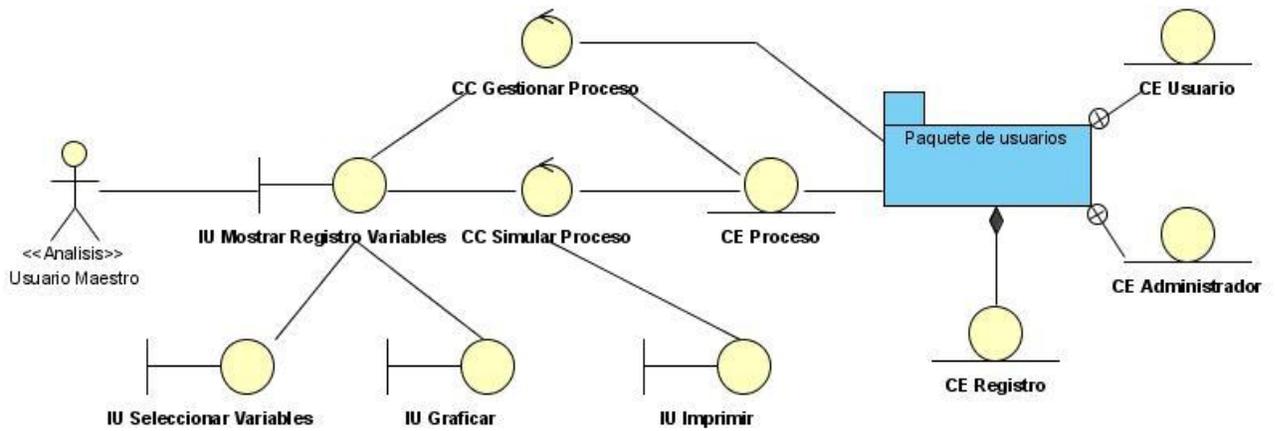


Figura 6: Diagrama de Clases del Análisis. CU Mostrar\_Registro\_Variables

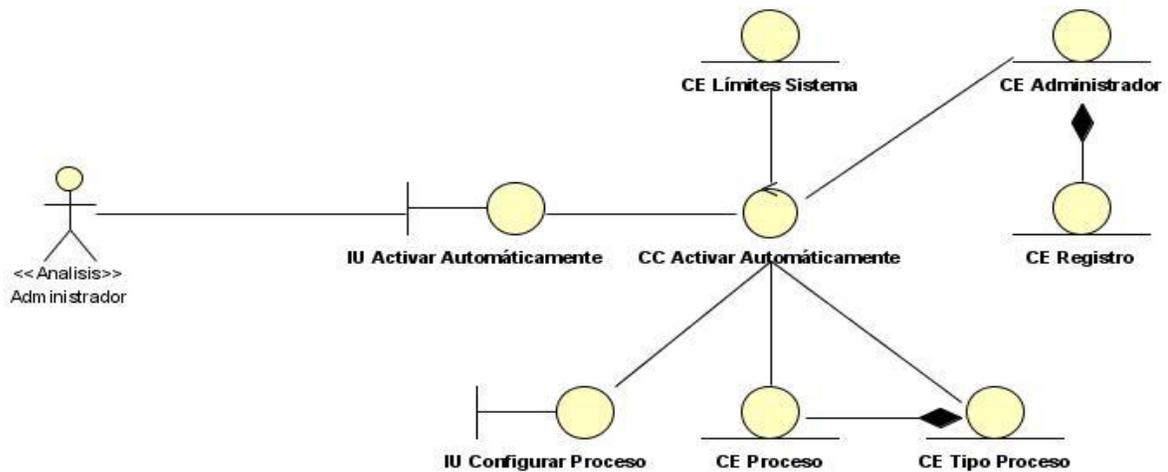


Figura 7: Diagrama de Clases del Análisis. CU Activar\_Automáticamente\_Procesos

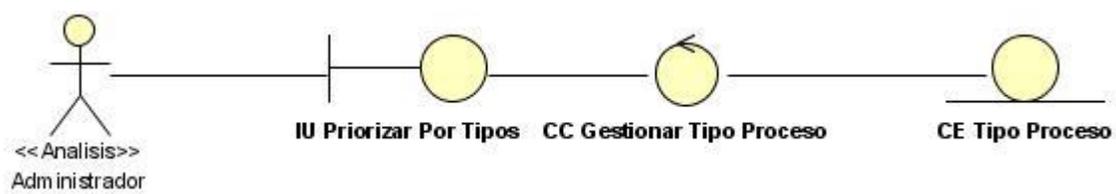


Figura 8: Diagrama de Clases del Análisis. CU Priorizar\_Número\_Procesos

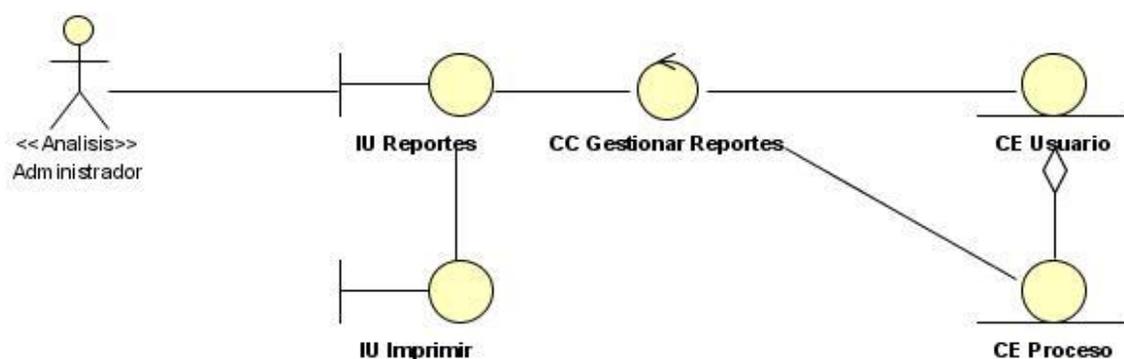


Figura 9: Diagrama de Clases del Análisis. Casos de Uso Asociados a Reportes (CU Mostrar\_Registro\_Usuario, CU Mostrar\_Procesos\_Activos, CU Mostrar\_Usuarios\_Conectados, CU Mostrar\_Usuarios\_Proceso, CU Mostrar\_Procesos\_Activados\_Usuario)

Consultar los diagramas de clases del análisis de los casos de uso críticos:

CU Gestionar\_Conexión\_Sistema: (Ver Anexo 15).

CU Gestionar\_Conexión\_Proceso: (Ver Anexo 16).

CU Establecer\_Límites: (Ver Anexo 17).

CU Gestionar\_Tipo\_Proceso: (Ver Anexo 18).

CU Gestionar\_Modelo\_Proceso: (Ver Anexo 19).

CU Configurar\_Proceso: (Ver Anexo 20).

CU Simular\_Proceso: (Ver Anexo 21).

CU Probar\_Aplicación: (Ver Anexo 21).

CU Gestionar\_Controlador: (Ver Anexo 22).

### **3.1.2 Realización de casos de uso del análisis.**

Para la realización de los casos de uso del análisis propuestos para el NVP se utilizarán los diagramas de interacción. De dichos diagramas se escogieron los de colaboración, los cuales destacan la organización estructural de los objetos que envían y reciben mensajes, ofrecen una visión clara del flujo de control en el contexto en el que se desarrollan, son útiles en la fase exploratoria para identificar objetos y la estructura estática viene dada por los enlaces; la dinámica por el envío de mensajes (30).

A continuación se presentan los diagramas de colaboración correspondientes a los casos de uso no críticos Desconectar\_Usuario\_Proceso y Gestionar\_Rol\_Administrador. En estos diagramas se podrá apreciar la inclusión para esta iteración de los flujos alternos con el propósito de detallar mejor el análisis. (Consultar los diagramas de los restantes casos de uso en los anexos desde el 23 hasta el 39).

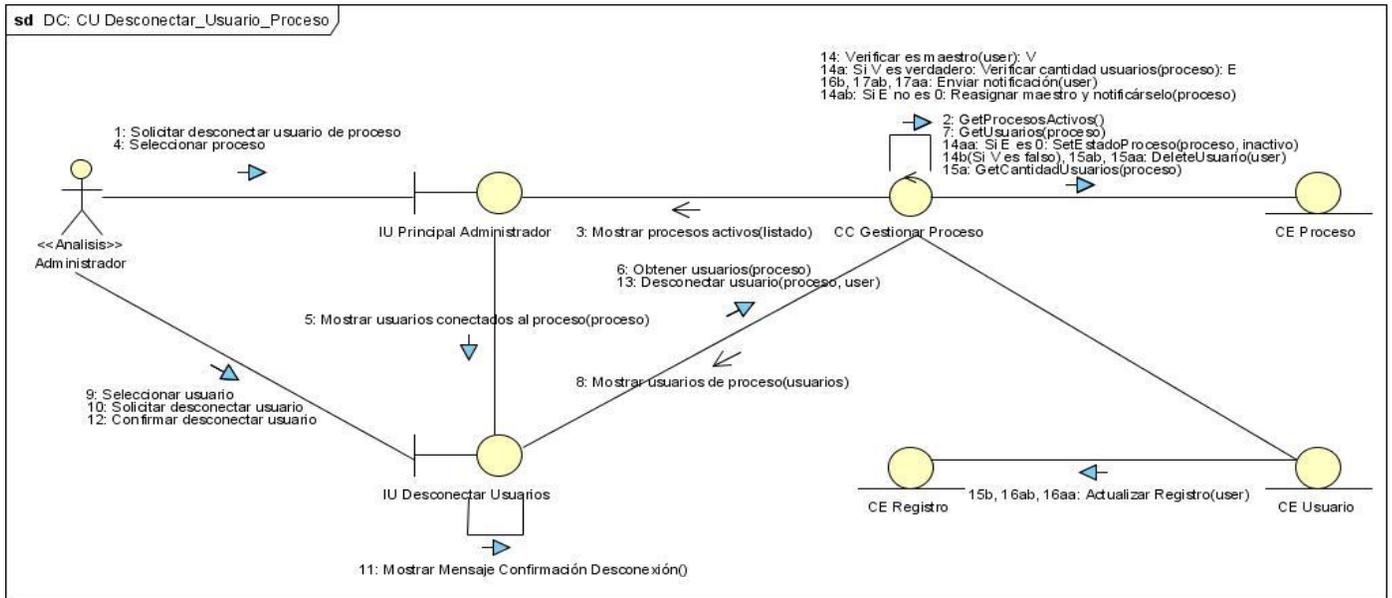


Figura 10: Diagrama de Colaboración del Análisis. CU Desconectar\_Usuario\_Proceso

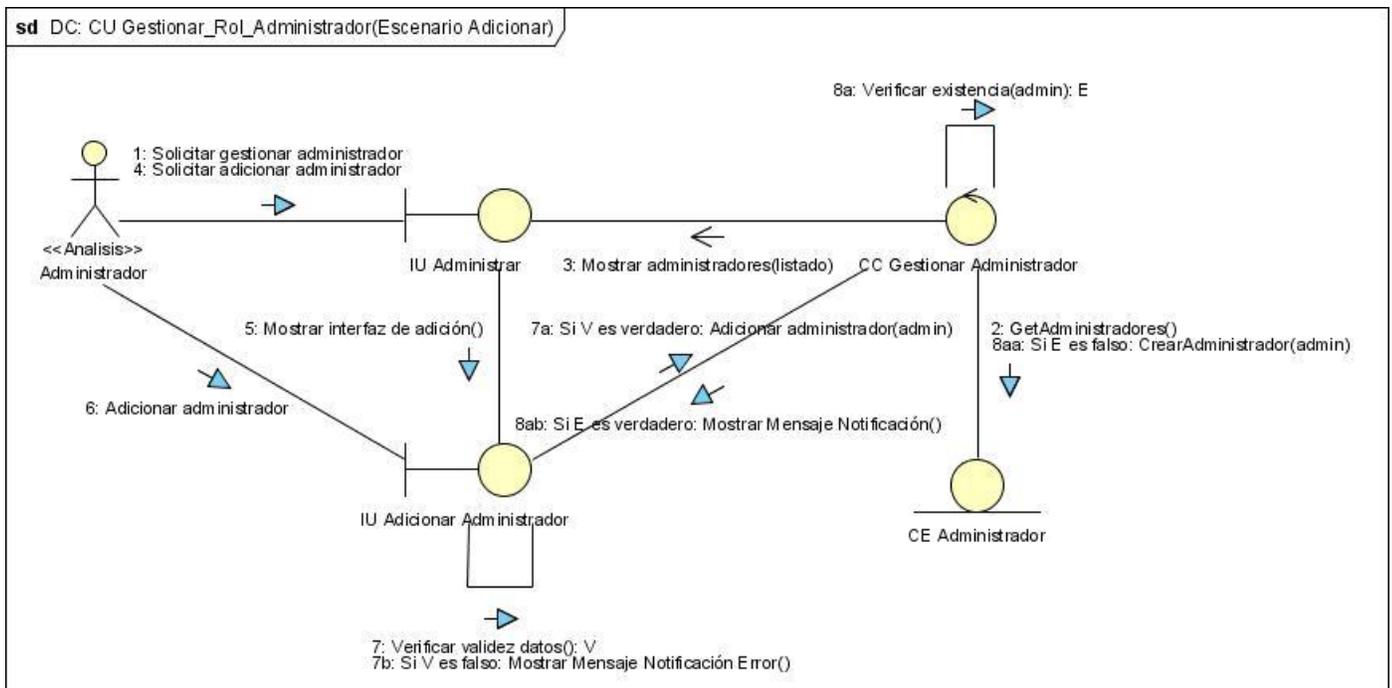


Figura 11: Diagrama de Colaboración del Análisis. CU Gestionar\_Rol\_Administrador (Escenario Adicionar)

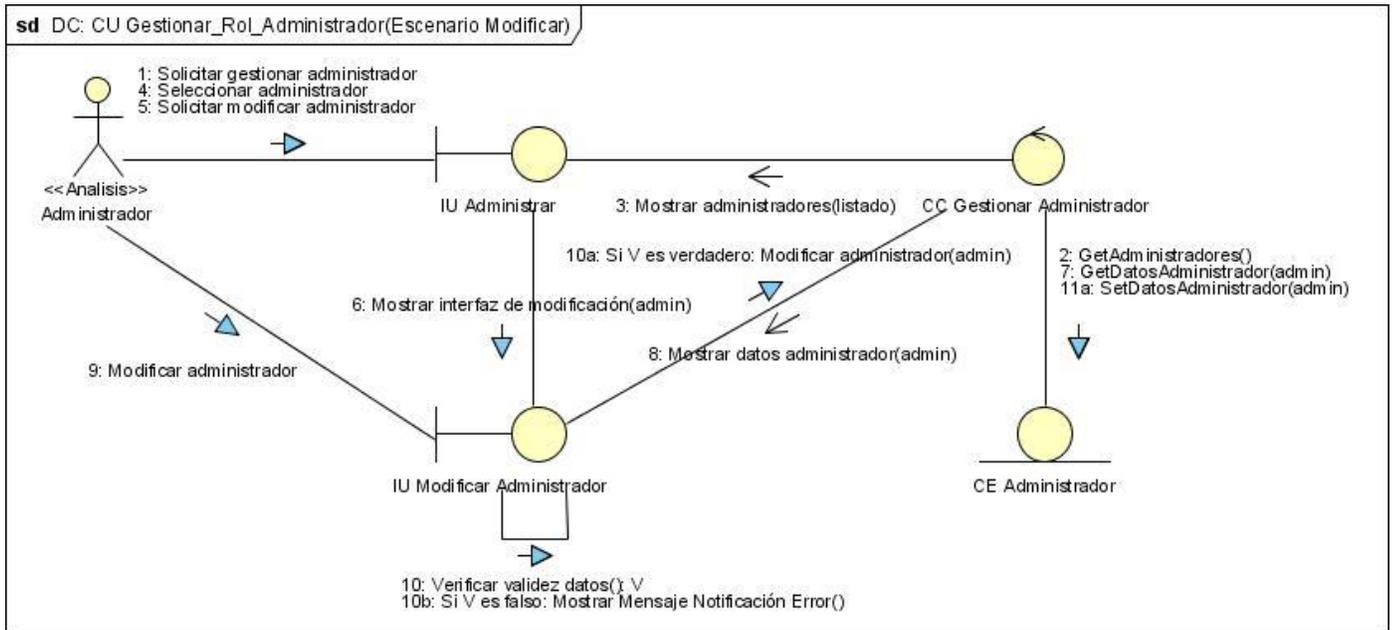


Figura 12: Diagrama de Colaboración del Análisis. CU Gestionar\_Rol\_Administrador (Escenario Modificar)

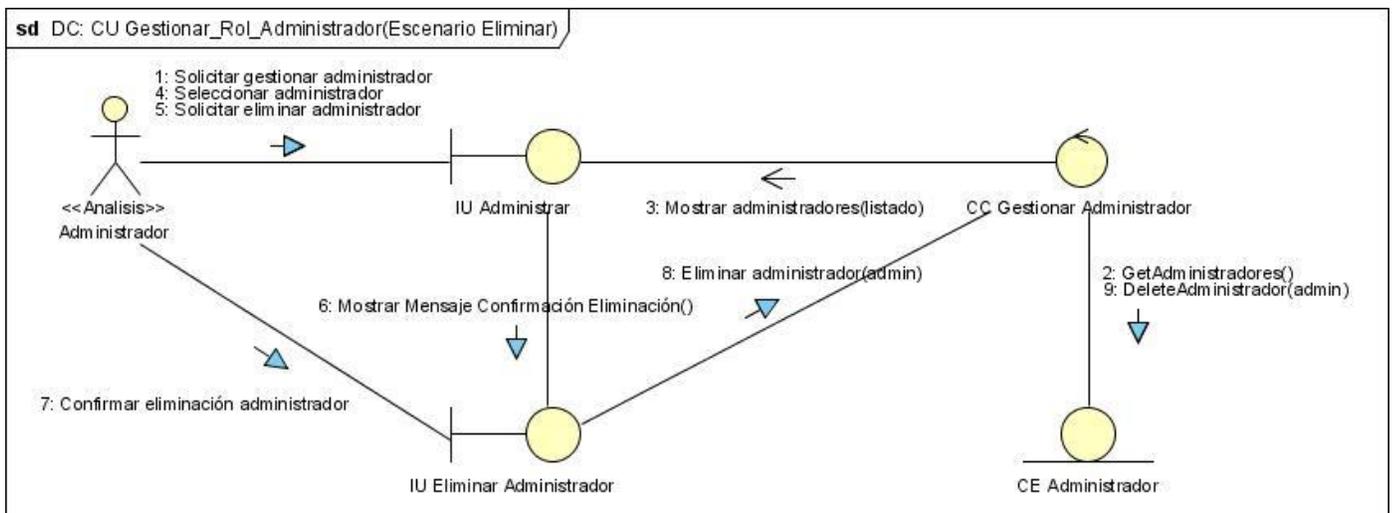


Figura 13: Diagrama de Colaboración del Análisis. CU Gestionar\_Rol\_Administrador (Escenario Eliminar)

### 3.2 Diseño.

El diseño es un refinamiento del análisis que tiene en cuenta los requisitos no funcionales, o sea cómo cumple el sistema sus objetivos. El diseño debe ser suficiente para que el sistema pueda ser implementado sin ambigüedades. En el diseño se modela el sistema y se define una arquitectura que soporte todos los requisitos. Concretamente se puede definir como propósitos del diseño:

- Adquirir una comprensión de los aspectos relacionados con los requisitos no funcionales y restricciones relacionadas con los lenguajes de programación, componentes reutilizables, sistemas operativos, tecnologías de distribución y concurrencia y tecnologías de interfaz de usuario.
- Crear una entrada apropiada y un punto de partida para actividades de implementación, capturando los requisitos o subsistemas individuales, interfaces y clases.
- Descomponer los trabajos de implementación en partes más manejables que puedan ser llevadas a cabo por diferentes equipos de desarrollo. (30)

#### 3.2.1 Arquitectura definida para el sistema.

La arquitectura es el esqueleto o base de una aplicación. En ésta se analiza la aplicación desde varios puntos de vista. En la arquitectura aparecen los artefactos más importantes y diferentes, para establecer un esquema de cómo deben ser los próximos artefactos a construir, o sea es puramente un semi-molde al que se deben ajustar sin muchos cambios los restantes artefactos a construir en la aplicación o solución. De obtenerse un artefacto demasiado diferente a los demás, éste formaría parte de la arquitectura. En RUP, ésta se compone de 4+1 vistas: Vista Lógica, Vista de Procesos, Vista de Implementación, Vista de Despliegue y estas cuatro regidas por la Vista de Casos de Uso. (30)

Para el estructurar el diseño del NVP, se escogió la Arquitectura en tres capas. Las interfaces y los formularios que permiten la interacción con el usuario se encuentran en la capa de Aplicación. Las clases encargadas de implementar las reglas del negocio y las restricciones que desea la parte interesada se encuentran en la capa Lógica o de Negocio, éstas son las clases controladoras y entidades que permiten la realización de los casos de uso. (12)

En el caso específico del NVP la capa de Aplicación estará compuesta por todos los elementos de la interfaz: formularios para la gestión de datos referentes a usuarios, procesos, controladores, tipos de

procesos y límites; además de representaciones gráficas de los diferentes procesos que se simularon y tablas que permitan visualizar reportes. La capa de Negocio abarca los subsistemas Simular-Graficar, Reportes, Conexión y Gestión. Por otro lado todo lo concerniente a la gestión de la persistencia de los datos estaría enmarcado en la capa de Acceso a Datos.

### 3.2.2 Justificación de los Patrones de Diseño utilizados.

Durante el diseño del NVP se proponen utilizar patrones generales de software para asignar responsabilidades (GRASP) (12), que constituyen principios básicos a tener en cuenta cuando se quiere construir eficazmente un software orientado a objetos. Entre los más evidentes en el diseño propuesto se encuentran los patrones: Experto, Creador, Bajo Acoplamiento, Alta Cohesión y Controlador.

Para contribuir a una implementación eficiente se hizo necesario el estudio de los patrones de la pandilla de los cuatro (GOF, Gang Of Four) (12), de ellos se seleccionaron tres: Fachada y Singleton, propuestos en el diseño inicial del NVP y Prototype para esta iteración, los cuales se explicarán a continuación.

#### **Patrón Solitario (Singleton)**

Con este patrón se garantiza una única instancia de aquellas clases que se desee tener una sola en toda la aplicación, proporcionando un punto de acceso global a dichas clases. Tiene como ventajas que reduce el espacio de nombres y es una mejora sobre las variables globales. Es usado debido a la necesidad de trabajar con el mismo objeto en distintos momentos y distintos subsistemas (12). Específicamente en el NVP, el administrador crea un modelo de proceso desde el subsistema de Gestión, ese mismo modelo va a ser llamado desde los subsistemas de Conexión y Simular-Graficar para realizar funcionalidades como conexión a un proceso o para comprobar los límites de usuarios para realizar la simulación. Este patrón se utiliza también en las clases fachadas de los subsistemas, garantizando una única instancia de ellas, esto suele hacerse cuando se aplica el Patrón Fachada, que se explicará a continuación.

#### **Patrón Fachada (Facade)**

Proporciona una interfaz sencilla unificada para un conjunto de clases o subsistemas, siendo más fácil de usar. Permite reducir la complejidad y minimizar las dependencias, el acceso de los usuarios a los subsistemas es por medio de la clase fachada. Ella es la encargada de reenviar las peticiones a los objetos de los subsistemas, para no acceder directamente a los mismos, ocultando la complejidad de ellos. Este patrón favorece a un Bajo Acoplamiento entre los usuarios y los subsistemas, respondiendo a

uno de los patrones GRASP, va a permitir variar las clases internas, de manera transparente a los usuarios que las utilizan; favoreciendo de esta forma a la división en capas de la aplicación. Como se dijo anteriormente estas clases van a utilizar el patrón Solitario lo que va a permitir un acceso global a ellas.  
(12)

### **Patrón Prototipo (Prototype)**

Es un patrón de creación, cuyo objetivo es especificar los tipos de objetos a crear por medio de una instancia que hace de prototipo, creando nuevos objetos copiando dicha instancia, todo se basa en clonar un prototipo dado. Específicamente en el NVP, cuando un usuario decide ser maestro de un proceso, el sistema le muestra una interfaz de configuración, el usuario puede realizar la configuración del proceso y guardarla en la Base de Datos para hacerla visible al resto de los usuarios o, puede replicar el proceso, en este caso, se crearía una copia del mismo usando el patrón Prototype y su configuración no persistiría en la Base de Datos y sólo sería visible para el usuario que lo haya replicado.

### **3.2.3 Modelo de Diseño.**

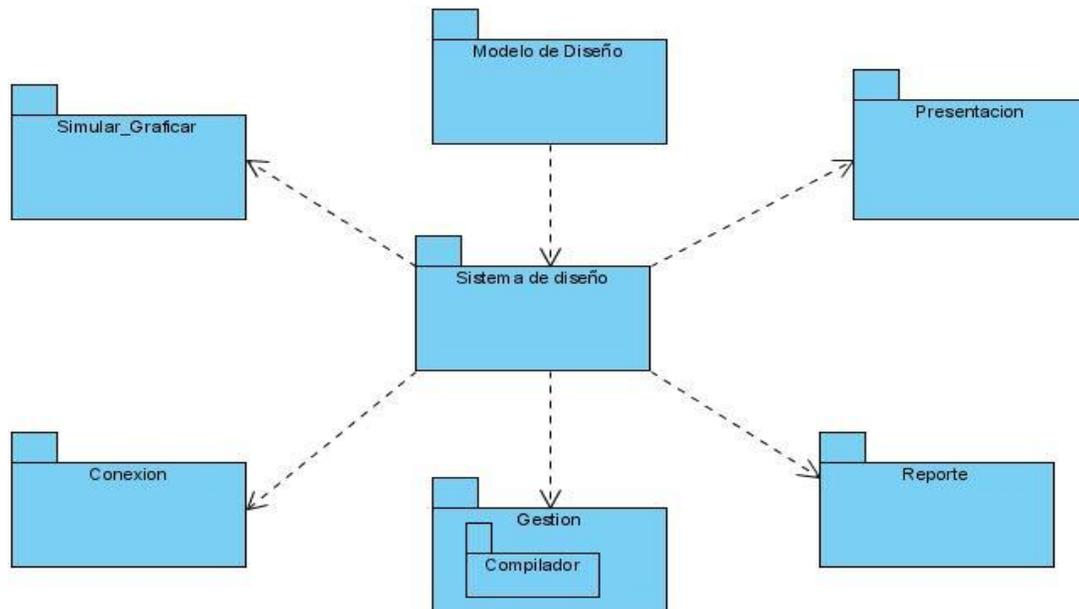
El Modelo de Diseño es un modelo de objetos que describe la realización de los casos de uso y al mismo tiempo constituye una abstracción del modelo de implementación y del código fuente, constituye una entrada esencial a las actividades de implementación y prueba. El Modelo de Diseño constituye un vehículo de análisis durante la fase de elaboración, pero se refina con un diseño detallado durante la fase de construcción.

RUP propone que el Modelo de Diseño básicamente contenga:

- Introducción: Una descripción textual que sirve como breve introducción al modelo.
- Paquetes y Subsistemas de Diseño: Los paquetes y subsistemas del diseño representados en una jerarquía y una breve descripción de ellos.
- Diagramas: los diagramas de clases del diseño y diagramas de interacción (colaboración y/o secuencia) del diseño. Estos últimos también llamados realización de casos de uso.
- Clases, interfaces, relaciones, etc.: contenidas en los paquetes y una breve descripción de ellos.

(30)

En este caso el modelo de diseño está conformado por cinco subsistemas: Presentación, Conexión, Gestión, Simular-Graficar y Reportes.



**Figura 14: Modelo de Diseño del Sistema**

### Subsistema Presentación

Este subsistema se encuentra en la capa de presentación de la arquitectura del sistema, en él solamente se encuentran aquellos componentes que permiten interactuar con el usuario, es decir formularios o ventanas para mostrar o capturar datos.

El diseño propuesto para este subsistema, al igual que los demás, está vinculado estrechamente a la plataforma y lenguaje de programación en el cual se va a implementar. Se usarán las vistas para mostrar diferentes interfaces. Esto permite que las vistas sean llamadas con la ocurrencia de un evento y se muestren dentro de una parte específica de la forma principal, evitando así la creación de varias formas.

(12)

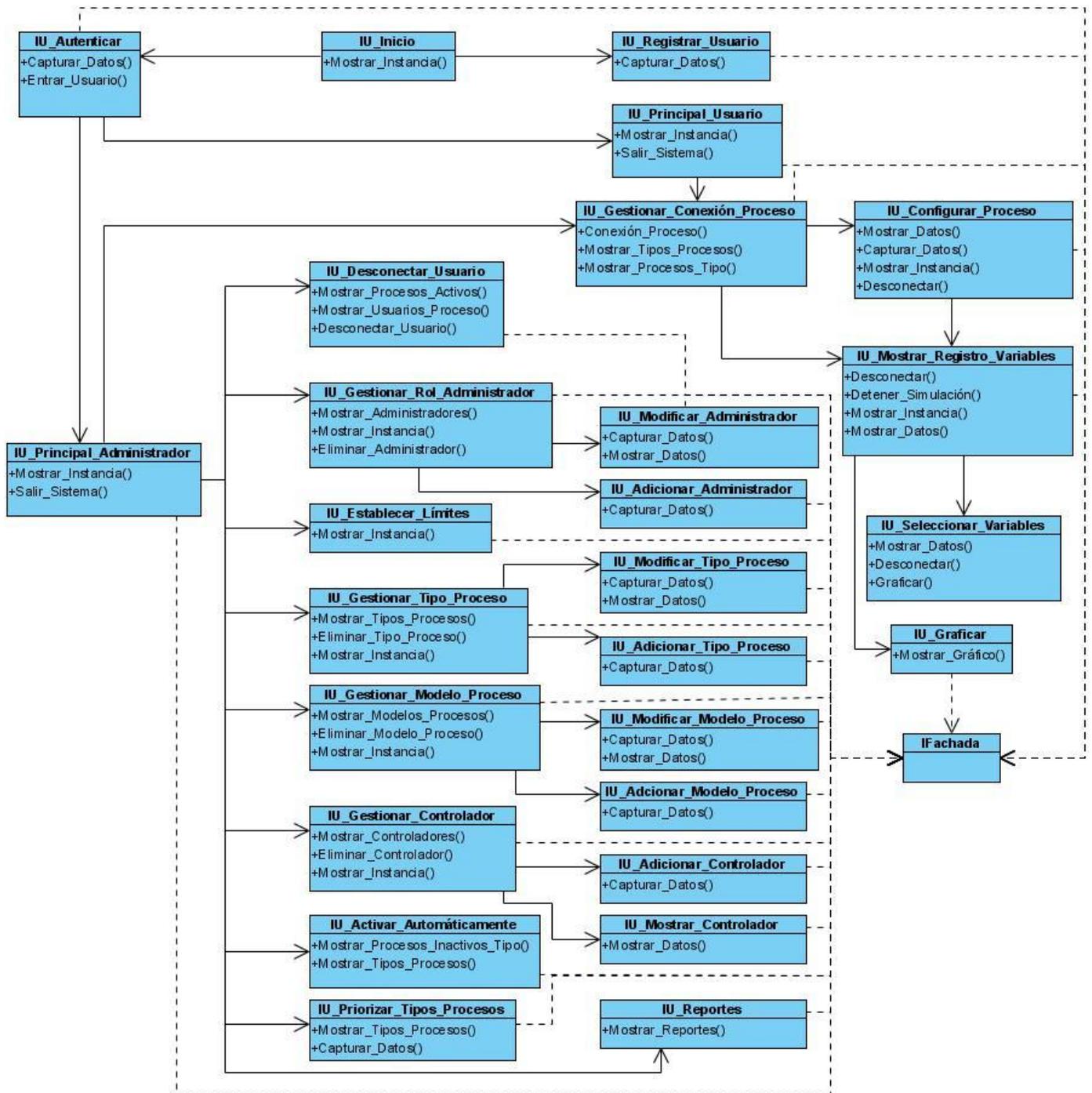


Figura 15: Subsistema de Presentación

- IU\_Inicio: Su función es mejorar la interfaz gráfica y hacer más amena la interacción con el usuario. Es un splash que carga al iniciar la aplicación con una imagen representativa del software. Dentro de ella se encuentran dos hipervínculos que dan paso a las interfaces para registrarse y autenticarse. (Ver Anexo 40).
- IU\_Registrar\_Usuario: La función de esta forma es obtener los datos de los usuarios que utilizarán el sistema para luego dar acceso al mismo a través de la IU\_Autenticar, esta forma tendrá varios campos que permitirán entrar el nombre del usuario, sus apellidos, el id que desea tener y la contraseña con la que accederá al sistema. (Ver Anexo 41).
- IU\_Autenticar: Los usuarios registrados y los administradores pueden acceder al sistema a través de una forma en la que deben introducir el id de usuario y la contraseña. (Ver Anexo 41).
- IU\_Principal\_Usuario: Es la forma principal de la aplicación que se le muestra al usuario. Contiene un menú a la izquierda con las distintas funcionalidades que puede realizar el usuario dentro del sistema. Mediante los eventos On-Click de cada uno de los componentes del menú se muestran las vistas y formas para capturar o mostrar información. (Ver Anexo 42).
- IU\_Principal\_Administrador: Sus funciones son las mismas que las de la IU\_Principal\_Usuario pero para el caso en que el autenticado sea un administrador. (Ver Anexo 42).
- IU\_Gestionar\_Conexión\_Proceso: Es la forma que permite a los usuarios conectarse a un proceso, para ello muestra un menú a la izquierda con los tipos de procesos que tiene el NVP y en la derecha en dependencia del tipo seleccionado un listado de procesos para escoger a cual conectarse. (Ver Anexo 43).
- IU\_Configurar\_Proceso: Es la forma que se invoca desde un desde la IU\_Gestionar\_Conexión\_Proceso, si el usuario se convierte en maestro del proceso al que se conectó. Tiene como función brindar la posibilidad de realizar la configuración de un proceso. Posee un grupo de campos encargados de capturar los datos y enviarlos hacia otros subsistemas para su procesamiento y posteriormente ser usados para realizar la simulación del proceso. (Ver Anexo 44).

- IU\_Mostrar\_Registro\_Variables: Esta forma se puede invocar desde dos eventos, uno desde la IU\_Gestionar\_Conexión\_Proceso y otro desde la IU\_Configurar\_Proceso. Tiene como función mostrar el resultado de la simulación del proceso en forma de tabla. (Ver Anexo 45).
- IU\_Seleccionar\_Variables: Esta forma permite al usuario seleccionar las variables con las que desea graficar el resultado de la simulación. (Ver Anexo 46).
- IU\_Graficar: Esta forma muestra la representación gráfica del resultado de la simulación, en dependencia de las variables seleccionadas en la IU\_Seleccionar\_Variables. (Ver Anexo 46).
- IU\_Desconectar\_Usuario: Esta forma permite al administrador desconectar a un usuario de un proceso. Primeramente se mostrará un listado de los procesos activos, en dependencia del proceso que se seleccione se mostrará un listado de los usuarios conectados al mismo para que se pueda realizar la operación de desconexión. (Ver Anexo 47).
- IU\_Gestionar\_Rol\_Administrador: Esta forma permite al administrador gestionar la información referente a los administradores del NVP. Se mostrará un listado de los administradores y las diferentes opciones de gestión. (Ver Anexo 48).
- IU\_Establecer\_Límites: Esta forma tiene como función brindar al administrador la posibilidad de establecer los límites del sistema. Tiene cargos que se encargan de capturar los datos y enviarlos a otros subsistemas para su procesamiento. (Ver Anexo 49).
- IU\_Gestionar\_Tipo\_Proceso: Esta forma permite al administrador gestionar la información referente a los tipos de procesos del NVP. Se mostrará un listado de los tipos de procesos y las diferentes opciones de gestión. (Ver Anexo 50).
- IU\_Gestionar\_Modelo\_Proceso: Esta forma permite al administrador gestionar la información referente a los modelos de procesos del NVP. Se mostrará un listado de los modelos de procesos y las diferentes opciones de gestión. (Ver Anexo 51).
- IU\_Gestionar\_Controlador: Esta forma permite al administrador gestionar la información referente a los controladores del NVP. Se mostrará un listado de los controladores y las diferentes opciones de gestión. (Ver Anexo 52).

- IU\_Activar\_Automáticamente: Esta forma permite al administrador activar automáticamente uno o varios procesos. Primeramente se mostrará un listado de los tipos de procesos y a continuación un listado con los modelos de procesos del tipo de proceso que se seleccione con las opciones de activar por defecto o activar por parámetros. (Ver Anexo 53).
- IU\_Priorizar\_Tipos\_Procesos: Esta forma permite al administrador establecer la cantidad límite de procesos que pueden haber activos por tipo de proceso. Se mostrará un listado de los tipos de procesos y en dependencia del tipo de proceso que se seleccione se mostrará un campo para introducir la cantidad que se desee. (Ver Anexo 54).
- IU\_Reportes: Esta forma permite al administrador solicitar e imprimir reportes necesarios para la administración del NVP. (Ver Anexo 55).

### **Subsistema Conexión**

Se encuentra en la capa de negocio, su función es gestionar la conexión tanto al sistema como a los procesos (12). Va a estar integrado por seis clases, tres controladoras y tres entidades. Su funcionamiento va a depender de los subsistemas Simular-Graficar y Gestión debido a que las clases CC\_Activar\_Automáticamente, CC\_Gestionar\_Conexión\_Sistema y CC\_Gestionar\_Conexión\_Proceso utilizan clases que se encuentran dentro del subsistema Gestión; y CC\_Gestionar\_Conexión\_Proceso utiliza clases que se encuentran dentro del subsistema Simular-Graficar.

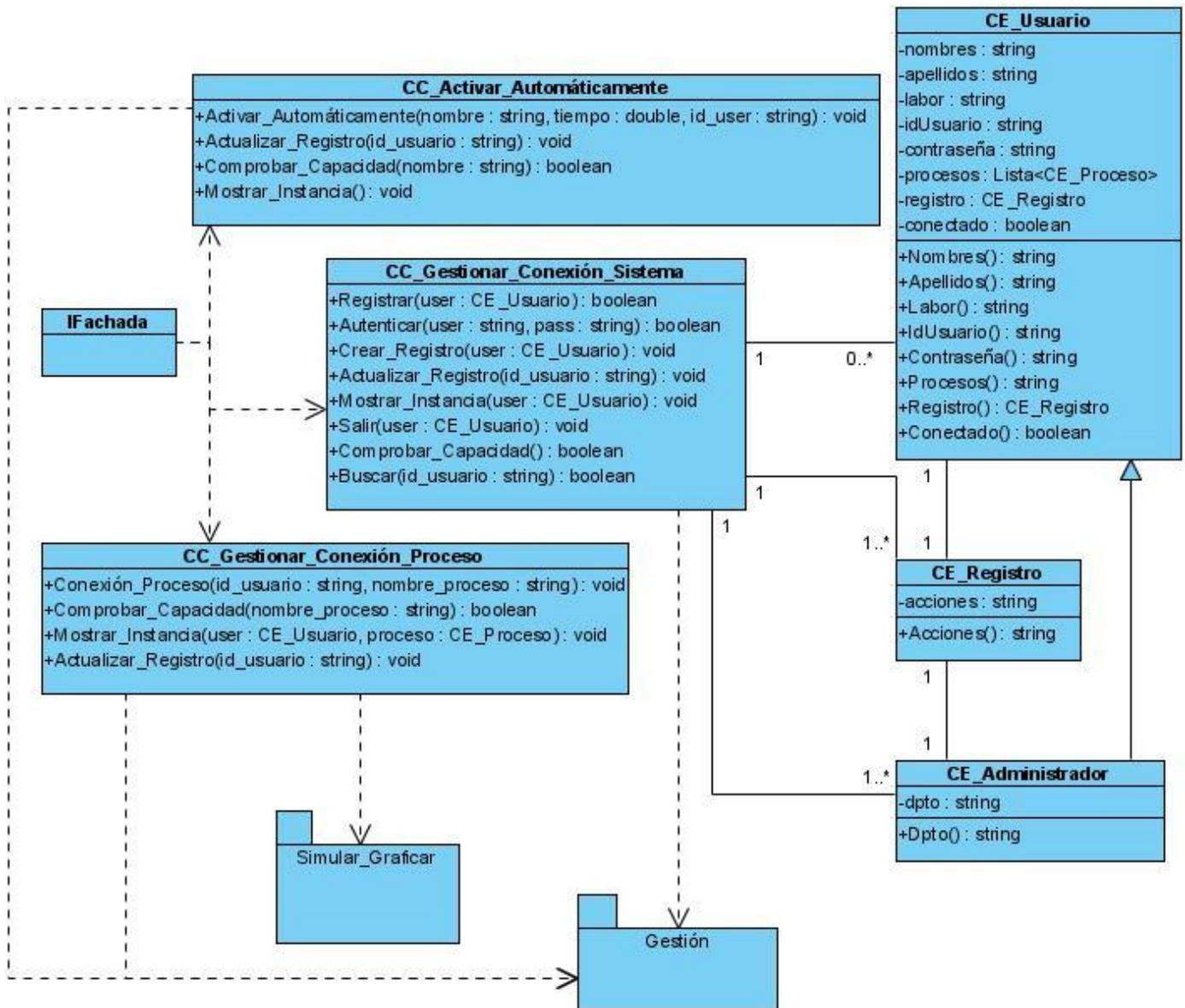


Figura 16: Subsistema de Conexión

### Subsistema Gestión

Se encuentra en la capa de negocio, su función es gestionar la información importante para el sistema como la referente a los tipos de procesos, los modelos de proceso, los controladores, los administradores; y establecer límites necesarios para su buen funcionamiento. Va a estar integrado por quince clases, cinco

clases controladoras y diez entidades. Su funcionamiento depende de los subsistemas Conexión, Compilador y Simular\_Graficar, debido a que las clases CC\_Gestionar\_Tipo\_Proceso, CC\_Gestionar\_Controlador, CC\_Gestionar\_Proceso y CC\_Gestionar\_Administrador utilizan clases que se encuentran dentro del subsistema Conexión; las clases CC\_Gestionar\_Controlador y CC\_Gestionar\_Proceso utilizan clases del subsistema Compilador y a su vez la CC\_Gestionar\_Proceso utiliza clases del subsistema Simular-Graficar.

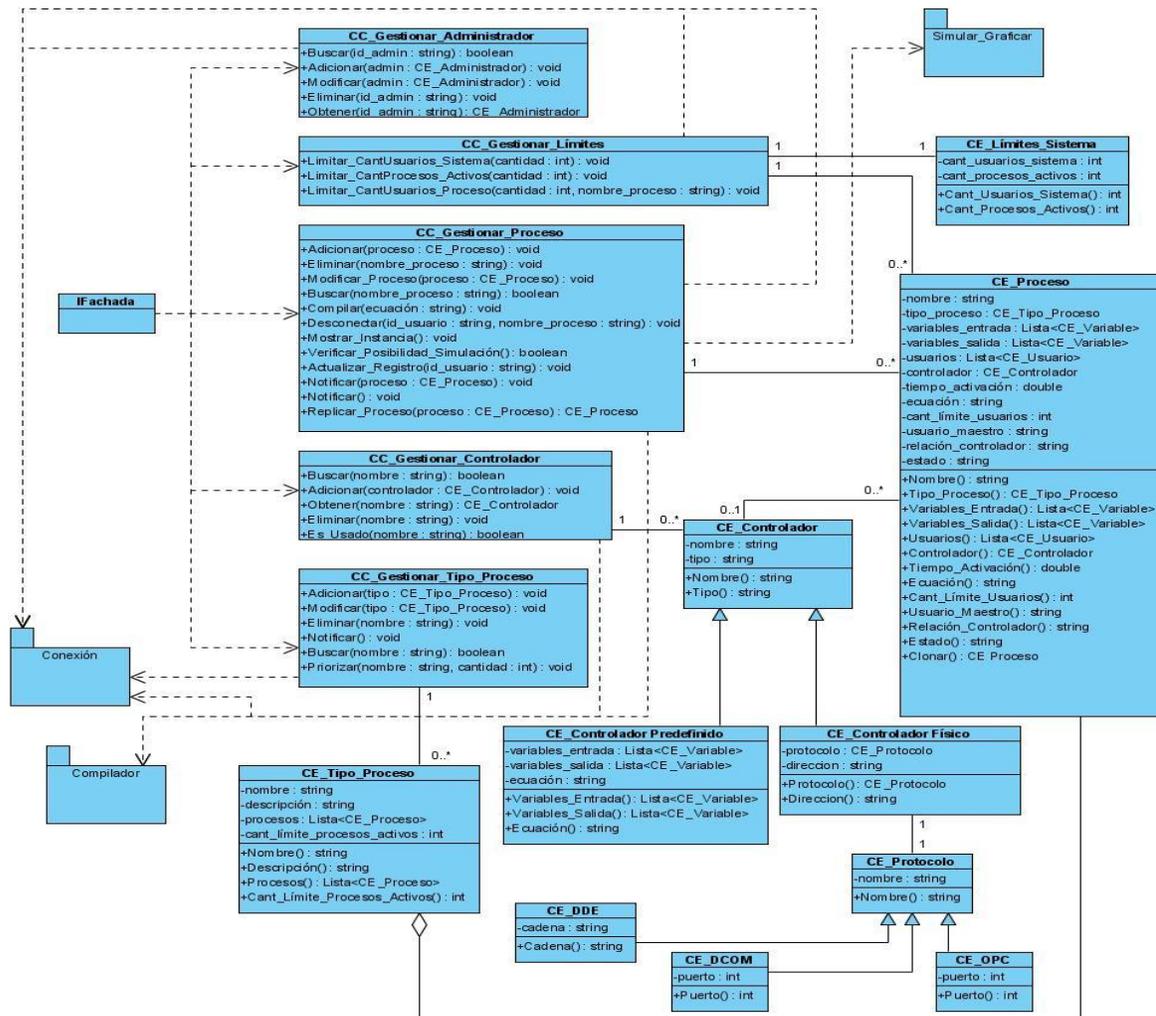


Figura 17: Subsistema de Gestión

El subsistema Compilador constituye un asistente matemático capaz de identificar y resolver el tipo de ecuación que se introduce cuando se está adicionando un controlador predefinido, un proceso o cuando se está configurando este último.

### Subsistema Simular-Graficar

Se encuentra en la capa de negocio, su función es permitir la simulación y la graficación de los procesos industriales. Los procesos luego de pasar por las diferentes actividades de gestión, entre ellas la configuración, pueden ser simulados o probados en tiempo real. El subsistema va a estar conformado por dos clases controladoras: CC\_Graficar y CC\_Simular\_Proceso, que utilizarán las clases entidades contenidas en los subsistemas Gestión y Conexión.

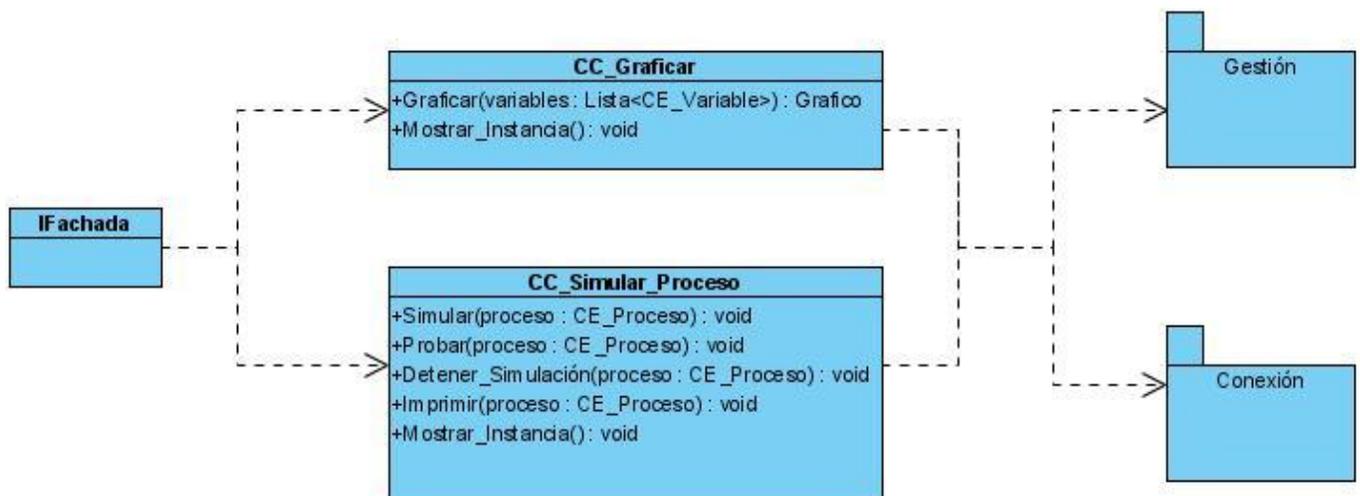


Figura 18: Subsistema Simular-Graficar

### Subsistema Reportes

Se encuentra en la capa de negocio y posee gran vínculo con la capa de acceso a datos debido a que su principal función es proveer información necesaria para la administración del NVP. Realiza reportes como los usuarios conectados a la aplicación, los usuarios conectados a un proceso, el registro de acciones de un usuario, los procesos activos y los que han sido activados por un usuario determinado. El subsistema va a estar conformado por una clase controladora CC\_Gestionar\_Reportes que utiliza entidades de los subsistemas Gestión y Conexión.

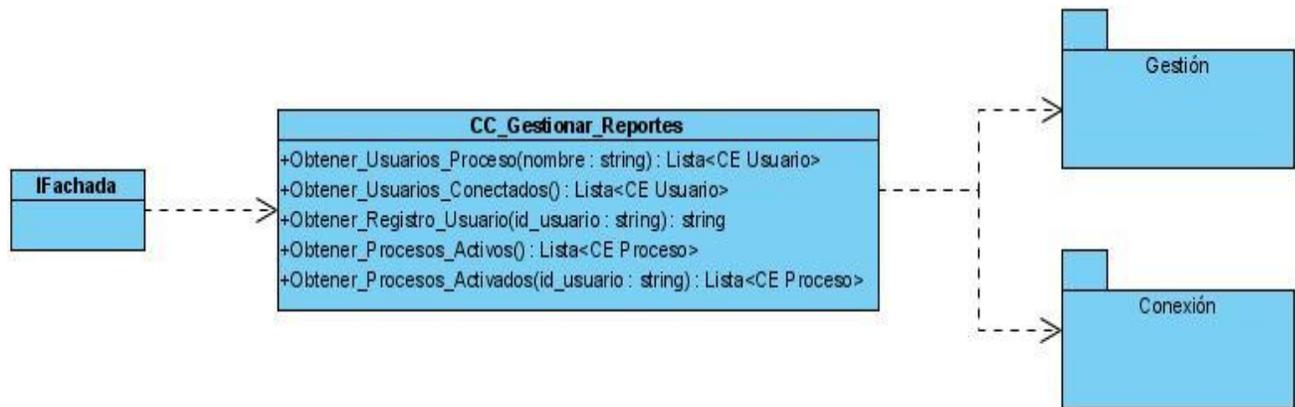


Figura 19: Subsistema de Reportes

### 3.2.4 Realización de Casos de Uso

Una realización de caso de uso-diseño es una colaboración en el modelo de diseño que describe cómo se realiza un caso de uso específico, y cómo se ejecuta, en términos de clases de diseño y sus objetos. Proporciona una traza directa a una realización de casos de uso-análisis en el modelo de análisis. (30)

A continuación se presentan los diagramas de clases del diseño correspondientes a los casos de usos no críticos del NVP.

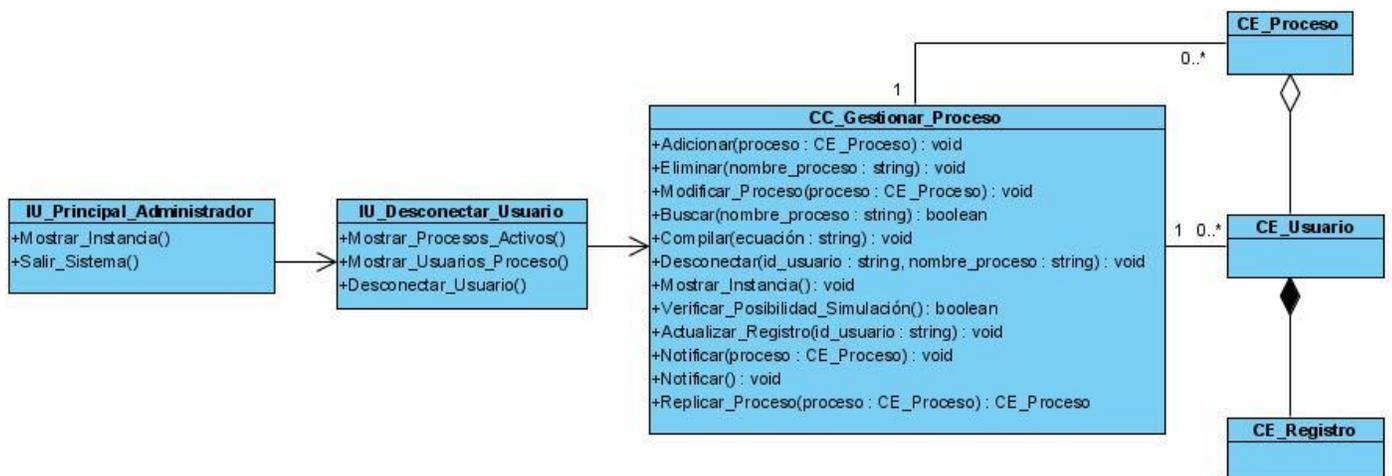


Figura 20: Diagrama de clases del Diseño. CU Desconectar\_Usuario\_Proceso

Ver en el Capítulo 2 la descripción del CU Desconectar\_Usuario\_Proceso.

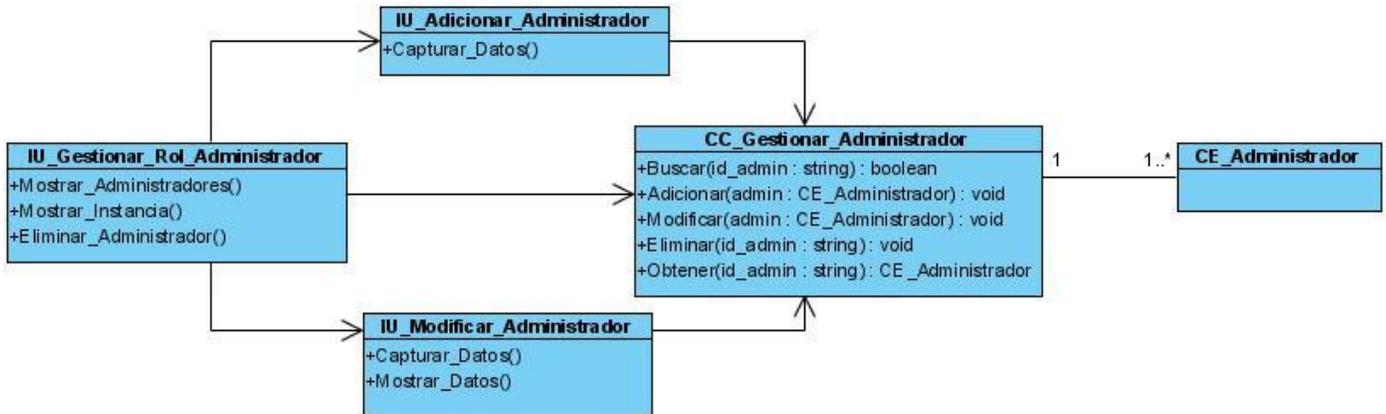


Figura 21: Diagrama de clases del Diseño. CU Gestionar\_Rol\_Administrador

Ver en el Capítulo 2 la descripción del CU Gestionar\_Rol\_Administrador.

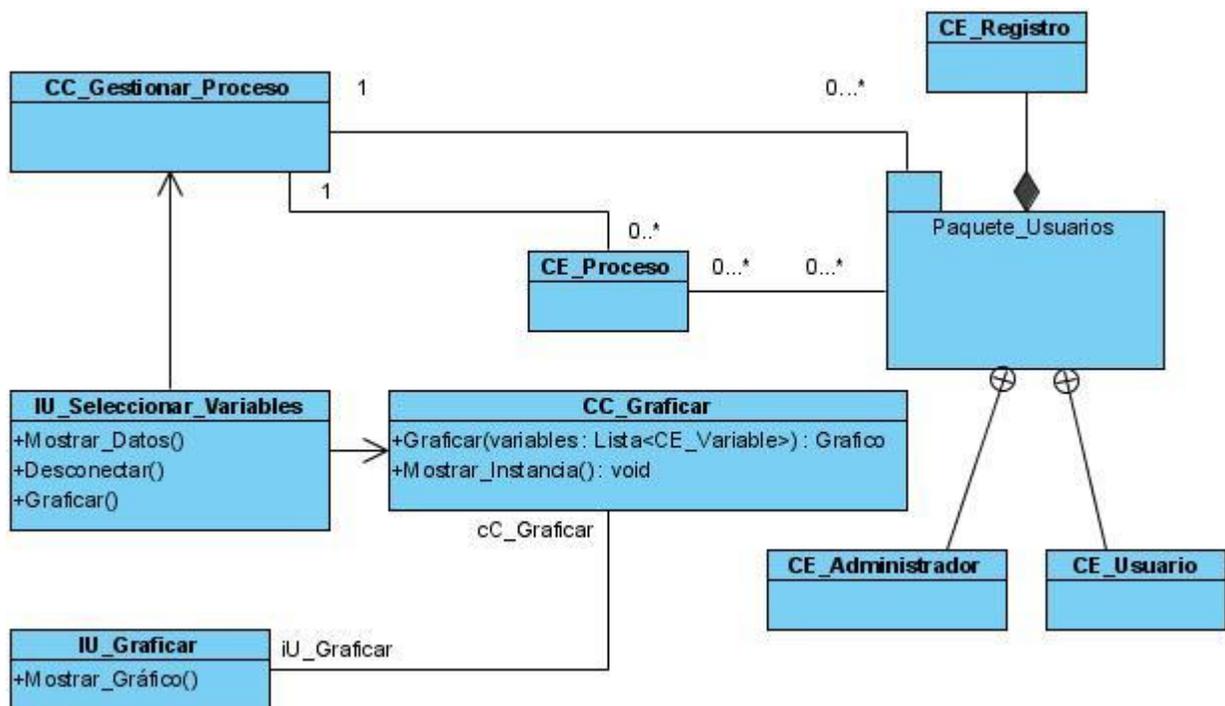


Figura 22: Diagrama de clases del Diseño. CU Graficar

Ver en el Capítulo 2 la descripción del CU Graficar.

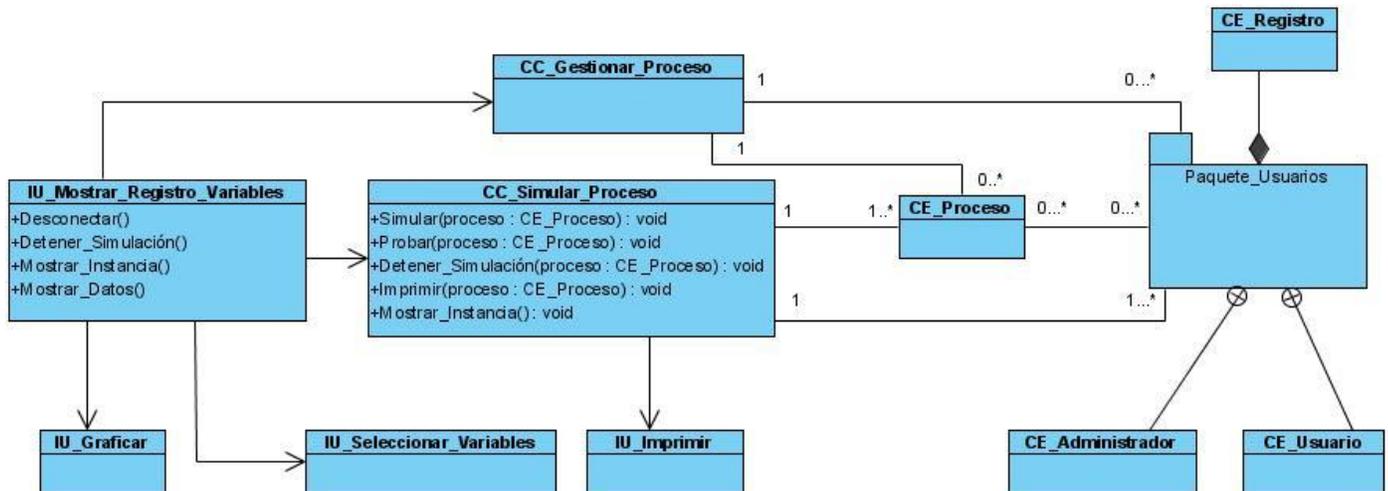


Figura 23: Diagrama de clases del Diseño. CU Mostrar\_Registro\_Variables

Este diagrama de clases recoge además las funcionalidades de los casos de uso Simular y Probar\_Aplicación debido a que ambos se traducen a métodos de una clase. Ver en el Capítulo 2 las descripciones de los casos de uso Mostrar\_Registro\_Variables, Simular y Probar\_Aplicación.

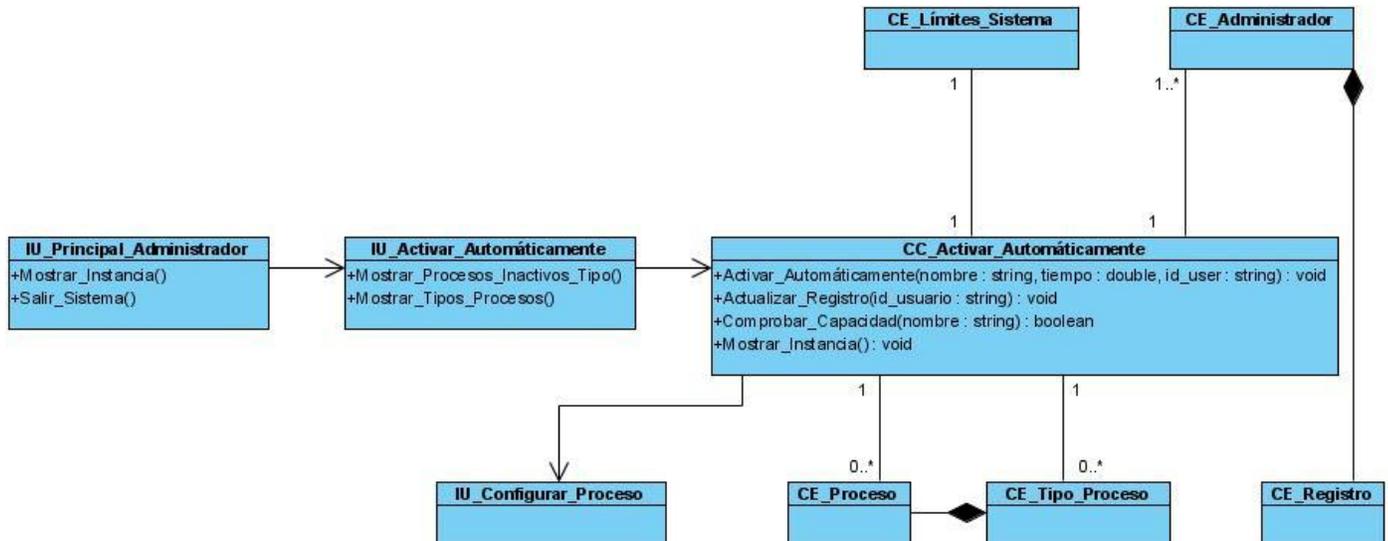


Figura 24: Diagrama de clases del Diseño. CU Activar\_Automáticamente\_Procesos

Ver en el Capítulo 2 la descripción del CU Activar\_Automáticamente\_Procesos.

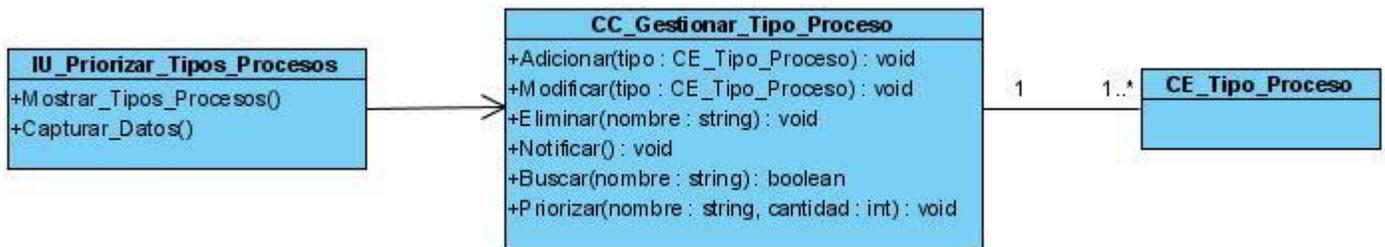


Figura 25: Diagrama de clases del Diseño. CU Priorizar\_Número\_Procesos

Ver en el Capítulo 2 la descripción del CU Priorizar\_Número\_Procesos.

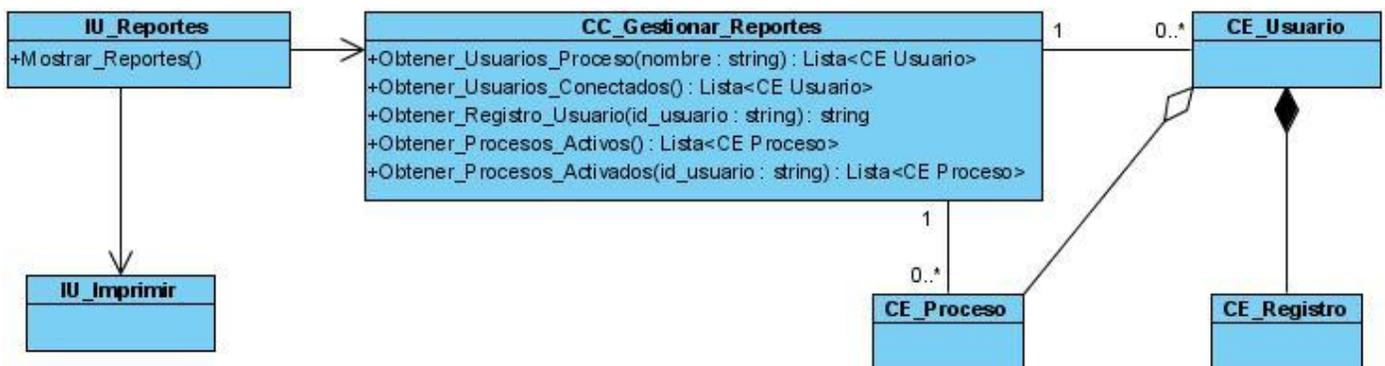


Figura 26: Diagrama de clases del Diseño. Casos de Uso Asociados a Reportes (CU Mostrar\_Registro\_Usuario, CU Mostrar\_Procesos\_Activos, CU Mostrar\_Usuarios\_Conectados, CU Mostrar\_Usuarios\_Proceso, CU Mostrar\_Procesos\_Activos\_Usuario)

Este diagrama de clases recoge las funcionalidades varios casos de uso por lo que se recomienda ver en el Capítulo 2 las descripciones de los casos de uso Mostrar\_Registro\_Usuario, Mostrar\_Procesos\_Activos, Mostrar\_Usuarios\_Conectados, Mostrar\_Usuarios\_Proceso, Mostrar\_Procesos\_Activos\_Usuario.

Consultar los diagramas de clases del diseño de los casos de uso críticos:

CU Gestionar\_Conexión\_Sistema: (Ver Anexos 56).

CU Gestionar\_Conexión\_Proceso: (Ver Anexos 57).

CU Establecer\_Límites: (Ver Anexos 58).

CU Gestionar\_Tipo\_Proceso: (Ver Anexos 59).

CU Gestionar\_Modelo\_Proceso: (Ver Anexos 60).

CU Configurar\_Proceso: (Ver Anexos 61).

CU Gestionar\_Controlador: (Ver Anexos 62).

Para la realización de los casos de uso se escogieron además los diagramas de secuencia como instrumento para lograr un mayor nivel de detalle en el diseño planteado. Estos diagramas funcionan de forma parecida a los de colaboración pero permiten reflejar de forma más sencilla la dinámica de cada caso de uso, así como los detalles de cada objeto participante en la interacción y sus funcionalidades.

A continuación se presentan los diagramas de secuencia correspondientes a los casos de uso no críticos Mostrar\_Registro\_Variables, Activar\_Automáticamente\_Procesos, Priorizar\_Número\_Procesos y Mostrar\_Registro\_Usuario. (Consultar los diagramas de los restantes casos de uso en los anexos desde el 63 hasta el 76).

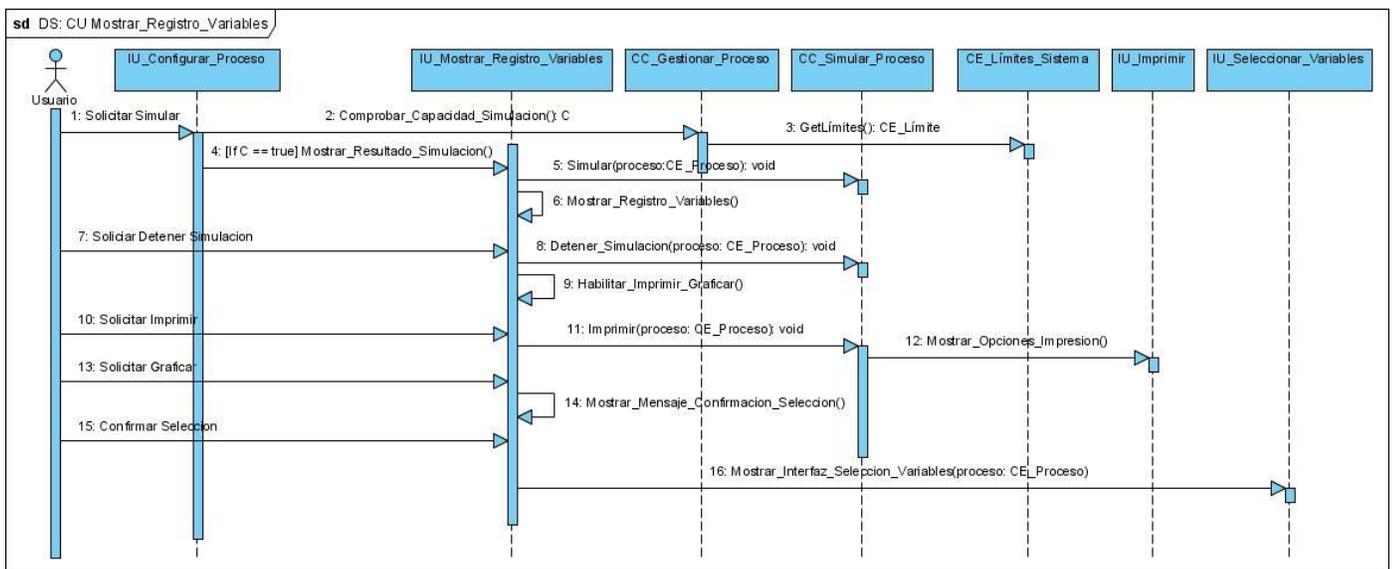


Figura 27: Diagrama de Secuencia. CU Mostrar\_Registro\_Variables

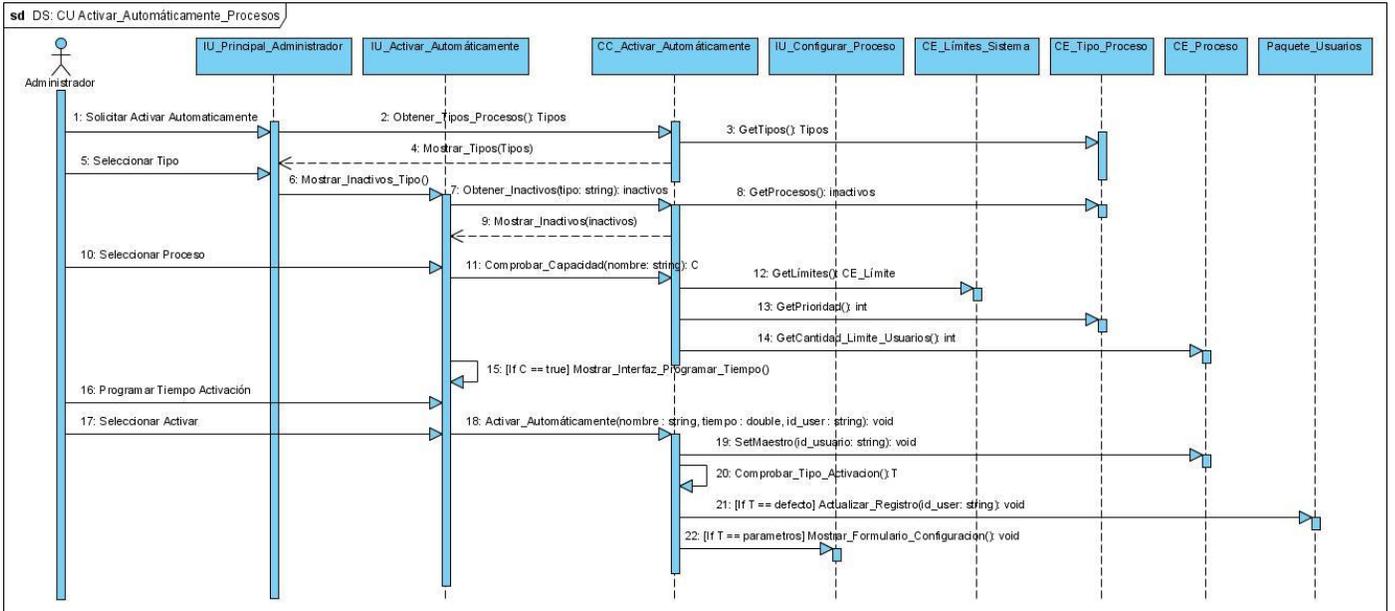


Figura 28: Diagrama de Secuencia. CU Activar\_Automáticamente\_Procesos

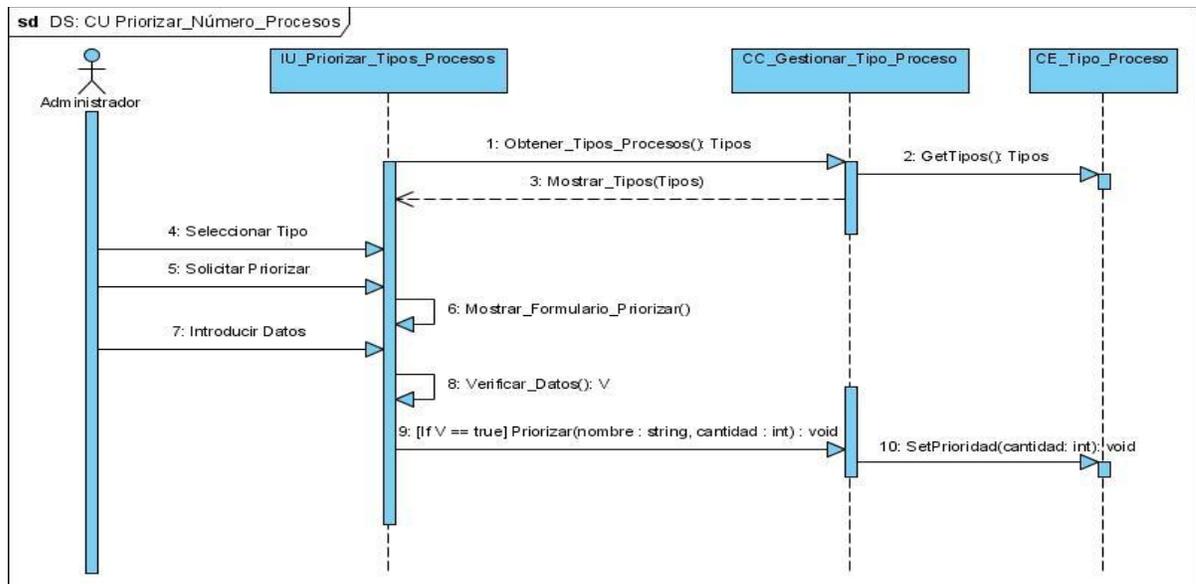


Figura 29: Diagrama de Secuencia. CU Priorizar\_Número\_Procesos

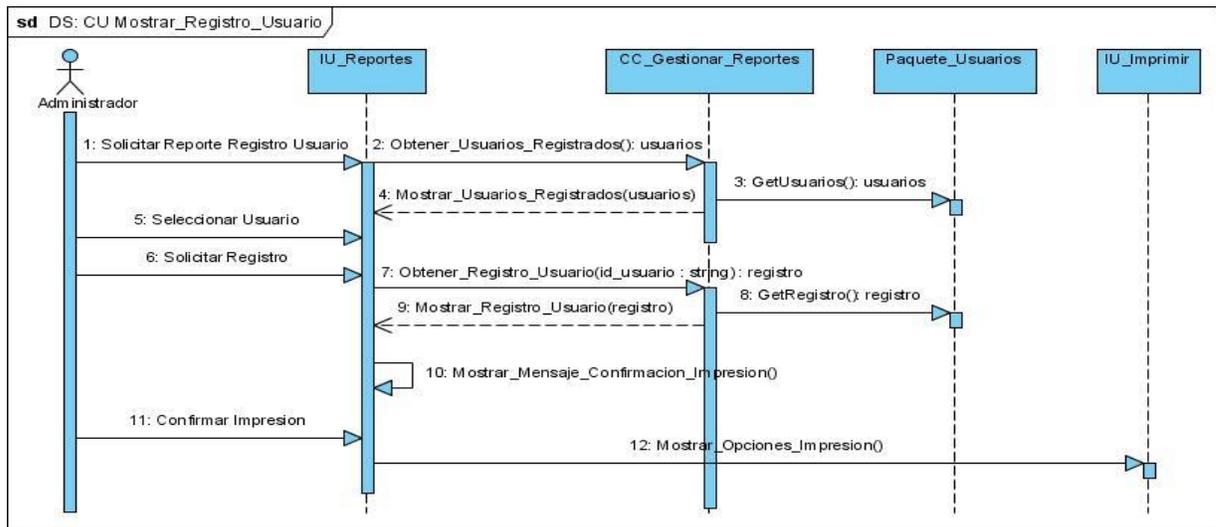


Figura 30: Diagrama de Secuencia. CU Mostrar\_Registro\_Usuario

## Conclusiones

Luego de tratar aspectos referentes al análisis y diseño del sistema, y obtener los principales artefactos de este flujo se puede concluir que:

- La realización del análisis de forma detallada facilitó la entrada al diseño.
- El uso de patrones de diseño y arquitectónicos permitió obtener un diseño flexible y de mayor calidad.
- La estructuración del diseño en subsistemas contribuirá a la organización del trabajo de los desarrolladores y a entender mejor el funcionamiento del futuro sistema.
- Las realizaciones de los casos de uso aportarán la información necesaria para el proceso de implementación.

## Capítulo 4: Validación de los Resultados.

### Introducción

Un factor fundamental para efectuar la evaluación de los productos y procesos de software en los diferentes dominios de aplicación lo constituyen las métricas, las cuales abarcan un gran escenario en cuanto a medidas de software computacional se refieren. Las métricas suelen ser aplicadas a muchas organizaciones, procesos y productos que estén relacionados y afecten a la estimación de costo.

Como se conoce la medición es muy común en el mundo de software y las mismas pueden englobarse en dos categorías principales: medidas directas y medidas indirectas. En las primeras está reflejado el costo y el esfuerzo aplicado, las líneas de código producidas, velocidad de ejecución, el tamaño de memoria, así como los defectos observados en un período de tiempo determinado. Las segundas abarcan la funcionalidad, calidad, complejidad, eficiencia, fiabilidad, facilidad de mantenimiento de un software, etc. (12)

A raíz del trabajo realizado para el análisis y diseño del NVP se obtuvieron una gran cantidad de artefactos, por parte del equipo de analistas. Para comprobar la efectividad de las actividades y validar los artefactos se empleará la Métrica de la Calidad de Especificación de los Requisitos, que propone Pressman, las Métricas de Diseño Arquitectónico propuestas por Card y Glass para calcular la complejidad de los módulos o subsistemas y las Métricas Orientadas a Clases.

#### 4.1 Métrica de la Calidad de Especificación de los Requisitos.

Para realizar la validación de los requisitos existe toda una lista de características que sugieren el uso de una o más métricas como son: especificidad (ausencia de ambigüedad), corrección, compleción, comprensión, capacidad de verificación, consistencia externa e interna, capacidad de logro, concisión, trazabilidad, capacidad de modificación, exactitud y capacidad de reutilización (40). Debido a la cantidad de requisitos capturados y las características de los mismos, se va a aplicar la métrica que mide su especificidad, haciendo que la parte interesada pueda entenderlos de una manera fácil y se puedan probar.

Para llevar a cabo este proceso se tiene que:  $n_r$  representa el número de requisitos del sistema:

$$n_r = n_f + n_{nf}$$

Donde  $n_f$  es el número de requisitos funcionales y  $n_{nf}$  es el número de requisitos no funcionales.

Luego se procede a medir la especificidad de los requisitos con la siguiente fórmula:

$$Q = n_{ui} / n_f$$

Donde  $n_{ui}$  es el número de requisitos para los que todos los revisores tuvieron interpretaciones idénticas. El valor de Q a medida que se acerca a uno, se va disminuyendo la ambigüedad de la especificación. (12)

Para realizar el proceso de medición anteriormente planteado y así proceder a la evaluación se llevaron a cabo dos revisiones y se consultaron tres revisores, con el objetivo de eliminar ambigüedades y especificar detalladamente los requisitos capturados. Luego del refinamiento se capturaron cuarenta y cinco requisitos funcionales y veinte y cinco requisitos no funcionales, setenta en total. Los resultados de las revisiones se muestran a continuación:

#### **Primera Revisión:**

Algunos requisitos funcionales presentaban ambigüedades y no estaban descritos correctamente. Representaban el 0.07% con ambigüedad.

De 70, los revisores comprendieron de la misma forma 65.

$$Q = 65 / 70$$

$$Q = 0.93$$

#### **Segunda Revisión:**

Luego de eliminar los problemas de comprensión detectados en la revisión anterior, elevar el nivel de especificidad y consultar nuevamente a los revisores, se redujo el nivel de ambigüedad a un 0.01%, por lo que casi la totalidad de los requisitos fueron interpretados de la misma forma.

$$Q = 69 / 70$$

$$Q = 0.99$$

Después de haber sido aplicada la métrica de especificación de los requisitos durante dos revisiones, se puede concluir que se logró un grado bajo de ambigüedad de forma general, por lo tanto, el nivel de calidad del proceso de refinamiento de requisitos fue bastante alto. Realizar este tipo de evaluaciones

contribuyó a obtener un análisis y un diseño en coherencia con lo que desea la parte interesada, desde el punto de vista de las funcionalidades y restricciones con las que debe contar el NVP.

#### 4.2 Complejidad de los Subsistemas.

Son consideradas métricas de alto nivel. Miden la eficiencia de la arquitectura y los módulos que conforman el sistema. No se necesita conocer el funcionamiento interno de los módulos para evaluarlos, por ello se consideran como métricas de caja negra.

Card y Glass definen tres medidas de la complejidad del diseño del software: complejidad estructural, complejidad de datos y complejidad del sistema. (40)

- La complejidad estructural, definida como:  $S(i) = f^2 out(i)$ , donde  $i$  es el módulo y  $f^2 out(i)$  es la expansión de  $i$ , lo que quiere decir que  $f_{out}$  indica el número de módulos que son invocados directamente por el módulo  $i$ .
- La complejidad de datos proporciona una indicación de la complejidad en la interfaz interna de un módulo, se define como:  $D(i) = v(i) / [f_{out}(i) + 1]$ , donde  $v(i)$  es el número de variables de entrada del módulo  $i$ .
- La complejidad del sistema se define como la suma de las complejidades estructural y de datos, se define como:  $C(i) = S(i) + D(i)$ .

Los valores de complejidad que se obtienen al aplicar esta métrica tienen gran significado para la arquitectura. Si aumentan los valores, entonces la complejidad arquitectónica del sistema también aumenta, provocando que aumente el esfuerzo para la integración y las pruebas.

A continuación se calcula la complejidad de los cuatro subsistemas principales del sistema:

#### Subsistema Conexión

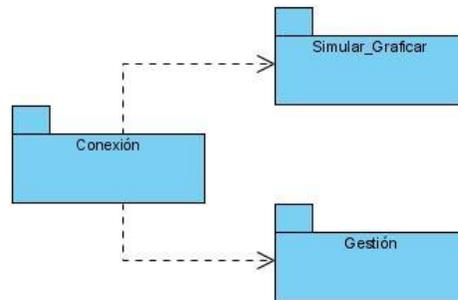


Figura 31: Expansión del Subsistema Conexión

Tabla 1: Complejidad del Subsistema Conexión

Complejidad Estructural $S(i)$	Complejidad de Datos $D(i)$	Complejidad del Sistema $C(i)$
$S(i) = 2^2 = 4$	$D(i) = 2 / [4 + 1] = 0.4$	$C(i) = 4 + 0.4 = 4.4$

**Subsistema Gestión**

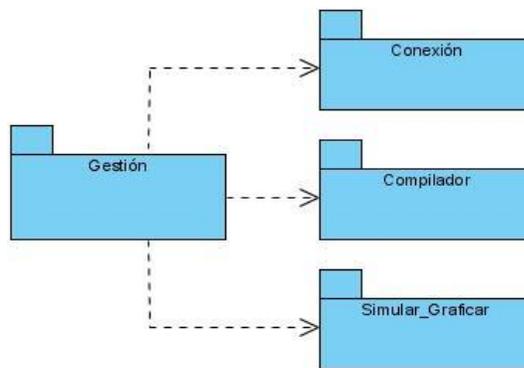
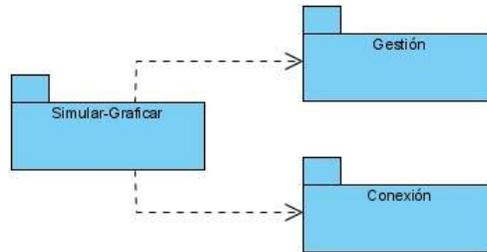


Figura 32: Expansión del Subsistema Gestión

Tabla 2: Complejidad del Subsistema Gestión

Complejidad Estructural $S(i)$	Complejidad de Datos $D(i)$	Complejidad del Sistema $C(i)$
$S(i) = 3^2 = 9$	$D(i) = 3 / [9 + 1] = 0.9$	$C(i) = 9 + 0.9 = 9.9$

**Subsistema Simular-Graficar**

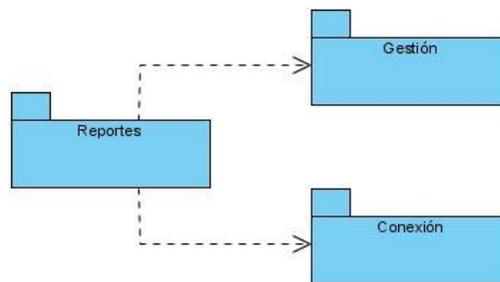


**Figura 33: Expansión del Subsistema Simular-Graficar**

**Tabla 3: Complejidad del Subsistema Simular-Graficar**

Complejidad Estructural $S(i)$	Complejidad de Datos $D(i)$	Complejidad del Sistema $C(i)$
$S(i) = 2^2 = 4$	$D(i) = 2 / [4 + 1] = 0.4$	$C(i) = 4 + 0.4 = 4.4$

**Subsistema Reportes**



**Figura 34: Expansión del Subsistema Reportes**

**Tabla 4: Complejidad del Subsistema Reportes**

Complejidad Estructural $S(i)$	Complejidad de Datos $D(i)$	Complejidad del Sistema $C(i)$
$S(i) = 2^2 = 4$	$D(i) = 2 / [4 + 1] = 0.4$	$C(i) = 4 + 0.4 = 4.4$

**Resultados****Tabla 5: Resultados**

<b>Conexión</b>	<b>Gestión</b>	<b>Simular-Graficar</b>	<b>Reportes</b>
4.4	9.9	4.4	4.4

Es necesario mencionar que la mayoría de los autores que tratan el tema de las métricas de diseño arquitectónico proponen tres tipos de sistema según su complejidad:

- No muy complejo.
- Complejo.
- Muy complejo.

Para saber cuando un sistema está dentro de alguno de los tres grupos se proponen umbrales. Para los no muy complejos tiene que cumplirse que  $D(i) \leq 7$  y  $S(i) \leq 32$ . (40)

A partir de los resultados obtenidos se puede concluir que los subsistemas Conexión, Gestión, Simular-Graficar y Reportes no son muy complejos. Los valores de la complejidad en ellos son muy bajos señalando que Gestión es el de mayor complejidad de datos, debido a la cantidad de variables de entrada que presenta. Por tanto, el NVP no es un sistema complejo.

**4.3 Métricas Orientadas a Clases.**

Teniendo en cuenta que para el equipo de desarrollo es importante conocer la complejidad del proceso de implementación y las características del diseño desde el punto de vista del nivel de dependencia entre las clases, la cantidad de pruebas que será necesario realizar, la flexibilidad del código y qué clases podrán ser reutilizables; se aplicarán métricas para evaluar el tamaño de las clases y las relaciones entre clases.

**4.3.1 Tamaño de Clase (TC).**

Para medir el tamaño de clase se tienen en cuenta los siguientes aspectos:

- Total de operaciones, ya sean las propias o las heredadas de las clases padres e interfaces que implementen.
- Cantidad de atributos, tanto los de ella, como lo de los padres.
- Promedio general de los dos anteriores para el sistema completo.

Para evaluar las métricas son necesarios los umbrales. En este caso las clases se clasifican en tres grupos según su tamaño, los que se representan en la siguiente tabla junto con los umbrales seleccionados para su clasificación. (12)

**Tabla 6: Valores de los umbrales para TC**

Clasificación	Valores de los umbrales
Pequeño	$\leq 20$
Medio	$> 20$ y $\leq 30$
Grande	$> 30$

**Tabla 7: Tamaño de las Clases**

No.	Nombre	Cantidad Atributos	Cantidad Operaciones	Tamaño
1	CC_Gestionar_Conexión_Sistema	2	9	Pequeño
2	CE_Usuario	8	9	Pequeño
3	CE_Administrador	9	10	Pequeño
4	CE_Registro	2	2	Pequeño
5	CE_Límites_Sistema	2	3	Pequeño
6	CC_Gestionar_Conexión_Proceso	4	5	Pequeño

7	CE_Proceso	12	14	Medio
8	CE_Tipo_Proceso	5	6	Pequeño
9	CE_Controlador	2	3	Pequeño
10	CE_Controlador Predefinido	5	6	Pequeño
11	CE_Controlador Físico	4	5	Pequeño
12	CE_Protocolo	1	2	Pequeño
13	CE_DDE	2	3	Pequeño
14	CE_DCOM	2	3	Pequeño
15	CE_OPC	2	3	Pequeño
16	CE_Variable	2	3	Pequeño
17	CC_Gestionar_Proceso	2	13	Pequeño
18	CC_Gestionar_Administrador	1	6	Pequeño
19	CC_Gestionar_Límites	3	4	Pequeño
20	CC_Gestionar_Tipo_Proceso	2	7	Pequeño
21	CC_Graficar	0	2	Pequeño
22	CC_Simular_Proceso	2	6	Pequeño
23	CC_Gestionar_Controlador	2	6	Pequeño
24	CC_Activar_Automáticamente	4	5	Pequeño

25	CC_Gestionar_Reportes	2	6	Pequeño
----	-----------------------	---	---	---------

Cuando existe un TC grande se afectan los parámetros de calidad definidos por esta métrica. Se reduce la reutilización de las clases, la implementación se hace más compleja, las pruebas son difíciles de realizar y aumenta la responsabilidad de las clases. (12)

Casi todas de las clases que conforman el diseño que será la entrada para el flujo de trabajo de implementación están dentro de la categoría de pequeñas, lo que demuestra que el proceso de construcción del NVP no será complejo. Los resultados obtenidos son positivos según esta métrica, como se puede ver en las siguientes tablas.

**Tabla 8: Cantidad de clases por clasificación**

Clasificación	Cantidad Clases
Pequeño	24
Medio	1
Grande	0

**Tabla 9: Resultados de la Métrica TC**

Cantidad Clases	Promedio Atributos	Promedio Operaciones
25	3.20	5.64

#### 4.3.2 Relaciones entre Clases (RC).

Esta métrica está dada por la cantidad de relaciones de uso que existe entre las distintas clases que forman el diseño propuesto. Se aplica a las mismas clases en las que fue aplicada la métrica TC. Los

aspectos de calidad que se miden son: Acoplamiento, Complejidad de mantenimiento, Reutilización y Cantidad de pruebas. (12)

**Tabla 10: Cantidad de relaciones de uso entre las clases**

No.	Nombre	Relaciones de Uso
1	CC_Gestionar_Conexión_Sistema	2
2	CE_Usuario	1
3	CE_Administrador	2
4	CE_Registro	0
5	CE_Límites_Sistema	0
6	CC_Gestionar_Conexión_Proceso	4
7	CE_Proceso	5
8	CE_Tipo_Proceso	1
9	CE_Controlador	1
10	CE_Controlador Predefinido	2
11	CE_Controlador Físico	2
12	CE_Protocolo	0
13	CE_DDE	1
14	CE_DCOM	1

15	CE_OPC	1
16	CE_Variable	0
17	CC_Gestionar_Proceso	3
18	CC_Gestionar_Administrador	1
19	CC_Gestionar_Límites	3
20	CC_Gestionar_Tipo_Proceso	2
21	CC_Graficar	0
22	CC_Simular_Proceso	2
23	CC_Gestionar_Controlador	2
24	CC_Activar_Automáticamente	4
25	CC_Gestionar_Reportes	2

Para medir el acoplamiento según los resultados de esta métrica, se plantean los siguientes valores.

**Tabla 11: Acoplamiento**

<b>Categoría</b>	<b>Relaciones de uso</b>	<b>Cantidad de Clases</b>
Ninguno	0	5
Bajo	1	7
Medio	2	8

Alto	>2	5
------	----	---

Los otros parámetros de calidad que mide esta métrica dependen del valor promedio de las relaciones de uso de todas las clases, en el caso del diseño del NVP dicho promedio es de 1.68.

**Tabla 12: Cantidad de Pruebas y Complejidad de Mantenimiento**

Categoría	Criterio	Cantidad de Clases
Baja	$\leq$ Prom.	12
Media	$>$ Prom. Y $\leq 2$ *Prom.	10
Alta	$> 2$ *Prom.	3

**Tabla 13: Reutilización**

Categoría	Criterio	Cantidad de Clases
Baja	$> 2$ *Prom.	3
Media	$>$ Prom. Y $\leq 2$ *Prom.	10
Alta	$\leq$ Prom.	12

Resumiendo el uso de esta métrica, los resultados pueden calificarse como positivos. El acoplamiento existente entre las clases es moderado, es decir, el necesario para contribuir a una alta cohesión, el 20% no tiene ningún acoplamiento, el 28% es bajo, el 32% es medio y sólo el 20% tiene alto acoplamiento. El nivel de reutilización de las clases es bastante bueno, el 48% de las clases pueden ser altamente reutilizables. Sucede de la misma manera con la cantidad de pruebas y la complejidad de mantenimiento, el 48% de las clases son fáciles de reparar y la cantidad de pruebas a realizar es relativamente pequeña.

## Conclusiones

Luego de aplicar las métricas para evaluar la calidad de la especificación de los requisitos, el diseño arquitectónico y la complejidad de los módulos o subsistemas se concluye que:

- El nivel de ambigüedad en los requisitos disminuyó, lográndose el nivel de especificidad requerido y el entendimiento casi total por parte de los revisores.
- El subsistema más complejo es el de Gestión teniendo en cuenta la complejidad de los datos, por lo que éste requerirá mayor esfuerzo y recursos para su construcción.
- El 96% de las clases que conforman el diseño están dentro de la categoría de pequeñas, lo que demuestra que el proceso de construcción y pruebas del NVP no será complejo.
- El 80% de las clases presenta clasificaciones menores o iguales al medio acoplamiento, contribuyendo a una alta cohesión. El 48% de las clases pueden ser altamente reutilizables.
- Las métricas aplicadas a los artefactos obtenidos validan de forma general la calidad del análisis y diseño del NVP.

### Conclusiones

- El estudio de las funcionalidades previstas durante la primera iteración para el NVP, en otros sistemas en el mundo, contribuyó a elevar el nivel de comprensión de los analistas y a redefinir los objetivos y funcionalidades necesarias para el sistema a construir.
- El refinamiento de requisitos permitió identificar nuevas funcionalidades, la posterior reestructuración del diagrama de casos de uso del sistema y un mayor nivel de detalle en las descripciones de éstos.
- Se completó la arquitectura del NVP mediante el uso de patrones GRASP, arquitectónicos y de diseño, para proveer a los desarrolladores de una entrada que permita la eficiente implementación del sistema.
- La aplicación del prototipado de interfaces y de métricas propuestas por varios autores permitió validar los resultados obtenidos durante el análisis y diseño del sistema.
- Se cumplió el objetivo general trazado para este Trabajo de Diploma: Realizar el análisis y diseño de un Nodo Virtual que simule procesos industriales.

### Recomendaciones

- Incluir para futuras versiones del NVP la simulación con el uso de controladores físicos, que soporten varios protocolos de comunicación.
- Añadir un Módulo de Gestión del Modelo Gráfico, que requiere de un número mayor de recursos humanos y de cómputo para su realización, por lo que no se incluyó en esta iteración.

## Bibliografía

### Referencias

1. **Capote Abreu, J, y otros.** Algunos conceptos y Definiciones del Modelado y Simulación Computacional de Incendios. [En línea] 2006.  
<http://grupos.unican.es/gidai/Algunos%20Conceptos%20y%20Definiciones%20de%20la%20Modelaci%C3%B3n%20y%20Simulaci%C3%B3n%20Computacional%20de%20Incendios.pdf>.
2. **Dormido, S., y otros.** *Estructura y tecnología de computadores II*. HAYES, J. P. : Sanz y Torres, 1994. pág. 538.
3. **Parra, Rafael Alberto Moreno.** Docencia 50 webs. [En línea] 2007.  
<http://docencia.50webs.com/simula01.htm>.
4. **Maisel, H. y Gnugnoli, G.** *Simulation of Discrete Stochastic Systems*. Chicago, Illinois : Science Research Associates, 1972. pág. 465.
5. **Shannon, Robert E.** *Introduction to Simulation*. 1992.
6. **Maier, S y Herrscher, K.** On node virtualization for scalable Network Emulation. [En línea] 2003.  
<http://www.ipvs.uni-stuttgart.de/abteilungen/vs/abteilung/mitarbeiter/eigenes/maiersn/start/maier05node.pdf>.
7. **Hosseinzaman, A. y Bargiela, A.** *ADA's Virtual Node based Water System*. [En línea] 1995.  
<http://www.intelligentmodelling.org.uk/Papers/rttg-publ10.pdf>.
8. **Wu, Y.** *A virtual node method for treatment of wells in modeling multiphase Flow in reservoirs*. 1999.
9. **Molino, Neil, Bao, Z. y R. Fedkiw, A.** Virtual Node Algorithm for Changing Mesh Topology During Simulation. [En línea] 2004. <http://citeseer.ist.psu.edu/666248.html>.
10. **Miyachi, Toshiyuki y Chinen, Ken-ichi.** *StarBED and SprinOS: Large scale general purpose network testbed and supporting software*. 2006.
11. **Batista, Ing. Yalice Gámez.** "Herramienta interactiva para la enseñanza en la carrera de Ingeniería Automática: Nodo Virtual de Procesos.". Instituto Superior Politécnico "José Antonio Echeverría". Habana : s.n., 2008. pág. preguntar a Yalice.

12. **Edith, Maylen y Ortiz, Leydis A.** *Análisis y Diseño de un Nodo Virtual de Procesos*. Universidad de las Ciencias Informáticas. Habana : s.n., 2008. pág. 169.
13. **De Sapiensman, Technical English - Spanish Vocabulary.** Fundamentos del Control Automático Industrial. [En línea] 2008. [http://www.sapiensman.com/control\\_automtico/](http://www.sapiensman.com/control_automtico/).
14. **Cantarero, Teodoro Álamo.** Diseño del Controlador PID. [En línea] 2005. [http://www.esi2.us.es/~alamo/Archivos/Certificaciones/Sec\\_9\\_Otras\\_Publicaciones/Docente/AlamoPIDTotal.pdf](http://www.esi2.us.es/~alamo/Archivos/Certificaciones/Sec_9_Otras_Publicaciones/Docente/AlamoPIDTotal.pdf).
15. **Zargham, Mehdi R.** *Computer Architecture Single and Parallel Systems*. Prentice Hall. USA : Madrid, 1995. págs. 371-389.
16. **Hernández Bello, Irmig Isaac y Ochoa Luna, Cristóbal.** Control difuso y construcción de un mecanismo capaz de golpear con distintos efectos una bola de billar. [En línea] 2004. [http://catarina.udlap.mx/u\\_dl\\_a/tales/documentos/lep/hernandez\\_b\\_ii/capitulo4.pdf](http://catarina.udlap.mx/u_dl_a/tales/documentos/lep/hernandez_b_ii/capitulo4.pdf).
17. **Sánchez, Yunier.** Control difuso. [En línea] 2007. <http://prof.usb.ve/montbrun/ps2320/fuzzy/fuzzy.html>.
18. **De Wikipedia, la enciclopedia libre.** Protocolos de Comunicación. [En línea] 2009. [http://es.wikipedia.org/wiki/Protocolo\\_de\\_red](http://es.wikipedia.org/wiki/Protocolo_de_red).
19. **MasterMagazine.** Definición de Driver - Significado y definición de Driver. [En línea] 2004. <http://www.mastermagazine.info/termino/4756.php>.
20. **De Wikipedia, la enciclopedia libre.** OPC. [En línea] 2009. <http://d.scribd.com/docs/16cl9tobtwd3am74kclt.pdf>.
21. **Barberán, Manuel.** *Funcionamiento del Protocolo TCP-IP*. 1998.
22. *Cubriendo un hueco en el mercado de buses de campo*. **Tolfo, Flavio.** 2002, ABB, Vol. 1, págs. 30-34.
23. **Herrera, Yamile Ramirez.** Componentes. [En línea] 1997. <http://www.monografias.com/trabajos16/componentes/componentes.shtml>.
24. **Petzold, Charles.** *Programación en Windows 95*. Madrid, España : McGraw-Hill, 1996. pág. 399.
25. **López Barrio, C.** Metodología de Desarrollo: Programación Extrema. [En línea] 2005. [http://www-lsi.die.upm.es/~carreras/ISSE/programacion\\_extrema\\_1.x2.pdf](http://www-lsi.die.upm.es/~carreras/ISSE/programacion_extrema_1.x2.pdf).

26. **Gómez, J.** ¿Metodología?... Sí, pero, ¿cuál? [En línea] 2006.  
<http://www.versioncero.com/articulo/469/metodologias-i-pero-cual>.
27. **Barbone, Víctor A. González.** XP: Extreme Programming (Programación Extrema. [En línea] 2006.  
<http://iie.fing.edu.uy/~nacho/blandos/seminario/XProg1.html>.
28. **Furini, Gastón.** Ingeniería del software - Nivel Intermedio. [En línea] 2008.  
<http://www.clubdesarrolladores.com/articulos/mostrar/63-metodologia-scrum>.
29. **Serrano, Jorge.** Explicando Scrum a mi abuela. [En línea] 2007.  
<http://www.pabloazorin.com.ar/index.php/2008/07/18/explicando-scrum-a-mi-abuela/>.
30. **Jacobson, I, Booch, G y Rumbaugh, J.** *El Proceso Unificado de Desarrollo de Software*. La Habana : Félix Varela, 2004. pág. 435.
31. **Giraldo, Z.** *Herramientas de Ingeniería de software*. 2003.
32. **De Wikipedia, la enciclopedia libre.** ArgoUML. [En línea] 2009. <http://es.wikipedia.org/wiki/ArgoUML>.
33. **Martín Martínez, Gema y Pareja León, Ángel.** HERRAMIENTAS CASE:ARGO UML. [En línea] 2006.  
<http://kybele.escet.urjc.es/documentos/HC/Exposiciones/ArgoUML.pdf>.
34. **De Wikipedia, la enciclopedia libre.** AndroMDA. [En línea] 2009.  
<http://es.wikipedia.org/wiki/AndroMDA>.
35. **Grupo Soluciones Innova.S.A. (GSI).** Rational Rose Enterprise. [En línea] 2007.  
<http://www.rational.com.ar/herramientas/roseenterprise.html>.
36. **Sixto.** ¿Patrones? [En línea] 2003. <http://www.code4net.com/archives/000030.html>.
37. **Visual Paradigm.** 10 Reasons to Choose Visual Paradigm. [En línea] 2008. <http://www.visual-paradigm.com/aboutus/10reasons.jsp>.
38. **Lucas Rodríguez F., Cabello Quintero A.** *Simulación de procesos de información y criptografía cuántica*. 2005.
39. **Geetanjali A., Balasubramaniam A.** *Programación Microsoft C#*. La Habana : Félix Varela, 2007. pág. poner.

40. **Pressman, R.** *Ingeniería del Software. Un enfoque práctico*. La Habana : Félix Varela, 2005. pág. 601.
41. **Koch, Nora y María, José.** *Ingeniería de Requisitos en Aplicaciones para la Web– Un estudio comparativo*. España : Universidad de Sevilla Lenguajes y Sistemas Informáticos, 2002. pág. 26.
42. *Ingeniería de Requerimientos una guía para extraer, analizar, especificar y validar los requerimientos de un proyecto*. **Dávila, Nicolás.** 2001.
43. **Zapata, Carlos y Arango, J.** Alineación entre Metas Organizacionales y Elicitación de Requisitos del Software. [En línea] 2004.  
<http://www2.unalmed.edu.co/dyna2005/143/ALINEACI%D3N%20ENTRE%20METAS.pdf>.
44. **Raghavan, Sridhar, Zelesnik, Gregory y Ford, Gary.** Lecture Notes on Requirements Elicitation. [En línea] 1994. <ftp://ftp.sei.cmu.edu/pub/education/94em010.pdf>.
45. **Robertson, Suzanne y Robertson, James.** *Mastering the Requirements Process*. 2. Portland, Oregon : Addison-Wesley Professional, 2006. pág. 592 .
46. **Fowler, M.** *Analysis Patterns, Reusable Object Models*. Portland, Oregon : Addison-Wesley Professional, 1996. pág. 384.
47. **Sánchez, Pedro Herrarte.** Consultar datos\_ SELECT - Tutorial SQL. [En línea] 2005.  
[http://www.devjoker.com/asp/ver\\_contenidos.aspx?co\\_conte](http://www.devjoker.com/asp/ver_contenidos.aspx?co_conte).
48. **De Wikipedia, la enciclopedia libre.** PostgreSQL. [En línea] 2009.  
<http://es.wikipedia.org/wiki/PostgreSQL>.

### Glosario de Términos

**Bus:** Representa básicamente una serie de cables mediante los cuales pueden cargarse datos en la memoria y desde allí transportarse a la CPU. El bus se controla y maneja desde la CPU.

**Switch:** Dispositivo analógico de lógica de interconexión de redes de computadoras que opera en la capa dos (nivel de enlace de datos). Su función es interconectar dos o más segmentos de red.

**Router:** Dispositivo de hardware para interconexión de red de ordenadores que opera en la capa tres (nivel de red). Permite asegurar el enrutamiento de paquetes entre redes o determinar la ruta que debe tomar el paquete de datos.

**Controlar:** Acción de mantener la medición dentro de límites aceptables.

**Driver:** Posibilita que el sistema operativo de una computadora pueda entenderse con un equipamiento periférico.

**AndroMDA:** Programa informático de tipo marco de trabajo, de generación extensible de código, que se adhiere al paradigma de la arquitectura dirigida por modelos.

**Tarjetas CRC (Clase - Responsabilidad - Colaborador):** Estas tarjetas se dividen en tres secciones que contienen la información del nombre de la clase, sus responsabilidades y sus colaboradores.

**OO:** Orientado a Objetos.

**XML (Extensible Markup Language):** Es un Lenguaje de Etiquetado Extensible muy simple, pero estricto que juega un papel fundamental en el intercambio de una gran variedad de datos.

**Triggers:** Objetos relacionados con tablas y almacenados en la base de datos que se ejecutan o se muestran cuando sucede algún evento sobre sus tablas asociadas.

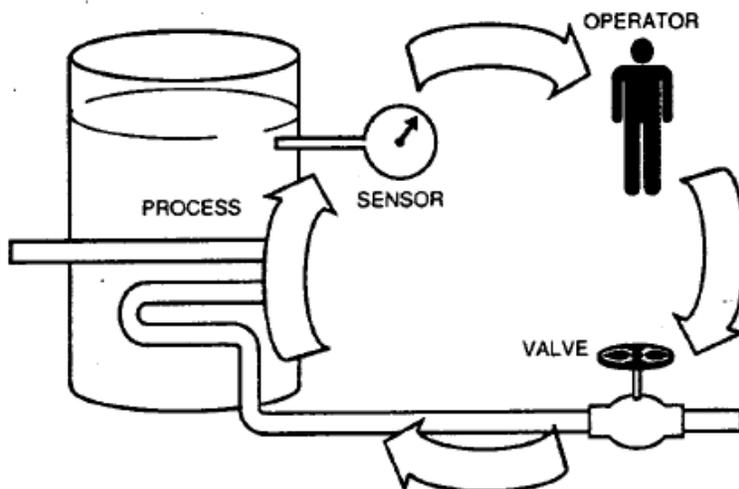
**Query:** Cadena de consulta, este término generalmente se utiliza para hacer referencias a una interacción con una base de datos.

**Orejas:** No escuchar alternativas en función de los requisitos del problema y los recursos disponibles.

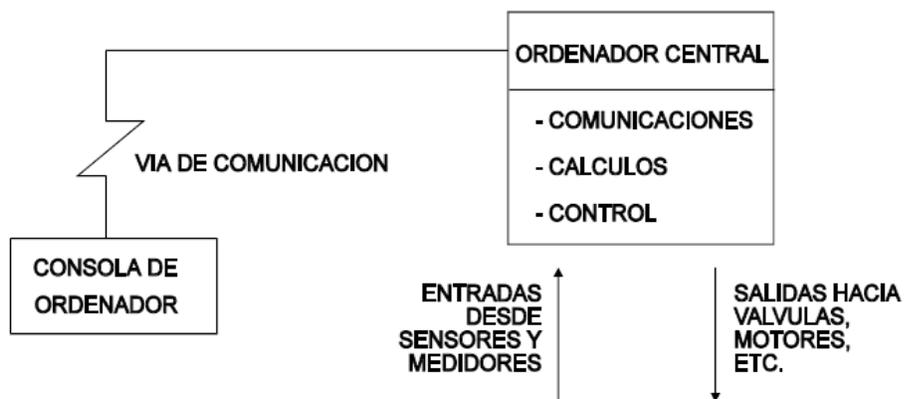
**Select:** Palabra clave que indica que la sentencia de SQL que queremos ejecutar es de selección.

## Anexos

Anexo 1: Control manual: un lazo de control que incluye el proceso, un sensor, el operador y un elemento final de control (válvula).



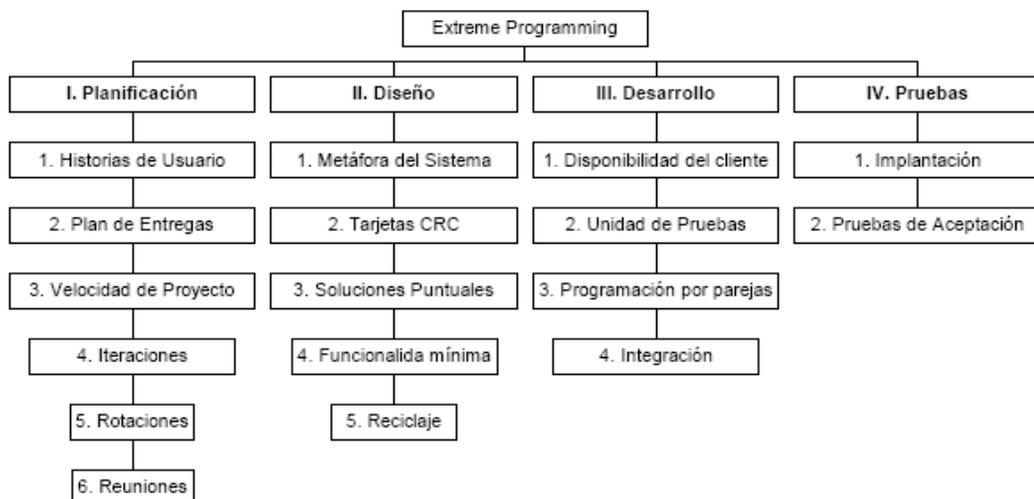
Anexo 2: Control centralizado por ordenador.



Anexo 3: Estructura de un controlador difuso.



Anexo 4: Fases de XP.



Anexo 5: Objetivos del Sistema.

<b>OBJ-1</b>	Conexión a Sistema.
<b>Descripción</b>	El sistema deberá permitir al usuario acceder a él o salir.

<b>Requisitos Asociados</b>	<p>RF 01. Conectarse al sistema.</p> <p>RF 01.1. Registrarse.</p> <p>RF 01.2. Autenticarse.</p> <p>RF 37. Crear registro del usuario.</p> <p>RF 01.4. Asignar roles.</p> <p>RF 02. Desconectarse del sistema.</p> <p>RF 38. Actualizar registro del usuario.</p>
-----------------------------	--

<b>OBJ-2</b>	Conexión a Proceso.
<b>Descripción</b>	El sistema deberá permitir al usuario conectarse y desconectarse de un proceso.
<b>Requisitos Asociados</b>	<p>RF 03. Conectarse a un proceso.</p> <p>RF 13. Mostrar listado de los tipos de procesos existentes.</p> <p>RF 03.1. Mostrar un listado de los modelos de procesos por tipo de proceso.</p> <p>RF 38. Actualizar registro del usuario.</p> <p>RF 04. Desconectarse de un proceso.</p>

<b>OBJ-3</b>	Desconectar a un usuario de un proceso.
<b>Descripción</b>	El sistema deberá permitir al administrador desconectar a un usuario de un proceso.

<b>Requisitos Asociados</b>	<p>RF 41. Mostrar listado de procesos activos.</p> <p>RF 43. Mostrar listado de usuarios conectados a un proceso.</p> <p>RF 05. Desconectar usuario de un proceso</p> <p>RF 38. Actualizar registro del usuario.</p>
-----------------------------	--

<b>OBJ-4</b>	Gestionar rol de administrador.
<b>Descripción</b>	El sistema deberá gestionar los administradores que tendrán privilegios sobre su funcionamiento: adicionarlos, modificar sus datos y como eliminarlos.
<b>Requisitos Asociados</b>	<p>RF 06. Mostrar listado de los administradores.</p> <p>RF 07. Adicionar administrador.</p> <p>RF 08. Modificar datos de un administrador.</p> <p>RF 09. Eliminar un administrador.</p>

<b>OBJ-5</b>	Establecer límites.
<b>Descripción</b>	El sistema deberá permitir al administrador modificar los datos de los límites en cuanto a: cantidad de usuarios conectados a él o a un proceso y la cantidad de procesos activos simultáneamente.
<b>Requisitos Asociados</b>	<p>RF 10. Limitar el número de usuarios conectados al sistema.</p> <p>RF 11. Limitar el número de usuarios conectados por procesos.</p> <p>RF 19. Mostrar listado de los modelos de procesos existentes.</p>

	<p>RF 12. Limitar el número de procesos activos.</p> <p>RF 36. Actualizar configuración del NVP.</p>
--	--

<b>OBJ-6</b>	Gestionar Tipos de Procesos.
<b>Descripción</b>	El sistema deberá permitir al administrador gestionar los tipos de procesos: adicionarlos, modificar sus datos y eliminarlos.
<b>Requisitos Asociados</b>	<p>RF 13. Mostrar listado de los tipos de procesos existentes.</p> <p>RF 14. Adicionar un tipo de proceso.</p> <p>RF 15. Modificar un tipo de proceso.</p> <p>RF 16. Eliminar un tipo de proceso.</p> <p>RF 17. Notificar resultado de la gestión de tipos de procesos.</p>

<b>OBJ-7</b>	Gestionar Modelos de Procesos.
<b>Descripción</b>	El sistema deberá permitir al administrador gestionar los modelos de procesos: adicionarlos, modificar sus datos y eliminarlos.
<b>Requisitos Asociados</b>	<p>RF 18. Adicionar un modelo de un proceso.</p> <p>RF 19. Mostrar un listado de los modelos de procesos existentes.</p> <p>RF 20. Modificar un modelo de un proceso.</p> <p>RF 21. Eliminar un modelo de un proceso.</p> <p>RF 38. Actualizar registro.</p> <p>RF 22. Notificar resultado de la gestión de modelos de procesos.</p>

<b>OBJ-8</b>	Configurar Proceso.
<b>Descripción</b>	El sistema deberá permitir al usuario maestro configurar un proceso.
<b>Requisitos Asociados</b>	RF 23. Configurar un modelo de proceso RF 38. Actualizar registro. RF 24. Notificar resultado de la configuración.

<b>OBJ-9</b>	Simular Proceso.
<b>Descripción</b>	El sistema deberá permitir al usuario maestro simular un proceso.
<b>Requisitos Asociados</b>	RF 25. Mostrar el resultado final de la simulación. RF 26. Mostrar el resultado de la simulación de forma gráfica.

<b>OBJ-10</b>	Probar Aplicaciones.
<b>Descripción</b>	El sistema deberá permitir al usuario maestro probar un proceso en tiempo real.
<b>Requisitos Asociados</b>	RF 27. Establecer conexión con el controlador. RF 25. Mostrar el resultado final de la simulación. RF 26. Mostrar el resultado de la simulación de forma gráfica.

<b>OBJ-11</b>	Gestionar controladores.
<b>Descripción</b>	El sistema deberá permitir al administrador gestionar controladores: adicionarlos, consultar sus datos y eliminarlos.
<b>Requisitos Asociados</b>	RF 28. Mostrar listado de los controladores. RF 29. Adicionar controladores. RF 30. Mostrar datos de controladores. RF 31. Eliminar controladores.

<b>OBJ-12</b>	Activar automáticamente uno o varios procesos.
<b>Descripción</b>	El sistema deberá permitir al administrador activar uno o varios procesos en un período de tiempo predefinido.
<b>Requisitos Asociados</b>	RF 13. Mostrar un listado de los tipos de procesos existentes. RF 32. Mostrar un listado de los modelos de procesos inactivos por tipo de proceso. RF 33. Programar el tiempo de duración de la activación. RF 34. Activar proceso automáticamente. RF 38. Actualizar registro.

<b>OBJ-13</b>	Priorizar procesos activos por tipo de proceso.
<b>Descripción</b>	El sistema deberá permitir al administrador priorizar el número de procesos activos para cada tipo de proceso.

<b>Requisitos Asociados</b>	<p>RF 13. Mostrar un listado de los tipos de procesos existentes.</p> <p>RF 35. Definir prioridades por tipo de proceso.</p> <p>RF 36. Actualizar configuración del NVP.</p>
-----------------------------	--

<b>OBJ-14</b>	Gestionar registro de usuarios.
<b>Descripción</b>	El sistema deberá permitir gestionar el registro de acciones de cada usuario: crearlo, consultarlo, y actualizarlo.
<b>Requisitos Asociados</b>	<p>RF 37. Crear registro del usuario.</p> <p>RF 38. Actualizar registro del usuario.</p> <p>RF 39. Mostrar registro del usuario.</p> <p>RF 40. Mostrar listado de usuarios existentes.</p>

<b>OBJ-15</b>	Gestionar reportes.
<b>Descripción</b>	El sistema deberá permitir gestionar reportes necesarios para el trabajo del administrador: procesos activos, usuarios conectados, usuarios conectados a un proceso, procesos activados por un usuario.
<b>Requisitos Asociados</b>	<p>RF 41. Mostrar un listado de procesos activos.</p> <p>RF 42. Mostrar usuarios conectados.</p> <p>RF 43. Mostrar listado de usuarios conectados a un proceso.</p> <p>RF 44. Mostrar listado de los procesos activados por un usuario.</p> <p>RF 45. Imprimir informes.</p>

## Anexo 6: Descripción detallada del CUS Gestionar\_Conexión\_Sistema.

<b>Caso de Uso – 1</b>	Gestionar_Conexión_Sistema	
<b>Actores</b>	Usuario	
<b>Propósito</b>	Permite que un usuario lleve a cabo la acción de conectarse y desconectarse del sistema.	
<b>Resumen</b>	El CUS se inicia cuando el usuario solicita la conexión o desconexión del sistema, el sistema verifica las solicitudes, realiza la acción seleccionada por el usuario y finaliza el caso de uso.	
<b>Descripción</b>		
<b>Pos-condiciones</b>	<ol style="list-style-type: none"> <li>1. Usuario conectado al sistema.</li> <li>2. Usuario desconectado del sistema.</li> </ol>	
<b>Flujo Normal de Eventos</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	
1. El Usuario selecciona la opción de entrar o de salir del sistema.	1.1. El sistema muestra una interfaz con las opciones de Registrarse y Autenticarse, o muestra un mensaje confirmando que el usuario quiere salir del sistema.	
<b>Escenario 1: Registrarse.</b>		
<ol style="list-style-type: none"> <li>1. El Usuario selecciona la opción de registrarse.</li> <li>2. El Usuario introduce los datos requeridos.</li> </ol>	<ol style="list-style-type: none"> <li>1.1. El sistema muestra el formulario para que el usuario se registre.</li> <li>2.1. El sistema verifica los datos introducidos por el usuario.</li> <li>2.2. Si el sistema no encuentra datos semejantes en la Base de Datos, crea un nuevo usuario con</li> </ol>	

	los datos introducidos y finaliza el caso de uso.
<b>Cursos Alternos</b>	
	2.2. Si el sistema encuentra datos semejantes, muestra un mensaje informando que ese usuario ya está registrado e indica retornar a la acción 2.
<b>Escenario 2:</b> Autenticarse.	
<p>1. El Usuario selecciona la opción de autenticarse.</p> <p>2. El Usuario introduce los datos requeridos.</p>	<p>1.1. El sistema verifica disponibilidad de conexión de usuarios.</p> <p>1.2. Si hay disponibilidad, el sistema muestra el formulario para la autenticación del usuario.</p> <p>2.1. El sistema verifica los datos introducidos por el usuario.</p> <p>2.2. Si los datos introducidos son correctos, el sistema le da acceso al usuario en dependencia del rol, al módulo de administrador o al módulo de usuario.</p> <p>2.3. El sistema crea o actualiza el registro de acciones del usuario y finaliza el caso de uso.</p>
<b>Cursos Alternos</b>	
	<p>1.2. Si no hay disponibilidad, el sistema muestra un mensaje informándole al usuario que no hay disponibilidad de conexión y finaliza el caso de uso.</p> <p>2.2. Si los datos introducidos por el usuario son incorrectos, el sistema muestra un mensaje de error e indica al usuario retornar a la acción 2.</p>

<b>Escenario 3:</b> Desconexión del sistema.	
1. El Usuario solicita salir del sistema.	1.1. El sistema verifica que el usuario no esté conectado a un proceso.
2. El Usuario confirma o no salir del sistema.	1.2. Si el usuario no está conectado a un proceso, el sistema muestra un mensaje de advertencia para confirmar la acción a realizar.
	2.1. Si el usuario confirma que desea salir, el sistema lo desconecta.
	2.2. El sistema actualiza el registro de acciones del usuario y finaliza el caso de uso.
<b>Cursos Alternos</b>	
	1.2. Si el usuario está conectado a un proceso, el sistema le informa que no puede desconectarse mientras no se desconecte del proceso antes, y lo envía a la acción 1.
	2.1. Si el usuario no confirma que desea salir, finaliza el caso de uso sin ejecutar ninguna acción.
<b>Referencia:</b>	RF 1, RF 2, RF 37, RF 38
<b>Prioridad:</b>	Crítico

## Anexo 7: Descripción detallada del CUS Gestionar\_Conexión\_Proceso.

<b>Caso de Uso – 2</b>	Gestionar_Conexión_Proceso
<b>Actores</b>	Usuario
<b>Propósito</b>	Permite que un usuario lleve a cabo la acción de conectarse y

	desconectarse de un proceso.	
<b>Resumen</b>	El CUS se inicia cuando el usuario solicita la conexión o desconexión a un proceso, el sistema verifica las solicitudes, realiza la acción seleccionada por el usuario y finaliza el caso de uso.	
<b>Descripción</b>		
<b>Pos-condiciones</b>	<ol style="list-style-type: none"> <li>1. Usuario conectado a un proceso activo.</li> <li>2. Usuario conectado a un proceso inactivo.</li> <li>3. Usuario desconectado de un proceso.</li> </ol>	
<b>Flujo Normal de Eventos</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	
<ol style="list-style-type: none"> <li>1. El Usuario solicita la opción de conectarse a un proceso.</li> <li>2. El Usuario selecciona un tipo de proceso.</li> </ol>	<ol style="list-style-type: none"> <li>1.1. El sistema muestra una interfaz con los tipos de procesos existentes.</li> <li>2.1. El sistema muestra los procesos, tanto activos como inactivos del tipo seleccionado.</li> <li>2.2. Una vez conectado a un proceso, el sistema da la posibilidad al usuario de desconectarse del mismo.</li> </ol>	
<b>Escenario 1:</b> Conexión a un proceso activo.		
<ol style="list-style-type: none"> <li>1. El Usuario selecciona el proceso activo al que se desea conectar.</li> <li>2. El Usuario selecciona la opción Conectar.</li> </ol>	<ol style="list-style-type: none"> <li>2.1. El sistema verifica la disponibilidad de conexión para dar acceso al usuario.</li> <li>2.2. Si hay disponibilidad de conexión, el sistema conecta el usuario (común) al proceso seleccionado.</li> <li>2.3. El sistema actualiza el registro de acciones del usuario y finaliza el caso de uso.</li> </ol>	

<b>Cursos Alternos</b>	
	<p>2.2. Si no hay disponibilidad de conexión, el sistema muestra un mensaje al usuario informándole que no hay capacidad de conexión, que lo intente más tarde y el caso de uso finaliza sin ninguna acción.</p>
<b>Escenario 2:</b> Conexión a un proceso inactivo.	
<p>1. El Usuario selecciona el proceso inactivo al que se desea conectar.</p> <p>2. El Usuario selecciona la opción Conectar.</p> <p>3. El Usuario elige o no ser maestro.</p>	<p>2.1. El sistema verifica la disponibilidad de conexión para dar acceso al usuario.</p> <p>2.2. Si hay disponibilidad de conexión, el sistema conecta el usuario al proceso seleccionado.</p> <p>2.3. El sistema muestra un mensaje de advertencia preguntándole al usuario si desea ser maestro del proceso.</p> <p>3.1. Si el usuario elige ser común, el sistema activa el proceso por defecto.</p> <p>3.2. El sistema actualiza el registro de acciones del usuario y finaliza el caso de uso.</p>
<b>Cursos Alternos</b>	
	<p>2.2. Si se ha superado alguno de los límites establecidos, el sistema va a mostrar un mensaje al usuario informándole que no hay capacidad de conexión, que lo intente más tarde y el caso de uso finaliza sin ninguna acción.</p> <p>3.1. Si el usuario elige ser maestro, el sistema muestra la interfaz correspondiente al CUS</p>

	Configurar_Proceso.
<b>Escenario 3: Desconexión de un proceso.</b>	
<p>1. El Usuario solicita la desconexión del proceso.</p> <p>2. El Usuario confirma o no la opción de desconexión.</p>	<p>1.1. El sistema muestra un mensaje de advertencia para confirmar la acción a realizar.</p> <p>2.1. Si el usuario confirma la desconexión, el sistema comprueba si es usuario maestro del proceso.</p> <p>2.2. Si el usuario es maestro, verifica la existencia de otros usuarios conectados al proceso.</p> <p>2.3. Si no existen otros usuarios conectados al proceso, éste se desactiva.</p> <p>2.4. El sistema elimina al usuario de la lista de usuarios conectados al proceso.</p> <p>2.5. El sistema actualiza el registro de acciones del usuario y finaliza el caso de uso.</p>
<b>Cursos Alternos</b>	
	<p>2.1. Si el usuario no confirma la desconexión, finaliza el caso de uso sin realizar ninguna acción.</p> <p>2.2. Si el usuario es común, el sistema pasa a la acción 2.3.</p> <p>2.3. Si existen otros usuarios conectados al proceso, el sistema desconecta al usuario del proceso, mediante un sistema de notificaciones gestiona un nuevo maestro y finaliza el caso de uso.</p>
<b>Referencia:</b>	RF 3, RF 4, RF 13, RF 38

<b>Prioridad:</b>	Crítico
-------------------	---------

Anexo 8: Descripción detallada del CUS Establecer\_Límites.

<b>Caso de Uso – 5</b>	Establecer_Límites	
<b>Actores</b>	Administrador	
<b>Propósito</b>	Permite al administrador modificar los datos de los límites en cuanto a la cantidad de usuarios conectados al sistema o a un proceso y la cantidad de procesos activos simultáneamente.	
<b>Resumen</b>	El CUS se inicia cuando el administrador de la aplicación selecciona la opción de Establecer Límite, luego selecciona el tipo de límite que va a modificar, introduce los datos necesarios, el sistema realiza la acción seleccionada por el administrador y finaliza el caso de uso.	
<b>Descripción</b>		
<b>Pos-condiciones</b>	1. Información de los límites adicionada a la Base de Datos.	
<b>Flujo Normal de Eventos</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	
1. El Administrador selecciona la opción de Establecer Límite.	1.1 El sistema muestra las opciones de Limitar cantidad de usuarios del sistema, Limitar cantidad de usuarios en proceso y Limitar cantidad de procesos activos.	
<b>Escenario 1:</b> Limitar cantidad de usuarios del sistema.		
1. El Administrador selecciona la opción de limitar el número de usuarios conectados al	1.1. El sistema muestra un formulario a llenar para hacer la limitación solicitada.	

<p>sistema.</p> <p>2. El Administrador introduce el dato requerido.</p>	<p>2.1. El sistema verifica el dato introducido.</p> <p>2.2. Si el dato introducido es correcto, el sistema verifica que es mayor o igual que la cantidad de usuarios conectados al sistema.</p> <p>2.3. Si la cantidad de usuarios especificada es válida el sistema guarda la información en la Base de Datos y finaliza el caso de uso.</p>
<b>Cursos Alternos</b>	
	<p>2.2. Si el dato introducido por el administrador es incorrecto, el sistema muestra un mensaje de error e indica al administrador retornar a la acción 2.</p> <p>2.3. Si el dato introducido es menor que la cantidad de usuarios conectados mostrar el mensaje indicando que debe introducir un número mayor o desconectar usuarios.</p>
<b>Escenario 2:</b> Limitar cantidad de usuarios en proceso.	
<p>1. El Administrador selecciona la opción de limitar el número de usuarios conectados a un proceso.</p> <p>2. El Administrador selecciona un modelo de proceso.</p> <p>3. El Administrador introduce el dato requerido.</p>	<p>1.1. El sistema muestra un listado con los modelos de los procesos existentes.</p> <p>2.1. El sistema muestra un formulario a llenar para hacer la limitación en el modelo de proceso seleccionado.</p> <p>3.1. El sistema verifica el dato introducido.</p> <p>3.2. Si el dato introducido es correcto, el sistema verifica que la cantidad especificada es mayor o igual que la cantidad de usuarios conectados al proceso seleccionado.</p> <p>3.3. Si la cantidad de usuarios especificada es</p>

	válida, el sistema guarda la información en la Base de Datos y finaliza el caso de uso.
<b>Cursos Alternos</b>	
	<p>3.2. Si el dato introducido por el administrador es incorrecto, el sistema muestra un mensaje de error y lo envía a la acción 3.</p> <p>3.3. Si la cantidad introducida es menor a la cantidad de usuarios conectados al proceso seleccionado, se muestra un mensaje indicándole introducir un número mayor o desconectar usuarios del proceso.</p>
<b>Escenario 3:</b> Limitar cantidad de procesos activos.	
<p>1. El Administrador selecciona la opción de limitar el número de procesos activos.</p> <p>2. El Administrador introduce el dato requerido.</p>	<p>1.1. El sistema muestra un formulario a llenar para hacer la limitación.</p> <p>2.1. El sistema verifica el dato introducido.</p> <p>2.2. Si el dato introducido es correcto, el sistema verifica que es mayor o igual que la cantidad de procesos activos en el sistema.</p> <p>2.3. Si la cantidad especificada es válida el sistema guarda la información en la Base de Datos y finaliza el caso de uso.</p>
<b>Cursos Alternos</b>	
	<p>2.2. Si el dato introducido por el administrador es incorrecto, el sistema muestra un mensaje de error y lo envía a la acción 2.</p> <p>2.3. Si el dato introducido es menor que la cantidad</p>

	de procesos activos, el sistema muestra un mensaje indicando que debe introducir un número mayor o que debe desactivar procesos.
<b>Referencia:</b>	RF 10, RF 11, RF 12, RF 19, RF 36
<b>Prioridad:</b>	Crítico

## Anexo 9: Descripción detallada del CUS Gestionar\_Tipo\_Proceso.

<b>Caso de Uso – 6</b>	Gestionar_Tipo_Proceso
<b>Actores</b>	Administrador
<b>Propósito</b>	Permite al administrador gestionar (adicionar, modificar o eliminar) tipos de procesos.
<b>Resumen</b>	El CUS se inicia cuando el administrador de la aplicación selecciona la opción de Gestionar Tipo de Proceso, luego selecciona el tipo de gestión, el sistema realiza la acción seleccionada por el administrador y finaliza el caso de uso.
<b>Descripción</b>	
<b>Pos-condiciones</b>	<ol style="list-style-type: none"> <li>1. Información del tipo de proceso adicionada a la Base de Datos.</li> <li>2. Información del tipo de proceso modificada en la Base de Datos.</li> <li>3. Información del tipo de proceso eliminada de la Base de Datos.</li> </ol>
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. El Administrador selecciona la opción de Gestionar Tipo de Proceso.	1.1. El sistema muestra el listado de los tipos de procesos existentes además de las opciones de Adicionar Tipo de Proceso, Modificar Tipo de

	Proceso y Eliminar Tipo de Proceso.
<b>Escenario 1:</b> Adicionar Tipo de Proceso.	
<p>1. El Administrador selecciona la opción de Adicionar Tipo de Proceso.</p> <p>2. El Administrador introduce los datos requeridos.</p> <p>3. El Administrador confirma o no la creación del tipo de proceso.</p>	<p>1.1. El sistema muestra el formulario a completar para la adición de un nuevo tipo de proceso.</p> <p>2.1. El sistema verifica los datos introducidos por el administrador.</p> <p>2.2. Si los datos introducidos son correctos, el sistema verifica que el tipo de proceso no está en el listado de los tipos de procesos existentes.</p> <p>2.3. Si el tipo de proceso a adicionar no existe, el sistema muestra un mensaje de advertencia para confirmar la creación del tipo de proceso.</p> <p>3.1. Si el administrador confirma crear el tipo de proceso, el sistema lo adiciona en la Base de Datos.</p> <p>3.2. El sistema envía una notificación a todos los usuarios conectados y finaliza el caso de uso.</p>
<b>Cursos Alternos</b>	
	<p>2.2. Si los datos introducidos por el administrador son incorrectos, el sistema muestra un mensaje de error indicándole donde está el dato erróneo y lo envía a la acción 2.</p> <p>2.3. Si el tipo de proceso a adicionar existe, el sistema muestra un mensaje indicándolo y envía al administrador a la acción 2.</p> <p>3.1. Si el administrador no confirma la creación del</p>

	tipo de proceso, finaliza el caso de uso sin realizar ninguna acción.
<b>Escenario 2: Modificar Tipo de Proceso.</b>	
<p>1. El Administrador selecciona el tipo de proceso a modificar.</p> <p>2. El Administrador selecciona la opción de Modificar Tipo de Proceso.</p> <p>3. El Administrador realiza los cambios necesarios a los datos.</p> <p>4. El Administrador confirma o no la modificación del tipo de proceso.</p>	<p>2.1. El sistema localiza los datos del tipo de proceso seleccionado y los muestra, listos para modificar.</p> <p>3.1. El sistema verifica los datos modificados por el administrador.</p> <p>3.2. Si los datos introducidos están correctos, el sistema muestra un mensaje de advertencia para confirmar la modificación del tipo de proceso.</p> <p>4.1. Si el administrador confirma la modificación del tipo de proceso, el sistema actualiza los datos del mismo en la Base de Datos.</p> <p>4.2. El sistema envía una notificación a todos los usuarios conectados y finaliza el caso de uso.</p>
<b>Cursos Alternos</b>	
	<p>3.2. Si los datos introducidos por el administrador son incorrectos, el sistema muestra un mensaje de error indicándole donde está el dato erróneo y lo envía a la acción 3.</p> <p>4.1. Si el administrador no confirma la modificación del tipo de proceso, finaliza el caso de uso sin realizar ninguna acción.</p>
<b>Escenario 3: Eliminar Tipo de Proceso.</b>	
1. El Administrador selecciona el Tipo de	2.1. El sistema comprueba si existen usuarios

<p>Proceso a eliminar.</p> <p>2. El Administrador selecciona la opción de Eliminar Tipo de Proceso.</p> <p>3. El Administrador confirma o no eliminar el Tipo de Proceso.</p>	<p>conectados a algún proceso del tipo seleccionado.</p> <p>2.2. Si no hay usuarios conectados algún proceso de este tipo, el sistema muestra un mensaje de advertencia para confirmar la acción a realizar.</p> <p>3.1. Si el administrador confirma la eliminación, el sistema elimina los datos del Tipo de proceso seleccionado, incluyendo los procesos asociados a él y finaliza el caso de uso.</p>
<b>Cursos Alternos</b>	
	<p>2.2. Si existen usuarios conectados a algún proceso del tipo seleccionado, el sistema muestra un mensaje indicándoselo al administrador y que lo intente más tarde.</p> <p>3.1 Si el administrador no confirma la eliminación, finaliza el caso de uso sin ejecutar ninguna acción.</p>
<b>Referencia:</b>	RF 13, RF 14, RF 15, RF 16, RF 17
<b>Prioridad:</b>	Crítico

Anexo 10: Descripción detallada del CUS Gestionar\_Modelo\_Proceso.

<b>Caso de Uso – 7</b>	Gestionar_Modelo_Proceso
<b>Actores</b>	Administrador
<b>Propósito</b>	Permite al administrador gestionar (adicionar, modificar o eliminar) modelos de procesos.
<b>Resumen</b>	El CUS se inicia cuando el administrador de la aplicación selecciona la opción de Gestionar Modelo de Proceso, luego selecciona el tipo de

	gestión, el sistema realiza la acción seleccionada por el administrador y finaliza el caso de uso.
<b>Descripción</b>	
<b>Pos-condiciones</b>	<ol style="list-style-type: none"> <li>1. Información del modelo de proceso adicionada a la Base de Datos.</li> <li>2. Información del modelo de proceso modificada en la Base de Datos.</li> <li>3. Información del modelo de proceso eliminada de la Base de Datos.</li> </ol>
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. El Administrador selecciona la opción de Gestionar Modelo de Proceso.	<ol style="list-style-type: none"> <li>1.1. El sistema muestra listado de los modelos de procesos existentes.</li> <li>1.2. El sistema muestra las opciones de Adicionar Modelo de Proceso, Modificar Modelo de Proceso y Eliminar Modelo de Proceso.</li> </ol>
<b>Escenario 1: Adicionar Modelo de Proceso.</b>	
<ol style="list-style-type: none"> <li>1. El Administrador selecciona la opción de Adicionar Modelo de Proceso.</li> <li>2. El Administrador introduce los datos requeridos.</li> <li>3. El Administrador confirma o no la creación.</li> </ol>	<ol style="list-style-type: none"> <li>1.1. El sistema muestra el formulario a completar para la adición de un nuevo modelo de proceso.</li> <li>2.1. El sistema verifica los datos introducidos por el administrador.</li> <li>2.2. Si los datos introducidos son correctos, el sistema verifica que el modelo de proceso a adicionar no existe.</li> <li>2.3. Si el modelo de proceso a adicionar no existe, el sistema muestra un mensaje de advertencia para confirmar la creación del mismo.</li> <li>3.1. Si el administrador confirma la creación, el</li> </ol>

	<p>sistema adiciona dicha información en la Base de Datos.</p> <p>3.2. El sistema actualiza el registro de acciones del administrador.</p> <p>3.3. El sistema envía una notificación a todos los usuarios conectados y finaliza el caso de uso.</p>
<b>Cursos Alternos</b>	
	<p>2.2. Si los datos introducidos por el administrador son incorrectos, el sistema muestra un mensaje de error indicándole donde está el dato erróneo y lo envía a la acción 2.</p> <p>2.3. Si el modelo de proceso a adicionar existe, el sistema muestra un mensaje indicándolo y envía al administrador a la acción 2.</p> <p>3.1. Si el administrador no confirma la creación del modelo de proceso, finaliza el caso de uso sin ejecutar ninguna acción.</p>
<b>Escenario 2:</b> Modificar Modelo de Proceso.	
<p>1. El Administrador selecciona el modelo de proceso a modificar.</p> <p>2. El Administrador selecciona la opción de Modificar Modelo de Proceso.</p> <p>3. El Administrador realiza los cambios necesarios a los datos.</p> <p>4. El Administrador confirma o no la modificación.</p>	<p>2.1. El sistema localiza los datos del modelo de proceso y los muestra, listos para modificar.</p> <p>3.1. El sistema verifica los datos modificados por el administrador.</p> <p>3.2. Si los datos introducidos por el administrador están correctos, el sistema muestra un mensaje de advertencia para confirmar la modificación del mismo.</p> <p>4.1. Si el administrador confirma la modificación, el</p>

	<p>sistema actualiza los datos del modelo del proceso en la Base de Datos.</p> <p>4.2. El sistema actualiza el registro de acciones del administrador.</p> <p>4.3. El sistema envía una notificación a todos los usuarios conectados y finaliza el caso de uso.</p>
<b>Cursos Alternos</b>	
	<p>3.2. Si los datos introducidos por el administrador son incorrectos, el sistema muestra un mensaje de error indicándole donde está el dato erróneo y lo envía a la acción 3.</p> <p>4.1. Si el administrador no confirma la modificación del modelo de proceso, finaliza el caso de uso sin ejecutar ninguna acción.</p>
<b>Escenario 3:</b> Eliminar Modelo de Proceso.	
<p>1. El Administrador selecciona el Modelo de Proceso a eliminar.</p> <p>2. El Administrador selecciona la opción Eliminar Modelo de Proceso.</p> <p>3. El Administrador confirma o no eliminar el Modelo de Proceso.</p>	<p>2.1. El sistema comprueba si existen usuarios conectados al modelo seleccionado.</p> <p>2.2. Si no existen usuarios conectados al modelo seleccionado, el sistema muestra un mensaje de advertencia para confirmar la acción a realizar.</p> <p>3.1. Si el administrador confirma la eliminación, el sistema elimina los datos del Modelo de proceso seleccionado.</p> <p>3.2. El sistema actualiza el registro de acciones del administrador.</p> <p>3.3. El sistema envía una notificación a todos los</p>

	usuarios conectados y finaliza el caso de uso.
<b>Cursos Alternos</b>	
	<p>2.2. Si existen usuarios conectados al modelo seleccionado, el sistema muestra un mensaje indicándole al administrador y que lo intente más tarde.</p> <p>3.1. Si el administrador no confirma la eliminación, finaliza el caso de uso sin ejecutar ninguna acción.</p>
<b>Referencia:</b>	RF 18, RF 19, RF 20, RF 21, RF 22, RF 38
<b>Prioridad:</b>	Crítico

Anexo 11: Descripción detallada del CUS Configurar\_Proceso.

<b>Caso de Uso – 8</b>	Configurar_Proceso (extendido de Gestionar_Conexión_Proceso)
<b>Actores</b>	Usuario Maestro
<b>Propósito</b>	Permite al usuario maestro configurar un proceso.
<b>Resumen</b>	El CUS se inicia en el flujo de acciones del caso de uso Gestionar_Conexión_Proceso cuando el usuario selecciona ser maestro, el usuario maestro introduce los datos necesarios, el sistema realiza la acción seleccionada y finaliza el caso de uso.
<b>Descripción</b>	
<b>Pos-condiciones</b>	Proceso Configurado.
<b>Flujo Normal de Eventos</b>	

Acción del Actor	Respuesta del Sistema
<p>2.1. El Usuario Maestro introduce los datos requeridos.</p> <p>4.1. El Usuario Maestro confirma o no la configuración.</p>	<ol style="list-style-type: none"> <li>1. Cuando se inicia el caso de uso Gestionar_Conexión_Proceso y el usuario selecciona ser maestro, el sistema muestra la interfaz correspondiente al CUS Configurar_Proceso.</li> <li>2. El sistema localiza y muestra los datos del modelo del proceso donde se encuentra el usuario maestro para ser configurados.</li> <li>3. EL sistema verifica los datos introducidos por el usuario maestro.</li> <li>4. Si los datos introducidos son correctos, el sistema muestra un mensaje de advertencia para confirmar la configuración.</li> <li>5. Si el usuario maestro confirma la configuración, el sistema activa el proceso y actualiza la información del mismo en la Base de Datos.</li> <li>6. El sistema actualiza el registro de acciones del usuario.</li> <li>7. El sistema envía una notificación al resto de los usuarios conectados al proceso configurado y finaliza el caso de uso.</li> </ol>
<b>Cursos Alternos</b>	
	<ol style="list-style-type: none"> <li>4. Si los datos introducidos por el usuario maestro son incorrectos, el sistema muestra un mensaje de error indicándole donde está el dato erróneo y lo</li> </ol>

	envía a la acción 2.1.  5. Si el usuario maestro no confirma la configuración, finaliza el caso de uso sin ejecutar ninguna acción.
<b>Referencia:</b>	RF 23, RF 24, RF 38
<b>Prioridad:</b>	Crítico

## Anexo 12: Descripción detallada del CUS Simular\_Proceso.

<b>Caso de Uso – 9</b>	Simular_Proceso (extendido de Configurar_Proceso)	
<b>Actores</b>		
<b>Propósito</b>	Permite al usuario maestro simular un proceso.	
<b>Resumen</b>	El CUS se inicia en el flujo de acciones del caso de uso Configurar_Proceso, cuando el usuario maestro de un proceso selecciona la opción de Simular Proceso sin controlador o con un controlador predefinido, el sistema realiza la acción seleccionada y finaliza el caso de uso.	
<b>Descripción</b>		
<b>Pos-condiciones</b>	Proceso Simulado.	
<b>Flujo Normal de Eventos</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	
4.1. El Usuario Maestro confirma o no comenzar la simulación.	1. Cuando se inicia el caso de uso Configurar_Proceso y el usuario selecciona la opción de simular sin controlador o con un	

	<p>controlador predefinido, el sistema activa el caso de uso Simular_Proceso.</p> <p>2. El sistema localiza los datos del proceso donde se encuentra el usuario maestro y verifica que el proceso esté configurado.</p> <p>3. Si el proceso está configurado, el sistema verifica si se puede efectuar la simulación.</p> <p>4. Si la simulación se puede efectuar, el sistema muestra un mensaje de advertencia para confirmar el inicio de la simulación.</p> <p>5. Si el usuario maestro confirma iniciar la simulación, el sistema realiza dicha acción.</p> <p>6. Durante la simulación, el sistema muestra la interfaz correspondiente al CUS Mostrar_Registro_Variables y finaliza el caso de uso.</p>
<b>Cursos Alternos</b>	
	<p>3. Si el proceso no está configurado, el sistema muestra un mensaje indicándole al usuario maestro que debe configurar antes el proceso.</p> <p>4. Si no se puede efectuar la simulación, el sistema muestra un mensaje de notificación informándole al usuario maestro que la simulación no se puede efectuar, que intente más tarde y finaliza el caso de uso sin realizar ninguna acción.</p> <p>5. Si el usuario maestro no confirma iniciar la simulación, finaliza el caso de uso sin ejecutar</p>

	ninguna acción.
<b>Referencia:</b>	RF 25, RF 26
<b>Prioridad:</b>	Crítico

## Anexo 13: Descripción detallada del CUS Probar\_Aplicación.

<b>Caso de Uso – 11</b>	Probar_Aplicación (extendido de Configurar_Proceso)	
<b>Actores</b>		
<b>Propósito</b>	Permite al usuario maestro probar un proceso en tiempo real.	
<b>Resumen</b>	El CUS se inicia en el flujo de acciones del caso de uso Configurar_Proceso, cuando el usuario maestro de un proceso selecciona la opción de Simular Proceso con un controlador físico, el sistema realiza la acción seleccionada y finaliza el caso de uso.	
<b>Descripción</b>		
<b>Pos-condiciones</b>	Proceso probado en tiempo real.	
<b>Flujo Normal de Eventos</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	
6.1. El Usuario Maestro confirma o no el inicio de la simulación.	<p>1. Cuando se inicia el caso de uso Configurar_Proceso y el usuario selecciona la opción de simular con un controlador físico, el sistema activa el caso de uso Probar_Aplicación.</p> <p>2. El sistema localiza los datos del proceso donde se encuentra el usuario maestro y verifica que el proceso esté configurado.</p>	

	<p>3. Si el proceso está configurado, el sistema verifica si se puede efectuar la simulación.</p> <p>4. Si la simulación se puede efectuar, el sistema verifica si el usuario maestro configuró correctamente un controlador.</p> <p>5. Si el controlador fue configurado correctamente, el sistema establece la conexión con el mismo.</p> <p>6. Si el sistema logró establecer la conexión, muestra un mensaje de advertencia para confirmar el inicio de la simulación.</p> <p>7. Si el usuario maestro confirma la acción, el sistema inicia la simulación.</p> <p>8. Durante la simulación, el sistema muestra la interfaz correspondiente al CUS Mostrar_Registro_Variables y finaliza el caso de uso.</p>
<b>Cursos Alternos</b>	
	<p>3. Si el proceso no está configurado, el sistema muestra un mensaje indicándole al usuario maestro que debe configurar antes el proceso.</p> <p>4. Si no se puede efectuar la simulación, el sistema muestra un mensaje de notificación informándole al usuario maestro que la acción no se puede efectuar, que intente más tarde y finaliza el caso de uso sin realizar ninguna acción.</p> <p>5. Si el controlador no ha sido configurado correctamente por el usuario maestro, el sistema</p>

	<p>muestra un mensaje de error indicándole donde está el dato erróneo y lo envía a la interfaz correspondiente al CUS Configurar_Proceso.</p> <p>6. Si el sistema no logra establecer la conexión, muestra un mensaje al usuario maestro indicándole que la conexión no se puede efectuar, que intente más tarde o que revise el dispositivo que seleccionó y lo envía a la interfaz correspondiente al CUS Configurar_Proceso.</p> <p>7. Si el usuario maestro no confirma el inicio de la simulación, finaliza el caso de uso sin ejecutar ninguna acción.</p>
<b>Referencia:</b>	RF 25, RF 26, RF 27
<b>Prioridad:</b>	Crítico

## Anexo 14: Descripción detallada del CUS Gestionar\_Controlador.

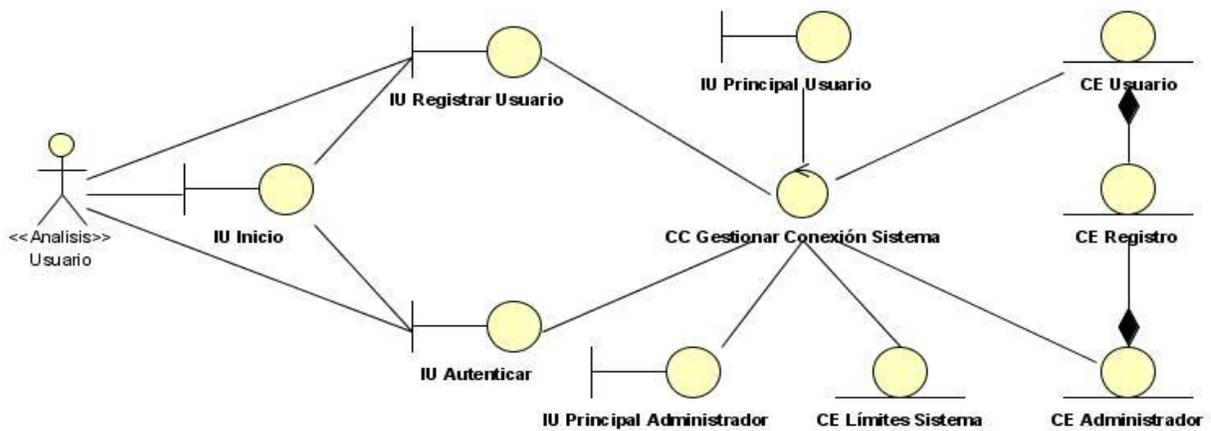
<b>Caso de Uso – 13</b>	Gestionar_Controlador
<b>Actores</b>	Administrador
<b>Propósito</b>	Permite al administrador gestionar (adicionar, visualizar o eliminar) controladores.
<b>Resumen</b>	El CUS se inicia cuando el administrador de la aplicación selecciona la opción de Gestionar Controlador, luego selecciona el tipo de gestión, el sistema realiza la acción seleccionada por el administrador y finaliza el caso de uso.
<b>Descripción</b>	

<b>Pos-condiciones</b>	<ol style="list-style-type: none"> <li>1. Información del controlador adicionada a la Base de Datos.</li> <li>2. Información del controlador mostrada al administrador.</li> <li>3. Información del controlador eliminada de la Base de Datos.</li> </ol>
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
<ol style="list-style-type: none"> <li>1. El Administrador selecciona la opción de Gestionar Controlador.</li> </ol>	<ol style="list-style-type: none"> <li>1.1. El sistema muestra el listado de los controladores existentes en la Base de Datos y las opciones de Adicionar Controlador, Mostrar Datos de Controladores y Eliminar Controlador.</li> </ol>
<b>Escenario 1:</b> Adicionar Controlador.	
<ol style="list-style-type: none"> <li>1. El Administrador selecciona la opción de Adicionar Controlador.</li> <li>2. El Administrador entra los datos requeridos.</li> </ol>	<ol style="list-style-type: none"> <li>1.1. El sistema muestra el formulario a completar para la adición de un nuevo controlador.</li> <li>2.1. El sistema verifica los datos introducidos por el administrador.</li> <li>2.2. Si los datos introducidos son correctos, el sistema verifica si el controlador a adicionar existe.</li> <li>2.3. Si el controlador a adicionar no existe, el sistema lo adiciona a la Base de Datos y finaliza el caso de uso.</li> </ol>
<b>Cursos Alternos</b>	
	<ol style="list-style-type: none"> <li>2.2. Si los datos introducidos por el administrador son incorrectos, el sistema muestra un mensaje de error indicándole donde está el dato erróneo y lo envía a la acción 2.</li> <li>2.3. Si el controlador a adicionar existe, el sistema</li> </ol>

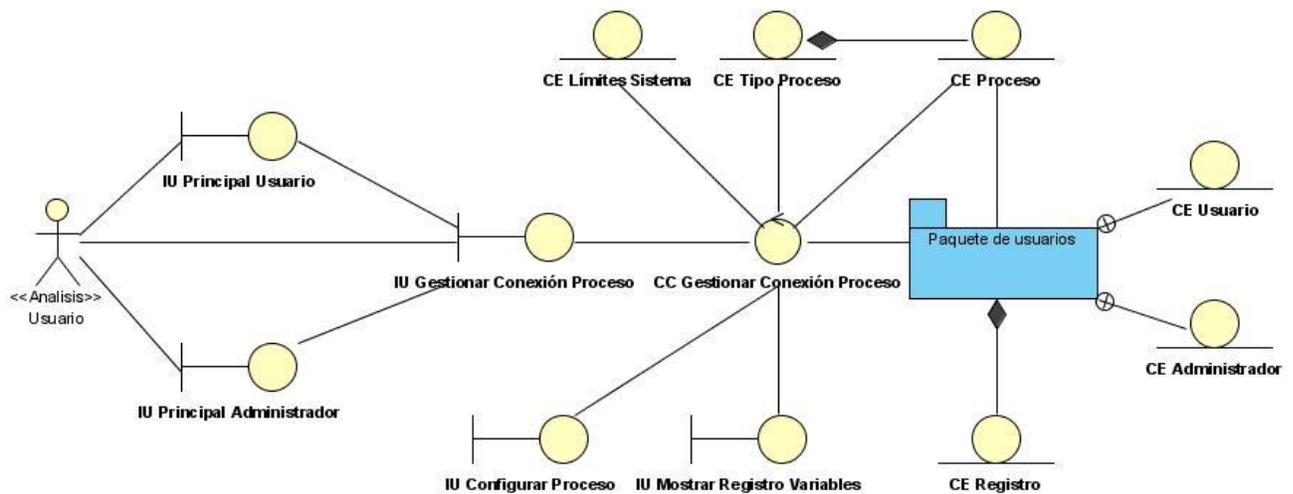
	muestra un mensaje indicándolo y envía al administrador a la acción 2.
<b>Escenario 2:</b> Mostrar Datos de Controlador.	
1. El Administrador selecciona el controlador del cual desea conocer los datos. 2. El Administrador selecciona la opción Mostrar Datos de Controladores.	2.1. El sistema localiza los datos del controlador seleccionado, los muestra al administrador y finaliza el caso de uso.
<b>Cursos Alternos</b>	
<b>Escenario 3:</b> Eliminar Controlador.	
1. El Administrador selecciona el controlador a eliminar. 2. El Administrador selecciona la opción Eliminar Controlador. 3. El Administrador confirma si quiere o no eliminar el Controlador.	2.1. Si el controlador seleccionado no está siendo utilizado por otro usuario, el sistema localiza los datos del controlador y los muestra, listos para eliminar. 2.2. El sistema muestra un mensaje de advertencia para confirmar la acción a realizar. 3.1. Si el administrador confirma la eliminación, el sistema elimina el controlador seleccionado y finaliza el caso de uso.
<b>Cursos Alternos</b>	
	2.1. Si el controlador seleccionado está siendo utilizado por otro usuario, el sistema muestra un mensaje indicándolo y envía al administrador a la acción 1. 3.1. Si el administrador no confirma la eliminación, se finaliza el caso de uso sin realizar ninguna

	acción.
<b>Referencia:</b>	RF 28, RF 29, RF 30, RF 31
<b>Prioridad:</b>	Crítico

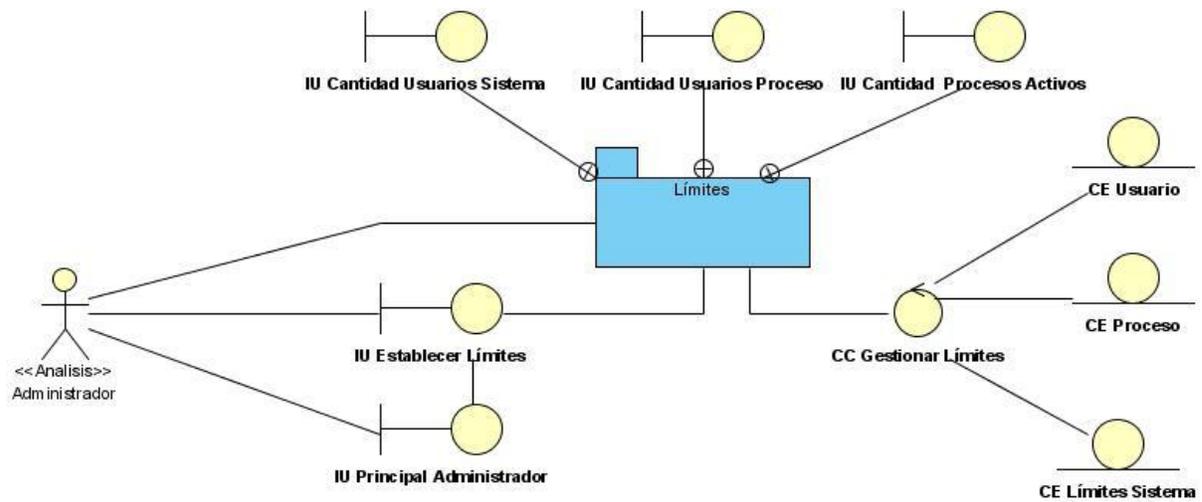
Anexo 15: Diagrama de Clases del Análisis. CU Gestionar\_Conexión\_Sistema.



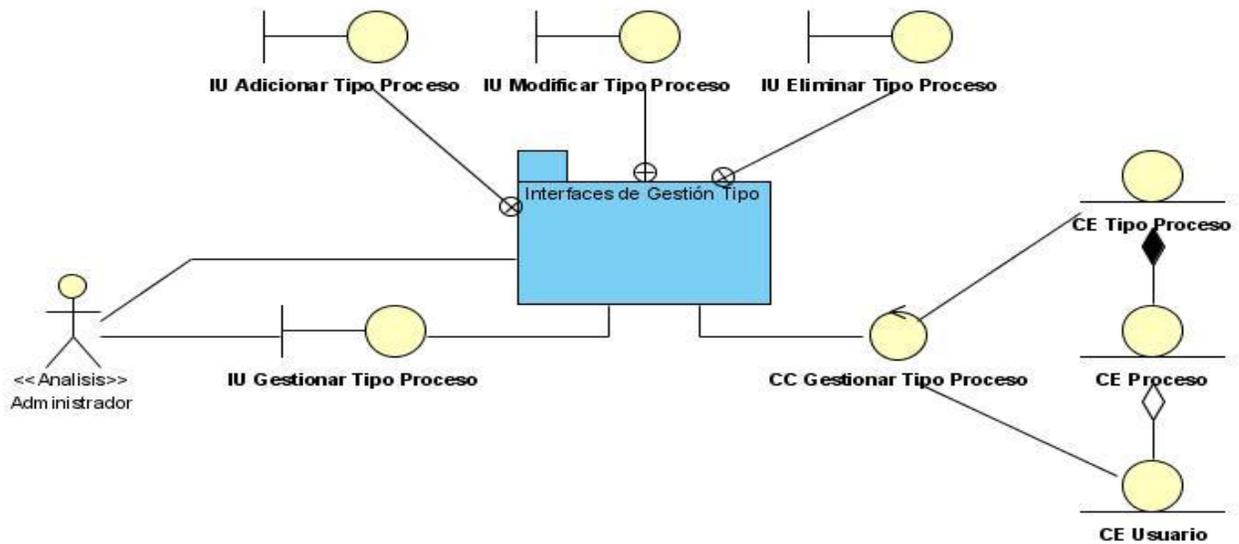
Anexo 16: Diagrama de Clases del Análisis. CU Gestionar\_Conexión\_Proceso.



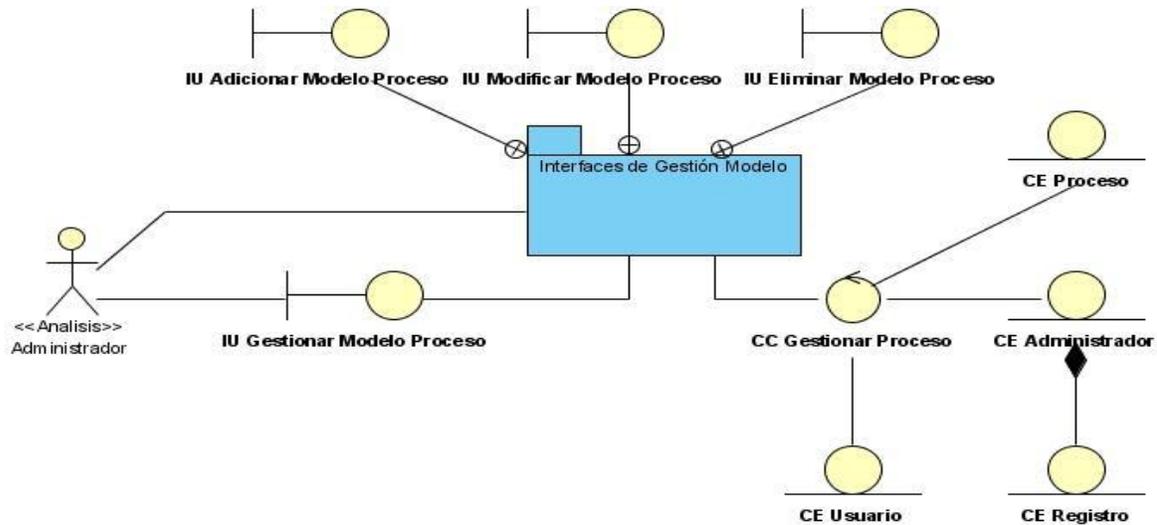
Anexo 17: Diagrama de Clases del Análisis. CU Establecer\_Límites.



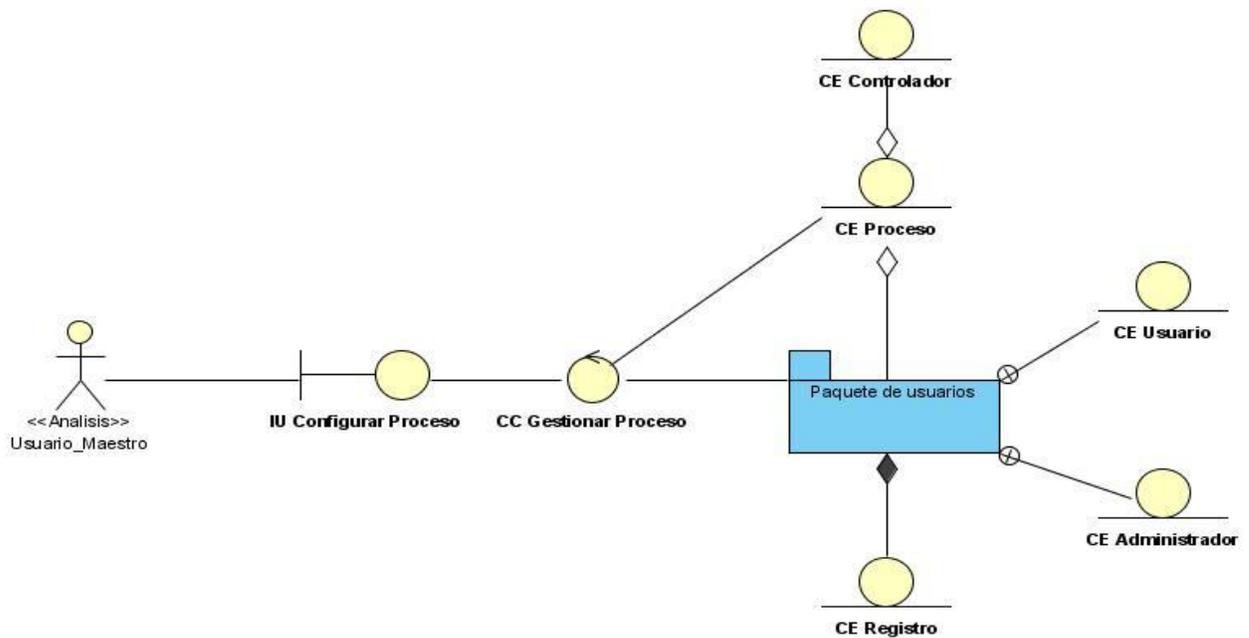
Anexo 18: Diagrama de Clases del Análisis. CU Gestionar\_Tipo\_Proceso.



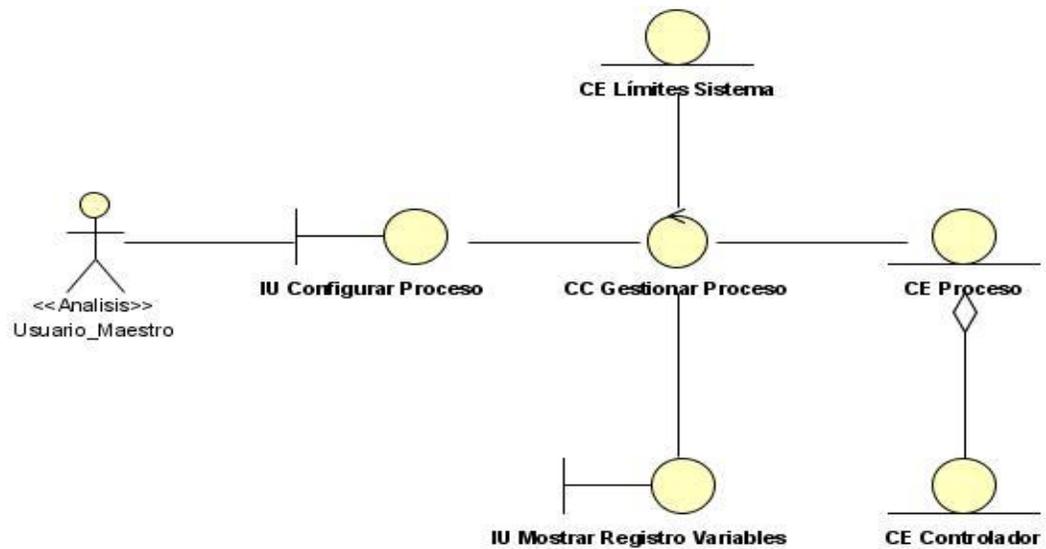
Anexo 19: Diagrama de Clases del Análisis. CU Gestionar\_Modelo\_Proceso.



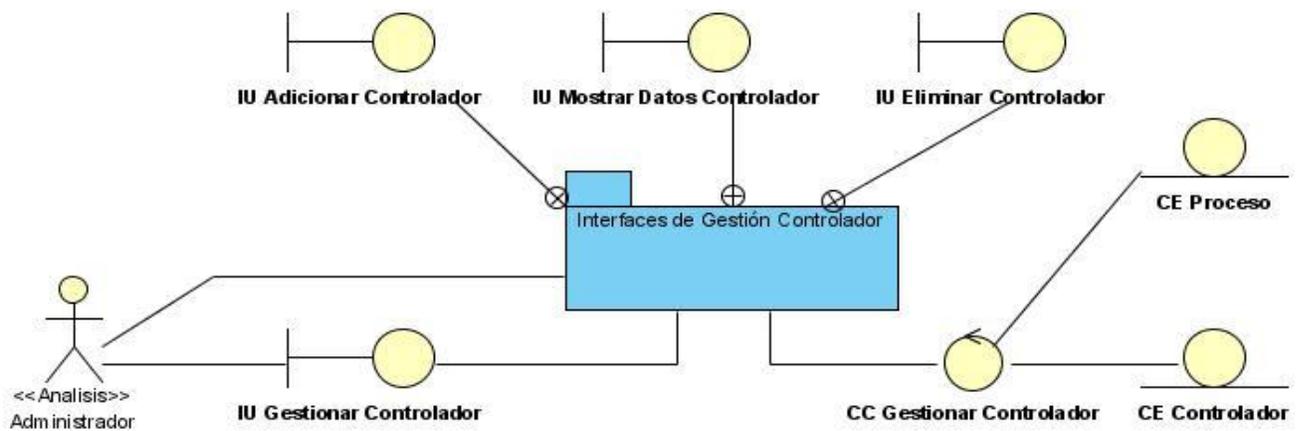
Anexo 20: Diagrama de Clases del Análisis. CU Configurar\_Proceso.



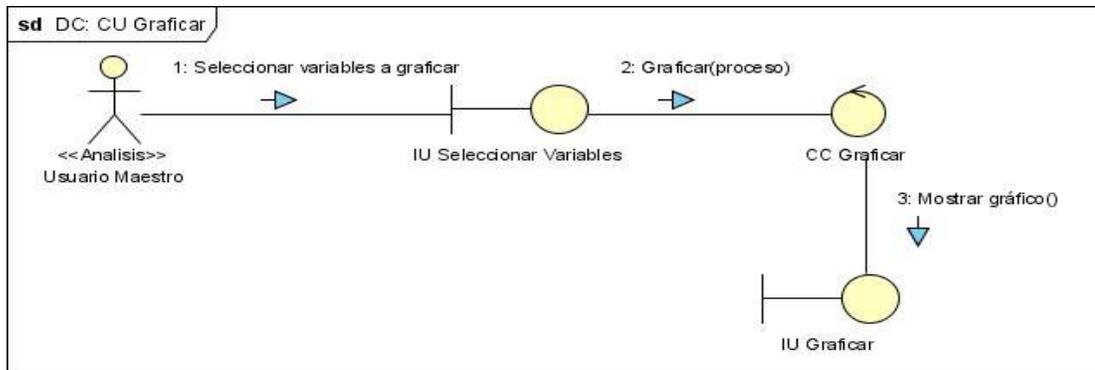
Anexo 21: Diagrama de Clases del Análisis. CU Simular\_Proceso, CU Probar\_Aplicación.



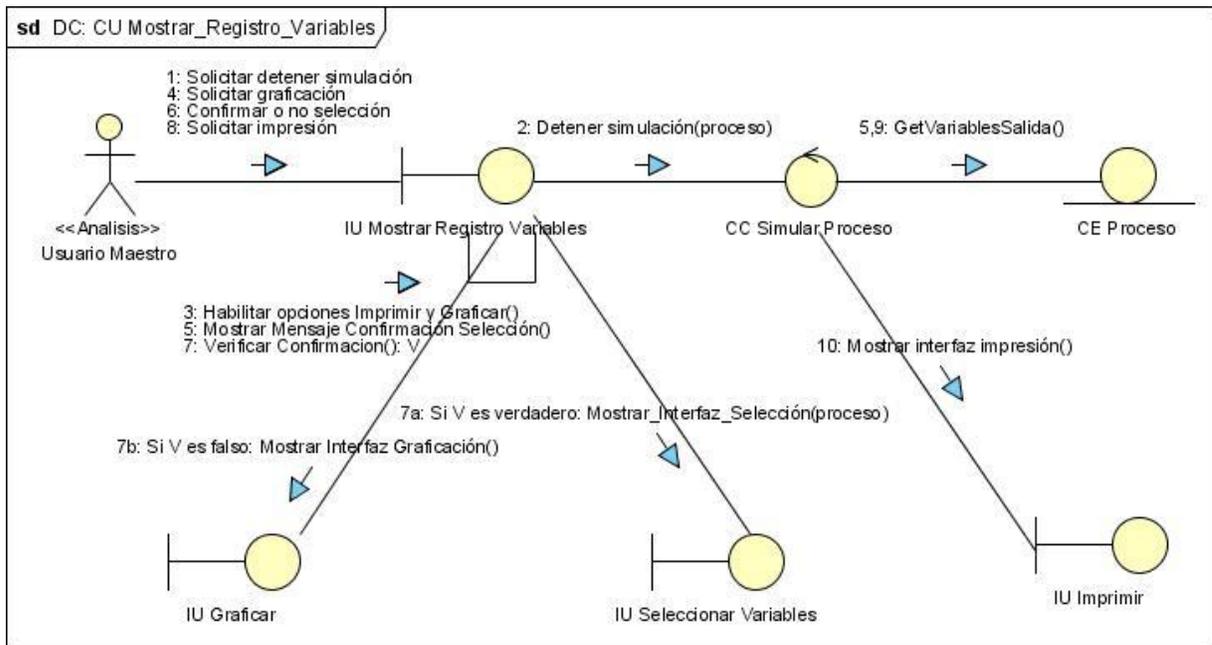
Anexo 22: Diagrama de Clases del Análisis. CU Gestionar\_Controlador.



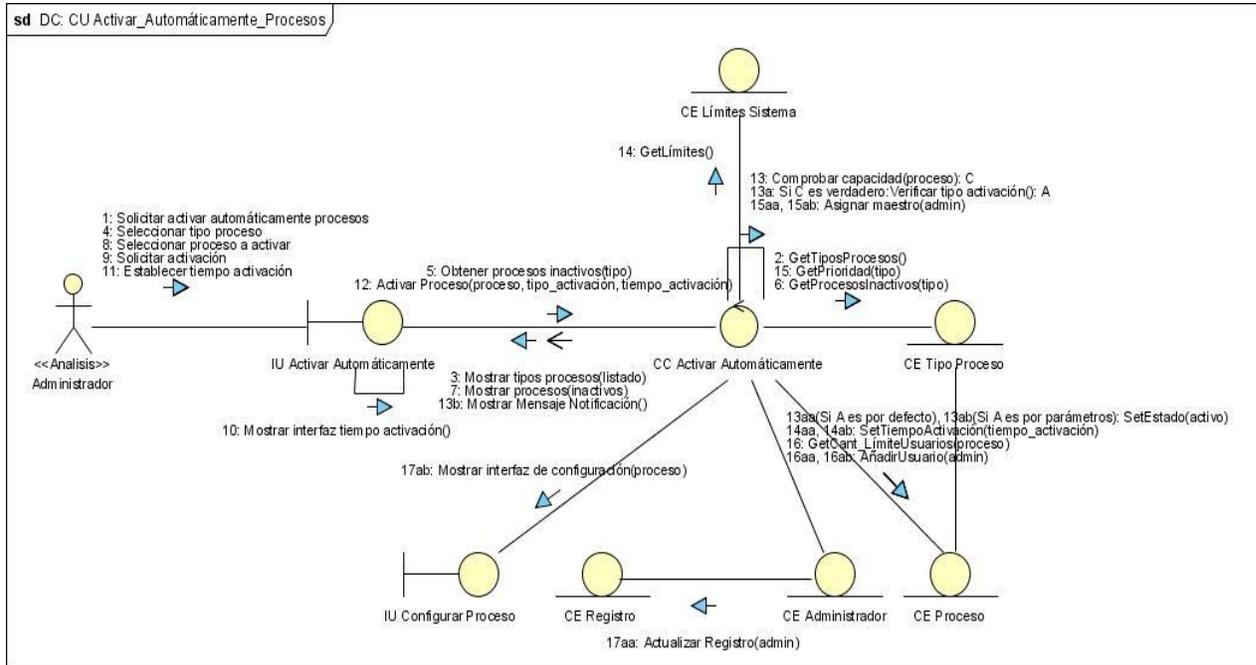
Anexo 23: Diagrama de Colaboración del Análisis. CU Graficar.



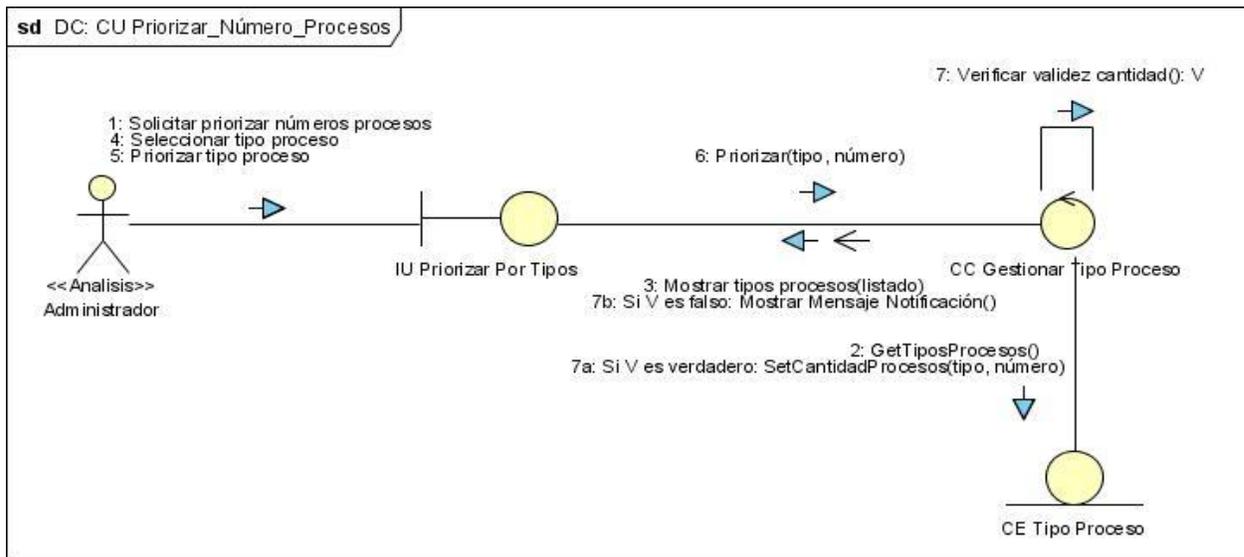
Anexo 24: Diagrama de Colaboración del Análisis. CU Mostrar\_Registro\_Variables.



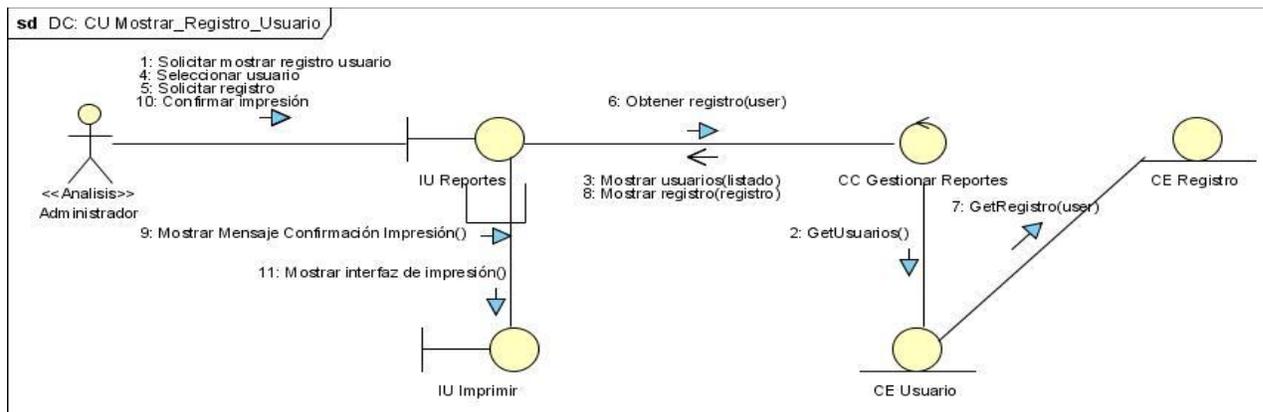
Anexo 25: Diagrama de Colaboración del Análisis. CU Activar\_Automáticamente\_Procesos.



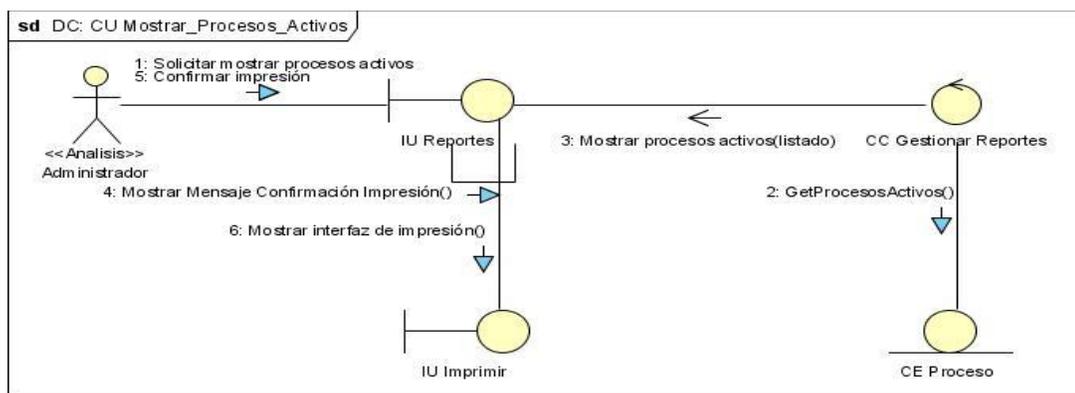
Anexo 26: Diagrama de Colaboración del Análisis. CU Priorizar\_Número\_Procesos.



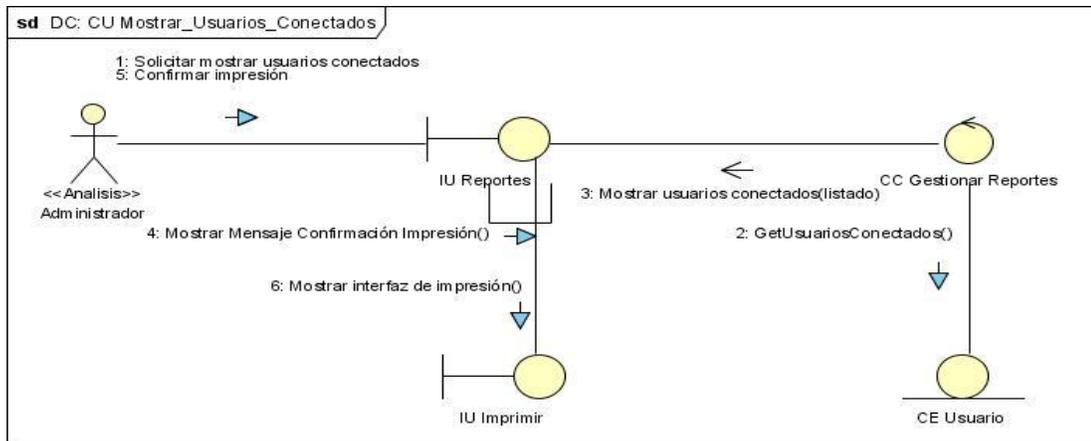
Anexo 27: Diagrama de Colaboración del Análisis. CU Mostrar\_Registro\_Usuario.



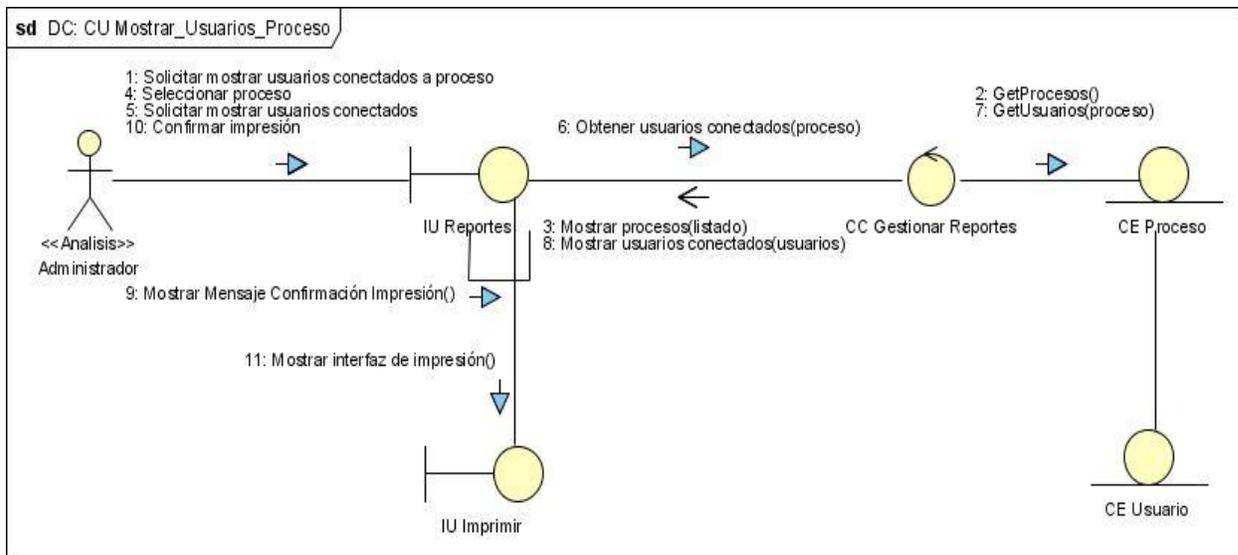
Anexo 28: Diagrama de Colaboración del Análisis. CU Mostrar\_Procesos\_Activos.



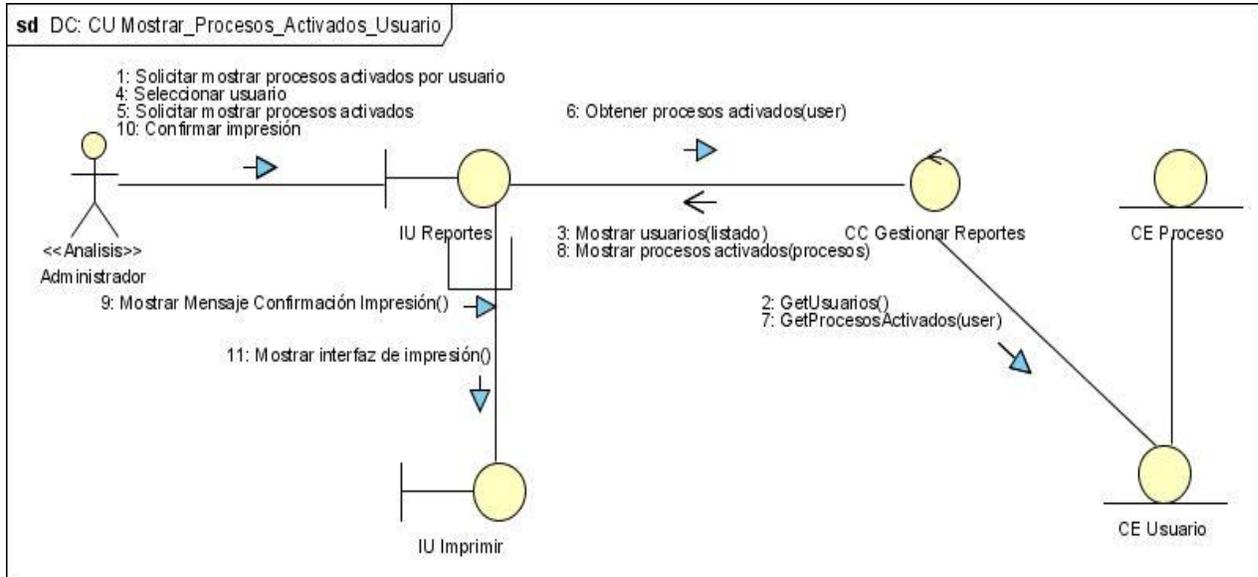
Anexo 29: Diagrama de Colaboración del Análisis. CU Mostrar\_Usuarios\_Conectados.



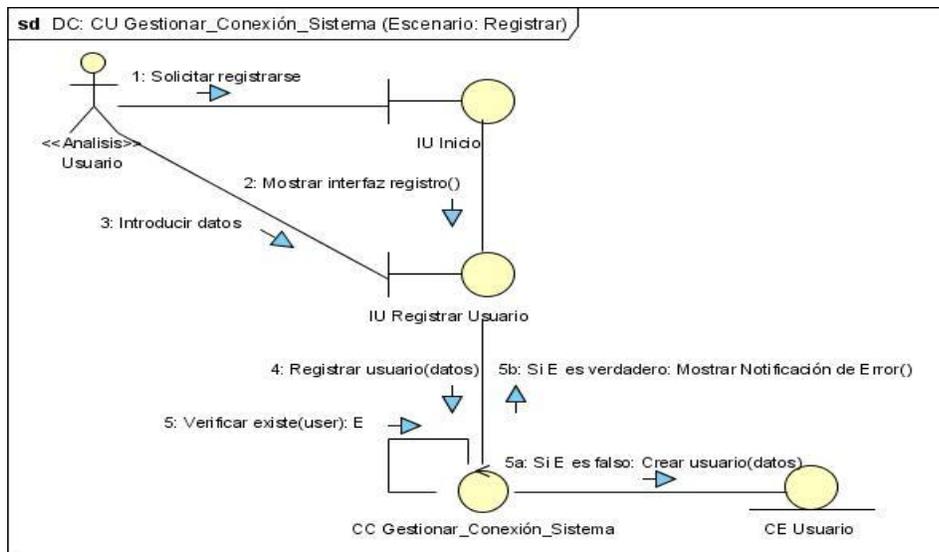
Anexo 30: Diagrama de Colaboración del Análisis. CU Mostrar\_Usuarios\_Proceso.



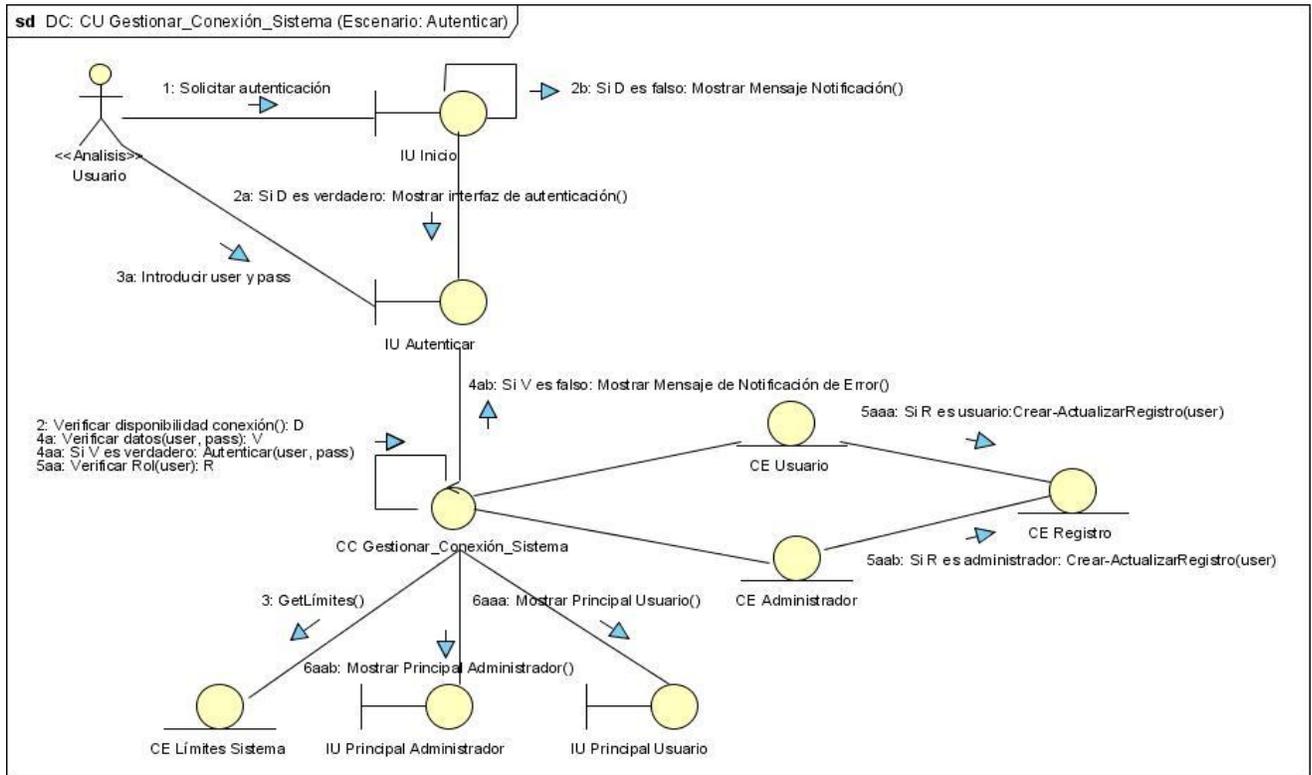
Anexo 31: Diagrama de Colaboración del Análisis. CU Mostrar\_Procesos\_Activados\_Usuario.



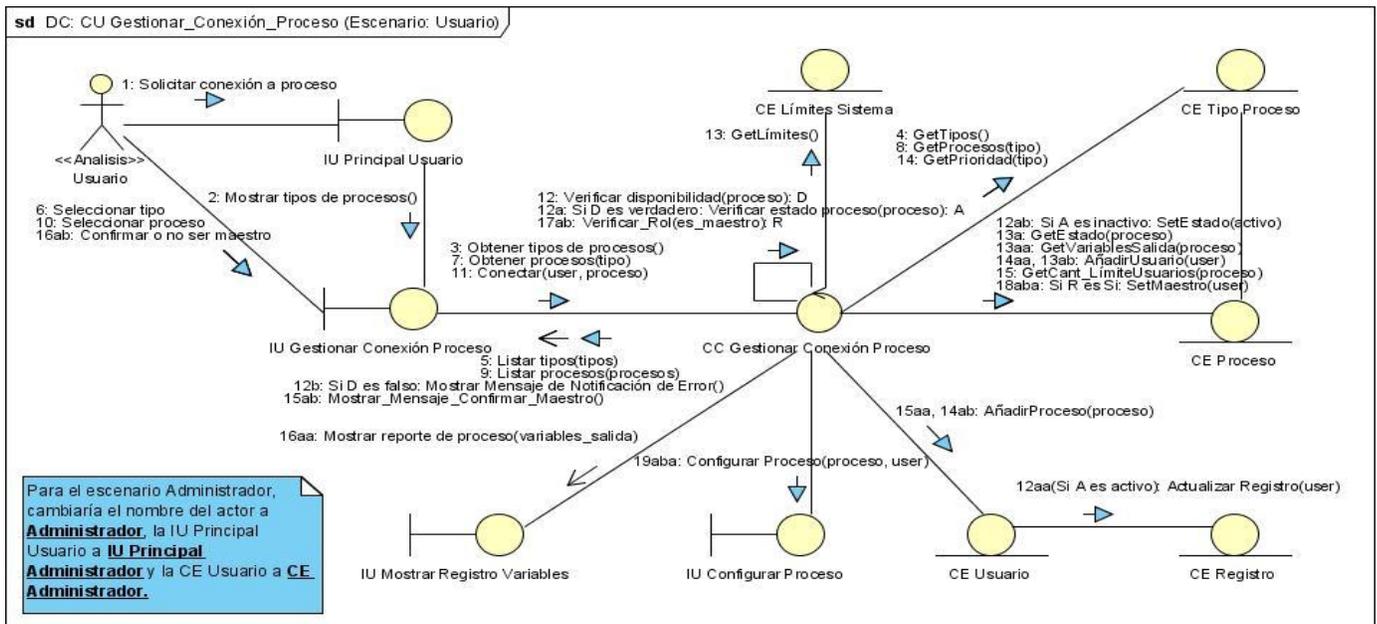
Anexo 32: Diagrama de Colaboración del Análisis. CU Gestionar\_Conexión\_Sistema.



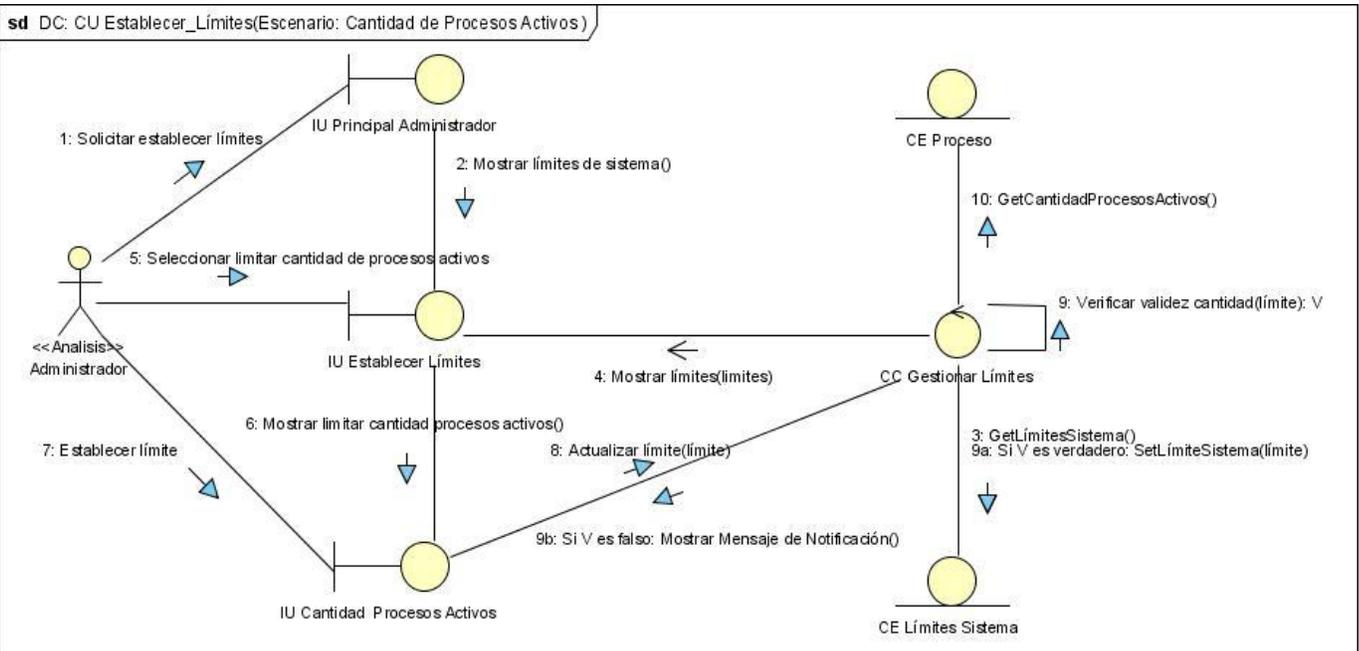
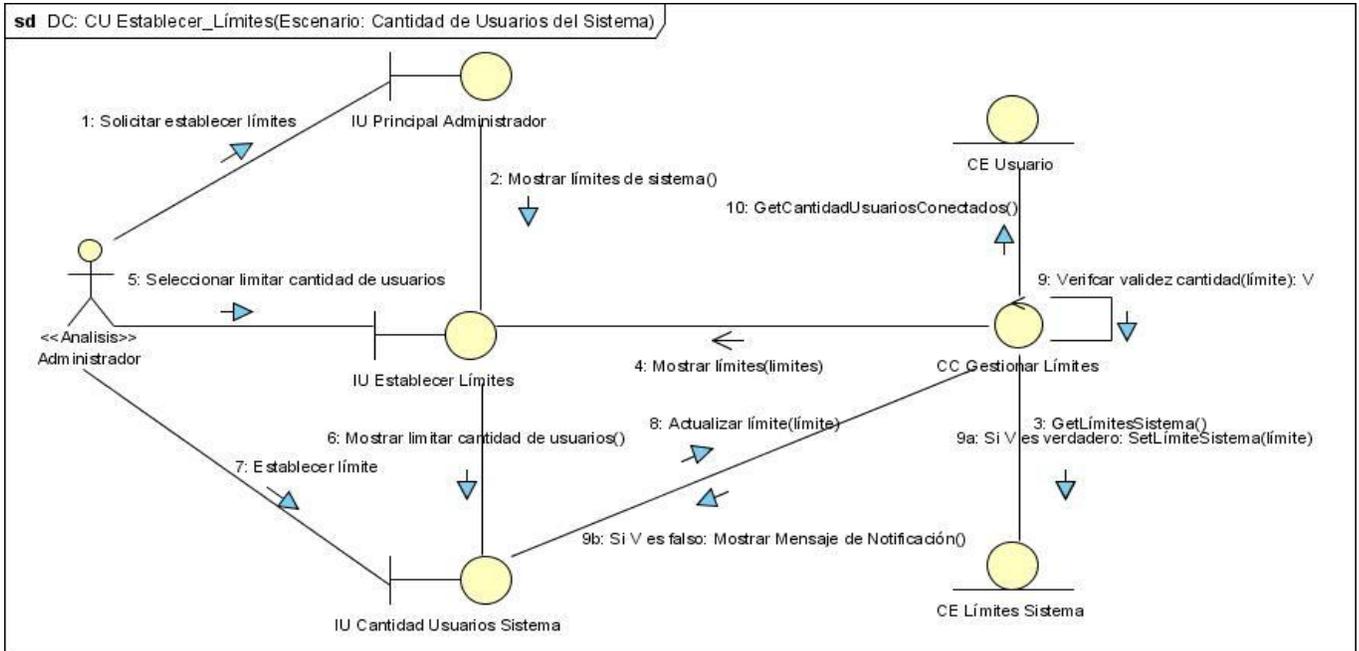
Análisis y Diseño de una herramienta interactiva de simulación de procesos: Nodo Virtual de Procesos



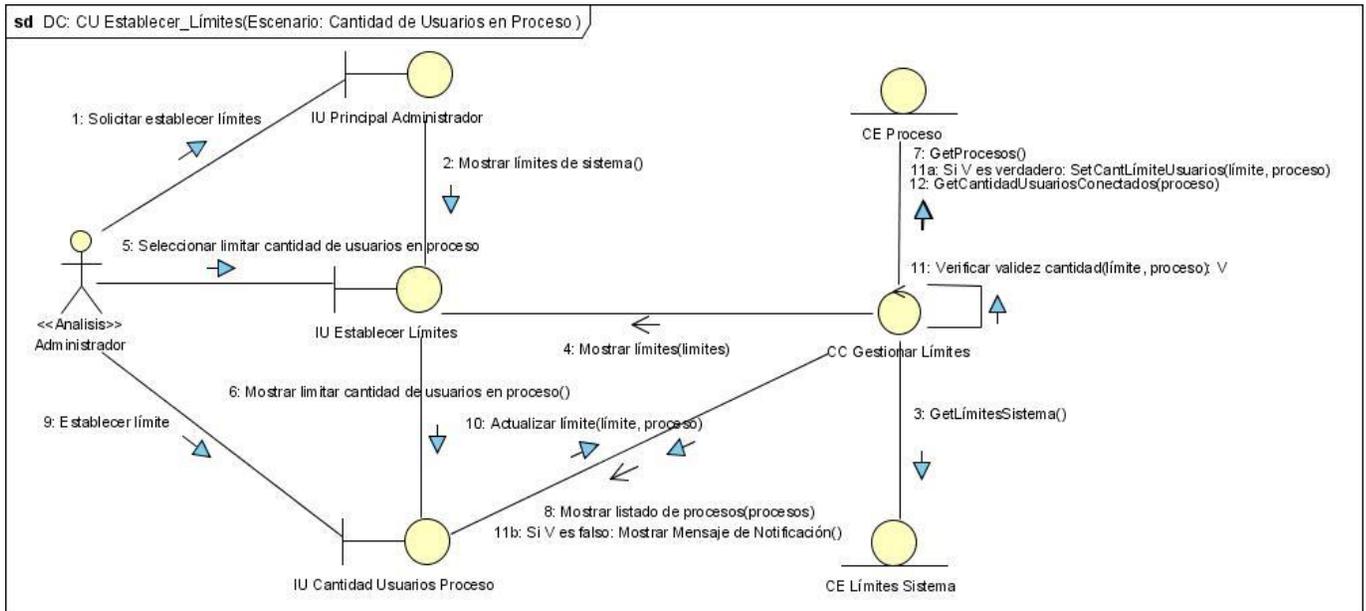
Anexo 33: Diagrama de Colaboración del Análisis. CU Gestionar\_Conexión\_Proceso.



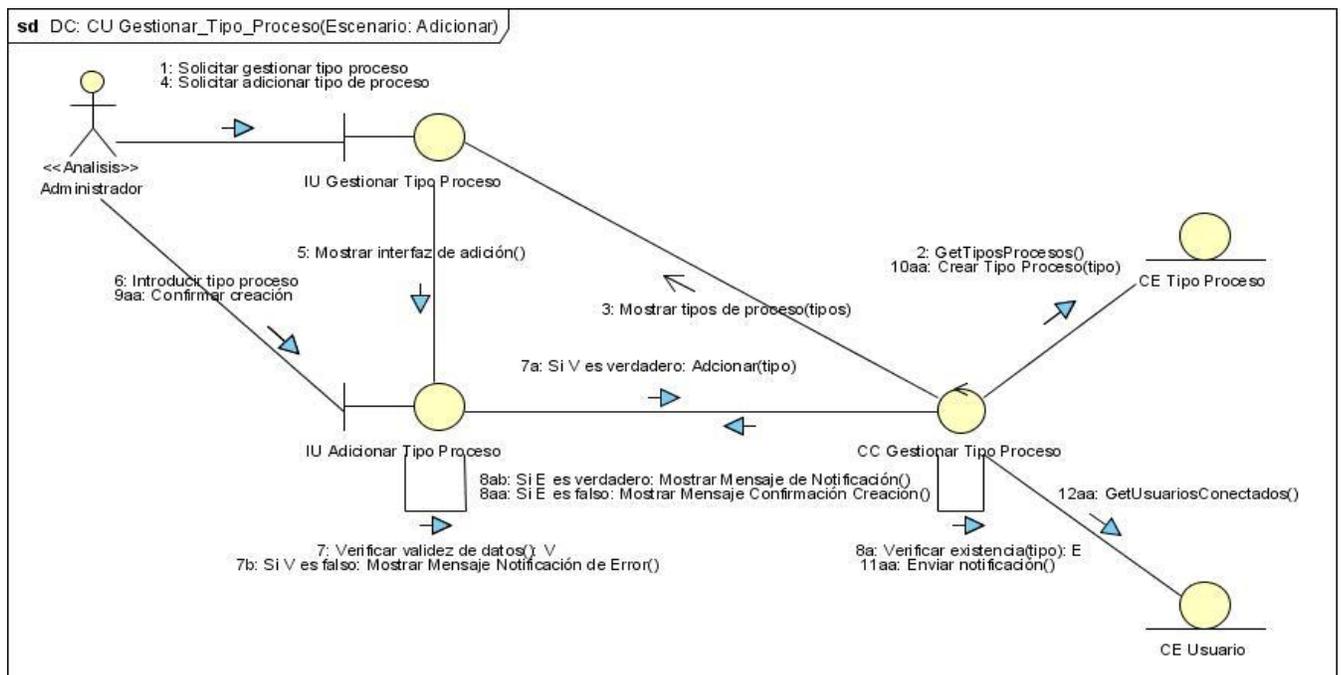
Anexo 34: Diagrama de Colaboración del Análisis. CU Establecer\_Límites.

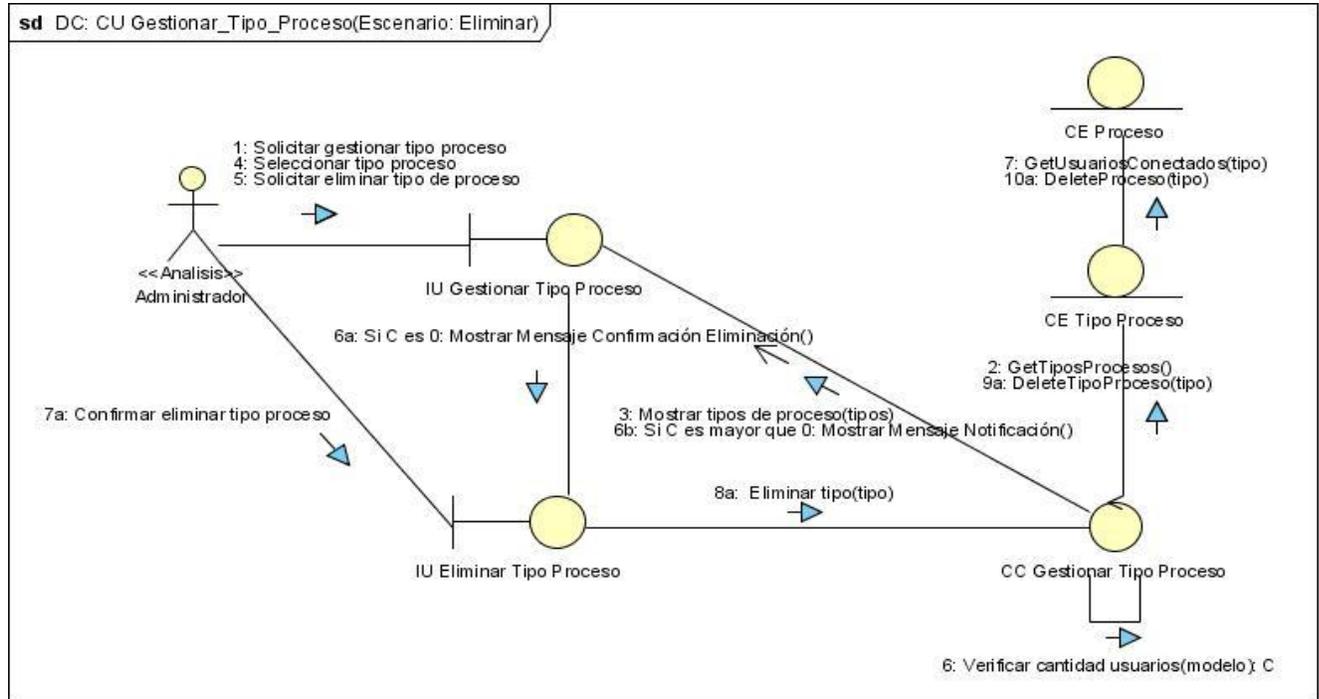
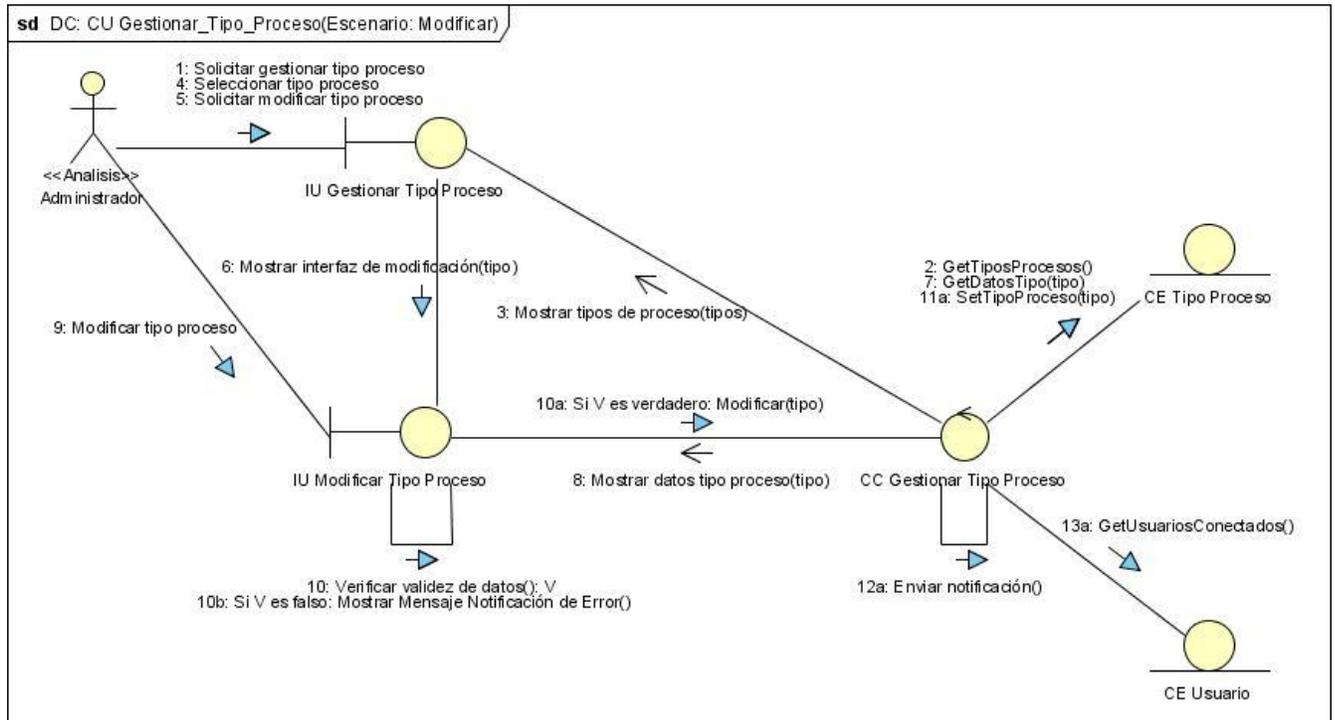


Análisis y Diseño de una herramienta interactiva de simulación de procesos: Nodo Virtual de Procesos

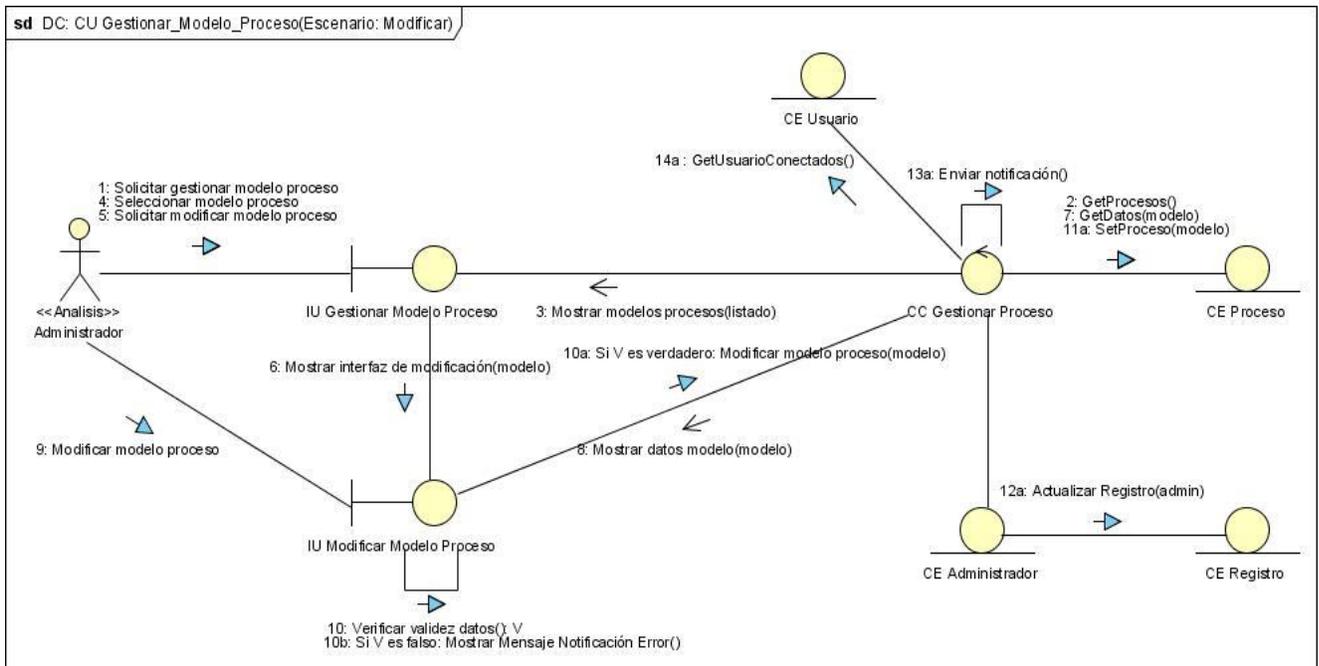
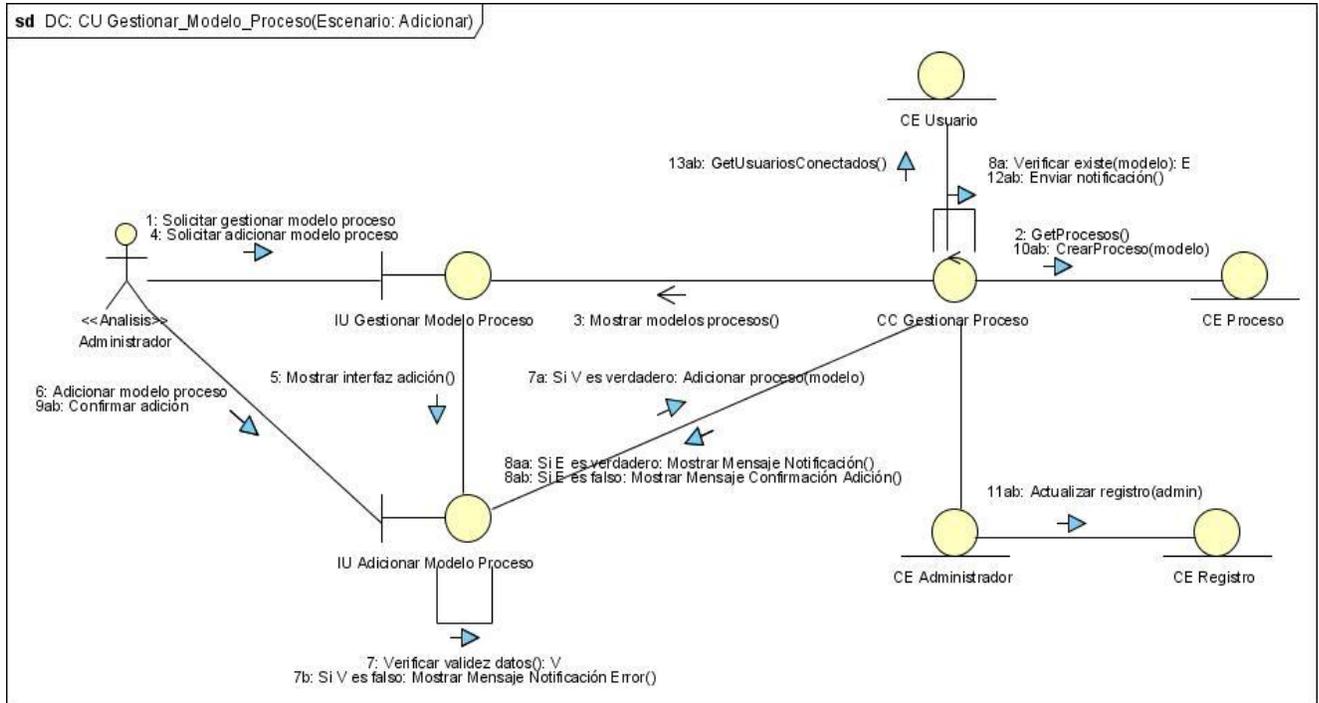


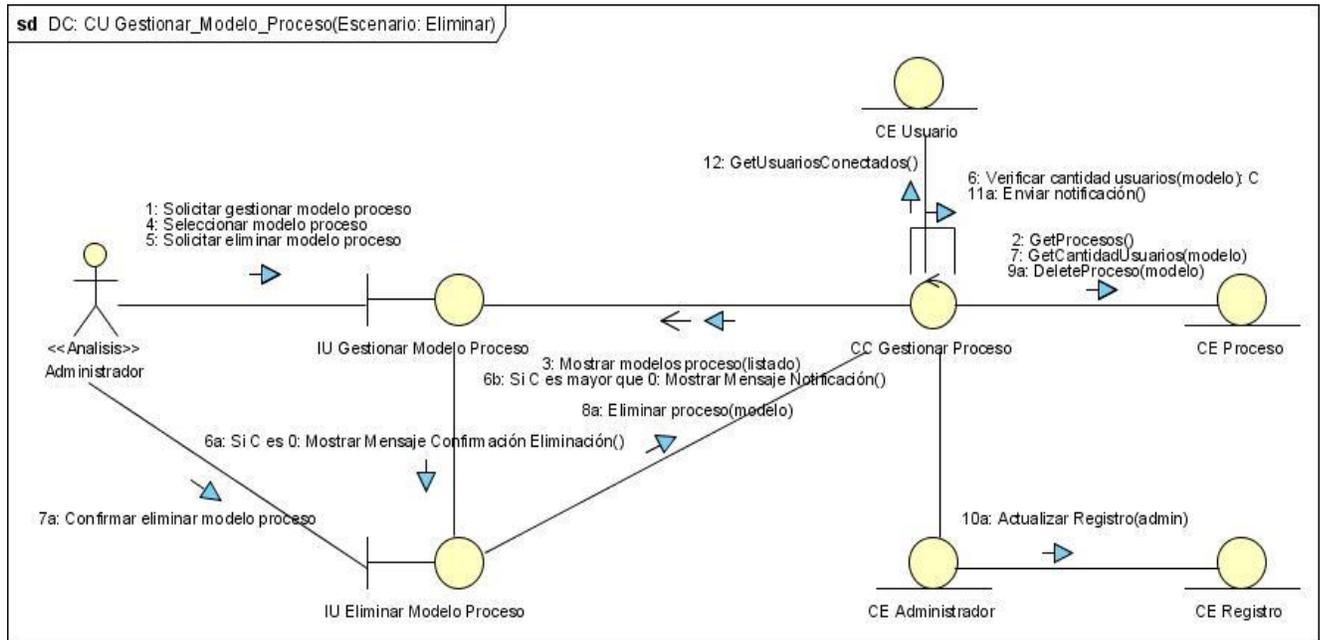
Anexo 35: Diagrama de Colaboración del Análisis. CU Gestionar\_Tipo\_Proceso.



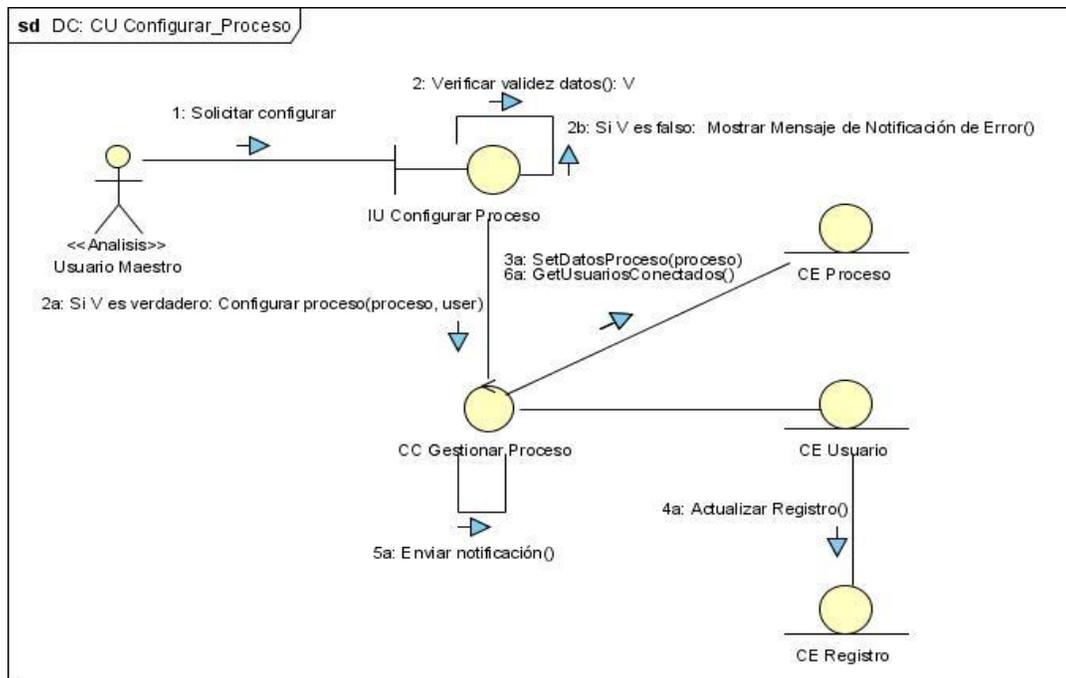


Anexo 36: Diagrama de Colaboración del Análisis. CU Gestionar\_Modelo\_Proceso.

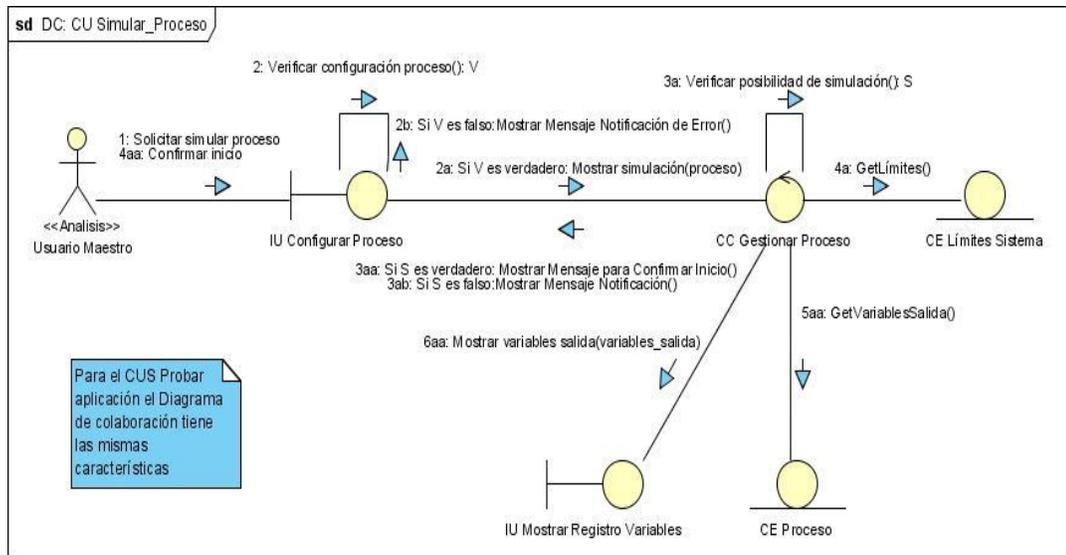




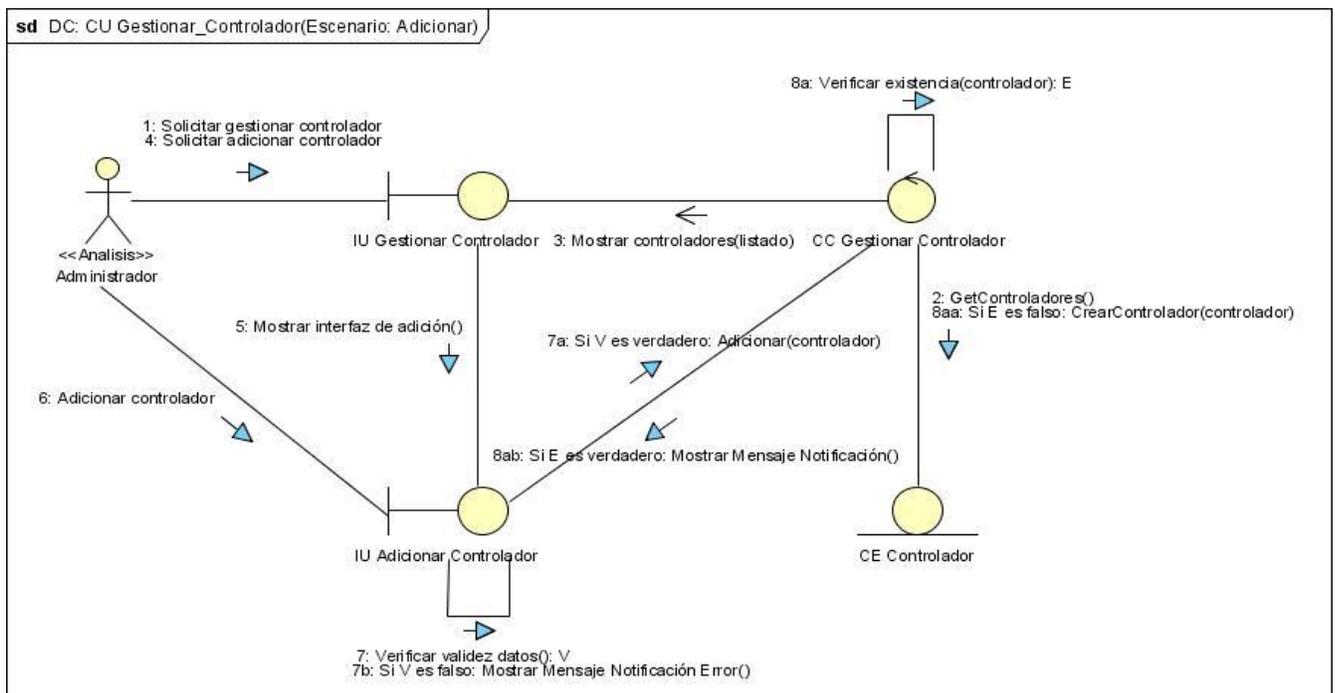
Anexo 37: Diagrama de Colaboración del Análisis. CU Configurar\_Proceso.

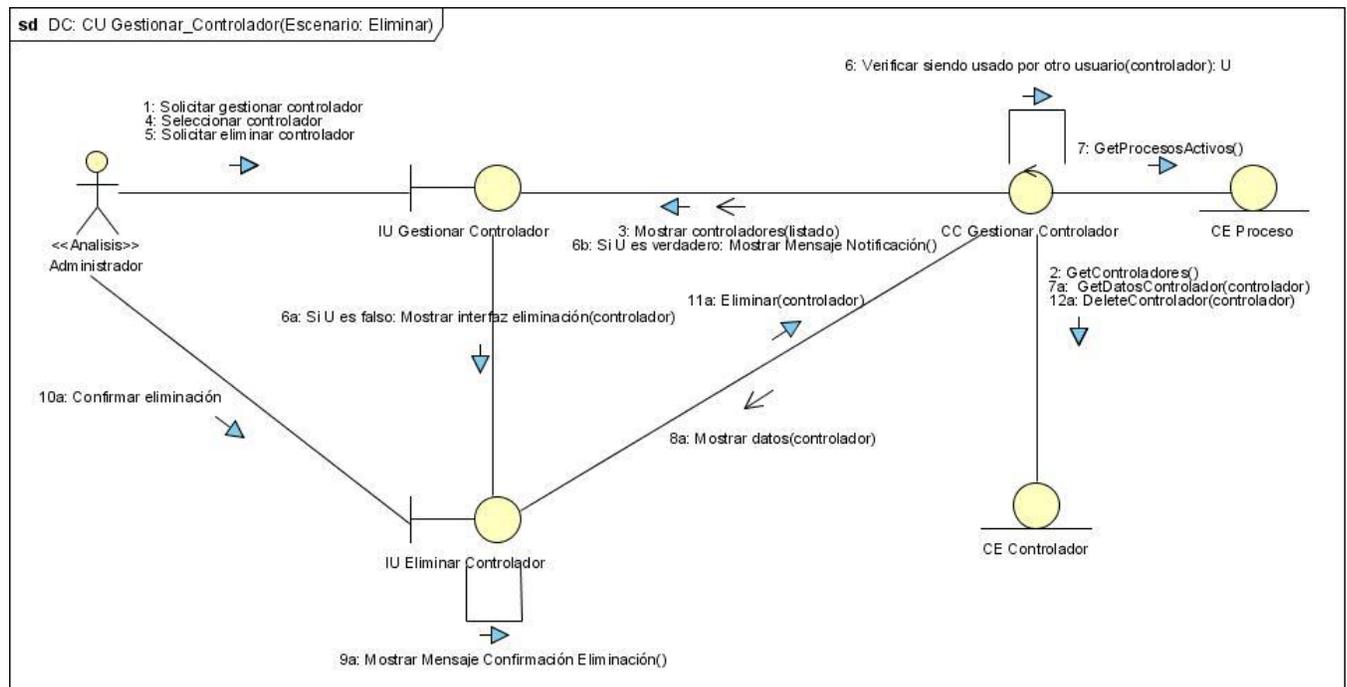
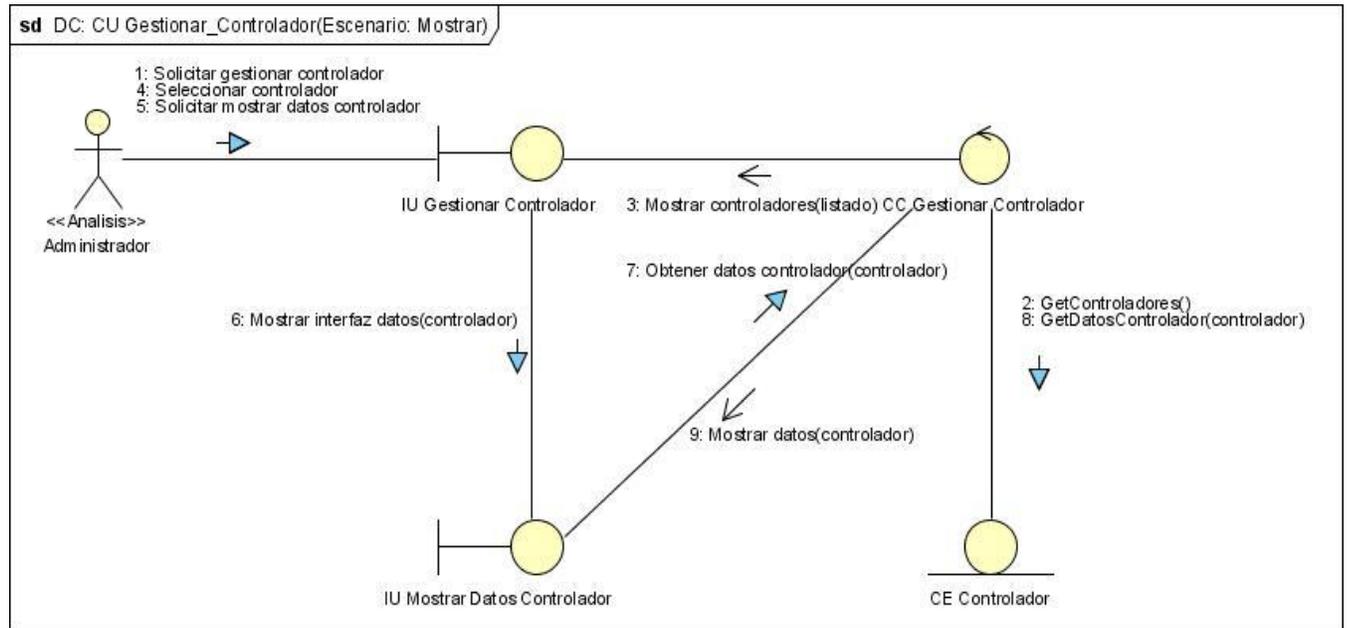


Anexo 38: Diagrama de Colaboración del Análisis. CU Simular\_Proceso.

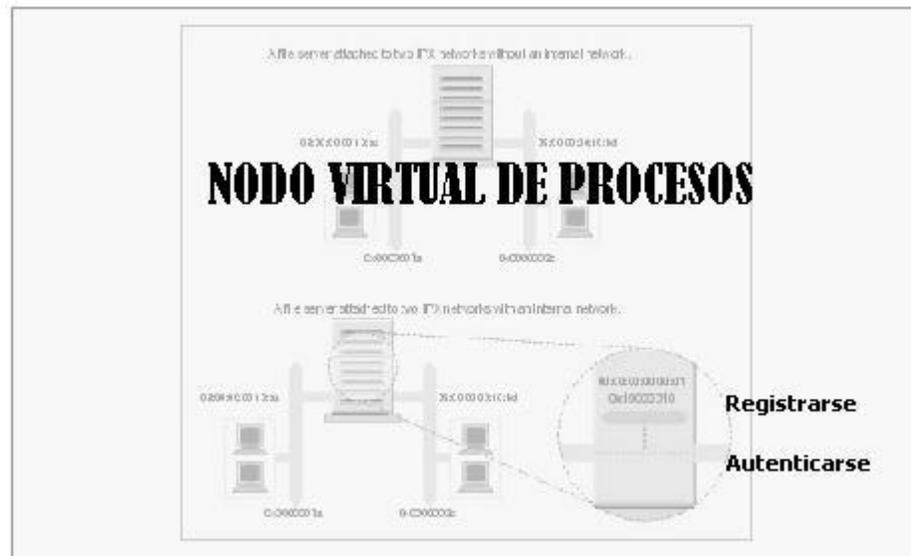


Anexo 39: Diagrama de Colaboración del Análisis. CU Gestionar\_Controlador.

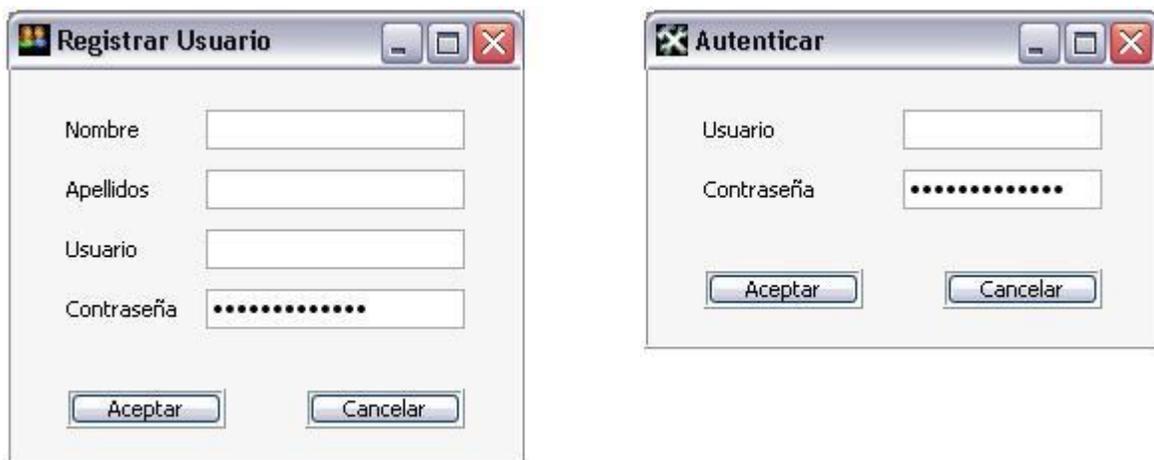




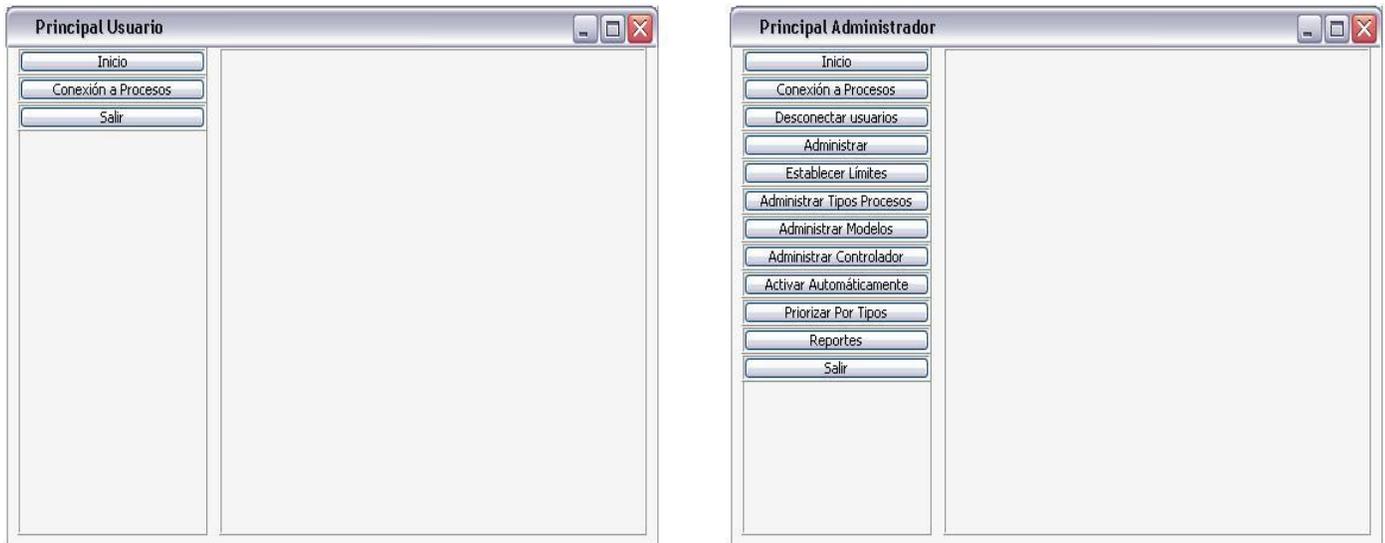
Anexo 40: Interfaz de Usuario. IU\_Inicio.



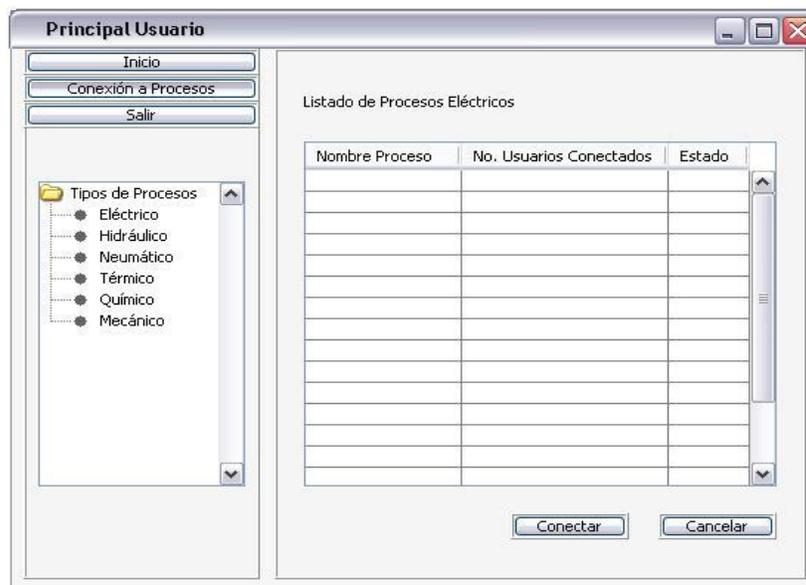
Anexo 41: Interfaz de Usuario. IU\_Registrar\_Usuario, IU\_Autenticar.



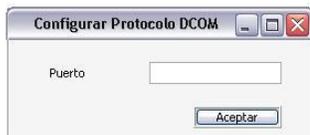
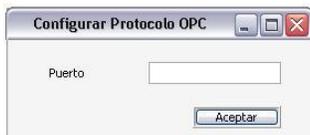
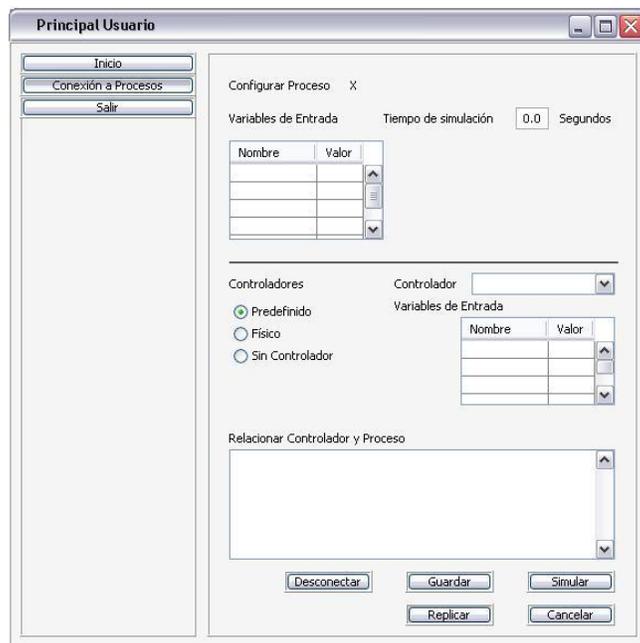
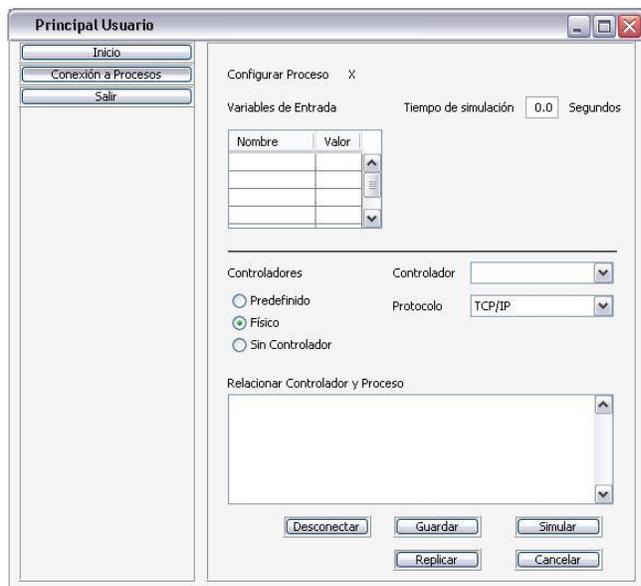
Anexo 42: Interfaz de Usuario. IU\_Principal\_Usuario, IU\_Principal\_Administrador.



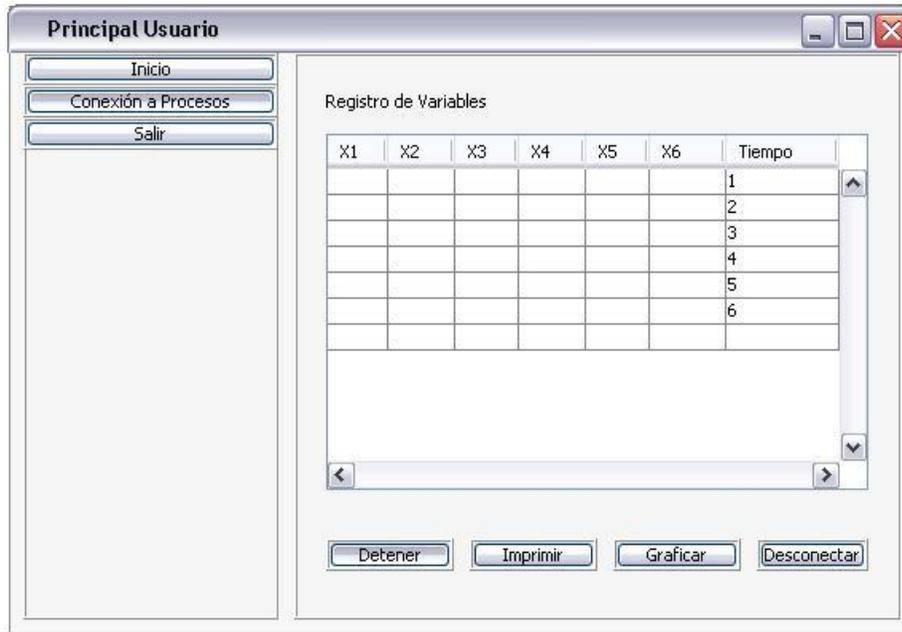
Anexo 43: Interfaz de Usuario. IU\_Gestionar\_Conexión\_Proceso.



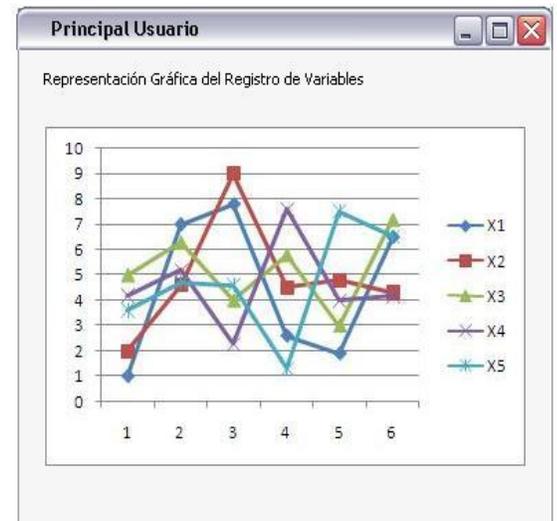
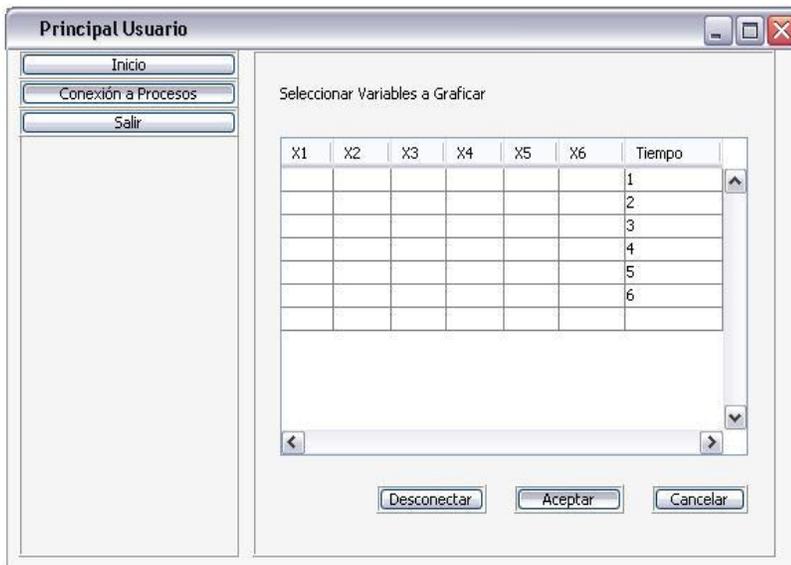
Anexo 44: Interfaz de Usuario. IU\_Configurar\_Proceso.



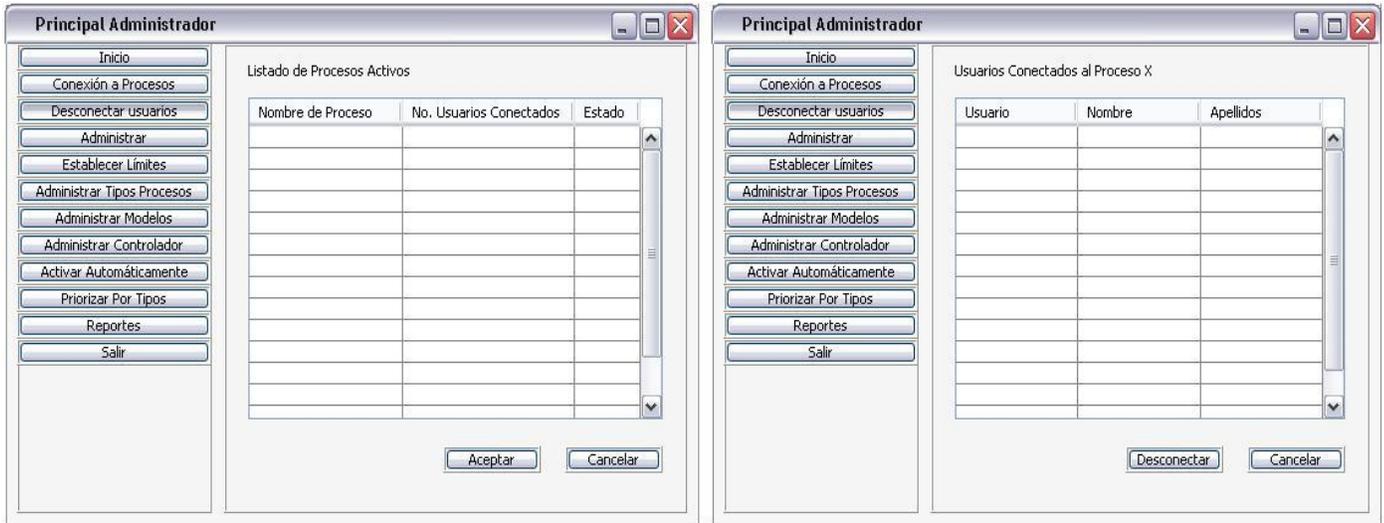
Anexo 45: Interfaz de Usuario. IU\_Mostrar\_Registro\_Variables.



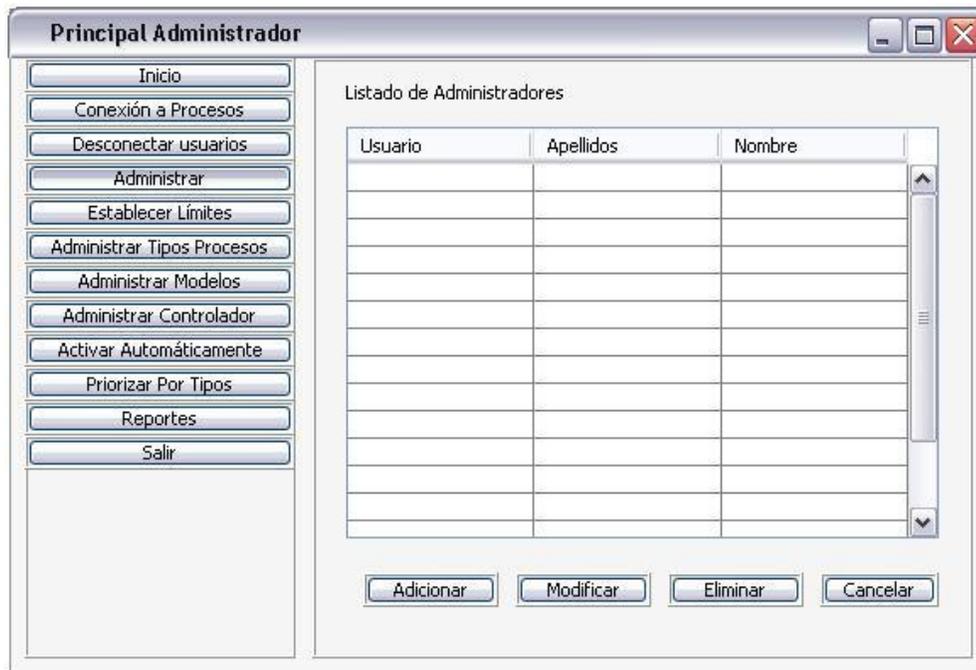
Anexo 46: Interfaz de Usuario. IU\_Seleccionar\_Variables, IU\_Graficar.



Anexo 47: Interfaz de Usuario. IU\_Desconectar\_Usuario.



Anexo 48: Interfaz de Usuario. IU\_Gestionar\_Rol\_Administrador.



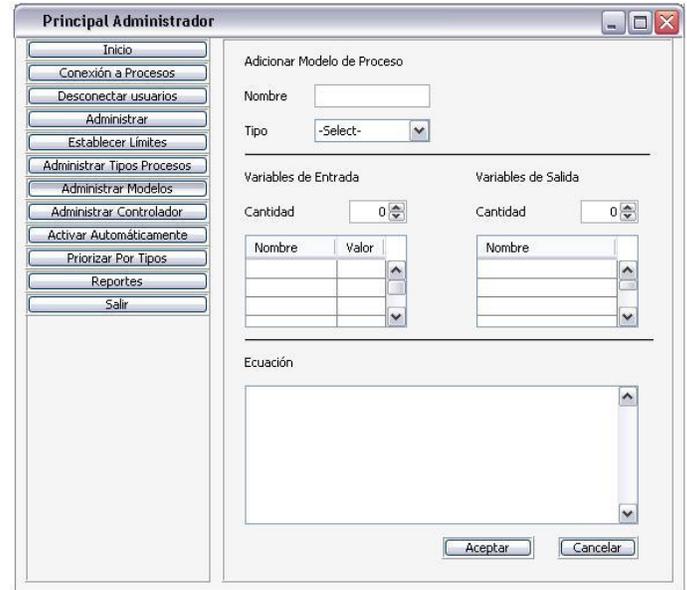
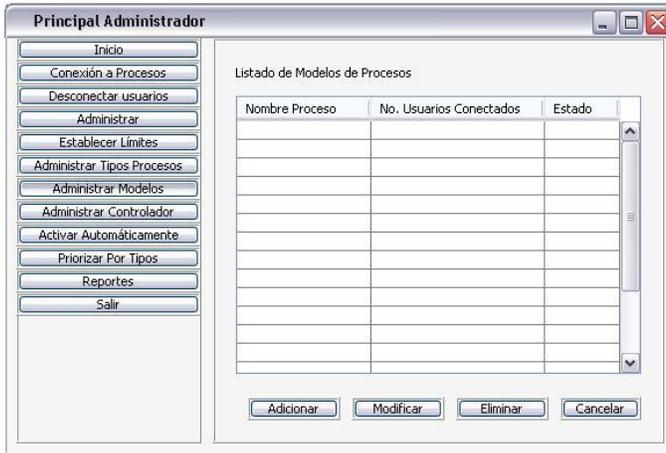
## Anexo 49: Interfaz de Usuario. IU\_Establecer\_Límites.



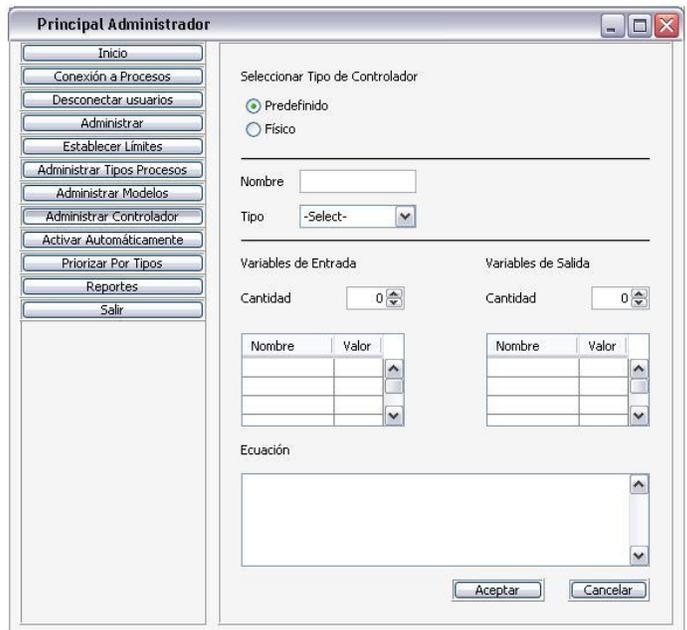
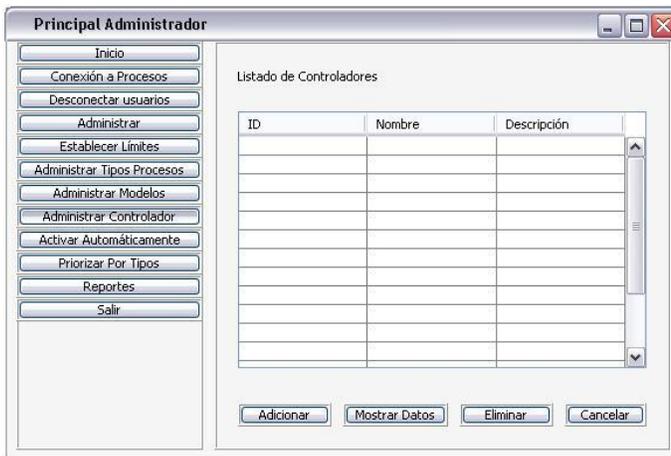
## Anexo 50: Interfaz de Usuario. IU\_Gestionar\_Tipo\_Proceso.



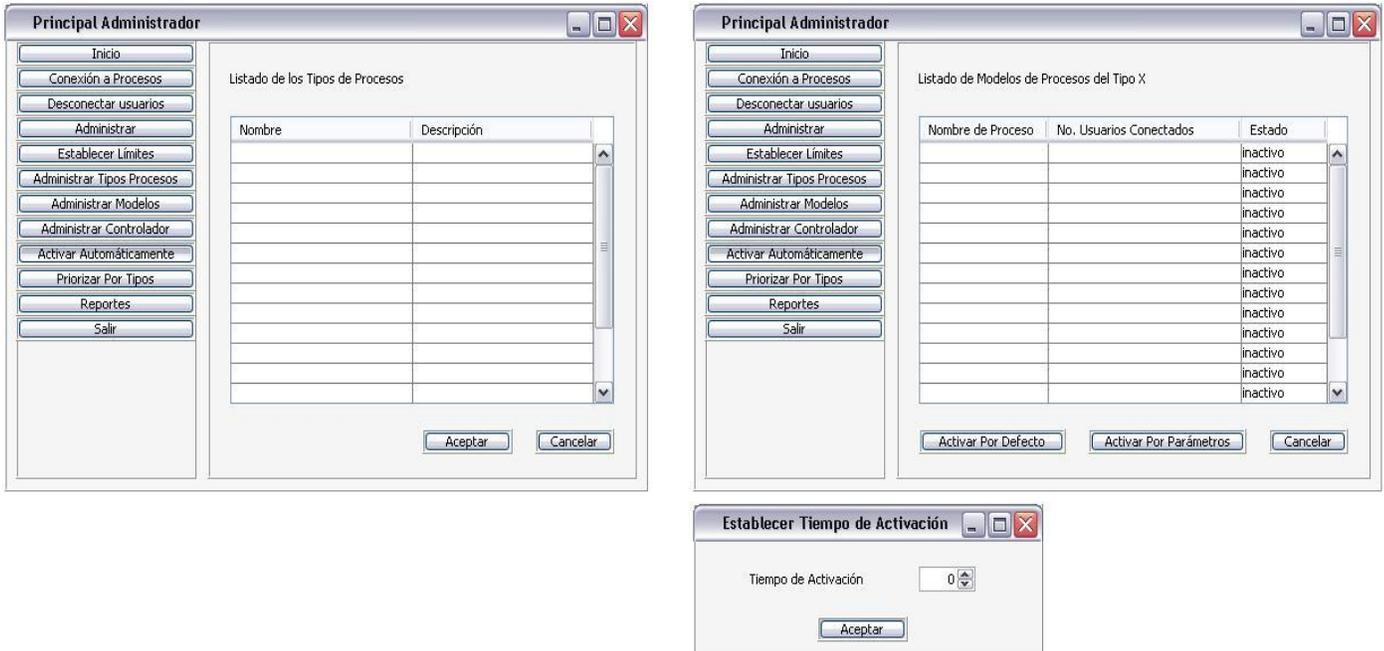
Anexo 51: Interfaz de Usuario. IU\_Gestionar\_Modelo\_Proceso.



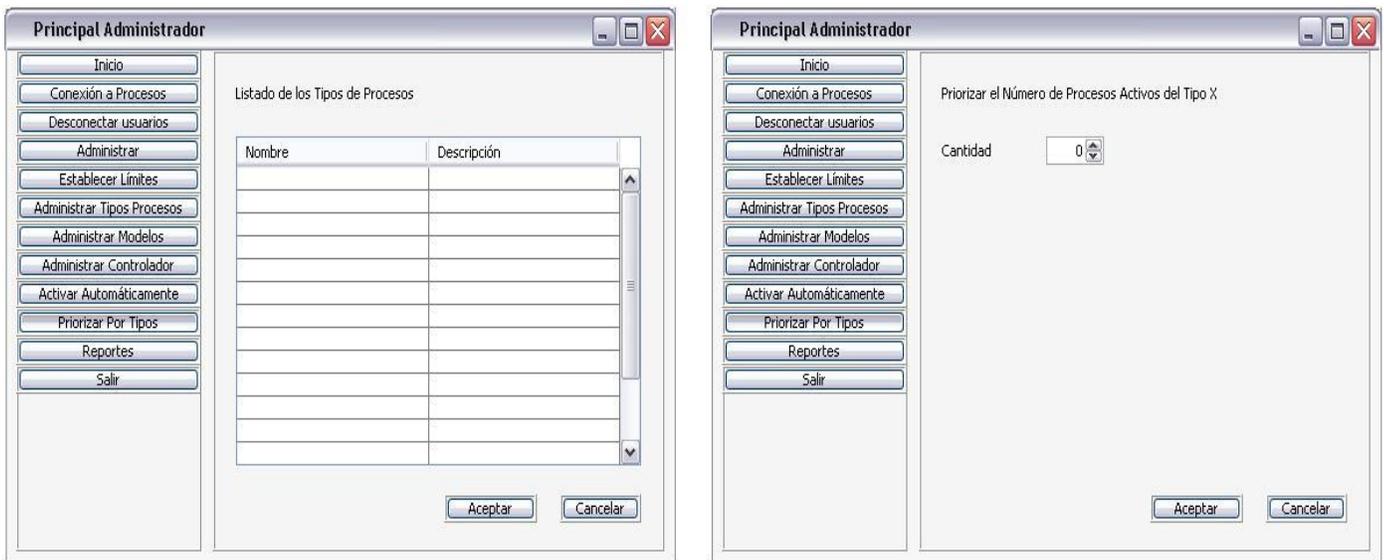
Anexo 52: Interfaz de Usuario. IU\_Gestionar\_Controlador.



Anexo 53: Interfaz de Usuario. IU\_Activar\_Automáticamente.



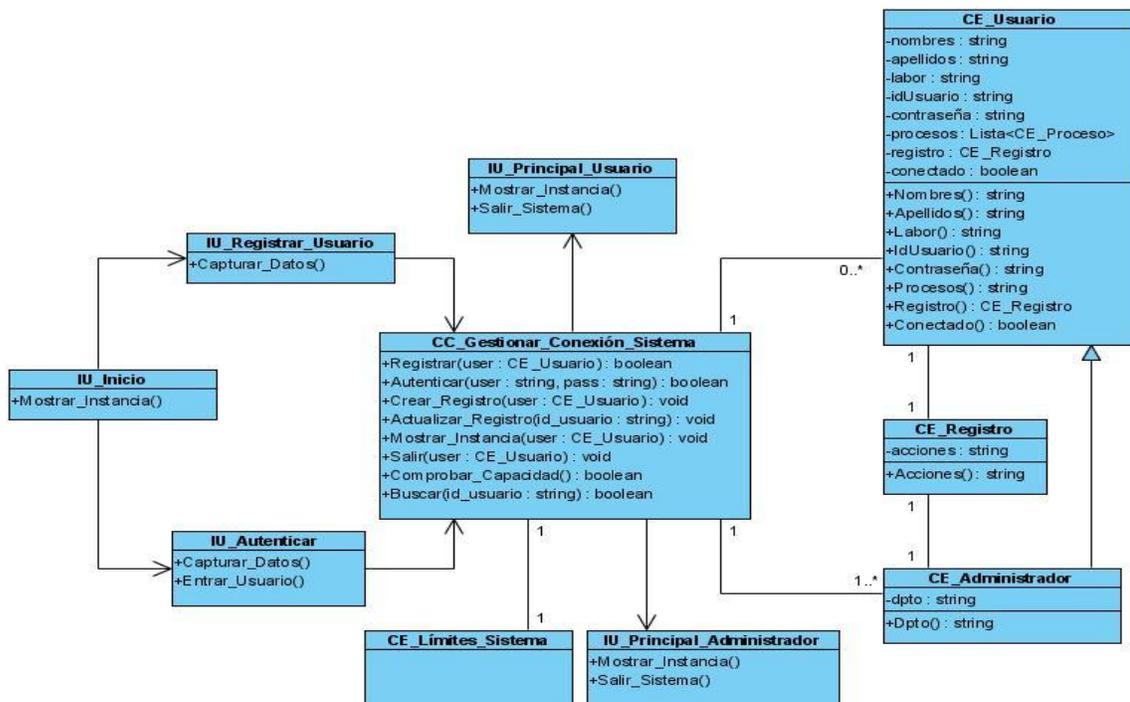
Anexo 54: Interfaz de Usuario. IU\_Priorizar\_Tipos\_Procesos.



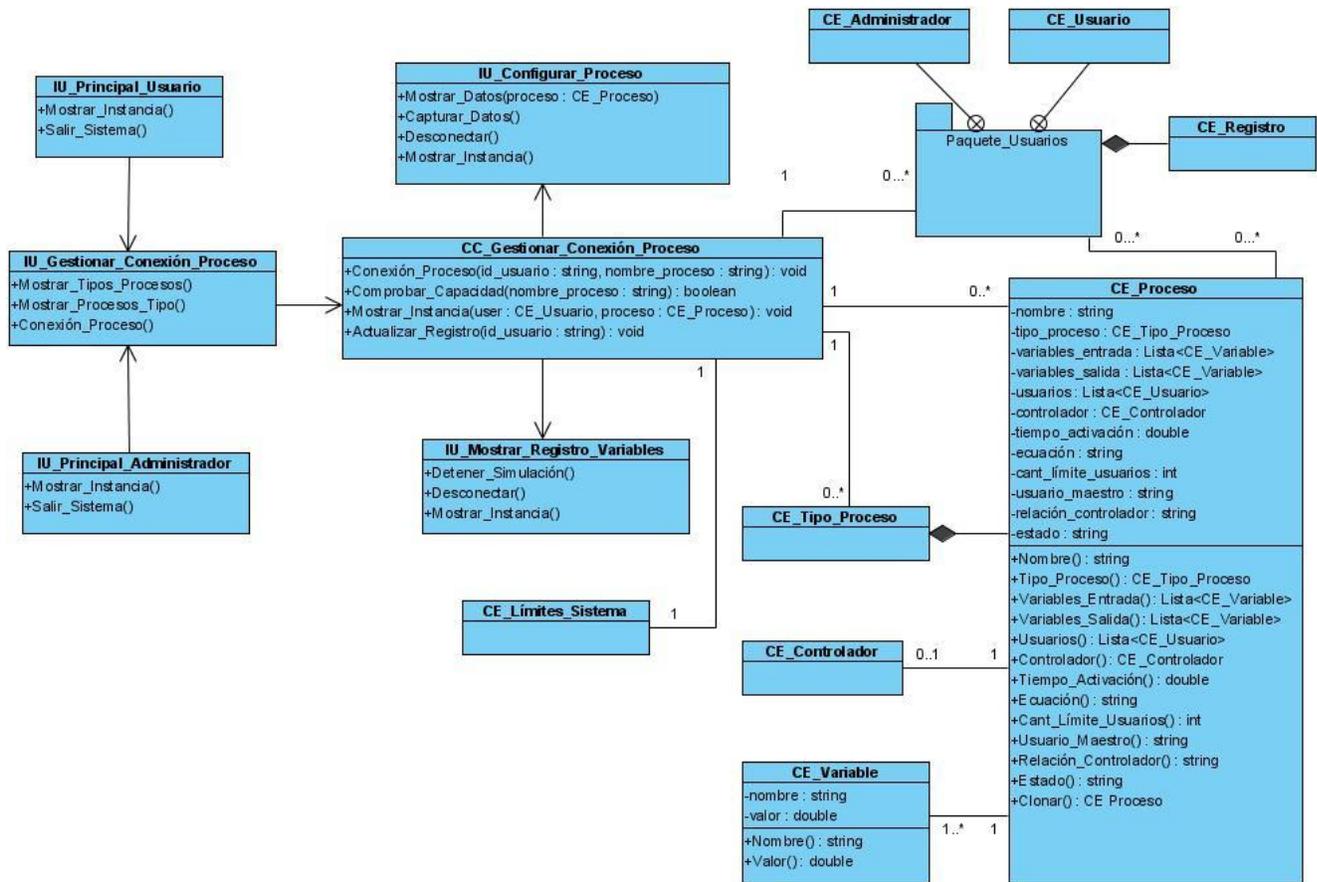
Anexo 55: Interfaz de Usuario. IU\_Reportes.



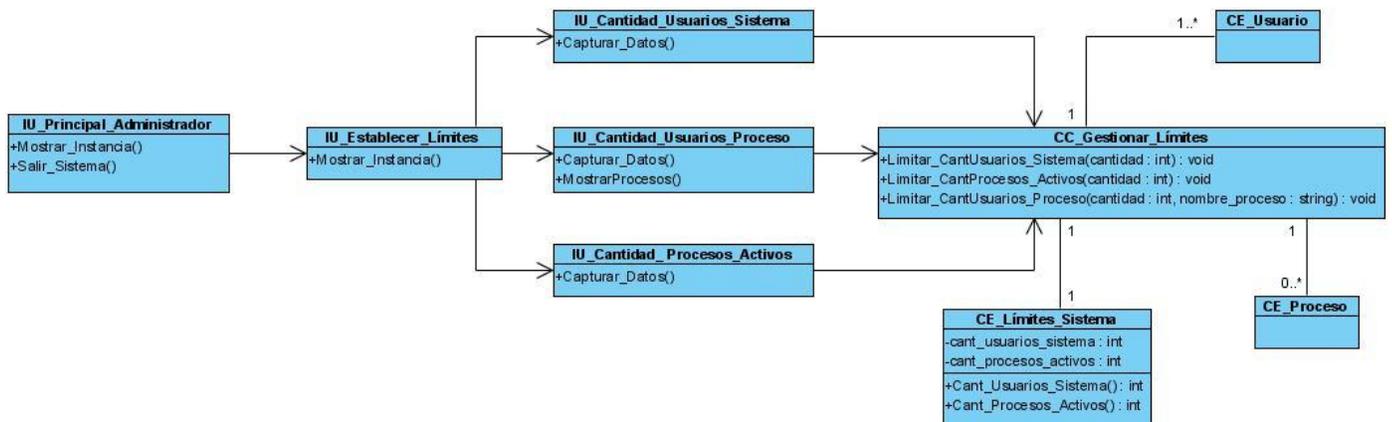
Anexo 56: Diagrama de clases del Diseño. CU Gestionar\_Conexión\_Sistema.



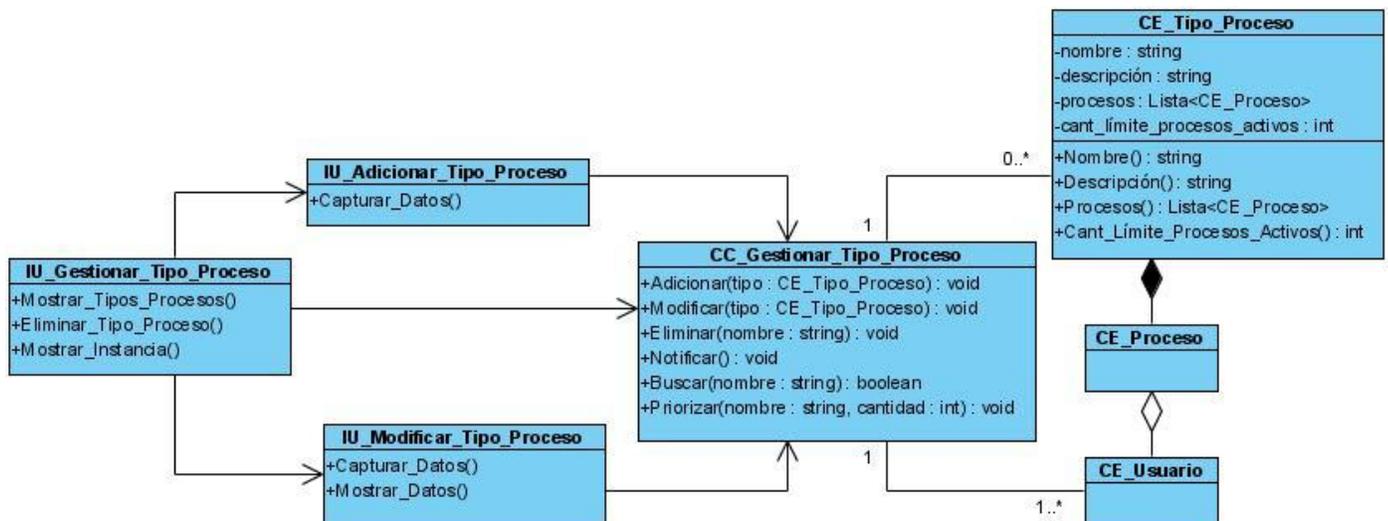
Anexo 57: Diagrama de clases del Diseño. CU Gestionar\_Conexión\_Proceso.



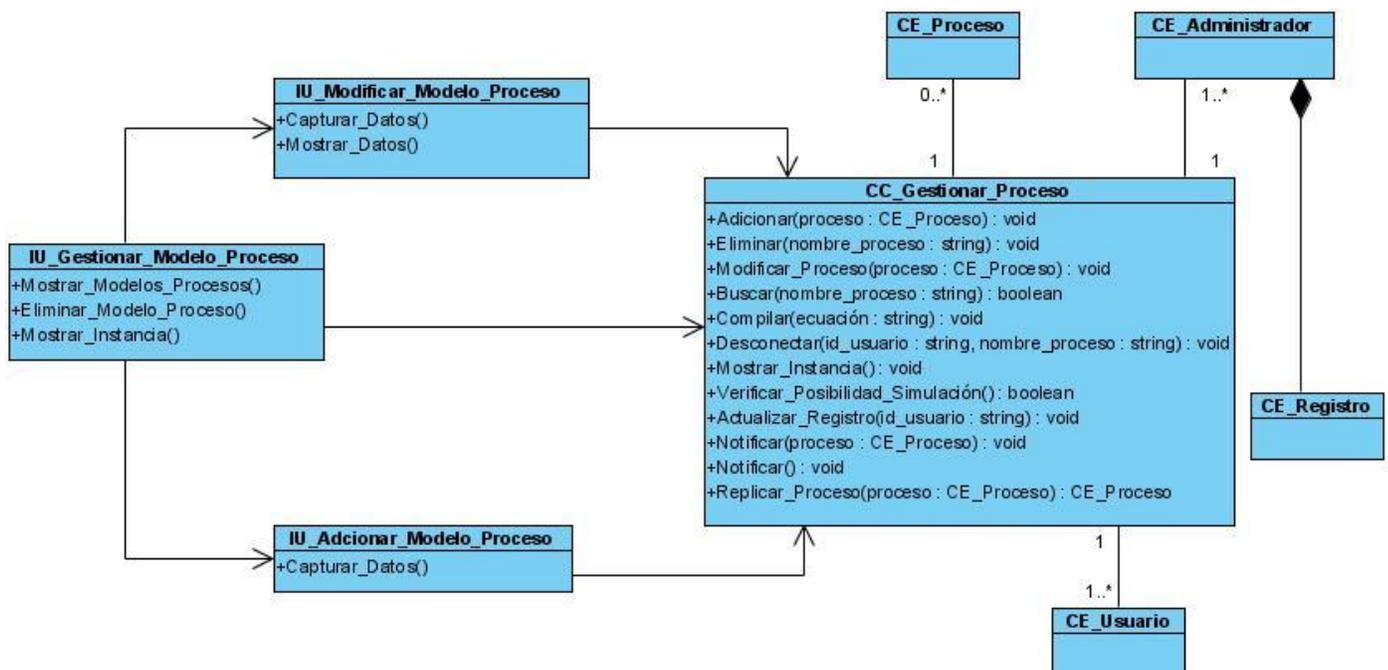
Anexo 58: Diagrama de clases del Diseño. CU Establecer\_Límites.



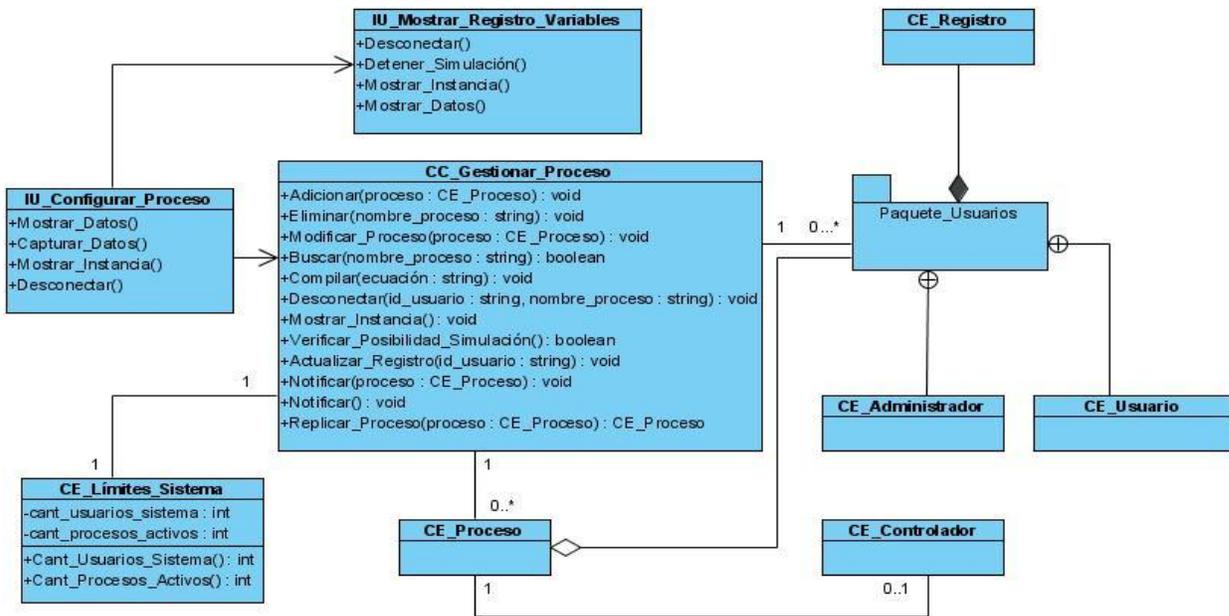
Anexo 59: Diagrama de clases del Diseño. CU Gestionar\_Tipo\_Proceso.



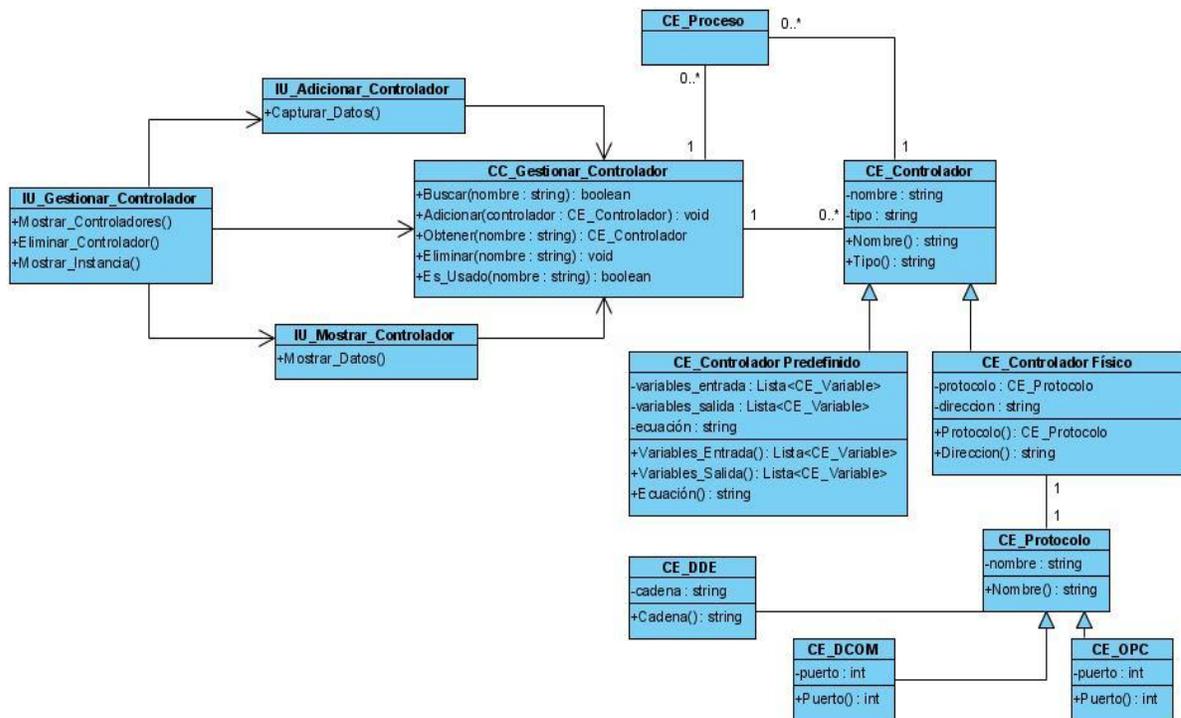
Anexo 60: Diagrama de clases del Diseño. CU Gestionar\_Modelo\_Proceso.



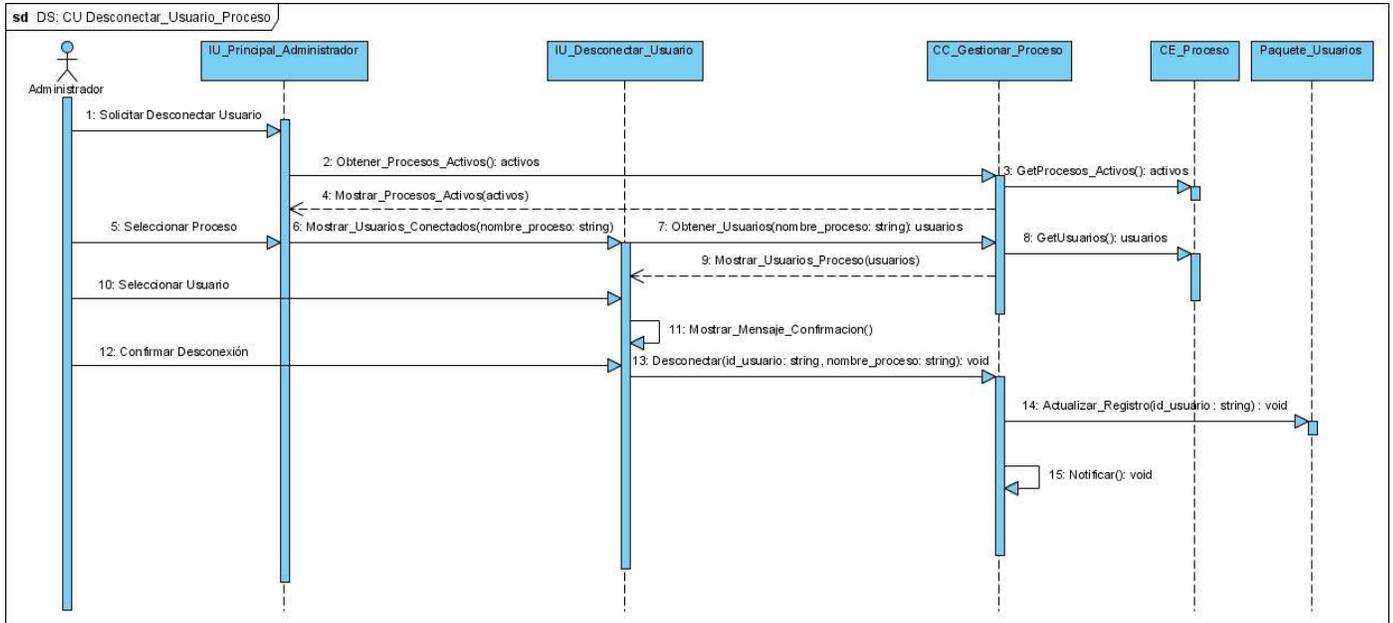
Anexo 61: Diagrama de clases del Diseño. CU Configurar\_Proceso.



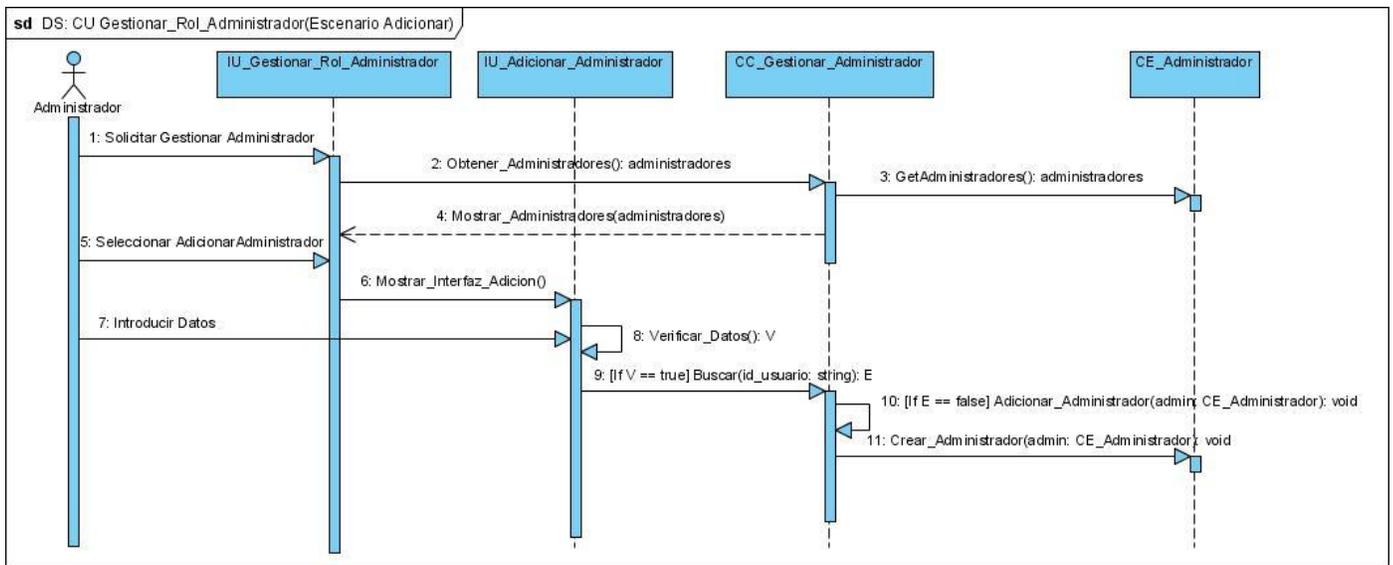
Anexo 62: Diagrama de clases del Diseño. CU Gestionar\_Controlador.

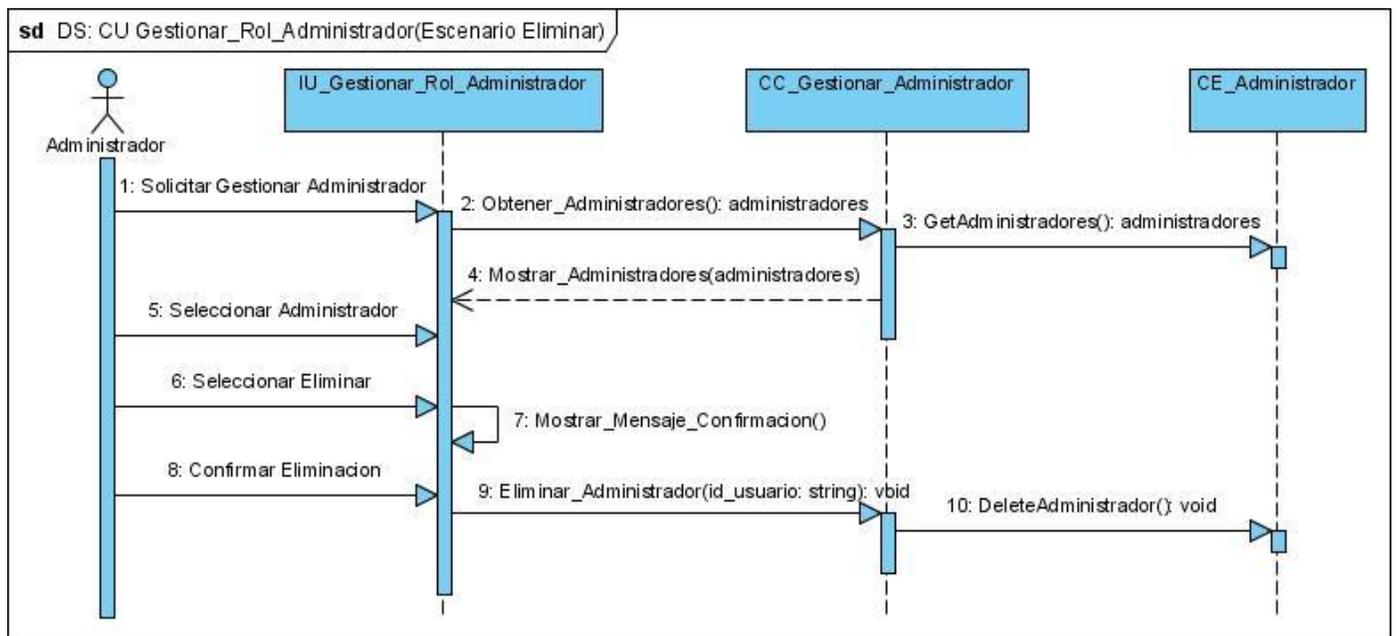
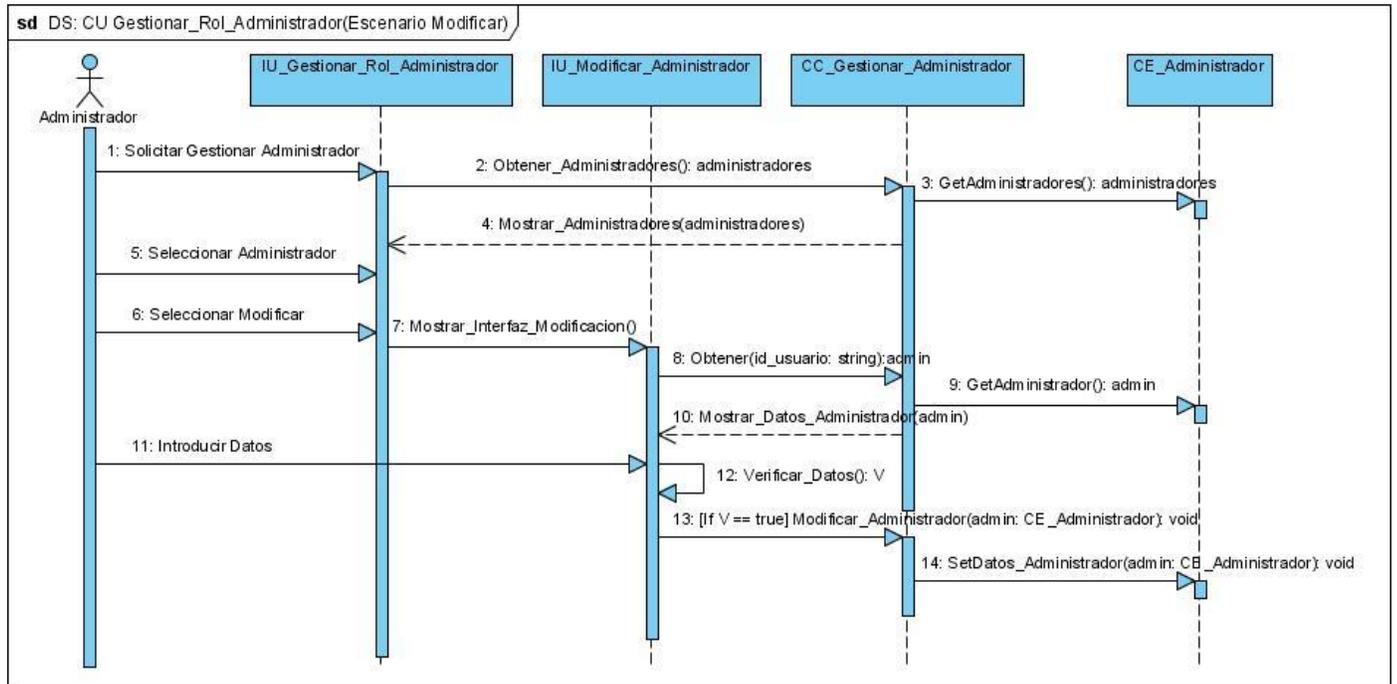


Anexo 63: Diagrama de Secuencia. CU Desconectar\_Usuario\_Proceso.

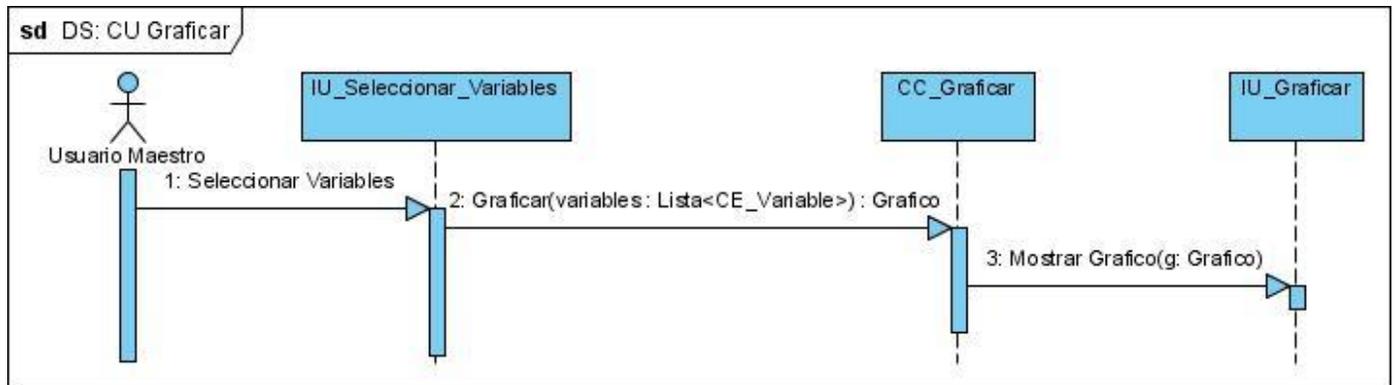


Anexo 64: Diagrama de Secuencia. CU Gestionar\_Rol\_Administrador.

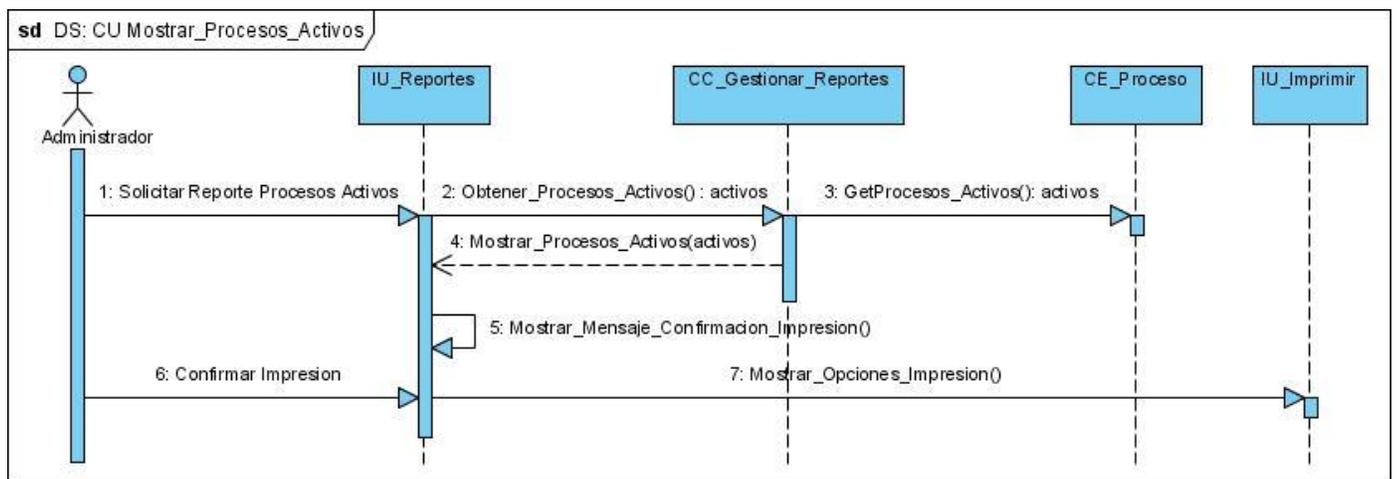




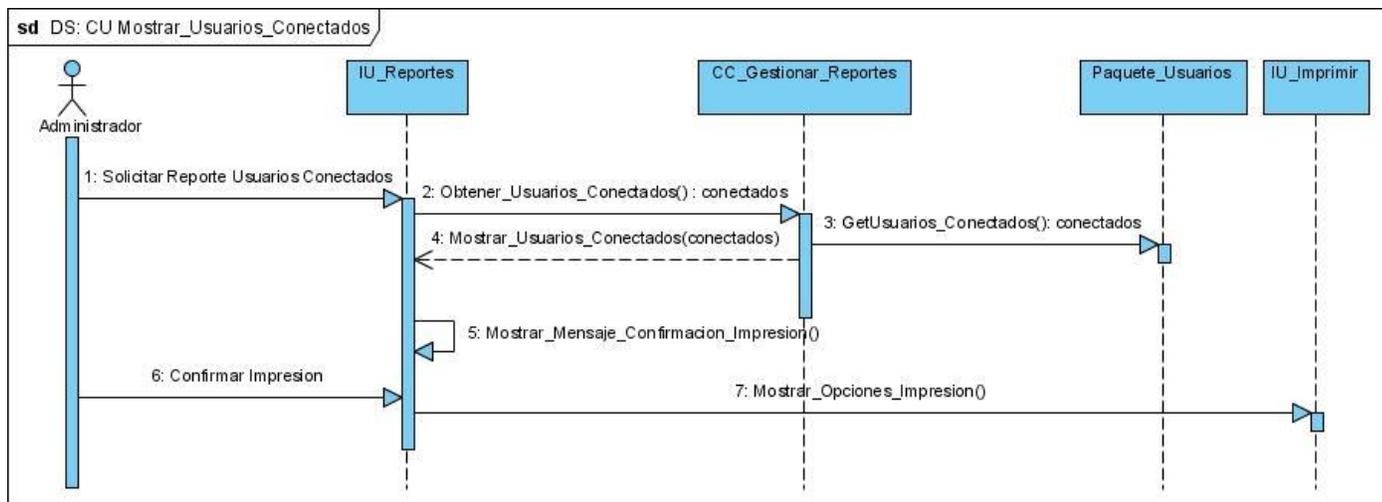
Anexo 65: Diagrama de Secuencia. CU Graficar.



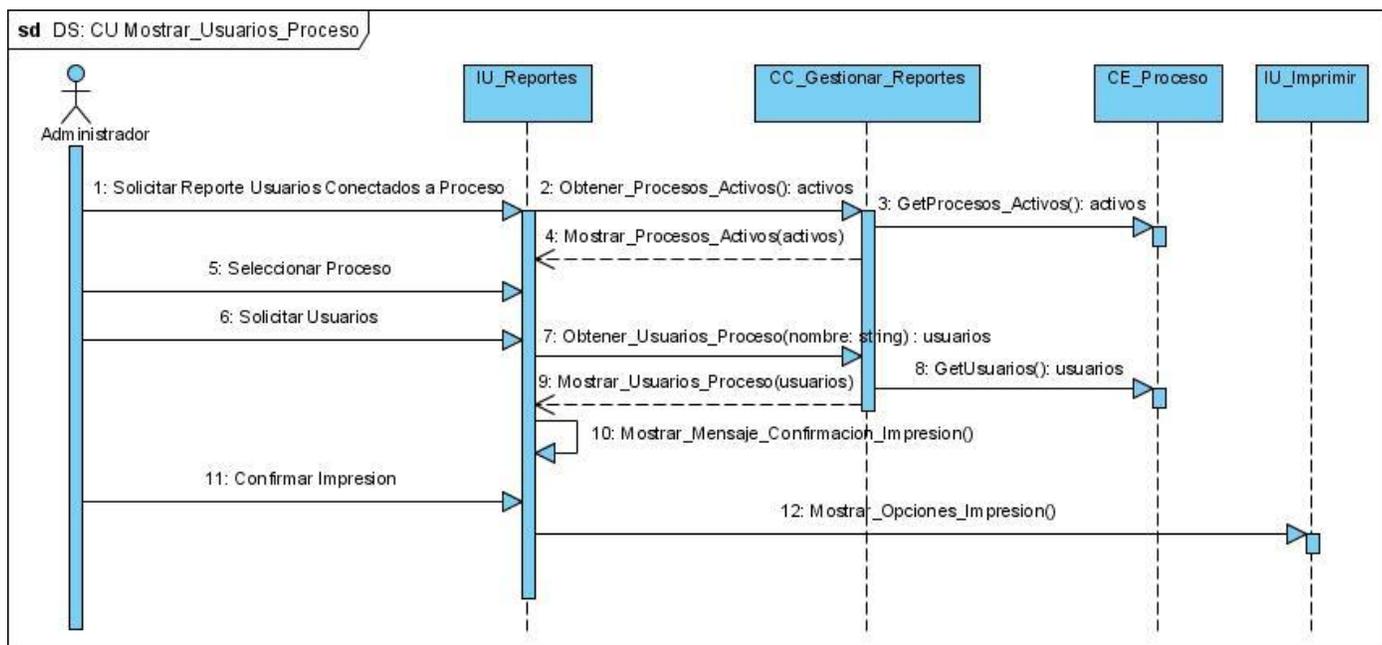
Anexo 66: Diagrama de Secuencia. CU Mostrar\_Procesos\_Activos.



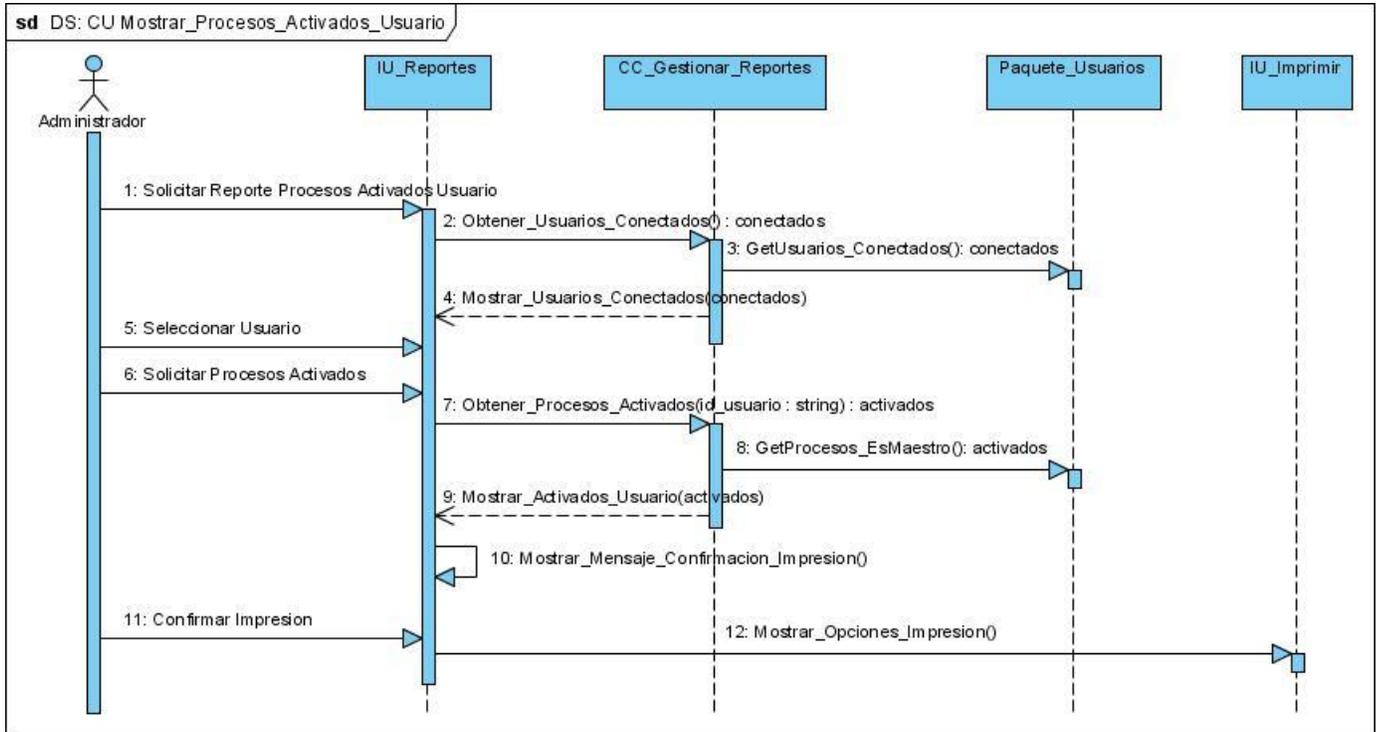
Anexo 67: Diagrama de Secuencia. CU Mostrar\_Usuarios\_Conectados.



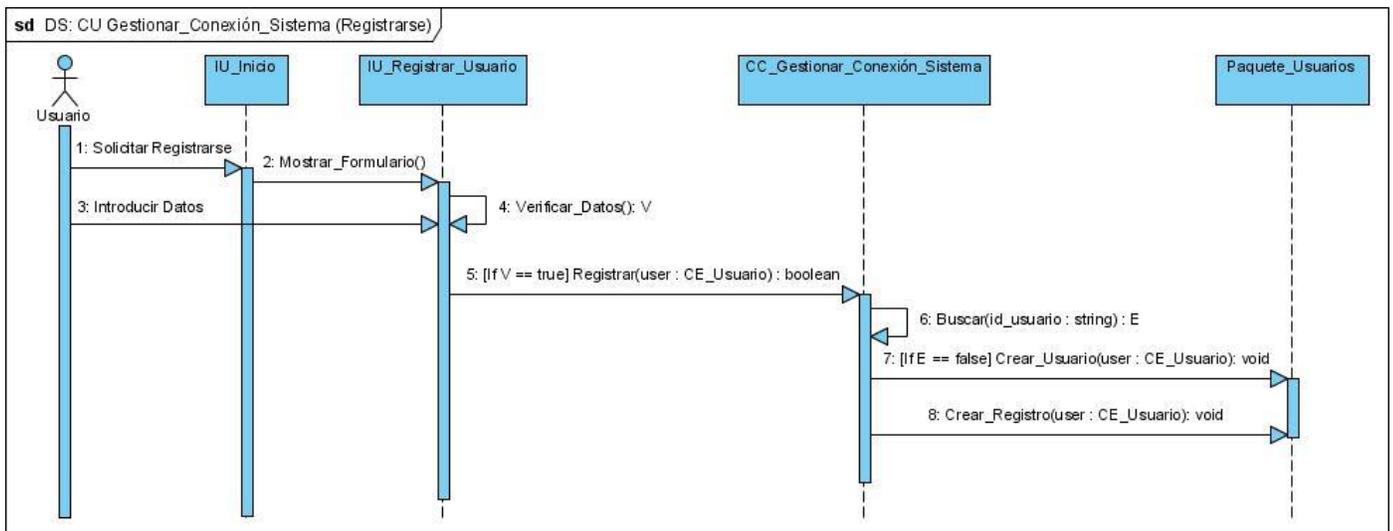
Anexo 68: Diagrama de Secuencia. CU Mostrar\_Usuarios\_Proceso.



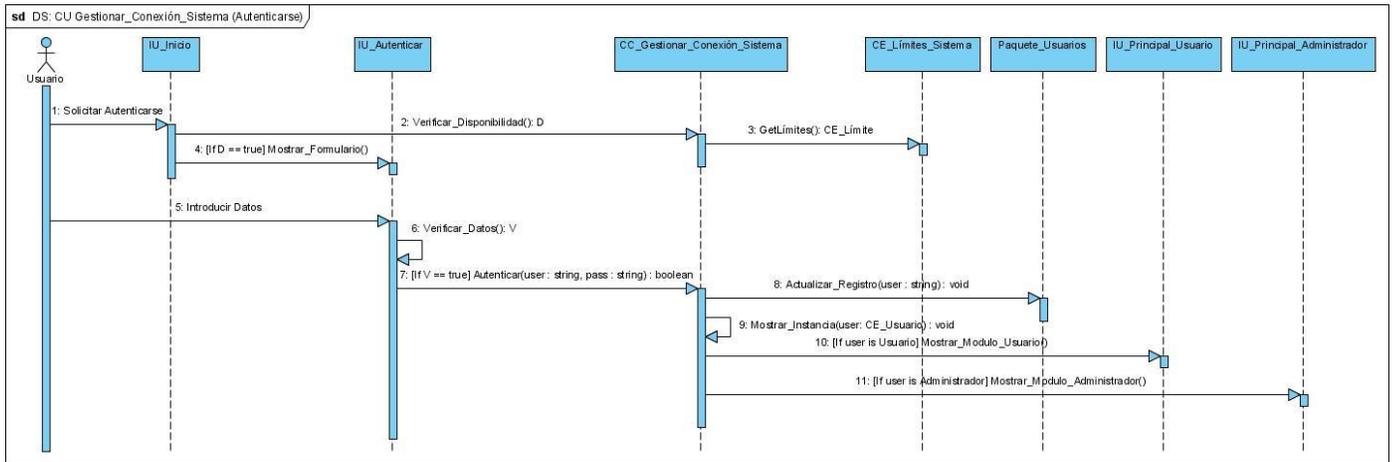
Anexo 69: Diagrama de Secuencia. CU Mostrar\_Procesos\_Activados\_Usuario.



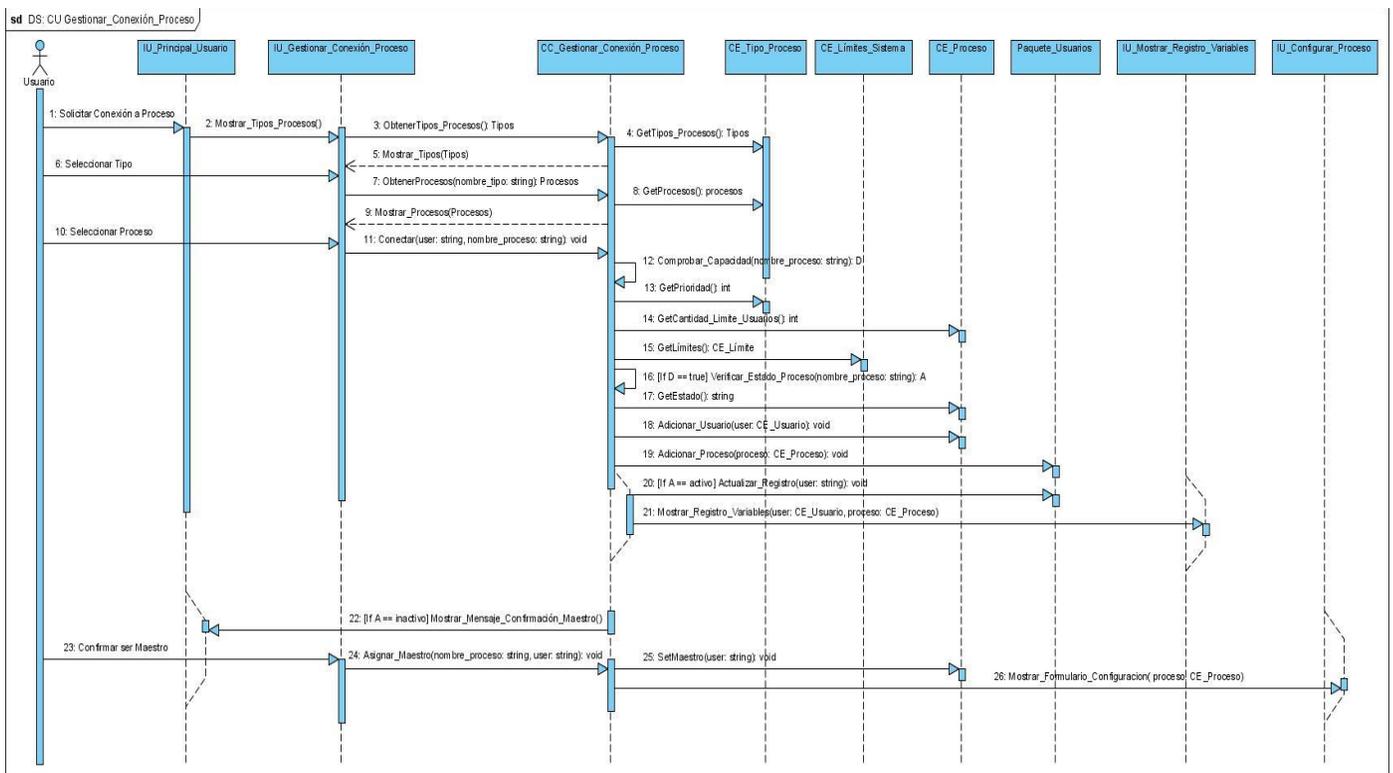
Anexo 70: Diagrama de Secuencia. CU Gestionar\_Conexión\_Sistema.



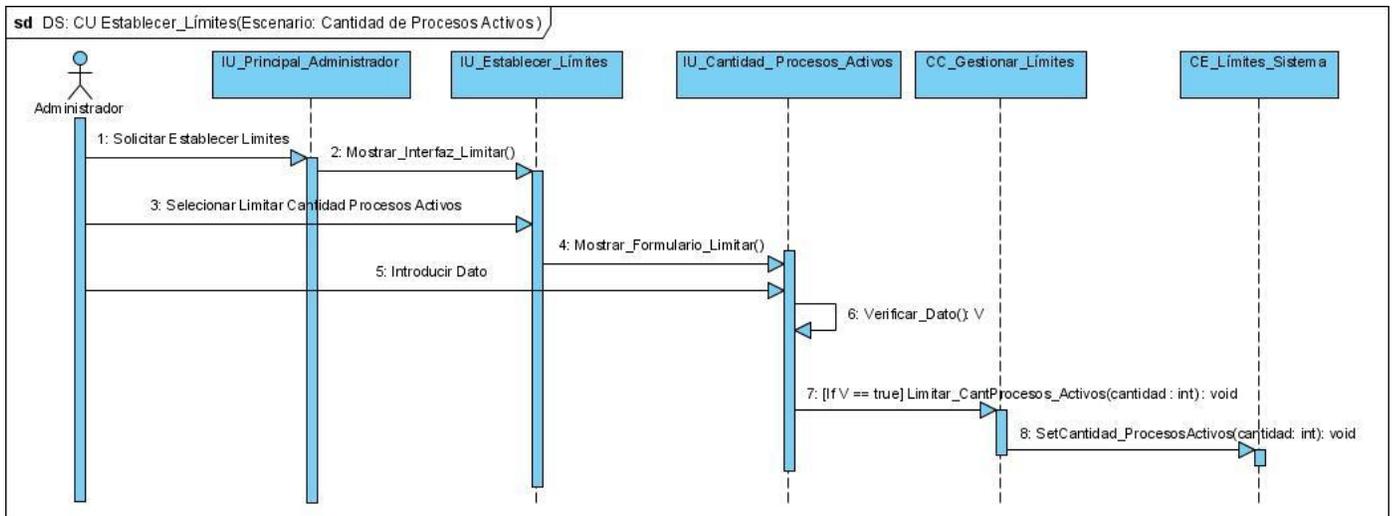
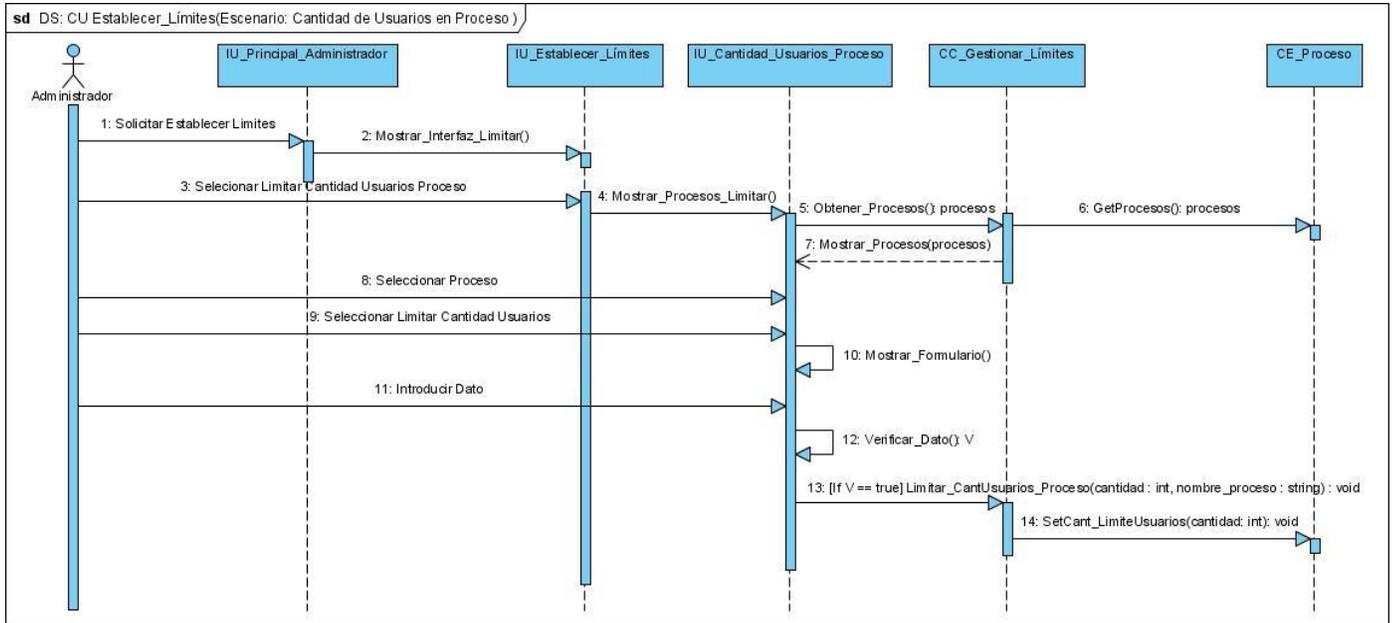
Análisis y Diseño de una herramienta interactiva de simulación de procesos: Nodo Virtual de Procesos

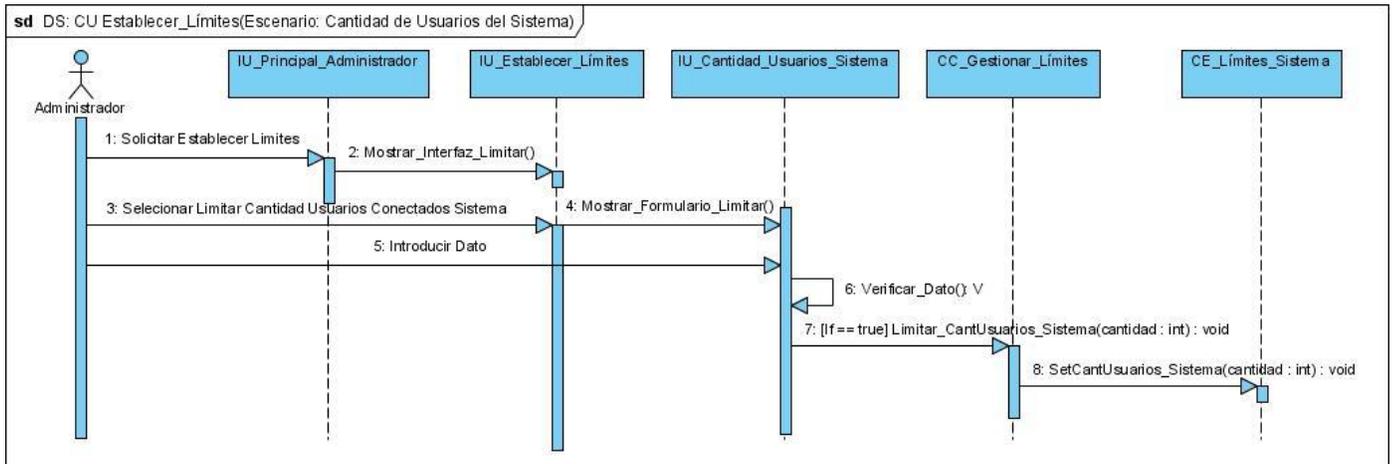


Anexo 71: Diagrama de Secuencia. CU Gestionar\_Conexión\_Proceso.

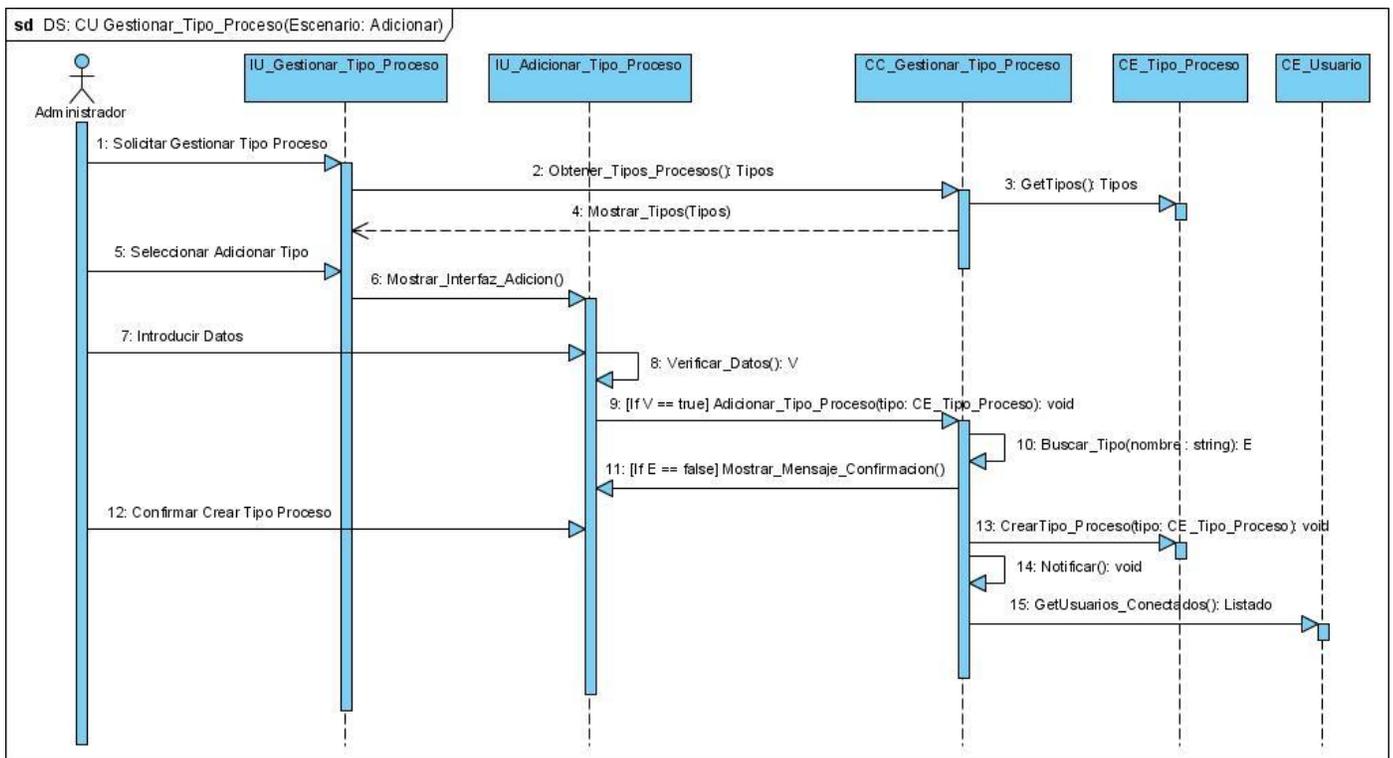


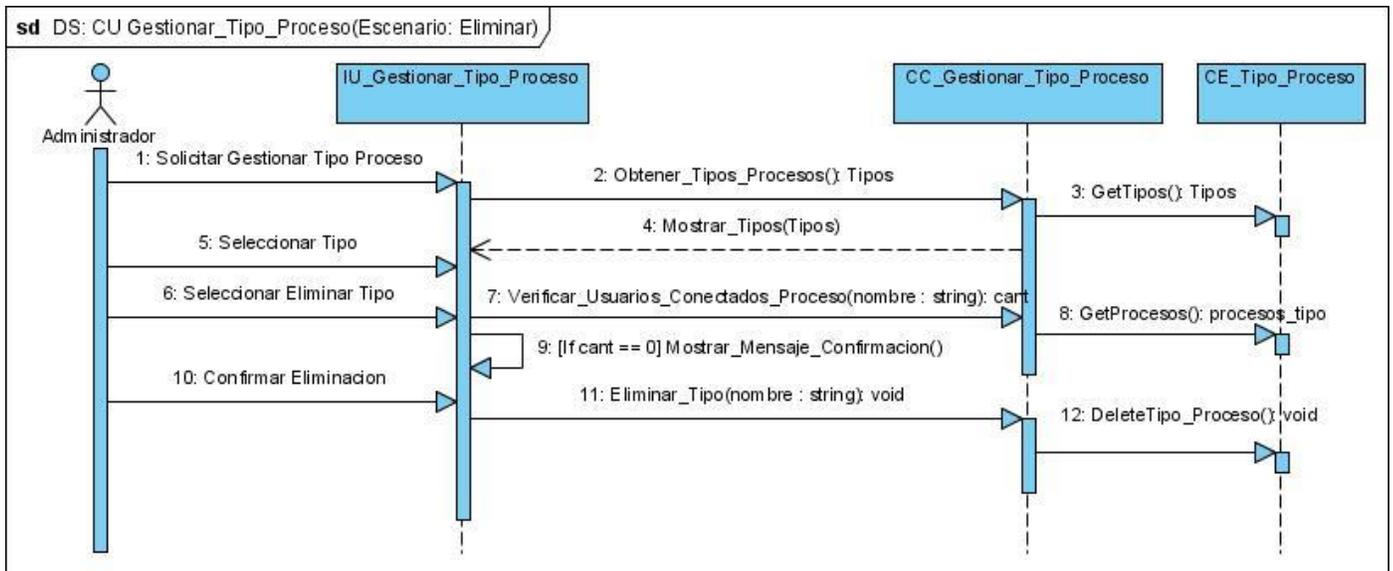
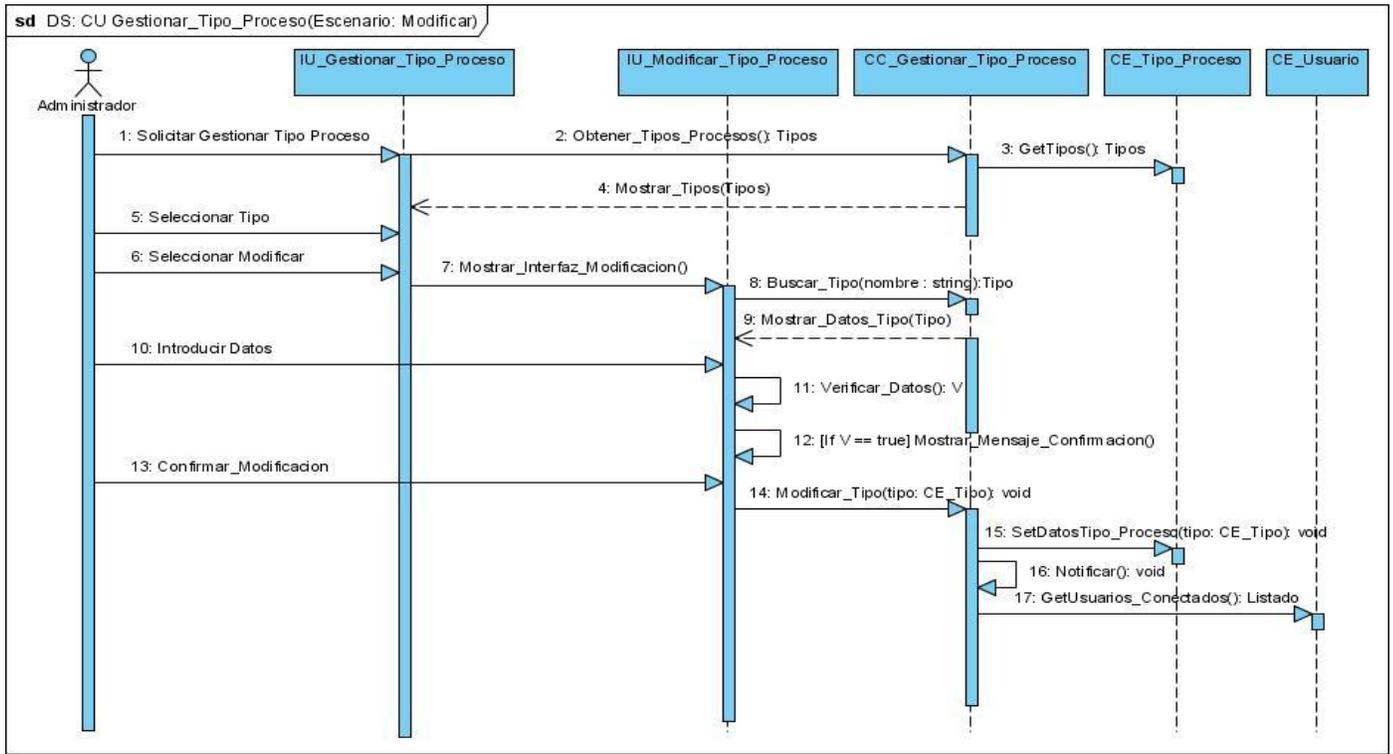
Anexo 72: Diagrama de Secuencia. CU Establecer\_Límites.



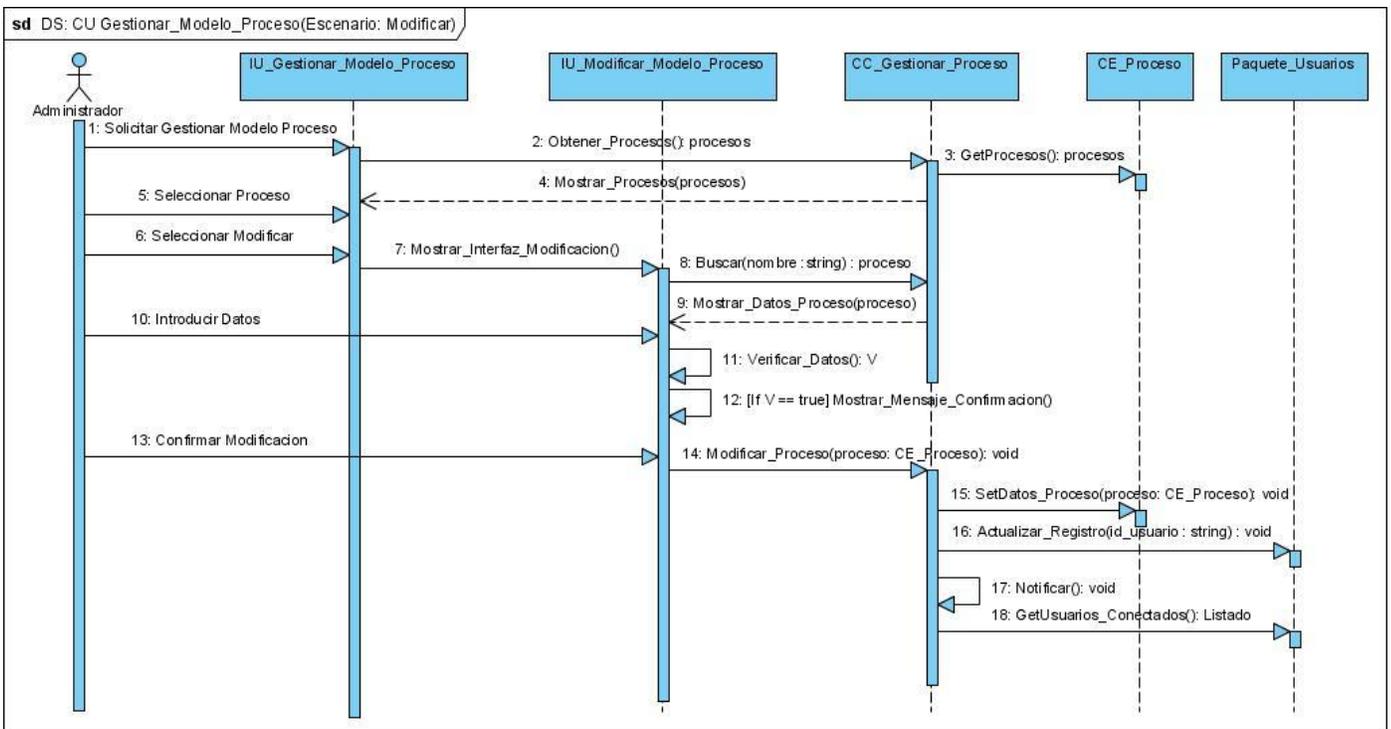
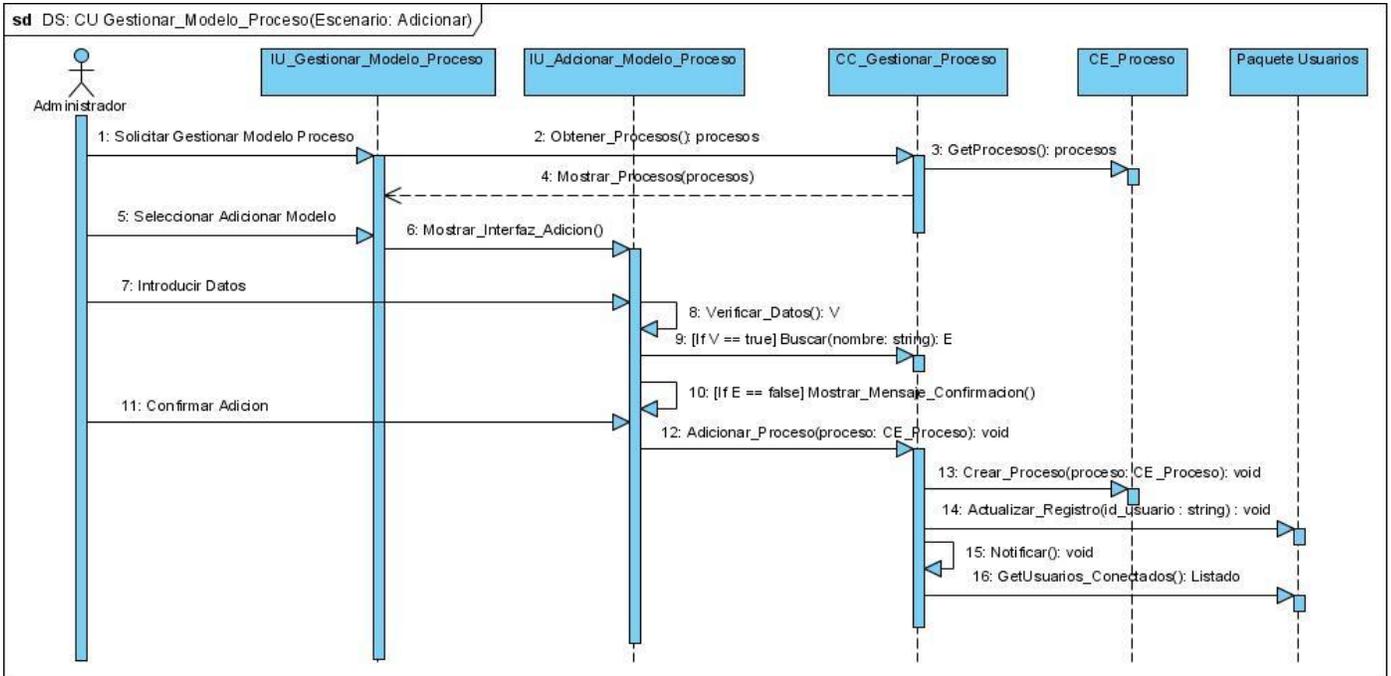


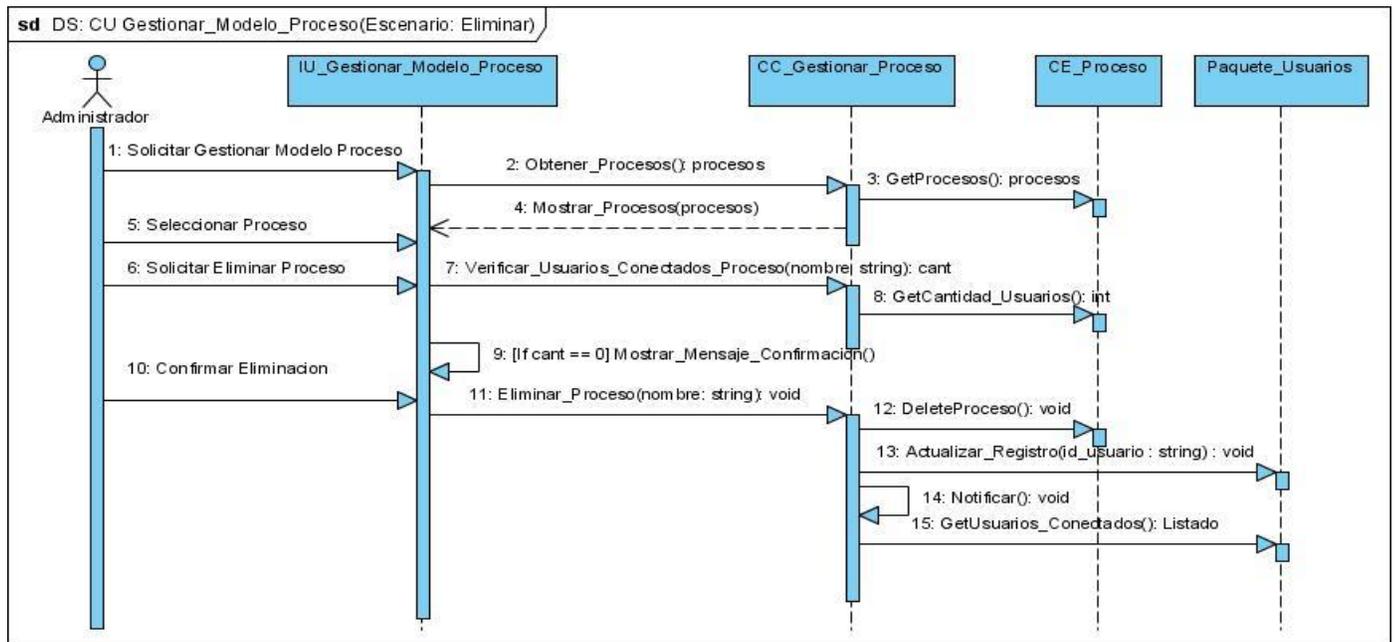
Anexo 73: Diagrama de Secuencia. CU Gestionar\_Tipo\_Proceso.



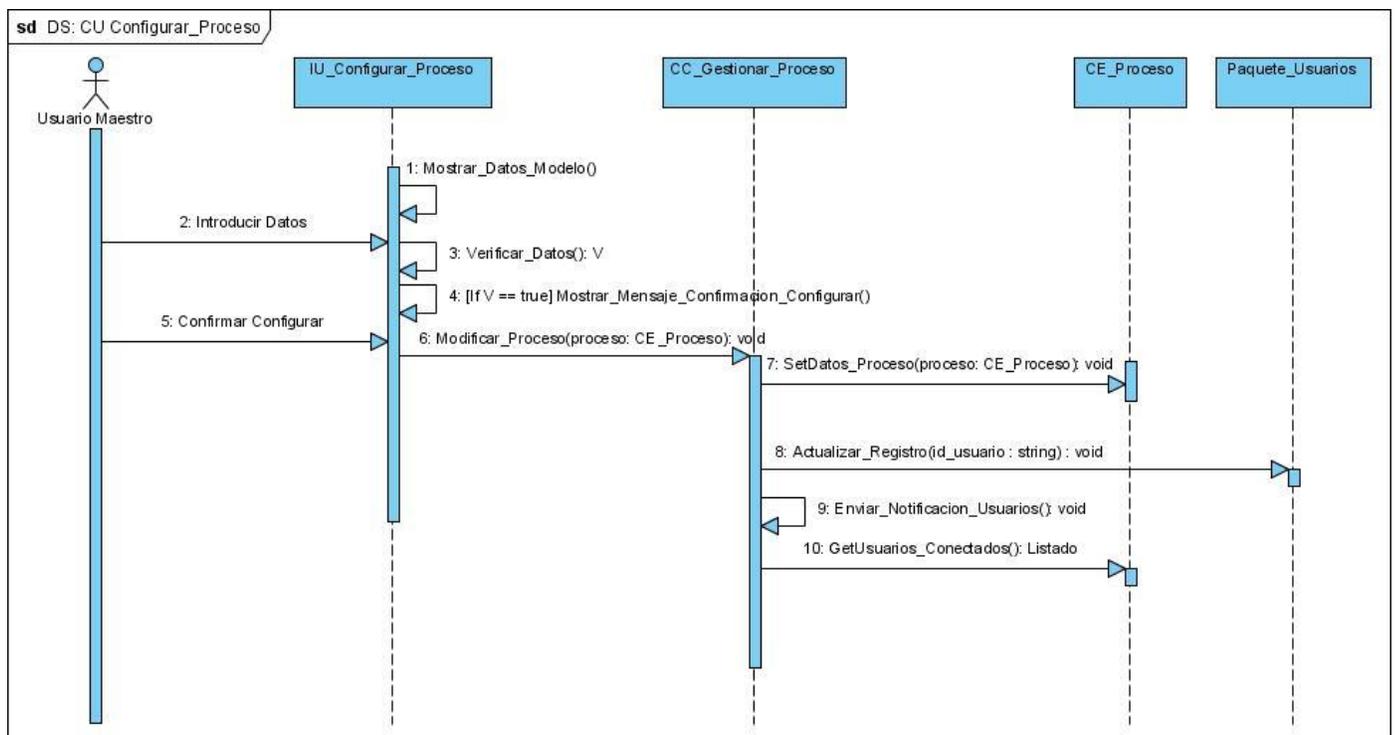


Anexo 74: Diagrama de Secuencia. CU Gestionar\_Modelo\_Proceso.





Anexo 75: Diagrama de Secuencia. CU Configurar\_Proceso.



Anexo 76: Diagrama de Secuencia. CU Gestionar\_Controlador.

