

*Universidad de las Ciencias Informáticas
"Facultad 9"*



Título:

“Análisis y diseño del subsistema de Análisis de Resultados de un Simulador de Procesos Químicos.”

Trabajo de diploma para optar por el título de Ingeniero en Ciencias Informáticas.

Autor: Erllys Carvajal Pérez

Tutor: Ing. Yurísnel Corrales Valdés

Asesora: MSc Yelény Zulueta Véliz

Ciudad de La Habana, 2009

Año 50 del triunfo de la Revolución

DATOS DE CONTACTOS

Tutor:

Ing. Yurisnel Corrales Valdés.

Universidad de las Ciencias Informáticas, Habana, Cuba.

Email: ycvaldes@uci.cu

Asesora:

MSc Yeleny Zulueta Véliz.

Universidad de las Ciencias Informáticas, Habana, Cuba.

Email: yeleny@uci.cu



AGRADECIMIENTOS

A mi país y a la revolución por darme la posibilidad de graduarme como ingeniero.

A mis padres queridos que siempre me han apoyado en todo lo que he necesitado, sobre todo en los momentos difíciles por los que he pasado.

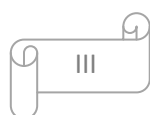
A mi querido hermano por brindarme sus buenos consejos a cada momento.

A mi novia Mercedes por ser mi ángel guardián aquí en la universidad y por ayudarme tanto en el desarrollo de este trabajo.

A mi tutor que también me ha aclarado cuanta duda me ha surgido, gracias por ser mi guía.

En fin a todos mis amigos que de una forma u otra contribuyeron en que yo alcanzara este logro.

A todos ellos muchas gracias, gracias por los momentos que pasamos juntos ya sea estudiando o divirtiéndonos.



DEDICATORIA

Sin duda ninguna a las personas más importantes para mí en la vida.

A mis padres Ada y Luis por hacer de mí el hombre que soy hoy, por brindarme su cariño, su amor y dedicación en toda mi vida, por ser un ejemplo a seguir por mí.

A mi hermano Asniobis que siempre me ha dejado la guía por donde yo debía coger, gracias por ser mi hermano, por brindarme tus consejos y servirme como un ejemplo de bien, por ser el motivo fundamental de mi superación.

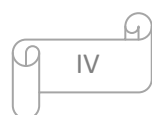
A mi abuelo Cheo por darme la posibilidad de verme convertido en ingeniero.

A Zulma que también me ha dado mucho cariño y apoyo cuando ha estado a mi lado.

A mi novia Mercedes que sin ella todo sería distinto, por su paciencia, cariño, entrega y apoyo.

En fin a toda mi familia.

A todos mis amigos que han sido siempre mi mejor compañía.



RESUMEN

La simulación se ha desarrollado crecientemente a nivel mundial y se han podido observar las mejoras que ofrece esta técnica en diferentes ramas como la educación, la salud, en el campo militar y la industria por solo mencionar algunas. En Cuba se hacen inmensos esfuerzos por estar a la par con los países desarrollados en este campo, pero cabe señalar que la experiencia adquirida en esta técnica aún se está consolidando y que los productos logrados poseen algunas dificultades. La esfera industrial es una de las más afectadas siendo la industria química una de ellas, los simuladores con los que se cuenta son ineficientes en el sentido de los métodos que proporcionan para analizar los resultados obtenidos de las simulaciones completas, siendo esta una herramienta utilizada en todo el mundo para mejorar y perfeccionar el rendimiento de las mismas. En estas condiciones surge la idea de crear un simulador de procesos donde la Universidad de las Ciencias Informáticas (UCI) tiene un importante papel por lo que se hace necesario el análisis y diseño del subsistema Análisis de Resultados del simulador de procesos químicos que se desarrolla actualmente en la facultad 9 de la UCI el cual tiene como objetivo procesar toda la información que arrojan las simulaciones a través de una serie de métodos seleccionados para analizar sus resultados.

Para realizar tal tarea se utiliza la metodología de desarrollo RUP (Proceso Unificado de Desarrollo) por ser una de las más usadas a nivel mundial y también por ser utilizada por la universidad para desarrollar los productos software, además de UML como lenguaje unificado de modelado y la herramienta Rational Rose Enterprise Edition como apoyo para realizar los diagramas y modelos pertinentes a dicho subsistema.

Con el desarrollo del análisis y diseño de este módulo se espera lograr un aporte al desarrollo del simulador propuesto y así contribuir de esta manera al mejoramiento de la producción en la industria química cubana.

PALABRAS CLAVE

Simulación, Análisis de Resultados

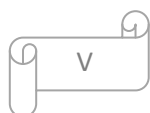


TABLA DE CONTENIDOS.

RESUMEN.....	V
INDICE DE FIGURAS.....	IX
INDICE DE TABLAS.....	X
INTRODUCCIÓN.....	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.....	6
1. Introducción.....	6
1.1 Simulación.....	6
1.1.1 Conceptos Asociados.....	7
1.1.2 Tipos de simulación.....	7
1.1.3 Clasificación de la simulación.....	8
1.1.4 Aplicaciones de la Simulación.....	9
1.1.5 Ventajas y Desventajas.....	10
1.1.6 Etapas de una simulación.....	11
1.2 Simulación de procesos.....	13
1.2.1 Aplicaciones de la simulación de procesos.....	13
1.2.2 Aplicaciones de la simulación de procesos químicos.....	14
1.3 Simuladores de Procesos.....	14
1.3.1 Simuladores internacionales.....	15
1.3.1.1 ASPEN PLUS.....	15
1.3.1.2 HYSYS.....	15
1.3.1.3 CHEMCAD.....	16
1.3.2 Simuladores nacionales.....	17
1.4 Generalidades sobre el análisis de resultados en los simuladores.....	19
1.5 Metodologías de Desarrollo de Software.....	21
1.5.1 Proceso Unificado de Desarrollo (RUP).....	22
1.5.2 Extreme Programming (XP).....	24
1.5.3 Microsoft Solution Framework (MSF).....	26
1.5.4 SCRUM.....	26

1.6 Lenguaje Unificado de Modelado (UML).....	27
1.7 Herramientas CASE	28
1.7.1 Rational Rose	28
1.7.2 Visual Paradigm.....	29
1.8 Metodología y herramientas a utilizar.	30
1.9 Conclusiones.....	30
CAPÍTULO 2: DESCRIPCIÓN DEL SISTEMA.	31
2 Introducción.....	31
2.1 Modelo de dominio	31
2.1.1 ¿Por qué hacer modelo de dominio?	31
2.1.2 Conceptos del dominio.....	32
2.1.3 Diagrama del modelo del dominio.	34
2.1.4 Descripción del Modelo de dominio.....	34
2.2 Especificación de los requisitos de software del sistema.....	35
2.2.1 Requerimientos Funcionales.	35
2.2.2 Requerimientos no funcionales.	36
2.3 Definición y descripción de los casos de uso del sistema.....	38
2.3.1 Definición de los actores	38
2.3.2 Diagrama de casos de uso del sistema.....	39
2.3.3 Descripción textual de los casos de uso del sistema.....	39
2.3.3.1 Descripción textual del caso de uso “Realizar_Análisis_Económico”	40
2.4 Patrones de casos de uso.	42
2.4.1 Patrones de casos de usos utilizados.	42
2.5 Conclusiones.....	44
CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA.....	45
3 Introducción.....	45
3.1 Modelo de Análisis	45
3.1.1 Diagramas de Clases del Análisis.	46
3.1.2 Diagramas de Interacción.	47

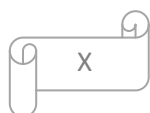
3.2 Modelo de Diseño.....	48
3.2.1 Diagramas de Clases de diseño.....	48
3.2.2 Diagramas de Interacción del diseño	50
3.3 Patrones de Diseño	51
3.3.1 Patrones a utilizar	53
3.4 Conclusiones.....	55
CONCLUSIONES GENERALES.....	56
RECOMENDACIONES	57
BIBLIOGRAFÍA.....	58
REREFENCIAS BIBLIOGRÁFICAS.....	60
ANEXOS.....	63
Anexo 1: Descripciones de Casos de Usos	63
Anexo 2: Diagramas de Clases del Análisis.	75
Anexo 3: Diagramas de Interacción del Análisis.....	78
Anexo 4: Diagramas de Clases del Diseño.....	82
Anexo 5: Diagramas de Interacción del Diseño.	85
GLOSARIO DE TÉRMINOS	90

INDICE DE FIGURAS

Figura 1: Clasificación de la simulación9
Figura 2: Metodología Extreme Programming24
Figura 3: Metodología MSF.....26
Figura 4: Diagrama del modelo de dominio.....34
Figura 5: Diagrama de casos de uso del sistema.....39
Figura 6: Concordancia (Modelo de Reuso).....43
Figura 7: Concordancia (Modelo de Adición)43
Figura 8: Extensión Concreta.....44
Figura 9: Inclusión Concreta44
Figura 10: DCA_CU_Realizar_Análisis_Económico.....46
Figura 11: DC_CU_Realizar_Análisis_Económico47
Figura 12: DCD_CU_Realizar Análisis Económico49
Figura 13: DS_CU_Realizar Análisis Económico50

INDICE DE TABLAS

Tabla 1: Definición de los actores	38
Tabla 2: Descripción del CU Realizar Análisis Económico	42
Tabla 3: Descripción del CU Analizar_Dependencia	65
Tabla 4: Descripción del CU Analizar_Indicadores.....	67
Tabla 5: Descripción del CU Generar_Reporte_Resultados	68
Tabla 6: Descripción del CU Especificar_Variables.....	69
Tabla 7: Descripción del CU Definir_Ingresos_Egresos	70
Tabla 8: Descripción del CU Mostrar_Gráfico_Resultados.....	71
Tabla 9: Descripción del CU Especificar_Ingresos_Egresos	72
Tabla 10: Descripción del CU Analizar_Sensibilidad.....	74



INTRODUCCIÓN

Hoy en día, las constantes transformaciones en el mundo tecnológico y la creciente evolución de nuevas y poderosas tecnologías y herramientas se hace cada día más fuerte debido a la demanda por parte de los usuarios y otras por la competitividad entre las empresas sin mencionar el desarrollo alcanzado de Internet que de una forma u otra también impacta y determina en el avance de estas.

El desarrollo de la informática y las TICs conjuntamente con la creación de computadoras que han evolucionado potencialmente incorporando mejoras en su funcionamiento ha permitido la introducción de nuevas tecnologías de visualización y modelación como lo es la Realidad Virtual, posibilitando la comprensión de los procesos mediante la simulación. Esta se ha convertido en una poderosa técnica que permite imitar los objetos reales en un mundo virtual, artificial o de fantasía cuando por problemas de tiempo, recursos o seguridad no es posible realizar la tarea en su ambiente original.

La disciplina de la simulación de procesos químicos (con la ayuda de computadores) tuvo sus primeras manifestaciones en el ámbito universitario y poco a poco fue penetrando en la industria química. Esta simulación tuvo sus orígenes en el auge mismo de la investigación operacional, por ser una novedosa herramienta para realizar confiable y velozmente diversos cálculos numéricos. La simulación de procesos químicos adquiere importancia por su potencialidad de aplicación en los siguientes campos: (BOGOYA, 1990)

- La docencia y la investigación académica, como ayuda didáctica mediante la cual se explica un fenómeno de difícil, inconveniente o costoso acceso físicamente.
- La investigación, evaluación y análisis de sensibilidad en redes de procesos existentes, para refrendar las condiciones de operación o sugerir otras, de acuerdo con algún nuevo criterio de optimización.
- La predicción de resultados en redes de procesos imaginados, con miras al diseño de procesos químicos.
- La evaluación compartida, conjuntamente con especificaciones de diseño de proceso y de diseño de equipo, para llegar al diseño de plantas químicas.

La simulación de procesos químicos ha tenido dos grandes subdisciplinas: la dinámica (para estados transitorios) y la de estado estable (para procesos estacionarios).

El uso de modelos en Ingeniería Química está consolidado, pero la utilización de "modelos dinámicos" en vez de los "modelos en estado estacionario" es más reciente. Esto se refleja en la existencia en el mercado actualmente de potentes programas destinados a la simulación dinámica de procesos químicos, tales como ASPEN PLUS, PROSIM PLUS, HYSYS, DESIGN II, solo por mencionar algunos (LINNINGER *et al.*, 2000). Estos programas de simulación poseen herramientas matemáticas sofisticadas, las cuales pueden solucionar tanto la condición inicial de estado estacionario así como los siguientes cambios dinámicos. En la actualidad los programas de simulación de procesos se han incorporado de forma generalizada en la industria química, llegando a convertirse en herramientas indispensables para el diseño, modificación y evaluación de equipos y plantas. Esto se debe principalmente a la gran versatilidad que ofrecen en cuanto a la manipulación de las muchas variables involucradas en los procesos y así como la evaluación económica, lo que proporcionan información importante que permite realizar un análisis completo de diversas alternativas para la mejora de un proceso, sin necesidad de materializar los cambios propuestos y obtener conclusiones confiables acerca de la conveniencia o no de implementar la alternativa en cuestión (WEBB y RAMIREZ, 2003).

En el mundo la simulación se ha utilizado en diferentes ramas de la vida humana proporcionando sustanciales beneficios a los mismos como por ejemplo, en el campo militar es asombroso el entrenamiento que pueden alcanzar los soldados y pilotos a través de estas herramientas (simuladores de tiro así como de conducción de tanques y de vuelo), en la salud es igual o mejor en la preparación de los profesionales en distintas situaciones para que cuando se enfrenten a la realidad estén preparados para ella, también es relevante en el proceso de enseñanza- aprendizaje puesto que está demostrado que se alcanzan mejores resultados aplicando simuladores al ser una herramienta interactiva que motiva a los estudiantes.

En los países desarrollados son utilizadas estas herramientas en todas las esferas y en especial en la industria pues hacen posible el aumento de producción y el mejoramiento de la calidad de los productos, además se determinan posibles fallos existentes, esto influye en el ahorro de recursos para el país.

Cuba no está ajena a esta situación por lo que se encuentra enfrascada en la obtención de tales medios que son indispensables en diferentes esferas para mejorar su desarrollo y aún cuando existen en algunas áreas se considera que no son suficientes con los que se cuenta actualmente, unas veces por no tener los recursos necesarios para obtenerlos en el mercado y otras debido al bloqueo que ha impuesto

por muchos años el gobierno de los Estados Unidos a Cuba que dificulta en gran medida poder contar con estas potentes herramientas que en la actualidad se presentan como una forma potencial de desarrollo. Una de las industrias cubanas que presenta problemas con la informatización y por ende del uso de la simulación lo es la industria química.

En la industria química moderna la búsqueda de una mayor competitividad hace indispensable el constante perfeccionamiento de los procesos, en especial los aspectos energéticos; componente determinante en los costos de operación. Una de las vías más exitosas para lograr lo anterior es el desarrollo de software para la simulación, análisis, síntesis y control de procesos apoyado en técnicas de inteligencia artificial, de integración de procesos y minería de datos. En Cuba han sido aplicadas simulaciones a los procesos de las producciones de azúcar crudo y refino, alcohol, ácido sulfúrico, sosa cáustica, refinación de petróleo, níquel, cobalto y otras. Lo logrado está limitado fuertemente porque los software cubanos carecen de varias de las facilidades de sus similares extranjeros, sobre todo de las necesarias para el análisis y la síntesis de los resultados de las simulaciones de procesos completos. (PÉREZ *et al.*, 1989)

Los simuladores cubanos están limitados en este sentido al carecer de una biblioteca de métodos de Análisis de Resultados que posibiliten el estudio y la síntesis de los datos obtenidos de las simulaciones. Estos son de vital importancia para transformar todos esos datos en información que pueda ser utilizada en el futuro por los usuarios.

Es por ello que el Análisis de Resultados es de gran importancia para interpretar correctamente los datos que se obtienen en las simulaciones que se realizan pues estos son necesarios en momentos críticos como la toma de decisiones. Además se puede obtener información para complementar la investigación, planeación y diseño de un sistema o proceso y así evitar errores que puedan ser costosos en el futuro. De esta manera se disminuye la incertidumbre del comportamiento del sistema y se tiene la oportunidad de tomar decisiones acertadas que seguramente aumentarán la productividad de los procesos. Por motivos como estos se hacen necesario que dichos software cuenten con diferentes métodos de análisis para poder obtener los resultados por diferentes vías de solución y así contar con mayores argumentos al ser analizados los datos que arroja cada simulación y a través de dichos métodos lograr que estos datos obtenidos se conviertan en información clara para todos y que puedan ayudar en el futuro.

Uno de los proyectos que se ha desarrollado al calor de la batalla de ideas lo constituye la Universidad de las Ciencias Informáticas (UCI), la cual en el futuro espera brindar grandes aportes a esta rama, ya se están dando los primeros pasos para el desarrollo de la misma. Han surgido algunos proyectos de simulación como lo es el simulador de procesos químicos que se desarrolla en la facultad 9 de la UCI aunque anteriormente ya se han desarrollado algunos e incluso se trabaja en otros. Este simulador tiene la misma limitante anteriormente dicha.

De aquí la necesidad de realizar el análisis y diseño al subsistema Análisis de Resultados al ser este de vital importancia para el software que se desea implementar. De esta manera el **problema a resolver** es la necesidad de una biblioteca de métodos de Análisis de Resultados de la simulación en el Simulador de Procesos Químicos que se desarrolla en la Facultad 9 de la UCI.

El **objetivo general** del trabajo es Modelar el subsistema de Análisis de Resultados del Simulador de Procesos Químicos que se desarrolla en la Facultad 9 de la UCI.

Para el cumplimiento del mismo se ha propuesto realizar las siguientes **tareas de investigación**:

1. Caracterización de algunas de las aplicaciones existentes para el Análisis de Resultados de la simulación.
2. Caracterización de la metodología y herramientas a utilizar en el desarrollo del subsistema de Análisis de Resultados.
3. Análisis del subsistema de Análisis de Resultados.
4. Diseño del subsistema de Análisis de Resultados.
5. Implementación del prototipo funcional del subsistema Análisis de Resultados que incluya:
 - a. Análisis Económico.

Se tiene el Proceso de desarrollo de simuladores de procesos para la industria química como **objeto de estudio** el cual estaría enmarcado en el **campo de acción** del Análisis de Resultados en los Simuladores de Procesos Químicos.

Con un correcto análisis y diseño del subsistema de Análisis de Resultados del Simulador de Procesos Químicos que se desarrolla en la Facultad 9 de la UCI permitirá un mejor entendimiento por parte del grupo de desarrollo en el momento de la implementación es la **idea a defender** de este trabajo.

Con el desarrollo de lo planteado anteriormente se espera lograr:

- Modelo de análisis del subsistema.
- Modelo de diseño del subsistema
- Prototipo de interfaz funcional del módulo.

Con el fin de resolver y dar cumplimiento a los objetivos y las tareas propuestas se han utilizado algunos métodos científicos categorizados como teóricos entre los que se encuentran:

Histórico-Lógico:

Este método se utiliza en la investigación para estudiar la teoría conocida hasta el momento, todo lo que se ha hecho acerca del tema que se trata en el trabajo.

Analítico-Sintético:

Este método es el que permite buscar la esencia del problema que se trata, los rasgos que lo caracterizan y los distinguen. Además permite la extracción de los elementos más importantes que se relacionan con el objeto de estudio.

Modelación:

Este método se utiliza a la hora de realizar los diferentes diagramas y modelos que ayudarán a un mejor entendimiento de las funcionalidades que debe cumplir el sistema y permitirá estudiar las relaciones entre las mismas.

Además de los métodos anteriores se utiliza otro que se encuentra dentro de la clasificación de los empíricos el cual es:

Entrevista:

Se ha utilizado con el objetivo de facilitar el levantamiento de requisitos que debe tener el producto así como la obtención de información para su desarrollo.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.

1. Introducción

La simulación de procesos es una de las más grandes herramientas de la ingeniería industrial, la cual se utiliza para representar un proceso mediante otro que lo hace mucho más simple y entendible. 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 su desarrollo.

El principal objetivo de este capítulo es ubicar al lector en el tema del trabajo, abordándose en él conceptos asociados con el contenido del mismo tales como simulación y algunas características generales de esta así como información sobre la simulación de procesos y el Análisis de Resultados específicamente. También se hace referencia a los principales programas de simulación existentes tanto internacional como nacionalmente brindando sus características generales. Además se dan a conocer algunas metodologías y herramientas para el desarrollo de software.

1.1 Simulación

La simulación resume toda la teoría relacionada con un proceso en el cual se sustituyen las situaciones reales por otras creadas artificialmente, de las cuales se deben aprender ciertas acciones, habilidades y hábitos.

A este novedoso término se le han dado varias definiciones, una de ellas es que “la simulación consiste simplemente en poner a un individuo en un ambiente que omita algún aspecto de la realidad y en idear dentro de ese marco un problema que exija la participación activa del alumno para iniciar y llevar a cabo una serie de indagaciones, decisiones y actos.” (ARGOTA, 1990)

En otra fuente se habla de que “la simulación consiste en experimentar con los modelos generados mediante la programación matemática, asignando valores a las variables de entrada y observando los valores de las de salida; es decir, es el laboratorio o rama experimental de la investigación operacional, donde la práctica y los resultados distan de la comprensión surgida de la física, la química o la biología.” (BOGOYA, 1990)

En un dialecto más formal y enfocado a la simulación que se desea abordar se puede encontrar que "simulación es una técnica numérica para conducir experimentos en una computadora digital. Estos experimentos comprenden ciertos tipos de relaciones matemáticas y lógicas, las cuales son necesarias para describir el comportamiento y la estructura de sistemas complejos del mundo real a través de largos periodos de tiempo". (NAYLOR, 1971)

Abordando este mismo aspecto también se plantea que "la simulación es el proceso de diseñar un modelo de un sistema real y llevar a término experiencias con el mismo con la finalidad de comprender el comportamiento del sistema o evaluar nuevas estrategias, dentro de los límites impuestos por un cierto criterio o un conjunto de ellos, para el funcionamiento del sistema". (SHANNON, 1988)

Para el autor la simulación es una herramienta que permite representar un sistema real determinado de forma artificial con el objetivo de predecir cómo funcionará este bajo determinadas condiciones a través de modelos matemáticos.

1.1.1 Conceptos Asociados

Modelo:

Es una representación de un objeto, sistema, o idea. Usualmente, su propósito es ayudar explicar, entender o mejorar un sistema.(SHANNON, 1988)

Sistema:

Conjunto de objetos o ideas que están interrelacionados entre sí como una unidad para la consecución de un fin.(SHANNON, 1988)

Proceso:

Es el conjunto de actividades que se llevan a cabo dentro de un sistema.(TARIFA, 2002)

Simulador:

Algún algoritmo tal que una vez ingresados ciertos valores de entradas, entregue como resultado, el comportamiento que tendría el sistema real.(HUGO, 2001)

1.1.2 Tipos de simulación

De acuerdo a la naturaleza del modelo empleado, la simulación puede ser por (FISHMAN, 1978):

- ❖ **Identidad:** Es cuando el modelo es una réplica exacta del sistema en estudio. Es la que utilizan las empresas automotrices cuando realizan ensayos de choques de automóviles utilizando unidades reales.
- ❖ **Cuasi-identidad:** Se utiliza una versión ligeramente simplificada del sistema real. Por ejemplo, los entrenamientos militares que incluyen movilización de equipos y tropas pero no se lleva a cabo una batalla real.
- ❖ **Laboratorio:** Se utilizan modelos bajo las condiciones controladas de un laboratorio. Se pueden distinguir dos tipos de simulaciones:
 - **Juego operacional:** Personas compiten entre ellas, estas forman parte del modelo, la otra parte consiste en computadoras, maquinaria y otros. Es el caso de una simulación de negocios donde las computadoras se limitan a recolectar la información generada por cada participante y a presentarla en forma ordenada a cada uno de ellos.
 - **Hombre-Máquina:** Se estudia la relación entre las personas y la máquina. Las personas también forman parte del modelo. La computadora no se limita a recolectar información, sino que también la genera. Un ejemplo de este tipo de simulación es el simulador de vuelo.
- ❖ **Simulación por computadora:** El modelo es completamente simbólico y está implementado en un lenguaje computacional. Las personas quedan excluidas del modelo. Un ejemplo es el simulador de un sistema de redes de comunicación donde la conducta de los usuarios está modelada en forma estadística. Este tipo de simulación a su vez puede ser:
 - **Digital:** Cuando se utiliza una computadora digital.
 - **Analógica:** Cuando se utiliza una computadora analógica. En este grupo también se pueden incluir las simulaciones que utilizan modelos físicos.

1.1.3 Clasificación de la simulación.

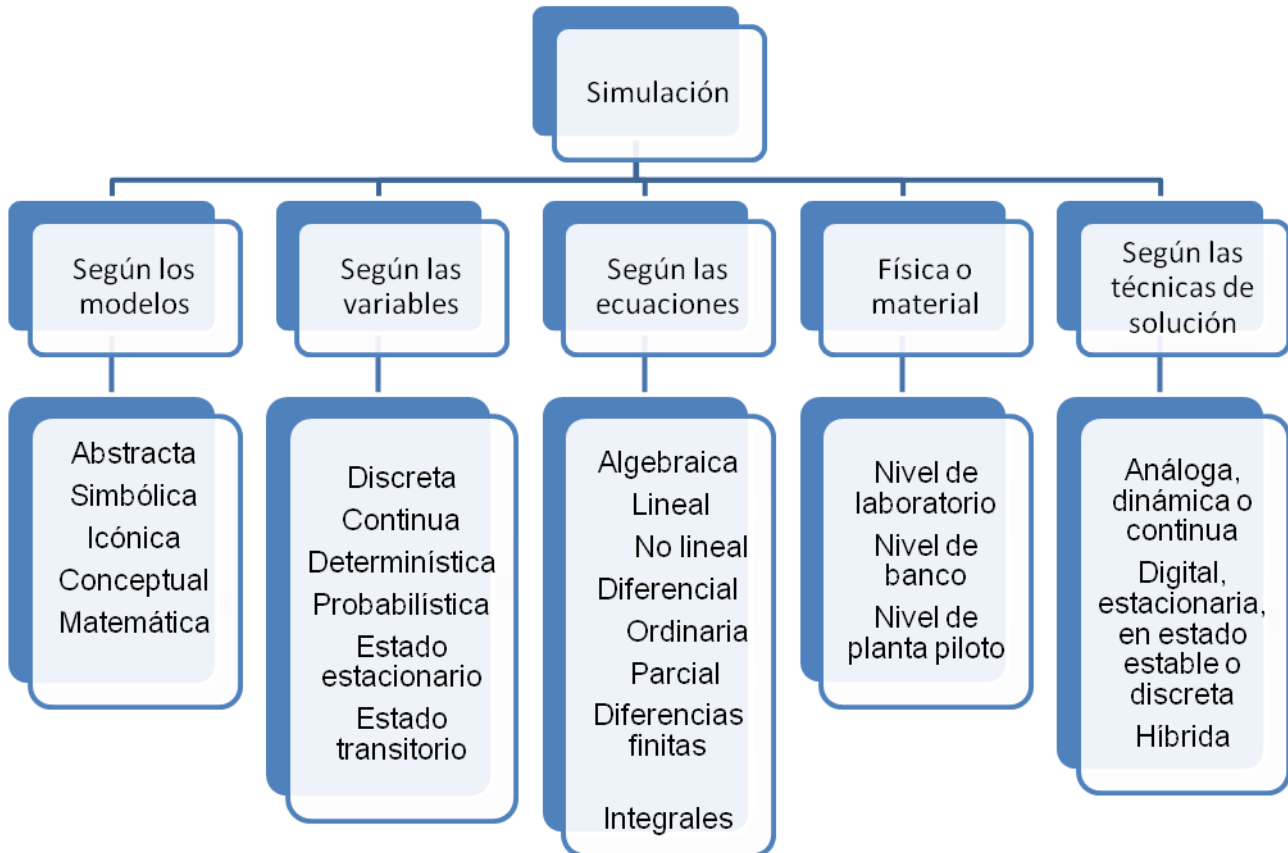


Figura 1: Clasificación de la simulación.(BOGOYA, 1990)

1.1.4 Aplicaciones de la Simulación.

Actualmente la simulación presta un invaluable servicio en casi todas las áreas posibles, algunas de ellas son:

- ❖ **Procesos de manufacturas:** Ayuda a detectar cuellos de botellas, a distribuir personal, determinar la política de producción.
- ❖ **Plantas industriales:** Brinda información para establecer las condiciones óptimas de operación, y para la elaboración de procedimientos de operación y de emergencias.
- ❖ **Sistemas públicos:** Predice la demanda de energía durante las diferentes épocas del año, anticipa el comportamiento del clima, predice la forma de propagación de enfermedades.
- ❖ **Sistemas de transportes:** Detecta zonas de posible congestionamiento, zonas con mayor riesgo de accidentes, predice la demanda para cada hora del día.
- ❖ **Construcción:** Predice el efecto de los vientos y temblores sobre la estabilidad de los edificios, provee información sobre las condiciones de iluminación y condiciones ambientales en el interior de los mismos, detecta las partes de las estructuras que deben ser reforzadas.
- ❖ **Diseño:** Permite la selección adecuada de materiales y formas. Posibilita estudiar la sensibilidad del diseño con respecto a parámetros no controlables.
- ❖ **Educación:** Es una excelente herramienta para ayudar a comprender un sistema real debido a que puede expandir, comprimir o detener el tiempo, y además es capaz de brindar información sobre variables que no pueden ser medidas en el sistema real.
- ❖ **Capacitación:** Dado que el riesgo y los costos son casi nulos, una persona puede utilizar el simulador para aprender por sí misma utilizando el método más natural para aprender: el de prueba y error.(TARIFA, 2002)

1.1.5 Ventajas y Desventajas

La simulación es conveniente cuando:

- ❖ No existe una formulación matemática analíticamente resoluble. Muchos sistemas reales no pueden ser modelados matemáticamente con las herramientas actualmente disponibles, por ejemplo la conducta de un cliente de un banco.
- ❖ Existe una formulación matemática, pero es difícil obtener una solución analítica. Los modelos matemáticos utilizados para modelar un reactor nuclear o una planta química son imposibles de resolver en forma analítica sin realizar serias simplificaciones.

- ❖ No existe el sistema real. Es problema del ingeniero que tiene que diseñar un sistema nuevo. El diseño del sistema mejorará notablemente si se cuenta con un modelo adecuado para realizar experimentos.
- ❖ Los experimentos son imposibles debido a impedimentos económicos, de seguridad, de calidad o éticos. En este caso el sistema real está disponible para realizar experimentos, pero la dificultad de los mismos hace que se descarte esta opción. Un ejemplo de esto es la imposibilidad de provocar fallas en un avión real para evaluar la conducta del piloto, tampoco se puede variar el valor de un impuesto para evaluar la reacción del mercado.
- ❖ El sistema evoluciona muy lentamente o muy rápidamente. Un ejemplo de dinámica lenta es el problema de los científicos que estudian la evolución del clima. Ellos deben predecir la conducta futura del clima dado las condiciones actuales, no pueden esperar a que un tornado arrase una ciudad para luego dar el mensaje de alerta. Por el contrario, existen fenómenos muy rápidos que deben ser simulados para poder observarlos en detalles, por ejemplo una explosión.

Entre las posibles desventajas de la simulación se pueden citar:

- ❖ El desarrollo de un modelo puede ser costoso, laborioso y lento.
- ❖ Existe la posibilidad de cometer errores. No se debe olvidar que la experimentación se lleva a cabo con un modelo y no con el sistema real; entonces, si el modelo está mal o se cometen errores en su manejo, los resultados también serán incorrectos.
- ❖ No se puede conocer el grado de imprecisión de los resultados. Por lo general el modelo se utiliza para experimentar situaciones nunca planteadas en el sistema real, por lo tanto no existe información previa para estimar el grado de correspondencia entre la respuesta del modelo y la del sistema real. (TARIFA, 2002)

1.1.6 Etapas de una simulación

En el desarrollo de una simulación se pueden distinguir las siguientes etapas:

- ❖ **Formulación del problema:** En este paso debe quedar perfectamente establecido el objeto de la simulación. El cliente y el desarrollador deben acordar lo más detalladamente posible los siguientes factores: los resultados que se esperan del simulador, el plan de experimentación, el tiempo

disponible, las variables de interés, el tipo de perturbaciones a estudiar, el tratamiento estadístico de los resultados, la complejidad de la interfaz del simulador y otros. Se debe establecer si el simulador será operado por el usuario o si el usuario sólo recibirá los resultados.

- ❖ **Definición del sistema:** El sistema a simular debe estar perfectamente definido. El cliente y el desarrollador deben acordar dónde estará la frontera del sistema a estudiar y las interacciones con el medioambiente que serán consideradas.
- ❖ **Formulación del modelo:** La misma comienza con el desarrollo de un modelo simple que captura los aspectos relevantes del sistema real. Los aspectos relevantes del sistema real dependen de la formulación del problema; para un ingeniero de seguridad los aspectos relevantes de un automóvil son diferentes de los aspectos considerados por un ingeniero mecánico para el mismo sistema. Este modelo simple se irá enriqueciendo como resultado de varias iteraciones.
- ❖ **Colección de datos:** La naturaleza y cantidad de datos necesarios están determinadas por la formulación del problema y del modelo. Los datos pueden ser provistos por registros históricos, experimentos de laboratorios o mediciones realizadas en el sistema real. Los mismos deberán ser procesados adecuadamente para darles el formato exigido por el modelo.
- ❖ **Implementación del modelo en la computadora:** El modelo es implementado utilizando algún lenguaje de computación. Existen lenguajes específicos de simulación que facilitan esta tarea; también, existen programas que ya cuentan con modelos implementados para casos especiales.
- ❖ **Verificación:** En esta etapa se comprueba que no se hayan cometido errores durante la implementación del modelo. Para ello, se utilizan las herramientas de *debugging* provistas por el entorno de programación.
- ❖ **Validación:** En esta etapa se comprueba la exactitud del modelo desarrollado. Esto se lleva a cabo comparando las predicciones del modelo con: mediciones realizadas en el sistema real, datos históricos o datos de sistemas similares. Como resultado de esta etapa puede surgir la necesidad de modificar el modelo o recolectar datos adicionales.
- ❖ **Diseño de experimentos:** En esta etapa se decide las características de los experimentos a realizar: el tiempo de arranque, el tiempo de simulación y el número de simulaciones.
- ❖ **Experimentación:** En esta etapa se realizan las simulaciones de acuerdo el diseño previo. Los resultados obtenidos son debidamente recolectados y procesados.

- ❖ **Interpretación:** Se analiza la sensibilidad del modelo con respecto a los parámetros que tienen asociados la mayor incertidumbre. Si es necesario, se deberán recolectar datos adicionales para refinar la estimación de los parámetros críticos.
- ❖ **Implementación:** Conviene acompañar al cliente en la etapa de implementación para evitar el mal manejo del simulador o el mal empleo de los resultados del mismo.
- ❖ **Documentación:** Incluye la elaboración de la documentación técnica y manuales de uso. La documentación técnica debe contar con una descripción detallada del modelo y de los datos; también, se debe incluir la evolución histórica de las distintas etapas del desarrollo. Esta documentación será de utilidad para el posterior perfeccionamiento del simulador. (BANKS *et al.*, 1996)

1.2 Simulación de procesos.

La simulación de procesos puede ser definida como una técnica para evaluar en forma rápida un proceso con base en una representación del mismo, mediante modelos matemáticos. (MARTÍNEZ *et al.*, 2000)

1.2.1 Aplicaciones de la simulación de procesos.

En principio, la simulación de procesos puede ser útil en todas las etapas del desarrollo de un proyecto industrial. En las diferentes etapas de un proyecto, puede haber necesidad de realizar simulaciones con diferentes niveles de sofisticación. La simulación de procesos puede usarse en las siguientes etapas del desarrollo de un proyecto industrial:

- a) Investigación y desarrollo: Una simulación sencilla se puede usar para probar la factibilidad técnica y económica del proyecto.
- b) Etapla crítica en la toma de decisiones: Se prueban diferentes alternativas de proceso y condiciones de operación y se toman decisiones.
- c) Planta piloto: Simulación con modelos más sofisticados para obtener mejores estimaciones de las condiciones de operación a escala industrial.
- d) Diseño: La simulación proporciona todos los datos de proceso requeridos para el diseño detallado de los diferentes equipos.
- e) Simulación de plantas existentes: Puede ser muy útil cuando es necesario cambiar las condiciones de operación, o cuando se quieren sustituir materias primas. (MARTÍNEZ *et al.*, 2000)

1.2.2 Aplicaciones de la simulación de procesos químicos

La simulación de procesos químicos es una herramienta moderna que se ha hecho indispensable para la solución adecuada de los problemas de proceso. Permite efectuar el análisis de plantas químicas en operación y llevar a cabo las siguientes tareas, las cuales son comunes en las diversas ramas de la industria química:

- ❖ Detección de cuellos de botella en la producción.
- ❖ Predicción de los efectos de cambios en las condiciones de operación y capacidad de la planta.
- ❖ Optimización de las variables de operación.
- ❖ Optimización del proceso cuando cambian las características de los insumos y/o las condiciones económicas del mercado.
- ❖ Análisis de nuevos procesos para nuevos productos.
- ❖ Evaluación de alternativas de proceso para reducir el consumo de energía.
- ❖ Análisis de condiciones críticas de operación.
- ❖ Transformación de un proceso para desarrollar otras materias primas.
- ❖ Análisis de factibilidad y viabilidad de nuevos procesos.
- ❖ Optimización del proceso para minimizar la producción de desechos y contaminantes.
- ❖ Entrenamiento de operadores e ingenieros de proceso.
- ❖ Investigación de la factibilidad de automatización de un proceso.

1.3 Simuladores de Procesos

En los últimos años, la simulación de procesos ha llegado a ser una herramienta adecuada y oportuna de apoyo para el diseño, caracterización, optimización y monitoreo del funcionamiento de procesos industriales y además su uso se está extendiendo en las instituciones de formación de ingenieros químicos

Existe una gran variedad de simuladores de procesos comerciales, algunos de las cuales son poderosas herramientas de cálculo, con inmensos bancos de datos que contienen las propiedades físicas de miles de compuestos y sustancias químicas, selección de modelos termodinámicos, cálculos de equipos (teórico y real), análisis de costo, estado de agregación y condiciones de operación, que le dan al simulador la ventaja de una gran versatilidad.

Algunos de estos simuladores de procesos comerciales son:

Aspen Plus (Aspen Technology, USA), HYSYS (Hyprotech, Canadá), CHEMCAD (Chemstations, USA), Pro II (Simulations Sciences, USA), DESIGN II (Advanced Engineering Software), SuperPro Designer (Intelligent INC), y otros.

1.3.1 Simuladores internacionales.

1.3.1.1 ASPEN PLUS

El Sistema Avanzado para Ingeniería de Procesos - Advanced System for Process Engineering (ASPEN) es un mercado líder en herramientas de modelado de proceso de diseño conceptual, optimización y monitoreo de desempeño para la industria química, polímeros, especialidades químicas, metales y minerales. Desarrollado en la década de 1970 por investigadores del Massachusetts Institute of Technology (MIT) y comercializado desde 1980 por una compañía denominada AspenTech. Aspen Plus es un simulador estacionario, secuencial modular (en las últimas versiones permite la estrategia orientada a ecuaciones). Actualmente es posible que sea el más extendido en la industria. Se ha utilizado para modelar procesos en industrias: química y petroquímica, refinado de petróleo, procesamientos de gas y aceites, generación de energía, metales y minerales, industrias del papel y la pulpa y otros. Aspen Plus tiene la base de datos más amplia entre los simuladores de procesos comerciales, e incluye comportamiento de iones y de electrolitos. Además modela y simula cualquier tipo de proceso para el cual hay un flujo continuo de materiales y energía de una unidad de proceso a otra. Posee herramientas para cálculos de costes y optimizaciones del proceso, generación de resultados en forma gráfica y en tablas y otros. (RODRÍGUEZ, 2000)

1.3.1.2 HYSYS

HYSYS es un programa interactivo enfocado a la ingeniería de procesos y la simulación, que se puede utilizar para solucionar toda clase de problemas relacionados con procesos químicos. Este simulador cuenta con una interfaz muy amigable para el usuario, además de permitir el empleo de operadores lógicos y herramientas que facilitan la simulación de diversos procesos. Fue adquirido por AspenTech en el 2004 por lo que es desarrollado en la actualidad por Aspen Technology. Es un simulador bidireccional,

ya que el flujo de información va en dos direcciones (hacia delante y hacia atrás). De esta forma, puede calcular las condiciones de una corriente de entrada a una operación a partir de las correspondientes a la corriente de salida sin necesidad de cálculos iterativos. Posee un entorno de simulación modular tanto para estado estacionario como para régimen dinámico. Es un software para la simulación de plantas petroquímicas y afines. (HENAO y VÉLEZ, 2002)

1.3.1.3 CHEMCAD

CHEMCAD nace en 1984 cuando un profesor universitario formó un equipo para desarrollar un simulador de procesos para computadoras personales PC. El simulador fue vendido a la sección de software de McGraw Hill (COADE) y luego siguió siendo desarrollado y distribuido por Chemstations Inc. CHEMCAD ha venido evolucionando durante estos años para convertirse en un paquete de módulos que abarca cálculo y diseño de intercambiadores de calor (CC-THERM), simulación de destilaciones dinámicas (CC-DCOLUMN), simulación de reactores por lotes (CC-ReACS), simulación de destilaciones por lotes (CC-BATCH), simulación de redes de tuberías (CC-SAFETY NET). Recientemente ha sido puesta a la venta la versión 6 de CHEMCAD con una nueva interface de usuario y otras propiedades adicionales.(VERA, 2008)

Este sistema es muy usado en todo el mundo, para el diseño, operación y mantenimiento de procesos químicos en una gran variedad de industrias incluyendo la exploración de petróleo y gas; y naturalmente en procesos químicos, farmacéuticos, biocombustibles y procesos de fábricas industriales.

De forma general este software, como una herramienta de productividad tiene muchas ventajas entre las que cabe mencionar las siguientes:(PACHECO, 2006)

- ❖ Incremento en la productividad por el uso de información obtenida a partir de la simulación diaria de cálculos relacionados con las condiciones de operación.
- ❖ Maximizar la rentabilidad de las operaciones por el diseño más eficiente de nuevos procesos y equipos.
- ❖ Reducción de costos e inversiones de capital por la optimización y solución de los cuellos de botella existentes en los procesos y en los equipos.

1.3.2 Simuladores nacionales.

El empleo de programas de simulación de la industria química comenzó en Cuba en el ISPJAE (Instituto Superior Pedagógico José Antonio Echeverría) en el año 73. Este desarrollo comienza con la introducción del: GEMCS, producto de la universidad de McMaster, Canadá lo que condujo posteriormente a la creación del TERMOAZÚCAR basado en las modificaciones hechas al GEMCS para adaptarlo a la industria azucarera.

El GEMCS está conformado mayoritariamente por modelos determinísticos del comportamiento químico-físico de los procesos y operaciones unitarias, pero también emplea correlaciones estadísticas experimentales para calcular las propiedades físico-químicas y algunos criterios de eficiencia.

El surgimiento del TERMOAZÚCAR (PÉREZ *et al.*, 1989), un tanto revolucionó lo que se venía haciendo con el GEMCS, pues finalmente se obtenía una primera versión de un simulador puramente nacional y mejor adaptado a nuestras necesidades.

Además de lo explicado anteriormente hay otros simuladores y aplicaciones de procesos completos o subprocesos en Cuba como los relacionados con la producción de etanol (VEGA y SABADÍ, 1988); refinación de petróleo (MORÓN, 2000) y de sulfato ferroso (PEREDA, 1993). En el ICIDCA (Instituto Cubano de Investigaciones de los Derivados de la Caña de Azúcar) se dispone de un simulador, orientado a ecuaciones, del proceso azucarero y de alcohol, SIMFAD 3.0 (SABADÍ, 1991), resultado de todo un amplio trabajo de modelación matemática y simulación de equipos y subprocesos de la industria azucarera. En La UCLV (Universidad Central “Marta Abreu” de Las Villas) también se desarrolló un simulador de procesos tecnológicos (IZADA y VEGA, 1995), pero no aparecen referenciadas aplicaciones en la industria cubana. Otro simulador desarrollado en el MINAZ (Villa Clara) es el denominado AGE que contiene recomendaciones de cómo actuar en función de los resultados. Pero al igual que otros paquetes de simulación está basado en métodos de cálculos simplificados ó rápidos que no permiten obtener todos los resultados importantes y necesarios que pueden dar los simuladores que usan modelos más precisos, además estos métodos que no tienen en cuenta el ahorro energético.

La Industria Azucarera Cubana en estos últimos años ha estado sumergida en un proceso de modernización, con el objetivo fundamental de aumentar la calidad de las producciones y lograr una disminución en los costos buscando competitividad en un mercado abarrotado de azúcar de alta calidad a

muy bajos precios. En esta estrategia el mayor peso lo lleva el aumento de la producción cañera y los trabajos en la calidad y los costos del azúcar, pero existen otras variantes como la diversificación de las producciones que puede llegar a ser una alternativa muy valiosa para lograr una mayor revalorización de la caña, además de una mejor y más variada oferta de surtidos en el mercado.

Vinculadas a la industria azucarera y alcoholera se han desarrollado en nuestro país diversas bases de cálculos para la simulación del proceso, entre las que se tienen a TERMOAZÚCAR (ALEJO, 1989), ACOPLA (SUÁREZ, 1973), SIMFAD (SABADÍ, 1991) y SIDEL (GARRIDO, 2002). De estos programas desarrollados en nuestro país TERMOAZÚCAR es el único de naturaleza modular-secuencial y que utiliza un sistema experto para el Análisis de los Resultados (GOZÁ, 2002). Los programas de simulación existentes para La Industria Alcoholera en nuestro país, ACOPLA, SIMFAD y SIDEL, se caracterizan por desarrollar una simulación poco flexible. Por otra TERMOAZÚCAR cuenta con un solo módulo de cálculo para la simulación de destilerías de alcohol con posibilidades muy limitadas para la evaluación de distintas tecnologías. Sin embargo, su naturaleza modular-secuencial hace que esté abierto a un continuo perfeccionamiento y que tenga una gran flexibilidad para la evaluación integral y sistémica de distintos esquemas. Estas características, junto con la posibilidad de implementar un análisis automatizado de la simulación con un sistema experto, hacen de TERMOAZÚCAR una base de cálculo muy propicia para poder analizar con rapidez y rigor estudios energéticos que integren las industrias azucarera y alcoholera.

El programa TERMOAZÚCAR, se desarrolló inicialmente para efectuar la simulación en fábricas de azúcar crudo y azúcar refino y cuenta en su biblioteca con sólo un modelo para efectuar la simulación en destilerías de alcohol de manera muy limitada y sin aprovechar las ventajas del enfoque modular secuencial. La naturaleza modular secuencial le confiere las principales ventajas y características a este tipo de simuladores, como son: posibilidad de obtener gran cantidad de información precisa sobre los equipos y corrientes del proceso, fácil adición de nuevos módulos de cálculo, y gran flexibilidad para la representación detallada de cualquier configuración de los sistemas de vapor en fábricas de azúcar crudo y refino.

SIMFAD, un paquete de programas orientados a la simulación de procesos de la producción azucarera con fines de análisis. Está orientado tanto a los técnicos de la industria como a aquellos que laboran en áreas de diseño y análisis de procesos, resultando una poderosa herramienta de consultoría técnica. Con este simulador se puede estudiar esquemas de calentamiento y evaporación más eficientes, realizar balances de materiales y cálculo de purezas en el área de cocción y otras operaciones. El SIMFAD

(Simulación Fábricas de Azúcar y Derivados) es el primer producto comercial con un enfoque para la industria azucarera, permite optimizar el funcionamiento de determinados procesos y utilizar un producto cubano similar al sistema SUGARS norteamericano al que no se tiene acceso.

1.4 Generalidades sobre el Análisis de Resultados en los simuladores

En la actualidad los programas de simulación de procesos están jugando un importante papel en el desarrollo industrial. Existe una gran variedad de simuladores de procesos comerciales, los cuales son poderosas herramientas de cálculo en procesos industriales, ya que cuentan con propiedades termodinámicas y de transporte, que son fundamentales para efectuar los balances de materia y energía.

Dentro de ellos juega un rol primordial el Análisis de los Resultados de las simulaciones, etapa en la cual se realiza la recopilación y análisis de los datos arrojados en las mismas y que serán de gran utilidad para otras etapas y para optimizar resultados obtenidos.

De este modo las ventajas principales de un simulador de procesos se centran en minimizar el tiempo necesario para realizar cálculos; más aún, minimizar las fuentes de error humano generando de este modo resultados más precisos, que permiten al ingeniero ocuparse de otras tareas tales como el análisis de los datos, y modificación de las condiciones de operación que le permitan tener un conocimiento más profundo de lo que sucede en el proceso.

Por otra parte, la introducción al mercado de lenguajes de programación orientados a objetos han mejorado por mucho las interface de entrada y salida de datos, permitiendo el manejo de estos con mucha más facilidad. Todo esto se traduce finalmente en un ahorro muy significativo de tiempo para el desarrollo de nuevos procesos, o mejora de los ya existentes y por tanto de dinero.

Por ello resulta de vital importancia los métodos de Análisis de los Resultados que se obtienen del proceso de simulación con estas potentes herramientas.

Se debe tener en cuenta que los resultados de una simulación no son siempre fiables y estos se deben analizar críticamente. Hay que tener presente que los resultados dependen de:

- ❖ La calidad de los datos de entrada
- ❖ De que las correlaciones empleadas sean las apropiadas (escoger bien el paquete termodinámico)

- ❖ Elección adecuada del proceso.

Los simuladores tienen algunas herramientas para el Análisis de Resultados entre las que se puedan mencionar:

- ❖ Análisis de Sensibilidad:

Es una herramienta para determinar cómo reacciona un proceso cuando se varían variables claves de diseño y de operación. Se puede utilizar para variar una o más variables dentro de un proceso. Es una herramienta valiosa para desarrollar estudios “what if”. Las variables que varían deben ser entradas a la simulación pues las variables que son calculadas en la simulación no pueden ser manipuladas.

Un Análisis de Sensibilidad puede ser usado para verificar si una solución de una especificación de diseño está entre los rangos de la variable manipulada. También se puede utilizar para realizar optimizaciones de procesos.

Algunos simuladores poseen diferentes opciones al mostrar los resultados obtenidos de las simulaciones como por ejemplo:

- ❖ ASPEN PLUS:

- Generación de resultados en forma de grafica o en tablas.
- Exportar los resultados a hojas de cálculos.
- Análisis de Sensibilidad: Es una de las herramientas disponibles en este simulador que se puede acceder: Model Analysis \ Tools Sensitivity.

- ❖ CHEMCAD:

- Herramienta de Análisis de Sensibilidad que le permite probar escenarios sin la necesidad de modificar la simulación base e identificar situaciones críticas y óptimas.
- Análisis de optimización de los procesos simulados optimizando una función objetivo.

- ❖ STA- TERMOAZÚCAR:

- De los simuladores desarrollados en Cuba es el único de naturaleza secuencial-modular y que utiliza un sistema experto para el Análisis de los Resultados.(GOZÁ y PÉREZ, 2003)
- Además este simulador realiza diferentes tipos de Análisis de Resultados que permiten procesar la información obtenida de la simulación entre los que se encuentran:

- ✓ Análisis de Sensibilidad: Es muy útil, ya que permite ver cómo varían las variables seleccionadas por el usuario, en función de otra también seleccionada por este, denominada también como variable objetivo. Esta herramienta se usa como una aproximación para luego hacer cálculos de optimización y disponer de un buen valor inicial.
- ✓ Análisis de Dependencia: Es utilizado fundamentalmente para ver la relación que existe entre dos variables determinadas por el usuario con el objetivo de ver la dependencia de una con respecto a la otra.
- ✓ Análisis Económico: Es una herramienta muy útil que se utiliza para ver la factibilidad económica de un proceso determinado. Permite hacer la evaluación económica de procesos de modo que puede generar flujos de caja a partir de los cuales se hace el estudio de rentabilidad y así puede ayudar a tomar la decisión de tratar o no dicho proceso en base a los resultados de la evaluación económica obtenida.

1.5 Metodologías de Desarrollo de Software

El mundo de la informática no se detiene en su continuo desarrollo y uno de los temas que más se trata es el de procesos de desarrollo de software, el modo de trabajar eficientemente para evitar catástrofes que llevan a que un gran porcentaje de proyectos terminen sin éxito.

Uno de los objetivos de un proceso de desarrollo es elevar la calidad del software (en todas las fases por las que pasa) a través de una mayor transparencia y control sobre el proceso. Da lo mismo que sea algo casero o para un cliente, lo más importante es que hay que producir lo esperado en el tiempo esperado y con el coste previsto.

Una metodología no es más que un conjunto de procedimientos, técnicas, herramientas y un soporte documental que ayuda a realizar un software. Se definen tareas que constituyen las actividades elementales en que se dividen los procesos y permiten ser ejecutadas a través de los procedimientos. Para ello se desarrollan técnicas que se pueden utilizar una o varias veces y herramientas software que automatizan la aplicación. Como resultado en cada etapa se obtiene un producto. Una metodología puede

seguir uno o varios modelos de ciclo de vida, es decir, el ciclo de vida indica que es lo que hay que obtener a lo largo del desarrollo del proyecto pero no como hacerlo.(PATÓN, 2007)

Se puede decir que en estos últimos años se han desarrollado dos corrientes en lo referente a los procesos de desarrollo. Por una parte se encuentran las metodologías tradicionales o pesadas entre las que se encuentran el Proceso Unificado (RUP), Microsoft Solution Framework (MSF) mientras que por la otra están las metodologías ágiles de las cuales se pueden mencionar Extreme Programming (XP), SCRUM y Dynamic System Development Method (DSDM). La diferencia fundamental entre ellas estada dada en que mientras las tradicionales intentan lograr el objetivo común por medio de orden y documentación las ágiles tratan de mejorar la calidad del software a través de una comunicación directa e inmediata entre las personas que intervienen en el proceso.

1.5.1 Proceso Unificado de Desarrollo (RUP)

La metodología RUP (Proceso Unificado de Desarrollo), llamada así por sus siglas en inglés Rational Unified Process, es un proceso de desarrollo de software. RUP es un proceso para el desarrollo de un producto que define claramente quién, cómo, cuándo y qué debe hacerse en el proyecto. Esta metodología constituye una guía rectora que dirige las actividades que se realizan por rol, les da un orden, especifica qué artefactos deberían ser desarrollados y puede además monitorear y medir los productos y actividades de un proyecto, teniendo en cuenta la calidad del producto final.

Las características esenciales de RUP son:

- ❖ **Dirigido por los Casos de Uso:** que orientan el proyecto a la importancia para el usuario y lo que este quiere.
- ❖ **Centrado en la arquitectura:** que relaciona la toma de decisiones que indican cómo tiene que ser construido el sistema y en qué orden.
- ❖ **Iterativo e incremental:** donde divide el proyecto en mini-proyectos donde los casos de uso y la arquitectura cumplen sus objetivos de manera más depurada.

Como filosofía RUP maneja 6 principios clave (GALLEGO, 2007):

- ❖ **Adaptación del proceso:** El proceso deberá adaptarse a las características propias de la organización. El tamaño del mismo, así como las regulaciones que lo condicionen, influirán en su diseño específico. También se deberá tener en cuenta el alcance del proyecto.

- ❖ **Balancear prioridades:** Los requerimientos de los diversos inversores pueden ser diferentes, contradictorios o disputarse recursos limitados. Debe encontrarse un balance que satisfaga los deseos de todos.
- ❖ **Colaboración entre equipos:** El desarrollo de software no lo hace una única persona sino múltiples equipos. Debe haber una comunicación fluida para coordinar requerimientos, desarrollo, evaluaciones, planes, resultados y otros.
- ❖ **Demostrar valor iterativamente:** Los proyectos se entregan, aunque sea de un modo interno, en etapas iteradas. En cada iteración se analiza la opinión de los inversores, la estabilidad y calidad del producto, y se refina la dirección del proyecto así como también los riesgos involucrados
- ❖ **Elevar el nivel de abstracción:** Este principio dominante motiva el uso de conceptos reutilizables tales como patrón del software, lenguajes 4GL o esquemas (*frameworks*) por nombrar algunos. Éstos se pueden acompañar por las representaciones visuales de la arquitectura, por ejemplo con UML.
- ❖ **Enfocarse en la calidad:** El control de calidad no debe realizarse al final de cada iteración, sino en todos los aspectos de la producción

El ciclo de vida de RUP

RUP divide el proceso en 4 fases, dentro de las cuales se realizan varias iteraciones en número variable según el proyecto y en las que se hace un mayor o menor hincapié en las distintas actividades. En las iteraciones de cada fase se hacen diferentes esfuerzos en diferentes actividades.

- ❖ **Inicio:** Se hace un plan de fases, se identifican los principales casos de uso y se identifican los riesgos, es donde se determina la visión del proyecto. Se define el alcance del proyecto.
- ❖ **Elaboración:** Se hace un plan de proyecto, se completan los casos de uso y se eliminan los riesgos. En esta etapa se determinan los cimientos de la arquitectura
- ❖ **Construcción:** se concentra en la elaboración de un producto totalmente operativo y eficiente y el manual de usuario. Se obtiene la capacidad operacional inicial del producto.
- ❖ **Transición:** Se obtiene la liberación del producto y se pone en manos de los usuarios finales.

Los elementos del RUP son:

- ❖ **Actividades,** Son los procesos que se llegan a determinar en cada iteración.

- ❖ **Trabajadores**, Vienen hacer las personas o entes involucrados en cada proceso.
- ❖ **Artefactos**, Un artefacto puede ser un documento, un modelo, o un elemento de modelo.

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.(SANCHEZ, 2004)

Ventajas de RUP

- ❖ Valoración en cada fase que permite cambios de objetivos.
- ❖ Funciona bien en proyectos de creación.
- ❖ Análisis y seguimiento detallado en cada una de las fases.

Desventajas de RUP

- ❖ La evaluación de riesgos es compleja.
- ❖ El cliente deberá ser capaz de describir y entender a un gran nivel de detalle para poder acordar un alcance del proyecto con él.

1.5.2 Extreme Programming (XP)

Es una de las metodologías de desarrollo de software más exitosas en la actualidad utilizada para proyectos de corto plazo, equipo pequeño y cuyo plazo de entrega era ayer. La metodología consiste en una programación rápida o extrema, cuya particularidad es tener como parte del equipo, al usuario final, pues es uno de los requisitos para llegar al éxito del proyecto.

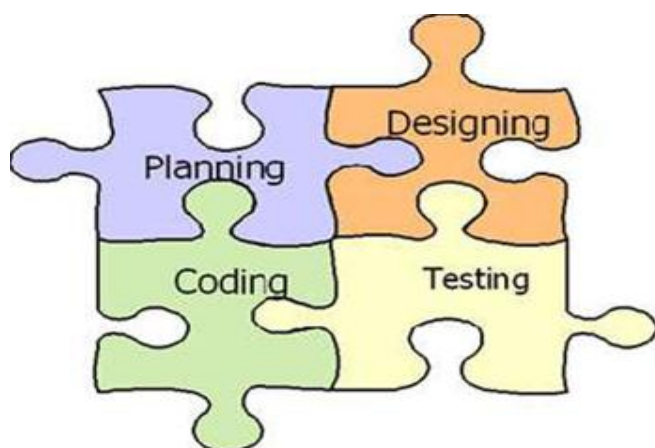


Figura 2: Metodología Extreme Programming

Características de XP

La metodología se basa en:

- ❖ **Pruebas Unitarias:** se basa en las pruebas realizadas a los principales procesos, de tal manera que adelantándose en algo hacia el futuro, poder hacer pruebas de las fallas que pudieran ocurrir. Es como si se adelantara a obtener los posibles errores.
- ❖ **Re-fabricación:** 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. Es como el chofer y el copiloto: mientras uno conduce, el otro consulta el mapa.

¿Qué es lo que propone XP?

- ❖ Empieza en pequeño y añade funcionalidad con retroalimentación continua.
- ❖ El manejo del cambio se convierte en parte sustantiva del proceso.
- ❖ El costo del cambio no depende de la fase o etapa.
- ❖ No introduce funcionalidades antes que sean necesarias.
- ❖ El cliente o el usuario se convierte en miembro del equipo.

Lo fundamental en este tipo de metodología 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.(SANCHEZ, 2004)

1.5.3 Microsoft Solution Framework (MSF)

Esta es una metodología flexible e interrelacionada con una serie de conceptos, modelos y prácticas de uso, que controlan la planificación, el desarrollo y la gestión de proyectos tecnológicos. MSF se centra en los modelos de proceso y de equipo dejando en un segundo plano las elecciones tecnológicas.



Figura 3: Metodología MSF

MSF tiene las siguientes características:

- ❖ **Adaptable:** es parecido a un compás, usado en cualquier parte como un mapa, del cual su uso es limitado a un específico lugar.
- ❖ **Escalable:** puede organizar equipos tan pequeños entre 3 o 4 personas, así como también, proyectos que requieren 50 personas a más.
- ❖ **Flexible:** es utilizada en el ambiente de desarrollo de cualquier cliente.
- ❖ **Tecnología Agnóstica:** porque puede ser usada para desarrollar soluciones basadas sobre cualquier tecnología.

MSF se compone de varios modelos encargados de planificar las diferentes partes implicadas en el desarrollo de un proyecto: Modelo de Arquitectura del Proyecto, Modelo de Equipo, Modelo de Proceso, Modelo de Gestión del Riesgo, Modelo de Diseño de Proceso y finalmente el modelo de Aplicación.(SANCHEZ, 2004)

1.5.4 SCRUM

SCRUM es un proceso ágil que sirve para administrar y controlar el desarrollo de software. El desarrollo se realiza en forma iterativa e incremental (una iteración es un ciclo corto de construcción repetitivo). Cada ciclo o iteración termina con una pieza de software ejecutable que incorpora nueva funcionalidad. Las

iteraciones en general tienen una duración entre 2 y 4 semanas. SCRUM se utiliza como marco para otras prácticas de ingeniería de software como RUP o Extreme Programming.

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, por ejemplo en un mercado de alta competitividad. 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. Se busca entregar software que realmente resuelva las necesidades, aumentando la satisfacción del cliente. En SCRUM, el equipo se focaliza en una única cosa: construir software de calidad. Por el otro lado, la gestión de un proyecto SCRUM se focaliza en definir cuáles son las características que debe tener el producto a construir (qué construir, qué no y en qué orden) y en remover cualquier obstáculo que pudiera entorpecer la tarea del equipo de desarrollo. Se busca que los equipos sean lo más efectivos y productivos que sea posible.

SCRUM tiene un conjunto de reglas muy pequeño y muy simple y está basado en los principios de inspección continua, adaptación, auto-gestión e innovación. El cliente se entusiasma y se compromete con el proyecto dado que ve crecer el producto iteración a iteración y encuentra las herramientas para alinear el desarrollo con los objetivos de negocio de su empresa. Por otro lado, los desarrolladores encuentran un ámbito propicio para desarrollar sus capacidades profesionales y esto resulta en un incremento en la motivación de los integrantes del equipo.

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.(SCHWABER, 2004)

1.6 Lenguaje Unificado de Modelado (UML).

UML (Unified Modeling Language) es un lenguaje de modelado visual que se usa para especificar, visualizar, construir y documentar artefactos de un sistema de software orientado a objetos.

Visualizar: UML permite expresar de una forma gráfica un sistema de forma que otro lo puede entender.

Especificar: UML permite especificar cuáles son las características de un sistema antes de su construcción.

Construir: A partir de los modelos especificados se pueden construir los sistemas diseñados.

Documentar: Los propios elementos gráficos sirven como documentación del sistema desarrollado que pueden servir para su futura revisión.

Actualmente UML ofrece un estándar para describir un “plano” del sistema incluyendo aspectos conceptuales como procesos del negocio y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software utilizados.

Es importante destacar que UML es un lenguaje de modelado, no un método o un proceso, se emplea para definir, detallar y documentar los artefactos de un sistema de software. En la última versión se adicionaron diversas novedades que resuelven carencias desde el punto de vista práctico fundamentalmente. Entre los diagramas que propone UML para modelar un sistema se encuentran (JACOBSON, I *et al.*, 2000a):

- ❖ Diagramas de estructura (clases, componentes, objetos, despliegue. paquetes)
- ❖ Diagramas de comportamiento (actividades, casos de uso, estado)
- ❖ Diagramas de interacción (secuencia, comunicación)

1.7 Herramientas CASE

Las herramientas de modelado visual permiten la representación de un producto de software de forma completa mediante diagramas que se van desarrollando durante los ciclos de vida del proyecto y proporcionan el desarrollo de un proceso. Entre las más usadas actualmente se encuentra el Rational Rose y el Visual Paradigm.

1.7.1 Rational Rose

Rational Rose es una herramienta software para el Modelado Visual mediante UML de sistemas software. Permite especificar, analizar y diseñar el sistema antes de codificarlo. Rational Rose es la herramienta CASE desarrollada por los creadores de UML (Booch, Rumbaugh y Jacobson), que cubre todo el ciclo de

vida de un proyecto: concepción y formalización del modelo, construcción de los componentes, transición a los usuarios y certificación de las distintas fases.

Características Rational Rose:

- ❖ “Mantiene la consistencia de los modelos del sistema software.
- ❖ Chequeo de la sintaxis UML.
- ❖ Generación de documentación automáticamente.
- ❖ Generación de código a partir de los modelos.
- ❖ Ingeniería Inversa (crear modelo a partir código). ”(JACOBSON, I *et al.*, 1999)

1.7.2 Visual Paradigm

Visual Paradigm para UML es una herramienta UML 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 UML ayuda a una más rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación.

Herramienta CASE que da soporte al modelado visual con UML 2.0. Dentro de las características que ofrece se encuentran las siguientes:

- ❖ Entorno de creación de diagramas para UML 2.0.
- ❖ Diseño centrado en casos de uso y enfocado al negocio que genera un software de mayor calidad.
- ❖ Uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación.
- ❖ Capacidades de ingeniería directa (versión profesional) e inversa.
- ❖ Modelo y código que permanece sincronizado en todo el ciclo de desarrollo.
- ❖ Disponibilidad de múltiples versiones, para cada necesidad.
- ❖ Disponibilidad de integrarse en los principales IDEs.
- ❖ Disponibilidad en múltiples plataformas.
- ❖ Es posible importar un área de trabajo (Workspace) (SIERRA, 2007)

1.8 Metodología y herramientas a utilizar.

Después de haber hecho un análisis de algunas de las principales herramientas y metodologías existentes se ha podido apreciar las principales ventajas y desventajas de las mismas. La metodología XP es muy utilizada pero requiere una fuerte vinculación del cliente con el equipo de desarrollo lo cual en ocasiones es un poco complicado de lograr, por otra parte se encuentra MSF que se presenta como una metodología flexible lo cual tiene gran ventaja puesto que se puede aplicar a diferentes tipos de proyectos pero su desarrollo está basado en Microsoft la cual es propietaria y su uso cuesta mucho dinero por lo que limita la utilización de muchos software. Por todo lo anterior se decide utilizar en el desarrollo del trabajo la metodología RUP por ser una de las más utilizada y difundida en la actualidad, por su flexibilidad, capacidad de adaptación para los distintos tipos de proyectos y robustez. Hay que señalar también que es una metodología que define claramente las actividades que se realizan por roles y la creación de artefactos de vital importancia en la construcción del producto software. Además se escoge como herramienta case a el Rational Rose ya que soporta muy bien esta metodología y al lenguaje de modelación visual que se utiliza (UML). Además permite una modelación absoluta de los procesos del negocio y del sistema y resulta de gran utilidad para los desarrolladores de proyectos, pues ella cubre todo el ciclo de vida de este, desde su fase inicial hasta su culminación.

1.9 Conclusiones

Después de abordar un tema tan importante para el desarrollo industrial como lo es hoy en día la simulación y más aún la simulación de procesos en particular, y de haber caracterizado algunos de los programas de simulación más conocidos en estos momentos se espera que los lectores puedan obtener el suficiente conocimiento sobre el contenido. Además se ofrecen elementos distintivos de las principales herramientas y metodologías que se utilizan en el mundo para el desarrollo de software y se da a conocer la escogida por el autor, pero lo más importante es que a través del trabajo los usuarios puedan elegir por sí mismos.

CAPÍTULO 2: DESCRIPCIÓN DEL SISTEMA.

2 Introducción

En este capítulo se aborda el funcionamiento general del subsistema mediante un modelo de dominio. Además se identifican los principales requisitos funcionales que el sistema debe cumplir así como los no funcionales, también se describen los actores que se identificaron al igual que los principales casos de usos que tendrá el subsistema y sus relaciones mostrados en el diagrama de casos de uso para mejorar su comprensión.

2.1 Modelo de dominio

El Modelo de Dominio (Modelo Conceptual) es una representación visual de los conceptos u objetos del mundo real significativos para un problema o área de interés. Representa clases conceptuales del dominio del problema, conceptos del mundo real, no de los componentes de software. En este modelo no se muestra comportamiento. Las clases conceptuales pueden tener atributos pero no métodos.(PRESSMAN, 2002)

El modelo del dominio se describe mediante diagramas UML, específicamente mediante diagramas de clases conceptuales significativas en el dominio del problema, donde se muestran a los clientes, usuarios, revisores y a otros desarrolladores las clases del dominio y cómo se relacionan unas con otras a través de asociaciones.(JACOBSON, IVAR *et al.*, 2000b)

Su principal objetivo es ayudar a los implicados en el desarrollo del producto a utilizar un vocabulario común que posibilite una mejor comprensión entre ellos, para poder entender y describir las clases más importantes dentro del contexto donde se ubica el sistema. Además permite y facilita el levantamiento de requisitos así como definir los procesos y roles más significativos.

2.1.1 ¿Por qué hacer modelo de dominio?

El primer paso del modelado del negocio consiste en capturar los procesos de negocio de la organización bajo estudio. La definición del conjunto de procesos del negocio es una tarea crucial, ya que define los límites del proceso de modelado posterior. Teniendo en cuenta lo anteriormente dicho no es posible

realizar un modelo de negocio en la presente investigación pues no existe ningún límite en el negocio que se desarrolla. Por todo esto y por el bajo nivel de estructuración que presenta los procesos se propone un modelo del dominio ayudando a los usuarios, clientes, desarrolladores y demás interesados, a utilizar un vocabulario común para poder entender el contexto en que se emplaza el sistema, además de que no es viable realizar un modelo de negocio en estas condiciones pues son necesarios un conjunto de datos manipulados por una serie de tareas en las que participan varios representantes del negocio de acuerdo a un flujo de trabajo, necesitándose así la realización de un conjunto de procesos que no están claros en este trabajo por lo que se hace necesario describir el problema de otra manera, es por eso que se hace el modelo de dominio también conocido con un modelo conceptual. Con este se logrará un glosario o diccionario de clases que sirva como común denominador a todos los componentes del sistema, incluyendo a las personas involucradas a lo largo del desarrollo. Además el modelo de dominio puede ser tomado como el punto de partida para el diseño del sistema pues al realizar una programación orientada a objetos como es el caso, el mapa de conceptos de este modelo constituye una primera versión del sistema imitando así en alguna medida la realidad y cuya función principal es ayudar a comprender el problema a tratar.

2.1.2 Conceptos del dominio.

Variables independientes: Son aquellas que se pueden manipular en un experimento con el objetivo de ver su influencia en las variables dependientes.

Variable objetivo: Es la variable que se toma de referencia en un método de análisis para ver su comportamiento cuando son manipuladas algunas de las variables independientes a ella. Es la que refleja los resultados de un estudio de investigación.

Indicador: Es un factor que determina el estado de un elemento determinado en la simulación de un proceso.

Simulación: Permite representar un proceso de forma virtual obteniendo resultados de su posible funcionamiento en la realidad.

Resultados: Se denomina resultado al proceso de análisis de los datos obtenidos de una o más simulaciones.

Métodos de análisis: Se pueden considerar herramientas que se utilizan para procesar los resultados.

Análisis de Sensibilidad: Es un tipo de análisis que consiste en manipular las variables independientes para observar cómo influye sobre la variable objetivo.

Análisis Económico: Tipo de análisis que dado las variables de entradas con los gastos que estas puedan tener y las de las salidas con el precio de las mismas permite verificar si es factible realizar un proceso determinado.

Análisis de Dependencia: Representa la dependencia entre variables en un rango especificado por el usuario.

Análisis de Indicadores: Tipo de análisis en el cual se calculan los indicadores que representan el estado en que se encuentran los elementos utilizados en la simulación de un proceso.

Especialista: Persona con conocimientos de ingeniería de procesos químicos que desea simular una fábrica o un proceso en el simulador.

Variable Económica: Variables que son especificadas por el usuario para realizar el Análisis Económico.

2.1.3 Diagrama del modelo del dominio.

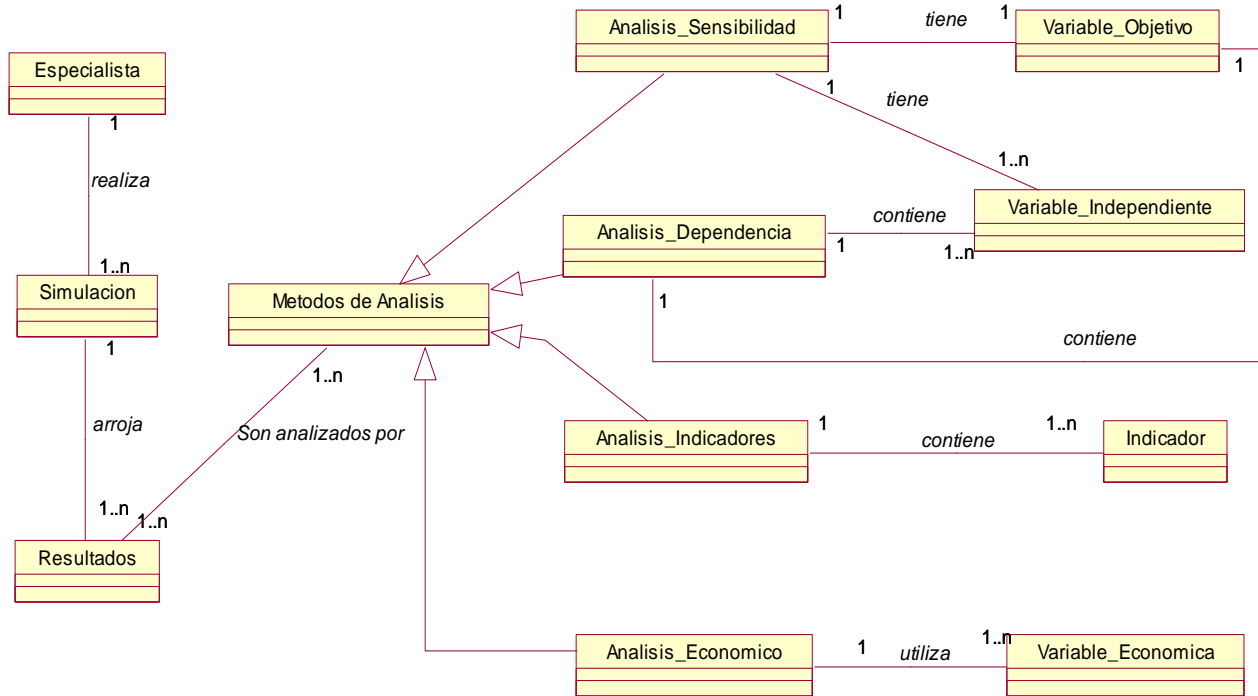


Figura 4: Diagrama del modelo de dominio

2.1.4 Descripción del Modelo de dominio.

El especialista se encarga de realizar una o más simulaciones de su interés de las cuales se obtienen numerosos resultados que pueden ser analizados a través de diferentes métodos de análisis los que permiten una mejor comprensión de estos datos obtenidos. Cada uno de estos métodos posee características diferentes y utilizan variables que hacen posible un análisis más completo de los resultados.

2.2 Especificación de los requisitos de software del sistema

Una especificación de requisitos de software es una descripción completa del comportamiento del sistema a desarrollar. Una vez en este punto, se hace necesario comenzar a definir qué es lo que debe hacer el sistema, siendo necesario ir a la captura de los requerimientos que este debe cumplir. Según Roger S Pressman para que un esfuerzo de desarrollo de software tenga éxito, es esencial comprender perfectamente los requisitos del software.

Los requerimientos se utilizan con un sentido general, refiriéndose a necesidades.(JACOBSON, I *et al.*, 1999)

Un requisito (o requerimiento) es una característica de diseño, una propiedad o un comportamiento de un sistema. Los requisitos constituyen la descripción de los deseos o de las necesidades de un producto. (JACOBSON, I *et al.*, 2004)

2.2.1 Requerimientos Funcionales.

Los requerimientos funcionales son capacidades o condiciones que el sistema debe cumplir. Son el punto de partida para identificar qué debe hacer el sistema. Los requerimientos funcionales definen las funciones que el sistema será capaz de realizar.(JACOBSON, I *et al.*, 1999)

R1. Realizar Análisis de Sensibilidad

R1.1. Especificar variables independientes del Análisis de Sensibilidad.

R1.2 Especificar variable objetivo del Análisis de Sensibilidad.

R1.3 Especificar valor nuevo de variable independiente.

R2 Mostrar gráfico con resultados del Análisis de Sensibilidad.

R3 Generar reporte con resultados del Análisis de Sensibilidad.

R4 Realizar Análisis de Dependencia.

R4.1 Especificar variable independiente del Análisis de Dependencia

R4.2 Especificar variable objetivo del Análisis de Dependencia.

R4.3 Especificar valores de la variable independiente.

R4.3.1 Analizar mediante un intervalo y un paso.

R4.3.2 Analizar indicando los valores.

R5 Mostrar gráfico con resultados del Análisis de Dependencia.

R6 Realizar Análisis de Indicadores

R6.1 Definir indicador

R6.2 Seleccionar indicadores a analizar

R6.3 Especificar los indicadores

R7 Mostrar reporte del Análisis de Indicadores.

R8 Realizar Análisis Económico.

R8.1 Definir ingresos

R8.2 Definir egresos

R8.3 Especificar variables económicas.

R8.4 Guardar Configuración de ingresos y egresos del análisis.

R8.5 Cargar Configuración de ingresos y egresos del análisis.

R8.6 Especificar valores de los ingresos.

R8.7 Especificar valores de los egresos.

R8.8 Especificar costos de los ingresos

R8.9 Especificar precios de los egresos.

R9 Reportar resultados del Análisis Económico.

2.2.2 Requerimientos no funcionales.

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Representan las características que hacen al producto atractivo, usable, rápido o confiable.

Los requisitos no funcionales especifican propiedades del sistema, como restricciones del entorno o de la implementación, rendimiento, dependencias de la plataforma, facilidad de mantenimiento, extensibilidad y fiabilidad.(JACOBSON, I *et al.*, 1999)

- Apariencia o interfaz externa.

La interfaz de manera general es sencilla, debe permitir iconos de acceso directo para las opciones de Análisis de Sensibilidad, Dependencia, Indicadores y Análisis Económico.

- Usabilidad.

La aplicación podrá ser usada por cualquier investigador capacitado, sobre todo ingenieros químicos. Contendrá un manual de usuario con las descripciones básicas sobre cómo trabajar con la aplicación incluyendo las especificaciones o características del módulo. No tiene restricciones de usuarios.

➤ Rendimiento.

Se realizará el Análisis de los Resultados una vez que se dispongan de los datos necesarios para ello.

➤ Soporte.

Una vez terminada la aplicación será entregada a los usuarios junto con un CD con las indicaciones de instalación del software.

➤ Seguridad.

- ❖ Confiability: Tiene relación con la protección de información frente a posibles accesos no autorizados. Es le encargada de asegurar que sólo los individuos autorizados tengan acceso a los recursos que se intercambian. En la aplicación no existen restricciones de acceso a la información manejada.(MARJEDAN, 2002)
- ❖ Integridad: Se refiere a la protección de información, datos, sistemas y otros activos informáticos contra cambios o alteraciones en su estructura o contenido ya sean intencionados, no autorizados o casuales. Los activos o la información solo pueden ser modificados por las personas autorizadas y de la forma autorizada.(MARJEDAN, 2002)
- ❖ Disponibilidad: Es la garantía de que los usuarios autorizados puedan acceder a la información y recursos cuando los necesiten. El sistema brinda información sobre al Análisis de los Resultados siempre que el personal lo estime conveniente.(MARJEDAN, 2002)

➤ De Software

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

➤ De Hardware

Para instalar la aplicación se necesitan máquinas con 2 GB de disco duro disponibles, 256 MB de RAM o superior y procesadores a 2.0 GHz o superior.

➤ Restricciones en el Diseño y la Implementación

- ❖ Se usará como lenguaje de programación C# (C Sharp).
- ❖ Como herramienta CASE Rational Rose para el modelado de los artefactos que se generen.
- ❖ Se utiliza UML como lenguaje de modelado.

2.3 Definición y descripción de los casos de uso del sistema

2.3.1 Definición de los actores

El término actor significa el rol que algo o alguien juega cuando interactúa con el sistema. Un candidato a actor del sistema es cualquier individuo, grupo, organización o máquina que interactúa en los casos de uso. De acuerdo con esta idea un actor del sistema representa un tipo particular de usuario del sistema más que un usuario físico, ya que varios usuarios físicos pueden realizar el mismo papel en relación al negocio, o sea, ser instancias de un mismo actor. Una vez que se han identificado los actores del sistema, se tiene identificado el entorno externo del sistema.(JACOBSON, IVAR *et al.*, 2000b)

Actor	Descripción
Especialista	Persona con conocimientos de ingeniería de procesos químicos que desea simular algún proceso o una fábrica en el simulador.

Tabla 1: Definición de los actores

2.3.2 Diagrama de casos de uso del sistema.

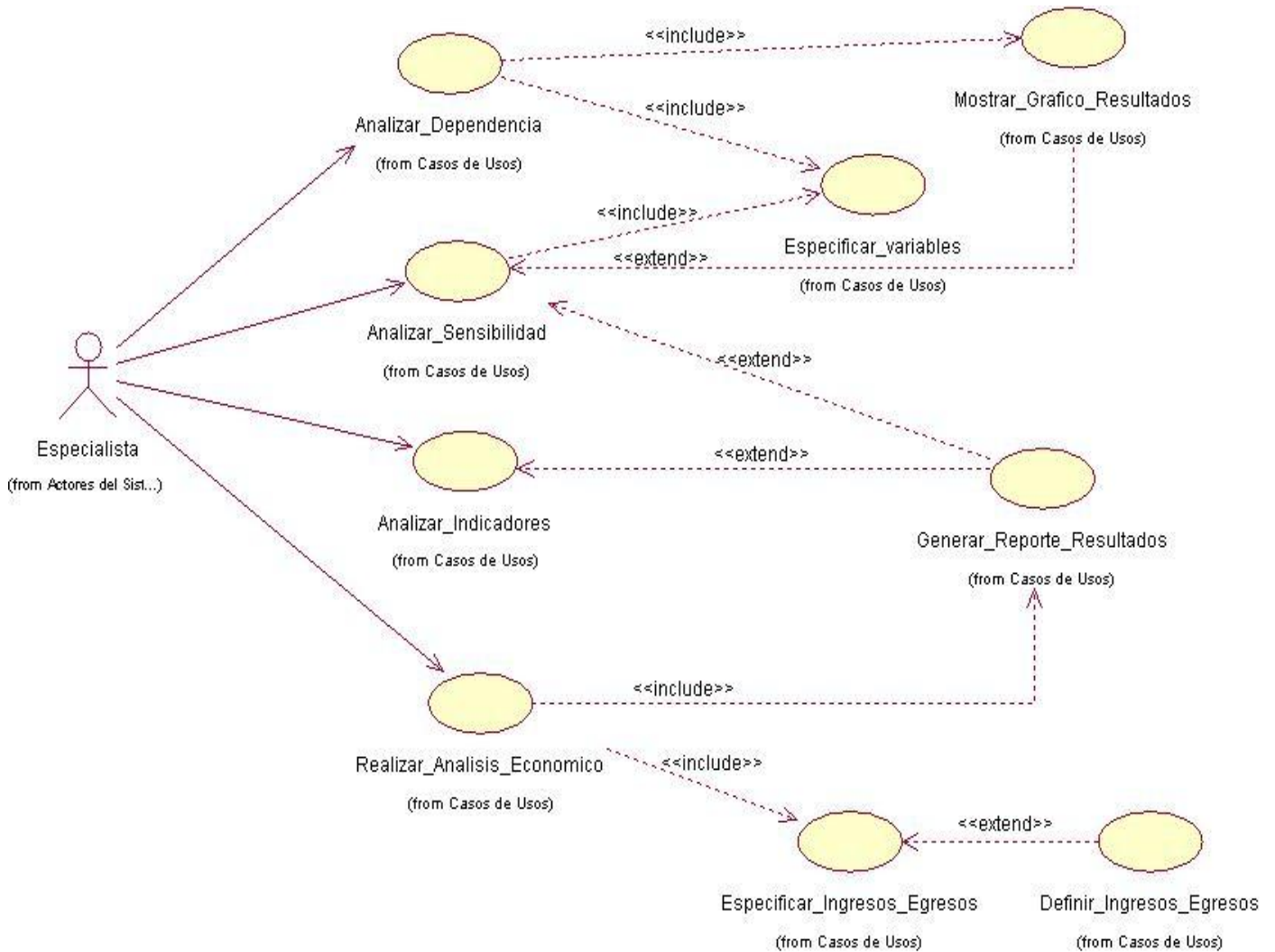


Figura 5: Diagrama de casos de uso del sistema

2.3.3 Descripción textual de los casos de uso del sistema.

A continuación se muestra un ejemplo correspondiente al caso de uso Realizar Análisis Económico. Para consultar los restantes diagramas dirigirse al [anexo 1](#).

2.3.3.1 Descripción textual del caso de uso “Realizar_Análisis_Económico”

Caso de Uso:	Realizar_Análisis_Económico
Actores:	Especialista
Resumen:	El CU comienza cuando el Especialista desea realizar el Análisis Económico a los resultados de la simulación. Esta funcionalidad permitirá hacer Análisis Económico sobre los resultados de la simulación.
Precondiciones:	El sistema debe haber simulado
Referencias	<p>R8 Realizar Análisis Económico.</p> <ul style="list-style-type: none"> R8.1 Definir ingresos R8.2 Definir egresos R8.3 Especificar variables económicas. R8.4 Guardar Configuración de ingresos y egresos del análisis. R8.5 Cargar Configuración de ingresos y egresos del análisis. R8.6 Especificar valores de los ingresos. R8.7 Especificar valores de los egresos. R8.8 Especificar costos de los ingresos R8.9 Especificar precios de los egresos. <p>R9 Reportar resultados del Análisis Económico.</p>
CU Asociados	<p>Definir_Ingresos_Egresos (<i>Extend</i>)</p> <p>Especificar_Ingresos_Egresos (<i>Include</i>)</p> <p>Generar_Reporte_Resultados (<i>Include</i>)</p>
Prioridad	--

Flujo Normal de Eventos

Acción del Actor

Respuesta del Sistema

<p>1 El especialista escoge la opción de realizar el Análisis Económico en el menú de la barra de herramientas.</p> <p>2 Especifica las variables económicas solicitadas y pulsa el botón ver análisis.</p>	<p>1.1 El sistema inicializa el caso de uso “Especificar_Ingresos_Egresos”. (Ver sección “Especificar_Ingresos_Egresos”)</p> <p>1.2 El sistema solicita la especificación de las variables económicas para el análisis.</p> <p>2.1 Realiza el Análisis Económico y muestra los resultados en un reporte generado con los mismos.</p>
-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Prototipo de Interfaz

Sección “Especificar_Ingresos_Egresos”

Flujo Normal de Eventos

Acción del Actor

Respuesta del Sistema

<p>2. El especialista especifica los ingresos y los egresos para el análisis.</p> <p>3. Selecciona los datos de los ingresos y de los egresos.</p> <p>4. Introduce los datos solicitados.</p>	<p>2.1 El sistema muestra las opciones necesarias para que el usuario especifique los ingresos y los egresos deseados y brinda la posibilidad de definir otros de interés para el usuario. En caso de escoger esta opción el sistema inicializa el caso de uso “Definir_Ingresos_Egresos”. (Ver descripción del caso de uso).</p> <p>3.1 El sistema solicita los datos de los precios de los ingresos y los costos de los productos egresados.</p> <p>4.1 El sistema guarda la configuración.</p>
Prototipo de Interfaz	
Poscondiciones	

Tabla 2: Descripción del CU Realizar Análisis Económico

2.4 Patrones de casos de uso.

Los patrones de casos de uso son comportamientos que deben existir en el sistema, ayudan a describir qué es lo que el sistema debe hacer, es decir, describen el uso del sistema y cómo este interactúa con los usuarios. Estos patrones son utilizados generalmente como plantillas que describen como debería ser estructurados y organizados los casos de uso.(LARMAN, 1999)

2.4.1 Patrones de casos de usos utilizados.

Patrón: Concordancia (Commonality)

Extrae una subsecuencia de acciones que aparecen en diferentes lugares del flujo de Casos de Uso y es expresado por separado.

Reuso

Consta de 3 casos de uso. El primero llamado subsecuencia común, modela una secuencia de acciones que aparecerán en múltiples casos de uso en el modelo. Los otros casos de uso modelan el uso del sistema que comparte la subsecuencia común de acciones. De manera que deben existir al menos dos de ellos. A continuación se muestra un ejemplo de su uso en el diagrama:

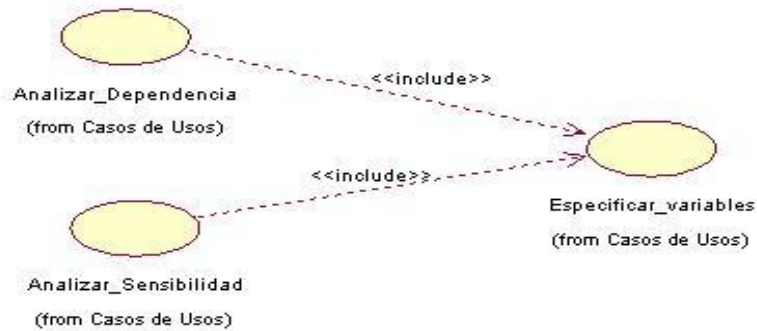


Figura 6: Concordancia (Modelo de Reuso)

Adición

En el caso de este patrón alternativo, la subsecuencia común de casos de uso, extiende los casos de uso compartiendo la subsecuencia de acciones. Los otros casos de uso modelan el flujo que será expandido con la subsecuencia. Este patrón es preferible usarlo cuando otros casos de uso se encuentran propiamente completos, o sea, que no requieren de una subsecuencia común de acciones para modelar los usos completos del sistema. Ejemplo de su uso, se muestra en la siguiente figura:

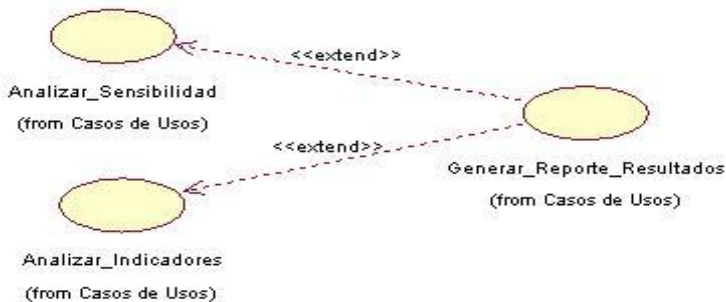


Figura 7: Concordancia (Modelo de Adición)

Patrón: Extensión Concreta o Inclusión

Este patrón está dividido en concreta extensión o concreta inclusión.

Extensión:

Consiste en dos casos de uso y una relación extendida entre ellos. Puede ser instalado en sí mismo, así como extendido en el caso de uso base. El referente puede ser concreto o abstracto. Este patrón se aplica cuando un flujo puede extender el flujo de otro caso de uso así como ser realizado en sí mismo. Un ejemplo de su uso se evidencia en la siguiente figura:

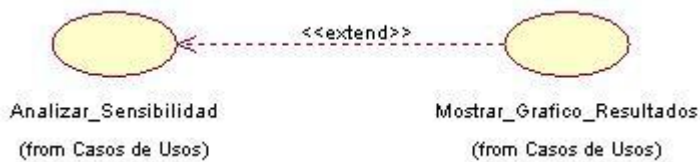


Figura 8: Extensión Concreta

Inclusión:

Se incluye una relación del caso de uso base al caso de uso de inclusión. El último puede ser instalado en sí mismo. El caso de uso base puede ser concreto o abstracto. Como ejemplo de su aplicación en el diagrama lo evidencia la figura a continuación mostrada:

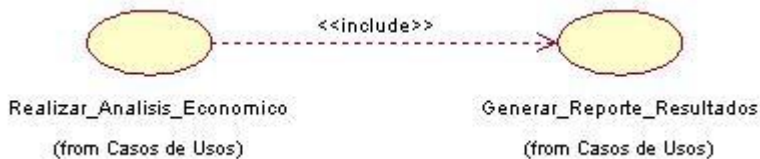


Figura 9: Inclusión Concreta

2.5 Conclusiones

Con el desarrollo de este capítulo se espera una mejor comprensión del sistema en desarrollo pues se han abordado temas importantes como los principales conceptos relacionados con el dominio del problema. Además se dieron a conocer las principales funcionalidades que tendrá el sistema y los requisitos que deben cumplirse para obtener un mejor resultado en su uso.

CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

3 Introducción

En el presente capítulo se expone el análisis y diseño propuesto para la solución del subsistema de Análisis de Resultados, modelándose los artefactos necesarios que contribuyen a la implementación del sistema. Estos artefactos son el Modelo de Análisis y el Modelo de Diseño, los cuales se encargan de describir en términos de clases cómo deben funcionar los casos de uso del sistema. Aquí también se logra representar la colaboración entre estas clases y se dan a conocer los patrones de diseño que se utilizan en la elaboración de las mismas.

3.1 Modelo de Análisis

El Modelo de Análisis, realmente un conjunto de modelos, es la primera representación técnica de un sistema. El Modelo de Análisis 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.

El propósito del análisis es definir todas las clases que son relevantes al problema que se va a resolver, las operaciones y atributos asociados, las relaciones y comportamientos asociadas con ellas.(PRESSMAN, 2002)

Este modelo es usado para representar la estructura global del sistema, describe la realización de casos de uso y sirve como una abstracción del Modelo de Diseño. El Modelo de Análisis puede contener: las clases y paquetes de análisis, las realizaciones de los casos de uso, las relaciones y los diagramas.

“Durante el análisis, analizamos los requisitos que se describen en la captura de requerimientos, refinándolos y estructurándolos. El objetivo de hacerlo es conseguir una comprensión más precisa de los requisitos y una descripción de los mismos que sea fácil de mantener y que nos ayude a estructurar el sistema entero, incluyendo su arquitectura”. (JACOBSON, I *et al.*, 1999)

3.1.1 Diagramas de Clases del Análisis.

El objetivo principal del flujo de trabajo del análisis son los diagramas de clases de análisis, los cuales muestran qué clases participan en las realizaciones de los distintos casos de usos.

En el diagrama de clases de análisis se representa los conceptos fundamentales que se abarcan en el dominio del problema. Estos diagramas representan las definiciones y relaciones entre las clases. Las clases del análisis se clasifican en Interfaz, de Control o Entidad.

- Clase Interfaz: Modela la interacción entre el sistema y sus actores.
- Clase Entidad: Modela información que posee una larga vida.
- Clase Control: Representa coordinación, secuencia, transacciones, y control de otros objetos y a menudo encapsula a un caso de uso en concreto.

A continuación se muestra uno de los diagramas de clases de análisis. Para consultar los diagramas de clases correspondientes a los casos de usos restantes remitirse al [Anexo 2](#).

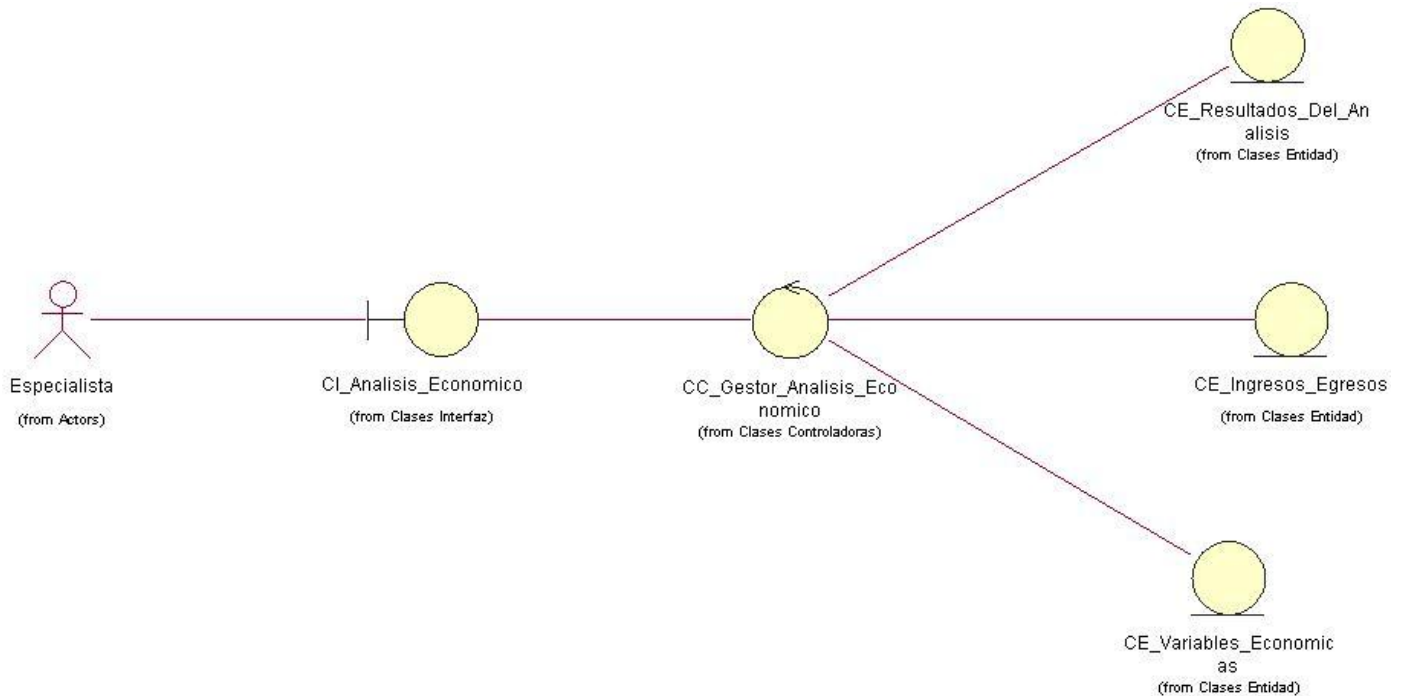


Figura 10: DCA_CU_Realizar_Análisis_Económico

3.1.2 Diagramas de Interacción.

Los diagramas de interacción se utilizan para modelar los aspectos dinámicos de un sistema, lo que conlleva modelar instancias concretas o prototípicas de clases interfaces, componentes y nodos, junto con los mensajes enviados entre ellos, todo en el contexto de un escenario que ilustra un comportamiento. En el contexto de las clases describen la forma en que grupos de objetos colaboran para proveer un comportamiento. (VILAS, 2001)

Se dividen en dos categorías los diagramas de colaboración y los diagramas de secuencia.

Un diagrama de secuencia es un diagrama de interacción que destaca la ordenación temporal de los mensajes. Muestran las interacciones expresadas en función de secuencias temporales.

Un diagrama de colaboración es un diagrama de interacción que destaca la organización estructural de los objetos que envían y reciben mensajes. Muestran las relaciones entre los objetos y los mensajes que intercambian.

Para consultar los diagramas de colaboración correspondientes a los casos de usos restantes remitirse al [Anexo 3](#).

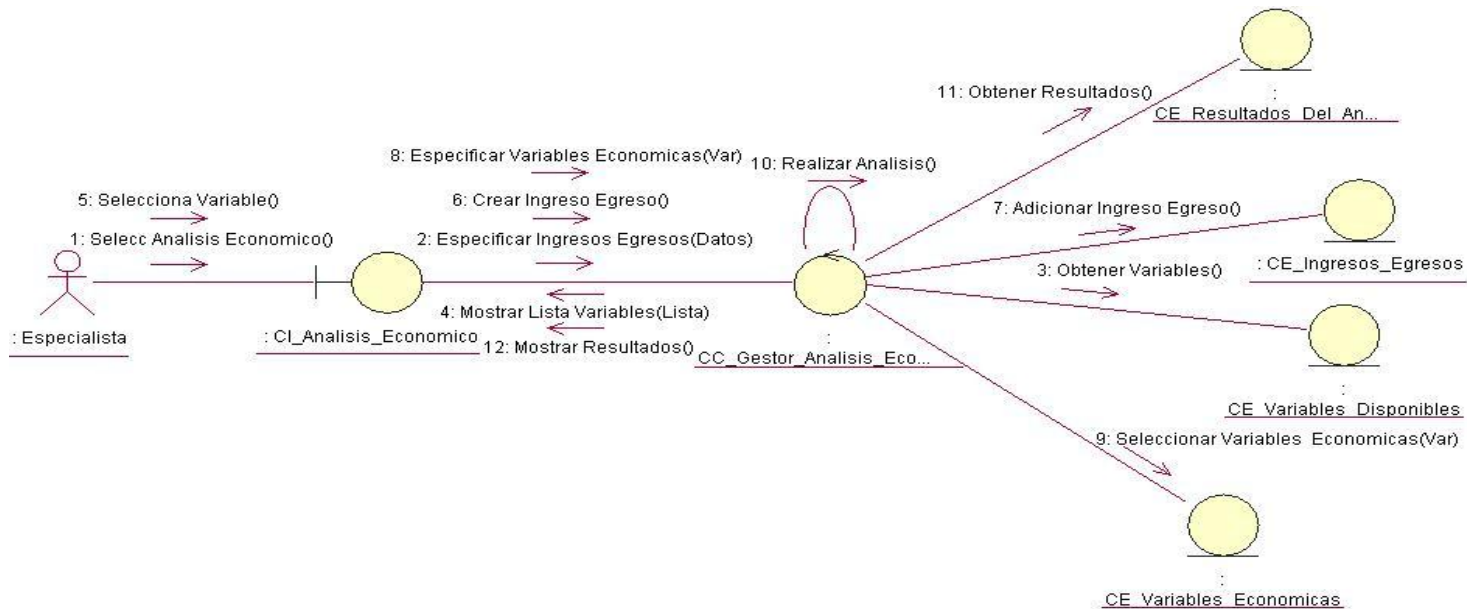


Figura 11: DC_CU_Realizar_Análisis_Económico

3.2 Modelo de Diseño

El diseño consiste en el refinamiento de los Modelos de Análisis para crear especificaciones adicionales que enriquecen el modelo de análisis con detalles próximos a la implementación. Una solución lógica, de forma que se cumplan los requerimientos (asignación de responsabilidades, interacciones entre objetos y otros.)(TORRES, 2002)

Sirve de abstracción de la implementación y es utilizada como entrada fundamental de las actividades de implementación. Este modelo puede contener: los diagramas, las clases, paquetes, subsistemas, cápsulas, protocolos, interfaces, relaciones, colaboraciones, atributos, las realizaciones de los casos de uso, entre otros que se puedan considerar para el sistema en desarrollo.

“El Modelo de Diseño es un modelo de objetos que describe la realización física de los casos de uso centrándose en cómo los requisitos funcionales y no funcionales, junto con otras restricciones relacionadas con el entorno de implementación, tienen impacto en el sistema a considerar. Además, el Modelo de Diseño sirve de abstracción de la implementación del sistema y es, de ese modo, utilizada como una entrada fundamental de las actividades de implementación”. (JACOBSON, I *et al.*, 1999)

3.2.1 Diagramas de Clases de diseño.

A través del flujo de diseño, uno de los artefactos más importantes a obtener son los diagramas de clases de diseño, donde se exponen las clases que intervienen en las realizaciones de los casos de uso del sistema. En este tipo de diagrama se representa un nivel de detalle más alto que los diagramas de clases del análisis, relacionándose con el lenguaje de programación del cual se hará uso en la implementación del sistema.

El diagrama de clases de diseño describe gráficamente las especificaciones de las clases de software y de las interfaces en una aplicación. Normalmente contiene la siguiente información:(VISCONTI y ASTUDILLO, 2003)

- Clases, asociaciones y atributos
- Interfaces, con sus operaciones y constantes
- Métodos

- Información sobre los tipos de los atributos, navegabilidad y dependencias

A continuación se muestra un ejemplo de los diagramas de clases, correspondiente al caso de uso Realizar Análisis Económico. Para consultar los restantes diagramas consultar el [anexo 4](#).

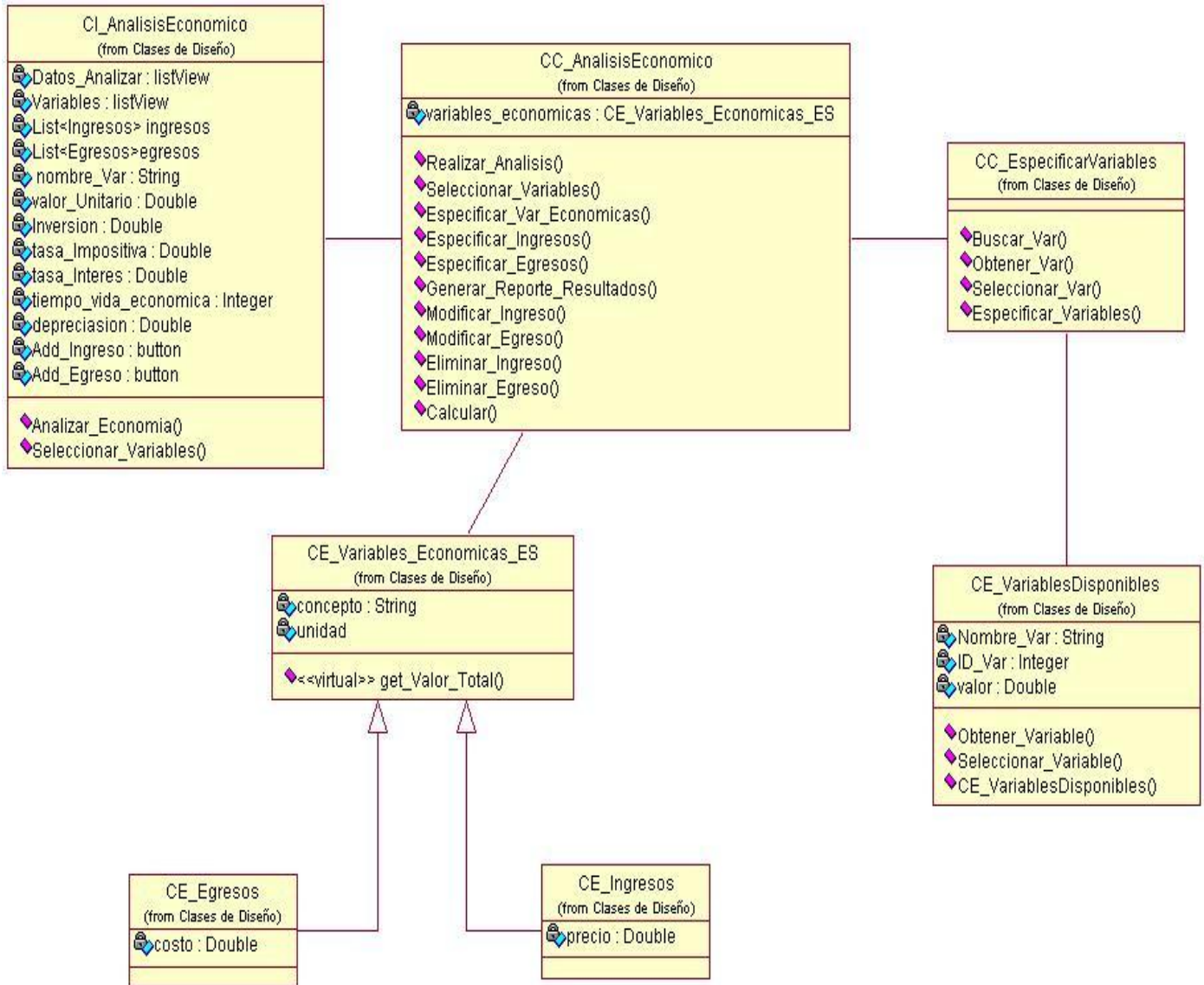


Figura 12: DCD_CU_Realizar Análisis Económico

3.2.2 Diagramas de Interacción del diseño

En este epígrafe se muestran los diagramas de interacción del diseño poniéndose como ejemplo uno de ellos. Para consultar los diagramas restantes dirigirse al [anexo 5](#).

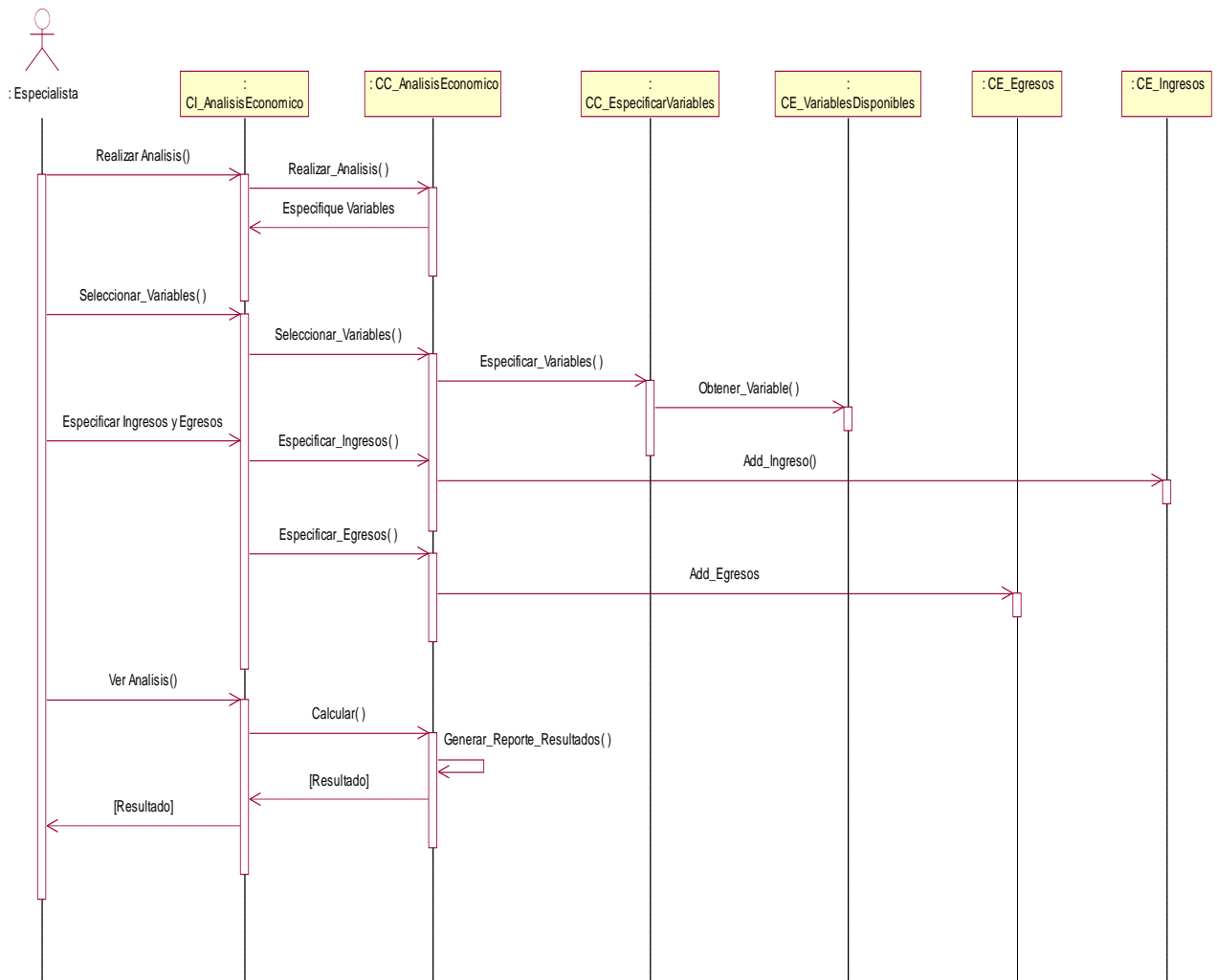


Figura 13: DS_CU_Realizar Análisis Económico

3.3 Patrones de Diseño

Un patrón de diseño es una solución a un problema de diseño no trivial que es efectiva (ya se resolvió el problema satisfactoriamente en ocasiones anteriores) y reusable (se puede aplicar a diferentes problemas de diseño en distintas circunstancias). Los patrones son soluciones de sentido común que deberían formar parte del conocimiento de un diseñador experto. Además facilitan la comunicación entre diseñadores, pues establecen un marco de referencia (terminología, justificación). (TORRES, 2002)

Por otro lado, los patrones de diseño, facilitan el aprendizaje al programador inexperto, pudiendo establecer parejas problema-solución. En la programación orientada a objetos resulta complicado descomponer el sistema en objetos (encapsulación, granularidad, dependencias, flexibilidad, reusabilidad y otros), estos patrones permitirán identificar a los objetos apropiados de una manera mucho más sencilla. Además, los patrones de diseño, también ayudarán a especificar las interfaces, identificando los elementos claves en las interfaces y las relaciones existentes entre distintas interfaces. De igual modo facilitarán la especificación de las implementaciones. También, y de forma casi automática, ayudan a reutilizar código, facilitando la decisión entre "herencia o composición" (favorece la composición sobre la herencia y hace uso de la delegación). Relacionan estructuras en tiempo de compilación y en tiempo de ejecución y permiten hacer un diseño preparado para el cambio. (TORRES, 2002)

Los patrones se pueden clasificar según su propósito:(TORRES, 2002)

- **Patrones de creación:** para creación de instancias.
- **Patrones estructurales:** relaciones entre clases, combinación y formación de estructuras mayores.
- **Patrones de comportamiento:** interacción y cooperación entre clases.

Patrones de creación

Los patrones de creación abstraen la forma en la que se crean los objetos, permitiendo tratar las clases a crear de forma genérica dejando para más tarde la decisión de qué clases crear o cómo crearlas. Según donde se tome dicha decisión se pueden clasificar a los patrones de creación en *patrones de creación de clase* (la decisión se toma en los constructores de las clases y usan la herencia para

determinar la creación de las instancias) y *patrones de creación de objeto* (se modifica la clase desde el objeto).

Entre los principales que se relacionan con esta clasificación se pueden mencionar los siguientes:

- Patrón Factoría
- Patrón Factoría Abstracta
- Patrón Singleton (Instancia Única)
- Patrón Prototipo

Patrones estructurales

Tratan de conseguir que cambios en los requisitos de la aplicación no ocasionen cambios en las relaciones entre los objetos. Lo fundamental son las relaciones de uso entre los objetos, y, estas están determinadas por las interfaces que soportan los objetos. Estudian cómo se relacionan los objetos en tiempo de ejecución. Sirven para diseñar las interconexiones entre los objetos.

De estos se pueden hacer referencia a los siguientes:

- Patrón Adaptador
- Patrón Puente
- Patrón Composición
- Patrón Decorador
- Patrón Fachada
- Patrón Proxy

Patrones de comportamiento

Los patrones de comportamiento estudian las relaciones entre llamadas entre los diferentes objetos, normalmente ligados con la dimensión temporal.

Dentro de ellos se encuentran los mencionados a continuación:

- Patrón Cadena de Responsabilidad
- Patrón Comando
- Patrón Intérprete
- Patrón Iterador
- Patrón Mediador
- Patrón Recuerdo (Memento)
- Patrón Observador
- Patrón Estado
- Patrón Estrategia
- Patrón Plantilla
- Patrón Visitante

Patrones GRASP

Los patrones GRASP describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones. Constituyen un apoyo para la enseñanza que ayuda a entender el diseño de objeto esencial y aplica el razonamiento para el diseño de una forma sistemática, racional y explicable. GRASP es un acrónimo que significa General Responsibility Assignment Software Patterns (patrones generales de software para asignar responsabilidades).

Dentro de esta clasificación de patrones se encuentran entre otros:

- Bajo Acoplamiento
- Alta Cohesión
- Experto
- Creador
- Controlador

3.3.1 Patrones a utilizar

Después de haber analizado de forma general los patrones de diseño vistos anteriormente se decidió utilizar para el desarrollo del trabajo los patrones GRASP. Estos son usados en el desarrollo de los

diagramas de clases del diseño proporcionando que estos sean más reusables, flexibles y tengan una mayor robustez.

El patrón experto se evidencia en los diagramas sobre todo en las clases controladoras de los distintos análisis puesto que estas poseen la mayor información necesaria para resolver las funcionalidades requeridas. También se evidencia el bajo acoplamiento una vez que las relaciones entre las clases que participan en el desarrollo de los distintos casos de usos son mínimas asegurando así que exista entre ellas una alta cohesión. Además se utiliza el patrón creador con el objetivo de asignar la responsabilidad de crear instancias de otras clases a la clase adecuada para este fin. Como un ejemplo de donde se evidencia el uso de este patrón se puede mencionar el diagrama de clases del caso de uso Realizar Análisis Económico en el cual la clase controladora Especificar Variables será la encargada de crear los objetos de de la clase entidad Variables Económicas.

De manera general a continuación se les brinda una breve descripción de lo que realiza cada uno de estos patrones y para qué son utilizados fundamentalmente.

- **Experto:** Asignar una responsabilidad al experto en información: La responsabilidad de realizar una labor es de la clase que tiene o puede tener los datos involucrados (atributos). Una clase, contiene toda la información necesaria para realizar la labor que tiene encomendada. Este patrón se usa más que cualquier otro al asignar responsabilidades; es un principio básico que suele utilizarse en el diseño orientado a objetos; expresa simplemente la "intuición" de que los objetos hacen cosas relacionadas con la información que poseen.(LARMAN, 1999)
- **Creador:** Asignar a la clase B la responsabilidad de crear una instancia de clase A en alguno de los siguientes casos (B agrega los objetos de A; B contiene a los objetos de A; B registra las instancias de los objetos de A; B utiliza específicamente los objetos de A; B tiene los datos de inicialización que serán transmitidos a A cuando este objeto sea creado). El propósito fundamental de este patrón es encontrar un creador que debemos conectar con el objeto producido en cualquier evento. Al escogerlo como creador, se da soporte al bajo acoplamiento.(LARMAN, 1999)

- **Controlador:** Asignar la responsabilidad de controlar el flujo de eventos del sistema, a clases específicas. Esto facilita la centralización de actividades (validaciones, seguridad y otros). La mayor parte de los sistemas reciben eventos de entrada externa, por lo cual, si se recurre a un diseño orientado a objetos, hay que elegir los controladores que manejen esos eventos de entrada.(LARMAN, 1999)
- **Alta Cohesión:** Asignar una responsabilidad de modo que la cohesión siga siendo alta. Una clase tiene responsabilidades moderadas en un área funcional y colabora con las otras para llevar a cabo las tareas. Una clase con mucha cohesión es útil porque es bastante fácil darle mantenimiento, entenderla y reutilizarla. Su alto grado de funcionalidad, combinada con una reducida cantidad de operaciones, también simplifica el mantenimiento y los mejoramientos. La ventaja que significa una gran funcionalidad también soporta un aumento de la capacidad de reutilización.(LARMAN, 1999)
- **Bajo Acoplamiento:** Asignar una responsabilidad para mantener bajo acoplamiento. Debe haber pocas dependencias entre las clases. Si todas las clases dependen de todas ¿cuánto software podemos extraer de un modo independiente y reutilizable en otro proyecto? .Para determinar el nivel de acoplamiento de clases, son muy buenos los diagramas de colaboración de UML. Uno de los principales síntomas de un mal diseño y alto acoplamiento es una herencia muy profunda. Siempre hay que considerar las ventajas de la delegación respecto de la herencia. Además soporta el diseño de clases más independientes, que reducen el impacto de los cambios, y también más reutilizables, que acrecienta la oportunidad de una mayor productividad.(LARMAN, 1999)

3.4 Conclusiones

El análisis y diseño de un software brinda la primera visión de lo que pudiera ser la solución en el desarrollo del mismo. En este capítulo se ha tratado de mostrar una idea de las posibles funcionalidades que podría tener el subsistema de Análisis de Resultados a través de una serie de modelos que permiten tener un mayor entendimiento de este. Se definieron las principales clases del análisis y el diseño mostrando sus relaciones así como la colaboración entre ellas mediante diferentes diagramas. Para lograr un mejor diseño se utilizaron algunos patrones que ayudan a la optimización del mismo.

CONCLUSIONES GENERALES

Una vez finalizada la investigación y realizado todas las tareas propuestas, se puede afirmar que se cumplió el objetivo planteado pues:

- ❖ Se realizó un estudio de los principales simuladores industriales existentes con el objetivo de encontrar las herramientas que poseen para el Análisis de Resultados.
- ❖ Se realizó una comparación de diferentes metodologías de desarrollo de software existentes escogiéndose RUP para el desarrollo del presente trabajo.
- ❖ Después de un análisis minucioso y crítico de herramientas CASE quedó definida para el trabajo la herramienta Rational Rose Enterprise Edition, la cual usa el lenguaje de modelado UML para la construcción de diagramas y modelos y se vincula muy bien con la metodología seleccionada.
- ❖ Se realizó un modelo del dominio logrando combinar los principales conceptos existentes en la investigación.
- ❖ Se obtuvieron todas las funcionalidades que debía cumplir el subsistema mediante los requisitos funcionales y no funcionales del mismo.
- ❖ Se especificaron los casos de uso del sistema y se utilizaron patrones para lograr un mejor entendimiento de los mismos.
- ❖ Se realizó el análisis del subsistema de Análisis de Resultados para el simulador de procesos químicos.
- ❖ Se realizó el diseño del subsistema de Análisis de Resultados para el simulador de procesos químicos.
- ❖ Se implementó un prototipo funcional del Análisis Económico perteneciente al subsistema de Análisis de Resultados.
- ❖ Se utilizaron patrones de diseño para optimizar la flexibilidad del diseño realizado.
- ❖ Se generaron todos los artefactos necesarios para completar la información necesaria sobre todas las actividades realizadas y lograr una mejor comprensión para la futura implementación del subsistema.

RECOMENDACIONES

- ❖ A partir de los artefactos generados seguir con la implementación de las funcionalidades propuestas en las próximas fases del ciclo de desarrollo.
- ❖ Integrar el subsistema a los demás subsistemas que componen el simulador de procesos químicos.
- ❖ Incorporar en próximas iteraciones otros métodos de Análisis de Resultados.

BIBLIOGRAFÍA

CREUS, A. Simulación y Control de Procesos por Ordenador. Barcelona: MARCOMBO S.A,

GOULD, H. y TOBOCHNIK, J. An Introduction to Computer Simulation Methods. Segunda ed. ADDISON-WESLEY, USA., 1996.

HUSAIN, A. Chemical Process Simulation. JOHN WILEY & SONS, New York, 1986.

RODRÍGUEZ, N. H. Informática Aplicada a la Ingeniería de Procesos. Ingeniería Química. Universidad Tecnológica Nacional, Facultad Regional Rosario, 2001.

SCENNA., N. J. Modelado, Simulación y Optimización de Procesos Químicos. 1999. ISBN 950-42-0022-2

TARIFA, E. Simulación de Sistemas Químicos. 2002

TARIFA, E. E.; HUMANA, D., et al. Simulación Dinámica De Tiempo Real. Universidad Nacional de Jujuy, 2003.

ALONSO, Y. y GONZÁLEZ, Y. Software para la Simulación de Sistemas Biológicos: Módulo de Simulación y Análisis. . Tutor: Moreno, N. Universidad de las Ciencias Informáticas, 2007.

CASAS, J. Aplicación de modelos de especiación iónica para la generación de sensores virtuales de apoyo al análisis y control de soluciones ácidas con metales disueltos procedentes de procesos industriales de lixiviación de minerales de cobre. 2001

DOMENECH, F. Simulación y optimización de procesos para un desarrollo industrial sostenible. Aplicación en la industria de producción de alcohol. ICIDCA. 2008

ERDMANN, E. y HUMANA, D. Gas transport analysis. 2007, vol. 27.

GUERRERO, L. A. UML- Diagramas de Interacción. Universidad de Chile. 2006

ASPEN TECHNOLOGY INC. Sensitivity Analysis. Introduction to Aspen Plus. 2003

LAGOS, M. Introducción al diseño de patrones. 2008.

MARTIN, R. C. Design Principles and Design Patterns. 2008.

MAYORCA, M. J. Simulación y estudio del reactor en el proceso de hidrogenación de aceite de girasol en condiciones supercríticas. Journal of Supercritical Fluids, 2005.

MORENO, G. Análisis y Diseño de sistemas. 2007.

ROJAS, G. UML. Diagramas de Secuencia. 2007.

SÁNCHEZ, H. y LLAP, F. Evaluación del software para la simulación de procesos químicos. Unidad Telecomunicaciones e Informática-Petroperu. 2006

SANTOS, C. Análisis y Diseño de sistemas informáticos. 2008, Disponible en:
<http://apuntes.rincondelvago.com/analisis-y-diseno-de-sistemas-informaticos.html>.

VALLE, M. D. y FUENTES, J. D. Simulación del sistema de absorción, enfriamiento de ácido y generación de vapor de una planta de ácido sulfúrico. Univesidad de Carabobo. 2006.

ZAVALAR, J. Ingeniería de Software. 2000.

REREFENCIAS BIBLIOGRÁFICAS

ALEJO, P. D. Manual de Usuario del Sistema de Programas TERMOAZUCAR. ISPJAE-ISTC. 1989

ARGOTA, J. V. Técnicas de simulación en la enseñanza de la Psiquiatría. 1990

BANKS, J.; CARSON, J. S., et al. Discrete-Event System Simulation Second Edition. ed. New Jersey: Prentice-Hall, 1996.

BOGOYA, D. SIMULACIÓN DE PROCESOS QUÍMICOS EN ESTADO ESTABLE. 1990

FISHMAN, G. S. Conceptos y métodos en la simulación digital de eventos discretos. Limusa, México: 1978.

GALLEGO, J. P. G. FUNDAMENTOS DE LA METODOLOGIA RUP (RATIONAL UNIFIED PROCESS). 2007

GARRIDO, N. SIDEL: Sistema de diseño y simulación de destilerías. 3er Taller Internacional de Análisis de Procesos (TIAP). 2002

GOZÁ, O. Use of simulation and expert systems to increase the energy efficiency in cane sugar factories. . Revista Developments in Chemical Engineering & Mineral Processing, 2002, nº Practical Applications of Process Systems Engineering,

GOZÁ, O. y PÉREZ, O. SIMULACION MODULAR SECUENCIAL DE DESTILERIAS DE ALCOHOL CON FINES ENERGETICOS Facultad Ingeniería Química. 2003

HENAO, C. y VÉLEZ, J. Manual del laboratorio diseño de procesos químicos - Uso del paquete de simulación HYSYS.Process. UPB.Medellín. 2002

HUGO, N. Modelado y Optimización. Departamento de Ingeniería Química.Grupo de Investigación Aplicada a la Ingeniería Química (GIAIQ). Universidad Tecnológica Nacional, Facultad Regional Rosario, 2001.

IZADA, D. y VEGA, E. C. Simulación de Procesos Tecnológicos Centro Azúcar, ISBN 0253-5777.

JACOBSON, I.; BOOCH, G., et al. El Lenguaje Unificado de Modelado. Manual de Referencia. Addison Wesley, 2000a.

---. El Proceso Unificado de Desarrollo de Software. 1999. vol. volumen 1,

---. El Proceso Unificado de Desarrollo de Software. Madrid: Addison Wesley, 2000b. ISBN 84-7829-036-2.

---. El Proceso Unificado de Desarrollo de Software, 2004. vol. 1,

LARMAN, C. UML Y Patrones. Introducción al análisis y diseño orientado a objeto. México 1999.

LINNINGER, A. A.; CHOWDHRY, S., et al. A system approach to mathematical modeling of industrial process, Computers and Chemical Engineering. 2000. 591-598 p.

MARJEDAN, A. Alerta - Antivirus. La seguridad de la información [Consultado el: 9 de mayo del 2009 Disponible en: http://www.alertaantivirus.es/seguridad/ver_pag.html?tema=S&articulo=4&pagina=1.

MARTÍNEZ, V. H.; ALONSO, P. A., et al. Simulación de procesos en Ingeniería Química. Primera Edición ed. México: 2000. 187 p. ISBN 968-856-755-8.

MORÓN, C. Una Herramienta para la Simulación de la Unidad de Destilación Atmosférica de la Refinería Níco López. Tesis de Doctorado, ISPJAE, 2000.

NAYLOR, T. H. Técnicas de simulación en computadoras. 1971

PACHECO, H. S. Evaluación del software para la simulación de procesos químicos 2006

PATÓN, D. Metodologías de Desarrollo de Software Disponible en: <http://alarcos.inf-cr.uclm.es/doc/ISOFTWAREI/Tema04.pdf>

PEREDA, E. S. Utilización de un Lenguaje de Simulación para la Determinación de un Esquema Tecnológico Eficiente en la Purificación del Sulfato Ferroso. 1993, vol. 14, ISBN 5-59, 0253-9276.

PÉREZ, H.; FERNÁNDEZ, J. P., et al. Sistema de Simulación de Centrales Azucareros TERMOAZUCAR. Manual de Usuario. Departamento de Información Científico Técnica. ISTC. 1989

PRESSMAN, R. Ingeniería de Software "Un enfoque práctico" 5ta ed. Madrid: Graw Hill, 2002.

RODRÍGUEZ, M. Modelado y simulación en Ingeniería Química [Consultado el: 12 de febrero del 2009 Disponible en: http://www.diquima.upm.es/docencia/modysim2000/docs/tema3_p4.pdf.

SABADÍ, R. SIMFAD un sistema para la simulación de fábricas de azúcar y derivados. Cuba Azúcar. 1991

SANCHEZ, M. A. M. Metodologías De Desarrollo De Software [Consultado el: 17 de febrero del 2009 Disponible en: <http://www.willydev.net/descargas/cualmetodologia.pdf>.

SCHWABER, K. Agile Project Management with Scrum. Microsoft Press, 2004.

SHANNON, R. E. Simulación de Sistemas. Diseño, desarrollo e implementación. Trillas, México: 1988.

SIERRA, M. Trabajando con Visual Paradigm for UML. Universidad de Cantabria - Facultad Ciencias.

[Consultado el: 17 de febrero del 2009 Disponible en: <http://www.visual-paradigm.com/>.

SUÁREZ, R. ACOPLA: Programa de computación para el balance de materiales, energía y costo de producción de una destilería. Departamento de Ingeniería. ICIDCA. 1973

TARIFA, E. E. Teoría de Modelos y Simulación. 2002,

TORRES, M. L. Introducción al diseño con patrones [Consultado el: 8 de abril del 2009 Disponible en: <http://www.elrincondelprogramador.com/default.asp?id=29&pag=articulos/leer.asp>.

VEGA, E. D. L. y SABADÍ, R. Modelación matemática del esquema de producción azúcar - alcohol. 1988,

VERA, A. M. Simulación de Procesos [Consultado el: 16 de febrero del 2009 Disponible en: <http://simulacioniq.blogspot.com/2008/05/chemcad-chemstations.html>.

VILAS, A. F. Diagramas de Interacción [Consultado el: 9 de abril del 2009 Disponible en: <http://tvdi.det.uvigo.es/~avilas/UML/node41.html>.

VISCONTI, M. y ASTUDILLO, H. Fundamentos de Ingeniería de Software. Departamento de Informática, 2003

WEBB, D. y RAMIREZ, W. F. Modeling, simulation and optimization software framework for dynamic systems. En American Control Conference. Denver, Colorado-USA. 2003.

ANEXOS

Anexo 1: Descripciones de Casos de Usos

Descripción textual del caso de uso “Analizar_Dependencia”

Caso de Uso:	Analizar_Dependencia
Actores:	Especialista
Resumen:	El CU comienza cuando el Especialista desea realizar el Análisis de Dependencia a los resultados de la simulación. Esta funcionalidad permitirá hacer Análisis de Dependencia sobre los resultados de la simulación.
Precondiciones:	El sistema debe haber simulado
Referencias	<p>R4 Realizar Análisis de Dependencia.</p> <p>R4.1 Especificar variable independiente del Análisis de Dependencia</p> <p>R4.2 Especificar variable objetivo del Análisis de Dependencia.</p> <p>R4.3 Especificar valores de la variable independiente.</p> <p>R4.3.1 Analizar mediante un intervalo y un paso.</p> <p>R4.3.2 Analizar indicando los valores.</p> <p>R5 Mostrar gráfico con resultados del Análisis de Dependencia.</p>
CU Asociados	Especificar_Variables (<i>Include</i>)
Prioridad	--

Flujo Normal de Eventos

Acción del Actor	Respuesta del Sistema
1. El especialista escoge la opción análisis de dependencia en el menú de la barra de herramientas.	<p>1.1 El sistema inicializa el caso de uso “Especificar_Variables”.</p> <p>1.2 El sistema solicita especificar el valor de las variables independientes y le brinda dos</p>

<p>2. Selecciona el valor de la variable independiente deseado.</p> <p>3. Elige la opción calcular.</p>	<p>opciones:</p> <ul style="list-style-type: none"> • Mediante un intervalo y un paso. Ver sección “Mediante intervalo con paso” • Indicando los valores. Ver sección “Indicar valores” <p>2.1 Si escoge la opción de especificar el valor mediante un intervalo y un paso ir a la sección “Mediante intervalo con paso”, si escoge la opción de indicar valores ir a la sección “Indicar valores”.</p> <p>3.1 Muestra los resultados en un gráfico.</p>
---------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Prototipo de Interfaz

Sección “Mediante intervalo con paso”

Flujo Normal de Eventos

Acción del Actor	Respuesta del Sistema
<p>2 Escoge la opción de especificar los valores mediante intervalos con paso.</p>	<p>2.1 Le solicita al usuario el límite inferior y el límite superior de los valores que va a tomar la variable y el paso.</p>
<p>2.2 Inserta los valores solicitados.</p>	<p>2.3 Calcula los valores que va a tomar la variable y los guarda.</p>

Sección “Indicar_Valores”

Flujo Normal de Eventos

Acción del Actor	Respuesta del Sistema
-------------------------	------------------------------

4. Escoge la opción de indicar valores.	4.1 Le solicita al usuario los diferentes valores que va a tomar la variable independiente.
4.2 Inserta los valores solicitados.	4.3 Guarda los valores.
Prototipo de Interfaz	
Poscondiciones	

Tabla 3: Descripción del CU Analizar_Dependencia

Descripción textual del caso de uso “Analizar_Indicadores”

Caso de Uso:	Analizar_Indicadores
Actores:	Especialista
Resumen:	El CU comienza cuando el Especialista desea realizar el Análisis de Indicadores a los resultados de la simulación.
Precondiciones:	El sistema debe haber simulado
Referencias	R6 Realizar Análisis de Indicadores R6.1 Definir indicador R6.2 Seleccionar indicadores a analizar R6.3 Especificar los indicadores R7 Mostrar reporte del Análisis de Indicadores.
CU Asociados	Definir_Indicador (<i>Extend</i>) Generar_Reporte_Resultados (<i>Extend</i>)
Prioridad	--
Flujo Normal de Eventos	

Acción del Actor	Respuesta del Sistema
1 El especialista escoge la opción análisis de indicadores en el menú de la barra de herramientas.	1.1 El sistema muestra un formulario solicitando los indicadores, ofreciendo la posibilidad de seleccionarlos directamente o definir otros de interés para el usuario. Si desea definir nuevos indicadores ver sección "Definir_Indicador".
2 Selecciona o define los indicadores.	2.1 Solicita los datos necesarios para cada indicador.
3. Inserta los datos solicitados.	3.1 Calcula los indicadores y brinda la opción de generar un reporte con los resultados.
4. Escoge la opción de generar un reporte con los resultados.	4.1 El sistema inicializa el caso de uso "Generar_Reporte_Resultados".

Prototipo de Interfaz

Sección "Definir_Indicador"

Flujo Normal de Eventos

Acción del Actor	Respuesta del Sistema
2 Escoge la opción de definir los indicadores. 2.2 Selecciona los indicadores deseados.	2.1 Muestra un formulario para que el usuario seleccione los nuevos indicadores. 2.3 Guarda los indicadores definidos. 2.4 Muestra una lista con los indicadores definidos y los que estaban predeterminados.

Prototipo de Interfaz	
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
4. No escoge esa opción.	4.1 El sistema termina el caso de uso.
Poscondiciones	

Tabla 4: Descripción del CU Analizar_Indicadores

Descripción textual del caso de uso “Generar_Reporte_Resultados”

Caso de Uso:	Generar_Reporte_Resultados
Actores:	Especialista
Resumen:	El CU comienza cuando el Especialista desea generar el reporte del resultado de alguno de los análisis realizados.
Precondiciones:	Haber realizado un análisis sobre los resultados.
Referencias	R3 Generar reporte con resultados del Análisis de Sensibilidad. R7 Mostrar reporte del Análisis de Indicadores. R9 Reportar resultados del Análisis Económico.
CU Asociados	---
Prioridad	--

Flujo Normal de Eventos

Acción del Actor	Respuesta del Sistema
-------------------------	------------------------------

1. El especialista solicita generar un reporte.	1.1 El sistema muestra un documento con los resultados del análisis realizado.
	1.2 El sistema brinda la posibilidad de guardar los resultados en un documento.
2. Elige la opción deseada	2.1 Si escoge la opción guardar el sistema salva los resultados en un documento especificado.
Prototipo de Interfaz	
Poscondiciones	Se genera un documento con los resultados del análisis realizado.

Tabla 5: Descripción del CU Generar_Reporte_Resultados

Descripción textual del caso de uso “Especificar_Variables”

Caso de Uso:	Especificar_Variables
Actores:	Especialista
Resumen:	El CU es iniciado una vez que el especialista desea realizar el Análisis de Sensibilidad o el Análisis de Dependencia sobre los resultados de la simulación.
Precondiciones:	Se debe proceder a realizar un Análisis de Resultados.
Referencias	R1.1 Especificar variables independientes del Análisis de Sensibilidad R1.2 Especificar variable objetivo del Análisis de Sensibilidad. R4.1 Especificar variable independiente del Análisis de Dependencia R4.2 Especificar variable objetivo del Análisis de Dependencia.
CU Asociados	---
Prioridad	--

Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. Selecciona variables (variable objetivo y variables independientes)	1.1 Muestra una lista de variables para que el usuario seleccione las variables a analizar.
2. Especifica nuevas variables	2.1 El sistema guarda las variables escogidas.
Prototipo de Interfaz	
Poscondiciones	

Tabla 6: Descripción del CU Especificar_Variables

Descripción textual del caso de uso “Definir_Ingresos_Egresos”

Caso de Uso:	Definir_Ingresos_Egresos
Actores:	Especialista
Resumen:	El CU comienza cuando el Especialista desea definir nuevos Ingresos y Egresos para realizar el Análisis Económico.
Precondiciones:	Se debe proceder a realizar un Análisis Económico.
Referencias	R8.1 Definir ingresos R8.2 Definir egresos
CU Asociados	---
Prioridad	--
Flujo Normal de Eventos	

Acción del Actor	Respuesta del Sistema
1 Selecciona definir los ingresos y los egresos.	1.1 Muestra un formulario para que el usuario introduzca los datos deseados.
2 Especifica los nuevos ingresos y egresos.	2.1 Guarda la nueva configuración con los datos introducidos.
Prototipo de Interfaz	
Poscondiciones	

Tabla 7: Descripción del CU Definir_Ingresos_Egresos

Descripción textual del caso de uso “Mostrar_Gráfico_Resultados”

Caso de Uso:	Mostrar_Gráfico_Resultados
Actores:	Especialista
Resumen:	El CU comienza cuando el Especialista especificar los ingresos y los egresos necesarios para realizar el Análisis Económico.
Precondiciones:	Se debe haber realizado un Análisis de Resultados de Sensibilidad o de Dependencia.
Referencias	R2 Mostrar gráfico con resultados del Análisis de Sensibilidad. R5 Mostrar gráfico con resultados del Análisis de Dependencia.
CU Asociados	---
Prioridad	--
Flujo Normal de Eventos	

Acción del Actor	Respuesta del Sistema
1. Selecciona la opción de mostrar resultados en forma de gráfico.	1.1 El sistema muestra los resultados en un gráfico.
Prototipo de Interfaz	
Poscondiciones	

Tabla 8: Descripción del CU Mostrar_Gráfico_Resultados

Descripción textual del caso de uso “Especificar_Ingresos_Egresos”

Caso de Uso:	Especificar_Ingresos_Egresos
Actores:	Especialista
Resumen:	El CU comienza cuando el Especialista desea mostrar un gráfico con el resultado de alguno de los análisis realizados.
Precondiciones:	Realizar el Análisis Económico.
Referencias	R8.1 Definir ingresos R8.2 Definir egresos
CU Asociados	Definir_Ingresos_Egresos (<i>Extend</i>)
Prioridad	--
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema

1. El especialista desea especificar los ingresos y los egresos para el análisis.	1.1 El sistema muestra un formulario para que el usuario especifique los ingresos y los egresos deseados y brinda la posibilidad de definir otros de interés para el usuario. En caso de escoger esta opción el sistema inicializa el caso de uso “Definir_Ingresos_Egresos”. (Ver descripción del caso de uso).
2. Selecciona o define los datos de los ingresos y de los egresos.	2.1 El sistema solicita los datos de los precios de los ingresos y los costos de los productos egresados.
3. Introduce los datos solicitados.	3.1 El sistema guarda la configuración.
Prototipo de Interfaz	
Poscondiciones	

Tabla 9: Descripción del CU Especificar_Ingresos_Egresos

Descripción textual del caso de uso “Analizar_Sensibilidad”

Caso de Uso:	Analizar_Sensibilidad
Actores:	Especialista
Resumen:	El CU comienza cuando el Especialista desea realizar el Análisis de Sensibilidad a los resultados de la simulación. El sistema brinda la opción adecuada al Especialista. Este efectúa las operaciones pertinentes y el sistema muestra los resultados terminando el CU.
Precondiciones:	El sistema debe haber simulado
Referencias	R1 Realizar Análisis de ensibilidad

CU Asociados	R1.1 Especificar variables independientes del Análisis de Sensibilidad
	R1.2 Especificar variable objetivo del Análisis de Sensibilidad.
	R1.3 Especificar valor nuevo de variable independiente.
Prioridad	R2 Mostrar gráfico con resultados del Análisis de Sensibilidad.
	R3 Generar reporte con resultados del Análisis de Sensibilidad.
	Mostrar_Gráfico_Resultados (<i>Extend</i>)
	Generar_Reporte_Resultados (<i>Extend</i>)
	Especificar_Variables (<i>Include</i>)
	--

Flujo Normal de Eventos

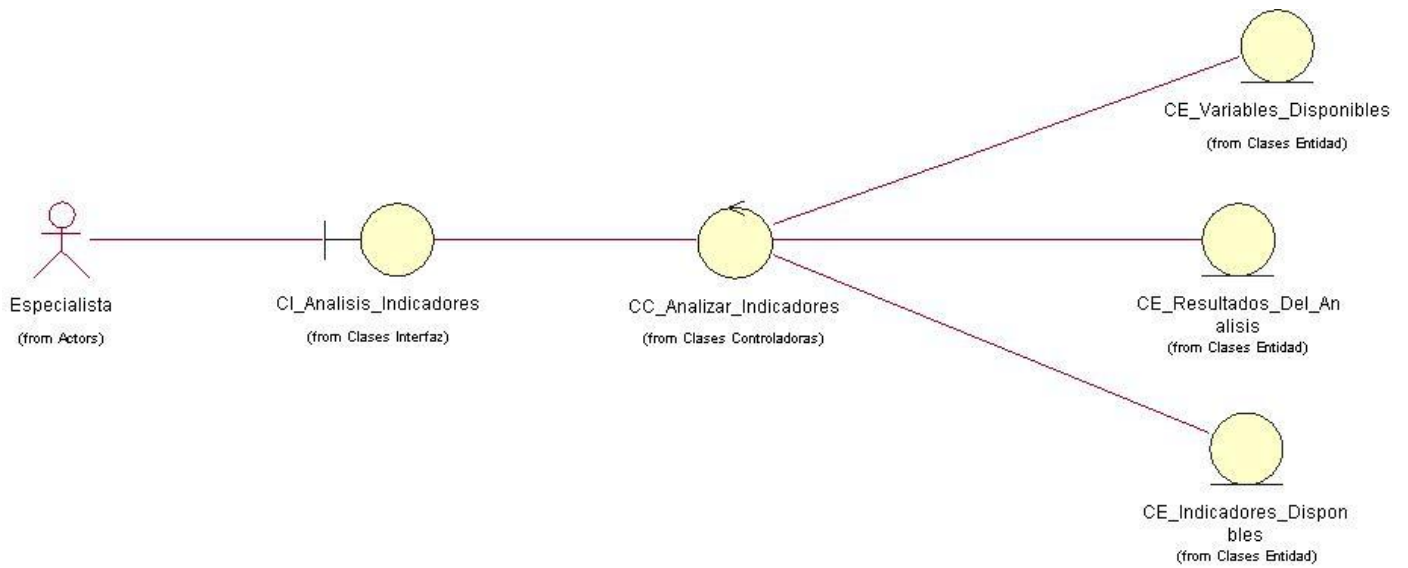
Acción del Actor	Respuesta del Sistema
<ol style="list-style-type: none"> 1. El especialista escoge la opción Análisis de Sensibilidad en el menú de la barra de herramientas. 2. Elige la opción de calcular para obtener los resultados. 	<ol style="list-style-type: none"> 1.1 El sistema inicializa el caso de uso "Especificar_Variables". 2.1 El sistema calcula las sensibilidades para cada una de las variables independientes y muestra una ventana con los resultados obtenidos brindando dos opciones: <ul style="list-style-type: none"> • Mostrar los resultados a través de un gráfico. Ver sección "Mostrar_gráfico_resultados" • Generar un reporte con ellos. Ver sección "Generar_Reporte_Resultados"

3. Selecciona la opción deseada para mostrar los resultados.	3.1 Si escoge la opción de Mostrar los resultados a través de un gráfico ir a la sección “Mostrar_gráfico_resultados”, si escoge la opción de generar un reporte ir a la sección “Generar_Reporte_Resultados”.
Prototipo de Interfaz	
Sección “Mostrar_gráfico_resultados”	
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
2.5 Selecciona la opción de mostrar resultados en forma de gráfico.	3.1 El sistema ejecuta el caso de uso “Mostrar_gráfico_resultados”
Sección “Generar_Reporte_Resultados”	
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
3 Selecciona la opción de Generar reporte de resultados.	3.1 El sistema ejecuta el caso de uso “Generar_Reporte_Resultados”
Prototipo de Interfaz	
Poscondiciones	

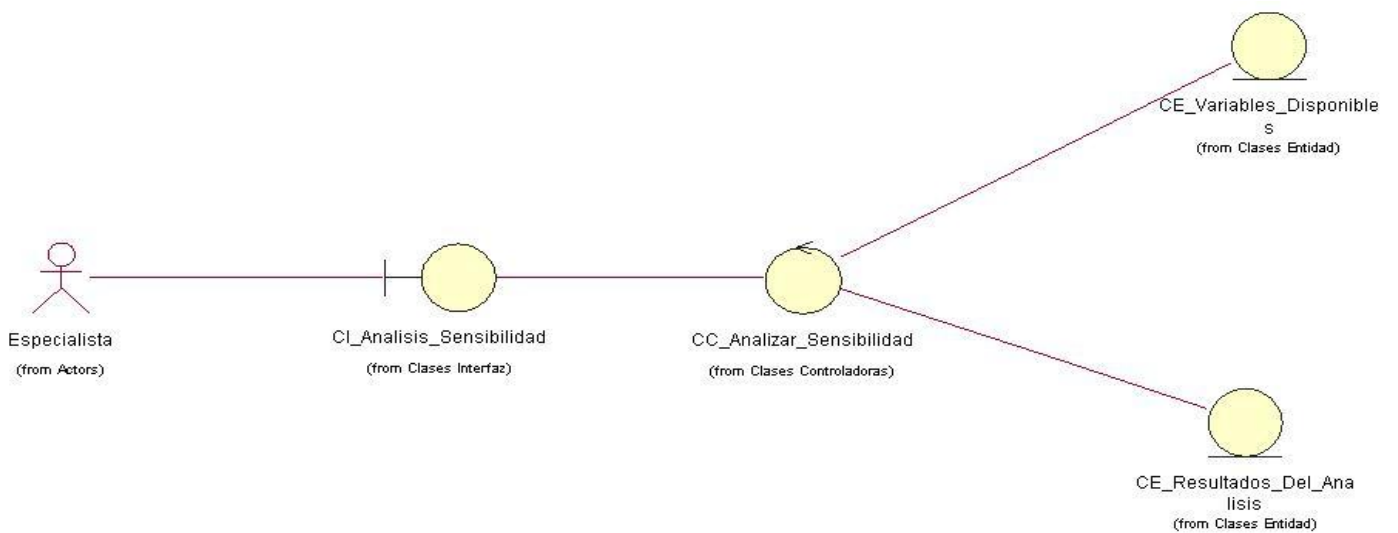
Tabla 10: Descripción del CU Analizar_Sensibilidad

Anexo 2: Diagramas de Clases del Análisis.

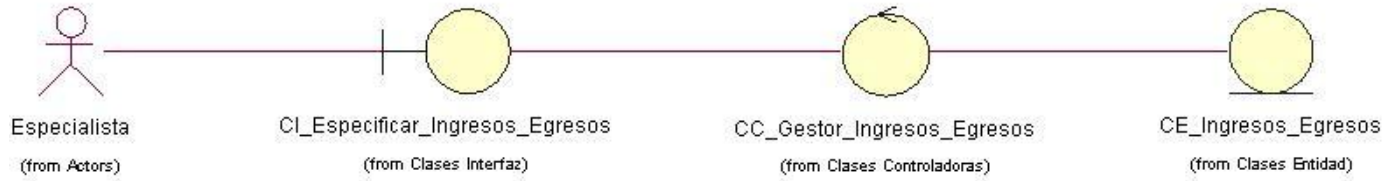
DCA_CU_Analizar Indicadores



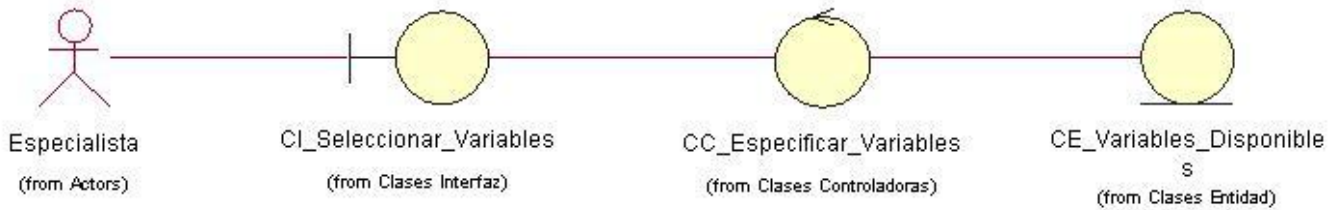
DCA_CU_Analizar Sensibilidad



DCA_CU_Especificar_Ingresos_Egresos



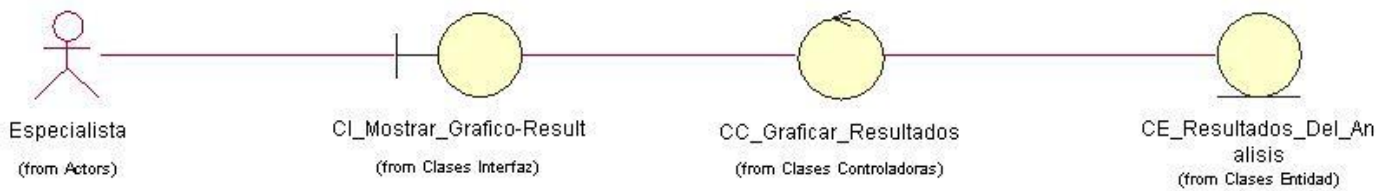
DCA_CU_Especificar Variables



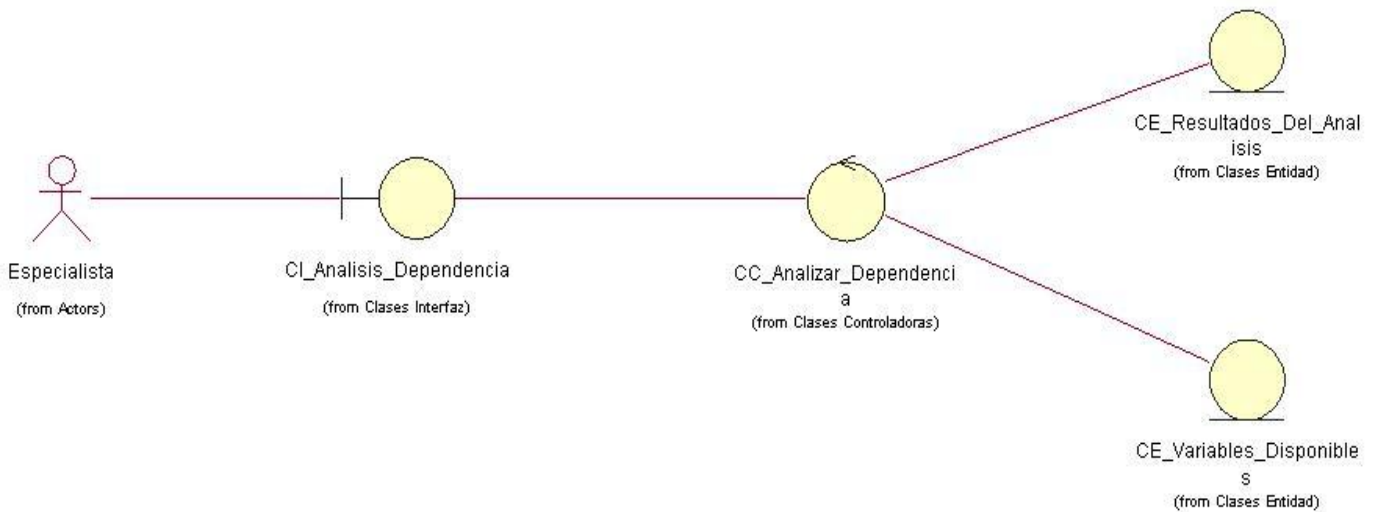
DCA_CU_Generar Reporte Resultados



DCA_CU_Mostrar Gráfico Resultados



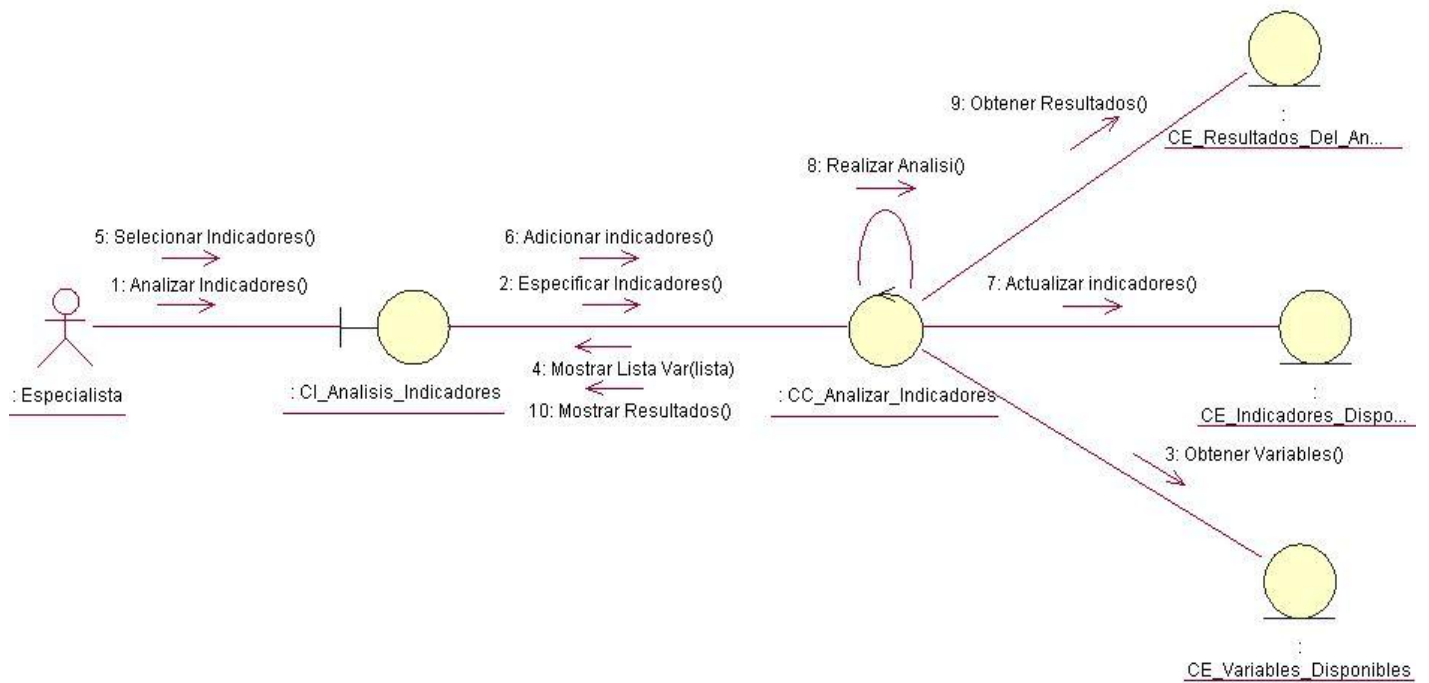
DCA_CU_Analizar Dependencia



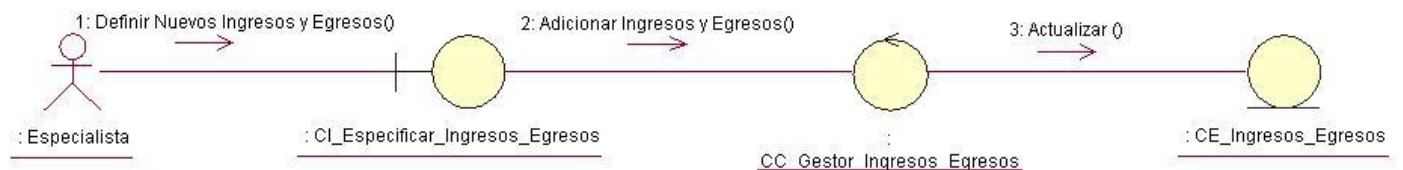
Anexo 3: Diagramas de Interacción del Análisis.

Diagramas de Colaboración del Análisis

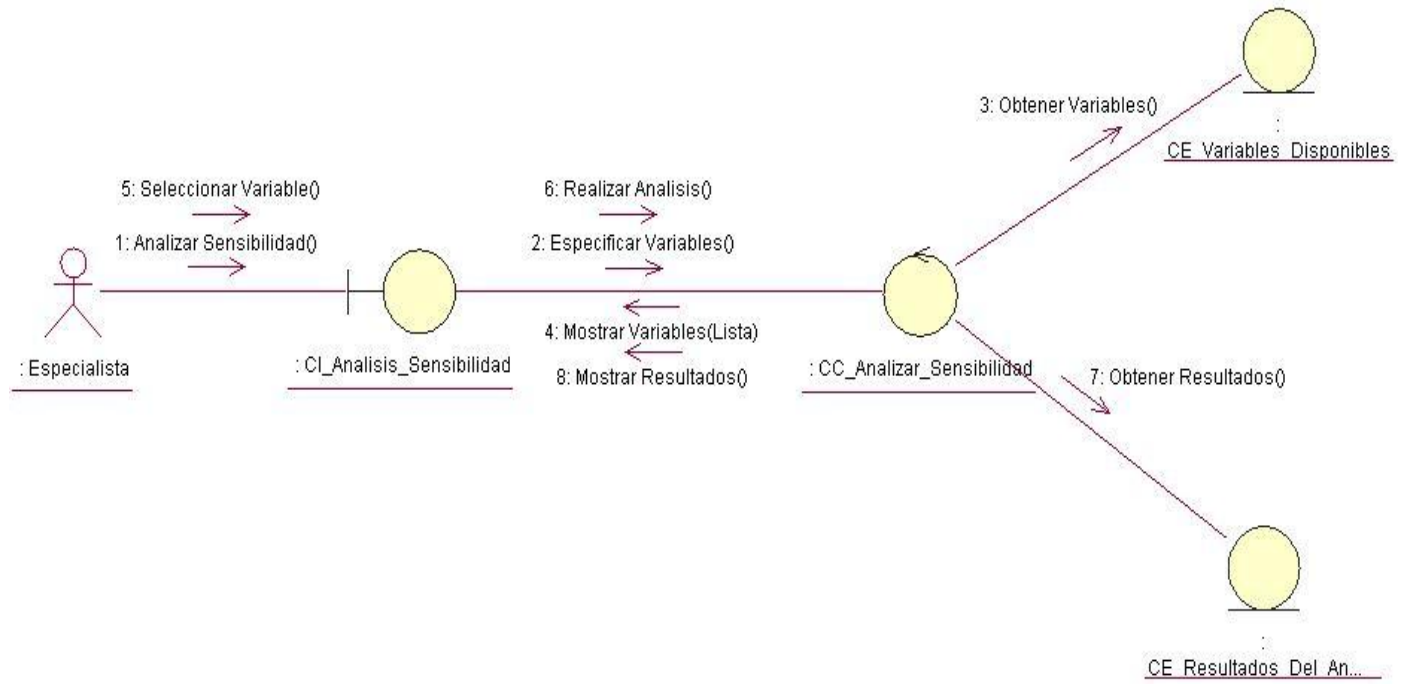
DC_CU_Analizar Indicadores



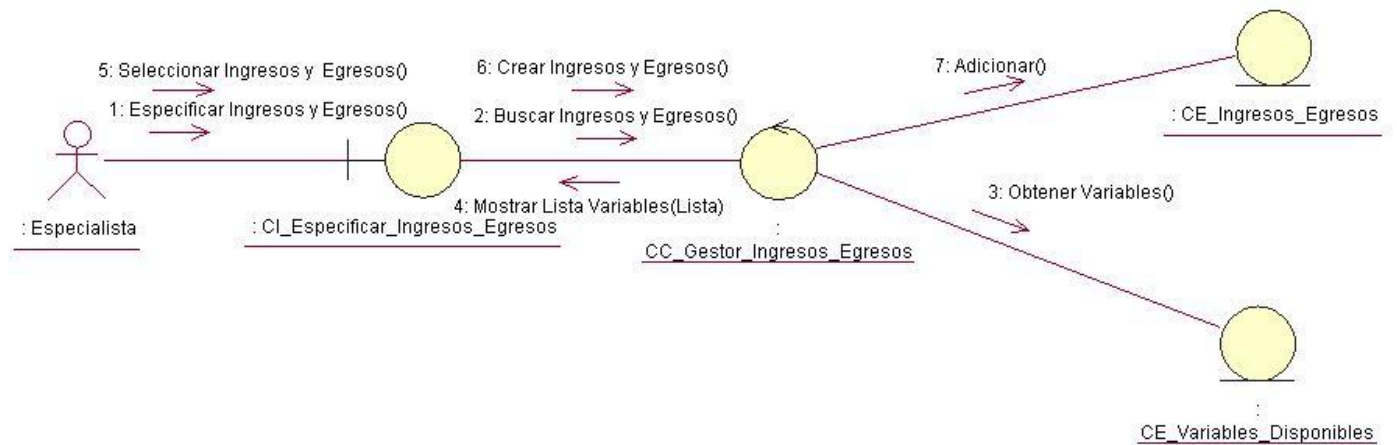
DC_CU_Definir Ingresos_Egresos



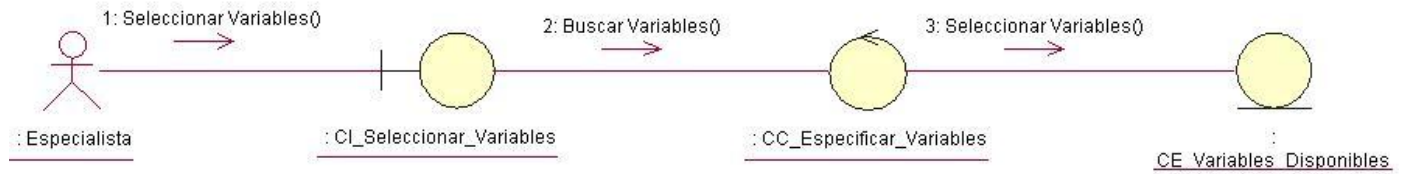
DC_CU_Analizar Sensibilidad



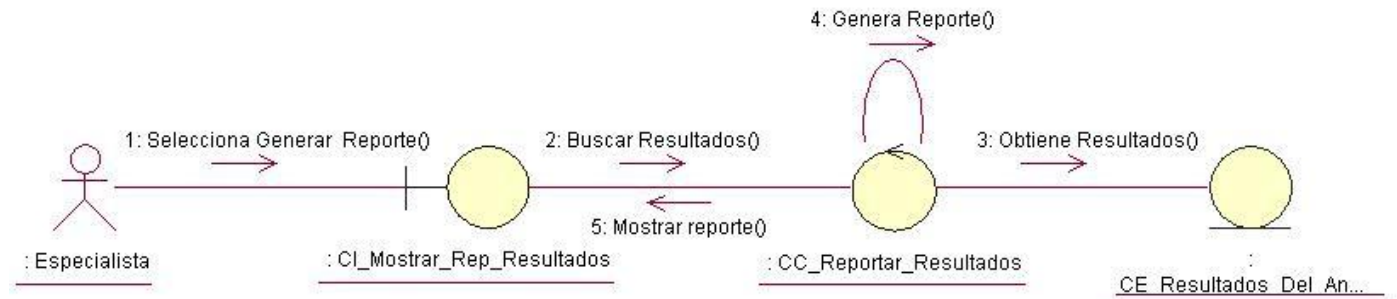
DC_CU_Especificar Ingresos_Egresos



DC_CU_Especificar Variables



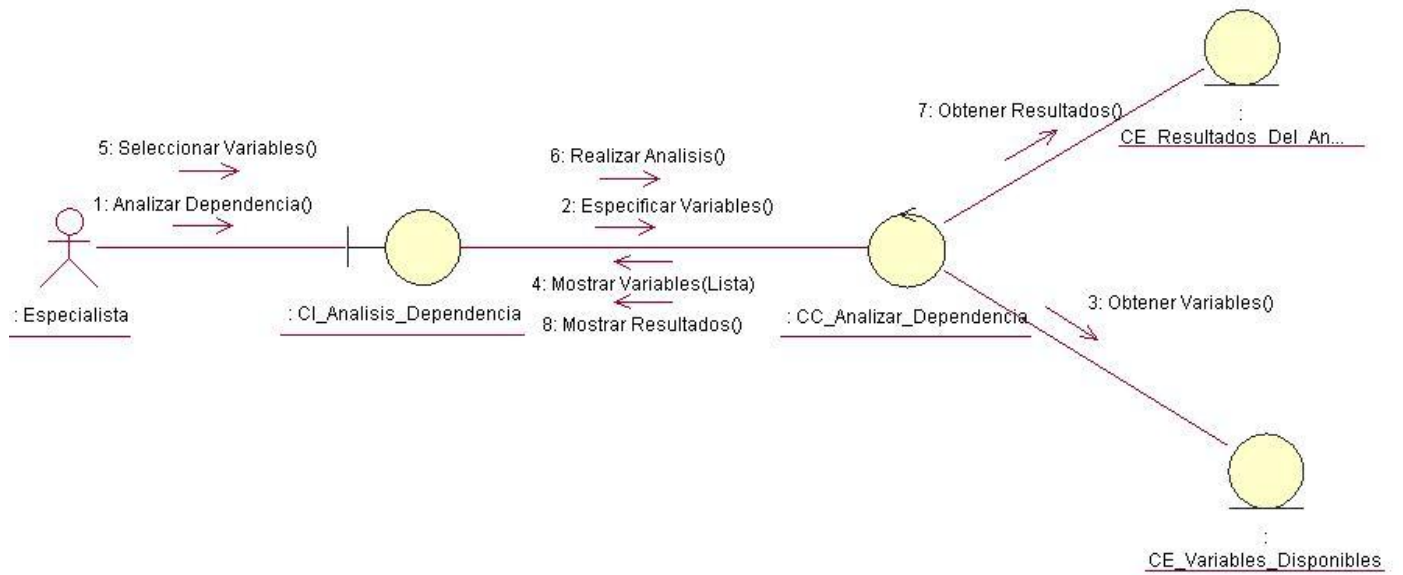
DC_CU_Generar Reporte Resultados



DC_CU_Mostrar Gráfico Resultados

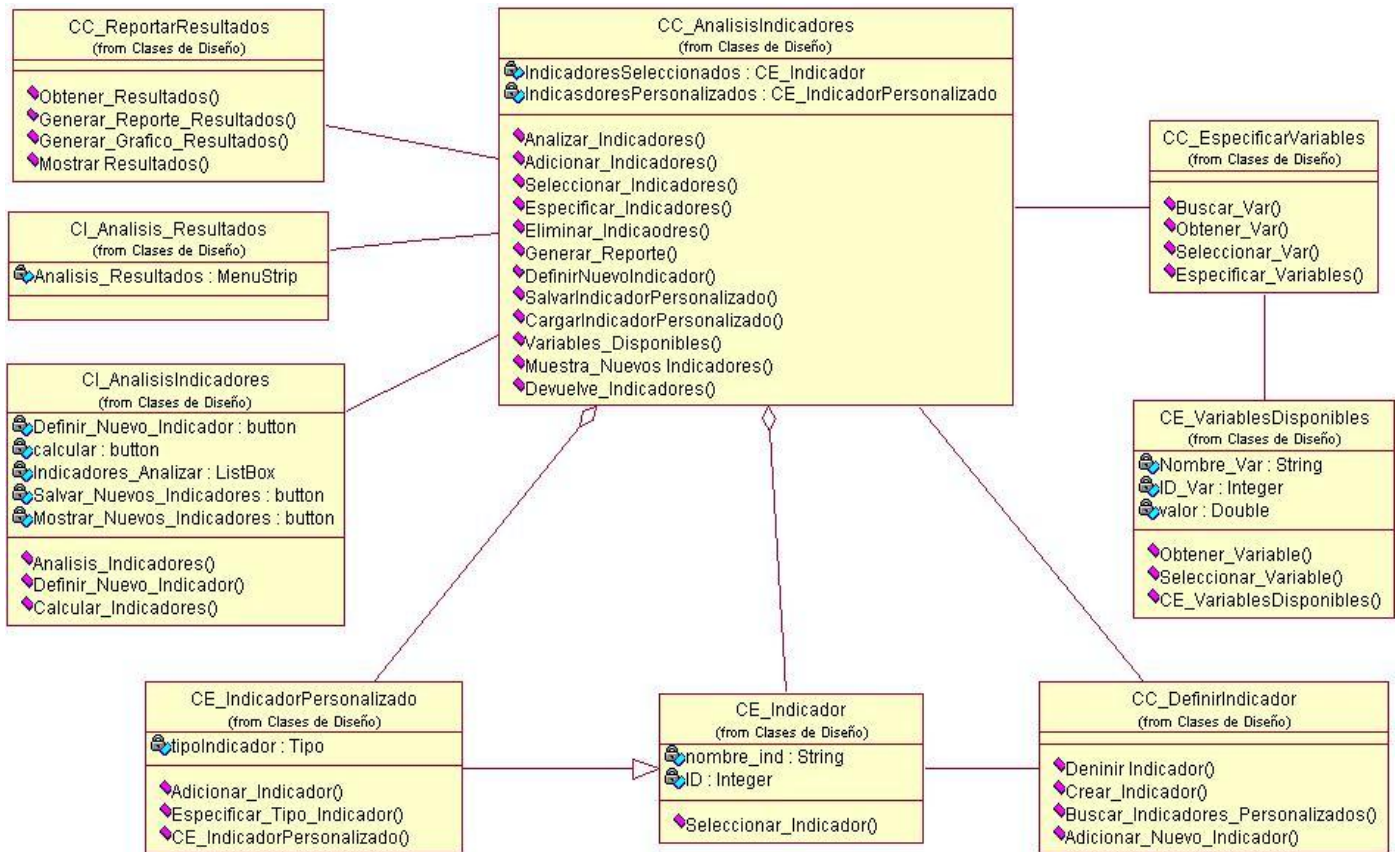


DC_CU_Analizar_Dependencia

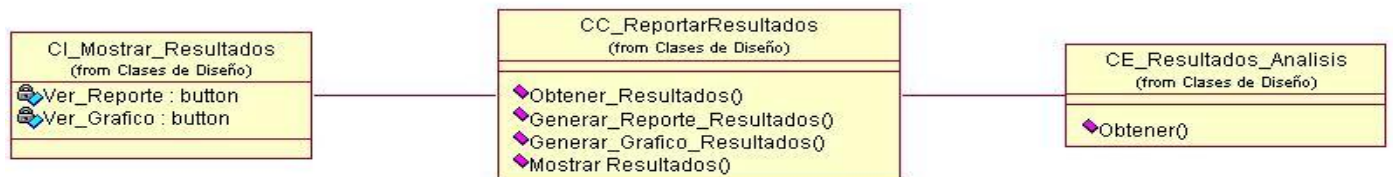


Anexo 4: Diagramas de Clases del Diseño.

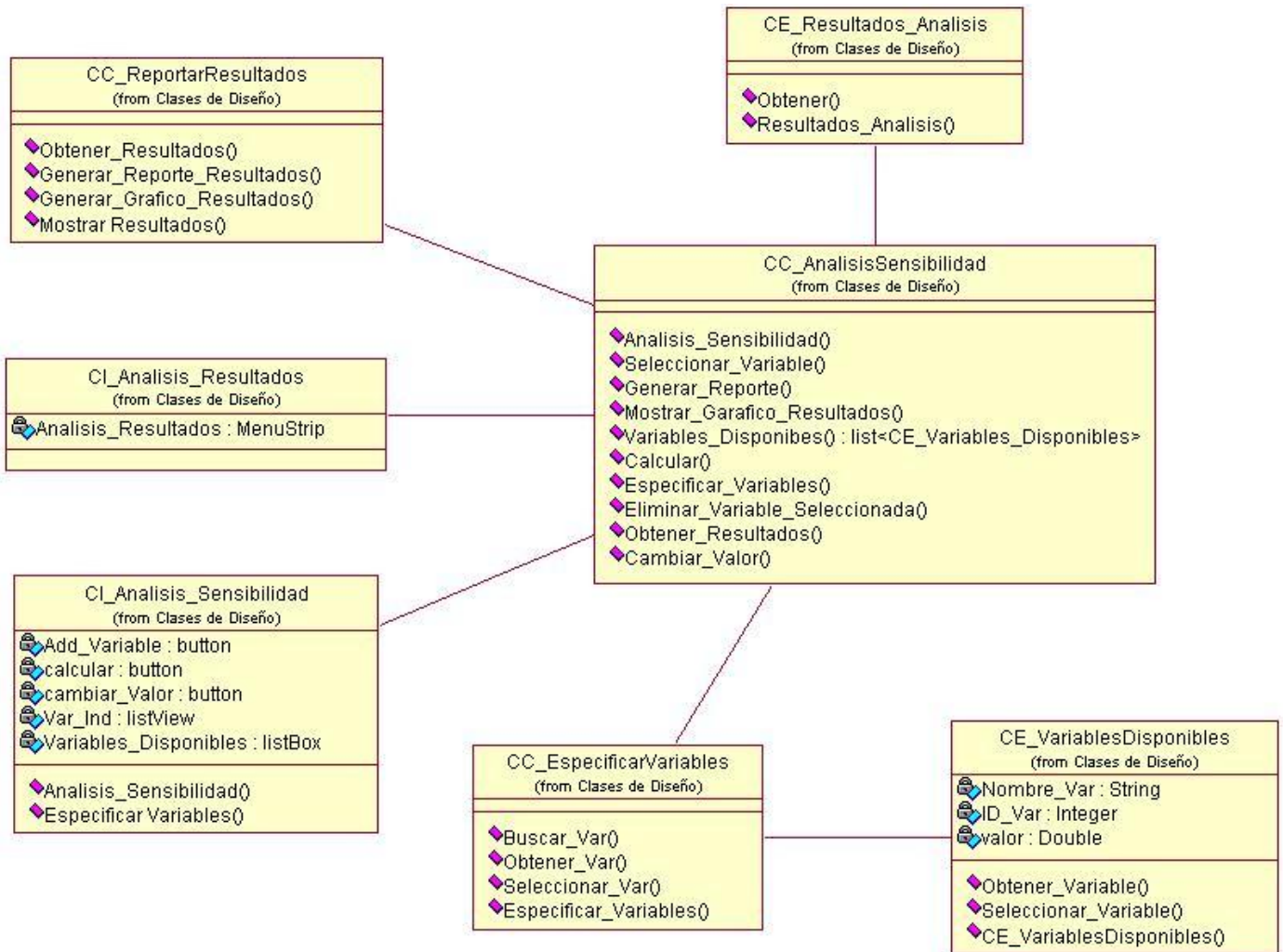
DCD_CU_Analizar_Indicadores



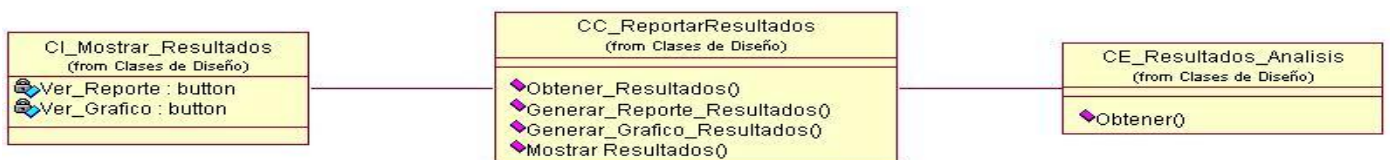
DCD_CU_Generar_Reporte_Resultados



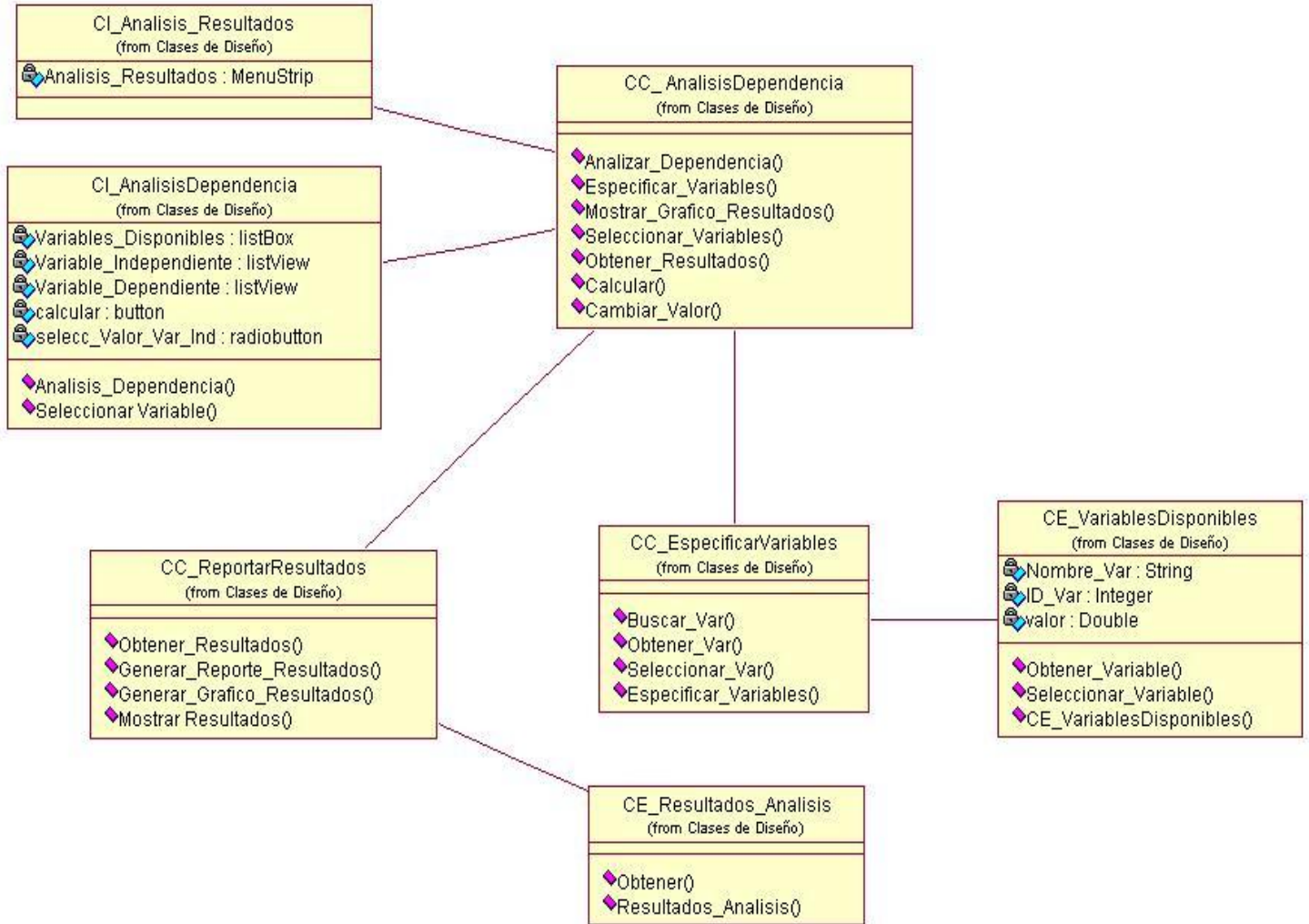
DCD_CU_Analizar_Sensibilidad



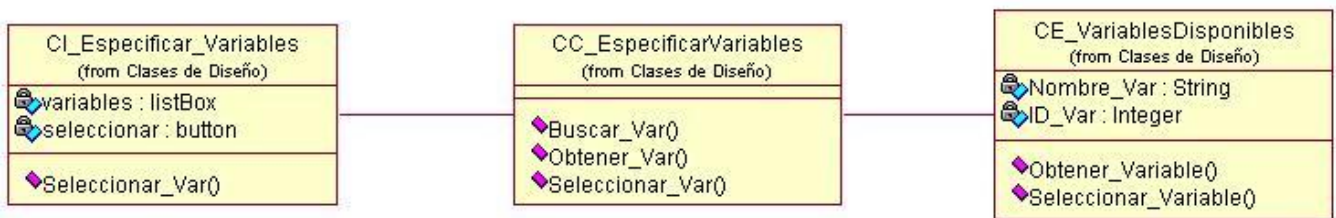
DCD_CU_Mostrar_Gráfico_Resultados



DCD_CU_Analizar_Dependencia



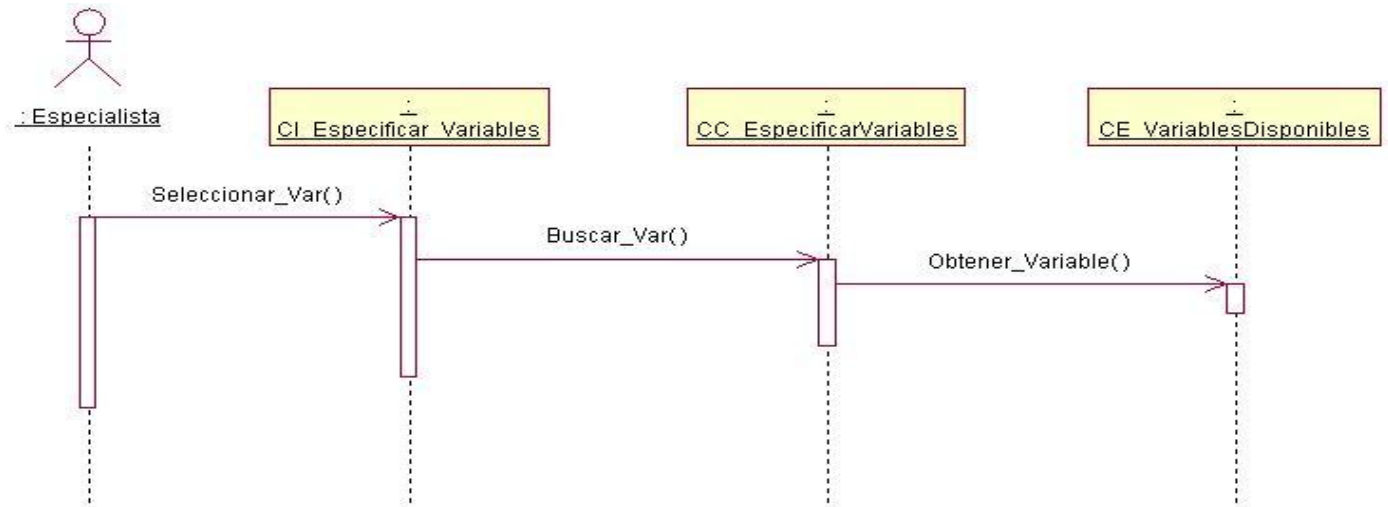
DCD_CU_Especificar_Variables



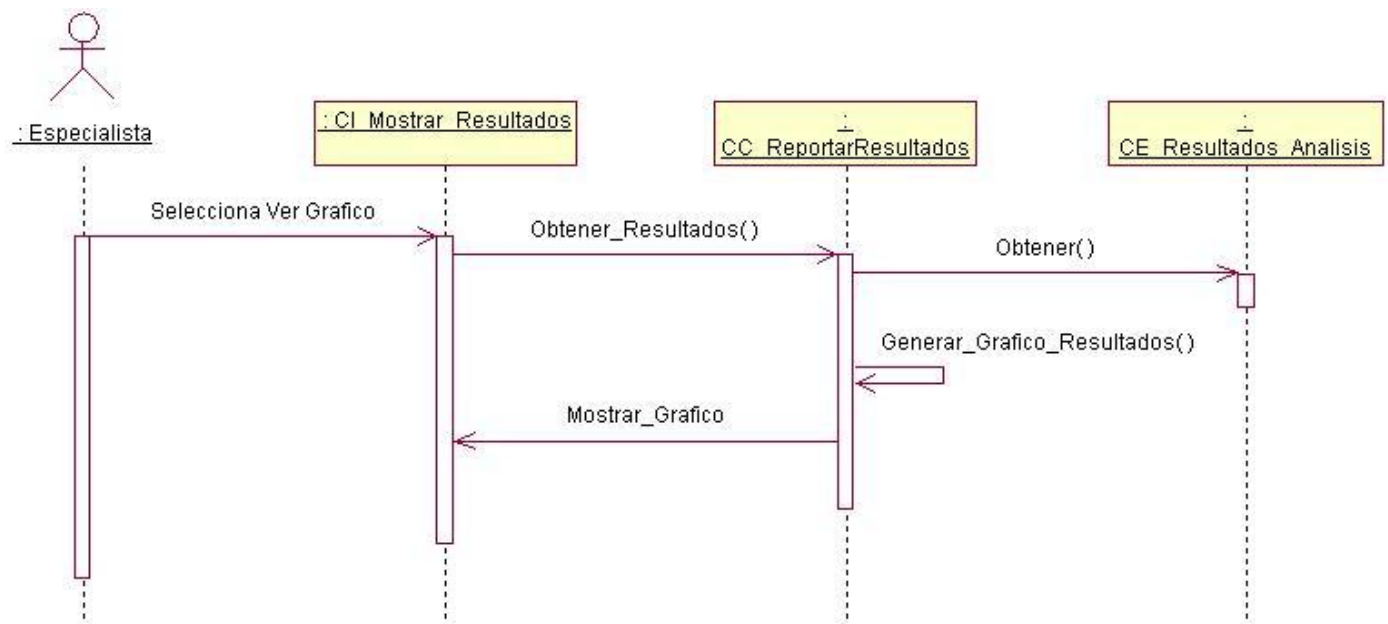
Anexo 5: Diagramas de Interacción del Diseño.

Diagramas de Secuencia del Diseño

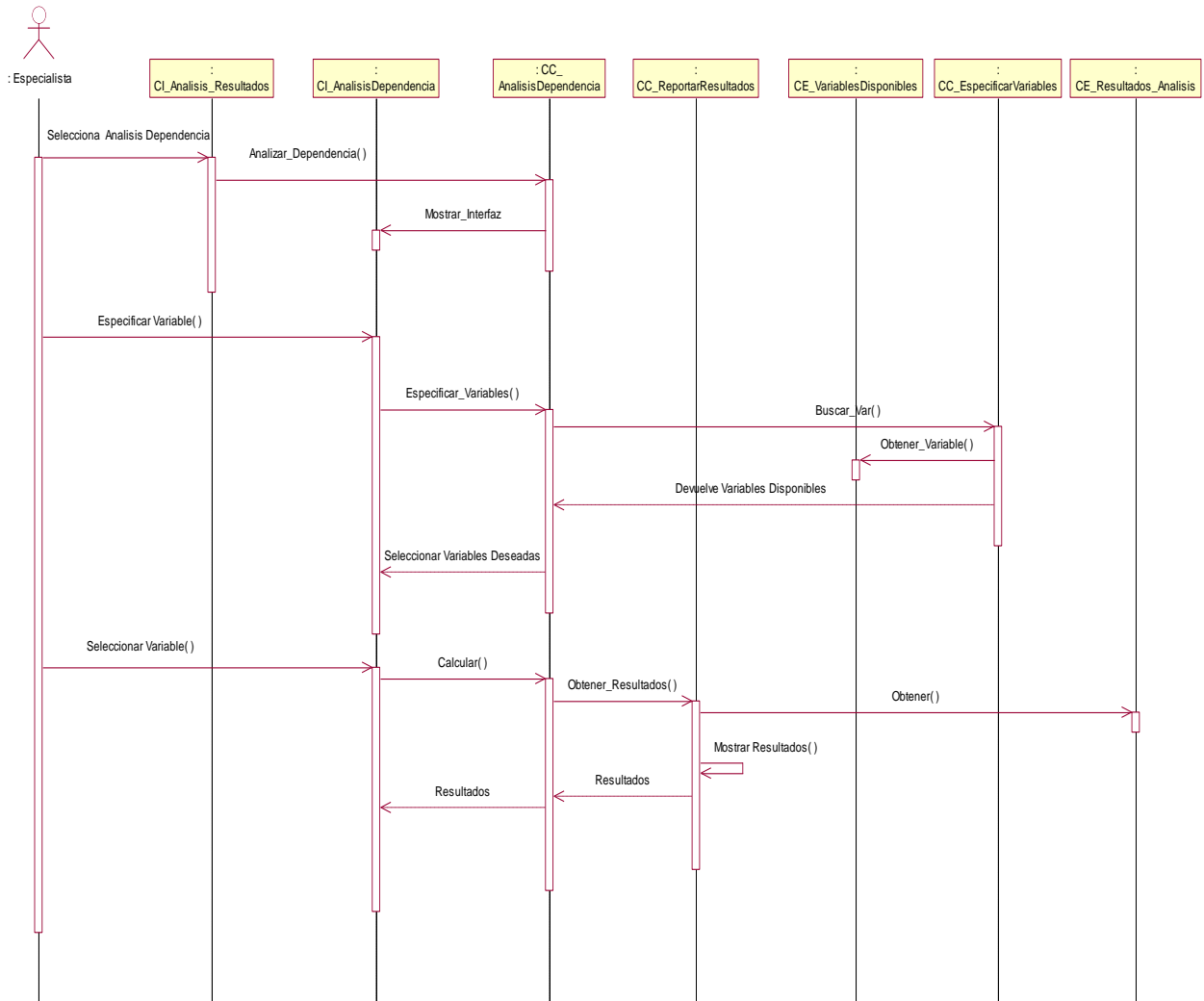
DS_CU_Especificar Variables



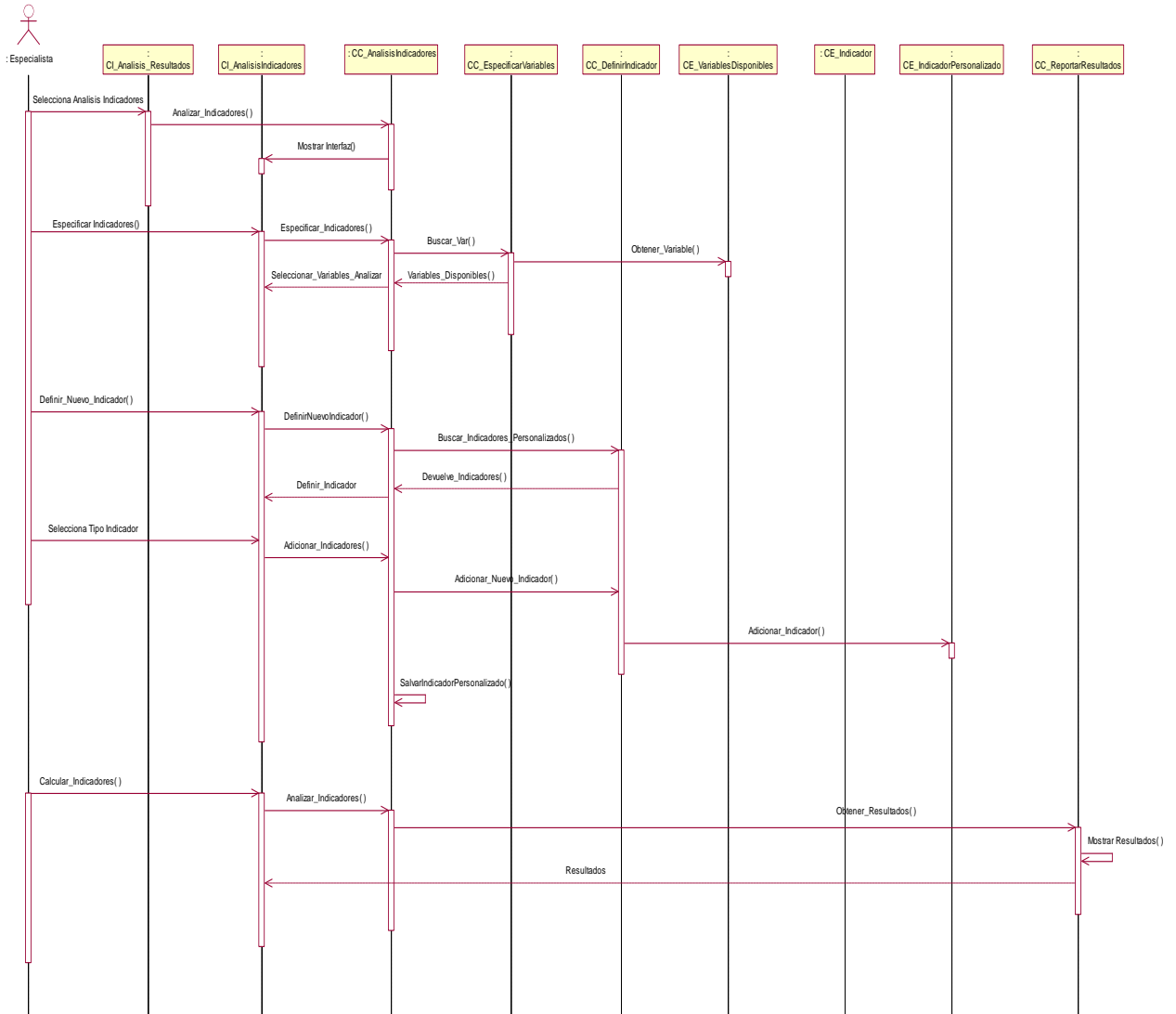
DS_CU_Mostrar Reporte Resultados



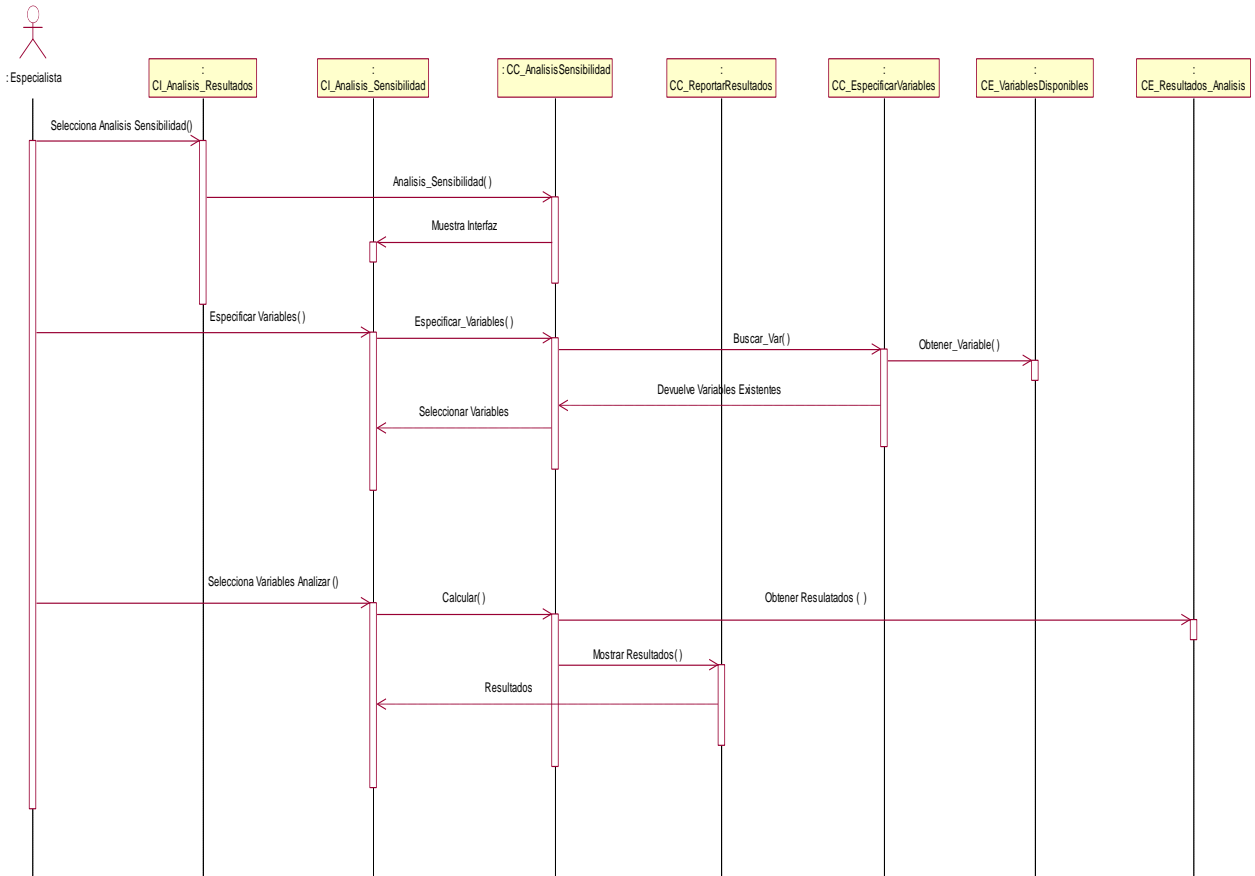
DS_CU_Analizar Dependencia



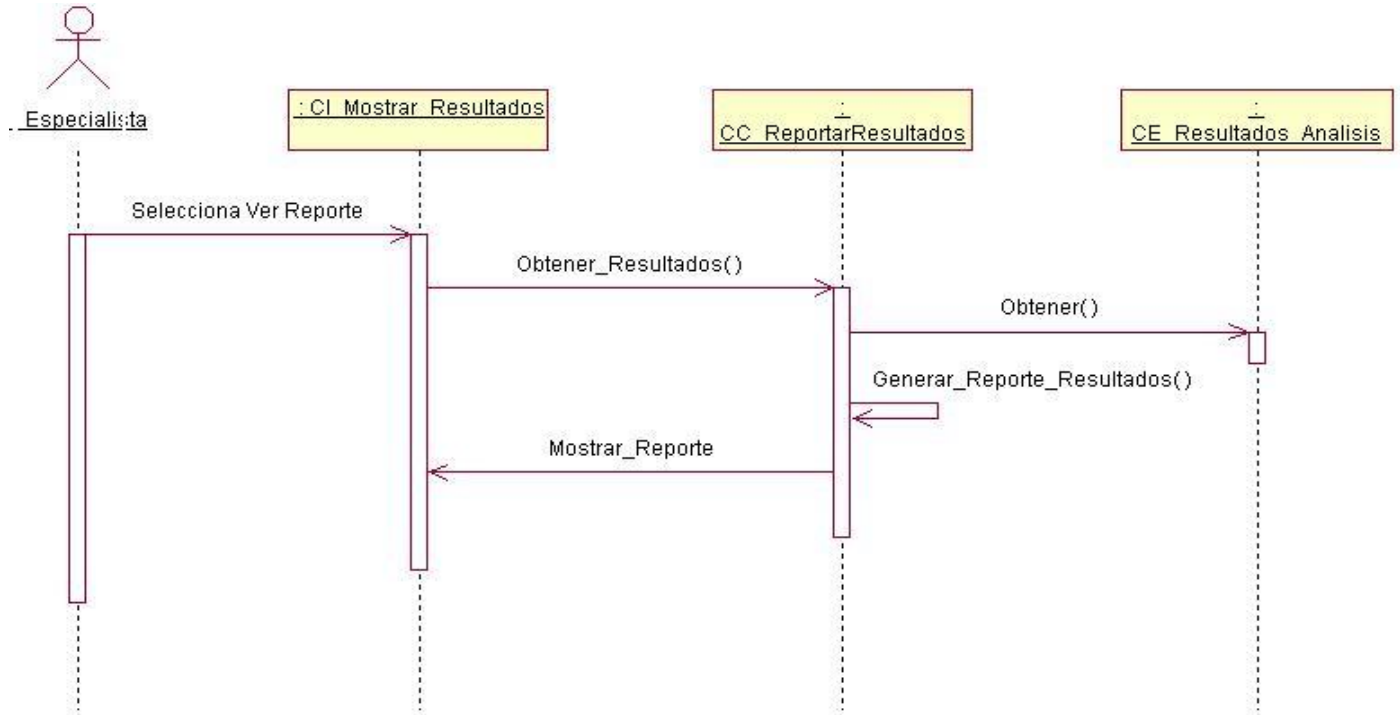
DS_CU_Analizar Indicadores



DS_CU_Analizar Sensibilidad



DS_CU_Generar_Reportes_Resultados



GLOSARIO DE TÉRMINOS

Actor: Abstracción de las entidades externas a un sistema, subsistemas o clases que interactúan directamente con el sistema. Un actor participa en un caso de uso o en conjunto coherente de casos de usos para llevar a cabo un propósito global.

Artefactos: Los productos tangibles del proceso, como por ejemplo, el modelo de casos de uso, el código fuente y otros.

Casos de uso: Especificación de las secuencias de acciones, incluyendo secuencias variantes y secuencias de errores, que pueden ser efectuadas por un sistema, subsistema o clase por interacción con autores externos.

CU: Caso de uso

Clases: Descriptor de un conjunto de objetos que comparten los mismos atributos, operaciones, métodos, relaciones y comportamientos. Una clase representa un concepto dentro del sistema que se está modelando.

DC: Diagrama de Colaboración.

DCA: Diagrama de Clases del Análisis.

DCD: Diagrama de Clases del Diseño.

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

DS: Diagrama de Secuencia.

Fases: Período de tiempo entre dos hitos principales de un proceso de desarrollo

Flujo de trabajo: Realización de un caso de negocio o parte de él. Puede describirse en términos de diagramas de actividad, que incluyen a los trabajadores participantes, las actividades que realizan y los artefactos que producen.

Herramienta: Software que se utiliza para automatizar las actividades definidas en el proceso.

Ingeniería de Software: Aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación y mantenimiento del software.

Modelo: Descripción de las características estáticas, dinámicas o ambas de un tema, presentada en varias vistas.

Proceso de Desarrollo: Definición del conjunto completo de actividades necesarias para transformar los requisitos de un usuario en un producto.

Producto: Artefactos que se crean durante la vida del proyecto, como los modelos, código fuente, ejecutables y documentación.

RUP: Rational Unified Process

Software: Es un término genérico que designa al conjunto de programas de distinto tipo (sistema operativo y aplicaciones diversas) que hacen posible operar con el ordenador.

Tecnología: Conjunto de teorías y de técnicas que permiten el aprovechamiento práctico del conocimiento científico.

Usuario: Un usuario puede ser definido como aquella persona que interactúa con la computadora a nivel de aplicación.