

Universidad de las Ciencias Informáticas

Facultad 9



**DISEÑO DEL SUBSISTEMA PARA EL TRATAMIENTO Y MODELADO
DE DIAGRAMAS DE FLUJO DE INFORMACIÓN.**

TRABAJO PARA OPTAR POR EL TÍTULO DE INGENIERO EN CIENCIAS INFORMÁTICAS

Autor: Joanner Paz Martínez

Tutor: Ing. Arnaldo Gandol

Cotutor: Ing. Yurisnel Corrales Valdés

Oponente: Ing. Yoandrys González González

Cuidad de la Habana. 18 de mayo del 2009.

“Emplearse en lo estéril cuando se puede hacer lo útil, dedicarse a lo fácil cuando se tienen bríos para intentar lo difícil, es despojar de su dignidad al talento. Todo el que deja de hacer lo que es capaz de hacer, peca.”

José Martí

AGRADECIMIENTOS

*En primer lugar a mis padres, que sin contar con sus ejemplos, apoyo, regaños, consejos y sabiduría, hoy no fuera ni la mitad de lo que soy. A Triny por dos razones: Una, por darme un hermano. Dos, por ser amiga, consultante y por la ayuda que siempre me has brindado en estos años. A mi hermano Alain, sin tu apoyo hoy no sería quien soy. A Naty, por ser mi novia, amiga, compañera y por compartirlo todo conmigo, por tu apoyo, ayuda, atenciones y en especial por tu amor y tu cariño. A mis amistades en general por ayudarme cuando lo necesité, por darme las fuerzas cuando me faltaron, por darme ánimos cuando pensé en no continuar, por confiar en mí y no defraudarme nunca. Al Pacho, brother sin ti esto hubiera sido mucho más difícil de lo que fue. Ya todo el que de una manera u otra contribuyó con el desarrollo del presente trabajo.
¡GRACIAS!*

DEDICATORIA

A mis padres, a mi hermano y a los amigos de toda la vida...

RESUMEN

En la actualidad una de las vías más aplicadas para lograr altos niveles de eficiencia en las industrias es la aplicación de herramientas de la ingeniería de procesos. Una de ellas es la simulación, que apoyada en técnicas de inteligencia artificial y minería de datos, entre otras, proveen a los ingenieros de potentes herramientas para el estudio de los procesos industriales. Pero aún son muchos los sectores que no aprovechan las ventajas que esta tecnología ofrece para la toma de decisiones, el ahorro de costos o la optimización de procesos industriales. La simulación comprende la creación de un escenario artificial y el análisis del mismo para detectar los problemas que presenta el proceso industrial representado. Dicha representación se realiza a través de distintos diagramas como son el Diagrama de Procesos (DP), Diagrama de Simulación (DS) y el Diagrama de Flujo de Información (DFI). Es por este motivo que un simulador debe poseer a demás de diferentes funcionalidades para cálculos termodinámicos y de equipos industriales, una serie de facilidades para la edición y tratamiento de los DFI. Por esta razón el presente trabajo de diploma tuvo como objetivo, lograr el diseño de un subsistema que permita el tratamiento y modelado de los DFI para el simulador DFISim. Aplicándose a lo largo de la presente investigación diferentes métodos científicos, tales como el Analítico-Sintético y la Modelación. Se estudiaron las características de algunos de los simuladores existentes en Cuba y el resto del mundo. Se analizó cuales serían las funcionalidades a incluir dentro de la propuesta a partir del levantamiento de los requerimientos funcionales y no funcionales. La información obtenida fue debidamente registrada siguiendo la metodología Proceso Unificado de Desarrollo (RUP) y el Lenguaje de Modelado Unificado (UML), además del Visual Paradigm for UML para realizar el modelado en todo el ciclo de vida del proyecto, específicamente el análisis y diseño, objetivo fundamental del presente trabajo de diploma.

ÍNDICE DE FIGURAS

Figura 1: Diagrama de Proceso.	9
Figura 2: Diagrama de Simulación.	9
Figura 3 Diagrama de Flujo de Información.	10
Figura 4: DCUS Sección de Gestión de Módulo.	43
Figura 5: Diagrama de CUS Sección de Gestión de Etiqueta.....	47
Figura 6: DCUS Sección de Gestión de Conexión.	49
Figura 7: Diagrama de CUS Sección de Gestión de Capa.	51
Figura 8: Diagrama de CUS Sección de Gestión de Selección.	56
Figura 9: Diagrama de CUS Sección de Gestión del DFI.	60
Figura 10: Diagrama de Clases del Análisis del CU Dibujar Módulo	67
Figura 11: Diagrama de Clases del Análisis del CU Mover Módulo.....	68
Figura 12: Diagrama de Clases del Análisis del CU Ocultar Módulo.	68
Figura 13: Diagrama de Clases del Análisis del CU Seleccionar Módulo.	68
Figura 14: Diagrama de Clases del Análisis del CU Dibujar Etiqueta.	69
Figura 15: Diagrama de Clases del Análisis del CU Crear Conexión.	69
Figura 16: Diagrama de Clases del Análisis del CU Mover Conexión.	69
Figura 17: Diagrama de Clases del Análisis del CU Agrupar en Capa.	70
Figura 18: Diagrama de Clases del Análisis del CU Bloquear Capa.	70
Figura 19: Diagrama de Clases del Análisis del CU Ocultar Capa.	71
Figura 20: Diagrama de Clases del Análisis del CU Mover Capa.	71
Figura 21: Diagrama de Clases del Análisis del CU Seleccionar Capa.	72
Figura 22: Diagrama de Clases del Análisis del CU Mover Multiselección.	72

Figura 23: Diagrama de Clases del Análisis del CU Multiseleccionar.....73

Figura 24: Diagrama de Clases del Análisis del CU Eliminar Multiselección.73

Figura 25: Diagrama de Clases del Análisis del CU Exportar Diagrama74

Figura 26: Diagrama de Clases del Análisis del CU Realizar Zoom.74

Figura 27: Diagrama de Clases del Análisis del CU Deshacer Última Acción.75

Figura 28: Diagrama de Clases del Análisis del CU Cortar Componentes.75

Figura 29: Diagrama de Clases del Análisis del CU Pegar Componentes.....76

Figura 30: Clases Involucradas en el Patrón Memento.79

Figura 31: Clases involucradas en el Patrón Observador.....80

Figura 32: Diagrama de Secuencia del CU Seleccionar Módulo.84

Figura 33: Diagrama de Secuencia del CU Ocultar Módulo.84

Figura 34: Diagrama de Secuencia del CU Mover Módulo.....85

Figura 35: Diagrama de Secuencia del CU Dibujar Módulo85

Figura 36: Diagrama de Secuencia del CU Dibujar Etiqueta.86

Figura 37: Diagrama de Secuencia del CU Crear Conexión.87

Figura 38: Diagrama de Secuencia del CU Agrupar en Capa.88

Figura 39: Diagrama de Secuencia del CU Bloquear Capa.....88

Figura 40: Diagrama de Secuencia del CU Ocultar Capa89

Figura 41: Diagrama de Secuencia del CU Mover Capa.....89

Figura 42: Diagrama de Secuencia del CU Seleccionar Capa.90

Figura 43: Diagrama de Secuencia del CU Mover Multiselección.91

Figura 44: Diagrama de Secuencia del CU Multiseleccionar.....92

Figura 45: Diagrama de Secuencia del CU Eliminar Multiselección.93

Figura 46: Diagrama de Secuencia del CU Cortar Componentes.94

Figura 47: Diagrama de Secuencia del CU Deshacer Última Acción.....94

Figura 48: Diagrama de Secuencia del CU Pegar Componentes.....95

Figura 49: Diagrama de Secuencia del CU Realizar Zoom.95

Figura 50: Diagrama de Secuencia del CU Exportar Diagrama.....96

ÍNDICE DE TABLAS

Tabla 1: Definición de los Actores del Sistema.	42
Tabla 2: Descripción del CUS Dibujar Módulo	43
Tabla 3: Descripción del CUS Seleccionar Módulo	44
Tabla 4: Descripción del CUS Mover Módulo.....	45
Tabla 5: Descripción del CUS Ocultar Módulo.	46
Tabla 6: Descripción del CUS Dibujar Etiqueta.	48
Tabla 7: Descripción del CUS Crear Conexión.	49
Tabla 8: Descripción del CUS Mover Conexión.	50
Tabla 9: Descripción del CUS Agrupar en Capa.	51
Tabla 10: Descripción del CUS Seleccionar Capa.	52
Tabla 11: Descripción del CUS Ocultar Capa.	53
Tabla 12: Descripción del CUS Bloquear Capa.....	54
Tabla 13: Descripción del CUS Mover Capa.	55
Tabla 14: Descripción del CUS Multiseleccionar.	57
Tabla 15 Descripción del CUS Mover Multiselección.	58
Tabla 16 Descripción de CUS Eliminar Multiselección.	59
Tabla 17: Descripción del CUS Realizar Zoom.	60
Tabla 18: Descripción del CUS Deshacer Última Acción.....	61
Tabla 19: Descripción del CUS Cortar Componentes.	62

Tabla 20: Descripción del CUS Pegar Componentes63

Tabla 21: Descripción del CUS Exportar Diagrama.....64

ÍNDICE

INTRODUCCIÓN.....	1
Tareas de la investigación:.....	3
Métodos Científicos utilizados en la investigación.....	4
Métodos Teóricos:.....	4
Métodos Empíricos:.....	4
CAPÍTULO # 1: FUNDAMENTACIÓN TEÓRICA.....	5
1.1 Introducción.....	5
1.2 Estado del Arte.....	5
1.3 Fundamentación del Tema.....	11
1.3.1 Definición de Simulación.....	11
1.3.2 Aplicaciones de la simulación de procesos.....	12
1.3.3 Tipos de Simulación.....	14
1.3.4 Etapas para realizar un estudio de simulación.....	15
1.3.5 Aceptación de la Simulación.....	17
1.4 Metodologías de desarrollo de Software.....	19
1.4.1 Rational Unified Process (RUP).....	19
1.4.2 Lenguaje de Modelado.....	22
1.4.3 Descripción de la Herramienta Case para el Modelado.....	23
1.4.4 Patrones de casos de uso.....	24
1.5 Rol de Analista.....	27
1.5.1 ¿Qué es un rol?.....	27
1.5.2 El rol de Analista.....	28
1.6 Conclusiones.....	28
CAPÍTULO # 2: PROPUESTA DE SOLUCIÓN.....	29
2.1 Introducción.....	29
2.2 Captura de Requisitos.....	29
2.2.1 Especificación de Requisitos.....	30

2.3	Requisitos Funcionales.....	31
2.3.1	Funcionalidad Gestionar Módulo.....	31
2.3.2	Funcionalidad Gestionar Conexión.....	33
2.3.3	Funcionalidad Gestionar Capas.....	34
2.3.4	Funcionalidad Gestionar Etiqueta.....	36
2.3.5	Funcionalidad Gestionar DFI.....	38
2.3.6	Funcionalidad Multiselección.....	40
2.4	Requisitos no Funcionales.....	41
2.4.1	Requisitos de software.....	41
2.4.2	Requisitos de hardware.....	41
2.4.3	Requisitos de Usabilidad.....	41
2.4.4	Requisitos de rendimiento.....	42
2.4.5	Restricciones de diseño o implementación.....	42
2.5	Definición de los Actores del Sistema.....	42
2.6	Diagrama de Casos de Uso del Sistema.....	42
2.6.1	Sección de Gestión de Módulo.....	43
2.6.2	Sección de Gestión de Etiqueta.....	47
2.6.3	Sección de Gestión de Conexión.....	49
2.6.4	Sección de Gestión de Capa.....	51
2.6.5	Sección de Gestión de Selección.....	56
2.6.6	Sección de Gestión del DFI.....	60
2.7	Conclusiones.....	65
CAPÍTULO # 3: ANÁLISIS Y DISEÑO DE LA SOLUCIÓN PROPUESTA.....		66
3.1	Introducción.....	66
3.2	Modelo del Análisis.....	66
3.3	Diagramas de Clases del Análisis.....	67
3.3.1	Clases del Análisis.....	67
3.3.2	Diagramas de Clases del Análisis. Sección de Gestión de Módulo.....	67
3.3.3	Diagramas de Clases del Análisis. Sección de Gestión de Etiqueta.....	69

3.3.4 Diagramas de Clases del Análisis Sección de Gestión de Conexión69

3.3.5 Diagramas de Clases del Análisis. Sección de Gestión de Capa.....70

3.3.6 Diagramas de Clases del Análisis. Sección de Selección.72

3.3.7 Diagramas de Clases del Análisis. Sección de Gestión del DFI.....74

3.4 Modelo de Diseño.76

3.4.1 Patrones de Diseño.76

3.5 Diagrama de Clases del Diseño.81

3.6 Diagramas de Interacción.....83

3.6.1 Sección de Gestión de Módulo.84

3.6.2 Sección de Gestión de Etiqueta.....86

3.6.3 Sección de Gestión de Conexión.87

3.6.4 Sección de Gestión de Capa.88

3.6.5 Sección de Selección.....91

3.6.6 Sección de Gestión del DFI.94

3.7 Conclusiones.....96

CONCLUSIONES GENERALES.....97

RECOMENDACIONES98

TRABAJOS CITADOS.....99

INTRODUCCIÓN

En la actualidad una de las vías más aplicadas para lograr mejores niveles de eficiencia en las industrias es logrando la correcta aplicación de herramientas de ingeniería de procesos. Una de ellas es la simulación, que apoyada en el análisis y la síntesis de procesos, en técnicas de inteligencia artificial, minería de datos; y complementada con los análisis ambientales, económicos y/o estadísticos proveen a los ingenieros de potentes herramientas para el estudio de los diferentes procesos industriales. Pero desafortunadamente, aún son muchos los sectores industriales que no aprovechan las ventajas que esta tecnología ofrece para la toma de decisiones, el ahorro de costos o la optimización de procesos industriales.

La simulación también puede ser aplicada a un amplio abanico de problemas y sectores, donde se pueden destacar el diseño de industrias de fabricación o modificación de plantas/líneas de producción ya sea en serie o no. En el sector logístico, para el diseño de flujos de transporte de personas o materiales, optimización de plataformas, playas o terminales. Además de estos sectores, la simulación también tiene aplicación en escenarios tales como la medicina, desarrollos urbanísticos, policía, sector bancario, entre otros.

En Cuba se han desarrollado algunos simuladores y se han aplicado en procesos completos o subprocesos relacionados con la producción de etanol, refinación de petróleo y de sulfato ferroso. En el ICIDCA se dispone de un Simulador, orientado a ecuaciones del proceso azucarero y de refinación de alcohol de caña llamado SIMFAD 3.0, resultado de todo un amplio trabajo anterior de modelación matemática y simulación de equipos y subprocesos de la industria azucarera. En la UCLV también se desarrolló un Simulador de procesos tecnológicos, pero no aparecen referenciadas aplicaciones en la industria cubana. Otro Simulador desarrollado en el MINAZ (Villa clara) es el denominado AGE de carácter determinístico y que contiene recomendaciones de cómo actuar en función de los resultados. Pero al igual que otro paquete de simulación de la UCLV está basado en métodos de cálculos simplificados o rápidos que no permiten obtener todos los resultados importantes y necesarios que pueden dar los Simuladores que usan modelos más precisos y sin simplificaciones innecesarias si se dispone de computadoras [1].

Actualmente existe en Cuba el Centro de Investigación y Desarrollo de Simuladores (SIMPRO), perteneciente a las Fuerzas Armadas Revolucionarias (FAR), en el cual se han desarrollado varios simuladores, con el objetivo de simular el comportamiento de tanques de guerra, simuladores de tiro entre otros, obteniéndose resultados importantes en la rama de la simulación de entornos virtuales, en la representación en tres dimensiones y en el tratamiento de imágenes.

La Universidad de las Ciencias Informáticas ha incursionado en este campo de la simulación, obteniendo resultados como es el caso del simulador Sistema Termo Azúcar (STA v4.0) basado en la industria química contando con posibilidades de cálculos y análisis de resultados.

Todo lo logrado en el campo de la simulación en Cuba, está limitado porque en la mayoría de las industrias cubanas a pesar de que existe la tecnología que pudiera soportar herramientas de simulación que ayudarían en la toma de decisiones o en el planeamiento de nuevos proyectos; estas no existen o son precarias, ya que carecen en su mayoría, de las facilidades que brindan sus similares extranjeras. La mayoría de los simuladores de procesos aplicados en la industria cubana, que no fueron desarrollados en Cuba; se encuentran bajo la licencia de Software Propietario y/o son copias no autorizadas de simuladores extranjeros, dificultando por este motivo la posible modificación de las facilidades que brindan con el objetivo de adecuarlas a las necesidades de determinada empresa o industria. Este tipo de herramientas de simulación solo puede usarse internacionalmente para fines académicos.

Tanto el STA v4.0, como algunos de los simuladores cubanos mencionados anteriormente, presentan deficiencias en lo que a su interfaz gráfica de usuario respecta. Ya sea porque en algunos casos necesitan muchos recursos de hardware o porque carecen de facilidades necesarias para los usuarios finales tales como las de ampliar y/o reducir los diagramas de flujo de información que se estén diseñando (zoom), la multiselección y redimensión de componentes en dichos diagramas, conexión de corrientes entre componentes con un grado mínimo de sencillez que permita ganar tiempo en el diseño, tratamiento de componentes a nivel de capas o grupos de componentes y otras facilidades para la edición y el trabajo con estos a nivel de interfaz.

Debido a estas deficiencias del STA v4.0, y a la inexistencia de un subsistema encargado del tratamiento y modelado de los diagramas de flujo de información que cumpla con las condiciones anteriores; se considera como **problema científico**:

Necesidad de la creación de un subsistema de interfaz gráfica de usuario encargado del tratamiento y modelado de Diagramas de Flujo de Información para el simulador DFISim.

Objeto de estudio: Proceso de desarrollo de los simuladores de procesos industriales.

Campo de acción: Diseño de subsistemas de interfaz gráfica de usuario para el tratamiento y modelado de Diagramas de Flujo de Información de simuladores de procesos en la facultad 9 de la Universidad de las Ciencias Informáticas.

Hipótesis: Diseñando un subsistema de interfaz gráfica de usuario para el simulador **DFISim** se logrará un mejor tratamiento y modelado de los Diagramas de Flujo de Información.

Objetivo General: Diseñar un subsistema de interfaz gráfica de usuario que permita el tratamiento y modelado de los Diagramas de Flujo de Información del simulador **DFISim**.

Tareas de la investigación:

1. Caracterización de otros sistemas con características similares al subsistema propuesto.
2. Evaluación de otros sistemas con características similares al subsistema propuesto.
3. Captura de los requisitos funcionales y no funcionales del subsistema propuesto.
4. Modelado del diagrama de casos de uso del subsistema propuesto.
5. Descripción textual de los casos de uso del subsistema propuesto.
6. Análisis del subsistema.
 - a. Creación de las realizaciones de casos de usos de análisis.
 - b. Complementación de la descripción de los casos de usos.
 - c. Definición de las clases de análisis, distribución del comportamiento y descripción las funcionalidades y atributos relacionados de estas clases.
7. Diseño del subsistema.
 - a. Distribuir el comportamiento del subsistema en los elementos del subsistema.

- b. Documentación de la estructura interna del subsistema.
- c. Definición de las realizaciones entre las interfaces del subsistema y las clases contenidas.

Con este trabajo se persiguen los siguientes resultados:

- Especificación de requisitos funcionales y no funcionales.
- Modelo de Casos de Usos.
- Modelo de Análisis.
- Modelo de Diseño.

Métodos Científicos utilizados en la investigación.

Métodos Teóricos:

- **Analítico–Sintético:** Este método ha servido para analizar y comprender la documentación relacionada con el tema de la investigación, permitiendo extraer los elementos teóricos relacionados con el objeto de estudio y su síntesis en este trabajo.
- **Análisis histórico–lógico:** Este método ha ayudado a entender el surgimiento y la evolución del tema de la investigación, así como otras temáticas estrechamente relacionadas con la misma a lo largo de la historia de la Informática.
- **Modelación:** Mediante este método se ha podido modelar cada artefacto que genera el proceso de desarrollo a través de UML. Toda esta modelación ayuda a descubrir y estudiar nuevas cualidades y relaciones del objeto de estudio.

Métodos Empíricos:

- **Análisis de Documentos:** Este método ha sido de gran ayuda porque a partir del estudio de un grupo de documentos que exponen en detalle el flujo que siguen los procesos que representan el objeto de la investigación, ha sido más fácil la comprensión de los mismos.

CAPÍTULO # 1: FUNDAMENTACIÓN TEÓRICA.

1.1 Introducción.

En este capítulo se desarrollarán los aspectos y conceptos generales relacionados con el tema de la simulación en general, se hará una breve descripción del estado del arte del tema a tratar, así como tecnologías y tendencias actuales que se usan para el desarrollo de productos similares al que se desea modelar. Se pretende además dejar fundamentadas las bases teóricas para un exhaustivo análisis del tema a desarrollar y también se explicarán las herramientas seleccionadas para dar solución a la problemática propuesta.

1.2 Estado del Arte.

Desde la antigüedad el ser humano ha intentado conseguir un mecanismo mediante el cual sea capaz de simular la realidad que le rodea. Los avances que se han producido en el diseño, desarrollo y construcción de máquinas que ayudan al hombre a simular la realidad, han pasado por distintas etapas. Desde los primeros mecanismos que intentaban imitar el comportamiento de los animales que datan desde el renacimiento, se ha llegado hasta los actuales ordenadores capaces de simular sistemas complejos; pero el uso moderno de la palabra simulación data de 1940, cuando los científicos Von Neumann y Stanislaw Ulam que trabajaban en el proyecto Monte Carlo, durante la Segunda Guerra Mundial, resolvieron problemas de reacciones nucleares cuya solución experimental resultaba muy cara y el análisis matemático demasiado complejo.

Durante todo este proceso de desarrollo, se han ido aprendiendo, mejorando y perfeccionando las técnicas que permiten representar la realidad, llegando a aplicarse en la actualidad complejos desarrollos matemáticos para construir modelos que representan una parte del mundo que nos rodea. Todo ello ha sido posible con la ayuda de las computadoras, y por supuesto, debido al desarrollo que se ha alcanzado en su diseño y construcción, haciéndolos más rápidos y con más potencia de cálculo por unidad de tiempo.

La simulación comprende la creación de un escenario artificial y el análisis del mismo para detectar los problemas que presenta el sistema operacional representado. Es una herramienta indispensable para resolver muchos de los problemas que se presentan en la realidad. Esta es empleada para representar y

analizar distintas alternativas de un sistema productivo y finalmente ayudarnos a decidir cuál es la mejor de todas [2].

Desde la década de los 60 se han desarrollado y aplicado Simuladores como el POWERFACTS de la Dow Chemical, GPSS II, CSL y CHIPS de IBM; GASP y GPS de la Corporación del Acero de USA; CHEOPS de la Compañía Petrolera Shell, Flexible FLOWSHEET de la Corporación Kellogg, PEDLAN de la Compañía Petrolera MOBIL. También en esos años en el sector académico de Canadá y USA se crearon el PACER y el SPEEDUP. Algunos de los Simuladores mencionados dieron lugar a nuevas versiones o nuevos desarrollos. Muchos de ellos han sido utilizados con fines académicos en Cuba, por ejemplo en Centro Universitario José Antonio Echeverría se han utilizado simuladores como el GEMCS, GASP II, CHEMCAD, HYSYS, SUGARS, ASPEN PLUS y otros [1].

La crisis del petróleo en la década del 70, obligó a la industria en general a enfocar sus esfuerzos a lograr mejores niveles de eficiencia, iniciándose así una etapa de desarrollo tecnológico encaminada a mejorar los procesos existentes y en algunos casos a diseñar otros, nuevos y más eficaces. Este fue el impulso necesario para motivar el desarrollo de la simulación de procesos.

El comienzo fue lento y se dio de forma conceptual, experimental y académica en algunas compañías y universidades de Estados Unidos, Canadá y Europa. Para 1975 ya se habían desarrollado algunos simuladores con algunas aplicaciones industriales reducidas. En 1980 empezaron a surgir compañías desarrolladoras de software, que se dedicaron a crear algunos paquetes de simulación para su comercialización [3]. Pero tenían la desventaja de que las entradas y salidas de datos eran muy rígidas y se presentaban en forma de listados o tablas de difícil interpretación y comprensión. A finales de los años 80 se inició el desarrollo de paquetes de simulación interactivos y su comercialización marcó el comienzo de un uso más intensivo y generalizado en la industria y en las universidades. Entre 1991 y 1995 se inicia la comercialización de paquetes de simulación dinámica y de integración de energía.

En Cuba se han desarrollado varios simuladores y se han llegado a aplicar en procesos completos o subprocesos en diferentes ramas de la industria. En el Instituto Cubano de Investigaciones de los Derivados de la Caña de Azúcar (ICIDCA) se dispone de un Simulador, orientado a ecuaciones del proceso azucarero y de refinación de alcohol de caña llamado SIMFAD 3.0, resultado de todo un amplio

trabajo anterior de modelación matemática y simulación de equipos y subprocesos de la industria azucarera. En la Universidad Central de Las Villas (UCLV) también se desarrolló un Simulador de procesos tecnológicos, pero no aparecen referenciadas aplicaciones en la industria cubana. Otro Simulador desarrollado en el MINAZ (Villa clara) es el denominado AGE de carácter determinístico y que contiene recomendaciones de cómo actuar en función de los resultados. Pero al igual que otro paquete de simulación de la UCLV está basado en métodos de cálculos simplificados o rápidos que no permiten obtener todos los resultados importantes y necesarios que pueden dar los Simuladores que usan modelos más precisos y sin simplificaciones innecesarias si se dispone de computadoras [1].

En el Centro de Investigación y Desarrollo de Simuladores (SIMPRO), perteneciente a las FAR, en coordinación con la industria militar cubana y con otras entidades de la economía nacional, han desarrollado a partir de la década del 90 decenas de simuladores virtuales diseminados por unidades de las FAR, escuelas de automovilismo del Ministerio de Transporte, y de patrullas del Ministerio del Interior. Unos permiten el entrenamiento para conducción de tanques, carros blindados y de distintos vehículos; otros, el tiro coheteril, de infantería, artillería y de tanques; algunos, los más complejos, capacitan a pilotos de aeronaves de transporte y de combate [4].

Existe una gran variedad de simuladores comerciales de procesos. Algunos de los cuales son poderosas herramientas de cálculo en procesos industriales, con enormes bases de datos y una fuente de respaldo de bibliotecas para cálculo de equipos y bibliotecas de modelos para cálculos termodinámicos, que dan a un simulador la ventaja de una gran versatilidad. Algunos de estos simuladores de procesos de propósitos generales son:

Aspen Plus y SpeedUp (de Aspen Technology, USA), Pro II (de Simulations Sciences, USA), Hysys (de Hyprotech, Canadá), Chemcad (de Chemstations, USA). Aspen Plus, Pro II y Chemcad son simuladores de procesos en estado estable, SpeedUp es un simulador de procesos dinámico y Hysys es útil para dos tipos de simulación.

Un aspecto importante de los simuladores de procesos, es la disponibilidad de propiedades termodinámicas y de transporte de las corrientes del proceso, estas propiedades son fundamentales para

realizar los balances de materia y energía al grado de que si se tienen datos confiables o buenas correlaciones para las propiedades, entonces los resultados de las simulaciones serán confiables también.

Los simuladores modernos deben permitir la selección de los modelos de propiedades termodinámicas adecuados para la naturaleza de los componentes químicos, estado de agregación y condiciones de operación. Las corrientes del proceso pueden ser:

- Gases a baja y alta presión.
- Soluciones líquidas con componentes polares, no polares y electrolitos a baja y alta presión.
- Sólidos en suspensión o finamente divididos.

Estas características son las que permiten que un simulador de procesos pueda representar una gran variedad de plantas industriales. En un estudio de simulación donde se cuente o no con el apoyo de una herramienta de simulación previamente elaborada, estos procesos son representados a través de varios diagramas como pueden ser el Diagrama de Simulación (DS), Diagrama de Procesos (DP), y el Diagrama de Flujo de Información (DFI).

A partir de la información del proceso que se está estudiando, generada a través del Diagrama de Procesos obtenido dentro del estudio de simulación, en la Fase de Formulación del Modelo. Se obtiene el Diagrama de Simulación, el cuál es básicamente igual al Diagrama de Procesos, pero en él aparecen los equipos virtuales, tales como mezcladores y divisores de corrientes. Con estos datos se construye el Diagrama de Flujo de Información de igual configuración que el anterior en donde los equipos de proceso se transforman en nodos unidos por corrientes de información (antes estas eran corrientes de procesos). Estas corrientes de información son líneas orientadas y flexibles, “mensajeras” del inventario másico energético que “fluye” entre nodos. A través de este diagrama el especialista obtiene el orden de cálculo de los diferentes componentes del mismo, estos representan a los equipos que intervienen en el proceso en la realidad. Existen dos tipos de DFI: DFI sin ciclos múltiples y con la presencia de estos. Este último se encuentra cuando existen subgrafos cíclicos dentro del diagrama.

Figura 1: Diagrama de Proceso.

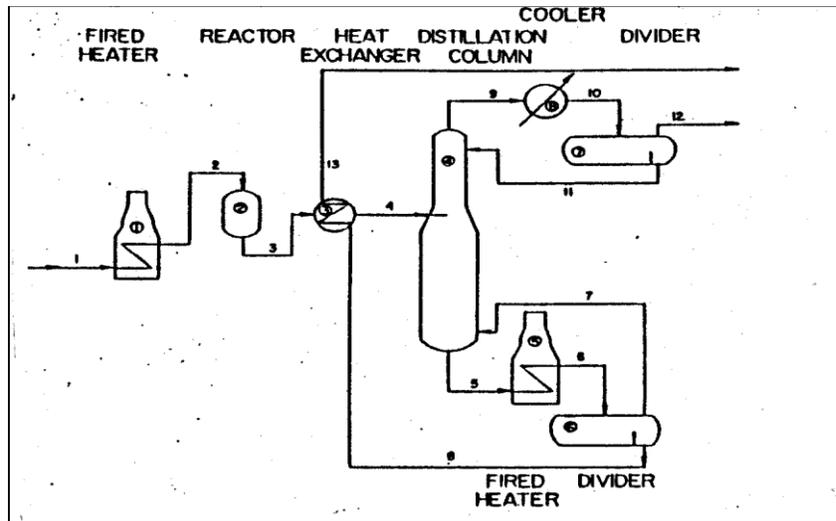


Figura 2: Diagrama de Simulación.

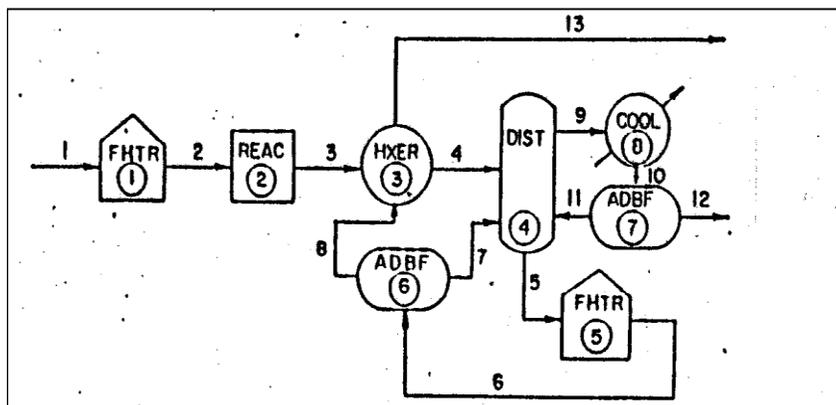
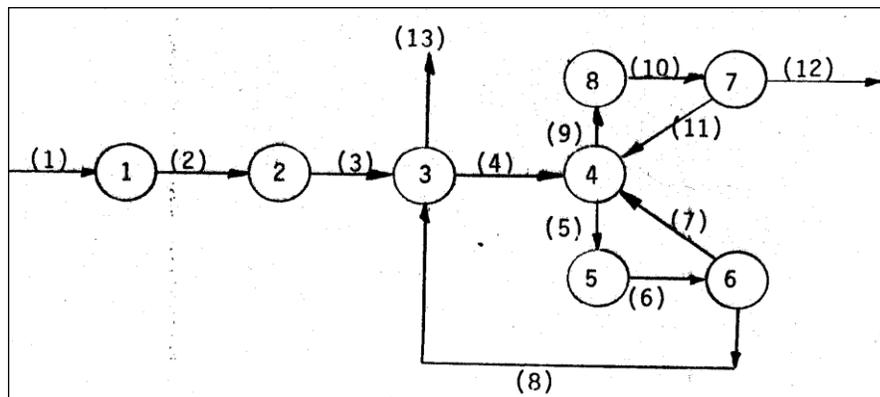


Figura 3 Diagrama de Flujo de Información.



En la Figura 1 (Diagrama de Proceso) se incluye sólo las corrientes involucradas en el balance másico-energético que se debe concretar; en la Figura 2 (Diagrama de simulación) se han incorporado dos divisores de corrientes y abreviado el nombre de los equipos que componen el diagrama, estas abreviaturas suelen ser únicas para cada componente dentro del diagrama. En la Figura 3 se ha generado el diagrama de flujo de información (DFI), en donde los equipos reales y virtuales se tratan como nodos, hacia donde llegan y salen corrientes de información, que son las líneas “mensajeras” del inventario de masa y energía entre ellos.

Las herramientas de simulación permiten la realización de estos diagramas de una manera más fácil e intuitiva, logrando una simplificación de este proceso ya que el mismo simulador se encarga de llevar a cabo la creación de manera interna del DFI y de la resolución del mismo.

Para introducir información al diagrama de flujo de información del proceso a un simulador usualmente se siguen los pasos básicos que a continuación se enlistan (el orden de los pasos y la manera en que estos pasos son cubiertos puede variar según el simulador que se esté utilizando):

- Aclarar el sistema de unidades en se desea trabajar.
- Definir los equipos o componentes que conformarán el DFI y las corrientes entre estos. Esto puede hacerse de diversas formas de acuerdo al simulador utilizando diferentes funcionalidades relacionadas con el diseño del DFI.

- Especificar la totalidad de los compuestos que van a intervenir en el proceso. Lo normal será que el usuario, o lo seleccione de un menú; en ocasiones es necesario que el usuario lo defina, cuando el compuesto no forma parte del banco de datos del simulador.
- Seleccionar modelos termodinámicos apropiados al problema a resolver para el cálculo de las propiedades. La selección se hace generalmente de un menú de opciones.
- Especificar régimen de flujo y condiciones termodinámicas de las corrientes de entrada al proceso. Es necesario aclarar en cada corriente los flujos (flujo de cada componente o flujo total y composición de todos los componentes menos uno) y dos propiedades de la corriente que frecuentemente son presión y temperatura.
- Especificar las condiciones de operación de los equipos o componentes en el diagrama de flujo de información. Este paso también varía significativamente entre un simulador y otro de acuerdo a la estructura del mismo.

Es por este motivo que el diseño y modelado del DFI es un importante proceso, mediante el cual un especialista o ingeniero realiza una síntesis virtual del proceso, representando los equipos que intervienen en la industria en la realidad y el flujo entre los mismos.

1.3 Fundamentación del Tema.

En esta sección se pretenden explicar algunos conceptos relacionados con el objeto de estudio y el dominio del problema.

1.3.1 Definición de Simulación.

A continuación se describen algunas de las definiciones de simulación:

Thomas H. Naylor la define así:

"Simulación es una técnica numérica para producir experimentos en una computadora digital. Estos experimentos comprenden criterios de 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 períodos de tiempo."

La definición anterior está en un sentido muy amplio. En sentido más estricto H. Maisel y G. Gnugnoli, la definen como:

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

Otros estudiosos del tema como Robert E, Shannon plantean que:

"Simulación es el proceso de diseñar y desarrollar un modelo computarizado de un sistema o proceso y concluir experimentos con este modelo con el propósito de entender el comportamiento del sistema o evaluar varias estrategias con las cuales se puede operar el sistema."

Las definiciones anteriores no especifican si los sistemas modelados son continuos o discretos.

1.3.2 Aplicaciones de la simulación de procesos.

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

- 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 procesos para reducir el consumo de energía.
- Análisis de condiciones críticas de operación.
- Transformación de un proceso para desarrollar otras materias primas o productos.
- 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.

La variedad de aplicaciones de los simuladores de procesos es muy grande, anteriormente solo los utilizaban los ingenieros que diseñaban procesos y ahora también manejan simuladores los ingenieros ambientales, ingenieros de procesos y hasta ingenieros de planta en su lugar de trabajo; ingenieros con poca o ninguna instrucción de programación pueden modelar procesos complejos utilizando estas herramientas.

En principio, la simulación de procesos puede ser útil en todas las etapas del desarrollo de un proyecto industrial. En estas puede surgir la necesidad de realizar simulaciones con diferentes niveles de sofisticación. Ejemplo de estas etapas son:

1. **Investigación y desarrollo.** Una simulación sencilla se puede aplicar para probar la factibilidad técnica y económica del proyecto.
2. **Etapla crítica en la toma de decisiones.** Se prueban diferentes alternativas de tamaño y localización de la planta industrial y se determinan las condiciones de operación óptimas.
3. **Planta piloto.** Simulación con modelos más sofisticados para obtener mejores estimaciones de las condiciones de operación a escala industrial.
4. **Diseño.** La simulación proporciona todos los datos de los procesos requeridos para el diseño detallado de los diferentes equipos.
5. **Simulación de plantas existentes.** Puede ser útil cuando es necesario cambiar las condiciones de operación, o cuando se quiere sustituir materias primas.

Hay tres tipos de problemas que se pueden resolver con la simulación de procesos.

1. En la simulación de un problema (análisis), deben especificarse las variables asociadas con las corrientes de alimentación y las variables de diseño de los módulos unitarios. Las incógnitas son variables asociadas con todas las corrientes adicionales y con las corrientes de productos que

salen del proceso. Es decir que se conocen las alimentaciones y los parámetros de los equipos, y las incógnitas son las condiciones de las corrientes de salida.

2. El problema de diseño es similar al problema de simulación, excepto en algunas de las variables de las corrientes (regularmente solo restricciones de igualdad). El número de restricciones es igual al número de variables de diseño sin especificar. En el diseño se conocen las alimentaciones y las condiciones principales de las corrientes de salida, y las incógnitas son las dimensiones y especificaciones de algunos parámetros de los equipos.
3. En el problema de optimización, las variables asociadas con las corrientes de alimentación y las variables de diseño pueden no estar especificadas, entonces es necesario agregar una función de costo al modelo. Las variables sin especificar se determinan de modo que se minimiza la función objetivo. En este caso, se pueden especificar restricciones de igualdad y desigualdad.

1.3.3 Tipos de Simulación.

Algunos de los tipos de Simulación aplicados en el desarrollo de Simuladores en la actualidad son:

1. **Simulación Discreta:** modelación de un sistema por medio de una representación en la cual el estado de las variables cambian instantáneamente en instantes de tiempo separados. (En términos matemáticos el sistema solo puede cambiar en instantes de tiempo contables). [5]
2. **Simulación Continua:** modelación de un sistema por medio de una representación en la cual las variables de estado cambian continuamente en el tiempo. Típicamente, los modelos de simulación continua involucran ecuaciones diferenciales que determinan las relaciones de las tasas de cambios de las variables de estado en el tiempo. [6]
3. **Simulación Combinada o Mixta (Combina la simulación Discreta y la Continua):** modelación de un sistema por medio de una representación en la cual unas variables de estado cambian continuamente con respecto al tiempo y otras cambian instantáneamente en instantes de tiempo separados. Es una simulación en la cual interactúan variables de estado discretas y continuas. Existen tres tipos de interacciones entre las variables de estado de este tipo de simulaciones:
 - Un evento discreto puede causar un cambio discreto en el valor de una variable de estado continua.

- Un evento discreto puede causar que la relación que gobierna una variable de estado continua cambie en un instante de tiempo en particular.
 - Una variable de estado continua de punto de partida puede causar que un evento discreto ocurra, o sea, programado. [6]
4. **Simulación Determinística y/o Estocástica:** una simulación determinística es aquella que utiliza únicamente datos de entrada determinísticos, utiliza datos de entrada conocidos y determinados (No hay incertidumbre). En cambio un modelo de simulación estocástico incorpora algunos datos de entrada azarosos al utilizar distribuciones de probabilidad (Existe incertidumbre).
 5. **Simulación con Orientación a eventos:** modelado con un enfoque hacia los eventos, en el cual la lógica del modelo gira alrededor de los eventos que ocurren instante a instante, registrando el estado de todos los eventos, entidades, atributos y variables del modelo en todo momento.
 6. **Simulación con Orientación hacia procesos:** modelado con un enfoque de procesos, en el cual la lógica del modelo gira alrededor de los procesos que deben seguir las entidades. En cierta forma, es un modelado basado en un esquema de flujo de procesos, el cual se hace mediante un seguimiento a una entidad a través de la secuencia de procesos que debe seguir.

1.3.4 Etapas para realizar un estudio de simulación.

Se ha escrito mucho acerca de los pasos necesarios para realizar un estudio de simulación. Sin embargo, la mayoría de los autores opinan que los pasos necesarios para llevar a cabo un experimento de simulación son:

- **Definición del sistema.** Para tener una definición exacta del sistema que se desea simular, es necesario hacer primeramente un análisis preliminar del mismo, con el fin de determinar la interacción del sistema con otros, las restricciones del sistema, las variables que interactúan dentro del sistema y sus interrelaciones, las medidas de efectividad que se van a utilizar para definir y estudiar el sistema y los resultados que se esperan obtener el estudio.
- **Formulación del modelo.** Una vez que están definidos con exactitud los resultados que se esperan obtener del estudio, el siguiente paso es definir y construir el modelo con el cual se obtendrán los resultados deseados. En la formulación del modelo es necesario definir todas las variables que forman parte de él, sus relaciones lógicas y los diagramas de flujo que describan en

forma completa el modelo.

- **Colección de datos.** Es posible que la facilidad de obtención de algunos datos o la dificultad de conseguir otros, pueda influenciar en el desarrollo y formulación del modelo. Por consiguiente, es muy importante que se definan con claridad y exactitud los datos que el modelo va a necesitar para producir los resultados deseados. Normalmente, la información requerida por un modelo se puede obtener de registros contables, de órdenes de trabajo o de compra, de opiniones de expertos y si no queda otra alternativa, por medio de la experimentación.
- **Implementación del modelo en la computadora.** Con el modelo definido, el siguiente paso es decir si se utiliza algún lenguaje de programación o algún paquete de simulación de propósito general como gpcss, simula o simscript, para procesarlo en la computadora y obtener los resultados.
- **Validación.** Una de las principales etapas de un estudio de simulaciones la validación. a través de esta etapa es posible detallar deficiencias en la formulación del modelo o de los datos alimentados al modelo. La formas más comunes de validar un modelo son:
 1. La opinión de expertos sobre resultados de la simulación o la aplicación de sistemas expertos.
 2. La exactitud con que se predicen los datos históricos.
 3. La exactitud en la predicción del futuro.
 4. La comprobación de falla del modelo de simulación al utilizar datos que hacen fallar al sistema real.
 5. La aceptación y confianza en el modelo de la persona que hará uso de los resultados que arroje el experimento de la simulación.
- **Experimentación.** La experimentación con el modelo se realiza después que este ha sido validado. La experimentación consiste en generar los datos deseados y en realizar el análisis de sensibilidad de los índices requeridos.
- **Interpretación.** En esta etapa del estudio, se interpretan los resultados que arroja la simulación y en base a esto se toma una decisión. Es obvio que los resultados que se obtienen de un estudio de simulación ayudan a soportar decisiones del tipo semiestructurado, es decir, la computadora en si no toma la decisión, sino que la información que proporciona ayuda a tomar mejores decisiones

de los especialistas que trabajan con el modelo y por consiguiente a obtener mejores resultados.

- **Documentación.** Dos tipos de documentación son requeridos para hacer un mejor uso del modelo de simulación. La primera se refiere a la documentación de tipo técnico, es decir, a la documentación que el departamento de Procesamiento de Datos debe tener del modelo. La segunda se refiere al manual del usuario, con el cual se facilita la interacción y el uso del modelo desarrollado, a través de una terminal de computadora.

1.3.5 Aceptación de la Simulación.

Durante las últimas dos o tres décadas consistentemente se ha reportado que la simulación es la herramienta de investigación operacional más popular [7]:

- Con respecto al valor práctico de aplicación que poseen los métodos de investigación de operaciones, los graduados de la maestría en ciencias M.S. del Departamento de Investigación de Operaciones de la Universidad Western Reserve, generaron un ranking. Los primeros 4 métodos más utilizados eran análisis estadístico, análisis de sistemas, sistemas de información. En quinto lugar a parece la simulación, por encima de programación lineal y teoría de colas.[8]
- A partir de un estudio realizado en 137 firmas, en el cual se les pedía que seleccionaran los métodos de investigación de operaciones que utilizaban. El resultado del estudio reportó que el 93% usaba análisis estadístico y en un 84% simulación. Nuevamente la simulación era más usada que métodos como programación lineal, PERT/CPM, teoría de inventarios y programación no lineal. [9]
- Según una encuesta realizada a los miembros de la División de Investigación de Operaciones del American Institute of Industrial Engineer, la simulación ocupa el primer lugar bajo la categoría de herramientas útiles y de interés, y el segundo lugar en la categoría de familiaridad. [10]
- Según los siguientes artículos, en términos de utilización de métodos por practicantes y grandes corporaciones, el análisis estadístico ocupa el primer lugar y la simulación el segundo lugar:

Harpell, J.L., M.S. Lane, y A.H. Masour, (1989), Operation Research in practice: A Longitudinal Study, Interfaces, vol19, pp 65-74

Harpell, J.L., M.S. Lane, y A.H. Masour, (1993) Operation Research Techniques: A Longitudinal Study, Interfaces, vol.23, pp 63-68

En un artículo publicado por C. L. Morgan en el cual se revisan muchos estudios como los anteriores, se reporta que existe un frecuente y fuerte uso de la simulación. Aún en aquellas industrias en donde se reporta un muy bajo uso de herramientas de investigación de operaciones, la simulación ocupa el primer lugar en uso.

Todos los resultados anteriores se deben a que desde un punto de vista general, el análisis y simulación de procesos presenta las ventajas que se señalan a continuación.

- Experimentación económica. Es posible estudiar procesos existentes de una forma más rápida, económica y completa que en la planta real. La simulación puede aumentar o reducir de una forma análoga a como una cámara cinematográfica acelera o retarda las imágenes; de esta forma se puede observar más fácilmente la operación del sistema.
- Extrapolación. Con un modelo matemático adecuado se puede ensayar intervalos extremos de las condiciones de operación, que pueden ser impracticables o imposibles de realizar en una planta real. También es posible establecer características de funcionamiento.
- Estudio de conmutabilidad y evaluación de otros planes de actuación. Se pueden introducir nuevos factores o elementos de un sistema y suprimir otros antiguos al examinar el sistema con el fin de ver si estas modificaciones son compatibles. La simulación permite comparar distintos diseños y procesos que todavía no están en operación y ensayar hipótesis sobre sistemas o procesos antes de llevarlos a la práctica.
- Repetición de experimentos. La simulación permite estudiar el efecto de la modificación de las variables y parámetros con resultados reproducibles. En el modelo matemático se puede introducir o retirar a voluntad un error, lo cual no es posible en una planta real.
- Control de cálculo. La simulación constituye una importante ayuda material para el estudio de los sistemas de control con lazos abiertos y cerrados.
- Ensayo de sensibilidad. Se puede ensayar la sensibilidad de los parámetros de costes y los parámetros básicos del sistema; por ejemplo un incremento del 10% de la velocidad de alimentación podrá tener, según los casos, un efecto mínimo o muy importante sobre el

funcionamiento de la instalación.

- Estudio de la estabilidad del sistema. Se puede examinar la estabilidad del sistema y subsistemas frente a diferentes perturbaciones.

1.4 Metodologías de desarrollo de Software.

Una metodología es el conjunto de filosofías, fases, procedimientos, reglas, técnicas, herramientas, documentación y aspectos de formación para los desarrolladores de Sistemas de Información. Proporcionan guías para estimar costos, políticas y procedimientos para garantizar la calidad del software y las descripciones de los roles y responsabilidades de cada uno. En un proyecto de desarrollo de software la metodología define: Quién debe hacer Qué, Cuándo y Cómo debe hacerlo.

En la actualidad existen dos grandes grupos de metodologías. Por una parte, se encuentran las más tradicionales que se centran especialmente en el control del proceso, estableciendo rigurosamente las actividades involucradas, los artefactos que se deben producir y las herramientas y notaciones que se usarán. Ejemplo de esta es RUP (Proceso Unificado de Desarrollo).

Por otra parte, existen las metodologías ágiles, las cuales dan mayor valor a individuo, a la colaboración con el cliente y al desarrollo incremental del software con iteraciones muy cortas. Este enfoque está mostrando su efectividad en proyectos con requisitos cambiantes y cuando se exige reducir drásticamente los tiempos de desarrollo pero manteniendo una alta calidad.

Una de las cualidades más destacables en una metodología ágil es su sencillez, tanto en su aprendizaje como en su aplicación, reduciéndose así los costos de implantación en un equipo de desarrollo. Ejemplo de esta metodología es XP (del inglés eXtreme Programing o Programación Extrema).

1.4.1 Rational Unified Process (RUP).

La metodología escogida por el arquitecto del equipo de trabajo que participa en el diseño del subsistema es la metodología RUP, es una metodología general, ya que está diseñada para adaptarse al contexto y necesidades de cada organización, principalmente con grupos de producción grandes, donde el ciclo de vida está dirigido por casos de uso que reflejan lo que los usuarios futuros necesitan y desean, estos se representan a través de los requerimientos. Los casos de uso guían a partir de aquí, todo el proceso de

desarrollo, donde la arquitectura del software es esencial y muestra una visión común del sistema, en la que el equipo de proyecto y los usuarios, deben estar de acuerdo, la arquitectura muestra la visión común del sistema y se expresa mediante modelos.

También esta metodología está basada en normas provenientes de estándares seguidos por un entorno de desarrollo y proceso más controlado, con numerosas políticas y normas, mientras que las ágiles están basadas en heurísticas provenientes de prácticas de producción de código y procesos menos controlados. Además de que para aplicar una metodología de desarrollo ágil hay que tener una muy buena comunicación con el cliente, cuestión con la cual no cumple el proyecto para el cual estamos diseñando este subsistema.

RUP organiza el proceso de desarrollo, en ciclos de iteración, obteniendo un producto final, al concluir cada ciclo y utilizando en cada uno artefactos, para lograr los hitos de cada fase, con el propósito de obtener un incremento en el proceso de desarrollo, siendo por este motivo una de las metodologías más importante para alcanzar un grado de calidad y de certificación en el desarrollo del software.

RUP toma en cuenta las mejores prácticas en el modelo de desarrollo de software en particular las siguientes:

- Desarrollo de software en forma iterativa (repite una acción).
- Manejo de requerimientos.
- Utiliza arquitectura basada en componentes.
- Modela el software visualmente (con UML).
- Verifica la calidad del software.
- Controla los cambios.

El Proceso Unificado de Rational (RUP) consta de cuatro fases o etapas:

- Inicio: Se describe el negocio y se delimita el proyecto describiendo sus alcances con la identificación de los casos de uso del sistema, en otras palabras el objetivo en esta etapa es determinar la visión del proyecto, es la puesta en marcha.

- **Elaboración:** Se define la arquitectura del sistema y se obtiene una aplicación ejecutable que responde a los casos de uso que la comprometen. A pesar de que se desarrolla a profundidad una parte del sistema, las decisiones sobre la arquitectura se hacen sobre la base de la comprensión del sistema completo y los requerimientos (funcionales y no funcionales) identificados de acuerdo al alcance definido.
- **Construcción:** Se obtiene un producto listo para su utilización que está documentado y tiene un manual de usuario. Se obtiene uno o varios release del producto que han pasado las pruebas. Se ponen estos release a consideración de un subconjunto de usuarios.
- **Transición:** El release ya está listo para su instalación en las condiciones reales. Puede implicar reparación de errores.[11]

Dentro de los Flujos de Trabajo propuestos por esta metodología, este trabajo solo desarrollará los siguientes flujos de trabajo:

- **Requerimientos**

La finalidad de este Flujo de Trabajo es:

- Establecer y mantener un acuerdo con los clientes y otros interesados acerca de lo que debe hacer el sistema.
- Proporcionar desarrolladores de sistema con un buen conocimiento de los requisitos del sistema.
- Definir los límites del sistema (delimitarlo).
- Proporcionar una base para planificar el contenido técnico de las iteraciones.
- Proporcionar una base para la estimación del coste y del tiempo en que desarrollar el sistema.
- Definir una interfaz de usuario para el sistema, centrándose en las necesidades y los objetivos de los usuarios.

Para alcanzar esos objetivos, es importante, en primer lugar, comprender la definición y el ámbito del problema que se intenta resolver con el sistema. Los interesados se identifican y las solicitudes de los interesados se obtienen, se reúnen y se analizan.

A partir de ahí se desarrollan los productos de trabajo de los requisitos para describir completamente el sistema (qué va a hacer el sistema) en un esfuerzo que percibe a todos los

interesados, incluidos los clientes y los usuarios potenciales, como fuentes importantes de información (además de los requisitos del sistema).

- **Análisis y Diseño**

La finalidad de este Flujo de Trabajo es:

- Transformar los requisitos en un diseño del sistema en creación.
- Evolucionar una arquitectura sólida para el sistema.

Adaptar el diseño para que se ajuste al entorno de implementación, con un diseño pensado para el rendimiento.

1.4.2 Lenguaje de Modelado.

Lenguaje Unificado de Modelado (UML, Unified Modeling Language), es un lenguaje de propósito general, que pueden usar todos los modeladores, es uno de los más conocidos y utilizados en la actualidad. UML es utilizado para visualizar, especificar, construir y documentar los artefactos de un sistema que involucra una gran cantidad de software [12]. UML es un lenguaje estándar que permite describir el sistema en forma de modelo, incluyendo aspectos conceptuales tales como procesos de negocios y funciones del sistema, y aspectos concretos como componentes de software reutilizables y expresiones de lenguajes de programación.

UML incluye todos los conceptos que se consideran necesarios para utilizar un proceso moderno iterativo, buscando construir una arquitectura para resolver requisitos dirigidos por casos de uso. Se utiliza para modelar el subsistema propuesto, para detallar los artefactos y para documentar y construir el mismo. Se puede aplicar en una gran variedad de formas para soportar una metodología de desarrollo de software (tal como el Proceso Unificado de Rational), pero no especifica en sí mismo qué metodología o proceso usar.

UML se puede usar para modelar distintos tipos de sistemas: sistemas de software, sistemas de hardware, y organizaciones del mundo real. UML ofrece diferentes tipos de diagramas [13]:

Diagramas de estructura estática

- Diagrama de clases
- Diagrama de objetos
- Diagrama de casos de uso

Diagramas de comportamiento

- Diagramas de interacción (secuencia y colaboración)
- Diagrama de estados
- Diagrama de actividad

Diagramas de implementación

- Diagrama de componentes
- Diagrama de despliegue

1.4.3 Descripción de la Herramienta Case para el Modelado.

Visual Paradigm para UML es una de las herramientas UML CASE fácil de usar y bastante completa, con soporte multiplataforma. Proporciona facilidades de interoperabilidad con otras aplicaciones. Fue creada para el ciclo vital completo del desarrollo del software, tratando de automatizarlo y acelerarlo dentro de sus posibilidades, permitiendo la captura de requisitos, análisis, diseño e implementación. Visual Paradigm-UML también proporciona características tales como la generación del código, ingeniería inversa y generación de informes. Tiene la capacidad de crear el esquema de clases a partir de una base de datos y crear la definición de base de datos a partir del esquema de clases. Permite invertir código fuente de programas, archivos ejecutables y binarios en modelos UML al instante, creando de manera simple toda la documentación. Está diseñada para usuarios interesados en sistemas de software de gran escala con el uso del modelado orientado a objeto, además apoya los estándares más recientes de las notaciones de Java, C# y otros lenguajes, así como también de UML. Incorpora el soporte para trabajo en equipo, que permite que varios desarrolladores trabajen a la vez en el mismo diagrama y vean en tiempo real los cambios hechos por sus compañeros.

1.4.4 Patrones de casos de uso.

Cada patrón describe un problema que ocurre una y otra vez en nuestro entorno, para describir después el núcleo de la solución a ese problema, de tal manera, que esa solución pueda ser usada más de un millón de veces sin hacerlo siquiera dos veces de la misma forma [13].

Los patrones han permitido en diferentes áreas del conocimiento humano rehusar la esencia de la solución de un problema al enfrentar nuevos problemas similares. Es así que los patrones constituyen una especie de mecanismo de registro y concentración de experticia. En la práctica se hace difícil, en algunas ocasiones, escoger qué patrón resulta más apropiado para resolver el problema.

La experiencia en la utilización de casos de uso ha evolucionado en un conjunto de patrones que permiten con más precisión reflejar los requisitos reales, haciendo más fácil el trabajo con los sistemas, y mucho más simple su mantenimiento.

Estas técnicas han mostrado ser una de las soluciones adoptadas comúnmente por los desarrolladores de software. Se presentan a modo de herramientas que permiten resolver algunos de los problemas de una forma ágil y sistemática. Estos patrones se enfocan hacia el diseño y las técnicas utilizadas en modelos de alta calidad, y no en cómo modelar usos específicos. Utilizando estos patrones, arquitectos, analistas, e ingenieros en general, pueden lograr mejores resultados de forma más rápida.

Algunos de los patrones de casos de uso son los siguientes:

- Reglas de negocio
- Concordancia (Commonality)
- Componente jerárquico (Component hierarchy)
- Extensión concreta o Inclusión
- CRUD (Creating, Reading, Updating, Deleting)
- Caso de uso grande (Large Use case)
- Sistema de Capas
- Múltiples actores
- Servicio opcional

- Vistas ortogonales
- Secuencia de casos de uso.

De estos mencionados sólo se describirán cinco patrones: Reglas de negocio, Concordancia, CRUD, Sistema de capas y Múltiples actores.

Reglas de Negocio

Se basan en la extracción de información originada de las políticas, reglas y regulaciones del negocio de la descripción del flujo y describe la información como una colección de reglas del negocio referenciadas a partir de las descripciones de los casos de uso.

- Definición estática:
Este patrón es aplicado a todos los casos de uso modelando los servicios que son afectados por las reglas del negocio definidas en la organización. Sin embargo, este patrón no influye en la estructura del modelo de casos de uso. Las reglas son descritas en un documento separado, referenciadas por las descripciones de los casos de usos relevantes. Este patrón es apropiado utilizarlo cuando no hay necesidad de cambiar dinámicamente las reglas del negocio mientras el sistema se esté utilizando.
- Modificación dinámica:
Este modelo del patrón contiene un caso de uso llamado Gestionar regla, que se encarga de crear, actualizar y eliminar las reglas del negocio (*Ver CRUD*). Este patrón es útil cuando la colección de reglas sea modificada dinámicamente, o sea, estas pueden ser modificadas mientras el sistema este corriendo.

Concordancia (Commonality)

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

- Reusabilidad:

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.

- **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.

- **Especialización:**

Otro patrón de concordancia que contiene casos de uso del mismo tipo. En este caso, estos son modelados como una especialización de casos de uso de tipo de uso común. Todas las acciones en estos casos de uso son heredadas por los casos de uso hijos, donde otras acciones serán adicionadas o acciones heredadas que serán especializadas. Este patrón es aplicable cuando la utilización de los casos de uso que han sido modelados son del mismo tipo, y este tipo debe hacerse visible en el modelo.

- **Reusabilidad interna:**

Si la subsecuencia de acciones es utilizada en diferentes lugares en un solo caso de uso, no existe la necesidad de extraer la subsecuencia dentro de un caso de uso separado. Además este debe ser descrito en una sub-sección separada en la descripción del caso de uso. Esta sub-sección será referenciada desde diferentes partes en la descripción del caso de uso donde las subsecuencias de acciones sean realizadas. Este patrón se utiliza cuando la subsecuencia común aparece en múltiples lugares en un mismo caso de uso.

CRUD (Creating, Reading, Updating, Deleting)

Este patrón se basa en la fusión de casos de uso simples para formar una unidad conceptual.

- **Completo:**

Este patrón consta de un caso de uso, llamado Información CRUD o Gestionar información, modela todas las operaciones que pueden ser realizadas sobre una parte de la información de un tipo específico, tales como creación, lectura, actualización y eliminación. Suele ser utilizado cuando todos los flujos contribuyen al mismo valor del negocio, y estos a su vez son cortos y simples.

- **Parcial:**

Este patrón alternativo modela una de las vías de los casos de uso como un caso de uso separado. Es preferiblemente utilizado cuando una de las alternativas de los casos de uso es más significativa, larga o más compleja que las otras.

Múltiples actores

- **Roles diferente:**

Captura la concordancia entre actores manteniendo roles separados. Consiste en un caso de uso y por lo menos dos actores. Es utilizado cuando dos actores juegan diferentes roles en un caso de uso, o sea, interactúan de forma diferente con el caso de uso.

- **Roles comunes:**

Puede suceder que los dos actores jueguen el mismo rol sobre el CU. Este rol es representado por otro actor, heredado por los actores que comparten este rol. Es aplicable cuando, desde el punto de vista del caso de uso, solo exista una entidad externa interactuando con cada una de las instancias del caso de uso.

1.5 Rol de Analista.

1.5.1 ¿Qué es un rol?

Un rol es una definición abstracta de un conjunto de actividades realizadas y de artefactos obtenidos. Los roles son realizados típicamente por un individuo, o un conjunto de individuos, trabajando juntos en equipo. Un miembro del equipo de proyecto cumple normalmente muchos roles. Los roles no son individuos; en lugar de ello, describen cómo los individuos se comportan en el negocio y qué responsabilidades tienen estos individuos. [14]

1.5.2 El rol de Analista.

En la disciplina de la ingeniería del software, un analista es aquel individuo que tiene como tarea analizar un problema y describirlo, con el propósito de darle solución mediante un sistema informático. Es imprescindible en cualquier organización, debido al abanico de destrezas que éste posee y los beneficios que produce. No sólo se encarga de estudiar la organización y desarrollar un sistema automatizado, la labor del analista es más que eso, es también la de asesorar, supervisar, recomendar y modificar procesos internos y algunas veces de modificar la estructura misma de la empresa, con el propósito de lograr los objetivos que se proponen.

El Analista de Sistema surge de la necesidad de recopilar, desglosar, catalogar y analizar información necesaria de una empresa para poder proponer nuevos métodos mejores o modificar los actuales para que así aumente el desempeño de los departamentos dentro de la organización. Un analista se vale de la información de entrada, los procesos modificadores y la información de salida, para así definir los procesos intermedios y poder entender con claridad a la organización. Todos estos flujos y procesos son estudiados sistemáticamente para poder determinar si son los adecuados, si se deben mejorar o si deben ser reemplazados por otros más idóneos.

Las cualidades que se esperan de un analista son esencialmente la capacidad de abstracción y de análisis, facilidad de comunicación que le permita relacionarse en forma significativa con diferentes personas diariamente, que tenga conocimientos básicos de usabilidad, pues cualquier sistema que no esté al servicio de los usuarios o diseñado pensado en él, no tiene mucho sentido.

1.6 Conclusiones

En este capítulo se abordaron los aspectos y conceptos generales relacionados con el tema de la simulación en general y la simulación de procesos en particular. Se realizó una reseña del estado del arte del tema a tratar, tecnologías y tendencias actuales que se utilizan internacionalmente para el Desarrollo de Software similares al que se propone modelar. Se hizo un estudio de algunas de los simuladores que se utilizan en la industria actual. Se dejaron sentadas las bases teóricas para un correcto análisis y se explicaron las herramientas seleccionadas para dar solución a la problemática propuesta.

CAPÍTULO # 2: PROPUESTA DE SOLUCIÓN.

2.1 Introducción.

En este capítulo se darán los primeros pasos para dar solución al problema planteado en este trabajo. En el mismo se encontrarán los principales resultados a obtener en el flujo de trabajo de requerimientos. Entre estos se encuentran los requerimientos funcionales y no funcionales de la solución propuesta, logrados a través de diferentes técnicas aplicadas en el proceso de levantamiento de requisitos. La descripción de los actores y casos de uso del sistema y los diagramas de caso de uso del sistema.

2.2 Captura de Requisitos.

Desde principios del desarrollo de aplicaciones informáticas, los desarrolladores de software se han encontrado con el problema de la identificación de requerimientos del sistema. Esto se debe a que es un proceso complejo, que no puede ser determinado matemáticamente, en el cual los datos son extraídos de las personas y estos varían en dependencia de las personas a las que se esté consultando y de su familiaridad y conocimiento del tema. Por este motivo se han desarrollado un grupo de diferentes técnicas que permiten hacer este proceso de forma más ágil y segura.

Para la elaboración de los diferentes procedimientos y artefactos se utilizaron las plantillas definidas por el Proceso Unificado del Software (RUP), ya que mediante estas se puede lograr una mayor organización y eficacia en el trabajo, queda constancia de todo lo que se hizo y se viabiliza el mismo.

Algunas de las técnicas utilizadas en este trabajo para la captura de requisitos son las siguientes:

- **Introspección:** esta técnica consiste en ponerse en el lugar del cliente y tratar de determinar cómo desearía éste el Sistema. Y en base a estas suposiciones comenzar a intercambiar con el cliente sobre las posibles funcionalidades encontradas.
- **Tormenta de Ideas:** técnica que puede ayudar a generar una gran variedad de vistas del problema y a formularlo de diferentes formas, sobre todo al comienzo del proceso de licitación, cuando los requisitos son todavía muy difusos. Mediante la misma se puede obtener una visión global de los requisitos que ayudaron a establecer un criterio común entre usuarios y desarrolladores sobre las funcionalidades del Sistema.

- **Discusiones:** Este tipo de entrevistas permite que el analista sostenga una discusión con el Cliente sobre su problemática para tratar de determinar en conjunto los requisitos del sistema.
- **Análisis de Protocolo:** Esta técnica parte de la idea de que el cliente cuenta con un modelo mental preexistente del sistema deseado y en base a este modelo ya existente se puede analizar y obtener los requisitos del sistema. Es posible su aplicación debido a que los clientes están familiarizados con las herramientas de simulación. Además de que poseen conocimiento sobre el tema de desarrollo de software.
- **VORD:** Esta técnica es utilizada para capturar requisitos en base a los puntos de vista de los diferentes implicados en el desarrollo del producto y en conjunto con el Cliente. Es utilizada gracias a que el subsistema propuesto va a ser desarrollado con el paradigma de programación orientados a objetos.

2.2.1 Especificación de Requisitos

La IEEE (Standard Glossary of Software Engineering Terminology) define un requerimiento cómo:

1. Condición o capacidad que necesita un usuario para resolver un problema o lograr un objetivo.
2. Condición o capacidad que tiene que ser alcanzada o poseída por un sistema o componente de un sistema para satisfacer un contrato, estándar, u otro documento impuesto formalmente.
3. Una representación documentada de una condición o capacidad como en 1 o 2.

Los requisitos generales del producto se obtienen del cliente. Estos requisitos comprenden necesidades de información y control, funcionalidad del producto y comportamiento, rendimiento general del producto, diseño, restricciones de interfaz y otras necesidades especiales. [15]

Todas las ideas que los clientes, usuarios y miembros del equipo de proyecto tengan acerca de lo que debe hacer el sistema, deben ser analizadas como candidatas a requisitos. Los requisitos se pueden clasificar en: funcionales y no funcionales.

Los requisitos funcionales pueden ser capturados también por medio de casos de uso. Estos a su vez, también capturan requisitos no funcionales específicos de un caso de uso determinado.

A continuación se identifican los principales requisitos funcionales que son las prestaciones que debe cumplir el subsistema y los requisitos no funcionales que son las características o cualidades que debe poseer el Subsistema de Interfaz Gráfica DFISim.

2.3 Requisitos Funcionales.

2.3.1 Funcionalidad Gestionar Módulo

RF 1: Dibujar Módulo.

Descripción El sistema permitirá al especialista a cargo de llevar a cabo el estudio de simulación la posibilidad de poder dibujar un nuevo módulo dentro del DFI que se esté diseñando.

Prioridad Crítico.

Estimado Final de la primera versión.

RF 2: Seleccionar Módulo.

Descripción El sistema permitirá seleccionar un módulo dentro del DFI que se esté diseñando.

Prioridad Crítico.

Estimado Final de la primera versión.

RF 3: Mover Módulo.

Descripción El sistema permitirá cambiar la posición de un módulo dentro del DFI que se esté diseñando.

Prioridad Crítico.

Estimado Final de la primera versión.

RF 6: Eliminar Módulo.

Descripción El sistema permitirá eliminar un módulo del DFI que se esté diseñando.

Prioridad Crítico.

Estimado Final de la primera versión.

RF 7: Redimensionar Módulo.

Descripción El sistema permitirá cambiar la dimensión un módulo del DFI que se esté diseñando.

Prioridad Secundario.

Estimado Final de la primera versión.

RF 8: Rotar Módulo.

Descripción El sistema permitirá rotar la imagen de un módulo del DFI que se esté diseñando.

Prioridad Secundario.

Estimado Final de la primera versión.

RF 9: Hacer espejo a Módulo.

Descripción El sistema permitirá hacer espejo a un módulo del DFI que se esté diseñando.

Prioridad Secundario.

Estimado Final de la segunda versión.

RF 10: Traer Módulo al frente.

Descripción El sistema permitirá cambiar el orden de precedencia de un módulo del DFI que se esté diseñando.

Prioridad Secundario.

Estimado Final de la primera versión.

RF 11: Enviar Módulo al fondo.

Descripción El sistema permitirá cambiar el orden de precedencia de un módulo del DFI que se esté diseñando.

Prioridad Secundario.

Estimado Final de la primera versión.

2.3.2 Funcionalidad Gestionar Conexión

RF 12: Crear Conexión

Descripción El sistema permitirá crear una conexión entre dos módulos del DFI que se esté diseñando.

Prioridad Crítico.

Estimado Final de la primera versión.

RF 13: Mover conexión.

Descripción El sistema permitirá mover una conexión existente entre dos módulos del DFI que se esté diseñando.

Prioridad Crítico.

Estimado Final de la primera versión.

RF 14: Seleccionar conexión.

Descripción El sistema permitirá seleccionar una conexión existente entre dos módulos del DFI que se esté diseñando.

Prioridad Crítico.

Estimado Final de la primera versión.

RF 15: Eliminar conexión.

Descripción El sistema permitirá eliminar una conexión existente entre dos módulos del DFI que se esté diseñando.

Prioridad Crítico.

Estimado Final de la primera versión.

2.3.3 Funcionalidad Gestionar Capas

RF 16: Agrupar en capa.

Descripción El sistema permitirá agrupar en una capa varios componentes existentes en el DFI que se esté diseñando.

Prioridad Crítico.

Estimado Final de la primera versión.

RF 17: Desagrupar capa.

Descripción El sistema permitirá desagrupar componentes de una capa existente en el DFI que se esté diseñando.

Prioridad Crítico.

Estimado Final de la primera versión.

RF 18: Seleccionar capa.

Descripción El sistema permitirá seleccionar una capa existente en el DFI que se esté diseñando.

Prioridad Secundario.

Estimado Final de la primera versión.

RF 19: Bloquear capa.

Descripción El sistema permitirá bloquear una capa existente en el DFI que se esté diseñando. Imposibilitando de tal forma que el especialista pueda realizar operaciones sobre la misma.

Prioridad Secundario.

Estimado Final de la primera versión.

RF 20: Desbloquear capa

Descripción El sistema permitirá desbloquear una capa existente en el DFI que se esté diseñando.

Prioridad Secundario.

Estimado Final de la primera versión.

RF 21: Ocultar capa.

Descripción El sistema permitirá ocultar una capa existente en el DFI que se esté diseñando. Se dejarán de mostrar todos los componentes de la misma.

Prioridad Secundario.

Estimado Final de la primera versión.

RF 22: Mover capa.

Descripción El sistema permitirá cambiar de posición todos los componentes pertenecientes a una capa existente en el DFI que se esté diseñando.

Prioridad Secundario.

Estimado Final de la primera versión.

RF 23: Eliminar capa.

Descripción El sistema permitirá eliminar una capa existente en el DFI que se esté diseñando. Se eliminarán todos los componentes de la misma.

Prioridad Secundario.

Estimado Final de la primera versión.

RF 24: Mostrar capa.

Descripción El sistema permitirá mostrar una capa oculta existente en el DFI que se esté diseñando. Se mostrarán todos los componentes ocultos de la misma.

Prioridad Secundario.

Estimado Final de la primera versión.

2.3.4 Funcionalidad Gestionar Etiqueta

RF 25: Dibujar etiqueta.

Descripción El sistema permitirá dibujar una nueva etiqueta dentro del DFI que se esté diseñando.

Prioridad Secundario.

Estimado Final de la primera versión.

RF 26: Mover etiqueta.

Descripción El sistema permitirá cambiar la posición de una etiqueta dentro del DFI que se esté diseñando.

Prioridad Secundario.

Estimado Final de la primera versión.

RF 27: Modificar etiqueta.

Descripción El sistema permitirá cambiar el texto de una etiqueta dentro del DFI que se esté diseñando.

Prioridad Secundario.

Estimado Final de la primera versión.

RF 28: Eliminar etiqueta.

Descripción El sistema permitirá eliminar una etiqueta dentro del DFI que se esté diseñando.

Prioridad Secundario.

Estimado Final de la primera versión.

RF 29: Asociar Etiqueta a Módulo.

Descripción El sistema permitirá asociar una etiqueta a un módulo existente dentro del DFI que se esté diseñando.

Prioridad Secundario.

Estimado Final de la primera versión.

RF 30: Asociar etiqueta a Conexión.

Descripción El sistema permitirá asociar una etiqueta a una conexión existente dentro del DFI que se esté diseñando.

Prioridad Secundario.

Estimado Final de la primera versión.

RF 31: Seleccionar etiqueta.

Descripción El sistema permitirá seleccionar una etiqueta existente dentro del DFI que se esté diseñando.

Prioridad Secundario.

Estimado Final de la primera versión.

2.3.5 Funcionalidad Gestionar DFI

RF 32: Cambiar porcentaje de visualización del DFI.

Descripción El sistema permitirá cambiar el tamaño del DFI que se esté diseñando.

Prioridad Crítico.

Estimado Final de la primera versión.

RF 33: Permitir Vista en Miniatura del DFI.

Descripción El sistema permitirá generar una vista más pequeña que muestre todos los componentes del DFI que se esté diseñando.

Prioridad Crítico.

Estimado Final de la primera versión.

RF 34: Deshacer la última acción (Undo).

Descripción El sistema permitirá deshacer alguna acción realizada al DFI que se esté diseñando.

Prioridad Crítico.

Estimado Final de la primera versión.

RF 35: Rehacer última acción deshecha (Redo).

Descripción El sistema permitirá rehacer alguna acción deshecha realizada al DFI

que se esté diseñando.

Prioridad Crítico.

Estimado Final de la primera versión.

RF 36: Exportar el DFI.

Descripción El sistema permitirá exportar como una imagen el DFI que se esté diseñando.

Prioridad Crítico.

Estimado Final de la primera versión.

RF 37: Realizar zoom al DFI.

Descripción El sistema permitirá cambiar el tamaño de todo el DFI que se esté diseñando.

Prioridad Crítico.

Estimado Final de la primera versión.

RF 38: Actualizar el DFI.

Descripción El sistema permitirá actualizar todo el DFI que se esté diseñando ante cualquier cambio en el estado de los elementos que lo componen.

Prioridad Crítico.

Estimado Final de la primera versión.

RF 39: Copiar componentes del DFI.

Descripción El sistema permitirá realizar copias de componentes existentes en el DFI que se esté diseñando y almacenarlas momentáneamente en la memoria de la PC.

Prioridad Crítico.

Estimado Final de la primera versión.

RF 40: Pegar componentes del DFI.

Descripción El sistema permitirá pegar los componentes almacenados en la memoria de la PC hacia el DFI que se esté diseñando.

Prioridad Crítico.

Estimado Final de la primera versión.

RF 41: Cortar componentes del DFI.

Descripción El sistema permitirá cortar componentes existentes en el DFI que se esté diseñando y almacenarlos momentáneamente en la memoria de la PC.

Prioridad Crítico.

Estimado Final de la primera versión.

2.3.6 Funcionalidad Multiselección

RF 42: Permitir la Multiselección de Componentes del DFI.

Descripción El sistema permitirá seleccionar varios componentes existentes en el DFI que se esté diseñando.

Prioridad Crítico.

Estimado Final de la primera versión.

RF 43: Mover varios elementos seleccionados.

Descripción El sistema permitirá cambiar la posición de varios componentes existentes en el DFI que se esté diseñando.

Prioridad Crítico.

Estimado Final de la primera versión.

RF 44: Eliminar varios elementos seleccionados.

Descripción El sistema permitirá eliminar varios componentes existentes en el DFI que se esté diseñando.

Prioridad Crítico.

Estimado Final de la primera versión.

2.4 Requisitos no Funcionales.

2.4.1 Requisitos de software

- ❖ El subsistema debe correr sobre la plataforma Windows XP o superior.
- ❖ Se requiere de la instalación del Framework .NET 2.0.

2.4.2 Requisitos de hardware

- ❖ Periféricos: Mouse, Teclado
- ❖ 256 MB de RAM o Superior.
- ❖ 150 MB mínimo de de espacio libre en el disco duro.

2.4.3 Requisitos de Usabilidad

- ❖ El subsistema deberá tener una interfaz intuitiva, organizada y de fácil entendimiento para el usuario.
- ❖ El subsistema deberá tener un grupo de menús que sugieran de forma intuitiva al usuario las opciones principales, además de las paletas contenedoras de módulos y corrientes, de edición por capas, y de herramientas de edición de diagramas, con íconos estén en correspondencia con lo que significan en la realidad.

- ❖ El sistema deberá ser utilizado por usuarios que tengan conocimiento acerca del funcionamiento y procesamiento de la información con que se trabaja en el sistema.

2.4.4 Requisitos de rendimiento

- ❖ El sistema deberá tener un tiempo de respuesta ante las peticiones del usuario de 0.3 seg. como máximo.

- ❖ El sistema deberá consumir como máximo 80 Mb de memoria RAM.

2.4.5 Restricciones de diseño o implementación

- ❖ El subsistema será una aplicación de escritorio.
- ❖ El subsistema será implementado en el lenguaje C#.
- ❖ El subsistema deberá utilizar como IDE de desarrollo el Visual Studio .NET 2005.
- ❖ El subsistema deberá usar como herramienta de modelación el Visual Paradigm for UML 6.0 Enterprise Edition.

2.5 Definición de los Actores del Sistema.

Tabla 1: Definición de los Actores del Sistema.

Actor	Descripción
Especialista	Es el encargado de llevar a cabo la edición del DFI, utilizando todas las funcionalidades que brinda el subsistema, es por este motivo que es el único Actor del Sistema.

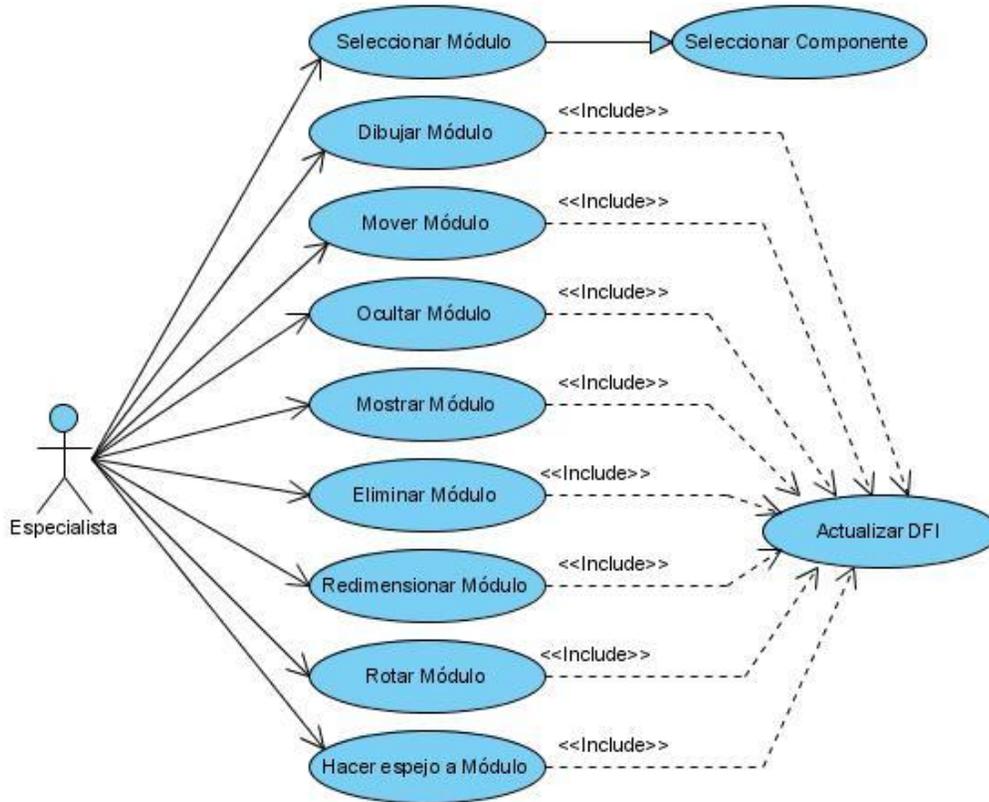
2.6 Diagrama de Casos de Uso del Sistema

Para lograr una mejor comprensión del subsistema se decidió dividir el Diagrama de Casos de Uso del Sistema en secciones, separando los casos de uso por funcionalidades. A continuación se encontrarán dichas secciones y las descripciones de los Casos de Uso más importantes que se trataran a lo largo de este trabajo.

2.6.1 Sección de Gestión de Módulo.

Diagrama de CU Sección de Gestión de Módulo.

Figura 4: DCUS Sección de Gestión de Módulo.



Descripciones de los CUS de la Sección de Gestión de Módulo.

Tabla 2: Descripción del CUS Dibujar Módulo

Caso de Uso:	Dibujar Módulo
Actores:	Especialista
Resumen:	El caso de uso se inicia cuando el Especialista se dispone Dibujar un Módulo en el diagrama.
Precondiciones:	
Referencias	RF 1
Prioridad	Crítico

Flujo Normal de Eventos	
Actor	Sistema
1. El Especialista escoge de la Paleta de Componentes el Módulo que desea colocar en diagrama.	2. El sistema cambia el ícono del Mouse.
3. El Especialista selecciona el área del diagrama donde este desea colocarlo.	4. El sistema dibuja el módulo en las coordenadas de la posición donde el Especialista seleccionó.
	5. El sistema adiciona el Módulo al Listado de Componentes perteneciente al DFI.
	6. Se inicializa el Caso de Uso Actualizar DFI descrito en la Sección de Gestión del DFI.
Flujos Alternos	
Actor	Sistema
Poscondiciones	

Tabla 3: Descripción del CUS Seleccionar Módulo

Caso de Uso:	Seleccionar Módulo
Actores:	Especialista
Resumen:	El caso de uso se inicia cuando el Especialista selecciona un Módulo del DFI
Precondiciones:	Debe existir al menos un Módulo en el DFI.
Referencias	RF 2
Prioridad	Crítico
Flujo Normal de Eventos	
Actor	Sistema
1. El Especialista selecciona un	2. El sistema dibuja el área de selección del Módulo.

Módulo existente en el DFI	
	3. Habilita las opciones de Rotar, Realizar Espejo, Redimensionar, Ocultar, Mostrar y Eliminar Módulo.
	4. El sistema destaca el Módulo seleccionado en el Listado de Componentes perteneciente al DFI.
Flujos Alternos	
Actor	Sistema
1. El Especialista selecciona un Módulo existente en el DFI	2. Si existe algún módulo seleccionado se inicializa el CU Multiselección descrito en la Sección de Gestión de Multiselección.
	3. El sistema dibuja el área de selección del Módulo.
	4. Habilita las opciones de Rotar, Realizar Espejo, Redimensionar, Ocultar, Mostrar y Eliminar Módulo.
	5. El sistema destaca el Módulo seleccionado en el Listado de Componentes perteneciente al DFI.
Poscondiciones	

Tabla 4: Descripción del CUS Mover Módulo.

Caso de Uso:	Mover Módulo.
Actores:	Especialista
Resumen:	El caso de uso se inicia cuando el Especialista selecciona un Módulo del DFI y cambia su posición dentro del diagrama.
Precondiciones:	Debe existir al menos un Módulo en el DFI. El Módulo debe estar previamente seleccionado. El Módulo seleccionado no debe estar oculto en el DFI. El Módulo seleccionado no debe estar Bloqueado.
Referencias	RF 3

Prioridad	Crítico
Flujo Normal de Eventos	
Actor	Sistema
1. El especialista selecciona un módulo del DFI y cambia su posición dentro del diagrama.	2. El sistema dibuja el Módulo en la nueva posición seleccionada por el Especialista.
	3. Se inicializa el Caso de Uso Actualizar DFI descrito en la Sección de Gestión del DFI.
Flujos Alternos	
Actor	Sistema
Poscondiciones	

Tabla 5: Descripción del CUS Ocultar Módulo.

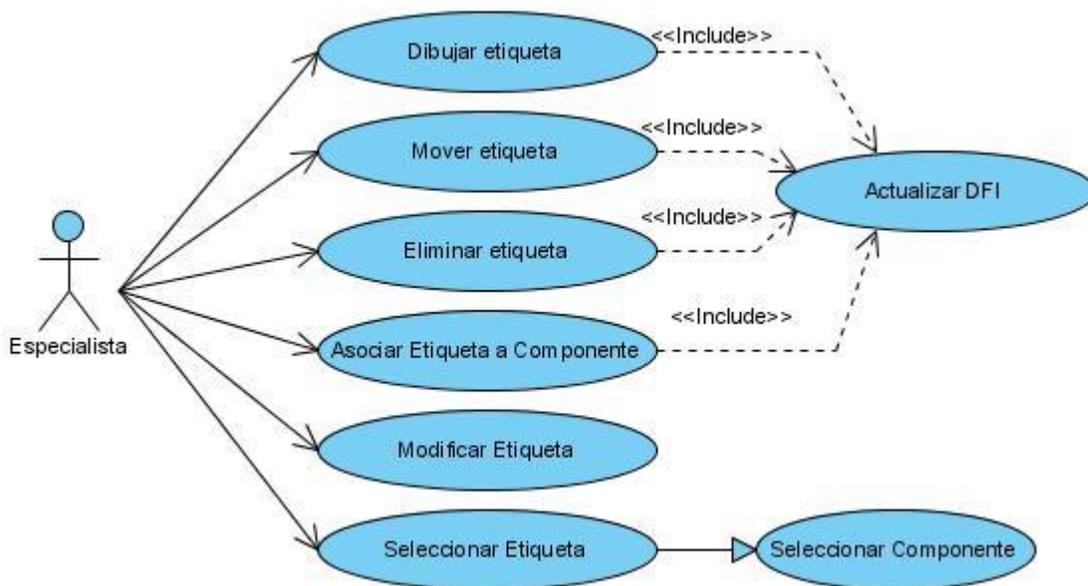
Caso de Uso:	Ocultar Módulo.
Actores:	Especialista
Resumen:	El caso de uso se inicia cuando el Especialista selecciona un Módulo del DFI y selecciona la opción de ocultar el Módulo.
Precondiciones:	Debe existir al menos un Módulo en el DFI. El Módulo debe estar seleccionado. El Módulo seleccionado no debe estar Bloqueado ni oculto.
Referencias	RF 4
Prioridad	Secundario
Flujo Normal de Eventos	
Actor	Sistema
1. El Especialista selecciona un módulo del	2. El sistema borra el Módulo del DFI.

DFI y luego selecciona la opción de ocultarlo.	
	3. El sistema cambia el ícono de la visibilidad en el Listado de Componentes perteneciente al DFI.
	4. El sistema oculta las conexiones del Módulo seleccionado.
	5. Se inicializa el Caso de Uso Actualizar DFI descrito en la Sección de Gestión del DFI.
Flujos Alternos	
Actor	Sistema
Poscondiciones	

2.6.2 Sección de Gestión de Etiqueta.

Diagrama de CUS Sección de Gestión de Etiqueta.

Figura 5: Diagrama de CUS Sección de Gestión de Etiqueta.



Descripciones de los CUS de la Sección de Gestión de Etiqueta.

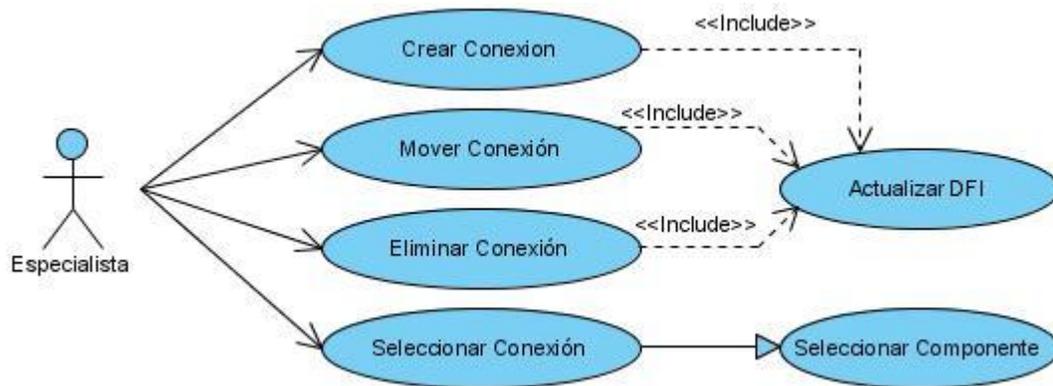
Tabla 6: Descripción del CUS Dibujar Etiqueta.

Caso de Uso:	Dibujar Etiqueta
Actores:	Especialista
Resumen:	El caso de uso se inicia cuando el Especialista se dispone Dibujar una Etiqueta.
Precondiciones:	
Referencias	RF 40
Prioridad	Crítico
Flujo Normal de Eventos	
Actor	Sistema
1. El Especialista escoge de la Paleta de Componentes la etiqueta.	2. El sistema cambia el ícono del Mouse.
3. El Especialista selecciona el área del diagrama donde este desea colocar la etiqueta.	4. El sistema dibuja la etiqueta en la posición escogida por el Especialista.
	5. El sistema adiciona la etiqueta al Listado de Componentes perteneciente al DFI.
	6. El sistema solicita el texto de la Etiqueta.
7. El Especialista teclea el texto de la etiqueta.	8. El sistema redibuja la etiqueta con el texto seleccionado.
	9. Se inicializa el Caso de Uso Actualizar DFI descrito en la Sección de Gestión del DFI.
Flujos Alternos	
Actor	Sistema
Poscondiciones	

2.6.3 Sección de Gestión de Conexión.

Diagrama de CUS Sección de Gestión de Conexión.

Figura 6: Diagrama de CUS Sección de Gestión de Conexión.



Descripciones de los CUS de la Sección de Gestión de Conexión.

Tabla 7: Descripción del CUS Crear Conexión.

Caso de Uso:	Crear Conexión
Actores:	Especialista
Resumen:	El caso de uso se inicia cuando el Especialista selecciona un Módulo del DFI y crea una conexión entre este y otro Módulo.
Precondiciones:	Deben existir al menos dos Módulos en el DFI.
Referencias	RF 24
Prioridad	Crítico
Flujo Normal de Eventos	
Actor	Sistema
1. El especialista selecciona un puerto del Módulo.	2. El sistema cambia el icono del mouse.
3. El especialista selecciona un	4. El sistema crea la conexión.

puerto del otro Módulo.	
	5. Se inicializa el Caso de Uso Actualizar DFI descrito en la Sección de Gestión del DFI.
Flujos Alternos	
Actor	Sistema
Poscondiciones	

Tabla 8: Descripción del CUS Mover Conexión.

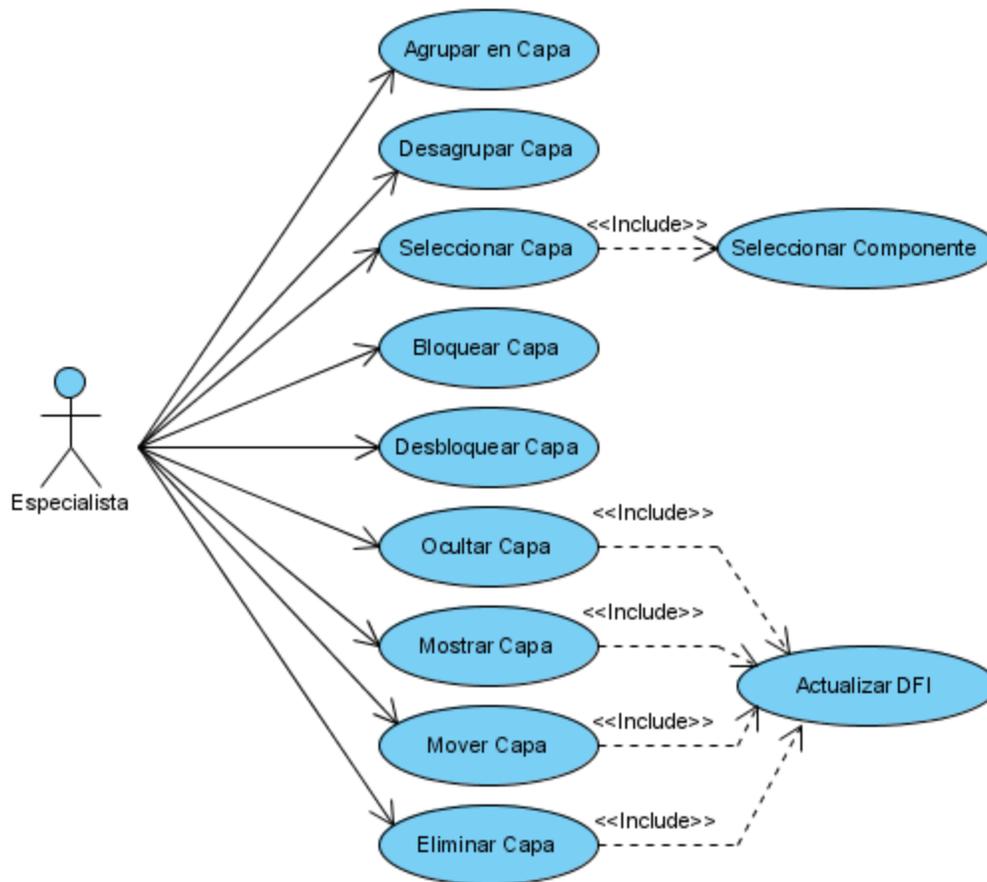
Caso de Uso:	Mover Conexión.
Actores:	Especialista
Resumen:	El caso de uso se inicia cuando el Especialista selecciona una Conexión y cambia su posición en el DFI.
Precondiciones:	Debe existir al menos una conexión en el DFI. La conexión debe estar seleccionada. La conexión no debe estar oculta, ni pertenecer a una capa que esté bloqueada.
Referencias	RF 25
Prioridad	Crítico
Flujo Normal de Eventos	
Actor	Sistema
1. El especialista selecciona una Conexión del DFI y cambia su posición.	2. El sistema cambia el icono del mouse.
	3. El sistema redibuja la conexión.
	4. Se inicializa el Caso de Uso Actualizar DFI descrito en la Sección de Gestión del DFI.
Flujos Alternos	

Actor	Sistema
Poscondiciones	

2.6.4 Sección de Gestión de Capa.

Diagrama de CUS Sección de Gestión de Capa.

Figura 7: Diagrama de CUS Sección de Gestión de Capa.



Descripciones de los CUS de la Sección de Gestión de Capa.

Tabla 9: Descripción del CUS Agrupar en Capa.

Caso de Uso:	Agrupar en Capa
--------------	-----------------

Actores:	Especialista
Resumen:	El caso de uso se inicia cuando el Especialista se dispone a agrupar un grupo de componentes del diagrama previamente seleccionados.
Precondiciones:	Deben estar seleccionados dos o más componentes dentro del DFI.
Referencias	RF 28
Prioridad	Crítico
Flujo Normal de Eventos	
Actor	Sistema
1. El Especialista selecciona la Opción de Agrupar Componentes en Capa.	2. El sistema agrupa los elementos seleccionados en una capa.
	3. El sistema dibuja el área de selección de la capa.
	4. El sistema coloca los elementos agrupados dentro del grupo jerárquico de Capa en el Listado de Componentes perteneciente al DFI.
	5. El sistema pide un nombre para la nueva Capa.
6. El usuario teclea un nombre para la Capa.	7. El sistema renombra la nueva Capa con el nombre seleccionado por el usuario y lo actualiza en el Listado de Componentes pertenecientes al DFI.
Flujos Alternos	
Actor	Sistema
Poscondiciones	

Tabla 10: Descripción del CUS Seleccionar Capa.

Caso de Uso:	Seleccionar Capa.
Actores:	Especialista

Resumen:	El caso de uso se inicia cuando el Especialista selecciona una Capa del DFI.
Precondiciones:	Debe existir al menos una Capa en el DFI.
Referencias	RF 30
Prioridad	Crítico
Flujo Normal de Eventos	
Actor	Sistema
1. El especialista selecciona una Capa del DFI.	2. El sistema dibuja el Área de Selección de la Capa seleccionada.
	3. Se habilitan las opciones de Desagrupar, Ocultar/Mostrar, Bloquear/Desbloquear y Eliminar Capa.
	4. Se resaltan las componentes dentro de la Estructura jerárquica de Capa y dicha Capa, dentro de la Lista de Componentes pertenecientes al DFI.
Flujos Alternos	
Actor	Sistema
Poscondiciones	

Tabla 11: Descripción del CUS Ocultar Capa.

Caso de Uso:	Ocultar Capa.
Actores:	Especialista
Resumen:	El caso de uso se inicia cuando el Especialista selecciona una Capa del DFI y selecciona la opción de Ocultar Capa.
Precondiciones:	Debe existir al menos una Capa en el DFI. La Capa debe estar seleccionada. La Capa seleccionada no debe estar Bloqueada ni oculta.

Referencias	RF 33
Prioridad	Secundario
Flujo Normal de Eventos	
Actor	Sistema
1. El Especialista selecciona una Capa del DFI y luego selecciona la opción de ocultarlo.	2. El sistema cambia el ícono de la visibilidad de la Capa en el Listado de Componentes perteneciente al DFI y de todos los Componentes pertenecientes a dicha Capa.
	3. Se inicializa el Caso de Uso Actualizar DFI descrito en la Sección de Gestión del DFI.
Flujos Alternos	
Actor	Sistema
Poscondiciones	

Tabla 12: Descripción del CUS Bloquear Capa.

Caso de Uso:	Bloquear Capa
Actores:	Especialista
Resumen:	El caso de uso se inicia cuando el Especialista selecciona una Capa del DFI y selecciona la opción de Bloquear Capa.
Precondiciones:	Debe existir al menos una Capa en el DFI. La Capa debe estar seleccionada. La Capa seleccionada no debe estar Bloqueada ni oculta.
Referencias	RF 31
Prioridad	Crítico
Flujo Normal de Eventos	
Actor	Sistema

1. El Especialista selecciona una Capa del DFI y luego selecciona la opción de Bloquear Capa.	2. El sistema cambia el ícono de movilidad de la capa en el Listado de Componentes perteneciente al DFI.
	3. El sistema cambia el ícono de movilidad de cada uno de los componentes de la capa en el Listado de Componentes perteneciente al DFI.
	4. El sistema cambia las propiedades de movilidad de la capa.
	5. El sistema cambia las propiedades de movilidad a cada uno de los componentes de la capa.
Flujos Alternos	
Actor	Sistema
Poscondiciones	

Tabla 13: Descripción del CUS Mover Capa.

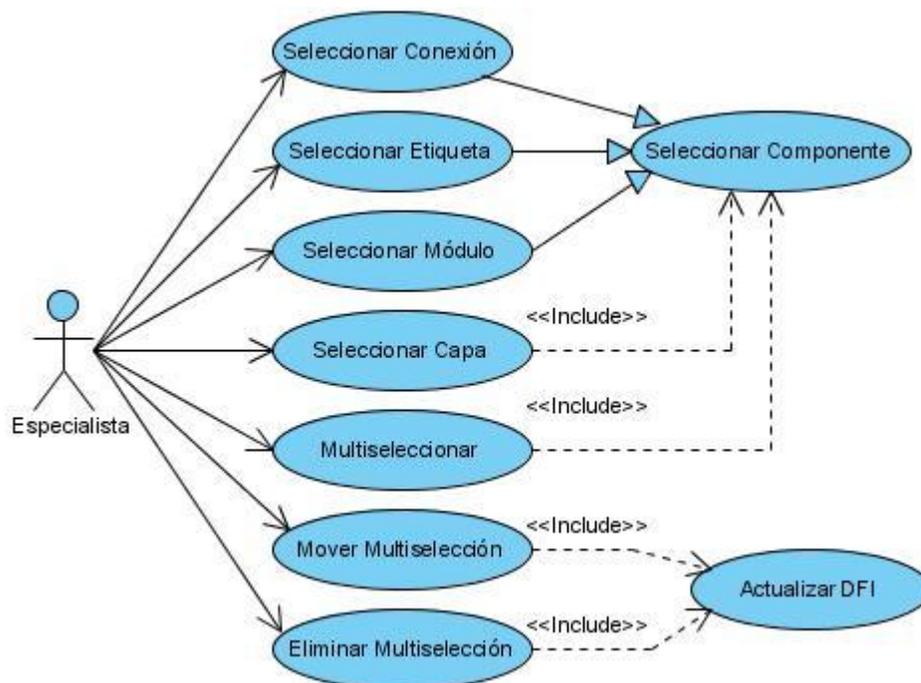
Caso de Uso:	Mover Capa
Actores:	Especialista
Resumen:	El caso de uso se inicia cuando el Especialista selecciona una Capa del DFI y cambia su posición dentro del diagrama.
Precondiciones:	Debe existir al menos una Capa en el DFI. La Capa debe estar seleccionada. La Capa seleccionada no debe estar Bloqueada, ni Oculta.
Referencias	RF 34
Prioridad	Crítico
Flujo Normal de Eventos	
Actor	Sistema
1. El Especialista selecciona una	2. El sistema calcula las coordenadas de localización para

Capa del DFI y luego cambia su posición dentro del DFI.	cada uno de los componentes de la Capa de acuerdo a la nueva posición seleccionada por el Especialista.
	3. Se inicializa el Caso de Uso Actualizar DFI descrito en la Sección de Gestión del DFI.
Flujos Alternos	
Actor	Sistema
Poscondiciones	

2.6.5 Sección de Gestión de Selección.

Diagrama de CUS Sección de Gestión de Selección.

Figura 8: Diagrama de CUS Sección de Gestión de Selección.



Descripciones de los CUS de la Sección de Gestión de Multiselección.

Tabla 14: Descripción del CUS Multiseleccionar.

Caso de Uso:	Multiseleccionar
Actores:	Especialista
Resumen:	El caso de uso se inicia cuando el Especialista se dispone a seleccionar un componente del DFI y hay al menos otro componente seleccionado.
Precondiciones:	Debe existir al menos un componente seleccionado en el DFI.
Referencias	RF 58
Prioridad	Crítico
Flujo Normal de Eventos	
Actor	Sistema
1. El Especialista presionando la tecla Ctrl o Control selecciona otro componente del DFI, ya sea seleccionándolo con el Mouse desde el Panel de Dibujo del editor o desde la Lista de Elementos pertenecientes al DFI.	2. El sistema adiciona en la Lista de Elementos Seleccionados el nuevo componente.
	3. El sistema redibuja el área de selección de los Componentes que se encuentran en la Lista de Elementos Seleccionados.
	4. El sistema habilita la opción de Agrupar en Capas.
Flujos Alternos	
Actor	Sistema
1. El actor sosteniendo el botón izquierdo del Mouse selecciona un área en el panel de dibujo.	2. El sistema Verifica si dentro de esa área de selección se encuentra algún componente.
	3. Adiciona los componentes que se

	encuentran dentro del área de selección a la lista de elementos seleccionados
	5. El sistema redibuja el área de selección de los Componentes que se encuentran en la Lista de Elementos Seleccionados.
	6. El sistema habilita la opción de Agrupar en Capas.
Poscondiciones	

Tabla 15 Descripción del CUS Mover Multiselección.

Caso de Uso:	Mover Multiselección
Actores:	Especialista
Resumen:	El caso de uso se inicia cuando el Especialista se dispone a mover un grupo de componentes seleccionados previamente en el DFI.
Precondiciones:	Deben existir al menos dos componentes seleccionados en el DFI.
Referencias	RF 42
Prioridad	Crítico
Flujo Normal de Eventos	
Actor	Sistema
1. El Especialista comienza a mover los elementos seleccionados.	2. El sistema dibuja un contorno de selección que engloba a todos los componentes y lo desplaza hasta la posición donde el usuario tiene el mouse.
3. El especialista deja de presionar el botón primario del mouse en la posición donde desea colocar los componentes.	4. El sistema cambia la posición de todos los elementos seleccionados.
	5. Se inicializa el Caso de Uso Actualizar DFI

	descrito en la Sección de Gestión del DFI.
Flujos Alternos	
Actor	Sistema
Poscondiciones	

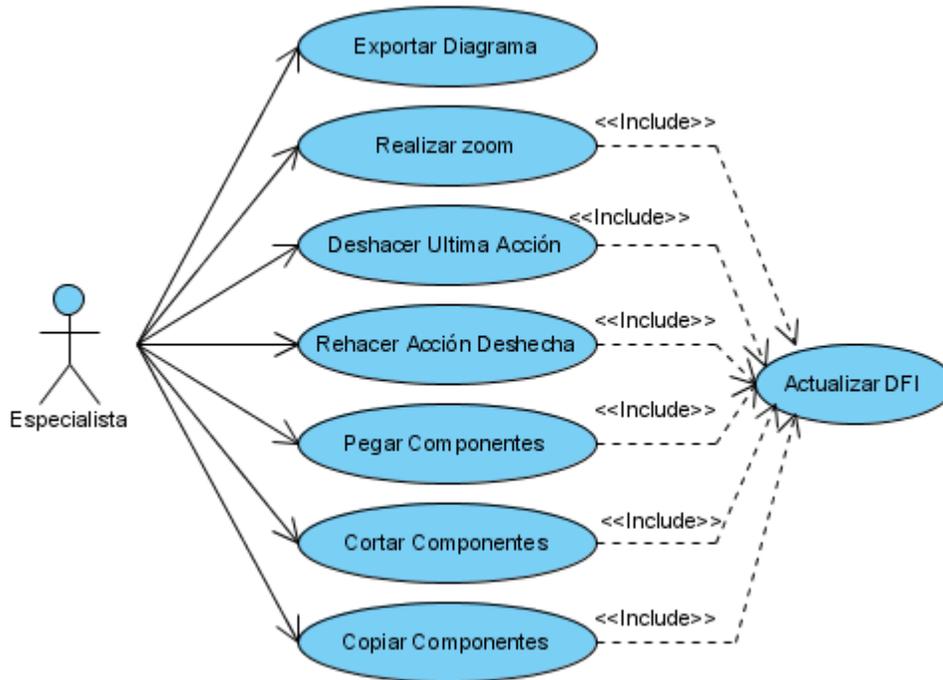
Tabla 16 Descripción de CUS Eliminar Multiselección.

Caso de Uso:	Eliminar Multiselección
Actores:	Especialista
Resumen:	El caso de uso se inicia cuando el Especialista se dispone a eliminar un grupo de componentes seleccionados previamente en el DFI.
Precondiciones:	Deben existir al menos dos componentes seleccionados en el DFI.
Referencias	RF 42
Prioridad	Crítico
Flujo Normal de Eventos	
Actor	Sistema
1. El Especialista selecciona la opción de eliminar multiselección.	2. El sistema elimina del DFI los componentes seleccionados.
	3. El sistema elimina del Listado de Componentes del DFI.
	4. Se inicializa el Caso de Uso Actualizar DFI descrito en la Sección de Gestión del DFI.
Flujos Alternos	
Actor	Sistema
Poscondiciones	

2.6.6 Sección de Gestión del DFI.

Diagrama de CUS Sección de Gestión DFI.

Figura 9: Diagrama de CUS Sección de Gestión del DFI.



Descripciones de los CUS de la Sección de Gestión del DFI.

Tabla 17: Descripción del CUS Realizar Zoom.

Caso de Uso:	Realizar Zoom
Actores:	Especialista
Resumen:	El caso de uso se inicia cuando el Especialista cambia el área de visualización del diagrama realizando Zoom a la misma.
Precondiciones:	
Referencias	RF 53

Prioridad	Crítico
Flujo Normal de Eventos	
Actor	Sistema
1. El usuario click en la opción de Hacer Zoom.	2. El sistema cambia la escala del DFI.
	3. El sistema fija las coordenadas del centro del área de visualización del DFI por las coordenadas actuales del mouse.
	4. Se inicializa el Caso de Uso Actualizar DFI descrito en la Sección de Gestión del DFI.
Flujos Alternos	
Actor	Sistema
Poscondiciones	

Tabla 18: Descripción del CUS Deshacer Última Acción.

Caso de Uso:	Deshacer Última Acción.
Actores:	Especialista
Resumen:	El caso de uso se inicia cuando el Especialista selecciona la opción de Deshacer Última Acción.
Precondiciones:	Se debe haber modificado el diagrama.
Referencias	RF 50
Prioridad	Crítico
Flujo Normal de Eventos	
Actor	Sistema
1. El Especialista selecciona la opción de Deshacer Última Acción.	2. El sistema busca en la Lista de Acciones Realizadas la última realizada.

	3. El sistema realiza la operación inversa a esta.
	4. Se adiciona a la lista de Acciones Deshechas la acción que se deshizo.
	5. Se inicializa el Caso de Uso Actualizar DFI descrito en la Sección de Gestión del DFI.
Flujos Alternos	
Actor	Sistema
Poscondiciones	

Tabla 19: Descripción del CUS Cortar Componentes.

Caso de Uso:	Cortar Componentes.
Actores:	Especialista
Resumen:	El caso de uso se inicia cuando el Especialista selecciona al menos un Componente del diagrama y Selecciona la opción de Cortar Componentes.
Precondiciones:	Debe existir al menos un componente en el diagrama. Los componentes deben estar seleccionados. Los componentes no deben estar Bloqueados ni ocultos.
Referencias	RF 57
Prioridad	Crítico
Flujo Normal de Eventos	
Actor	Sistema
1. El Especialista selecciona al menos un Componente del diagrama y Selecciona la opción de Cortar Componentes.	2. El sistema copia los componentes con sus propiedades y los almacena temporalmente en la lista de elementos copiados.
	3. El sistema elimina los elementos del DFI.

	4. El sistema elimina de la Lista de Componentes perteneciente al DFI, los elementos cortados.
	5. El sistema habilita la opción de Pegar Componentes.
	6. Se inicializa el Caso de Uso Actualizar DFI descrito en la Sección de Gestión del DFI.
Flujos Alternos	
Actor	Sistema
Poscondiciones	

Tabla 20: Descripción del CUS Pegar Componentes.

Caso de Uso:	Pegar Componentes.
Actores:	Especialista
Resumen:	El caso de uso se inicia cuando el Especialista selecciona la opción de Pegar Componentes.
Precondiciones:	Deben existir componentes en la lista de elementos copiados.
Referencias	RF 56
Prioridad	Crítico
Flujo Normal de Eventos	
Actor	Sistema
1. El Especialista selecciona la opción de Pegar Componentes.	2. El sistema dibuja los componentes que están en la lista de elementos copiados en el diagrama.
	3. El sistema adiciona a la Lista de Componentes perteneciente al DFI, los elementos pegados.
	4. Se inicializa el Caso de Uso Actualizar DFI descrito en la Sección de Gestión del DFI.
Flujos Alternos	
Actor	Sistema

Poscondiciones	

Tabla 21: Descripción del CUS Exportar Diagrama.

Caso de Uso:	Exportar Diagrama
Actores:	Especialista
Resumen:	El caso de uso se inicia cuando el Especialista cambia el porcentaje de visualización del diagrama.
Precondiciones:	
Referencias	RF 52
Prioridad	Crítico
Flujo Normal de Eventos	
Actor	Sistema
1. El Especialista selecciona la opción de Exportar el diagrama.	2. El sistema pide al usuario la dirección donde se va a guardar la imagen del DFI.
3. El usuario selecciona la localización donde quiere guardar la imagen del DFI.	4. El sistema genera la imagen del DFI y la guarda en la dirección seleccionada por el usuario.
Flujos Alternos	
Actor	Sistema
Poscondiciones	

Tabla 22: Descripción del CUS Actualizar DFI.

Caso de Uso:	Actualizar DFI
Actores:	
Resumen:	El caso de uso iniciado por un caso de uso base.

Precondiciones:	
Referencias	RF 48.
Prioridad	
Flujo Normal de Eventos	
Actor	Sistema
	1. El sistema redibuja los componentes del diagrama, teniendo en cuenta el orden de precedencia y visibilidad de los mismos.
	2. El sistema redibuja las conexiones entre Módulos teniendo en cuenta el orden de precedencia y visibilidad.
Flujos Alternos	
Actor	Sistema
Poscondiciones	

2.7 Conclusiones

En este capítulo se ha desarrollado la propuesta de la solución del problema a través de la obtención de los Requisitos Funcionales y no Funcionales del subsistema y el modelado del diagrama de CUS utilizando diferentes patrones de caso de uso. La descripción detallada de algunos de los Casos de Uso más importantes, en un lenguaje común. Además de su clasificación según la prioridad, atendiendo su significación dentro de la arquitectura de la aplicación propuesta.

CAPÍTULO # 3: ANÁLISIS Y DISEÑO DE LA SOLUCIÓN PROPUESTA.

3.1 Introducción.

El objetivo principal de este capítulo es transformar los requerimientos a una especificación que describa cómo implementar el sistema. El análisis fundamentalmente consiste en obtener una visión de qué hace el sistema de software a desarrollar, por tal motivo éste se centra en los requerimientos funcionales obtenidos anteriormente. En él se estructuran los requisitos de manera que se facilite su comprensión, preparación, modificación y en general su mantenimiento. Esta estructura es independiente de la estructura que se dio a los requisitos (basada en casos de uso). Sin embargo existe una trazabilidad directa entre esas distintas estructuras, la cual se define entre casos de uso del modelo de casos de uso y realizaciones del caso de uso en el modelo de análisis. En este flujo se refinan y estructuran los requisitos obtenidos con anterioridad.

Por otra parte, el diseño es un refinamiento que toma en cuenta los requerimientos no funcionales, por lo cual se centra en como el sistema cumple sus objetivos.

Los objetivos específicos del análisis y diseño son:

1. Transformar los requerimientos al diseño del futuro sistema.
2. Desarrollar una arquitectura para el sistema.
3. Adaptar el diseño para que sea consistente con el entorno de implementación.

Para darle cumplimiento a estos objetivos se desarrollaran los artefactos más importantes propuestos por la metodología RUP.

3.2 Modelo del Análisis.

Este modelo es usado para representar la estructura global del sistema, describe la realización de casos de uso, sirve como una abstracción del Modelo de Diseño y se centra en los requerimientos no funcionales.

Este modelo de análisis no es un diagrama final que describe todos los posibles conceptos y sus relaciones, es un primer intento por definir los conceptos claves que describen el sistema. Su utilidad radica en que permite una apreciación conceptual de una manera global del sistema. A diferencia del Modelo de Casos de Uso que captura la funcionalidad del sistema, el Modelo de Análisis da forma a la arquitectura para soportar estas funcionalidades.

3.3 Diagramas de Clases del Análisis

Un Diagrama de clases del análisis, es un artefacto en el que se representan los conceptos en un dominio del problema. Representa los conceptos del mundo real, no de la implementación automatizada de estos. La representación de estos conceptos se realiza a través de clases del análisis y las diferentes relaciones que pueden existir entre las mismas.

3.3.1 Clases del Análisis

Se centran en los requisitos funcionales y son evidentes en el dominio del problema porque representan conceptos y relaciones del dominio. Estas clases del análisis contienen atributos y entre ellas se establecen relaciones de asociación, agregación/composición, generalización/especialización y tipos asociativos. RUP propone clasificar a las clases en: Entidad, Interfaz y Control.

A continuación se muestran los diagramas de clases del análisis de algunos de los casos de uso del sistema divididos en sus respectivas secciones.

3.3.2 Diagramas de Clases del Análisis. Sección de Gestión de Módulo

Figura 10: Diagrama de Clases del Análisis del CU Dibujar Módulo

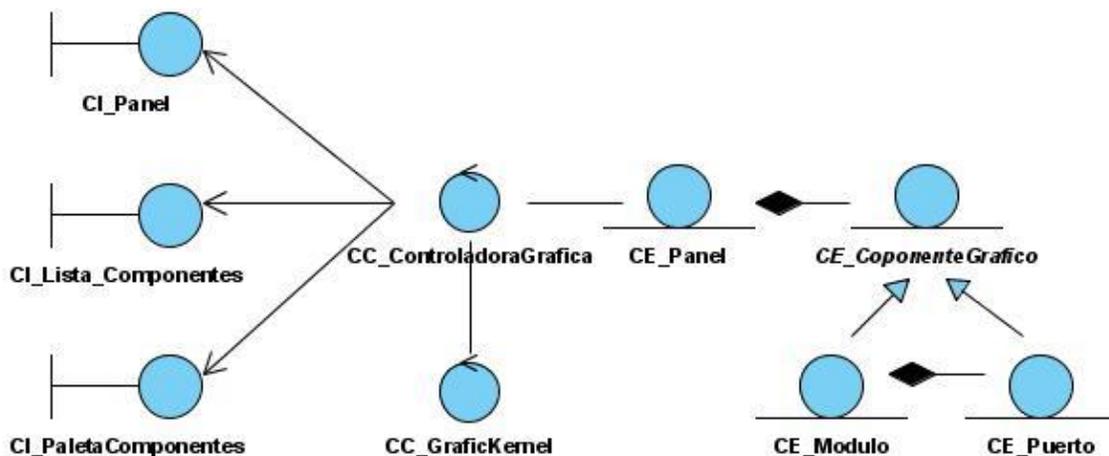


Figura 11: Diagrama de Clases del Análisis del CU Mover Módulo.

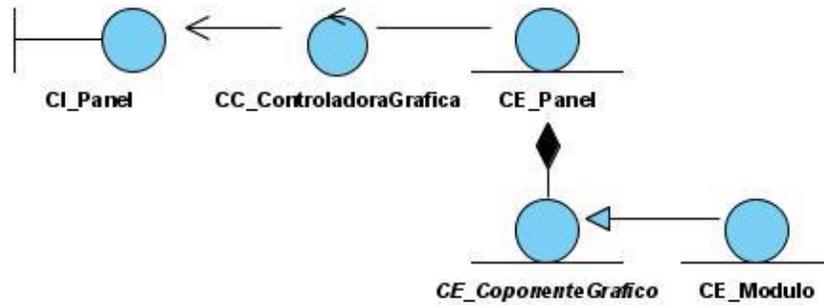


Figura 12: Diagrama de Clases del Análisis del CU Ocultar Módulo.

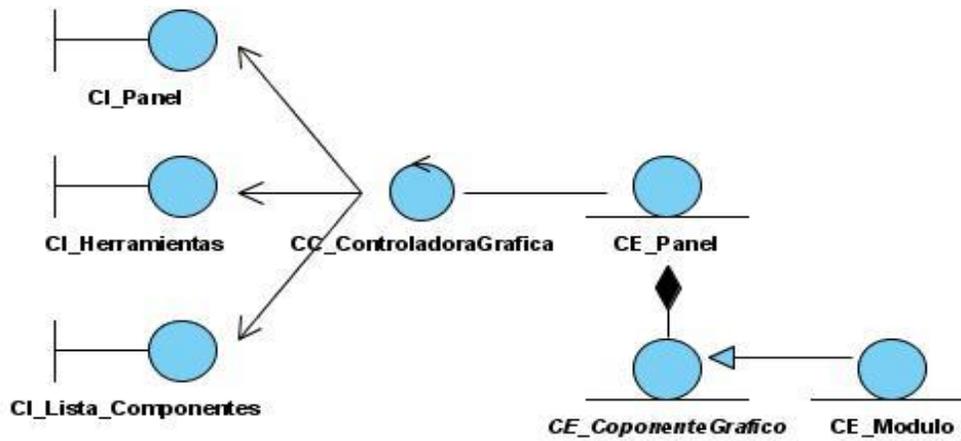
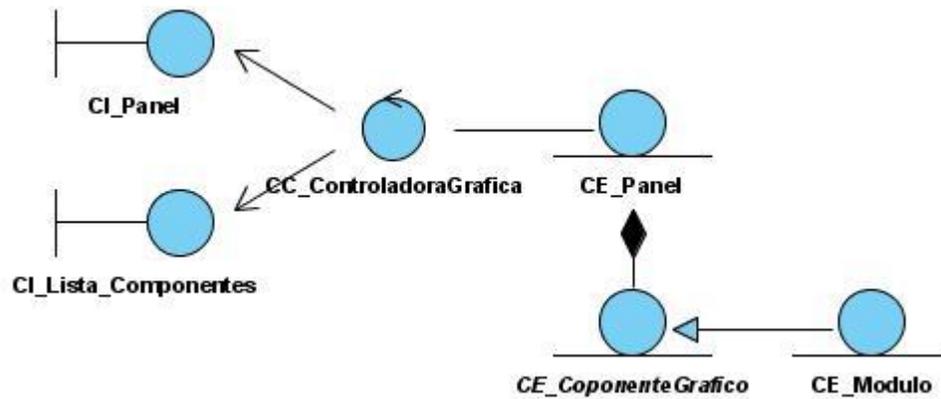
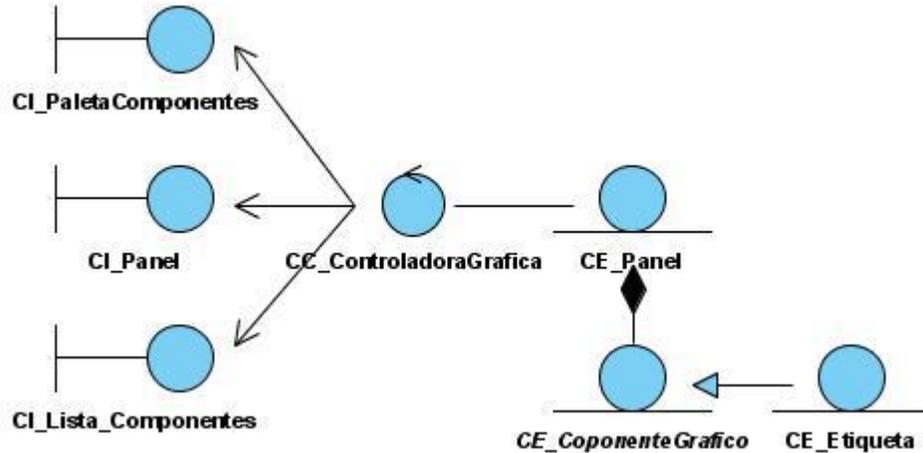


Figura 13: Diagrama de Clases del Análisis del CU Seleccionar Módulo.



3.3.3 Diagramas de Clases del Análisis. Sección de Gestión de Etiqueta

Figura 14: Diagrama de Clases del Análisis del CU Dibujar Etiqueta.



3.3.4 Diagramas de Clases del Análisis Sección de Gestión de Conexión

Figura 15: Diagrama de Clases del Análisis del CU Crear Conexión.

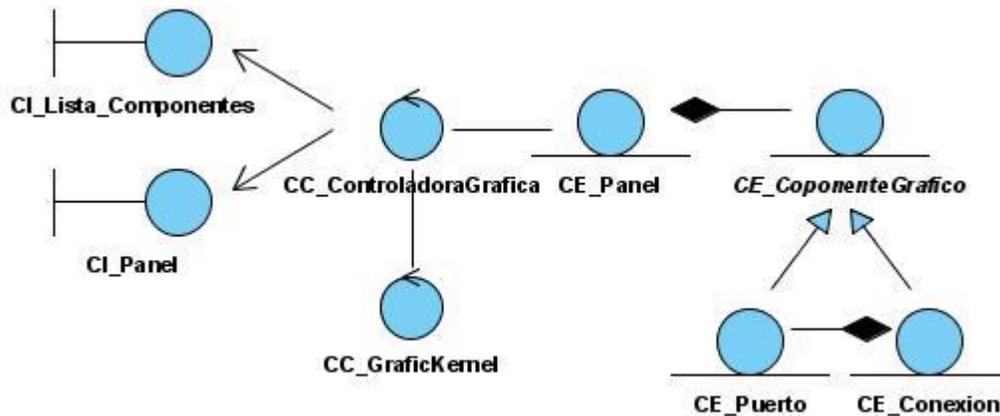
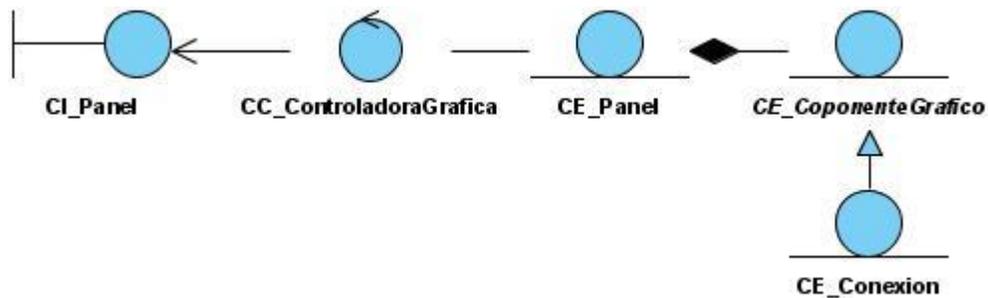


Figura 16: Diagrama de Clases del Análisis del CU Mover Conexión.



3.3.5 Diagramas de Clases del Análisis. Sección de Gestión de Capa

Figura 17: Diagrama de Clases del Análisis del CU Agrupar en Capa.

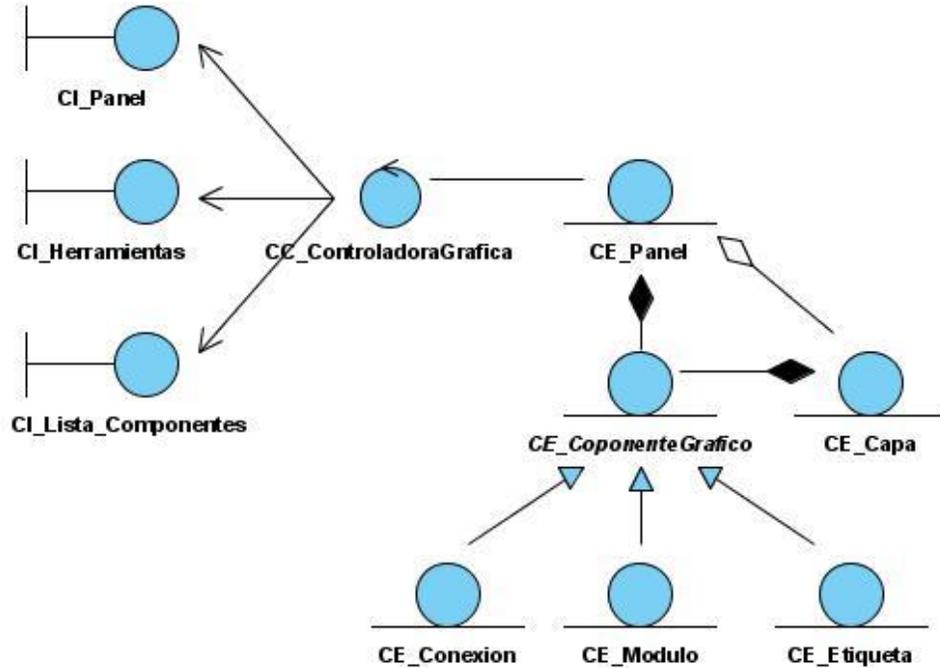


Figura 18: Diagrama de Clases del Análisis del CU Bloquear Capa.

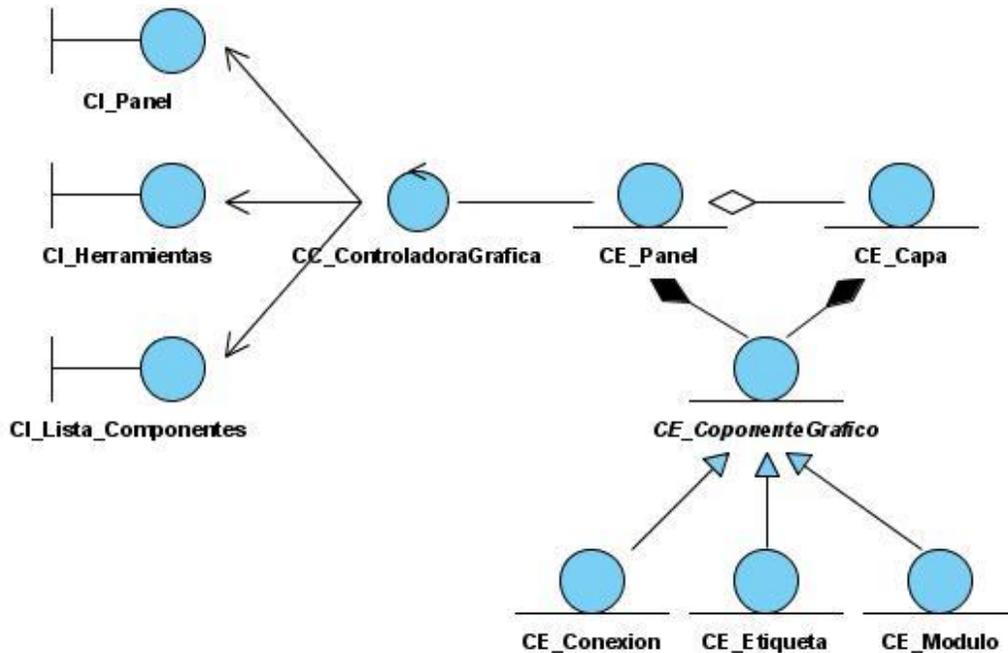


Figura 19: Diagrama de Clases del Análisis del CU Ocultar Capa.

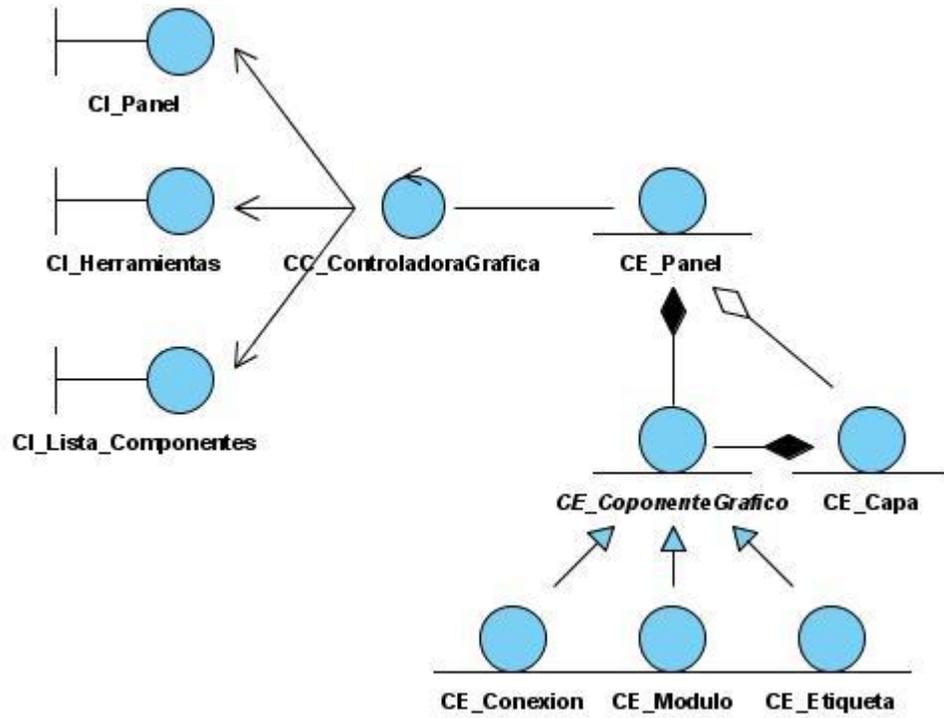


Figura 20: Diagrama de Clases del Análisis del CU Mover Capa.

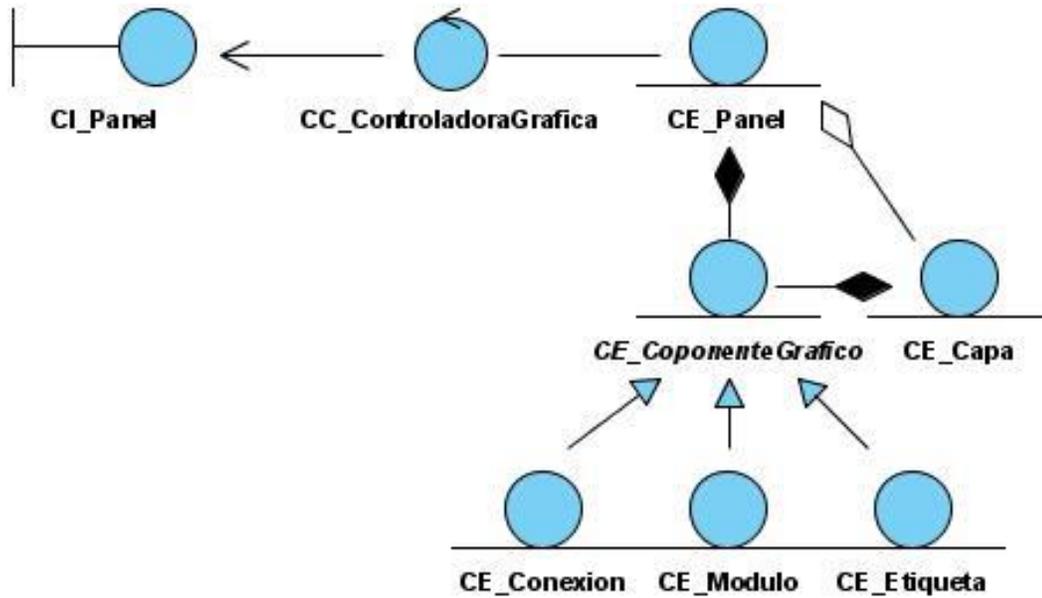
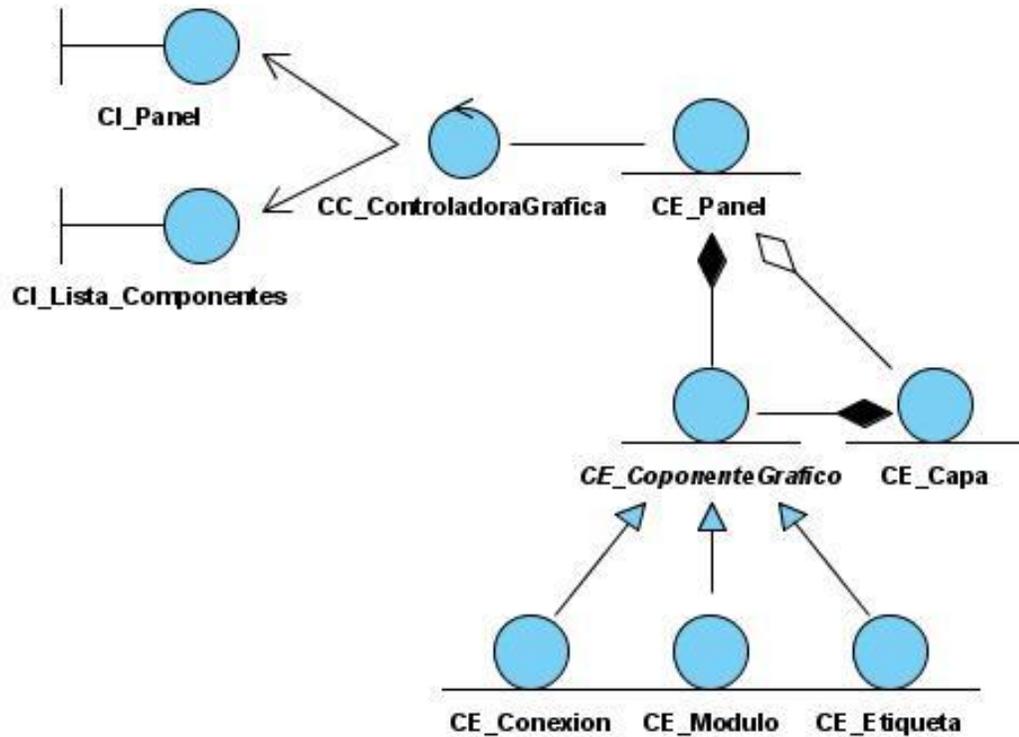


Figura 21: Diagrama de Clases del Análisis del CU Seleccionar Capa.



3.3.6 Diagramas de Clases del Análisis. Sección de Selección.

Figura 22: Diagrama de Clases del Análisis del CU Mover Multiselección.

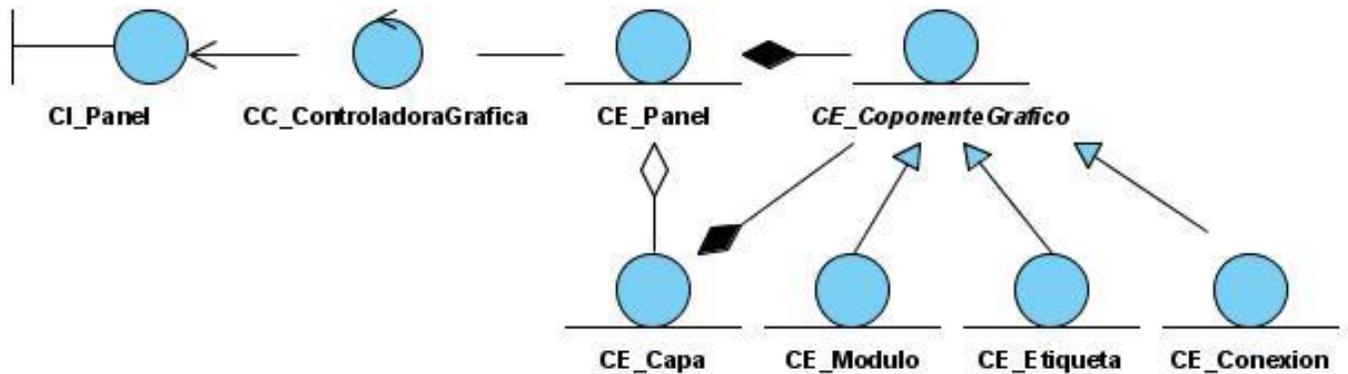


Figura 23: Diagrama de Clases del Análisis del CU Multiseleccionar.

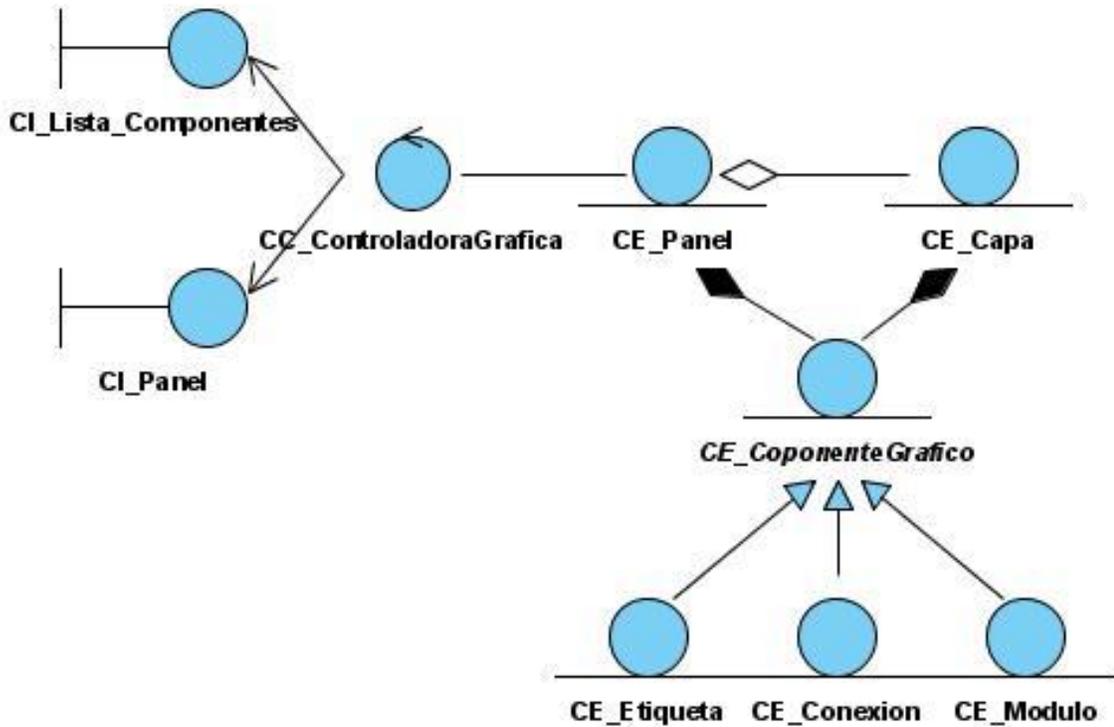
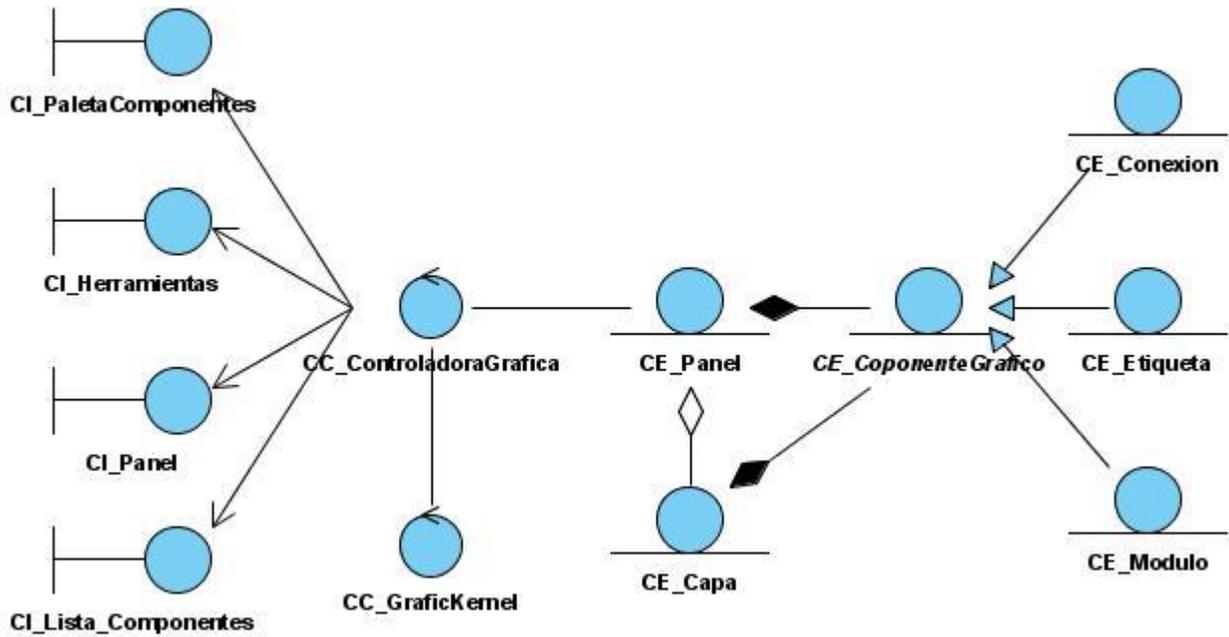


Figura 24: Diagrama de Clases del Análisis del CU Eliminar Multiselección.



3.3.7 Diagramas de Clases del Análisis. Sección de Gestión del DFI

Figura 25: Diagrama de Clases del Análisis del CU Exportar Diagrama

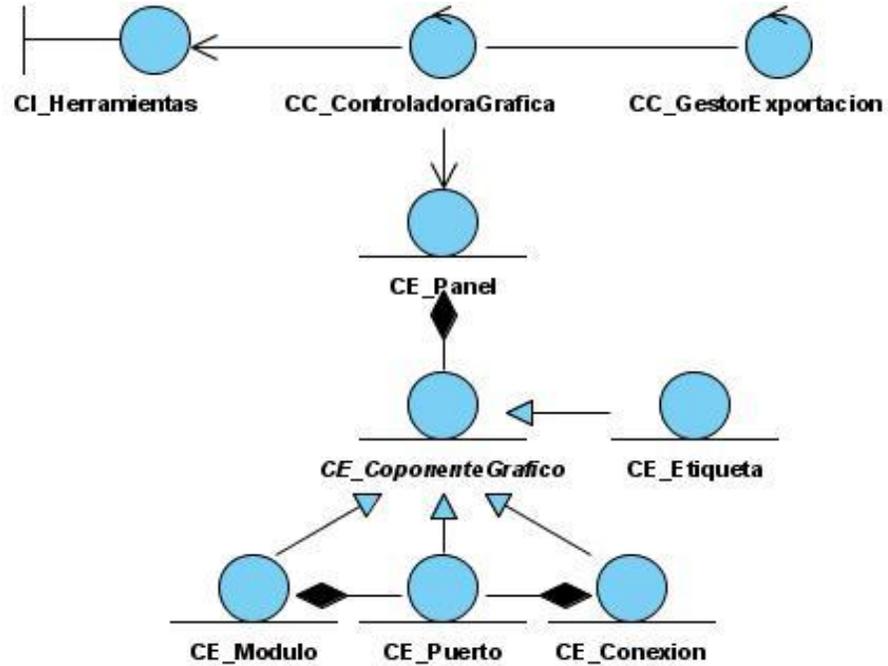


Figura 26: Diagrama de Clases del Análisis del CU Realizar Zoom.

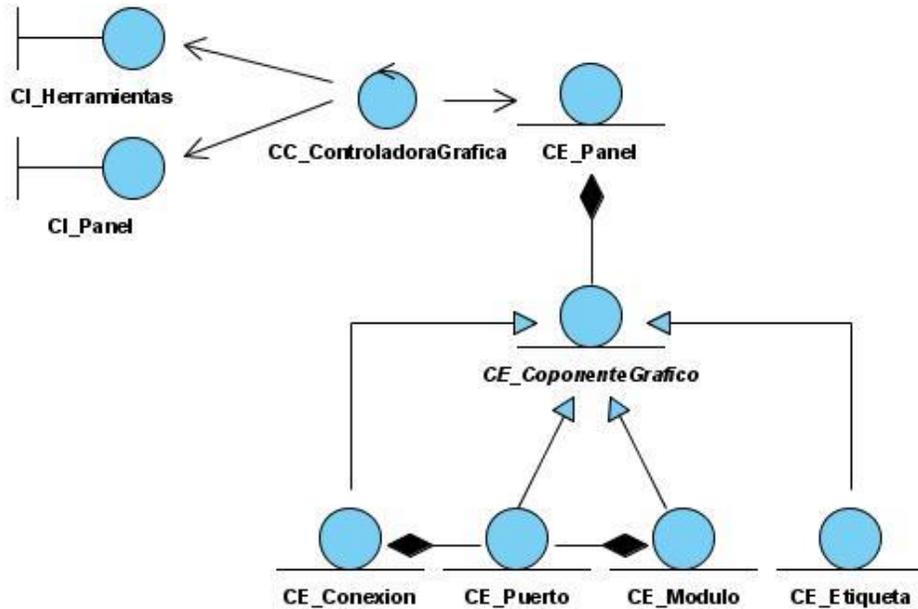


Figura 27: Diagrama de Clases del Análisis del CU Deshacer Última Acción.

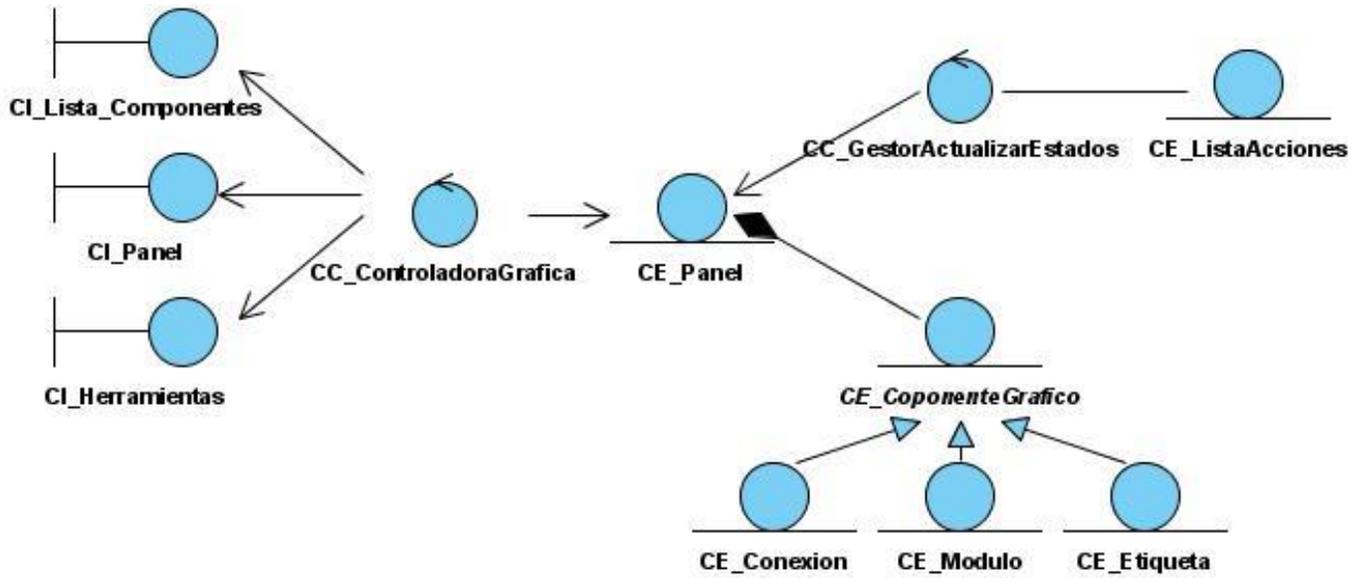


Figura 28: Diagrama de Clases del Análisis del CU Cortar Componentes.

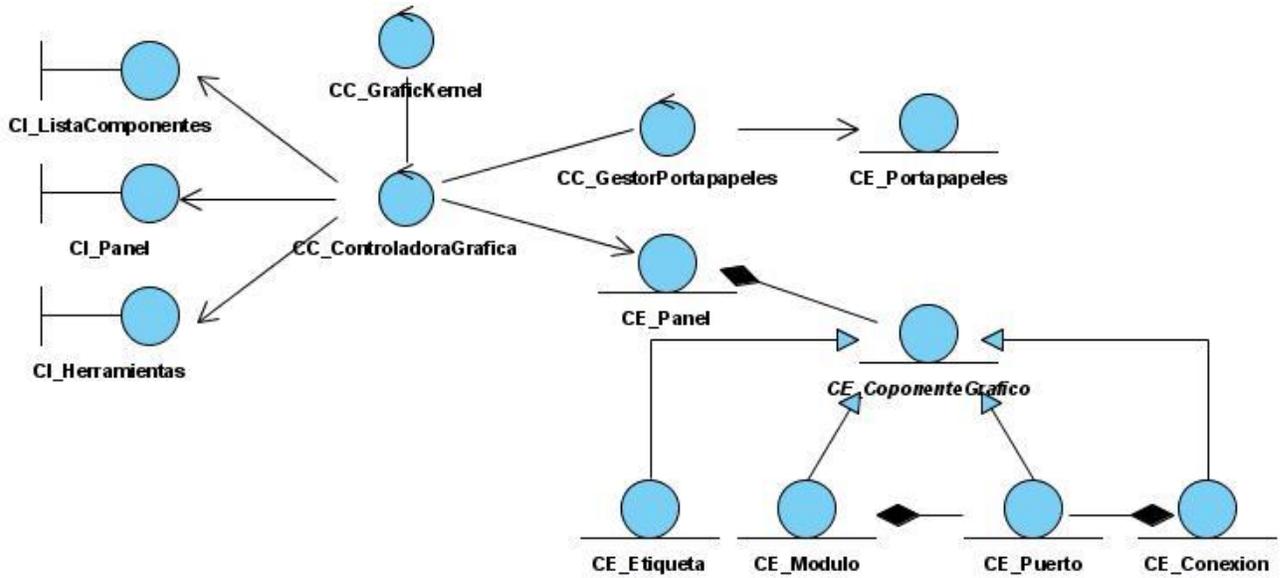
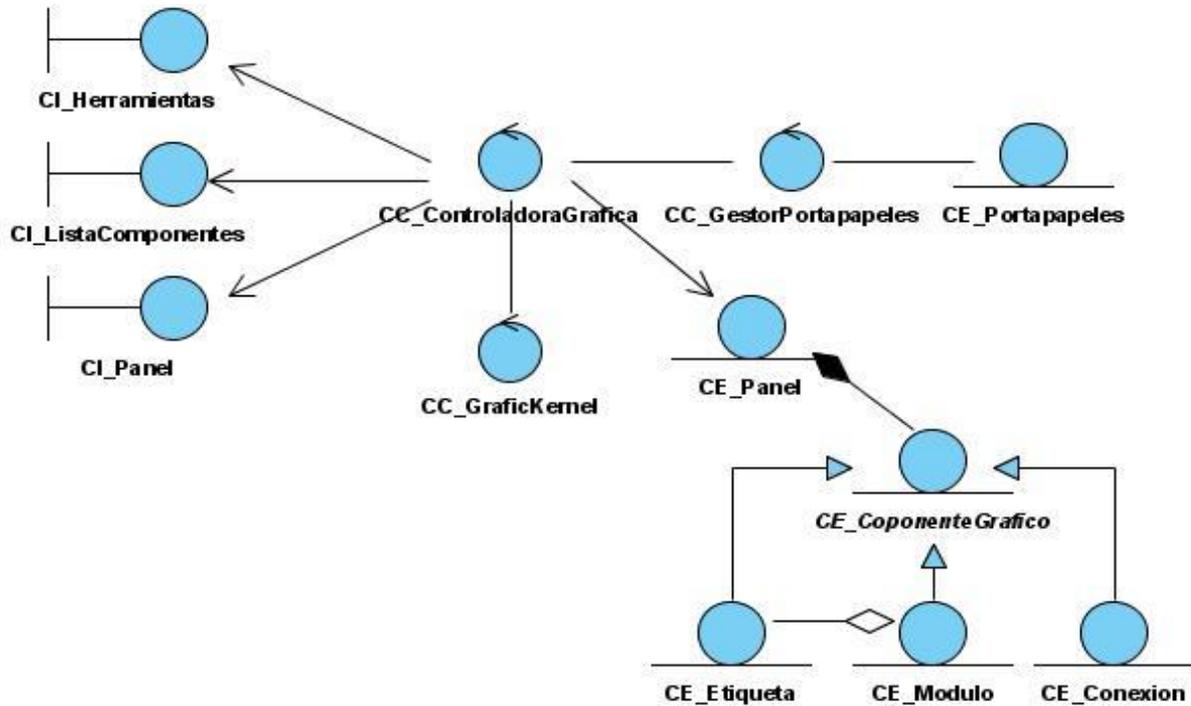


Figura 29: Diagrama de Clases del Análisis del CU Pegar Componentes.



3.4 Modelo de Diseño.

El diseño es un refinamiento del análisis, donde se tienen en cuenta los requisitos no funcionales. Este sirve como puente entre el modelo de análisis y el de implementación. Entre los propósitos del diseño se puede destacar la adquisición de una comprensión de los aspectos relacionados con los requisitos no funcionales, y restricciones relacionadas con los lenguajes de programación, componentes reutilizables, sistemas operativos, tecnologías de distribución y concurrencia y tecnologías de interfaz de usuario. Además de comprender a su vez la descomposición de los trabajos de implementación en partes más manejables, que puedan ser llevadas a cabo por diferentes equipos de desarrollo y tratando de lograr un nivel lo más alto posible de reutilización. Para lograr este último objetivo y resolver los problemas que surgen a lo largo del diseño se suele buscar solución a estos en los Patrones de Diseño.

3.4.1 Patrones de Diseño.

Ante los problemas comunes existentes en el desarrollo del software y otros ámbitos referentes al diseño de interacción o interfaces, los patrones de diseño (*design patterns*) son la base para la búsqueda de soluciones. Un patrón de diseño es una solución a un problema de diseño. Para que una solución sea

considerada un patrón debe poseer ciertas características. Una de ellas es que debe haber comprobado su efectividad resolviendo problemas similares en ocasiones anteriores. Otra es que debe ser reusable, lo que significa que es aplicable a diferentes problemas de diseño en distintas circunstancias.

En el diseño de la solución propuesta se contemplaron los patrones que se explican a continuación.

Patrones Generales de Software para Asignar Responsabilidades (GRASP).

Los patrones GRASP describen los principios fundamentales de diseño de objetos para la asignación de responsabilidades y de cierto modo ayudan a determinar las clases que estarán en el diseño, sus responsabilidades y relaciones con el resto. A continuación se ofrece una relación de los GRASP aplicados en el diseño de la propuesta de solución y el objetivo específico que cumple cada uno de ellos dentro de esta:

1. Experto:

Este es un patrón que se usa más que cualquier otro al asignar responsabilidades; es un principio básico que suele utilizarse en el diseño orientado a objetos. Asignando una responsabilidad determinada a la clase o “experto en información” que contiene todos los datos necesarios y suficientes para desempeñar dicha responsabilidad.

Beneficios:

- Se conserva el encapsulamiento, ya que los objetos se valen de su propia información para hacer lo que se les pide. Esto soporta un bajo acoplamiento, lo que favorece el hecho de tener sistemas más robustos y de fácil mantenimiento.
- El comportamiento se distribuye entre las clases que cuentan con la información requerida, alentando con ello definiciones de clases "sencillas y más cohesivas que son fáciles de comprender y de mantener. Así se brinda soporte a una alta cohesión.

2. Creador:

Este patrón como su nombre lo indica, guía la asignación de responsabilidades relacionadas con la creación de objetos. Asignando la responsabilidad de que una clase B cree un Objeto de la clase A solamente cuando:

- B contiene a A.

- B es una agregación (o composición) de A.
- B almacena a A.
- B tiene los datos de inicialización de A (datos que requiere su constructor)
- B usa a A.

A la hora de crear objetos se deben tener en cuenta las características específicas de la clase.

Beneficios:

- Se brinda soporte a un bajo acoplamiento, lo cual supone menos dependencias respecto al mantenimiento y mejores oportunidades de reutilización.

3. Bajo Acoplamiento:

El acoplamiento es la medida de cuánto una clase está conectada (tiene conocimiento) de otras clases por lo que éste patrón es evaluativo.

Beneficios:

- Un bajo acoplamiento permite que el diseño de clases sea más independiente reduciendo el impacto de los cambios y aumentando la reutilización.

4. Controlador:

Asigna la responsabilidad de controlar o coordinar el flujo de eventos del sistema, a clases específicas o especializadas, manteniendo un modelo de alta cohesión. Esto facilita la centralización de actividades (validaciones, seguridad, etc.). Un error muy común es asignarle demasiada responsabilidad al controlador y alcanzando un alto nivel de acoplamiento con el resto de los componentes del sistema.

En el Proceso Unificado de desarrollo de Software, al construir el modelo de análisis, existen estereotipos predefinidos que favorecen la separación de las responsabilidades en entidades, interfaces y controladoras.

Existen varios tipos de controladores asociados a este patrón. Entre ellos se encuentran los controladores de fachada. Estos representan al sistema global, dispositivo o subsistema. El otro tipo de controlador que se puede encontrar son los controladores de casos de uso. Estos constituyen

construcciones artificiales que dan soporte al sistema y se utilizan cuando los controladores de fachada conducen a diseños de baja cohesión o alto acoplamiento.

Patrones GOF.

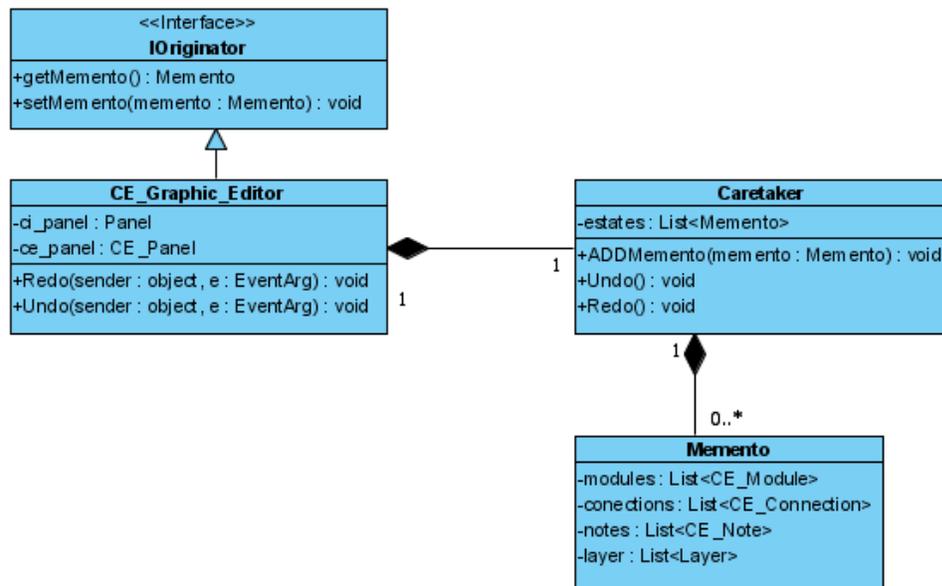
A continuación se ofrece una relación de los patrones GOF aplicados en el diseño de la propuesta de solución y el objetivo específico que cumple cada uno de ellos dentro de esta:

1. Memento:

Este patrón es el encargado de capturar el estado de los componentes que conforman el diagrama ante cualquier acción realizada por el Especialista, para soportar las operaciones de deshacer acciones y rehacerlas, brindando la posibilidad al usuario de poder corregir errores accidentales.

En este patrón intervienen las siguientes clases:

Figura 30: Clases Involucradas en el Patrón Memento.



La clase Memento es la encargada de almacenar los componentes modificados dentro del diagrama en la última acción realizada por el Especialista. La clase CE_Graphic_Editor implementa la interfaz IOriginator, definiendo los métodos getMemento(): Memento y setMemento(): void.

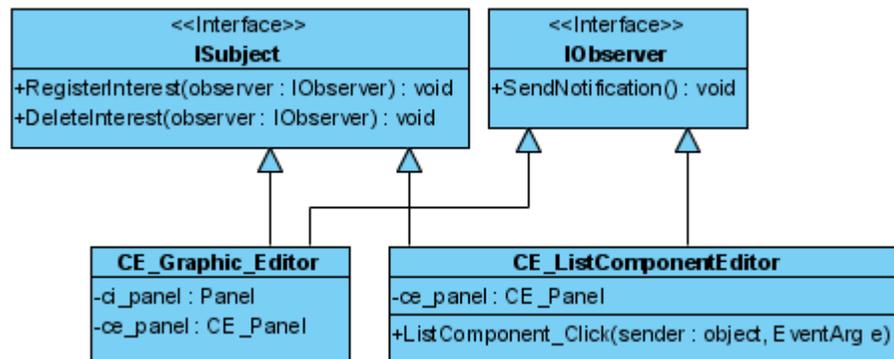
Encargados de salvar el estado y cambiarlo. La clase Caretaker sería la encargada de almacenar la colección de los diferentes estados por los cuales han pasado todos los componentes durante la edición del DFI. Para lograr un mejor rendimiento del software se hace necesario que en la implementación de la operación ADDMemento(memento: Memento): void de la clase Caretaker, sea tomado en cuenta que no es necesario salvar todos los componentes del diagrama, sino solos los que han sido modificados. Esto ahorraría considerablemente memoria y ayudaría al rendimiento del subsistema.

2. Observador:

Este patrón es aplicado ante la necesidad de mantener actualizados varios objetos que dependen de uno o varios datos. Estos objetos dependientes son llamados “observadores”, ya que están pendientes a cualquier modificación que ocurra en el objeto “sujeto”, que sería el encargado de manipular los datos que los observadores necesitan. No existe razón alguna por la cual el número de observadores tenga que ser restringido a dos, pueden utilizarse cuantos sean necesarios. Cada uno de ellos será notificado cuando el sujeto sea objeto de alguna modificación. Este comportamiento se conoce como Suscripción-Notificación, ya que previamente los observadores deberán suscribir su interés de sincronización al sujeto, este notificara a cada uno de los observadores interesados cuando ocurra alguna modificación en sus datos.

En este trabajo el patrón fue aplicado en las siguientes clases:

Figura 31: Clases involucradas en el Patrón Observador



Las clases `CE_Graphic_Editor` y `CE_ListComponentEditor` ambas tienen que mantener sus atributos `ce_panel: CE_Panel` actualizados por los que cualquier modificación en el estado de alguno de estos, debe ser informado a la otra clase. Con este objetivo ambas implementan las dos interfaces `ISubject` e `IObserver`.

3.5 Diagrama de Clases del Diseño.

Las clases de diseño se especifican utilizando la sintaxis del lenguaje de programación elegido y tienen correspondencia directa con los métodos en la implementación. Un diagrama de clases de diseño es una representación más concreta que el diagrama de clases del análisis, representa la parte estática del sistema, las clases que lo componen, atributos y métodos de estas y sus relaciones. A continuación se muestra el Diagrama General de Clases del Diseño.

3.6 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, en el contexto de un escenario que ilustra un comportamiento para dar soporte a una funcionalidad determinada. En el contexto de las clases, los diagramas de interacción describen la forma en que grupos de objetos colaboran para proveer un comportamiento. Mientras que un diagrama de casos de uso presenta una visión externa del sistema, la funcionalidad de dichos casos de uso se recoge como un flujo de eventos utilizando para ello interacciones entre sociedades de objetos.

Cada caso de uso puede presentar varios escenarios primarios (flujo normal del caso de uso) y secundarios (flujos excepcionales y alternativos). Por tanto, para un caso de uso podemos definir diferentes instancias (escenarios) que nos ayudan a la identificación de objetos, clases e interacciones entre objetos necesarios para llevar a cabo la parte de la funcionalidad que especifica el caso de uso. Existen dos tipos de diagramas de interacción:

- Diagramas de secuencia.
- Diagramas de colaboración.

Un diagrama de secuencia es un diagrama de interacción que destaca la ordenación temporal de los mensajes; un diagrama de colaboración es un diagrama de interacción que destaca la organización estructural de los objetos que envían y reciben mensajes.

Se deberían utilizar diagramas de interacción cuando se quiere analizar el comportamiento de varios objetos dentro del mismo caso de uso, resultando apropiados para mostrar colaboraciones entre objetos.

A continuación se muestra una representación de los Diagramas de secuencia de los casos de uso más importantes divididos en sus respectivas secciones.

3.6.1 Sección de Gestión de Módulo.

Figura 32: Diagrama de Secuencia del CU Seleccionar Módulo.

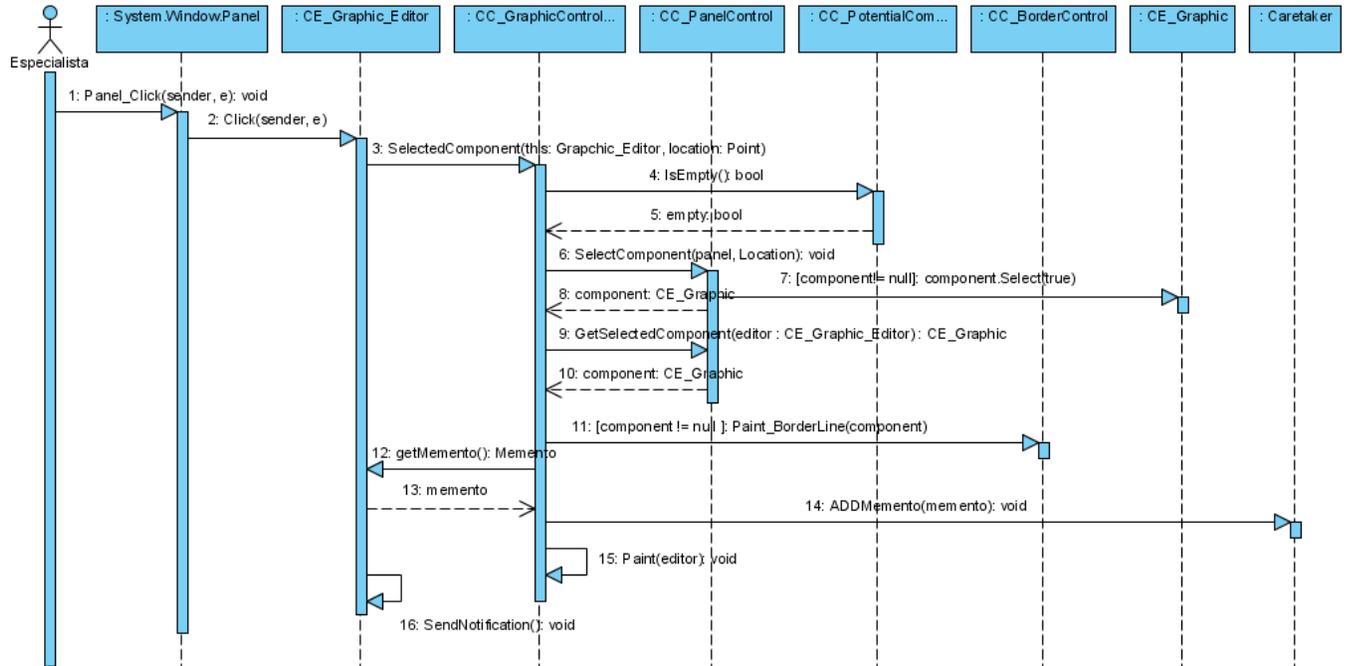


Figura 33: Diagrama de Secuencia del CU Ocultar Módulo.

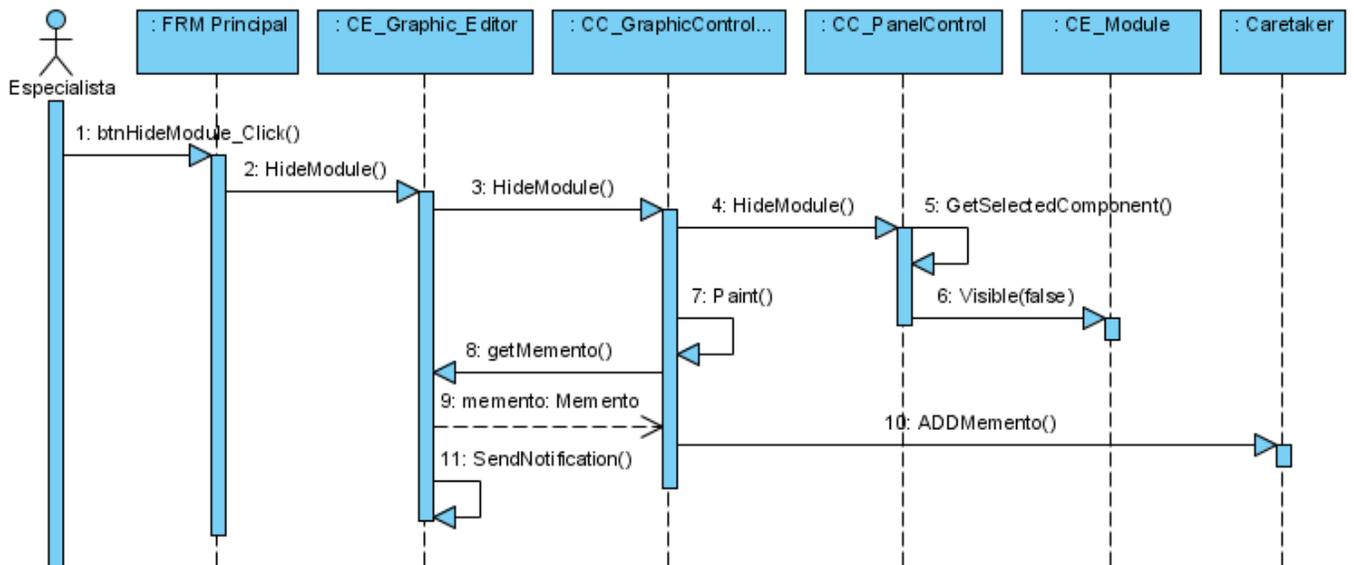


Figura 34: Diagrama de Secuencia del CU Mover Módulo.

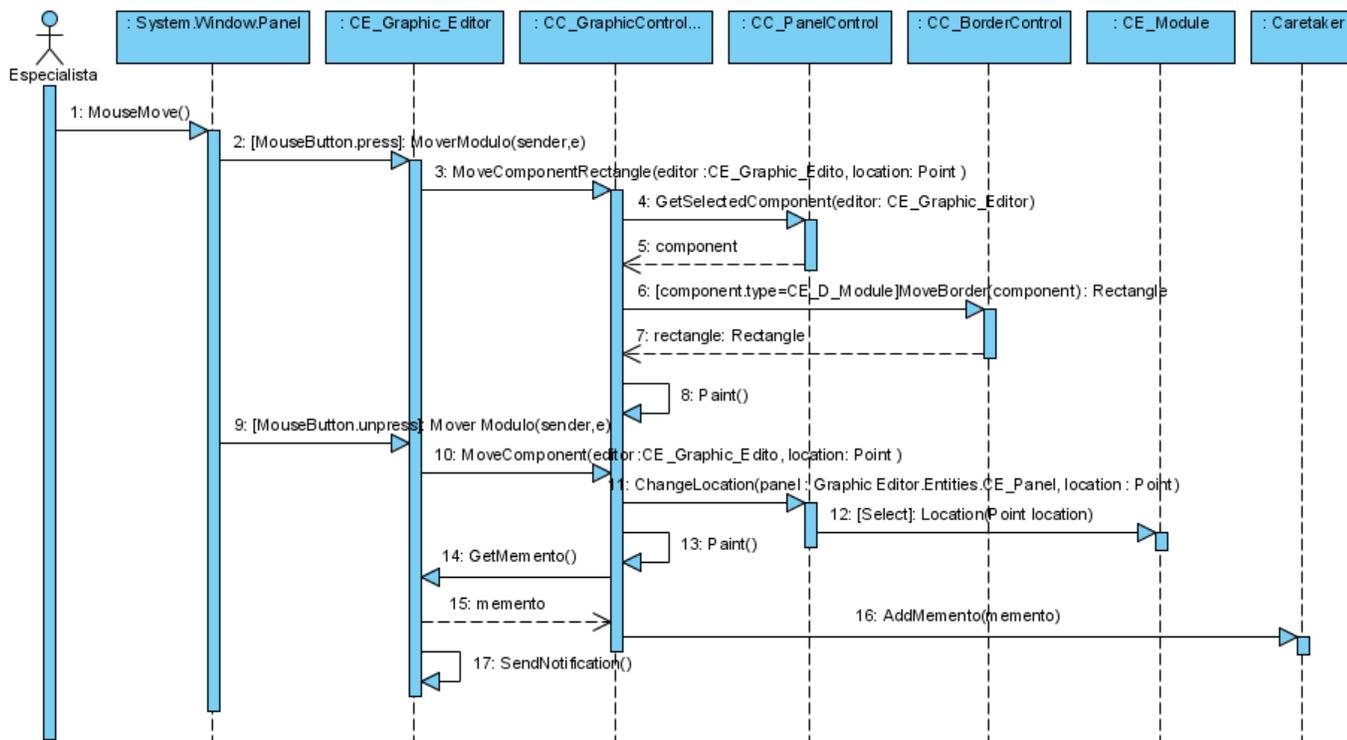
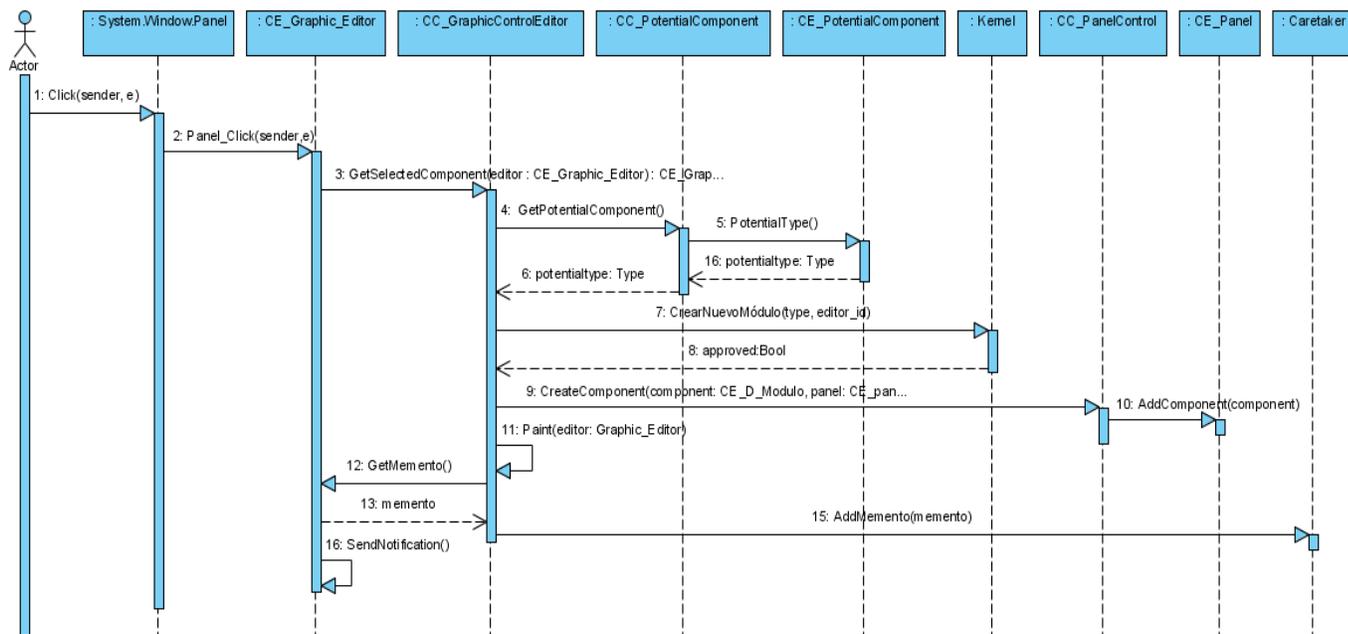
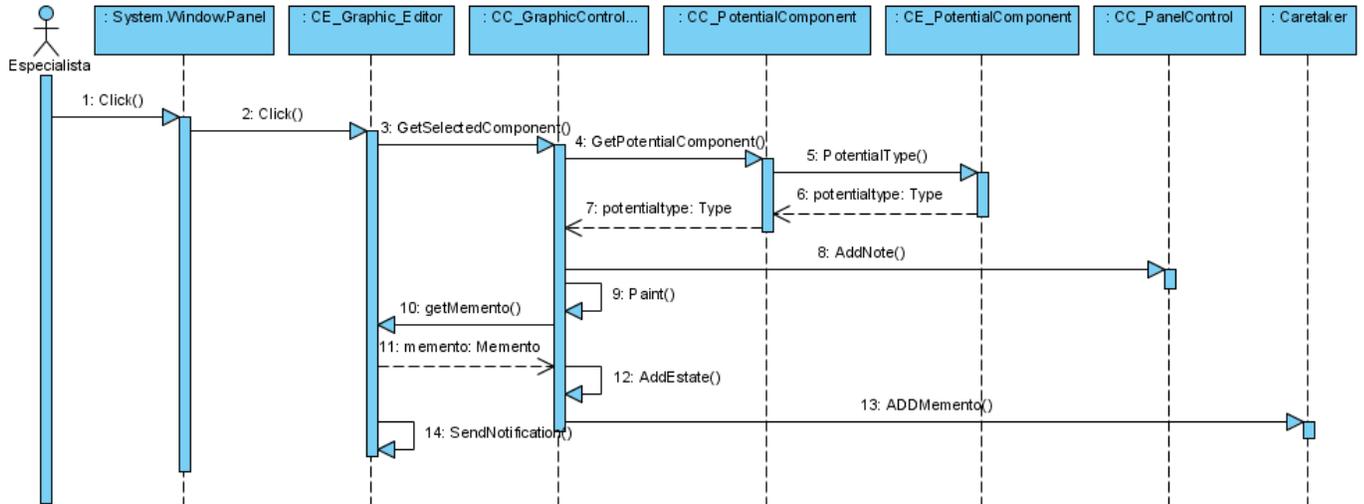


Figura 35: Diagrama de Secuencia del CU Dibujar Módulo



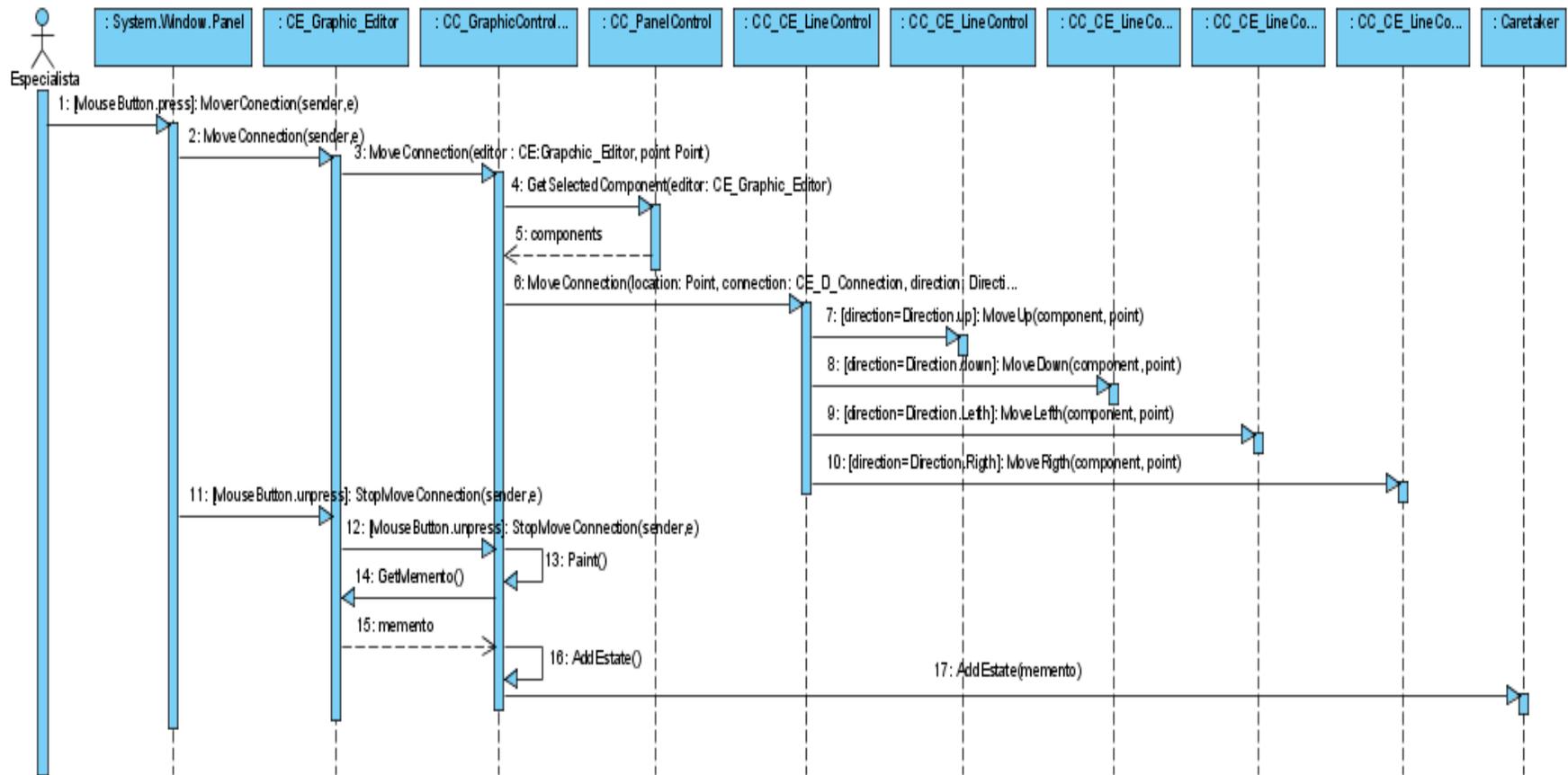
3.6.2 Sección de Gestión de Etiqueta.

Figura 36: Diagrama de Secuencia del CU Dibujar Etiqueta.



3.6.3 Sección de Gestión de Conexión.

Figura 37: Diagrama de Secuencia del CU Crear Conexión.



3.6.4 Sección de Gestión de Capa.

Figura 38: Diagrama de Secuencia del CU Agrupar en Capa.

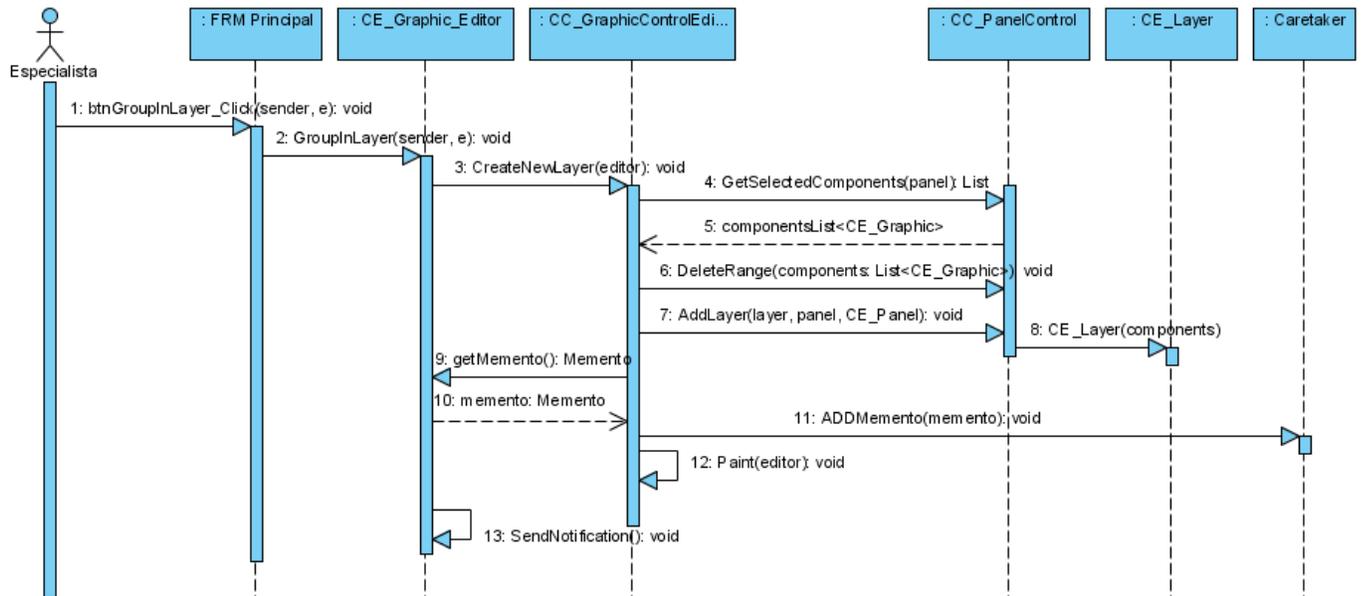


Figura 39: Diagrama de Secuencia del CU Bloquear Capa

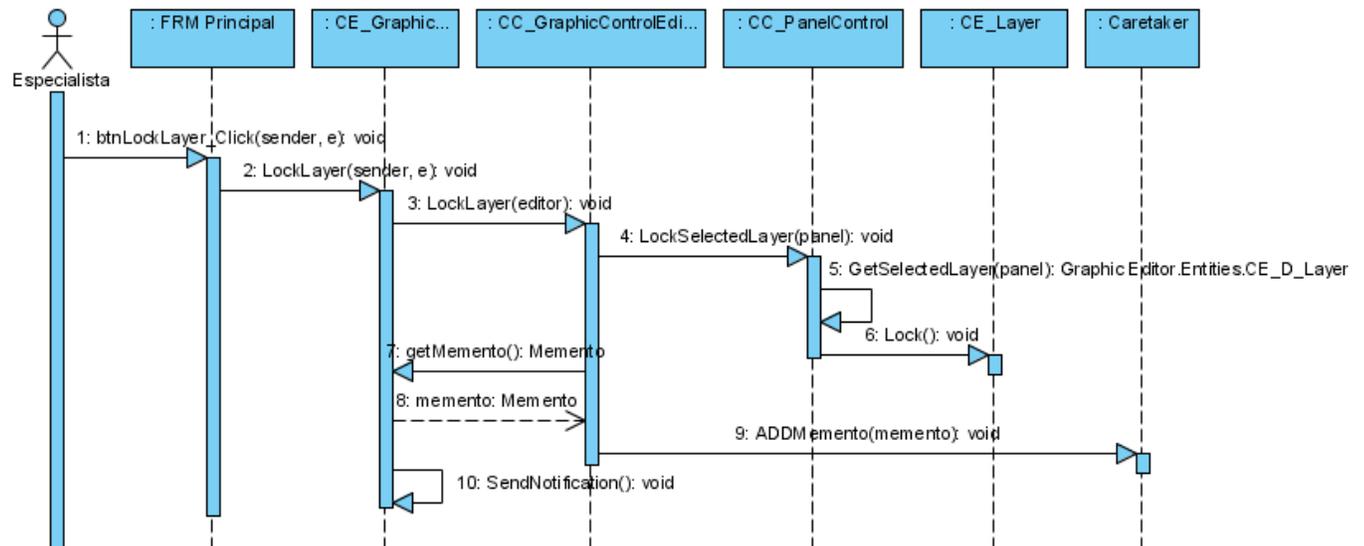


Figura 40: Diagrama de Secuencia del CU Ocultar Capa

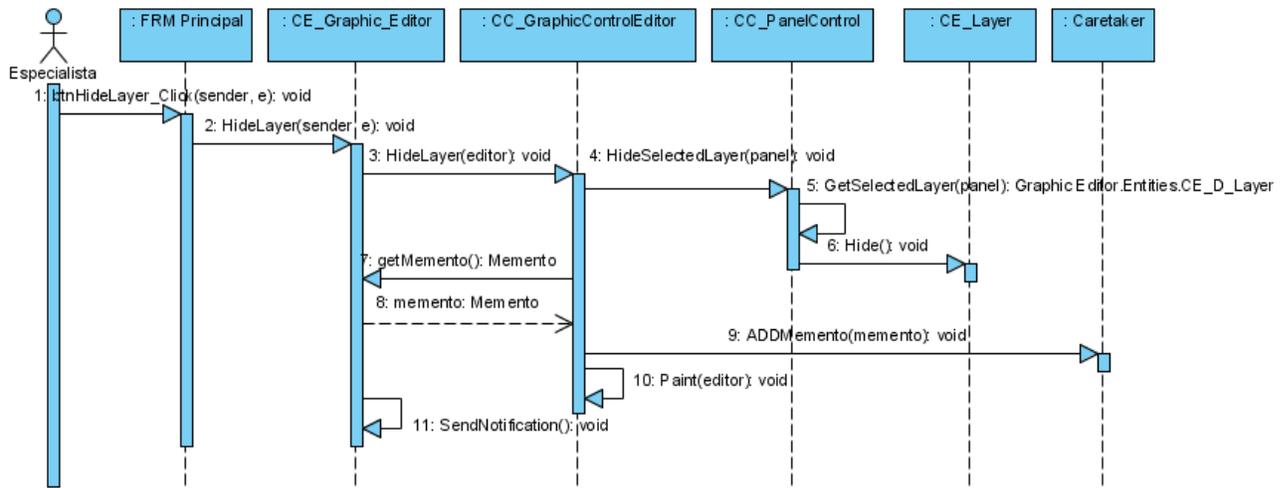


Figura 41: Diagrama de Secuencia del CU Mover Capa.

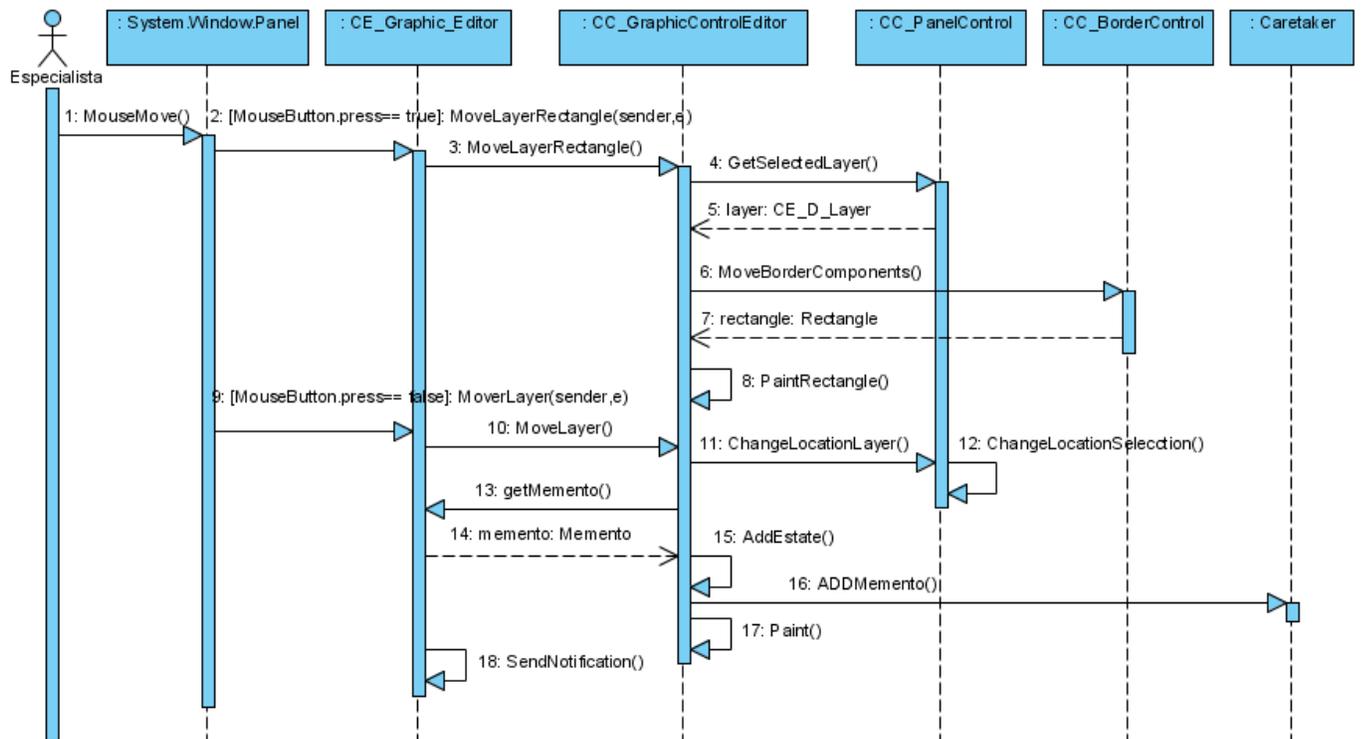
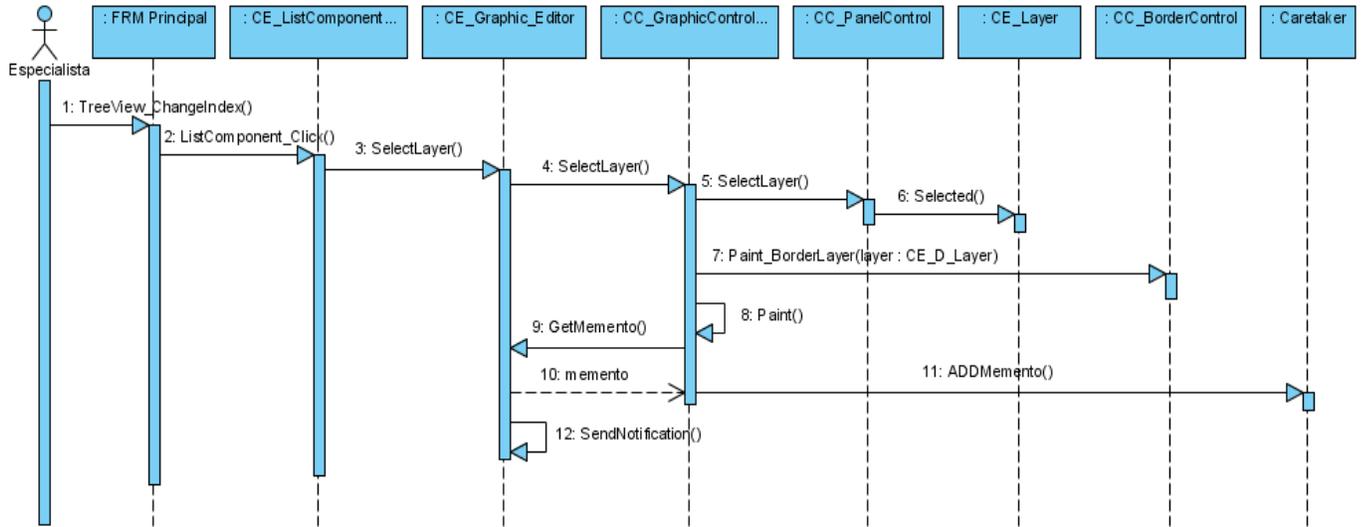


Figura 42: Diagrama de Secuencia del CU Seleccionar Capa.



3.6.5 Sección de Selección.

Figura 43: Diagrama de Secuencia del CU Mover Multiselección.

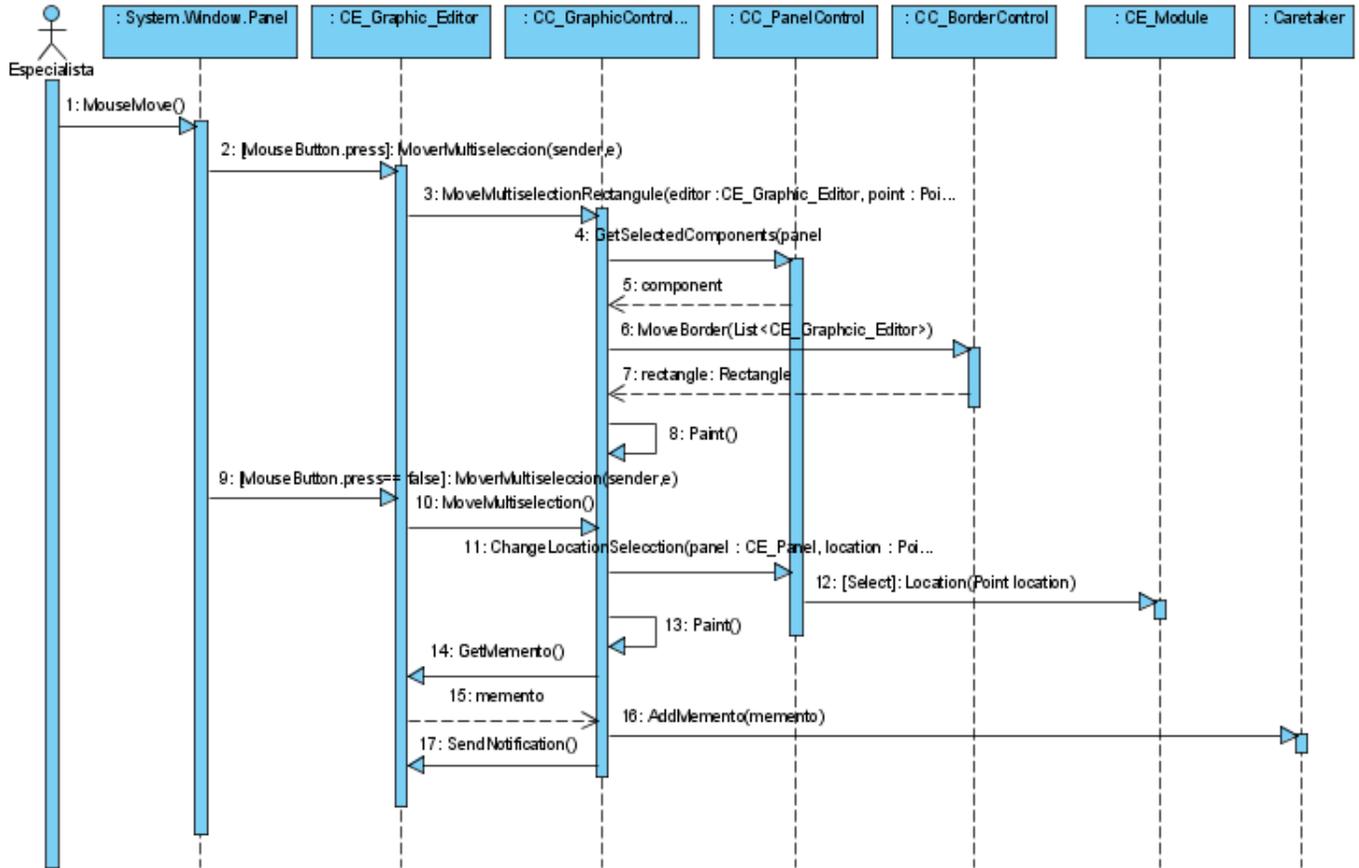


Figura 44: Diagrama de Secuencia del CU Multiseleccionar.

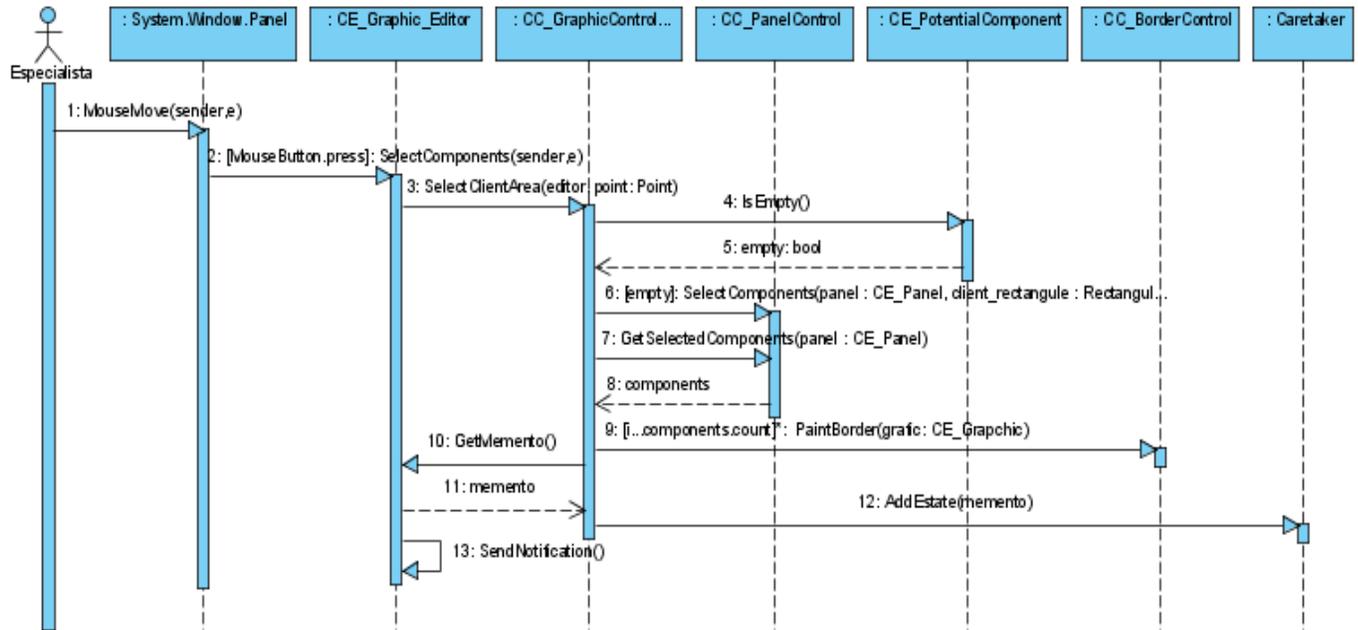
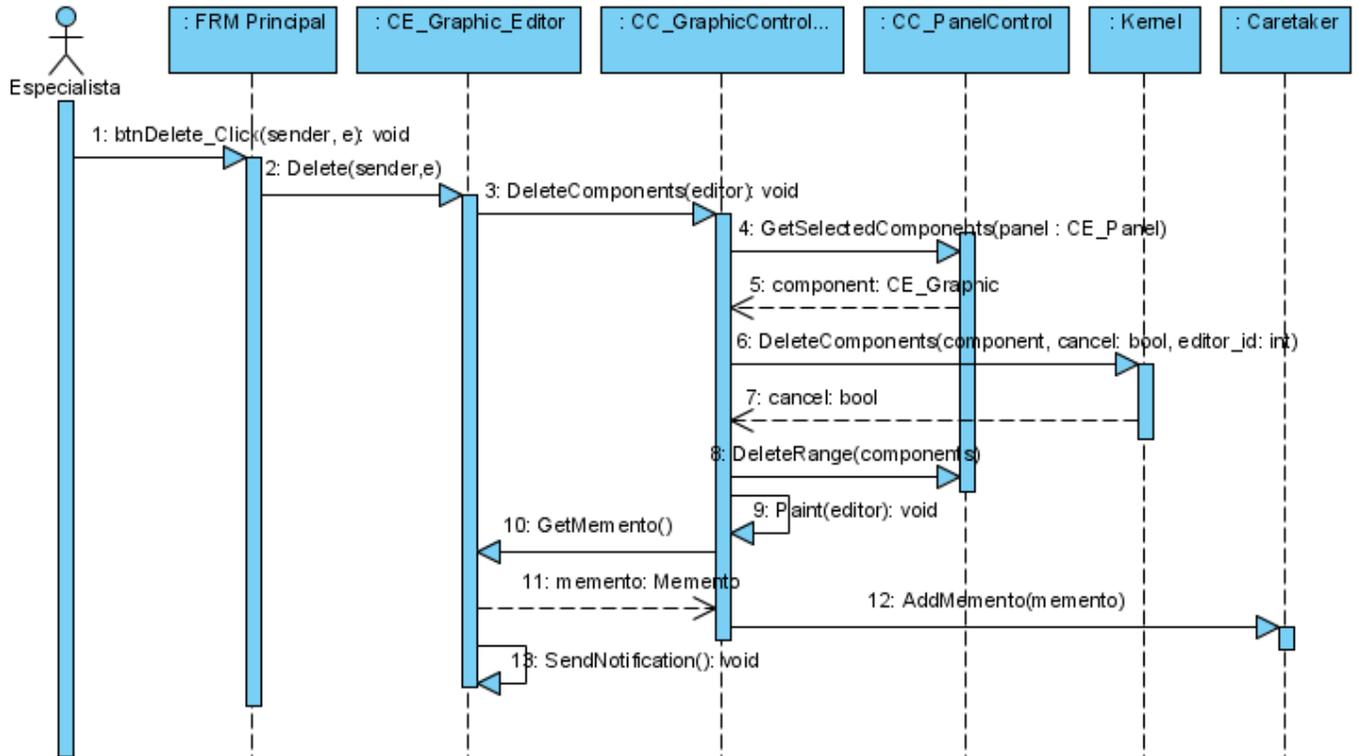


Figura 45: Diagrama de Secuencia del CU Eliminar Multiselección.



3.6.6 Sección de Gestión del DFI.

Figura 46: Diagrama de Secuencia del CU Cortar Componentes.

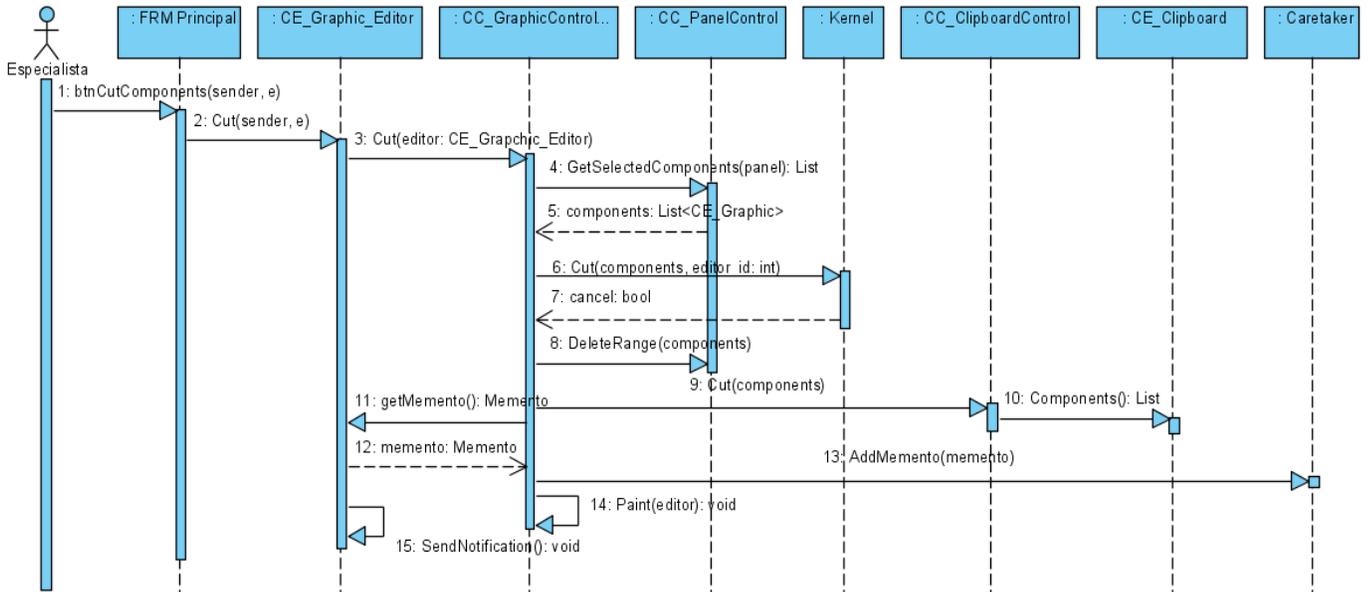


Figura 47: Diagrama de Secuencia del CU Deshacer Última Acción.

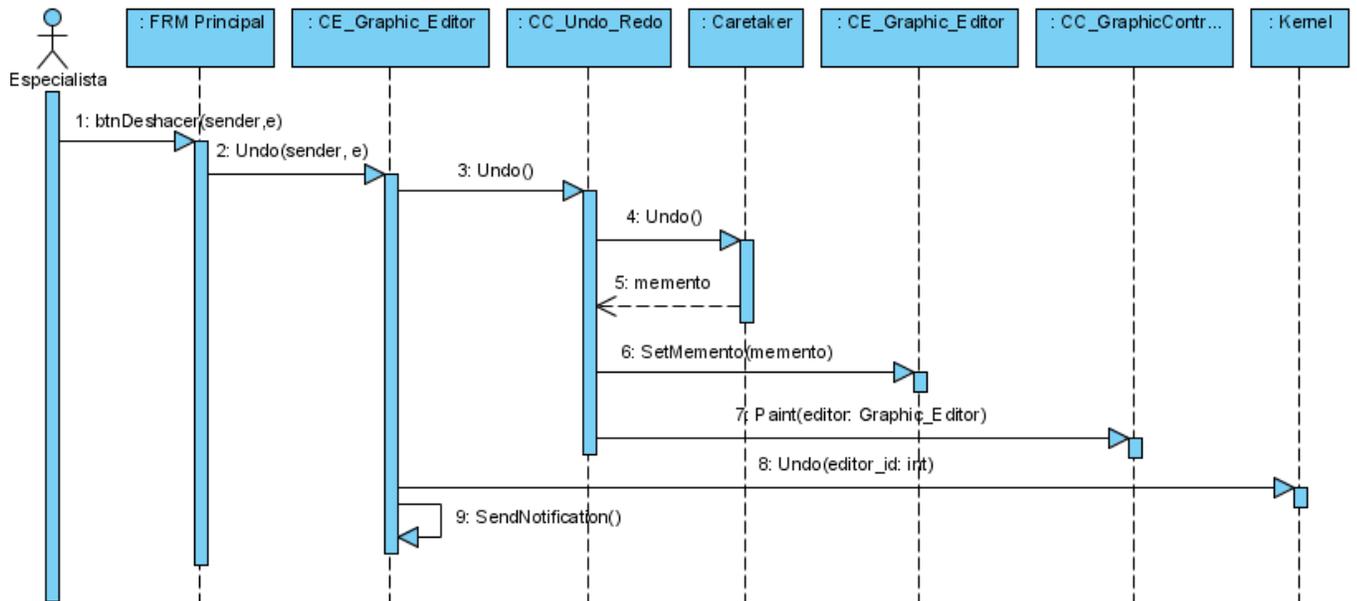


Figura 48: Diagrama de Secuencia del CU Pegar Componentes.

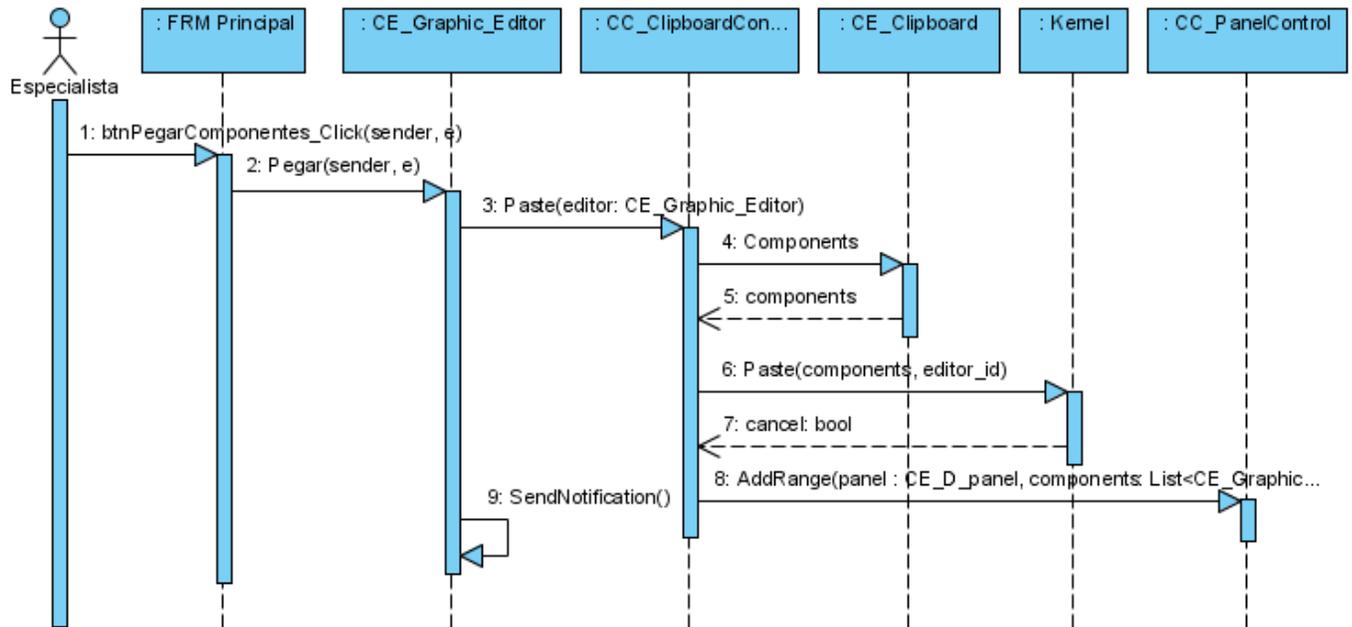


Figura 49: Diagrama de Secuencia del CU Realizar Zoom.

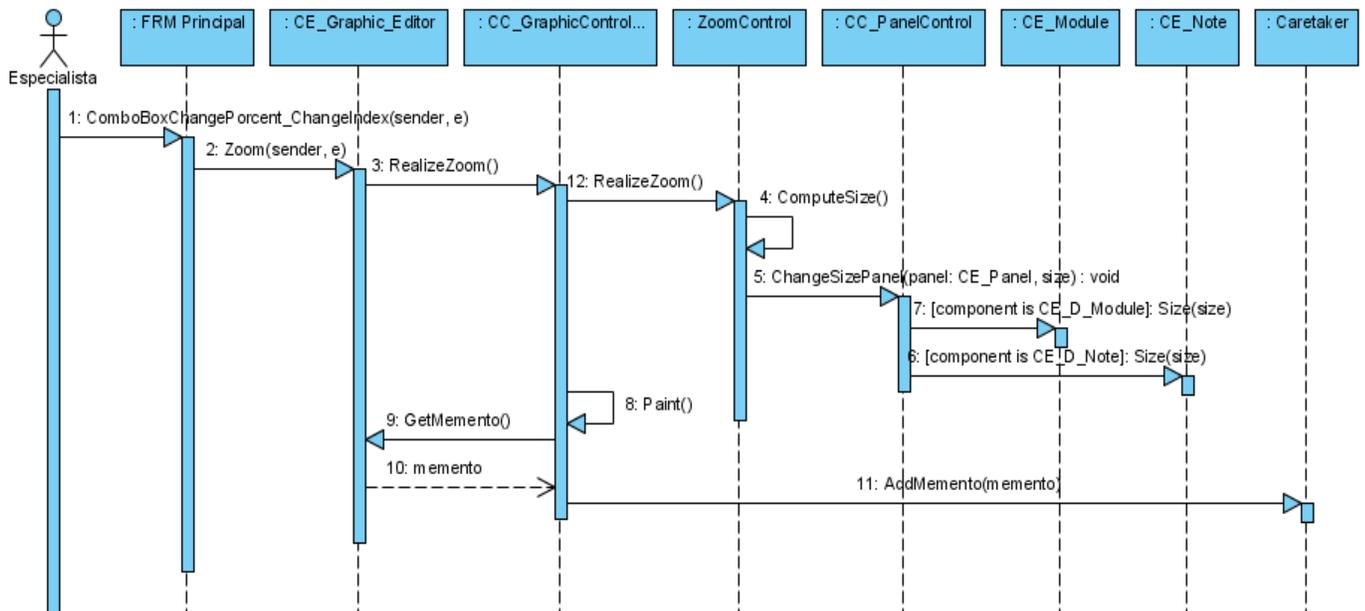
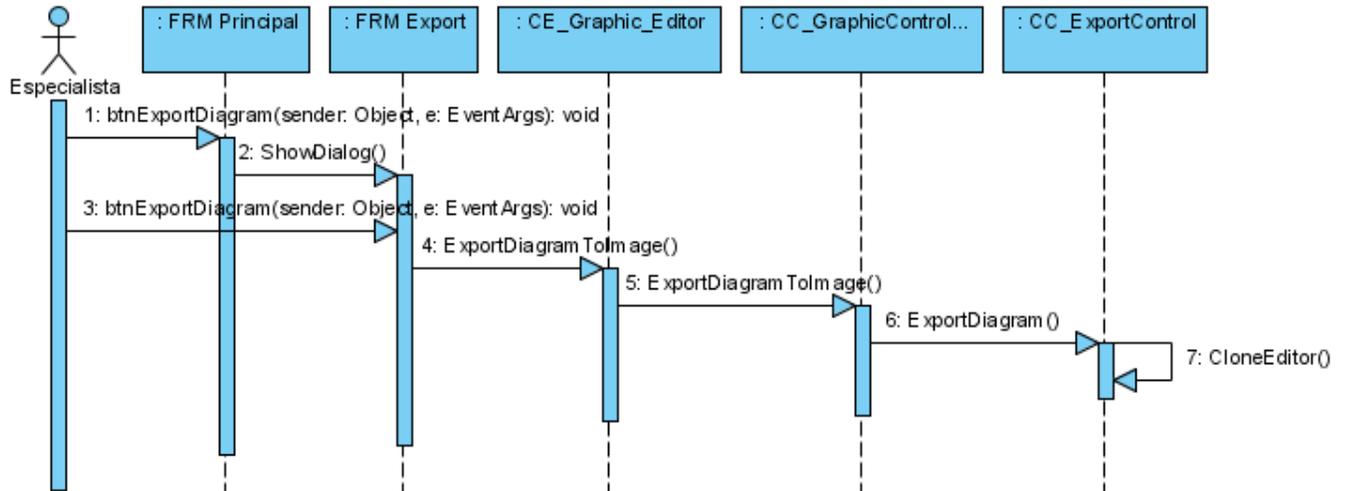


Figura 50: Diagrama de Secuencia del CU Exportar Diagrama.



3.7 Conclusiones.

En este capítulo se presentaron los modelos de análisis y diseño, presentando la estructura principal del Subsistema de Interfaz Gráfica DFISim, mostrando los diagramas de clases del Análisis y del Diseño y la interacción entre los objetos pertenecientes a estas, siguiendo los Patrones de Diseño GRASP y alguno de los Patrones GOF.

CONCLUSIONES GENERALES

La investigación realizada permitió arribar a las siguientes conclusiones:

- El estudio y análisis de varios simuladores de procesos comerciales y específicos disponibles hoy en el mercado internacional y en la UCI, sirvieron de apoyo para entender cuáles fueron las funcionalidades necesarias para incluir en la herramienta a diseñar y dar respuesta a la problemática existente, explotando así, algunas oportunidades en el mercado.
- Se definieron los requerimientos del sistema, tanto funcionales como no funcionales, y a continuación se elaboró el modelo de casos de uso del sistema, describiéndose cada caso de uso para un mejor conocimiento de las funcionalidades que brindan.
- Se logró el análisis y diseño del Subsistema de Interfaz Gráfica de Usuario para el tratamiento y modelado de Diagramas de Flujo de Información, lográndose así el cumplimiento del objetivo principal del presente trabajo de diploma.

RECOMENDACIONES

Se recomienda para investigaciones futuras:

- Realizar los restantes flujos de trabajo que propone la metodología utilizada, RUP, llegando a implementar las funcionalidades que se han propuesto y logrado un refinando del diseño actual a lo largo de ese proceso.
- Realizar una investigación que permita llegar a la conclusión de cuán factible puede ser implementar el subsistema en un lenguaje de programación orientado a objeto bajo la licencia de software libre.
- Aplicar alguna métrica de calidad que pueda servir para evaluar la calidad de los diagramas de casos de uso y las especificaciones de requisitos.

TRABAJOS CITADOS

1. Suárez López, J.C. and Y. Corrales Valdés, *Desarrollo de la arquitectura del Sistema para la Simulación de Procesos en la Industria Química Cubana*. 2007, Universidad de las Ciencias Informáticas: Cuba.
2. Banks, J., *Handbook of simulation: Principles, Methodology, Advances, Applications, and Practice*, ed. J.W. Sons. 1998, New York.
3. Sifuentes, M.n. and V. Hugo, *Simulación de procesos en ingeniería química*. 2000, México: Plaza y Valdés.
4. Pérez Navarro, L., *Conducir, tirar y volar en realidad virtual*. Granma, 2007. **No. 159**.
5. Himmelblau, D.M. and K.B. Bischoff, *Análisis y simulación de procesos*. 1976, Barcelona: Reverté.
6. Law, A. and D. Kelton, *Simulation Modeling and Analysis*. Vol. Tercera Edición. 2004: McGraw Hill.
7. Forgionne, G.A., *Corporate Management Science Activities: An Update Interfaces*. Vol. 13. 1978. 257-260.
8. Rasmussen, J.J. and T. George, *After 25 Years: A Survey of Operations Research Alumni*. Case Western Reserve University Interfaces, 1978. **Vol 8**: p. 48-52.
9. Thomas , G. and J. DaCosta, *A Sample Survey Of Corporate Operations Research*, in *Interfaces*. 1978. p. 48-52.
10. Shannon, R.E., S.S. Long, and B.P. Buckles, *Operations Research Methodologies Industrial Engineering*. AIIE Trans., 1981. **vol. 12**: p. 364-367.
11. M. Sánchez, J.M.G., A. F. Gómez, H. Martínez, *Generación de simuladores eficientes para procesos complejos basados en Arquitecturas Distribuidas*. X JORNADAS DE PARALELISMO, LA MANGA DEL MAR MENOR, 1999.

12. Booch, G., J. Rumbaugh, and I. Jacobson, *El Lenguaje Unificado de Modelado*. 1999: Addison Wesley. 432.
13. Jacobson, I., G. Booch, and J. Rumbaugh, *El proceso unificado de desarrollo de software*. 2006: Addison Wesley.
14. *Patrones de diseño*, in *Conferencia de Ingeniería de Software*. 2005-2006: UCI.
15. PRESSMAN, R.S., *Ingeniería de Software, un enfoque práctico*. Addison Wesley Professional ed. 1998.

BIBLIOGRAFÍA CONSULTADA

1. Durán Toro, A.R.C., R. Corchuelo Gil, M. Toro Bonilla, *Identificación de Patrones de Reutilización de Requisitos de Sistemas de Información*, in *Departamento de Lenguajes y Sistemas Informáticos, Facultad de Informática y Estadística*. 2000, Universidad de Sevilla: Sevilla.
2. David M. Raffo, W.W., *Moving Up the CMMI Capability and Maturity Levels Using Simulation*, ed. S.E.M.a. Analysis. 2008.
3. GARCÍA, I.G., *SIMULACIÓN HÍBRIDA COMO NÚCLEO DE SIMULACIÓN*
4. *DE APLICACIONES GRÁFICAS EN TIEMPO REAL*. 2004, UNIVERSITAT DE VALENCIA: València.
5. González, Y. *SIMULACIÓN Y MODELAJE HERRAMIENTAS PARA MEJORAR EL PROCESO Y EL NEGOCIO*. in *Foro Rueda de Negocios CNTQ*. 2007. Valencia: Universidad Nacional Experimental Politécnica (UNEXPO) “Antonio José de Sucre” Vice-rectorado Varquisimeto Centro de Investigación de Procesos (CENIPRO).
6. Jay April , M.B., Fred Glover, James P. Kelly, Manuel Laguna, *Enhancing Business Process Management With Simulation Optimization*. Enero 2005: OptTek Systems, Inc.
7. José Manuel Rodríguez Barrios, D.S., Toni Monleón, Jaime Caro. *Los modelos de simulación de eventos discretos en la evaluación económica de tecnologías y productos sanitarios*. 2008 [cited 2009-05-20]; Available from:
http://scielo.isciii.es/scielo.php?script=sci_arttext&pid=S0213-91112008000200012&lng=es&nrm=iso.
8. L. J. Enstonem, M.F.C., *BPMN and Simulation*, in *Lanner Group. Product Discussion Paper*. April 2006.
9. Luis R. Izquierdo, J.M.G., José I. Santos, Ricardo del Olmo, *Modelado de sistemas complejos mediante simulación basada en agentes y mediante dinámica de sistemas*. EMPIRIA.

10. M. Sánchez, J.M.G., A. F. Gómez, H. Martínez, *Generación de simuladores eficientes para procesos complejos basados en Arquitecturas Distribuidas*. X JORNADAS DE PARALELISMO, LA MANGA DEL MAR MENOR, 1999.
11. Meléndrez, E.H., *METODOLOGÍA DE LA INVESTIGACIÓN, Cómo escribir una tesis*, ed. E.N.d.S. Pública. 2006.
12. Reinier García Ramos, R.H.L. *SIMULADOR DE MICROSCOPIO ÓPTICO SIMOPTIC*. in "VII Congreso Internacional de Informática en Salud". 2008.
13. Riveros, J.C.G., *Modelo Computacional de Simulación de Procesos Constructivos "MOCSPROC"*. 2003.
14. Torres, J.L. *Especificación de requisitos en Ingeniería de Software*. 2006 [cited 2008 27 de Noviembre]; Available from: <http://www.uag.mx/ieee/contsep01/requerimientos.htm>.
15. Villareal, D., *Simulación y Optimización de Procesos Discretos y Continuos: Estado del Arte y Tendencias*. 2000: México.