

**UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS**

**Facultad 5**



**TÍTULO: Ingeniería Inversa del Módulo Diseño e Instalación  
de una Red de Área Local (LAN)**

**TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO  
DE INGENIERO EN CIENCIAS INFORMÁTICAS**

**AUTOR: Yanelky Hernández Farray**

**TUTORES: Ing. Sailyn Salas Hechavarria.**

**Ciudad de la Habana, septiembre de 2011  
“Año 53 de la Revolución”**

# *Declaración de Autoría*

## **DECLARACIÓN DE AUTORÍA**

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

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

Yanelky Hernández Farray

Ing. Sailyn Salas Hechavarria

---

(Firma del autor)

---

(Firma del tutor)

# *Datos de Contacto*

## **DATOS DE CONTACTO**

### **Tutor:**

Ing. Sailyn Salas Hechavarria

Universidad de las Ciencias Informáticas, Habana, Cuba.

Graduada de Ingeniera en Informática en el año 2008 y profesora instructora con tres años de experiencia docente en la Universidad de Ciencias Informáticas y ocho en la producción de software.

Email: [ssalas@uci.cu](mailto:ssalas@uci.cu)

# *Agradecimientos*

## **AGRADECIMIENTOS**

A la revolución y a Fidel por hacer posible una Universidad de excelencia.

A mis padres por apoyarme y ayudarme en todo momento y por hacerme la mujer que soy hoy.

A mi hermano que es mi vida por estar presente siempre que lo necesito.

A mi novio por su apoyo y comprensión.

A mi familia por ser la familia más linda del mundo.

A mi tutora por ser tan paciente y comprensiva.

A mis compañeros de grupo por estar siempre presentes.

A mi tribunal de tesis por guiarme por el camino correcto para realizar una tesis mejor.

A todos los que de una manera u otra ayudaron al desarrollo de esta investigación, lleguen mis agradecimientos más sinceros.

# *Dedicatoria*

## **DEDICATORIA**

### **A mis padres**

Clara y Yoel por haberme apoyado en todo momento y por ser las personas más importantes en mi vida. Los quiero mucho y por ello le dedico esta tesis.

### **A mi hermano**

Yoerky por ser la persona que es, por quererme tanto, por ser una de las personas que más quiero en este mundo.

### **A mi novio**

Rolando por apoyarme y ayudarme siempre que lo necesitaba, por hacerme la mujer más feliz de este mundo.

### **A mis familiares**

A mis cuatro abuelitos por ser tan lindas personas conmigo. A mis todos mis primos, especialmente Michel y Maikel por estar siempre presentes en los momentos buenos y malos. A todos mis tíos, especialmente Efrén y Sofía por ser tan cariñosos y buenas personas.

### **A mis amigos**

A todas las personas con que he compartido estos cinco años de carrera y que una manera u otra me han hecho sentir bien.

**Yanelky Hernández Farray**

## RESUMEN

El desarrollo alcanzado en las tecnologías informáticas ha dado paso a la creación de laboratorios virtuales en todo el mundo. Cuba también se ha trazado la meta de avanzar en la creación de estos, por lo que en la Universidad de las Ciencias Informáticas (UCI) surge el proyecto “Prolavi<sup>1</sup>”, con el propósito de realizar prácticas de laboratorio que servirán de apoyo al aprendizaje en diferentes centros de enseñanza, ejemplo de ello el módulo “Diseño e instalación de una red de área local (LAN)”, el cual no cuenta con los artefactos ingenieriles requeridos que soporten lo implementado.

Por esta razón en el presente trabajo de diploma se realiza una investigación sobre los principales conceptos relacionados con los laboratorios virtuales y la ingeniería inversa, que sirven de base para un mejor entendimiento de la situación a modelar. Como resultado del trabajo se obtienen los diagramas correspondientes al análisis y diseño del sistema donde se visualizan los detalles empleados en el mismo. Se realizan los diagramas de componentes para visualizar los elementos físicos del sistema y sus relaciones. Además se validan los requisitos funcionales a partir de la matriz de trazabilidad.

**Palabras clave:** Artefactos ingenieriles, ingeniería inversa, laboratorio virtual, red de área local.

---

<sup>1</sup> Proyecto Laboratorios Virtuales.

# *Índice de Contenido*

## **ÍNDICE DE CONTENIDO**

<b>Capítulo 1: “Fundamentación Teórica”</b> .....	<b>15</b>
Introducción.....	15
1.1 Laboratorio virtual .....	15
1.1.1 Ventajas y desventajas de los laboratorios virtuales .....	15
1.2 Red de área local (LAN) .....	16
1.2.1 Componentes de una LAN .....	16
1.3 Ingeniería inversa .....	19
1.3.1 Tipos de Ingeniería Inversa .....	19
1.3.2 Sistemas a los que se le aplica la Ingeniería Inversa .....	20
1.3.3 Proceso de ingeniería inversa .....	20
1.3.4 Herramientas para la Ingeniería Inversa.....	21
1.3.5 Ventajas y desventajas de la Ingeniería Inversa .....	22
1.4 Análisis de soluciones existente de laboratorios virtuales de redes .....	22
1.5 Metodología, Lenguaje, Herramienta de modelado.....	24
1.5.1 Rational Unified Procces (RUP).....	24
1.5.2 Lenguaje Unificado de Modelado (UML).....	25
1.5.3 Visual Paradigm .....	26
Consideraciones parciales.....	26
<b>Capítulo 2: “Características del sistema”</b> .....	<b>27</b>
Introducción.....	27
2.1 Modelo del dominio.....	27
2.2 Descripción del sistema.....	28
2.3 Especificación de los requisitos.....	28
2.3.1 Requisitos funcionales .....	28
2.3.2 Requisitos no funcionales.....	29
2.4 Modelo de casos de uso del sistema.....	30
2.4.1 Descripción del actor del sistema.....	31
2.4.2 Diagrama de casos de uso del sistema .....	31
2.4.3 Descripción de los casos de uso del sistema .....	31
2.5 Modelo del análisis .....	43
2.5.1 Clases del análisis .....	43
2.5.2 Diagramas de clases del análisis .....	44
2.5.3 Diagrama de interacción del análisis (colaboración) .....	45
Consideraciones parciales.....	46
<b>Capítulo 3: “Diseño de la aplicación”</b> .....	<b>47</b>

# *Índice de Contenido*

Introducción.....	47
3.1 Patrones de diseño .....	47
3.2 Arquitectura del sistema .....	49
3.3 Diagrama de paquetes .....	50
3.4 Modelo de diseño.....	51
3.4.1 Diagrama de clases del diseño .....	51
3.5 Modelo de Implementación .....	58
3.5.1 Diagrama de componentes .....	58
3.6 Diagrama de despliegue.....	62
3.7 Matriz de trazabilidad.....	62
Consideraciones parciales.....	64
<b>Conclusiones generales .....</b>	<b>65</b>
<b>Bibliografía Referenciada .....</b>	<b>67</b>
<b>Bibliografía Consultada .....</b>	<b>70</b>
<b>Glosario de Términos.....</b>	<b>71</b>
<b>Anexos .....</b>	<b>72</b>



# Índice de Figuras

## ÍNDICE DE FIGURAS

Figura 1: Servidor. (6) .....	16
Figura 2: Estación de trabajo. (6) .....	17
Figura 3: Puentes. (6) .....	17
Figura 4: Tarjeta de red. (6) .....	17
Figura 5: El medio. (6).....	18
Figura 6: Proceso Unificado de Desarrollo de Software. (17).....	25
Figura 7: Vocabulario de UML. (20).....	25
Figura 8: Visual Paradigm. (24) .....	26
Figura 9: Modelo del dominio.....	27
Figura 10: Diagrama de casos de uso del sistema.....	31
Figura 11 Prototipo de interfaz “Registrar” .....	32
Figura 12: Prototipo de interfaz “Mostrar Objetos de la Escena”.....	35
Figura 13: Prototipo de interfaz “Seleccionar Objeto Gráfico”.....	36
Figura 14: Prototipo de interfaz “Mostrar Tiempo de Duración de la Práctica”.....	37
Figura 15: Prototipo de interfaz “Mostrar Errores”.....	37
Figura 16: Prototipo de interfaz “Salvar Archivo”.....	38
Figura 17: Prototipo de interfaz “Eliminar Objeto Gráfico”.....	39
Figura 18: Prototipo de interfaz “Limpiar Escena”.....	40
Figura 19: Prototipo de interfaz “Agrupar Objetos Gráficos”.....	41
Figura 20: Prototipo de interfaz “Desagrupar Objetos Gráficos”.....	42
Figura 21: Prototipo de interfaz “Exportar Escena”.....	42
Figura 22 : Diagrama de clases del análisis del CU Registrar.....	44
Figura 23: Diagrama de clases del análisis del CU Agrupar Objetos Gráficos.....	44
Figura 24: Diagrama de clases del análisis del CU Mostrar Errores.....	44
Figura 25: Diagrama de clases del análisis del CU Seleccionar Objetos Gráficos.....	45
Figura 26: Diagrama de interacción (colaboración) del análisis del CU Registrar.....	45
Figura 27: Diagrama de interacción (colaboración) del análisis del CU Agrupar Objetos Gráficos.....	45
.....	45
Figura 28: Diagrama de interacción (colaboración) del análisis del CU Mostrar Errores.....	46
Figura 29: Diagrama de interacción (colaboración) del análisis del CU Seleccionar Objetos Gráficos.....	46
.....	46
Figura 30: Uso del patrón Singleton.....	49
Figura 31: Arquitectura tres capas. (28).....	50
Figura 32: Diagrama de paquetes.....	50
Figura 33: Diagrama de clases del diseño CU Registrar.....	51

# *Índice de Figuras*

Figura 34: Diagrama de Clases del diseño del CU Agrupar Objetos Gráficos.....	52
Figura 35: Diagrama de Clases del diseño del CU Mostrar Errores.....	52
Figura 36: Diagrama de clases del diseño del CU Seleccionar Objetos Gráficos. ....	53
Figura 37: Diagrama de Clases del diseño del CU Desagrupar Objetos Gráficos.....	53
Figura 38: Diagrama de clases del diseño del CU Eliminar Objetos Gráficos. ....	54
Figura 39: Diagrama de Clases del diseño del CU Exportar Escena.....	54
Figura 40: Diagrama de clases del diseño del CU Limpiar Escena.....	55
Figura 41: Diagrama de Clases del diseño del CU Mostar Tiempo de Duración de la Práctica.	55
Figura 42: Diagrama de Clases del diseño del CU Mostrar Objetos de la Escena.....	56
Figura 43: Diagrama de Clases del diseño del CU Salvar Archivos.....	57
Figura 44: Diagrama de componentes del CU Registrar. ....	58
Figura 45: Diagrama de componente del CU Agrupar Objetos Gráficos. ....	58
Figura 46: Diagrama de componente del CU Desagrupar Objetos Gráficos. ....	59
Figura 47: Diagrama de componente del CU Eliminar Objetos Gráficos. ....	59
Figura 48: Diagrama de componente del CU Exportar Escena.....	59
Figura 49: Diagrama de componente del CU Limpiar Escena.....	60
Figura 50: Diagrama de componente del CU Mostrar Objetos de la Escena.....	60
Figura 51: Diagrama de componente del CU Salvar Archivos. ....	60
Figura 52: Diagrama de componente del CU Seleccionar Objetos Gráficos. ....	61
Figura 53: Diagrama de componente del CU Mostrar Tiempo de Duración de la Práctica. ....	61
Figura 54: Diagrama de componente del CU Mostrar Errores. ....	61
Figura 55: Diagrama de despliegue.....	62
Figura 56: Diagrama de clases del análisis del CU Desagrupar Objetos Gráficos.....	72
Figura 57: Diagrama de clases del análisis del CU Exportar Escena. ....	72
Figura 58: Diagrama de clases del análisis del CU Limpiar Escena. ....	73
Figura 59: Diagrama de clases del análisis del CU Mostrar Objetos de la Escena. ....	73
Figura 60: Diagrama de clases del análisis del CU Mostrar tiempo de Duración de la Práctica.	73
Figura 61: Diagrama de clases del análisis del CU Salvar Archivo. ....	73
Figura 62: Diagrama de clases del análisis del CU Eliminar Objeto Gráfico.....	74
Figura 63: Diagrama de colaboración del CU Desagrupar Objetos Gráficos.....	74
Figura 64: Diagrama de colaboración del CU Exportar Escena. ....	74
Figura 65: Diagrama de colaboración del CU Limpiar Escena. ....	75
Figura 66: Diagrama de colaboración del CU Mostrar Objetos de la Escena.....	75
Figura 67: Diagrama de colaboración del CU Mostrar Tiempo de Duración de la Práctica.....	75
Figura 68: Diagrama de colaboración del CU Salvar Archivo.....	76
Figura 69: Diagrama de colaboración del CU Eliminar Objeto Gráfico. ....	76

# *Índice de Tablas*

## **ÍNDICE DE TABLAS**

Tabla 1: Descripción del actor del sistema.....	31
Tabla 2: Especificación del requisito Registrar. ....	32
Tabla 3: Especificación del requisito Mostrar Objetos de la Escena.....	35
Tabla 4: Especificación del requisito Seleccionar Objetos Gráficos.....	36
Tabla 5: Especificación del requisito Mostrar Tiempo de Duración de la Práctica.....	37
Tabla 6: Especificación del requisito Mostrar Errores.....	38
Tabla 7: Especificación del requisito Salvar Archivo.....	38
Tabla 8: Especificación del requisito Eliminar Objeto Gráfico. ....	39
Tabla 9: Especificación del requisito Limpiar Escena.....	40
Tabla 10: Especificación del requisito Agrupar Objetos Gráficos.....	41
Tabla 11: Especificación del requisito Desagrupar Objetos Gráficos.....	42
Tabla 12: Especificación del requisito Exportar Escena.....	42
Tabla 13: Estereotipos de las clases del análisis.....	43
Tabla 14: Matriz de trazabilidad.....	63

## **Introducción**

Las tecnologías de la información y las comunicaciones, actualmente son elementos fundamentales para la superación y desarrollo de un país. Los Sistemas de Información (SI) y las Tecnologías de Información (TI) han cambiado la forma en que operan las organizaciones actuales. A través de su uso se logran importantes mejoras, pues automatizan los procesos operativos y suministran una plataforma de información necesaria para la toma de decisiones.

Cuba enfrenta hoy, una serie de retos que requieren ser superados como parte del proceso de desarrollo en áreas fundamentales como lo son: educación, salud, tecnología, entre otras. Esto hace indispensable el desarrollo de la informática en diferentes sectores de la sociedad buscando así soluciones para estos problemas que, por su trascendencia para el país se considera indispensable solucionar.

La educación es uno de los aspectos más importantes y con mayor prioridad en Cuba, donde siempre se ha trabajado intensamente con el objetivo de alcanzar grandes mejoras en esta rama. Es por ello que se han buscado diferentes métodos de enseñanza que contribuyan al mejoramiento de la misma, como por ejemplo los laboratorios tradicionales para que el alumno pueda interactuar con el medio y poner en práctica sus conocimientos, pero muchas veces estos laboratorios no tienen los recursos necesarios para su uso, por lo que se hace indispensable la búsqueda de nuevas estrategias. La Universidad de las Ciencias Informáticas (UCI) se ha trazado crear laboratorios virtuales para la educación, con el objetivo de erradicar dichos problemas de recursos. En la Facultad 5, una de las principales desarrolladoras de software de la universidad, surge el proyecto Prolavi, donde uno de los módulos que desarrolla es el de “Diseño e instalación de redes de área local (LAN)”. Este módulo que se ha desarrollado principalmente con fines educativos no cuenta con los artefactos ingenieriles como los diagramas de clases, la especificación de casos de uso, los diagramas de secuencias entre otros que avalen el proceso de desarrollo de software sostenido en lo implementado.

Los artefactos ingenieriles de un sistema son de gran importancia ya que estos se pueden utilizar en otros proyectos que compartan situaciones comunes, minimizando en tiempo y trabajo. Sirve de guía para el usuario para el mejor entendimiento de la aplicación y garantiza la recuperación del sistema en caso de que la misma sufra alguna pérdida o fallo.

Por lo expuesto anteriormente se plantea como **problema científico**:

Inexistencia de los artefactos ingenieriles que avalen el proceso de desarrollo de software seguido en el módulo “Diseño e instalación de una red de área local (LAN)”.

El **objeto de estudio de la investigación** es el proceso de ingeniería inversa para software de realidad virtual y se delimita en el **campo de acción**: ingeniería inversa para el modulo “Diseño e Instalación de una red de área local (LAN)”.

# *Introducción*

Para dar solución al problema científico declarado, esta investigación tiene como **objetivo general**: Realizar los artefactos ingenieriles del módulo “Diseño e Instalación de una red de área Local (LAN)” desarrollado en el proyecto Laboratorios Virtuales (Prolavi).

Para dar cumplimiento al objetivo general se han trazado las siguientes **tareas de investigación**:

- Estudio del estado actual de los laboratorios virtuales para la elaboración del marco teórico.
- Estudio del proceso de ingeniería Inversa para seleccionar la vía idónea en la propuesta de solución.
- Estudio de los principales elementos que componen al módulo de diseño e instalación de una red de área local (LAN) para un mejor entendimiento de la situación a modelar.
- Realización de un estudio del negocio para un mejor entendimiento del sistema desarrollado.
- Realización de la captura de requisitos funcionales y no funcionales para plasmar las funcionalidades y restricciones que tiene el sistema desarrollado.
- Realización del análisis y diseño del sistema para plasmar la documentación del sistema desarrollado.
- Realización del modelo de implementación para visualizar los detalles empleados en el desarrollo del sistema.
- Validación de los requisitos funcionales a partir de la matriz de trazabilidad.

Con este trabajo se defiende la idea de que si se realiza la ingeniería inversa del módulo Diseño e Instalación de una Red de Área Local, se dejará documentado cada uno de los artefactos generados durante el proceso de desarrollo de software seleccionado.

Los **Métodos científicos** a utilizar en la investigación son los siguientes:

## **Método empírico<sup>2</sup>:**

**La entrevista:** Se utilizará para recopilar información con el equipo de trabajo que desarrollo el módulo para un mejor entendiendo del sistema.

## **Métodos Teóricos:**

**Histórico-lógico:** El análisis histórico de laboratorios virtuales posibilitará indagar sobre las funcionalidades que estos realizan, lo que permitirá comprenderlos y aumentar los conocimientos sobre el tema.

**Método de la modelación:** Se utilizará para realizar los modelos que son el resultado de la ingeniería inversa de este proyecto.

---

<sup>2</sup> Conocimiento basado en la experiencia, todo lo que sabemos y que lo repetimos continuamente sin tener un conocimiento científico.

# *Introducción*

El trabajo de Diploma está estructurado en **tres Capítulos**:

## **Capítulo 1 Fundamentación teórica:**

Se abordan los conceptos fundamentales para un mejor entendimiento del sistema desarrollado y algunos ejemplos de laboratorios virtuales de redes en Cuba y en el mundo. Se hace referencia a la metodología de desarrollo utilizada, conjunto a la herramienta y el lenguaje de modelado.

## **Capítulo 2 Características del sistema:**

Se hará un estudio del negocio para un mejor entendimiento del sistema desarrollado. Además se realizará la captura de requisitos funcionales y no funcionales para plasmar las funcionalidades y restricciones que tiene el sistema y los diagramas correspondientes al análisis del mismo.

## **Capítulo 3 Diseño de la aplicación:**

Se realizarán los diagramas correspondientes al diseño de la aplicación. Se modelarán los diagramas de componentes, los cuales describen los elementos físicos del sistema y sus relaciones y validarán los requisitos funcionales a partir de una matriz de trazabilidad.

# *Fundamentación Teórica*

## **Capítulo 1: “Fundamentación Teórica”**

### **Introducción**

En el presente capítulo se abordan algunas definiciones importantes que sustentan el trabajo de diploma, como por ejemplo los laboratorios virtuales, los componentes que integran una red de área local (LAN) y la ingeniería inversa. Además se describe la metodología y herramienta de modelado utilizada en el proyecto.

### **1.1 Laboratorio virtual**

Un Laboratorio Virtual es una herramienta pedagógica utilizada principalmente con fines educativos, donde el alumno puede aprender sin la necesidad de contar con recursos materiales. Depende solamente de un ordenador para llevarse a cabo.

Un laboratorio virtual es diferente de un “laboratorio verdadero”. Sin embargo, no se considera que estos vayan a suplantar a los tradicionales o competir con ellos. En cambio, los laboratorios virtuales constituyen una posible extensión de los verdaderos y abren nuevas perspectivas que garantizan una buena práctica, a un costo asequible. (1)

#### **1.1.1 Ventajas y desventajas de los laboratorios virtuales**

El uso de laboratorios virtuales para la educación tiene muchas ventajas:

- Los alumnos aprenden por cuenta propia fomentando la capacidad de análisis, el pensamiento, la utilización de la tecnología informática, etc.
- No supone gasto económico alguno por parte de los departamentos de ciencia para adquirir materiales.
- Evita los desdoblados al tener cada alumno a su disposición todo el material interactivo que necesiten.
- Permite que el profesor analice los resultados desde su ordenador y a cualquier hora del día.
- En el caso de los laboratorios que tengan que ver con productos químicos al no verter estos productos se favorece el medio ambiente.

También se tienen algunos inconvenientes:

- Es necesario que todos los alumnos dispongan de un ordenador personal.
- El centro y las aulas han de disponer de conexión a Internet de banda ancha.
- No tienen en cuenta las ideas de los alumnos durante el proceso de aprendizaje.
- Hay ciertos experimentos que son imposibles de realizar virtualmente.
- Los resultados son menos llamativos para los alumnos perdiendo calidad en la educación. (1)

# Fundamentación Teórica

## 1.2 Red de área local (LAN)

Es la interconexión de varias computadoras y periféricos. Su extensión está limitada físicamente a un edificio o a un entorno de 200 metros, con repetidores podría llegar a la distancia de un campo de 1 kilómetro. Su aplicación más extendida es la interconexión de computadoras personales y estaciones de trabajo en oficinas, fábricas y otros. (2)

Con una LAN, un concepto que se remonta a 1970, los empleados de una compañía pueden:

- intercambiar información.
- comunicarse.
- acceder a diversos servicios.

Por lo general, una red de área local conecta equipos (o recursos, como impresoras) a través de un medio de transmisión cableado (frecuentemente pares trenzados o cables coaxiales) dentro de un perímetro de unos cien metros. Para espacios más grandes, se considera como parte de una red denominada MAN (red de área metropolitana), en la que el medio de transmisión está mejor preparado para enviar señales a través de grandes distancias. (3)

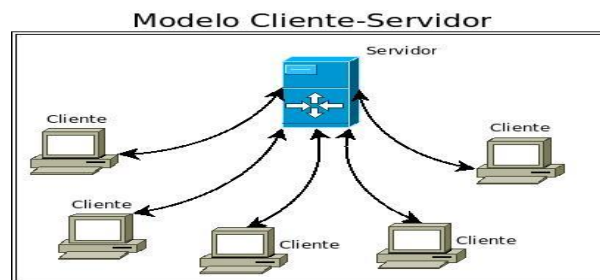
Una LAN es un grupo de equipos que pertenecen a la misma organización y están conectados dentro de un área geográfica pequeña a través de una red, generalmente con la misma tecnología (la más utilizada es Ethernet). (4)

### 1.2.1 Componentes de una LAN

Una red de área local está compuesta por varios componentes, los cuales se describen a continuación:

#### Servidor

Es aquel o aquellos ordenadores que van a compartir sus recursos hardware y software con los demás equipos de la red. Sus características son potencia de cálculo, importancia de la información que almacena y conexión con recursos que se desean compartir. (5)



**Figura 1: Servidor. (6)**



# Fundamentación Teórica

## Estación de trabajo

Los ordenadores que toman el papel de estaciones de trabajo aprovechan o tienen a su disposición los recursos que ofrece la red, así como los servicios que proporcionan los servidores a los cuales pueden acceder. (5)



Figura 2: Estación de trabajo. (6)

## Puentes (bridges)

Es un hardware o software que permite que se conecten dos redes locales entre sí. Un puente interno se instala en un servidor de la red, y uno externo es el que se hace sobre una estación de trabajo de la misma red. Los puentes también pueden ser locales o remotos. Los locales se conectan a redes de un mismo edificio, usando tanto conexiones internas como externas. Los puentes remotos conectan redes distintas entre sí, llevando a cabo la conexión a través de redes públicas, como por ejemplo la red telefónica. (5)

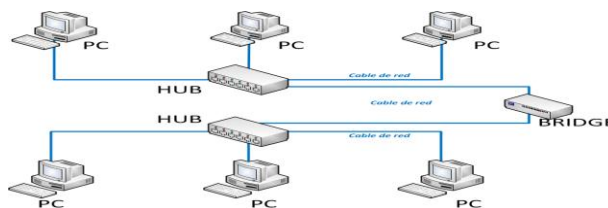


Figura 3: Puentes. (6)

## Tarjeta de red

También denominadas NIC (Network Interface Card), realizan la función de intermediario entre el ordenador y la red. En ella se encuentran grabados los protocolos de comunicación de la red. La comunicación con el ordenador se realiza normalmente a través de las ranuras de expansión que éste dispone. Aunque algunos equipos disponen de este adaptador integrado directamente en la placa base. (5)



Figura 4: Tarjeta de red. (6)

# Fundamentación Teórica

## El medio

Está constituido por el cableado y los conectores que enlazan los componentes de la red. Los medios físicos más utilizados son el cable de par trenzado, cable coaxial y la fibra óptica, siendo esta última la más utilizada. (5)

### Cable de par trenzado:

Consiste en dos alambres de cobre aislados, que se trenzan de forma helicoidal. De esta forma constituye un circuito que puede transmitir datos. (7)

### Cable coaxial:

Está formado por dos conductores metálicos concéntricos separados por un material aislante. El conjunto se encuentra protegido por un material plástico que lo aísla del exterior. El conductor exterior realiza, entre otras funciones, un blindaje de la señal que se transmite por el conducto interior. (8)

### Fibra óptica:

Es un medio de transmisión empleado habitualmente en redes de datos; un hilo muy fino de material transparente, vidrio o materiales plásticos, por el que se envían pulsos de luz que representan los datos a transmitir. (9)

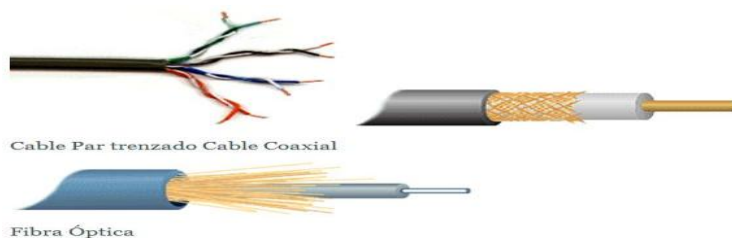


Figura 5: El medio. (6)

## Concentradores de cableado

Una LAN en bus usa solamente tarjetas de red en las estaciones y cableado coaxial para interconectarlas, además de los conectores, sin embargo este método complica el mantenimiento de la red ya que si falla alguna conexión, la misma deja de funcionar. Para impedir estos problemas las redes de área local usan concentradores de cableado, para realizar las conexiones de las estaciones, en vez de distribuirlas, el concentrador las centraliza en un único dispositivo manteniendo indicadores luminosos de su estado e impidiendo que una de ellas pueda hacer fallar toda la red. (5)

Existen dos tipos de concentradores de cableado:

- Concentradores pasivos: actúan como un simple concentrador cuya función principal consiste en interconectar toda la red.
- Concentradores activos: además de su función básica de concentrador también amplifican y regeneran las señales recibidas antes de ser enviadas y ejecutadas.

# *Fundamentación Teórica*

Los concentradores de cableado tienen dos tipos de conexiones: para las estaciones y para unirse a otros concentradores y así aumentar el tamaño de la red. Estos se clasifican dependiendo de la manera en que internamente realizan las conexiones y distribuyen los mensajes. A esta característica se le llama topología lógica.

Existen dos tipos principales:

- Concentradores con topología lógica en bus (HUB): estos dispositivos hacen que la red se comporte como un bus enviando las señales que les llegan por todas las salidas conectadas.
- Concentradores con topología lógica en anillo (MAU): se comportan como si la red fuera un anillo enviando la señal que les llega por un puerto al siguiente.

## **1.3 Ingeniería inversa**

Se conoce como Ingeniería Inversa o Reverse Engineering al proceso de duplicar una pieza, componente o conjunto, sin la ayuda de planos, documentación o modelos auxiliares. Se parte siempre de un modelo físico y se usan métodos de ingeniería de medida, análisis, diseño y adquisición de datos para finalmente obtener una réplica idéntica o mejorada del objeto.

La ingeniería inversa se ha definido como el proceso de construir especificaciones de un mayor nivel de abstracción partiendo del código fuente de un sistema, software o cualquier otro producto (se puede utilizar como punto de partida cualquier otro elemento de diseño y otros).

El método se denomina así porque avanza en dirección opuesta a las tareas habituales de ingeniería, que consisten en utilizar datos técnicos para elaborar un producto determinado.

Esta tiene la misión de desentrañar los misterios y secretos de los sistemas en uso. Consiste principalmente en recuperar el diseño de una aplicación a partir del código. Esto se realiza mediante herramientas que extraen información de los datos, procedimientos y arquitectura del sistema existente. (10) En este caso se realizará mediante la herramienta de modelado Visual Paradigm.

La ingeniería inversa no es más que la realización de la documentación de un proyecto ya realizado, es decir, obtener toda la información necesaria a partir del código del programa. Esta permite obtener el producto que indique cómo se ha construido el mismo.

### **1.3.1 Tipos de Ingeniería Inversa**

La ingeniería inversa puede ser de varios tipos:

Ingeniería inversa de datos: Se aplica sobre algún código de base de datos (aplicación, código SQL, y otros) para obtener los modelos relacionales o sobre el modelo relacional, para obtener el diagrama entidad-relación.

Ingeniería inversa de lógica o de proceso: Cuando se aplica sobre código de un programa para averiguar su lógica o sobre cualquier documento de diseño para obtener documentos de análisis o de requisitos.

# *Fundamentación Teórica*

Ingeniería inversa de interfaces de usuario: Se aplica con objeto de mantener la lógica interna del programa para obtener los modelos y especificaciones que sirvieron de base para la construcción de la misma, con objeto de tomarlas como punto de partida en procesos de ingeniería directa que permitan modificar dicha interfaz.

Ingeniería inversa de protocolos de comunicación: Independientemente del medio físico, se analiza el comportamiento de la comunicación, analizan tramas y documentan protocolos, para la implementación de nuevos sistemas, replicas o interfaces. (10)

Se pone en práctica en este caso la ingeniería inversa de lógica o de proceso ya que se aplica sobre el código de un programa para averiguar su lógica.

Habitualmente, este tipo de ingeniería inversa se usa para:

- Entender mejor la aplicación y regenerar el código.
- Migrar la aplicación a un nuevo sistema operativo.
- Generar/completar la documentación.
- Comprobar que el código cumple las especificaciones de diseño.

## **1.3.2 Sistemas a los que se le aplica la Ingeniería Inversa**

Es aplicable a sistemas con las siguientes características:

- Documentación inexistente.
- Programación en bloque de códigos muy grandes y/o sin estructurar.
- Inexistencia de documentación interna en los programas.
- La aplicación cubre gran parte de los requisitos y del rendimiento esperado.
- La aplicación está sujeta a cambios frecuentes, que pueden afectar a parte del diseño.

## **1.3.3 Proceso de ingeniería inversa**

La primera tarea para comenzar la ingeniería inversa es limpiar el código fuente, es decir, dejar solamente las clases que sean significativas en el proyecto, de manera que sea más fácil de leer, y proporciona la base para todas las actividades subsiguientes.

El núcleo de la ingeniería inversa es una actividad denominada extracción de abstracciones. El ingeniero tiene que evaluar el viejo programa y a partir del código fuente (que no suele estar documentado), tiene que extraer una especificación significativa del procesamiento que se realizará, la interfaz de usuario que se aplica y las estructuras de datos de programa o de base de datos que se utiliza.

Antes de iniciar el trabajo detallado debe comprenderse totalmente la funcionalidad general de todo el sistema de aplicaciones sobre el que se está operando. Esto es lo que establece un contexto para un análisis posterior, y proporciona ideas generales acerca de los problemas de interoperabilidad entre aplicaciones dentro del sistema. Cada uno de los programas del sistema de aplicaciones representará

# *Fundamentación Teórica*

una abstracción funcional con un elevado nivel de detalle, creándose un diagrama de bloques como representación de la iteración entre estas abstracciones funcionales. Cada uno de los componentes de estos diagramas efectúa una sub-función, y representa una abstracción definida de procedimientos. En algunas situaciones ya existen especificaciones de sistema, programa y componente. Cuando ocurre tal cosa, se revisan en busca de posibles errores.

Para los sistemas grandes, la ingeniería inversa suele efectuarse mediante el uso de un enfoque semi-automatizado. Las herramientas CASE se utilizan para “analizar” la semántica del código existente. La salida de este proceso se pasa entonces a unas herramientas de reestructuración y de ingeniería directa que completarán el proceso de reingeniería.

A la hora de realizar ingeniería inversa de procesos se suelen seguir los siguientes pasos:

1. Buscamos el programa principal.
2. Ignoramos inicializaciones de variables.
3. Inspeccionamos la primera rutina llamada y la examinamos si es importante.
4. Inspeccionamos las rutinas llamadas por la primera rutina del programa principal, y examinamos aquellas que nos parecen importantes.
5. Repetimos los pasos 3-4 a lo largo del resto del software.
6. Recopilamos esas rutinas “importantes”, que se llaman componentes funcionales.
7. Asignamos significado a cada componente funcional, esto es (a) explicamos qué hace cada componente funcional en el conjunto del sistema y (b) explicamos qué hace el sistema a partir de los diferentes componentes funcionales.

Una vez encontrados los posibles componentes funcionales, conviene repasar la lista teniendo en cuenta que este es funcional si su ausencia impide seriamente el funcionamiento de la aplicación, dificulta la legibilidad del código, impide la comprensión de todo o de otro componente funcional o cuando hace caer a niveles muy bajos la calidad, fiabilidad, mantenibilidad entre otros. (10)

## **1.3.4 Herramientas para la Ingeniería Inversa**

### **AthTek Software:**

Puede generar automáticamente diagramas de flujo desde el código fuente, y ayuda a los usuarios a entender las complejas estructuras del programa por medio de diagramas visuales. El diagrama de flujo puede ser completamente ampliado y utiliza gráficos de alta calidad para permitir el código visual.

### **DB-MAIN:**

Está diseñada para ayudar a los desarrolladores y analistas en la mayor parte de los procesos de ingeniería de datos incluyendo: procesos de diseño, transformaciones, ingeniería inversa y comprensión de programas, mantenimiento, evolución e integración, bases de datos temporales y activos. (11)

# *Fundamentación Teórica*

## **Visual Paradigm for UML:**

Es una herramienta profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El software de modelado UML ayuda a una rápida construcción de aplicaciones de calidad, y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. (12)

Con esta herramienta de modelado se pudieron generar los diagramas de clase del diseño a partir del código del programa.

## **1.3.5 Ventajas y desventajas de la Ingeniería Inversa**

Gracias al uso de Ingeniería Inversa se obtienen los siguientes beneficios:

Reducir la complejidad del sistema: al intentar comprender el software se facilita su mantenimiento y la complejidad existente disminuye.

Generar diferentes alternativas: del punto de partida del proceso, principalmente código fuente, se generan representaciones gráficas lo que facilita su comprensión.

Recuperar y/o actualizar la información perdida (cambios que no se documentaron en su momento): en la evolución del sistema se realizan cambios que no se suele actualizar en las representaciones de nivel de abstracción más alto, para lo cual se utiliza la recuperación de diseño.

Facilitar la reutilización: por medio de la ingeniería inversa se pueden detectar componentes de posible reutilización de sistemas existentes, pudiendo aumentar la productividad, reducir los costes y los riesgos de mantenimiento. La finalidad de la ingeniería inversa es la de desentrañar los misterios y secretos de los sistemas en uso a partir del código. Para ello, se emplean una serie de herramientas que extraen información de los datos, procedimientos y arquitectura del sistema existente. (10)

El proceso de ingeniería inversa no contiene ninguna desventaja ya que con ella solo se alcanzan beneficios.

## **1.4 Análisis de soluciones existente de laboratorios virtuales de redes**

### Laboratorio de Redes del Programa de Ingeniería en Telecomunicaciones:

En la facultad de ingeniería, Bogotá, Colombia se realizó dicho laboratorio, el cual constituye el espacio que requiere el estudiante para proyectar sus trabajos de diseño e implementación de redes de datos. Para atender la demanda del laboratorio se cuenta con personal altamente capacitado en el manejo de los equipos para la configuración de redes LAN. Este laboratorio tiene como objetivo apoyar a los estudiantes en el desarrollo de las prácticas e investigación en redes de datos y seguridad de la información del programa. (13)

El laboratorio cuenta con equipos adecuados para el desarrollo de los cursos enfocados hacia las prácticas propias de las asignaturas de redes y telemática.

# *Fundamentación Teórica*

Este está compuesto por:

- Equipos para el diseño, implementación y simulación de redes LAN/WAN y Wireless.
- Equipos y elementos para Cableado estructurado.
- Equipos de cómputo para verificación e implementación de sistemas de seguridad para la información.

## Laboratorio virtual de Redes y Comunicaciones:

La Facultad de Ingeniería y ciencias Hídricas de la Universidad Nacional del Litoral, posee un laboratorio destinado a la carrera de ingeniería en Informática, el cual es utilizado para el desarrollo de trabajos prácticos en las materias Redes y Comunicaciones. Este laboratorio al ser compartido con otras materias debe proveer soporte a distintos sistemas operativos y aplicaciones.

El laboratorio permite colocar en una Intranet los equipos de hardware, de tal manera que sean accesibles por los alumnos desde la biblioteca u otros laboratorios, con posibilidad también de ser accesible desde Internet.

Existen también distintos paquetes de software que simulan con cierta fidelidad el comportamiento de equipos de hardware, donde podemos mencionar:

- Router Sim 3.11
- Boson Router Sim 5.27. (14)

La Revolución Virtual llegó también a las universidades cubanas, donde se crean nuevas propuestas con gran impacto social. La informática ha resuelto la carencia de las instalaciones de laboratorios reales, insuficientes en todo el país por problemas económicos. Cuba no cuenta con un laboratorio de redes para la enseñanza.

La Universidad de las Ciencias Informáticas se encuentra en proceso de creación de un laboratorio virtual de redes para la educación. El sistema en general permitirá la interacción del usuario con la escena, logrando alcanzar los objetivos para el cual fue desarrollado. Este laboratorio virtual servirá como material de apoyo para la unidad curricular “Redes de Computadoras”, que se encuentra dentro de las unidades curriculares que forman el Programa Nacional de Formación en Informática de la República Bolivariana de Venezuela. Este laboratorio de redes servirá también para las universidades cubanas.

Después de analizar los laboratorios virtuales de redes seleccionados, se puede evidenciar que realizan funcionalidades muy buenas que podrían agregarse al módulo de diseño e instalación de una red. Un ejemplo pudiera ser, realizar un video con una simulación de la instalación de una red y ubicarlo en el laboratorio donde el usuario tenga la posibilidad de acceder a él en caso de que este tenga alguna duda en la realización de la práctica. Otra funcionalidad sería agregar un módulo de comunicación a la aplicación para lograr la interacción entre los usuarios de la práctica de laboratorio.



# *Fundamentación Teórica*

## **1.5 Metodología, Lenguaje, Herramienta de modelado**

Para la confección del proyecto se utilizó la metodología RUP (Rational Unified Procces). A pesar de ser utilizada principalmente para proyectos grandes se decidió por política de la facultad trabajar basados en dicha metodología. A continuación se describe detalladamente las características de RUP.

### **1.5.1 Rational Unified Procces (RUP)**

El Proceso Unificado de Rational (RUP) define un manejo entero de las actividades y de los artefactos que se necesita elegir. Es el proceso de desarrollo más general de los existentes.

Las iteraciones tempranas de proyectos basados en la metodología RUP se enfocan fuertemente sobre la arquitectura del software; la puesta en práctica rápida de características se retrasa hasta que se ha identificado y se ha probado una arquitectura firme. (15)

A continuación se caracteriza este proceso de desarrollo:

- RUP realiza un levantamiento exhaustivo de requerimientos.
- Busca detectar defectos en las fases iniciales.
- Intenta reducir al número de cambios tanto como sea posible.
- Realiza el análisis y diseño, tan completo como sea posible.
- Diseño genérico, intenta anticiparse a futuras necesidades.
- Las necesidades de clientes no son fáciles de discernir.
- Existe un contrato prefijado con los clientes.
- El cliente interactúa con el equipo de desarrollo mediante reuniones.
- Está dirigido durante el proceso por los casos de uso.
- Centrado en la arquitectura.
- Proporciona una guía de cómo usar de forma efectiva UML.
- Es un proceso configurable. (16)

RUP es una metodología muy extensa y en la mayoría de los casos en el momento de su implantación se considera un proceso demasiado costoso para la cantidad de actividades y entregables que se definen. Hay que tener en cuenta que no es obligado hacer estas tareas, sino que se puede configurar el proceso para adaptarlo únicamente a aquellas partes que se consideran necesarias.

Repite en una serie de ciclos, cada ciclo concluye con una versión del producto (release) y está dividido en 4 fases: inicio, elaboración, construcción y transición. Cada fase está a su vez dividida en iteraciones, y en cada iteración se realizan 6 procesos o etapas principales: requerimientos, análisis, diseño, implementación, pruebas y entrega. (16)



# Fundamentación Teórica



Figura 6: Proceso Unificado de Desarrollo de Software. (17)

## 1.5.2 Lenguaje Unificado de Modelado (UML)

Es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad para visualizar, especificar, construir y documentar un sistema. (18)

Surge ante la necesidad que tenían los desarrolladores del software de comunicar sus modelos a otros, y no solo entre ellos mismos, sino a todos los que estuvieran vinculados con el proyecto e incluso, a futuras generaciones. Además, se hacía necesario proporcionar un marco en el que desarrolladores individuales pudieran pensar o analizar y registrar estos modelos en documentos. (19)

La creación de UML, permitió a los creadores de software visualizar los resultados de su trabajo en esquemas o diagramas estandarizados que capturan sus ideas de forma convencional y fácil de comprender para comunicarlas a otros.

Este lenguaje propone un vocabulario que incluye tres categorías: elementos, relaciones y diagramas, y estas a su vez están compuestas por diferentes categorías. En la figura que a continuación se muestra se especifican cada una de ellas. (19)

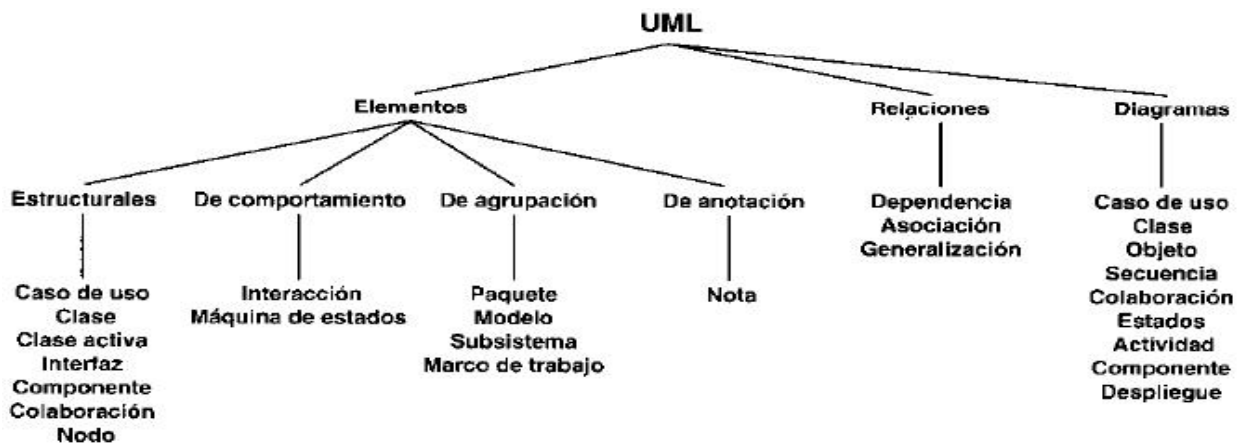


Figura 7: Vocabulario de UML. (20)

# Fundamentación Teórica

## 1.5.3 Visual Paradigm

Presenta una estabilidad de ejecución en diferentes sistemas operativos y la facilidad de abrir y trabajar con un modelo UML utilizando el mismo programa sin afectar en absoluto el trabajo hecho; además destacar que esta herramienta guarda todo el modelo en un solo fichero y trabaja bastante decente en ordenadores “poco potentes”. (21)

Es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El software de modelado UML ayuda a una rápida construcción de aplicaciones de calidad y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. (22)

Se integra con varios ambientes de desarrollo (IDE) lo cual posibilita pasar del código al modelado y viceversa. Establece interoperabilidad con otras aplicaciones como el Visio (Es un software de dibujo vectorial para Microsoft Windows.) y el Rational Rose. Disponible en múltiples lenguajes y plataformas: Microsoft Windows (98, 2000, XP, o Vista) Linux, Mac OS X, Solaris o Java. (23)

Tiene la ventaja de ser multiplataforma y una herramienta CASE muy potente y fácil de utilizar. Soporta aplicaciones web, varios idiomas, es fácil de instalar y actualizar, tiene compatibilidad entre ediciones.

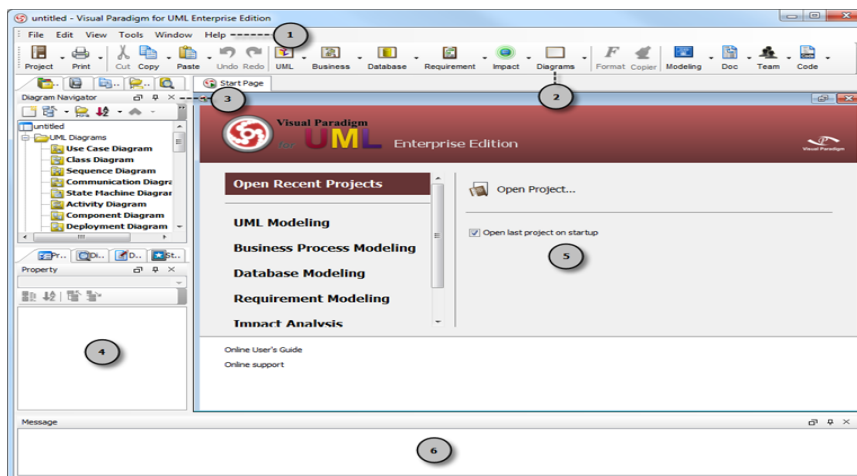


Figura 8: Visual Paradigm. (24)

### Consideraciones parciales

Con la realización de este capítulo se dejaron fundamentadas algunas definiciones que ayudaron a un mejor entendimiento de la situación planteada. Se definieron las características de la ingeniería inversa y se plasmó la metodología, lenguaje y herramienta con la que se realizará el modelado del sistema.

# Características del Sistema

## Capítulo 2: “Características del sistema”

### Introducción

En el presente capítulo se abordan las características que el sistema debe tener. Se realiza la especificación de los requisitos funcionales y no funcionales con los que debe cumplir la aplicación. También se obtienen los diagramas de casos de usos del sistema y los de clases del análisis.

### 2.1 Modelo del dominio

El modelo de dominio es una representación gráfica de las clases conceptuales del mundo real, no de componentes de software. Este modelo describe como se estructura y enlaza la información, cuya función principal es ayudar a comprender el problema. A continuación se presenta el modelo de dominio correspondiente al sistema en cuestión.

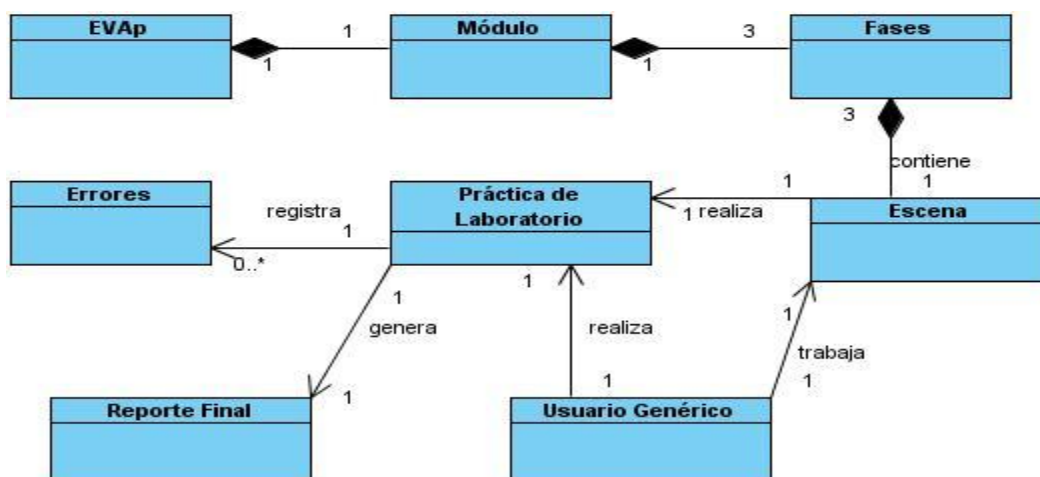


Figura 9: Modelo del dominio

**EVAp:** Entorno Virtual de Aprendizaje de la República de Venezuela donde se ubicará el proyecto Laboratorios Virtuales, en el cual está contenido el módulo “Diseño e Instalación de una red de Área Local (LAN)”.

**Módulo:** En el laboratorio virtual se desarrolla el módulo “Diseño e Instalación de una red de Área Local (LAN)”, el cual estará compuesto por tres fases de trabajo.

**Fases:** Cada una de las fases de trabajo contendrá una escena donde se llevará a cabo la práctica de laboratorio.

**Escena:** Constituye el escenario donde se realizará la práctica de laboratorio.

**Práctica de Laboratorio:** El usuario tendrá la oportunidad de desarrollar su trabajo donde finalmente obtendrá un reporte de las actividades realizadas.

**Usuario Genérico:** Cualquier persona que se registre podrá acceder a la realización de la práctica.

# *Características del Sistema*

**Errores:** Con el desarrollo del trabajo de laboratorio el usuario podrá cometer algunos errores los cuales se mostrarán en la aplicación.

**Reporte Final:** Al concluir la práctica de laboratorio se deberá generar un reporte donde se muestren los resultados del trabajo realizado.

## **2.2 Descripción del sistema**

El laboratorio virtual: "Diseño e Instalación de una red de área local (LAN)" cuenta con tres fases que permiten al usuario desarrollar los procesos de diseño e instalación de una red de área local.

Fase I: Construcción de un cable de red, el objetivo de esta fase es construir cables de red, mediante el uso de un entorno virtual.

Fase II: Diseño de una red de área local, su objetivo es construir una red de área local en 2D, mediante el uso de un entorno virtual.

Fase III: Instalación de una red de área local, cuyo objetivo es instalar una red de área local en 3D de un edificio, mediante el uso de un entorno virtual.

## **2.3 Especificación de los requisitos**

Para modelar el sistema, se identifican sus requisitos, tanto funcionales como no funcionales, y se modelan los funcionales en términos de casos de uso del sistema.

### **2.3.1 Requisitos funcionales**

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

#### **RF 1 Registrar:**

La aplicación mostrará una ventana donde el usuario inserte su nombre, apellido(s), número de cédula o pasaporte. Además aparecerán las tres fases a las cuales el usuario deberá entrar.

#### **RF 2 Mostrar Objetos de la Escena:**

La aplicación deberá mostrar los objetos que contiene la escena según en el escenario y fase en la que se encuentre el usuario.

#### **RF 3 Salvar Archivos:**

La aplicación permitirá salvar los archivos correspondientes a las fases I, II y III para su posterior utilización.

#### **RF 4 Eliminar Objetos Gráficos:**

La aplicación permitirá eliminar los objetos gráficos que el usuario genérico desee.

#### **RF 5 Seleccionar Objetos Gráficos:**

La aplicación permitirá seleccionar el objeto gráfico que el usuario desee.

#### **RF 6 Mostrar Tiempo de Duración de la Práctica:**

# *Características del Sistema*

La aplicación deberá mostrar el tiempo que demoró el usuario desde que accedió a la práctica hasta su culminación. El formato en que se mostrará el tiempo es cantidad de horas-minutos-segundos transcurridos desde que el usuario accede al primer escenario y fase.

## **RF 7 Limpiar Escena:**

La aplicación dejará todo el escenario vacío.

## **RF 8 Mostrar errores:**

La aplicación mostrará todos los errores cometidos en la práctica en una ventana.

## **RF 9 Agrupar Objetos Gráficos:**

La aplicación permitirá agrupar los objetos gráficos que el usuario genérico haya seleccionado.

## **RF 10 Desagrupar Objetos Gráficos:**

La aplicación permitirá desagrupar los objetos gráficos que el usuario genérico tenga agrupados.

## **RF 11 Exportar Escena:**

La aplicación permitirá guardar la escena para su posterior utilización.

### **2.3.2 Requisitos no funcionales**

Los requisitos no funcionales son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido y confiable. Son los aspectos del sistema visibles para el usuario, que no están relacionados de forma directa con el comportamiento funcional del sistema. Los requisitos no funcionales forman una parte significativa de la especificación. Son importantes para que clientes y usuarios puedan valorar las características no funcionales del producto, esto puede marcar la diferencia entre un producto bien aceptado y uno con poca aceptación.

#### **RNF Usabilidad:**

Será usable por cualquier tipo de usuario con experiencia básica, media o avanzada en el proceso de instalación de una red de área local (LAN).

#### **RNF Apariencia o interfaz externa:**

El sistema tiene una interfaz amigable con una buena utilización de los elementos de diseño y adecuada combinación de colores. Contiene tres fases en las cuales se muestran los objetos gráficos correspondientes a cada una de ellas, los cuales estarán visibles todo el tiempo para facilitar la interacción del usuario con la aplicación.

El sistema tendrá una buena navegabilidad, propiciando varias opciones para acceder a cada servicio brindado.

# Características del Sistema

## **RNF Rendimiento:**

El sistema debe garantizar un tiempo de respuesta de 3 a 5 segundos en dependencia de la complejidad de la acción que se esté realizando.

## **RNF Seguridad:**

Confidencialidad: Todos los usuarios que accedan a la realización de la práctica deben registrarse. Al concluir las actividades realizadas se guardará un reporte con los datos del mismo y los resultados de la instalación de la red.

Integridad: La información manejada por el sistema será objeto de cuidadosa protección contra la corrupción y estados inconsistentes.

Disponibilidad: La aplicación estará disponible para todos los usuarios que quieran realizar la práctica de laboratorio.

## **RNF Software:**

Se utilizará como lenguaje de programación Qt, como Sistema Operativo Windows y Linux, para el modelado la herramienta Visual Paradigm for UML y como entorno de desarrollo integrado (IDE) Visual Studio 2008.

## **RNF Hardware:**

El ordenador donde se instale el componente deberá cumplir con las siguientes características:

Se podrá contar con los periféricos Teclado y Mouse.

Requisito Mínimo: Procesador: Celeron con CPU 1.8 GHz, Memoria: 256 Mb, Disco Duro: 60 Gb.

Requisitos Recomendados: Procesador: Pentium IV 2.66 GHz, Memoria: 512 Mb, Disco Duro: 80 Gb.

## **RNF Restricciones de diseño:**

La aplicación se desarrollará utilizando Ogre como motor gráfico y QT Creator como entorno de desarrollo integrado (IDE). En el trabajo con gráficos tridimensionales se utilizará la herramienta Blender.

## **RNF Portabilidad:**

El sistema debe ser multiplataforma, siendo compatible con los sistemas operativos Window y Linux.

## **2.4 Modelo de casos de uso del sistema**

La modelación de un sistema, muestra la forma en que este tiene que funcionar, cuyo objetivo es responder a una necesidad específica del cliente. Describen los requisitos funcionales de un actor partiendo de las interacciones que ejecuta con la aplicación.

# Características del Sistema

## 2.4.1 Descripción del actor del sistema

Actor	Descripción
Usuario genérico	Persona con privilegios para interactuar con la aplicación. Haciendo uso de todas las funcionalidades con que cuenta la misma con fines educativos. Este usuario genérico puede ser el estudiante, el profesor o cualquier persona con el objetivo de interactuar con dicha aplicación.

Tabla 1: Descripción del actor del sistema.

## 2.4.2 Diagrama de casos de uso del sistema

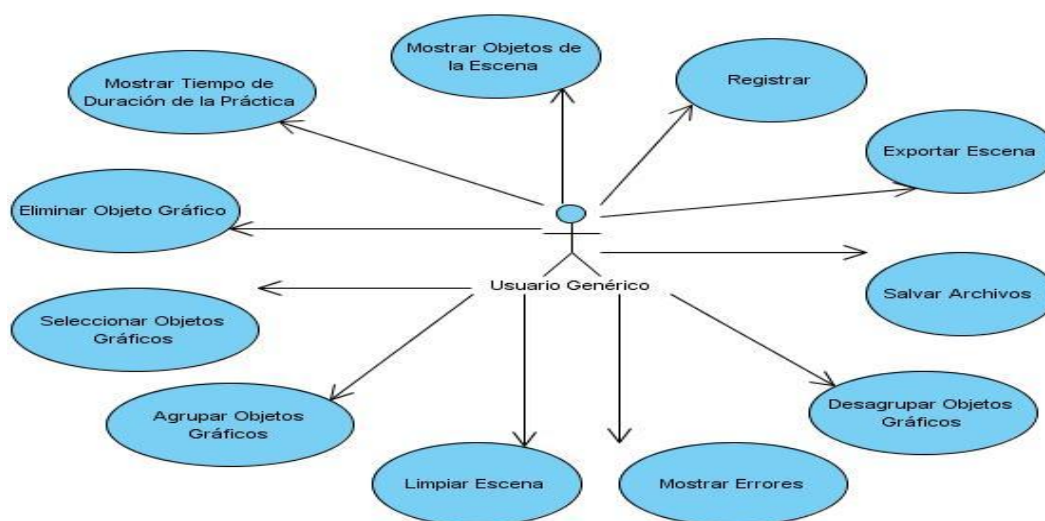


Figura 10: Diagrama de casos de uso del sistema.

## 2.4.3 Descripción de los casos de uso del sistema

<b>Caso de Uso del sistema</b>	Registrar
<b>Actor</b>	Usuario Genérico
<b>Propósito</b>	Permitir mostrar al usuario genérico una ventana donde se pueda registrar y seleccionar la fase en la que desee trabajar .
<b>Resumen</b>	El caso de uso se inicia cuando el usuario genérico selecciona la opción registrar y finaliza cuando llene los campos y seleccione la opción guardar.
<b>Precondiciones</b>	El usuario genérico debe de haber leído las indicaciones expuestas en el Entorno Virtual de Apoyo (EVAp).
<b>Postcondiciones</b>	El usuario genérico es registrado.
<b>Referencias</b>	RF 1
<b>Prioridad</b>	Crítico



# Características del Sistema

Flujo normal de eventos	
Acción del actor	Respuesta del Sistema
1. El usuario genérico comienza la práctica y le sale la opción registrar.	1.1 La aplicación muestra una ventana con el siguiente formato: <ul style="list-style-type: none"> <li>▪ Nombre y apellidos</li> <li>▪ Número de cédula.</li> <li>▪ Número de pasaporte</li> <li>▪ Fases de trabajo</li> </ul>
2. El usuario genérico llena todos los campos pedidos y luego selecciona la opción aceptar.	2.1 La aplicación verifica que los datos de los campos estén llenos y correctamente escritos y le da acceso a comenzar dicha práctica.

## Prototipo de interfaz "Registrar"

Figura 11 Prototipo de interfaz "Registrar"

Flujos Alternos	
Acción del Actor	Respuesta del Sistema
1. El usuario genérico no llena todos los campos pedidos.	1.1 La aplicación emitirá el mensaje "No debe dejar ningún campo vacío".
2. El usuario genérico escribe símbolos no permitidos en el nombre y apellidos.	2.1 La aplicación emitirá el mensaje "El nombre no es válido, por favor indique el nombre correctamente".
3. El usuario genérico escribe símbolos no permitidos en el identificador.	3.1 La aplicación emitirá el mensaje "El carácter no es válido, por favor indique el número de cédula correctamente".

Tabla 2: Especificación del requisito Registrar.

<b>Caso de Uso del Sistema</b>	Mostrar Objetos de la Escena
--------------------------------	------------------------------



# *Características del Sistema*

<b>Actor</b>	Usuario Genérico
<b>Propósito</b>	Permitir mostrar al usuario genérico los objetos gráficos de la escena.
<b>Resumen</b>	El caso de uso se inicia cuando el usuario se ha registrado correctamente y ha comenzado la realización de la práctica y termina cuando el sistema muestra la escena correspondiente a la fase elegida por el usuario.
<b>Precondiciones</b>	Comprobar que los objetos sean mostrados según la fase en la que se encuentre el usuario.
<b>Postcondiciones</b>	Se muestran los objetos de la escena de acuerdo a la fase seleccionada.
<b>Referencias</b>	RF 2
<b>Prioridad</b>	Crítico
Flujo normal de eventos	
Acción del actor	Respuesta del Sistema
1. El usuario genérico accederá a la realización de la fase 1 del laboratorio.	<p>1.1 La aplicación solo mostrará los objetos que corresponden a la fase1.</p> <p>Objetos-Fase1-Escenario1:</p> <ul style="list-style-type: none"> <li>▪ Cables: <ul style="list-style-type: none"> <li>• Coaxial.</li> <li>• UTP_Cat_1.</li> <li>• UTP_Cat_6.</li> </ul> </li> <li>▪ Conectores: <ul style="list-style-type: none"> <li>• Conector_coaxial.</li> <li>• Conector RJ11.</li> <li>• Conector RJ45.</li> </ul> </li> <li>▪ Herramientas: <ul style="list-style-type: none"> <li>• Comprobador.</li> <li>• Cortador.</li> <li>• Crinpeadora.</li> </ul> </li> <li>▪ Número de pasaporte</li> <li>▪ Fases de trabajo</li> </ul>
2. El usuario genérico accederá a la realización	2.1 La aplicación solo mostrará los objetos que

# Características del Sistema

de la fase 2 del laboratorio.	corresponden a la fase 2.  Objetos-Fase2-Escenario2: En la escena se encontrarán los siguientes objetos 2D: <ul style="list-style-type: none"><li>▪ Un plano en blanco donde se diseñará la red de área local (LAN).</li><li>▪ Equipos informáticos.</li><li>▪ Cables UTP, coaxial, fibra óptica.</li><li>▪ Enrutador.</li><li>▪ Repetidora.</li><li>▪ Bridges.</li><li>▪ Dispositivos inalámbricos</li><li>▪ MAU (Multistation Access Unit).</li><li>▪ Hubs y Switch.</li></ul>
3. El usuario genérico accederá a la realización de la fase 3 del laboratorio.	3.1 La aplicación solo mostrara los objetos que corresponden a la fase 3.  Objetos-Fase3-Escenario3: En la escena se encontrarán los siguientes objetos 3D: <ul style="list-style-type: none"><li>▪ Equipos informáticos.</li><li>▪ Cables UTP, coaxial, fibra óptica.</li><li>▪ Enrutador.</li><li>▪ Repetidora.</li><li>▪ Bridges.</li><li>▪ Dispositivos inalámbricos.</li><li>▪ MAU (Multistation Access Unit).</li><li>▪ Hubs y Switch.</li><li>▪ Gabinete.</li></ul>

**Prototipo de interfaz “Mostrar Objetos de la Escena”**

# Características del Sistema

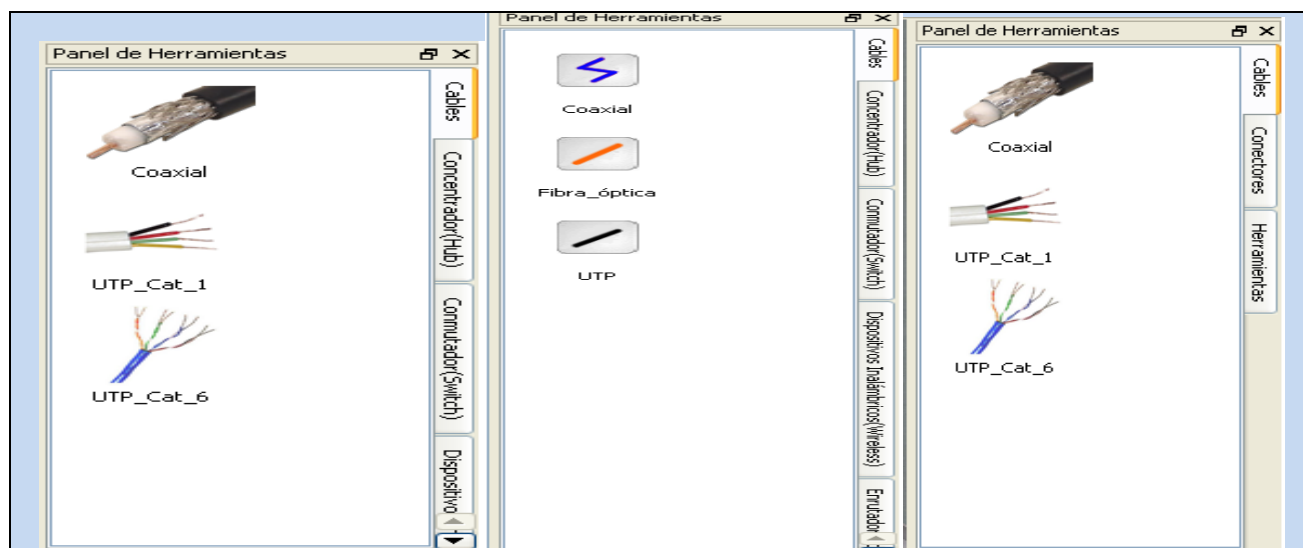


Figura 12: Prototipo de interfaz “Mostrar Objetos de la Escena”

Fase 3

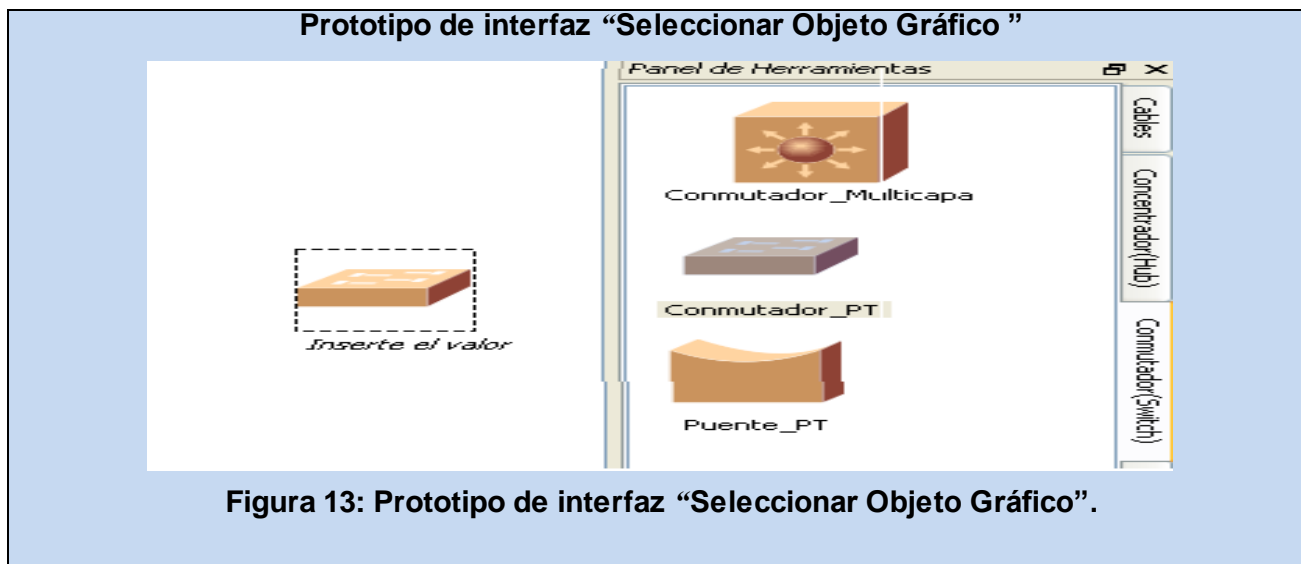
Fase 2

Fase 1

Tabla 3: Especificación del requisito Mostrar Objetos de la Escena.

<b>Caso de Uso del Sistema</b>	Seleccionar Objetos Gráficos.	
<b>Actor</b>	Usuario Genérico	
<b>Propósito</b>	Permitir al usuario genérico seleccionar el objeto gráfico deseado.	
<b>Resumen</b>	La aplicación le permitirá al usuario seleccionar los objetos gráficos dentro de la escena.	
<b>Precondiciones</b>	El usuario debe acceder a la realización de la práctica.	
<b>Postcondiciones</b>	El usuario selecciona los objetos gráficos de acuerdo a la fase elegida.	
<b>Referencias</b>	RF 5	
<b>Prioridad</b>	Crítico	
<b>Flujo normal de eventos</b>		
<b>Acción del actor</b>	<b>Respuesta del Sistema</b>	
1. El usuario accede a la realización de la práctica.	1.1 El sistema le muestra todos los objetos gráficos.	
2. El usuario selecciona los objetos gráficos.	2.1 El sistema le permite seleccionar el objeto gráfico.	

# Características del Sistema



**Tabla 4: Especificación del requisito Seleccionar Objetos Gráficos.**

<b>Caso de Uso del Sistema</b>	Mostrar Tiempo de Duración de la Práctica
<b>Actor</b>	Usuario Genérico
<b>Propósito</b>	Permitir que el usuario genérico pueda observar el tiempo que se demora en la realización de práctica.
<b>Resumen</b>	El caso de uso se inicia cuando el usuario genérico comienza la práctica y finaliza cuando termina dicha práctica.
<b>Precondiciones</b>	El usuario genérico debe haber comenzado la práctica.
<b>Postcondiciones</b>	El sistema muestra al usuario genérico el tiempo de realización de la práctica.
<b>Referencias</b>	RF 6
<b>Prioridad</b>	Auxiliar
<b>Flujo normal de eventos</b>	
<b>Acción del actor</b>	<b>Respuesta del Sistema</b>
1. El usuario genérico una vez registrado los datos se inicia la realización de la práctica.	1.1. La aplicación muestra en la esquina inferior derecha de la pantalla el tiempo de duración desde que fue iniciada la práctica hasta la culminación de la misma. 1.2. La aplicación muestra el tiempo de duración de la práctica con el siguiente formato: Horas-minutos-segundos.

# Características del Sistema

Prototipo de interfaz “Mostrar Tiempo de Duración de la Práctica”

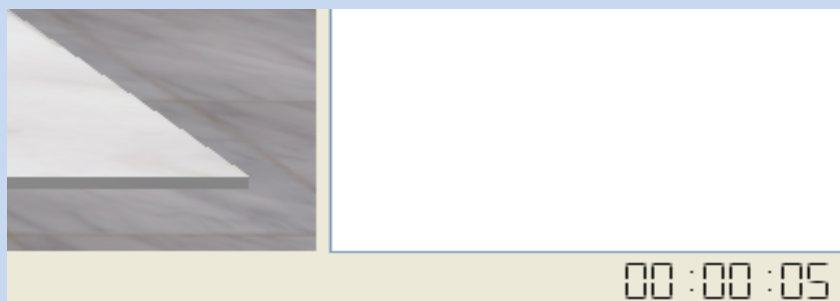


Figura 14: Prototipo de interfaz “Mostrar Tiempo de Duración de la Práctica”.

Tabla 5: Especificación del requisito Mostrar Tiempo de Duración de la Práctica.

<b>Caso de Uso del Sistema</b>	Mostrar Errores	
<b>Actor</b>	Usuario Genérico	
<b>Propósito</b>	Mostrar los errores que el usuario genérico cometa en la práctica.	
<b>Resumen</b>	El caso de uso se inicia cuando el usuario genérico comienza la práctica de la fase uno y concluye cuando haya terminado la misma.	
<b>Precondiciones</b>	El usuario genérico debe haber realizado la práctica.	
<b>Postcondiciones</b>	El sistema muestra al usuario genérico los errores cometidos.	
<b>Referencias</b>	RF 8	
<b>Prioridad</b>	Secundario	
<b>Flujo normal de eventos</b>		
	<b>Acción del actor</b>	<b>Respuesta del Sistema</b>
	1. El usuario genérico comienza la realización de la práctica.	1.1. La aplicación muestra una pantalla donde se registrarán los errores.
	2. El usuario genérico comete errores.	2.1. La aplicación automáticamente mostrará en el panel inferior derecho los errores cometidos.

Prototipo de interfaz “Mostrar Errores”

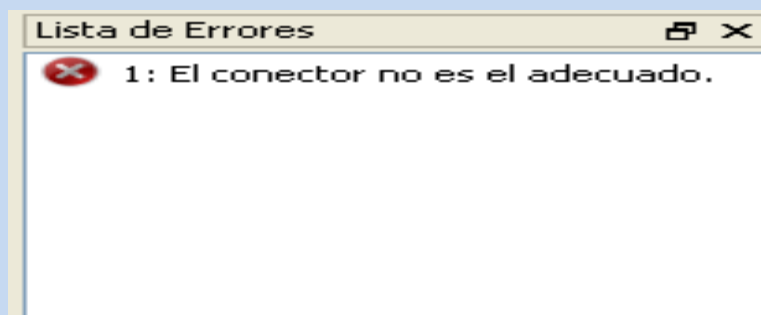


Figura 15: Prototipo de interfaz “Mostrar Errores”.

# Características del Sistema

Flujos Alternos	
Acción del Actor	Respuesta del Sistema
1.1. El usuario genérico no ha cometido errores.	1.1. El sistema muestra el panel de errores vacío.

**Tabla 6: Especificación del requisito Mostrar Errores.**

<b>Caso de Uso del Sistema</b>	Salvar Archivo
<b>Actor</b>	Usuario Genérico
<b>Propósito</b>	Generar un reporte con los datos de la práctica realizada.
<b>Resumen</b>	El caso de uso se inicia cuando el usuario genérico presiona clic sobre la opción guardar, finalizando así el caso de uso.
<b>Precondiciones</b>	El usuario genérico debe haber realizado la práctica.
<b>Postcondiciones</b>	El usuario salva la realización de la práctica en PDF.
<b>Referencias</b>	RF 3
<b>Prioridad</b>	Secundario

## Flujo normal de eventos

Acción del actor	Respuesta del Sistema
1. El usuario genérico concluye la práctica y desea guardarla.	1.1. La aplicación muestra en la barra de menú la opción guardar.
2. El usuario genérico presiona clic en la opción guardar.	2.1. La aplicación automáticamente salvará la práctica con extensión PDF.

## Prototipo de interfaz “ Salvar Archivo ”



**Figura 16: Prototipo de interfaz “Salvar Archivo”.**

Flujos Alternos	
Acción del Actor	Respuesta del Sistema
1.1. El usuario genérico no ha salvado la práctica.	1.1. El sistema no guarda cambios.

**Tabla 7: Especificación del requisito Salvar Archivo.**

# Características del Sistema

<b>Caso de Uso del Sistema</b>	Eliminar Objeto Gráfico
<b>Actor</b>	Usuario Genérico
<b>Propósito</b>	Permitir eliminar el objeto gráfico seleccionado.
<b>Resumen</b>	El caso de uso se inicia cuando el usuario genérico selecciona el objeto que quiere eliminar y finaliza cuando presiona clic en la opción eliminar.
<b>Precondiciones</b>	El usuario genérico debe haber seleccionado el objeto gráfico.
<b>Postcondiciones</b>	El usuario ha eliminado el objeto gráfico.
<b>Referencias</b>	RF 4
<b>Prioridad</b>	Crítico

## Flujo normal de eventos

Acción del actor	Respuesta del Sistema
1. El usuario genérico selecciona el objeto que desea eliminar.	1.1. La aplicación muestra en la barra de menú la opción eliminar.
2. El usuario genérico presiona clic en la opción eliminar.	2.1. La aplicación automáticamente eliminará el objeto gráfico.

## Prototipo de interfaz “ Eliminar Objeto Gráfico ”



Figura 17: Prototipo de interfaz “Eliminar Objeto Gráfico”.


Tabla 8: Especificación del requisito Eliminar Objeto Gráfico.

<b>Caso de Uso del Sistema</b>	Limpiar Escena
<b>Actor</b>	Usuario Genérico
<b>Propósito</b>	Eliminar todos los objetos gráficos que se encuentren en la escena.
<b>Resumen</b>	El caso de uso se inicia cuando el usuario genérico presiona clic sobre la opción eliminar todo, finalizando así el caso de uso.
<b>Precondiciones</b>	El usuario genérico debe haber realizado la práctica.
<b>Postcondiciones</b>	La escena se encuentra vacía.
<b>Referencias</b>	RF 7
<b>Prioridad</b>	Secundario

# Características del Sistema

Flujo normal de eventos	
Acción del actor	Respuesta del Sistema
1. El usuario genérico comienza la práctica y desea limpiar lo que hizo en la escena.	1.1. La aplicación muestra en la barra de menú la opción eliminar todo.
2. El usuario genérico presiona clic sobre la opción eliminar todo.	2.1. La aplicación automáticamente limpiará la escena.

**Prototipo de interfaz “Limpiar Escena”**



**Figura 18: Prototipo de interfaz “Limpiar Escena”.**

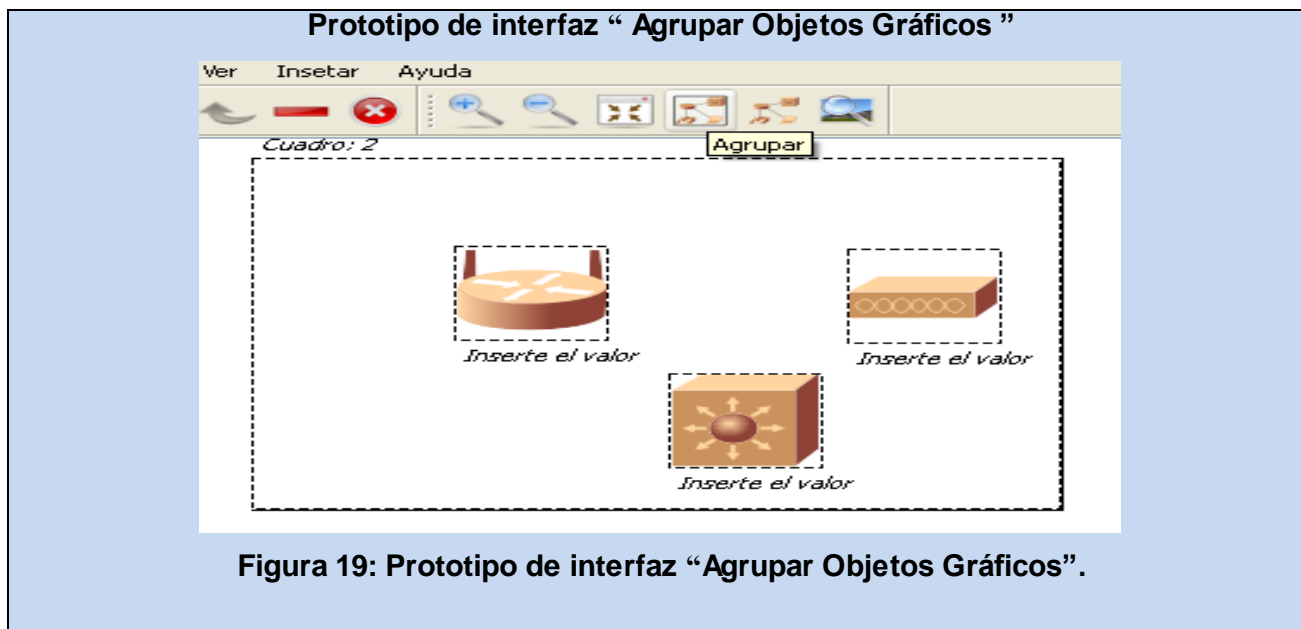
**Tabla 9: Especificación del requisito Limpiar Escena.**

<b>Caso de Uso del Sistema</b>	Agrupar Objetos Gráficos
<b>Actor</b>	Usuario Genérico
<b>Propósito</b>	Permitir que el sistema agrupe una serie de objetos dentro de un cuadro.
<b>Resumen</b>	El caso de uso se inicia cuando el usuario genérico ha comenzado la práctica de la fase dos y selecciona los objetos que quiere agrupar y finaliza cuando presiona clic en la opción agrupar.
<b>Precondiciones</b>	El usuario genérico debe haber seleccionado los objetos gráficos.
<b>Postcondiciones</b>	El usuario ha agrupado los objetos gráficos deseado.
<b>Referencias</b>	RF 9
<b>Prioridad</b>	Secundario

Flujo normal de eventos	
Acción del actor	Respuesta del Sistema
1. El usuario genérico selecciona los objetos que quiere agrupar.	1.1. La aplicación muestra en la barra de menú la opción agrupar.
2. El usuario genérico presionar clic sobre la opción agrupar.	2.1. La aplicación automáticamente agrupará los objetos gráficos.



# Características del Sistema



**Tabla 10: Especificación del requisito Agrupar Objetos Gráficos.**

<b>Caso de Uso del Sistema</b>	Desagrupar Objetos Gráficos	
<b>Actor</b>	Usuario Genérico	
<b>Propósito</b>	Permitir que el sistema desagrupe una serie de objetos agrupados.	
<b>Resumen</b>	El caso de uso se inicia cuando el usuario genérico ha comenzado la práctica de la fase dos y desea desagrupar los objetos agrupados y finaliza cuando presiona clic en la opción desagrupar.	
<b>Precondiciones</b>	El usuario genérico debe haber agrupado los objetos gráficos.	
<b>Postcondiciones</b>	El usuario ha desagrupado los objetos agrupados.	
<b>Referencias</b>	RF 10	
<b>Prioridad</b>	Secundario	
<b>Flujo normal de eventos</b>		
<b>Acción del actor</b>	<b>Respuesta del Sistema</b>	
1. El usuario genérico desea desagrupar los objetos agrupados.	1.1 La aplicación muestra en la barra de menú la opción desagrupar.	
2. El usuario genérico presiona clic sobre la opción desagrupar.	2.1 La aplicación automáticamente desagrupará los objetos gráficos.	

# Características del Sistema

Prototipo de interfaz “ Desagrupar Objetos Gráficos ”

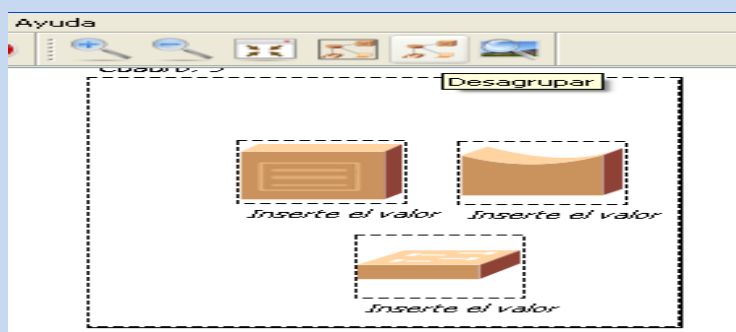


Figura 20: Prototipo de interfaz “Desagrupar Objetos Gráficos”.

Tabla 11: Especificación del requisito Desagrupar Objetos Gráficos.

<b>Caso de Uso del Sistema</b>	Exportar Escena
<b>Actor</b>	Usuario Genérico
<b>Propósito</b>	Guardar la escena
<b>Resumen</b>	El caso de uso se inicia cuando el usuario genérico ha finalizado la práctica de la fase tres y quiere exportar lo realizado.
<b>Precondiciones</b>	El usuario genérico debe haber realizado la práctica.
<b>Postcondiciones</b>	El usuario genérico ha exportado lo realizado en la práctica.
<b>Referencias</b>	RF 11
<b>Prioridad</b>	Secundario

### Flujo normal de eventos

Acción del actor	Respuesta del Sistema
1. El usuario genérico desea exportar lo realizado en la escena de la fase tres.	1.1. La aplicación muestra en la barra de menú la opción exportar escena.
2. El usuario genérico presiona clic en la opción exportar escena.	2.1. La aplicación automáticamente guardará la práctica en un fichero con extensión scene.

Prototipo de interfaz “ Exportar Escena ”

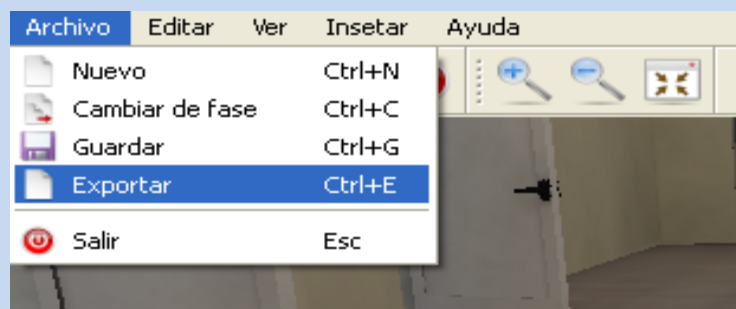


Figura 21: Prototipo de interfaz “Exportar Escena”.

Tabla 12: Especificación del requisito Exportar Escena.

# Características del Sistema

## 2.5 Modelo del análisis

El modelo de análisis ayuda a refinar los requisitos, representa la estructura global del sistema, describe la realización de casos de uso, sirve como una abstracción del modelo de diseño, se centra en los requisitos funcionales y permite razonar sobre los aspectos internos del sistema, incluidos sus recursos compartidos. Se centra además en aspectos tales como la flexibilidad ante los cambios y la reutilización. Esta estructura se utiliza como entrada en las actividades del diseño. Consiste en obtener una visión del sistema que se preocupa de ver qué se debe hacer, sin tener que preocuparse por cuestiones propias del lenguaje.

Aunque en el modelo del análisis hay un refinamiento de los requisitos, no se tiene en cuenta el lenguaje de programación a usar en la construcción, la plataforma en la que se ejecutará la aplicación, los componentes prefabricados o reusables de otras aplicaciones, entre otras características que afectan al sistema, ya que el objetivo del análisis es comprender perfectamente los requisitos del software y no precisar cómo se implementará la solución. (25)

### 2.5.1 Clases del análisis

Las clases del análisis se centran en los requisitos funcionales y son evidentes en el dominio del problema, ya que representan sus conceptos y relaciones. La representación se muestra en la siguiente tabla. [2]

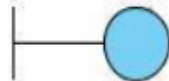

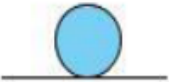
Nombre	Características	Representación
Interfaz	Modelan la interacción entre el sistema y sus actores.	 Clase de interfaz
Control	Representa coordinación secuencial, transacciones y control de otros objetos. Encapsula el control de un CU en concreto. Se usa para representar derivaciones y cálculos complejos.	 Clase de control
Entidad	Modelan información que posee larga vida y que es a menudo persistente.	 Clase de entidad

Tabla 13: Estereotipos de las clases del análisis.

# Características del Sistema

## 2.5.2 Diagramas de clases del análisis

A continuación se muestran los diagramas de clases del análisis correspondiente a los CUS:

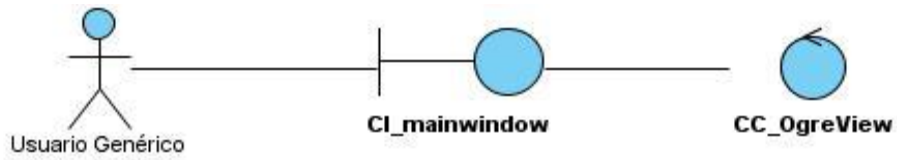


Figura 22 : Diagrama de clases del análisis del CU Registrar.

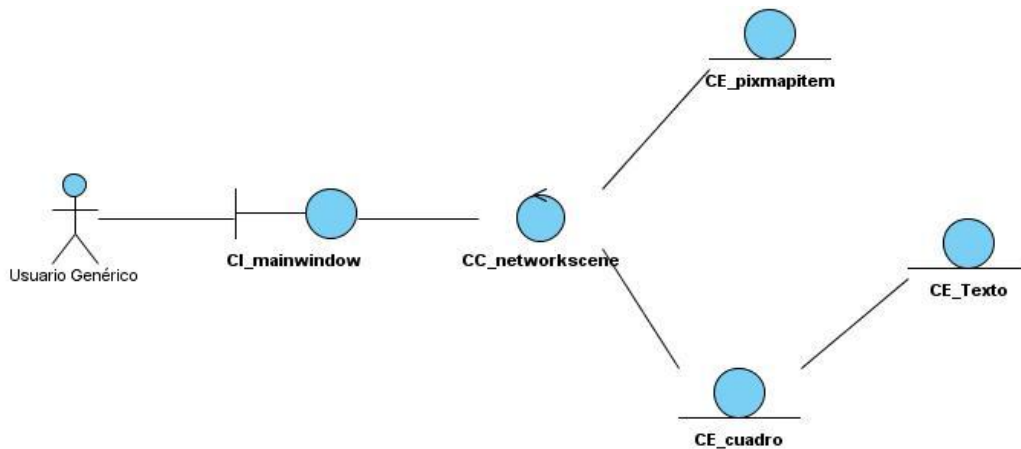


Figura 23: Diagrama de clases del análisis del CU Agrupar Objetos Gráficos.

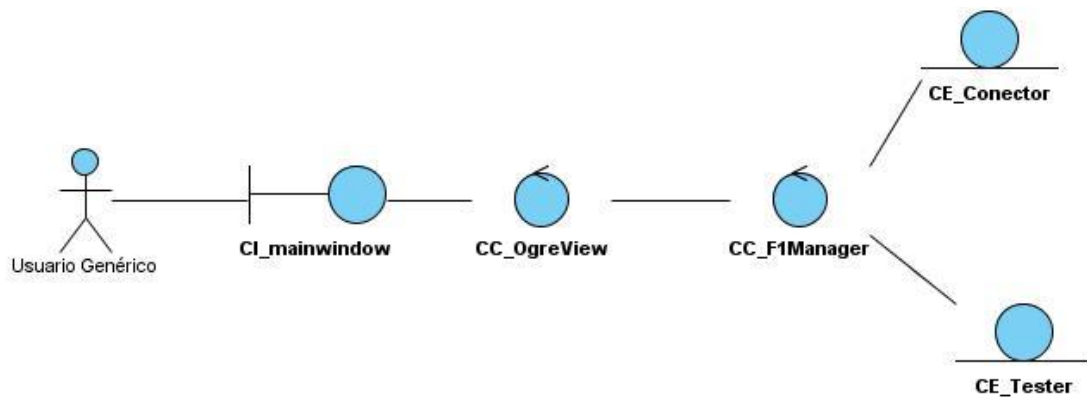
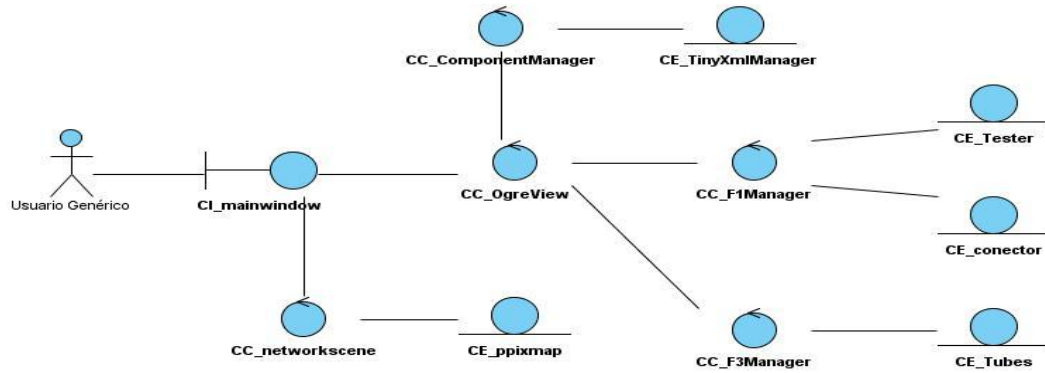


Figura 24: Diagrama de clases del análisis del CU Mostrar Errores.

# Características del Sistema

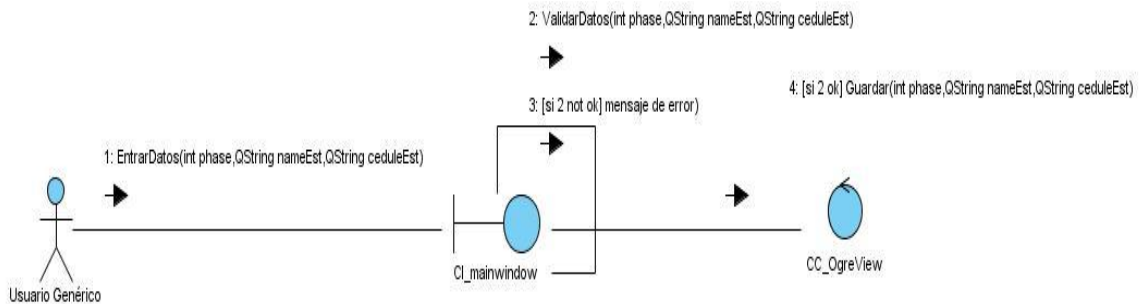


**Figura 25: Diagrama de clases del análisis del CU Seleccionar Objetos Gráficos.**

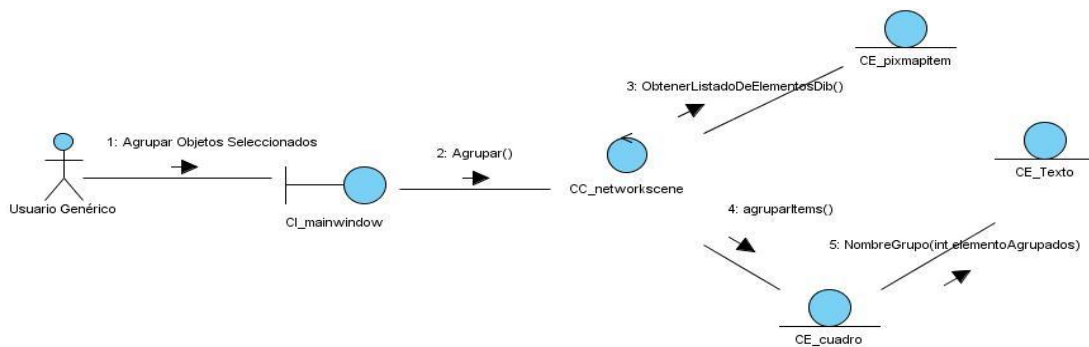
## 2.5.3 Diagrama de interacción del análisis (colaboración)

Los diagramas de colaboración son usados para modelar la dinámica del sistema a través de los objetos de las diferentes clases del análisis, sus relaciones y los mensajes que se puedan enviar entre ellos. Un diagrama de colaboración destaca la organización estructural de los objetos que envían y reciben mensajes. [1]

A continuación se muestran los diagramas de interacción (colaboración) del análisis correspondiente a los casos de uso del sistema:

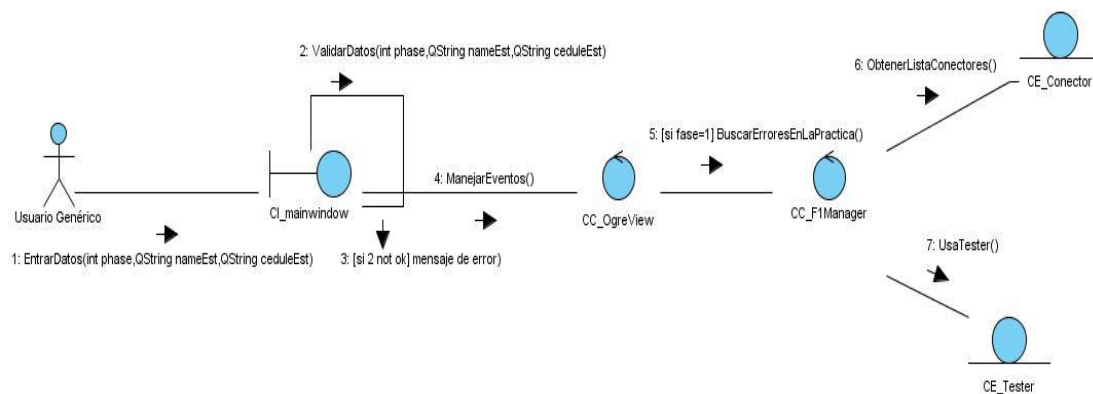


**Figura 26: Diagrama de interacción (colaboración) del análisis del CU Registrar.**

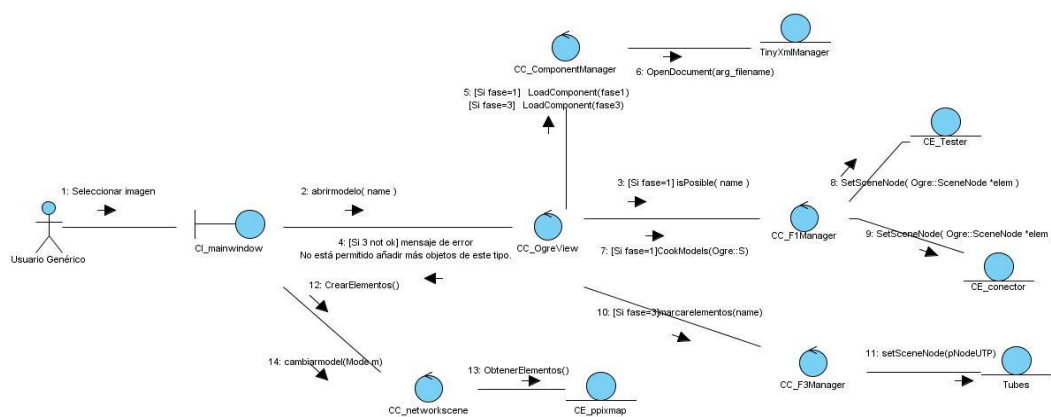


**Figura 27: Diagrama de interacción (colaboración) del análisis del CU Agrupar Objetos Gráficos.**

# Características del Sistema



**Figura 28: Diagrama de interacción (colaboración) del análisis del CU Mostrar Errores.**



**Figura 29: Diagrama de interacción (colaboración) del análisis del CU Seleccionar Objetos Gráficos.**

## Consideraciones parciales

Con la realización de este capítulo se logró identificar el actor del sistema, que será el encargado de interactuar con el mismo; se realizó la descripción de los casos de uso del sistema para un mejor entendimiento. Además se modeló los diagramas de clases del análisis y de iteración (colaboración), los cuales servirán de entrada y comprensión para la realización del diseño de la aplicación.

# *Diseño de la Aplicación*

## Capítulo 3: “Diseño de la aplicación”

### Introducción

En este capítulo se obtendrá el diseño de la aplicación donde quedarán plasmadas todas las clases principales de dicho programa. Además se realizará el diagrama de componentes, los cuales describen los elementos físicos del sistema y sus relaciones y se validarán los requisitos funcionales a partir de la matriz de trazabilidad.

### 3.1 Patrones de diseño

En la tecnología orientada a objetos, un patrón es una descripción del problema y la solución, a la que se da un nombre, y que se puede aplicar a nuevos contextos; idealmente, proporciona consejos sobre el modo de aplicarlo en varias circunstancias, y considera los puntos fuertes y compromisos. (26)

En el diseño del sistema se utilizaron los siguientes patrones para dar solución a las diferentes problemáticas.

#### **Patrones Grasp<sup>3</sup>:**

Experto:

Problema: ¿qué principio usar para asignar responsabilidades a los objetos?

Solución: asignar una responsabilidad al experto en información (la clase que tiene la información necesaria para realizar la responsabilidad).

Si esta actividad se realiza bien, los sistemas tienden a ser más fáciles de entender, mantener y ampliar, y existen más oportunidades para reutilizar componentes en futuras aplicaciones. El experto en información se utiliza con frecuencia en la asignación de responsabilidades; es un principio de guía básico que se utiliza continuamente en el diseño de objetos.

“El experto expresa la “intuición” común de que los objetos hacen las cosas relacionadas con la información que tienen. Se debe notar que el cumplimiento de la responsabilidad a menudo requiere información que se encuentra dispersa por diferentes clases de objetos”. (26)

Beneficios: Se mantiene el encapsulamiento de la información, puesto que los objetos utilizan su propia información para llevar a cabo las tareas. Normalmente, esto conlleva un bajo acoplamiento, lo que da lugar a sistemas más robustos y más fáciles de mantener.

Se distribuye el comportamiento entre las clases que contienen la información requerida, por tanto, se estimula las definiciones de clases más cohesivas y “ligeras” que son más fáciles de entender y mantener. Se soporta normalmente una alta cohesión.

---

<sup>3</sup> Son patrones generales de software para asignación de responsabilidades.

# *Diseño de la Aplicación*

## Creador:

Problema: ¿quién debería ser el responsable de la creación de una nueva instancia de alguna clase?

Solución: asignar a la clase B la responsabilidad de crear una instancia de la clase A si se cumple una o más de los casos siguientes:

- B agrega objetos de A.
- B contiene objetos de A.
- B registra instancias de objetos de A.
- B utiliza más estrechamente objetos de A.
- B tiene los datos de inicialización que se pasará a un objeto de A cuando sea creado, por tanto, B es un experto con respecto a la creación de A.

Si se asigna bien la responsabilidad de creación, el diseño puede soportar un bajo acoplamiento, mayor claridad, encapsulación y reutilización.

## Alta cohesión:

Problema: ¿cómo mantener la complejidad manejable?

Solución: asignar una responsabilidad de manera que la cohesión permanezca alta.

La cohesión es una medida de la fuerza con la que se relacionan y del grado de focalización de las responsabilidades de un elemento. Un elemento con responsabilidades altamente relacionadas, y que no hace una gran cantidad de trabajo, tiene alta cohesión. Estos elementos pueden ser clases, subsistemas, entre otros.

Una clase con baja cohesión hace muchas cosas no relacionadas, o hace demasiado trabajo. Tales clases no son convenientes, adolecen de los siguientes problemas:

1. Difíciles de entender.
2. Difíciles de reutilizar.
3. Difíciles de mantener.
4. Delicadas, constantemente afectadas por los cambios.

“Como regla empírica, una clase con alta cohesión tiene un número relativamente pequeño de métodos, con funcionalidad altamente relacionada, y no realiza mucho trabajo. Colabora con otros objetos para compartir el esfuerzo si la tarea es extensa”. (26)



# Diseño de la Aplicación

## Patrones Gof<sup>4</sup>:

### Singleton:

Permite la creación, a nivel de objetos. Tiene como propósito garantizar que una clase sólo tenga una única instancia, proporcionando un punto de acceso global a la misma. A continuación se muestra un ejemplo donde se aplica este patrón.

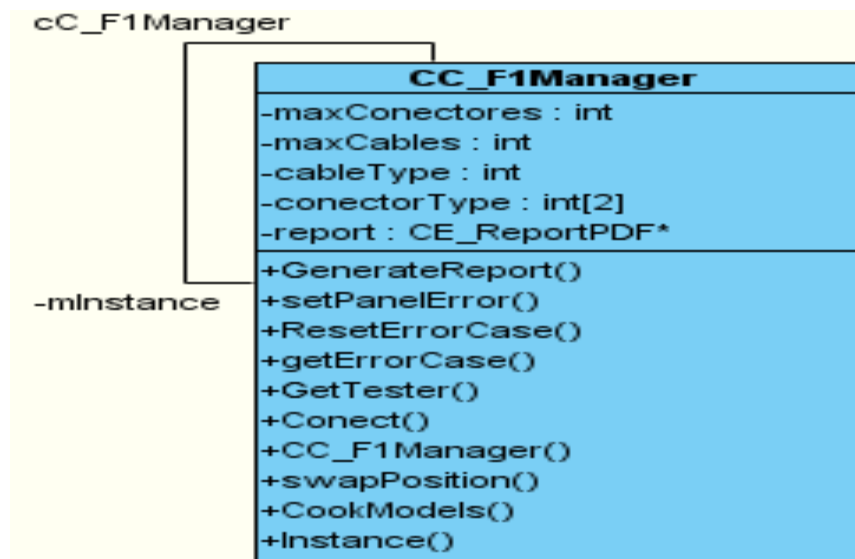


Figura 30: Uso del patrón Singleton.

## 3.2 Arquitectura del sistema

La arquitectura de una aplicación es la vista conceptual de la estructura de ésta. Toda aplicación contiene código de presentación, de procesamiento y de almacenamiento de datos. La arquitectura de los sistemas difiere según como está distribuido el código.

Se utilizará en el diseño del sistema la arquitectura de n\_capas (3 capas). Esta “se enfoca en la distribución de roles y responsabilidades de forma jerárquica, brindando una forma muy efectiva de separación de responsabilidades. El rol indica el modo y tipo de interacción con otras capas, y la responsabilidad indica la funcionalidad que está siendo desarrollada”. (27)

La arquitectura en capas se encuentra centrada básicamente en dividir un problema en pequeñas partes que puedan ser manejadas e implementadas de forma independiente, dichas partes poseerán responsabilidades específicas que no dependan del funcionamiento de las otras, o al menos que su dependencia sea mínima. Este aspecto constituye una ventaja considerable, pues proporciona una

<sup>4</sup> Son los patrones de creación, estructurales y de comportamiento. describen 23 patrones de diseño comúnmente utilizados y de gran aplicabilidad en problemas de diseño usando modelamiento UML.

# Diseño de la Aplicación

amplia reutilización de las clases implementadas, al hacer abstracciones de las distintas funcionalidades o responsabilidades del sistema agrupándolas en capas.

Las tres capas son:

- Capa de presentación: resuelve la presentación de datos al usuario, posibilitando la interacción entre él y el sistema. En ella están contenidas todas las interfaces a través de las cuales el usuario puede decirle a la aplicación que almacene determinados datos, o pedirle datos ya almacenados y que él necesita.
- Capa de lógica del negocio: resuelve la lógica de la aplicación. Contiene los algoritmos, validaciones y coordinación necesaria para resolver la problemática. Los elementos fundamentales de esta capa son los objetos de dominio. Estos objetos se representan como los objetos principales del negocio y contienen la lógica del mismo. Esta capa contiene las soluciones de los problemas planteados en el estudio del negocio.
- Capa de acceso a datos: resuelve el acceso a datos, abstrayendo a su capa superior, aislándola de la complejidad del acceso e interacción con los diferentes orígenes de datos y suministrándole a su vez lo que el usuario necesita.



Figura 31: Arquitectura tres capas. (28)

## 3.3 Diagrama de paquetes

Se utilizan para reflejar la organización de paquetes y sus elementos. Se usa frecuentemente para organizar diagramas de casos de uso y de clase. Se pueden construir para representar relaciones tanto físicas como lógicas. (29) En la siguiente figura se muestra el diagrama de paquetes de la aplicación.

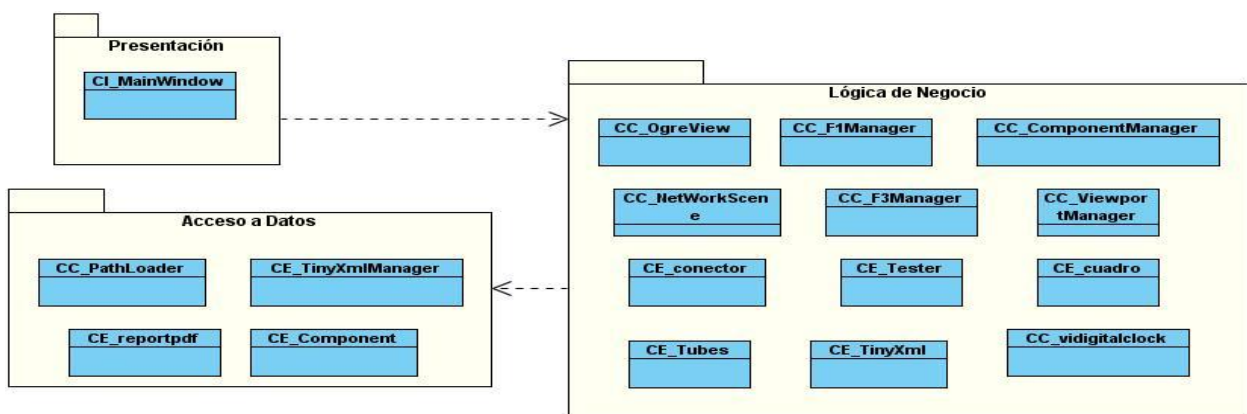


Figura 32: Diagrama de paquetes.

# Diseño de la Aplicación

## 3.3.1 Descripción de las clases

A continuación se muestra la descripción de las clases principales del diseño:

**MainWindow:** Representa la clase interfaz. Muestra al usuario las funcionalidades del sistema.

**OgreView:** Se encarga de manejar los eventos en la escena (mover objetos).

**netWorkScene:** Contiene la información necesaria para dar soporte al diseño de una red LAN. Constituye la fase2.

**F1Manager:** Representa la fase 1, es decir, permite que el usuario pueda construir un cable de red.

**F3Manager:** Brinda las funcionalidades necesarias para que el usuario una vez registrado instale una red de área local.

**ComponentManager:** Controla los elementos de una red, tales como: router, switch, pc, entre otros.

**TinyXmlManager:** Carga los componentes de la red, a partir de un fichero XML.

**Reportpdf:** Salva un archivo PDF que contiene los resultados de la práctica del usuario.

## 3.4 Modelo de diseño

El diseño ocupa el final de la fase de elaboración y el comienzo de las iteraciones de construcción. Contribuye a una arquitectura estable y sólida. En el diseño se modela el sistema teniendo en cuenta la arquitectura, para lograr que soporte todos los requisitos, funcionales no funcionales y las restricciones que se le suponen.

### 3.4.1 Diagrama de clases del diseño

Las clases de diseño son abstracciones de clases claramente aprovechables en la implementación del software. Sus relaciones tienen un significado directo en el lenguaje de programación. En una clase del diseño, los métodos son los mismos que en una clase implementada.

A continuación se representarán los diagramas de clases del diseño correspondientes a los CUS:

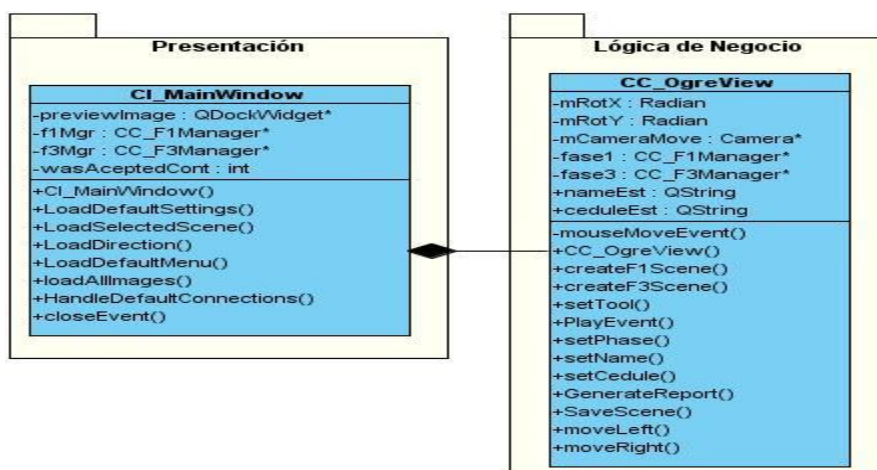


Figura 33: Diagrama de clases del diseño CU Registrar.

# Diseño de la Aplicación

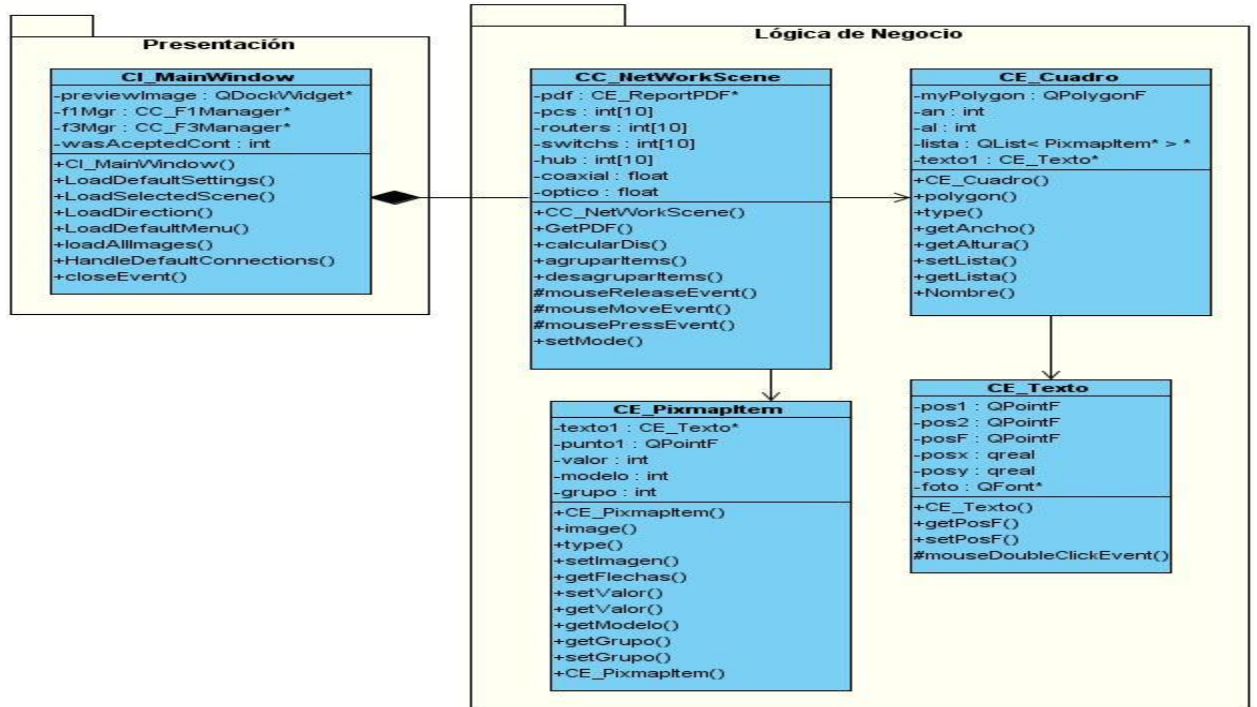


Figura 34: Diagrama de Clases del diseño del CU Agrupar Objetos Gráficos.

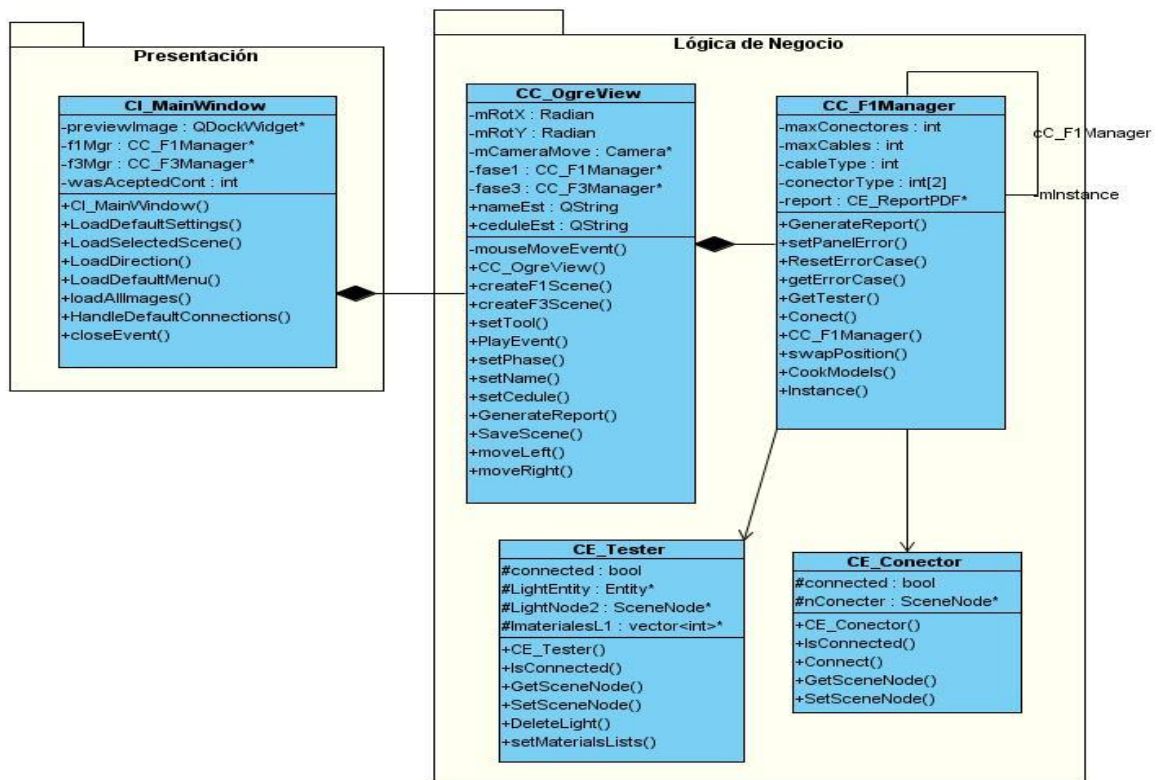


Figura 35: Diagrama de Clases del diseño del CU Mostrar Errores.



# Diseño de la Aplicación

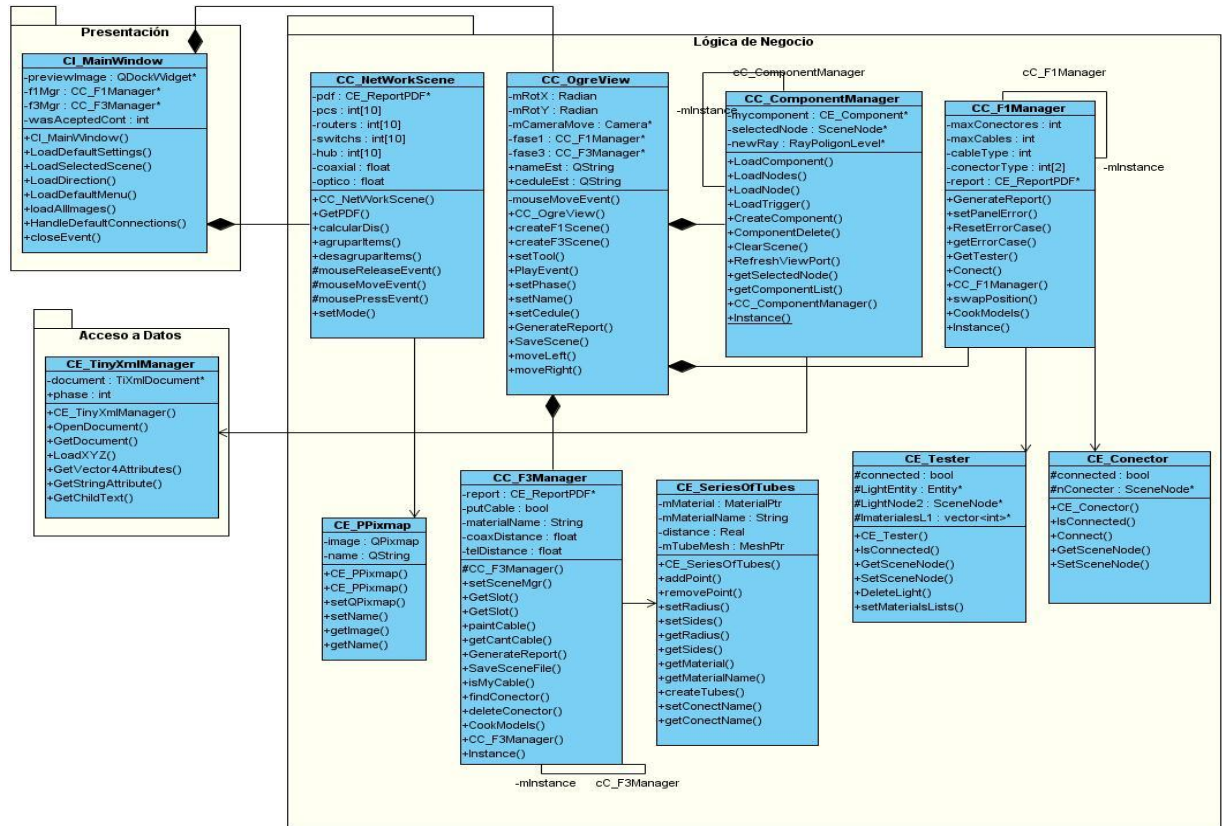


Figura 36: Diagrama de clases del diseño del CU Seleccionar Objetos Gráficos.

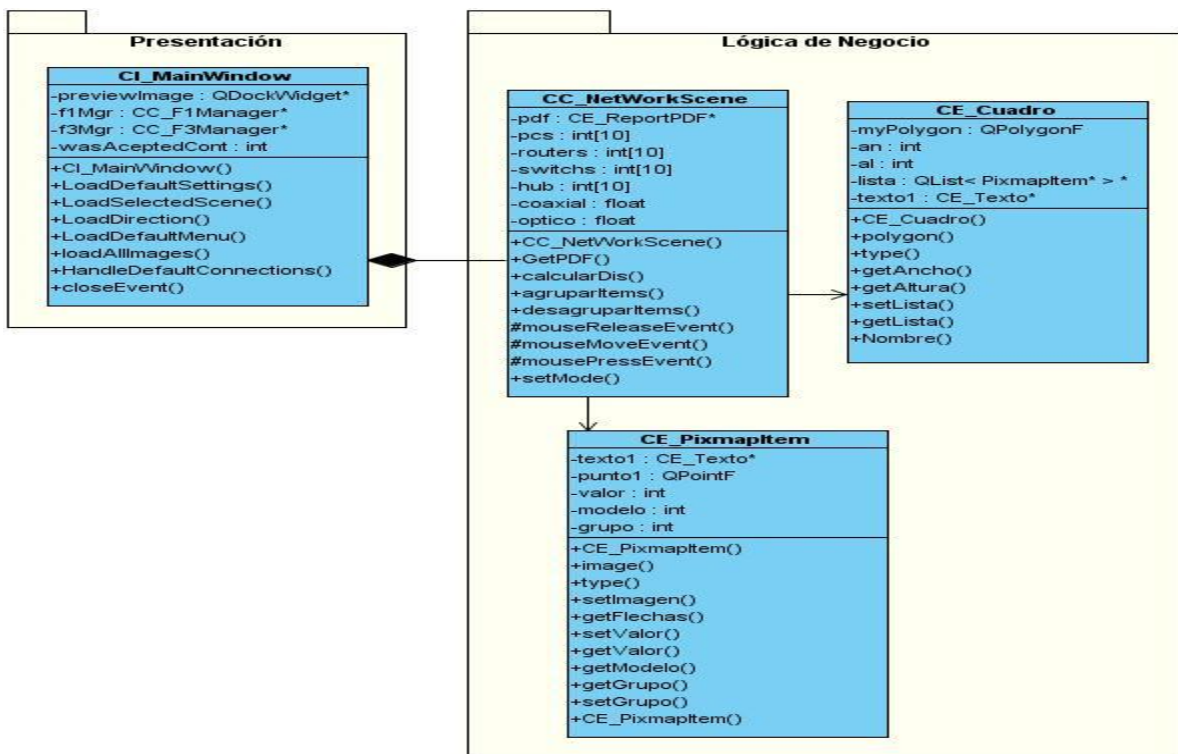


Figura 37: Diagrama de Clases del diseño del CU Desagrupar Objetos Gráficos.

# Diseño de la Aplicación

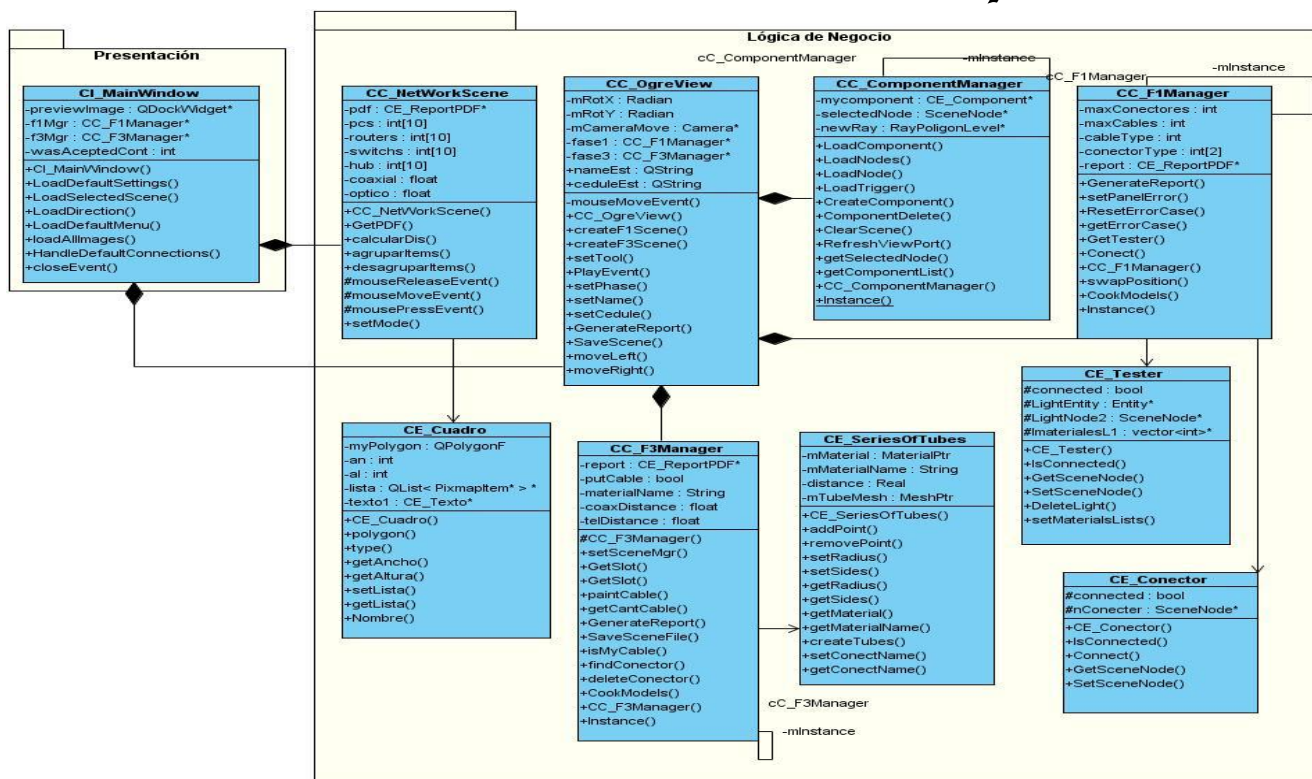


Figura 38: Diagrama de clases del diseño del CU Eliminar Objetos Gráficos.

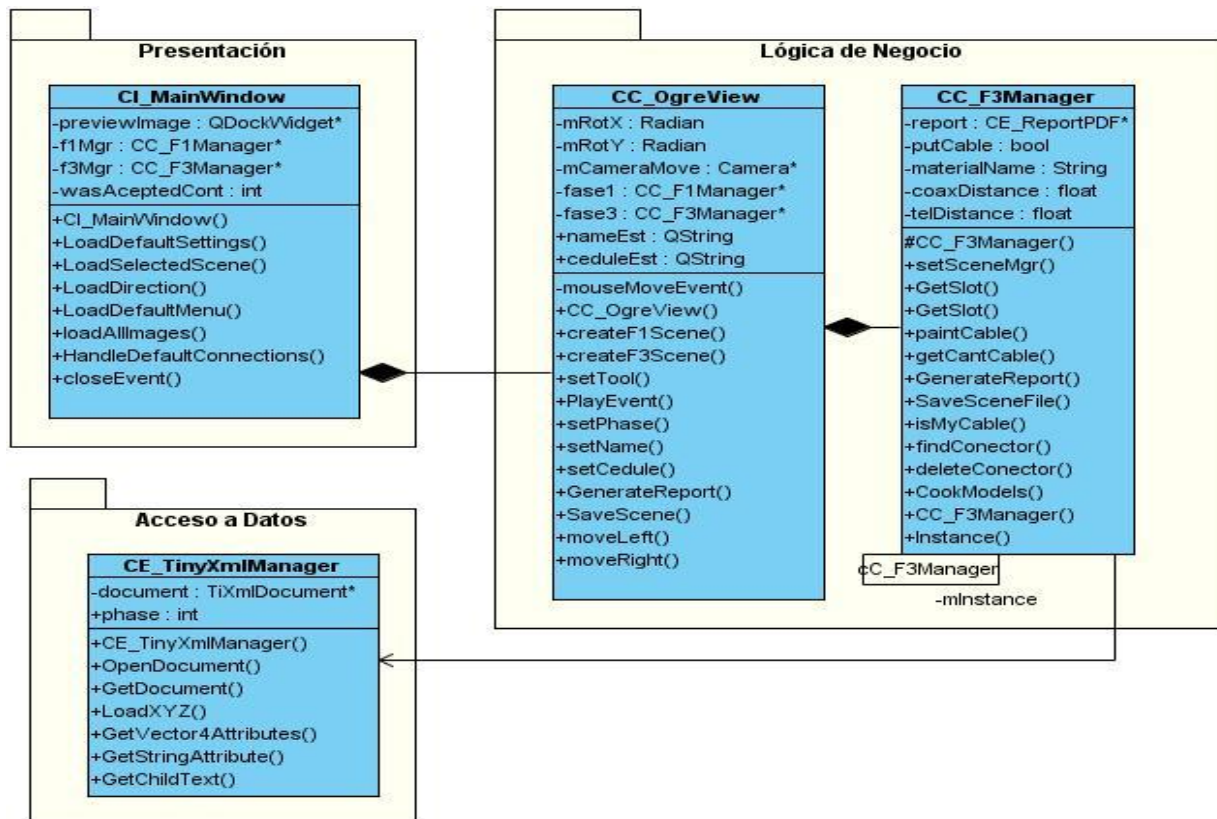


Figura 39: Diagrama de Clases del diseño del CU Exportar Escena.



# Diseño de la Aplicación

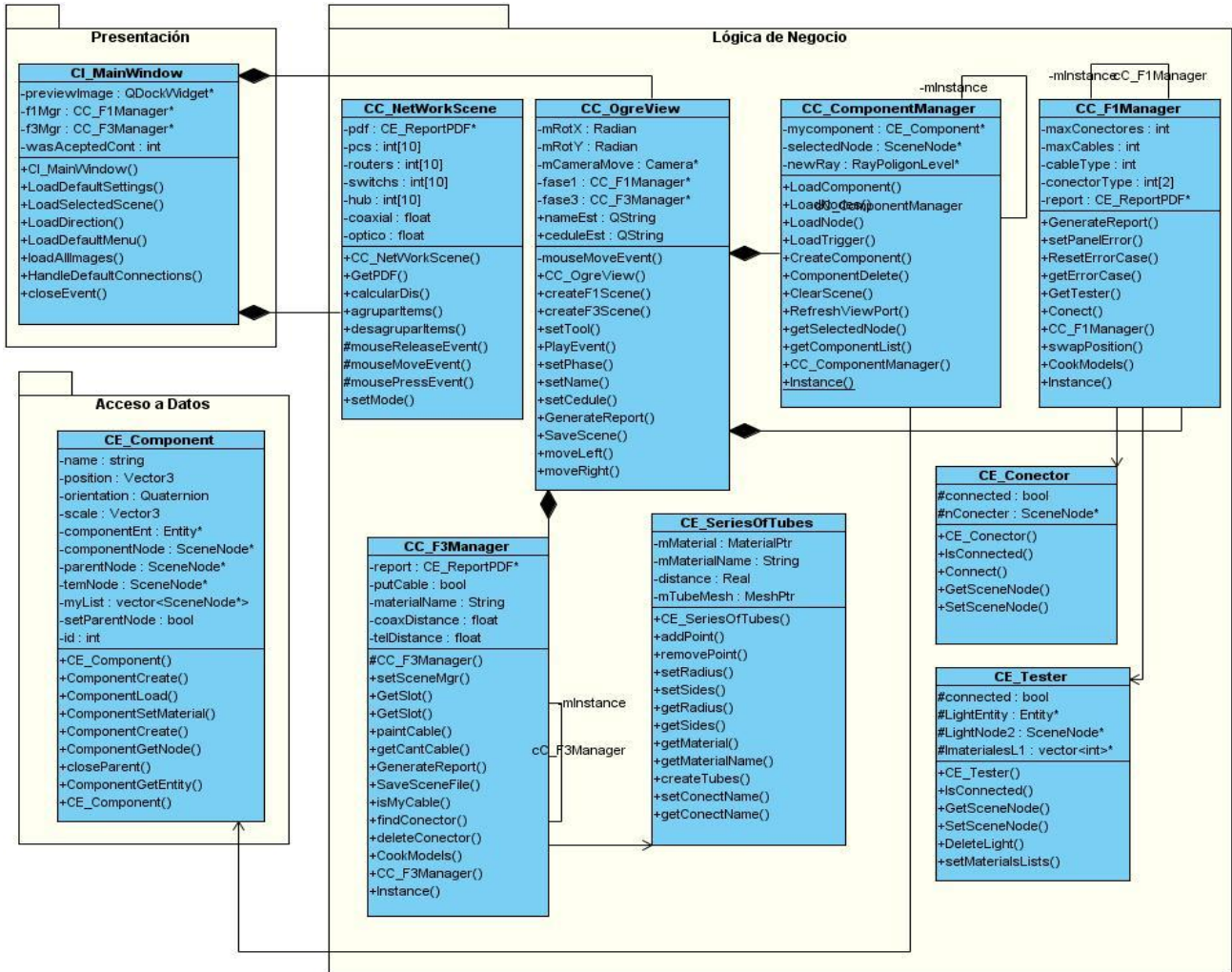


Figura 40: Diagrama de clases del diseño del CU Limpiar Escena.

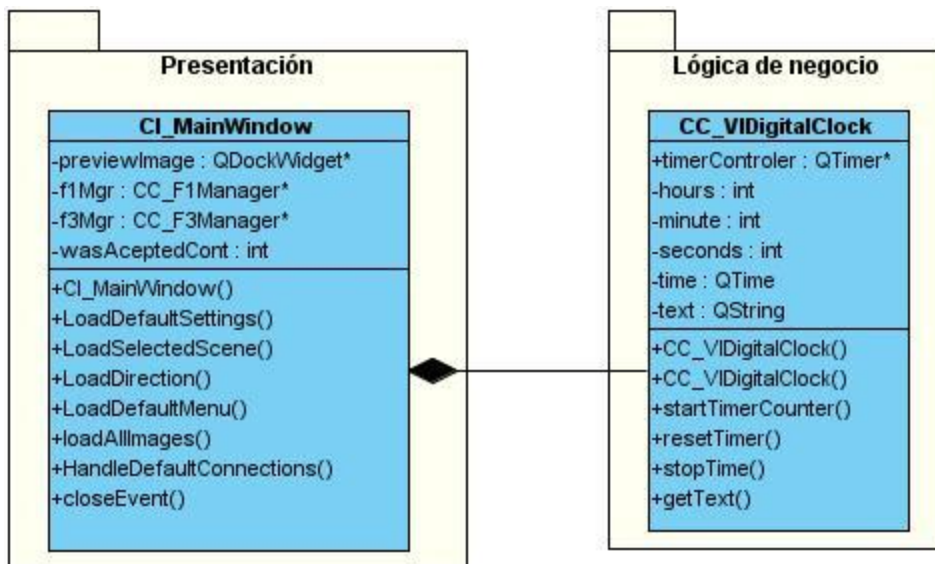
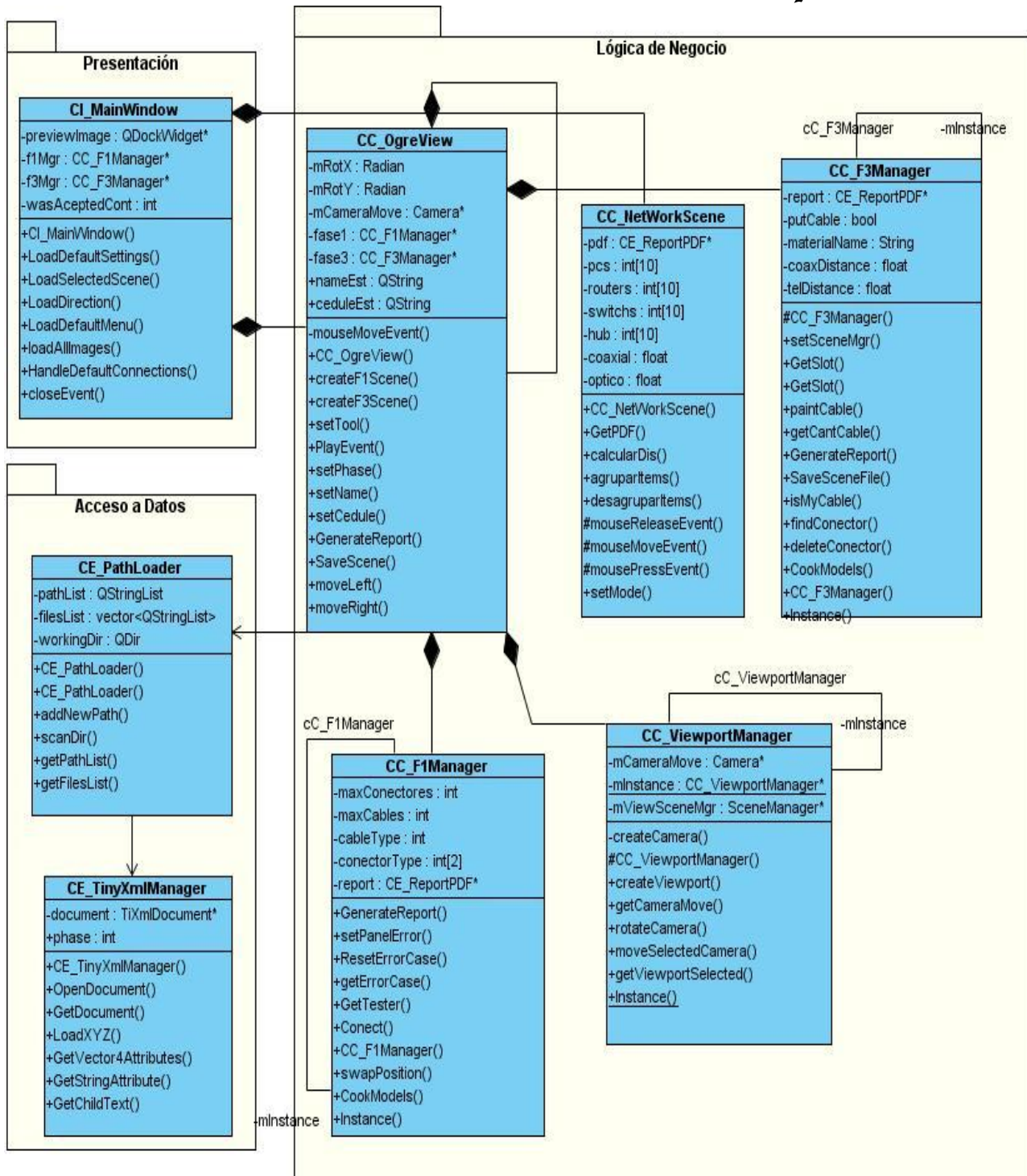


Figura 41: Diagrama de Clases del diseño del CU Mostrar Tiempo de Duración de la Práctica.

# Diseño de la Aplicación



**Figura 42: Diagrama de Clases del diseño del CU Mostrar Objetos de la Escena.**



# Diseño de la Aplicación

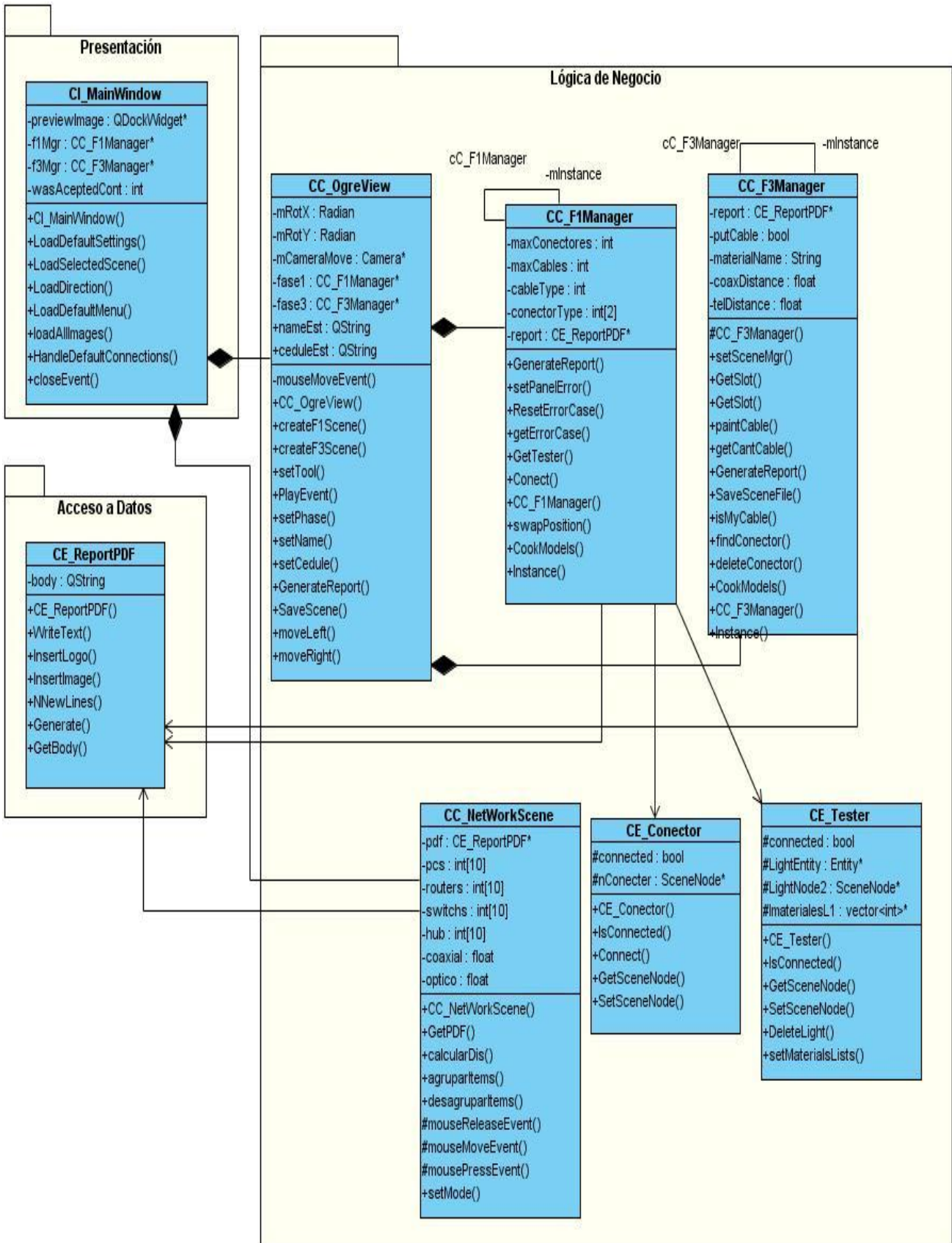


Figura 43: Diagrama de Clases del diseño del CU Salvar Archivos.

# Diseño de la Aplicación

## 3.5 Modelo de Implementación

El Modelo de Implementación describe como las clases, se implementan en términos de componentes, ficheros de código fuente, ejecutables, etc. Especifica cómo se organizan los componentes de acuerdo con los mecanismos de estructuración y modelación disponible en el entorno de implementación y en el lenguaje o lenguajes de programación utilizados. [3]

### 3.5.1 Diagrama de componentes

Los diagramas de componentes describen los elementos físicos del sistema y sus relaciones. Muestran las opciones de realización incluyendo código fuente, binario y ejecutable. Los componentes representan todos los tipos de elementos software que entran en la fabricación de aplicaciones informáticas. Son usados para estructurar el modelo de implementación en términos de subsistemas de implementación y mostrar las relaciones entre sus elementos. Los componentes representados pueden ser datos, archivos, directorios y otros. [3] A continuación se muestran los correspondientes al sistema desarrollado.

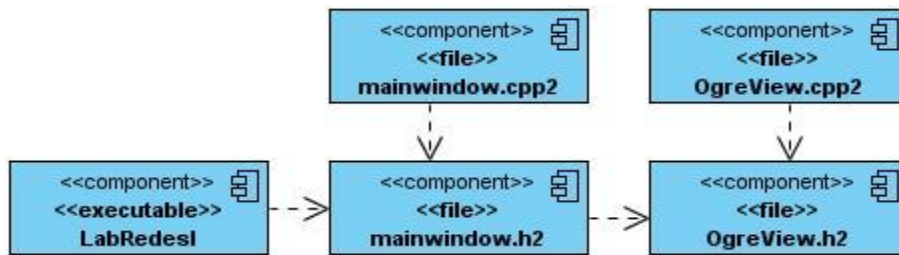


Figura 44: Diagrama de componentes del CU Registrar.

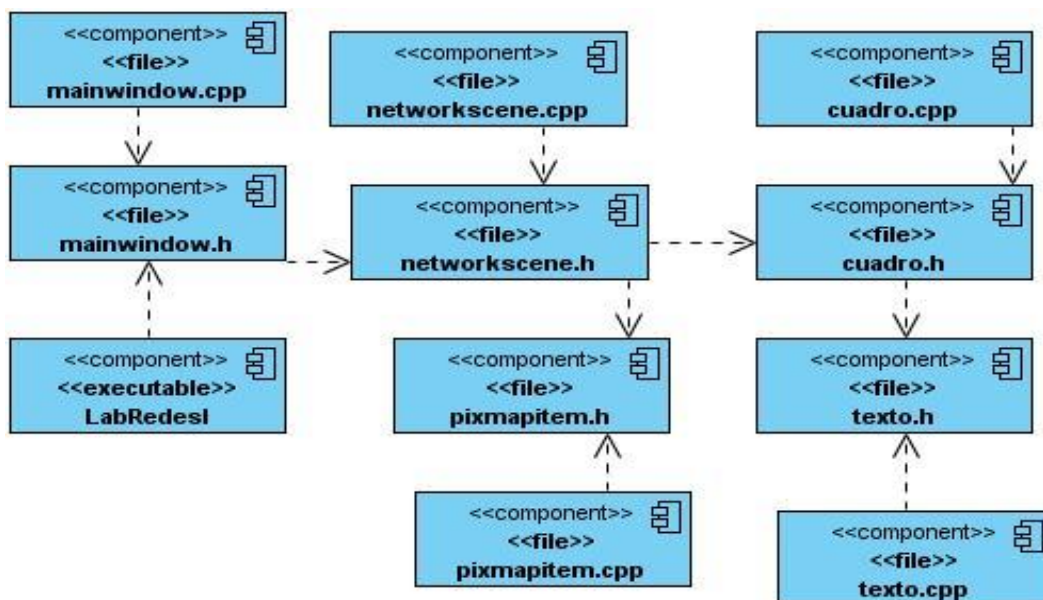


Figura 45: Diagrama de componente del CU Agrupar Objetos Gráficos.

# Diseño de la Aplicación

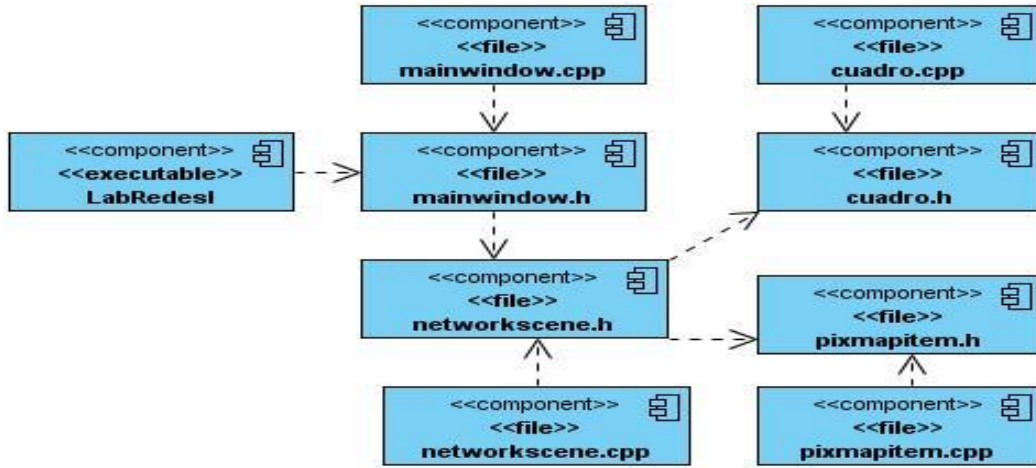


Figura 46: Diagrama de componente del CU Desagrupar Objetos Gráficos.

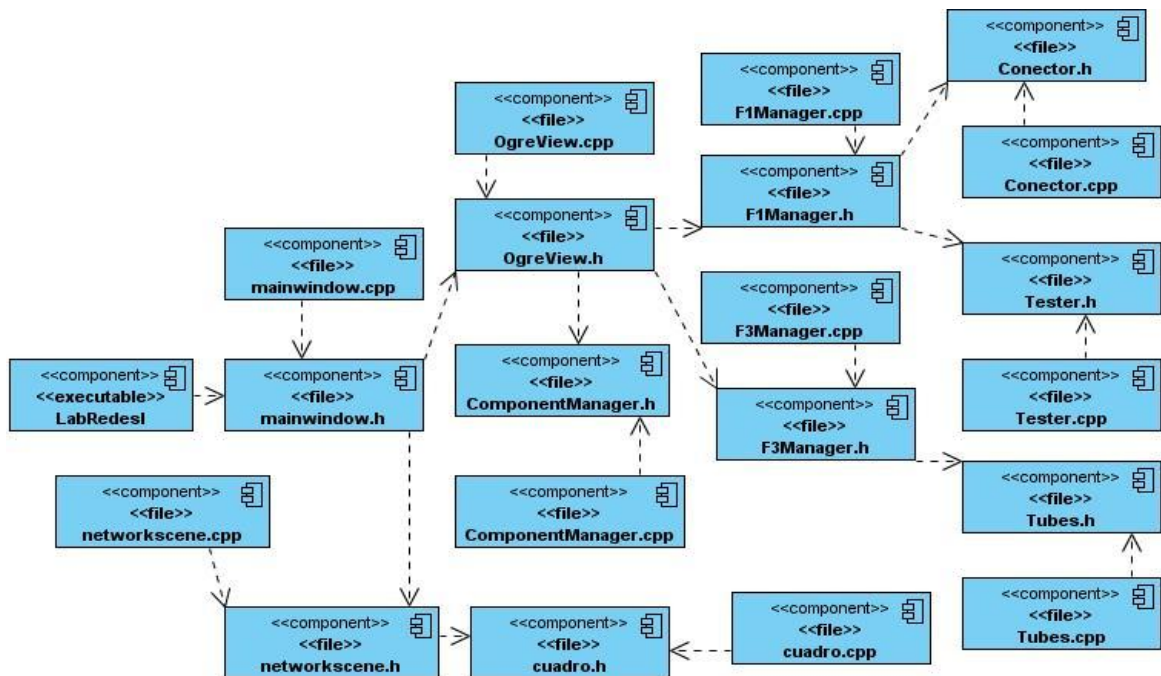


Figura 47: Diagrama de componente del CU Eliminar Objetos Gráficos.

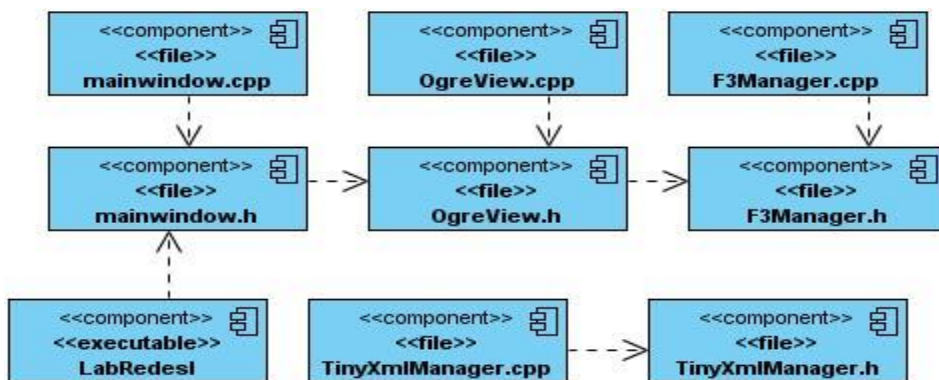


Figura 48: Diagrama de componente del CU Exportar Escena.



# Diseño de la Aplicación

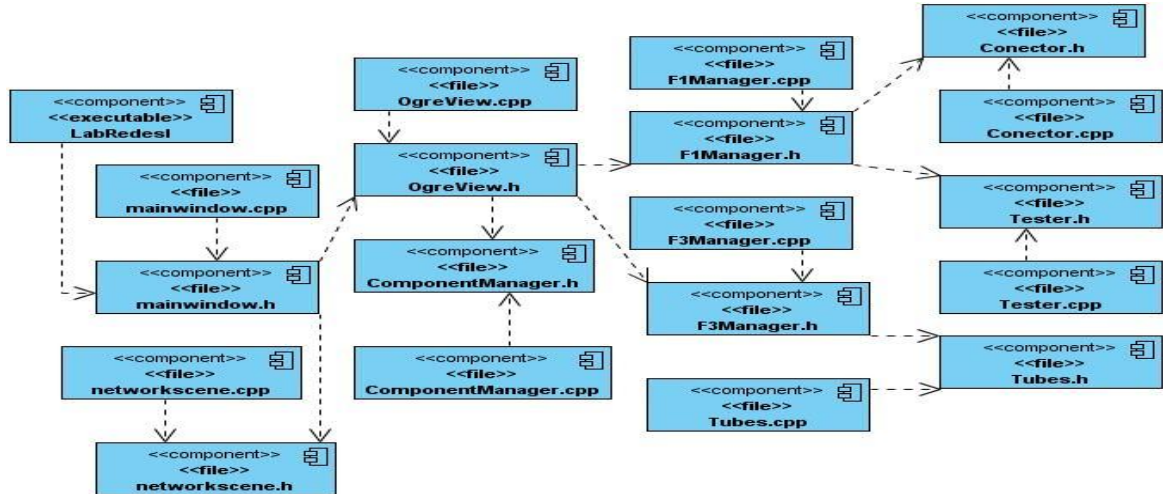


Figura 49: Diagrama de componente del CU Limpiar Escena.

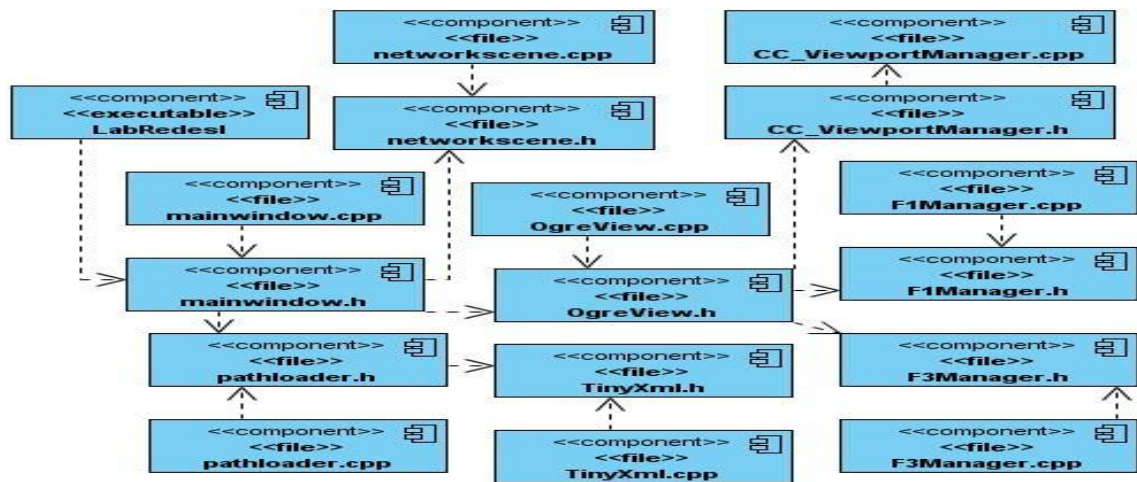


Figura 50: Diagrama de componente del CU Mostrar Objetos de la Escena.

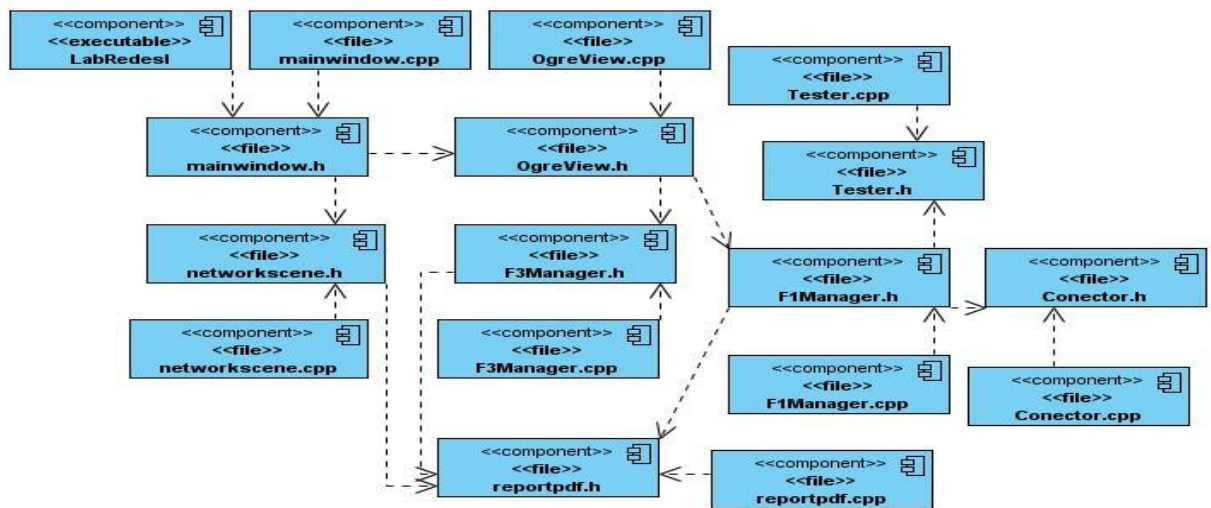


Figura 51: Diagrama de componente del CU Salvar Archivos.

# *Diseño de la Aplicación*

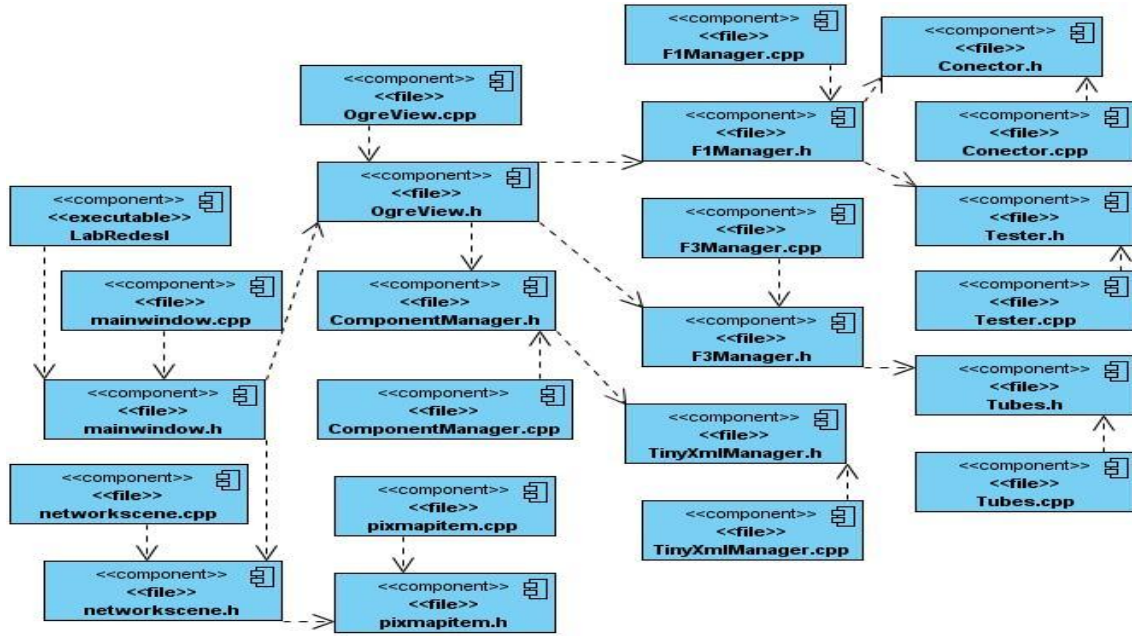


Figura 52: Diagrama de componente del CU Seleccionar Objetos Gráficos.

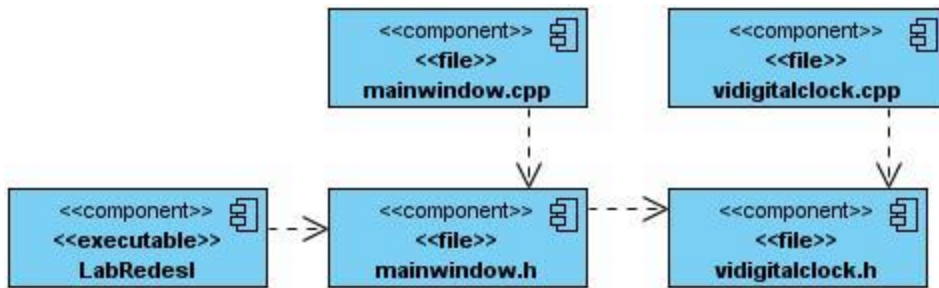


Figura 53: Diagrama de componente del CU Mostrar Tiempo de Duración de la Práctica.

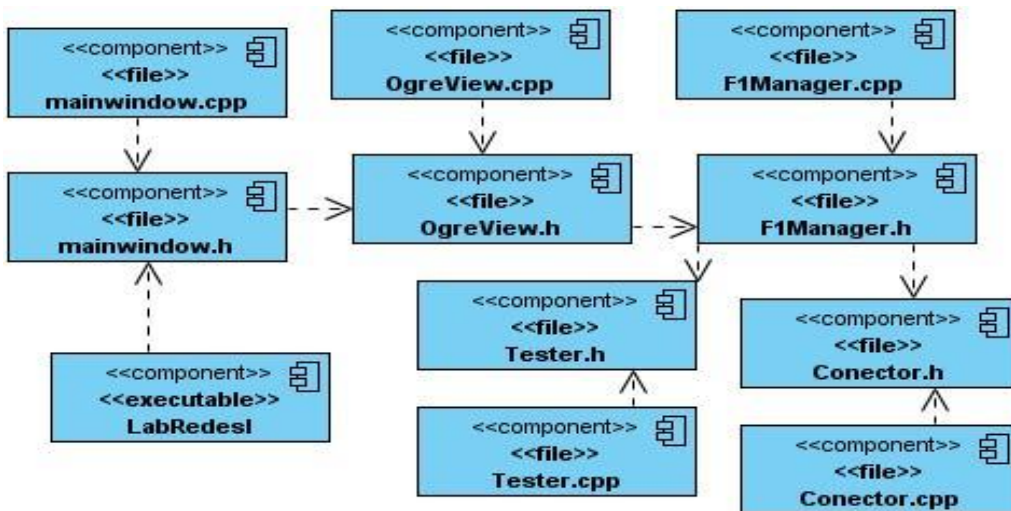


Figura 54: Diagrama de componente del CU Mostrar Errores.

# Diseño de la Aplicación

## 3.6 Diagrama de despliegue

Los Diagramas de Despliegue muestran las relaciones físicas de los distintos nodos que componen un sistema y el reparto de los componentes sobre dichos nodos. Un nodo es un objeto físico en tiempo de ejecución que representa un recurso computacional, generalmente con memoria y capacidad de procesamiento. (29) A continuación se muestra el diagrama de despliegue del sistema:



Figura 55: Diagrama de despliegue.

## 3.7 Matriz de trazabilidad

Durante el ciclo de vida del desarrollo de software de un proyecto existen dos fases muy importantes como son la definición de requisitos o especificaciones y el diseño y ejecución de pruebas.

Cuando el proyecto es muy grande o complejo es difícil poder saber que test ejecutados o diseñados cubren cada una de las especificaciones o requerimientos del proyecto. Es por este motivo que existe lo que se conoce como la matriz de trazabilidad. (30)

La matriz de trazabilidad es una herramienta que se utiliza para saber que requerimientos quedan cubiertos por una prueba. A continuación se muestra un ejemplo:

Se tiene un proyecto con 5 requerimientos (R1-R5) y se ha diseñado tres casos de prueba (T1-T3).

- El caso de prueba T1 cubre los requerimientos R1 y R4.
- El caso de prueba T2 cubre los requerimientos R3 y R5.
- El caso de prueba T3 cubre el requerimiento R3.

En este caso la matriz resultante será:

	T1	T2	T3
R1	x		
R2			
R3		x	x
R4	x		
R5		x	

Observando la matriz se muestra claramente dos cosas:

1. El requerimiento R3 está probado en 2 casos de prueba.

# *Diseño de la Aplicación*

2. El requerimiento R2 no está cubierto.

Gracias a estos datos se puede ver que partes o módulos del software no están cubiertos y deberían probarse por otras pruebas o identificar los requerimientos más críticos para saber si están suficientemente cubiertos (más de un caso de prueba es diseñado y ejecutado para ese requerimiento). (30)

También podemos identificar los casos de prueba que han fallado y a partir de ahí ver que requerimiento está en riesgo para poder evaluar la criticidad y el riesgo de este. A continuación se muestra la matriz de trazabilidad correspondiente al proyecto de redes.

### Casos de Uso del sistema:

CUS 1: Registrar

CUS 2: Mostrar Objetos de la Escena

CUS 3: Salvar Archivo

CUS 4: Eliminar Objetos Gráficos

CUS 5: Mostrar Tiempo de Duración de la Práctica

CUS 6: Limpiar Escena

CUS 7: Mostrar Errores

CUS 8: Agrupar objetos Gráficos

CUS 9: Desagrupar Objetos Gráficos

CUS 10: Exportar Escena

CUS 11: Seleccionar Objeto Gráfico

Requisitos / Casos de Uso	CUS 1	CUS 2	CUS 3	CUS 4	CUS 5	CUS 6	CUS 7	CUS 8	CUS 9	CUS 10	CUS 11
RF Registrar	X										
RF Mostrar Objetos de la Escena		X									
RF Salvar Archivo			X								
RF Eliminar Objetos Gráficos				X							
RF Mostrar Tiempo de Duración de la Práctica					X						
RF Limpiar Escena						X					
RF Mostrar Errores							X				
RF Agrupar Objetos Gráficos								X			
RF Desagrupar Objetos Gráficos									X		
RF Exportar Escena										X	
RF Seleccionar objeto Gráfico											X

**Tabla 14: Matriz de trazabilidad.**

# *Diseño de la Aplicación*

## **Consideraciones parciales**

Con el desarrollo de este capítulo se concluyeron todos los diagramas de clases del diseño del sistema. Además se realizó el diagrama de componentes que describen los elementos físicos del sistema y sus relaciones y la matriz de trazabilidad para validar los requisitos del sistema.



# *Conclusiones Generales*

## **Conclusiones generales**

Con la realización de la ingeniería inversa del módulo de diseño e instalación de una red de área local (LAN), perteneciente al proyecto de Laboratorios Virtuales de la facultad 5 y luego de haber cumplido con las tareas de investigación científica, se ha arribado a las siguientes conclusiones:

- Se trataron conceptos importantes relacionados con el problema en cuestión para su mejor entendimiento.
- Se realizó un estudio de la herramienta, lenguaje y metodología informática para un mejor análisis de la aplicación.
- Se definieron los requerimientos funcionales y no funcionales, los que permitieron tener una guía de las funcionalidades del sistema.
- Se realizó el análisis y el diseño del sistema, lo que dio la posibilidad de obtener una guía de la aplicación desarrollada.
- Se realizó la matriz de trazabilidad para validar los requisitos del sistema.

## **Recomendaciones**

Una vez concluido el análisis y diseño y demás documentación del módulo de redes perteneciente al proyecto Laboratorios Virtuales de la facultad 5 se recomienda:

- Agregar nuevas funcionalidades a la aplicación que mejoren la comunicación del usuario con estas herramientas educativas.
- Desarrollar una ayuda más interactiva, que responda al momento a las necesidades del usuario.
- Poner este laboratorio al alcance de los estudiantes de la UCI para su puesta en práctica y hacerlo extensible a todas las universidades del país.
- Colocar en el laboratorio un video donde se simule la instalación de una red, que sirva de apoyo al usuario a la hora de realizar la práctica, en caso de que el mismo tenga dudas.
- Agregar al proyecto un módulo de comunicación que permita intercambiar información entre los usuarios que realicen la práctica.

# Bibliografía Referenciada

## Bibliografía Referenciada

1. Very, P. V. Informe de reunión de expertos sobre laboratorios virtuales. [Consultado el 8 de febrero del 2010]. [Disponible en : <http://unesdoc.unesco.org/images/0011/001191/119102s.pdf>]
2. [En línea] [Consultado el: 7 de diciembre de 2010]. [Disponible en: <http://es.kioskea.net/contents/initiation/lan.php3>]
3. Redes de área local. [En línea][Consultado el: 20 de enero de 2011]. [Disponible en: <http://es.kioskea.net/contents/lan/lanintro.php3> ]
4. LAN. [En línea] [Consultado el: 20 de enero de 2011]. [Disponible en: <http://es.kioskea.net/contents/initiation/lan.php3> ]
5. [En línea] [Consultado el: 7 de diciembre de 2010]. [Disponible en: <http://es.kioskea.net/contents/initiation/lan.php3>]
6. Modelo Cliente-Servidor [Consultado el 20 de mayo del 2011]. [Disponible en: <http://www.google.com/search?um=1&hl=es&biw=1024&bih=629&tbm=isch&sa=1&q=componentes+de+una+red+lan&aq=f&aql=&aql=&oq=>]
7. Wikitel [En línea] [Consultado el 8 de febrero del 2010]. [Disponible en : [http://es.wikel.info/wiki/Cable\\_coaxial](http://es.wikel.info/wiki/Cable_coaxial)]
8. Fibra óptica [En línea] [Consultado el 8 de febrero del 2010]. [Disponible en : <http://neo.lcc.uma.es/evirtual/cdd/tutorial/fisico/fibra.html>]
9. Valle, P, P. Informática en la administración pública [Consultado el: 7 de febrero de 2011]. [Disponible en: <http://www.mailxmail.com/curso-informatica-administracion-publica-1/topologia-red-fisica-logica> ]
10. Cicilia, M, A. Qué es la Ingeniería Inversa [Consultado el: 7 de diciembre de 2010]. [Disponible en: <http://cnx.org/content/m17432/latest/> ]
11. Sitio de software. [En línea] 29 de 5 de 2009. [Consultado el: 31 de 02 de 2011.] [Disponible en: [http://www.freedownloadmanager.org/es/downloads/ingenier%C3%ADa\\_inversa\\_gratis/](http://www.freedownloadmanager.org/es/downloads/ingenier%C3%ADa_inversa_gratis/) ]
12. Universidad Militar Nueva Granada. Laboratorio de redes Química [Consultado el: 8 de diciembre de 2010]. [Disponible en: <http://www.umng.edu.co/www/section-6962.jsp> ]
13. Días, M, M. Escuela de Ingeniería de Sistemas sociales [Consultado el: 8 de diciembre de 2010]. [Disponible en: <http://www.usmp.edu.pe/publicaciones/boletin/fia/info49/articulos/RUP%20vs.%20XP.pdf> ]
14. Laboratorio virtual [Consultado el 20 de mayo del 2011]. [Disponible en: <http://www.cea-ifac.es/actividades/jornadas/XXIV/documentos/econ/44.pdf>]
15. Días, M, M. Escuela de Ingeniería de Sistemas sociales [Consultado el: 8 de diciembre de 2010]. [Disponible en: <http://www.usmp.edu.pe/publicaciones/boletin/fia/info49/articulos/RUP%20vs.%20XP.pdf> ]

# Bibliografía Referenciada

16. Jacobson, Ivar, Grady y Rumbaugh, James. El proceso Unificado de Desarrollo de Software. La Habana: Feleix Varela, 2004
17. Metodología RUP. [Consultado el 20 de mayo del 2011]. [Disponible en: <http://www.google.com/search?um=1&hl=es&biw=1024&bih=629&tbm=isch&sa=1&q=metodologia+rup&aq=f&aqi=&aql=&oq>]
18. [En línea] [Consultado el: 9 de diciembre de 2010]. [Disponible en: <http://docirs.cl/uml.htm>]
19. UML. [En línea][Consultado el: 9 de diciembre de 2010]. [Disponible en: <http://www.google.com.cu/images?hl=es&q=el+vocabulario+de+UML&um=1&ie=UTF-8&source=og&sa=N&tab=wj>]
20. Vocabulario de UML. [Consultado el 20 de mayo del 2011]. [Disponible en: <http://www.google.com/search?um=1&hl=es&biw=1024&bih=629&tbm=isch&sa=1&q=vocabulario+de+UML&aq=f&aqi=&aql=&oq>]
21. Mora, J, A. Herramientas UML [Consultado el: 9 de diciembre de 2010]. [Disponible en: <http://www.jams.name/2010/04/18/herramientas-uml-cual-utilizar/> ]
22. 2007. Visitado en el Sitio Universidad de Jaén (Departamento de Informática). [Consultado el 5 de febrero del 2010]. [Disponible en: <http://wwdi.ujaen.es/asignaturas/isg/Software.html> ]
23. Peláez, Juan Carlos. Extractado de La Guía de Arquitectura. [En línea] 29 de 5 de 2009. [Consultado el: 31 de 01 de 2011.] [Disponible en: <http://geeks.ms/blogs/jkpelaez/archive/2009/05/29/arquitectura-basada-en-capas.aspx.>]
24. Visual Paradigm [Consultado el 20 de mayo del 2011]. [Disponible en: <http://www.google.com/search?um=1&hl=es&biw=1024&bih=629&tbm=isch&sa=1&q=visual+paradigm&aq=o&aqi=&aql=&oq>]
25. EVA, Introducción a la disciplina de análisis y diseño [Consultado el: 22 de febrero del 2011]. [Disponible en: <http://eva.uci.cu/mod/resource/view.php?id=35381> ]
26. Lescano, German. Patrones fundamentales en el diseño orientado a objetos (DOO). [En línea] [Consultado el: 26 de 04 de 2010] [Disponible en: <http://germanlescano.wordpress.com/2010/03/10/patrones-.>]
27. Peláez, Juan Carlos. Extractado de La Guía de Arquitectura. [En línea] 29 de 5 de 2009. [Consultado el: 31 de 01 de 2011.] [Disponible en: <http://geeks.ms/blogs/jkpelaez/archive/2009/05/29/arquitectura-basada-en-capas.aspx.>]
28. Arquitectura tres capas [Consultado el 20 de mayo del 2011]. [Disponible en: <http://www.google.com/search?um=1&hl=es&biw=1024&bih=629&tbm=isch&sa=1&q=arquitectura+tres+capas&aq=f&aqi=g1&aql=&oq>]
29. Diagrama de despliegue [Consultado el 20 de mayo del 2011]. [Disponible en: <http://www.google.com/url?sa=t&source=web&cd=7&ved=0CCwQFjAG&url=http%3A%2F%>

# *Bibliografía Referenciada*

[2Fvirtual.usalesiana.edu.bo%2Fweb%2Fpractica%2Farchiv%2Fdespliegue.doc&rct=j&q=diagrama%20de%20despliegue&ei=SO3qTfzlCsbV0QG7qfi6AQ&usg=AFQjCNGu7s\\_JDdYCEwEmbkYnxa2XI2Q2Q&cad=rja](http://virtual.usalesiana.edu.bo/web/practica/archiv/despliegue.doc&rct=j&q=diagrama%20de%20despliegue&ei=SO3qTfzlCsbV0QG7qfi6AQ&usg=AFQjCNGu7s_JDdYCEwEmbkYnxa2XI2Q2Q&cad=rja) ]

30. Lescano, German. Patrones fundamentales en el diseño orientado a objetos (DOO). [En línea] [Consultado el: 26 de 04 de 2010] [Disponible en: <http://germanlescano.wordpress.com/2010/03/10/patrones->.]

# *Bibliografía Consultada*

## **Bibliografía Consultada**

1. [En Línea] [Consultado el 20 de febrero del 2010]. [Disponible en:  
<http://www.slideshare.net/jent46/diagramas-de-interaccion-presentation-892165>]
2. Clases del análisis. Conferencia #4. Fase de Elaboración. Flujo de trabajo Pruebas. [Disponible  
<http://www.monografias.com/trabajos14/ciencias-social/ciencias-social.shtml>]
3. [En Línea] [Consultado el 20 de febrero del 2010]. [Disponible en:  
<http://msdn.microsoft.com/es-es/library/zw4w595w.aspx>]

# *Glosario de Términos*

## **Glosario de Términos**

**Periféricos:** Se denomina periféricos a los aparatos o dispositivos auxiliares e independientes conectados a la unidad central de procesamiento de una computadora. Se consideran periféricos tanto a las unidades o dispositivos a través de los cuales la computadora se comunica con el mundo exterior, como a los sistemas que almacenan o archivan la información, sirviendo de memoria auxiliar de la memoria principal.

**Ethernet:** Ethernet es un estándar de redes de computadoras de área local con acceso al medio por contienda CSMA/CD. ("Acceso Múltiple por Detección de Portadora con Detección de Colisiones"), es una técnica usada en redes Ethernet para mejorar sus prestaciones. El nombre viene del concepto físico de ether.

**Helicoidal:** El movimiento helicoidal es un movimiento rototraslatorio que resulta de combinar un movimiento de rotación en torno a un eje dado con un movimiento de traslación a lo largo de ese mismo eje; el resultado es un movimiento helicoidal.

**Software:** Se conoce como software al equipamiento lógico o soporte lógico de una computadora digital; comprende el conjunto de los componentes lógicos necesarios que hacen posible la realización de tareas específicas, en contraposición a los componentes físicos, que son llamados hardware.

# Anexos

## Anexo 1: Diagramas de clases del análisis

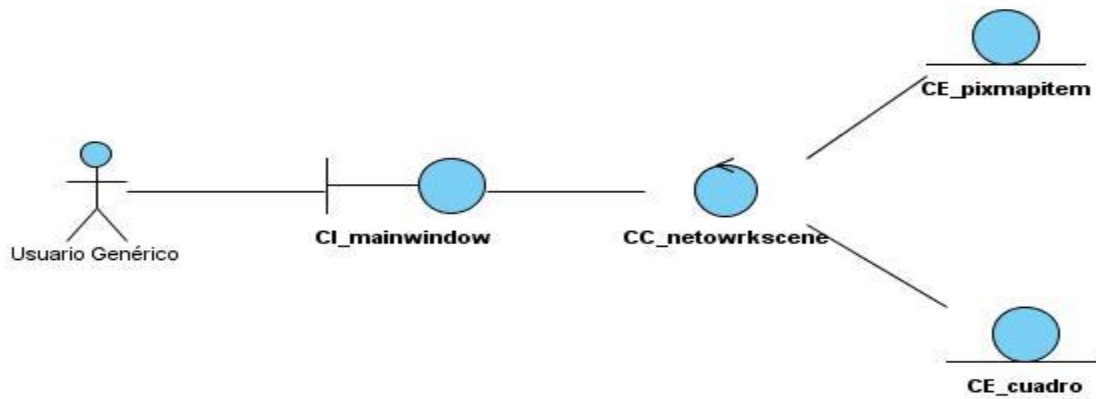


Figura 56: Diagrama de clases del análisis del CU Desagrupar Objetos Gráficos.

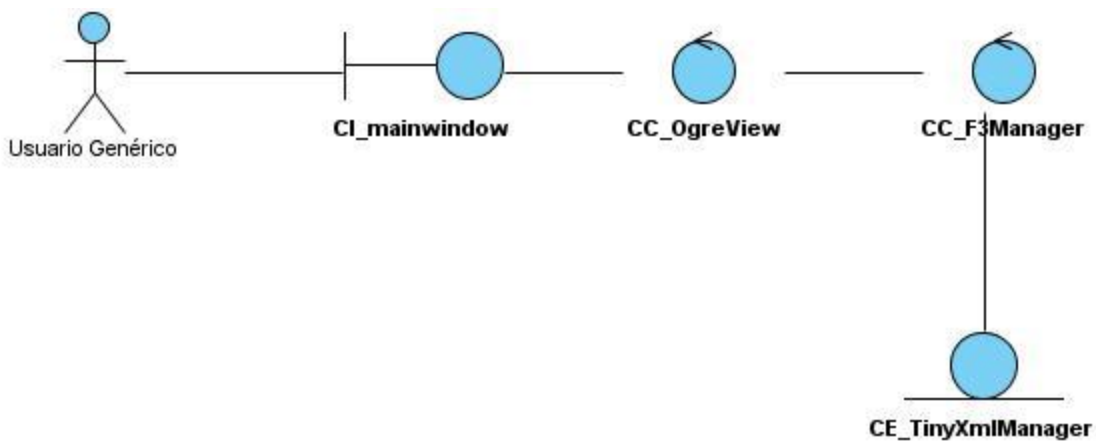


Figura 57: Diagrama de clases del análisis del CU Exportar Escena.

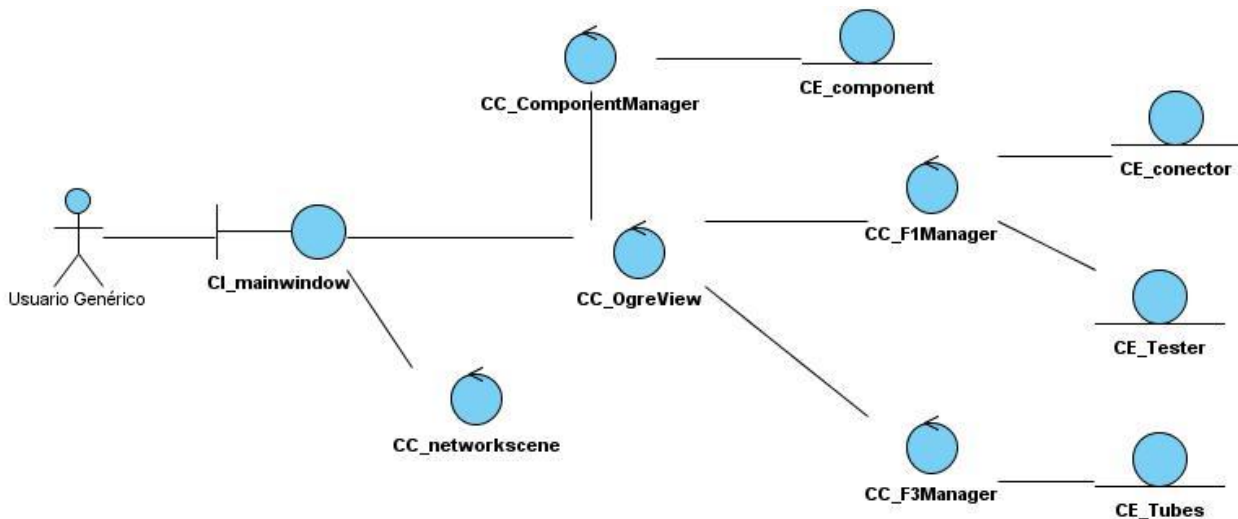




Figura 58: Diagrama de clases del análisis del CU Limpiar Escena.

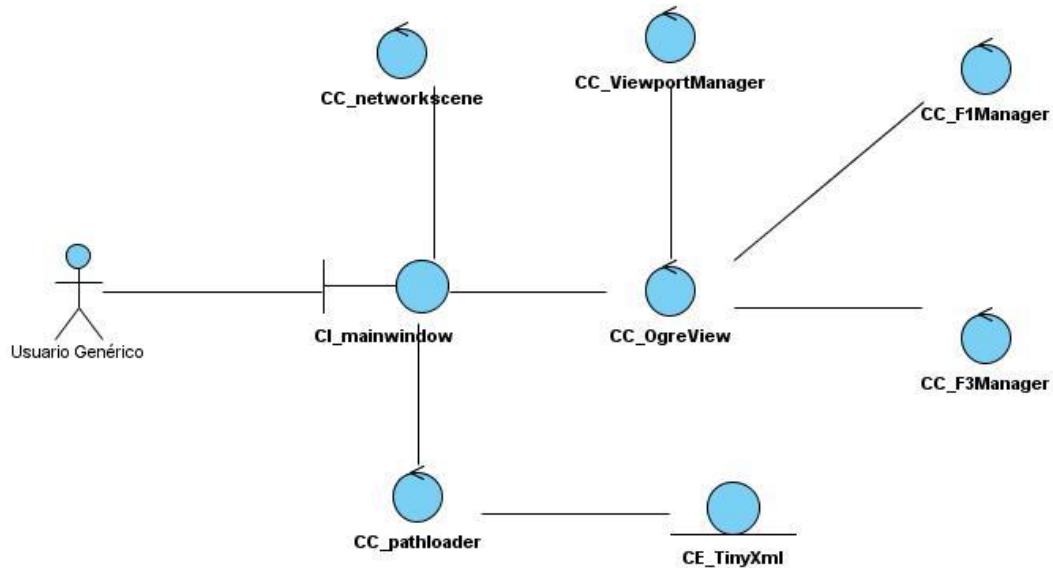


Figura 59: Diagrama de clases del análisis del CU Mostrar Objetos de la Escena.

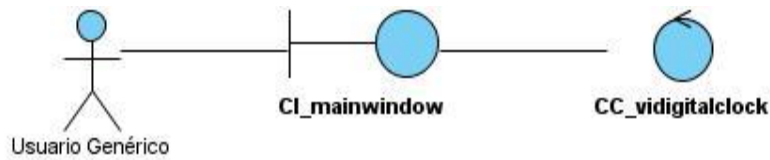


Figura 60: Diagrama de clases del análisis del CU Mostrar tiempo de Duración de la Práctica.

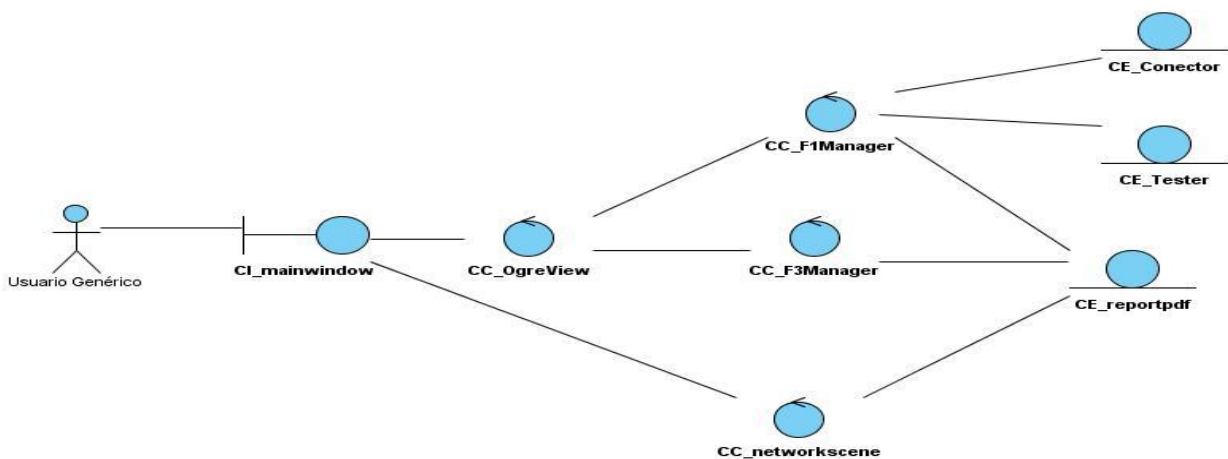


Figura 61: Diagrama de clases del análisis del CU Salvar Archivo.

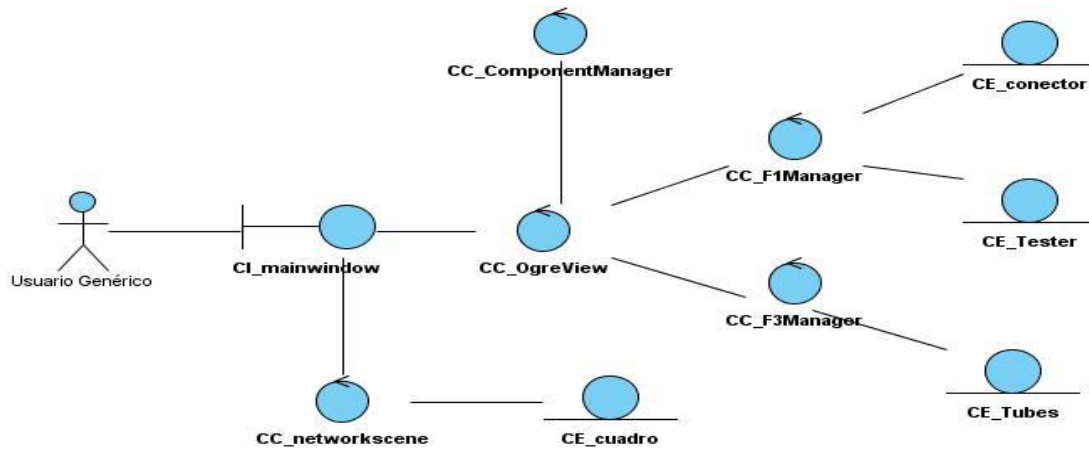


Figura 62: Diagrama de clases del análisis del CU Eliminar Objeto Gráfico.

## Anexo 2: Diagrama de interacción (colaboración) del análisis

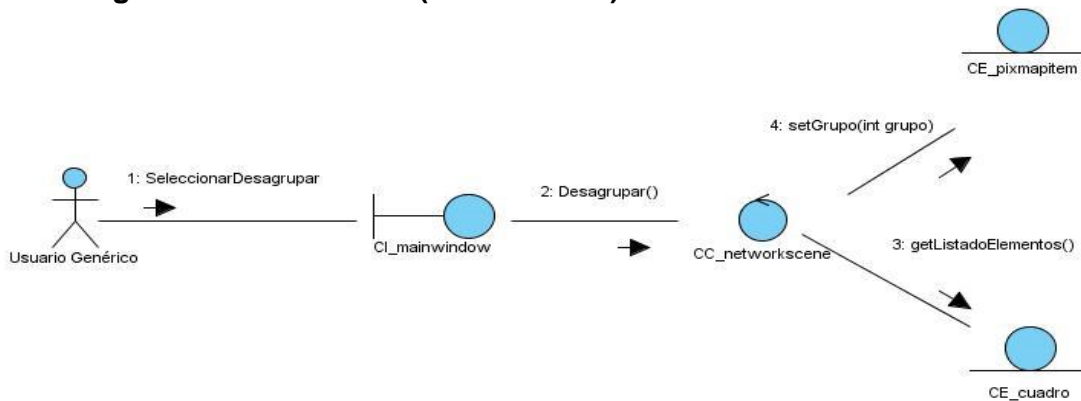


Figura 63: Diagrama de colaboración del CU Desagrupar Objetos Gráficos.

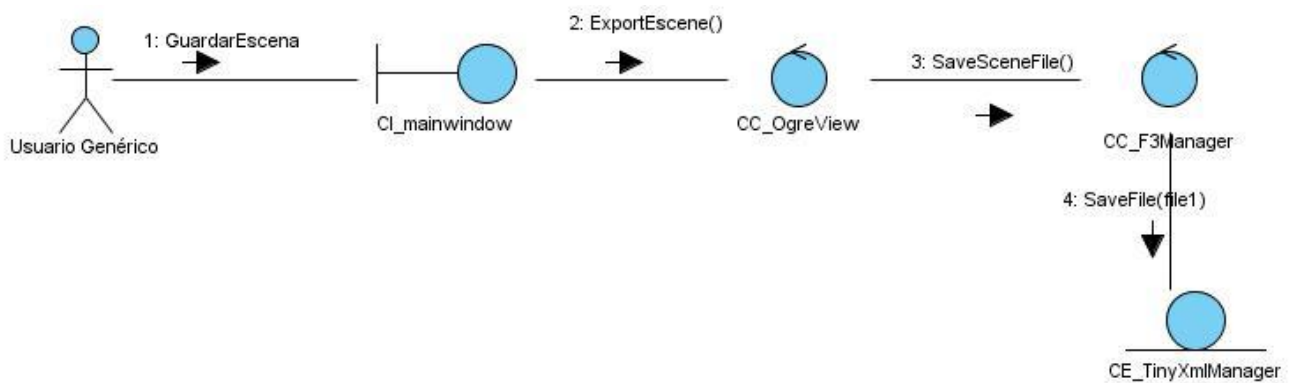
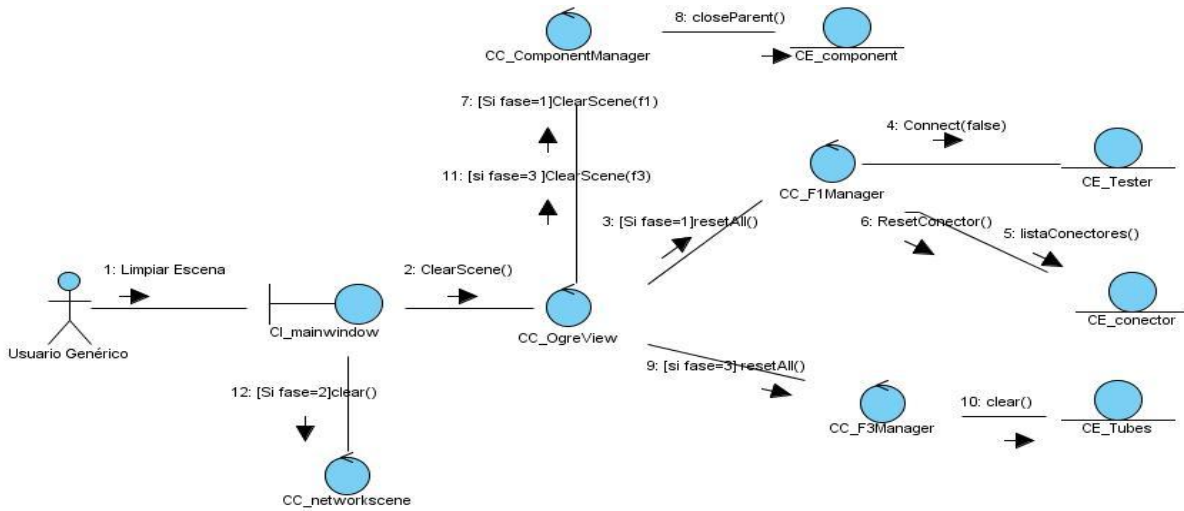
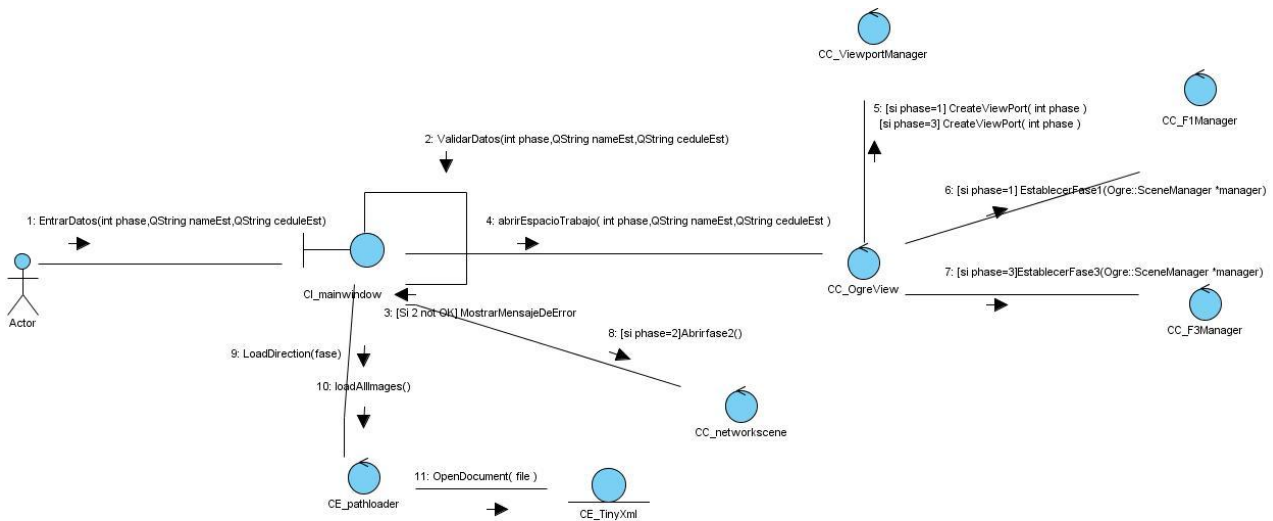


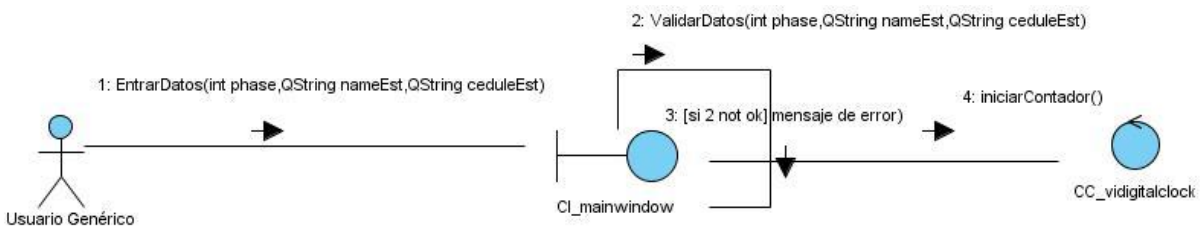
Figura 64: Diagrama de colaboración del CU Exportar Escena.



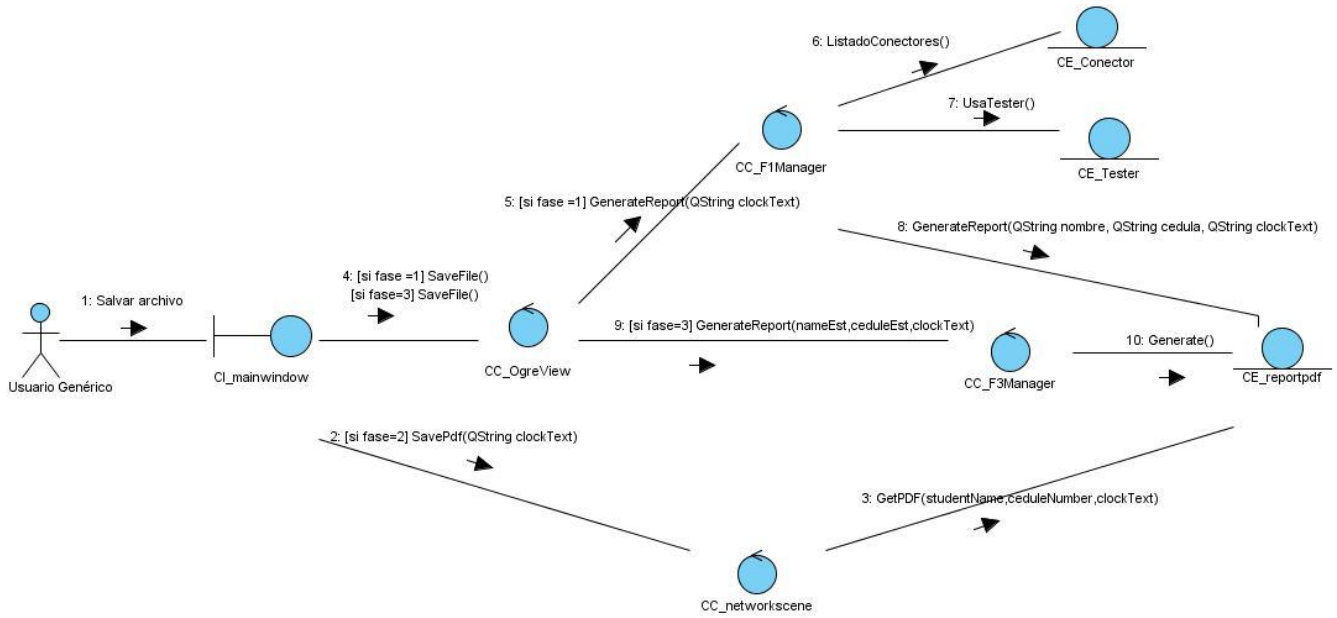
**Figura 65: Diagrama de colaboración del CU Limpiar Escena.**



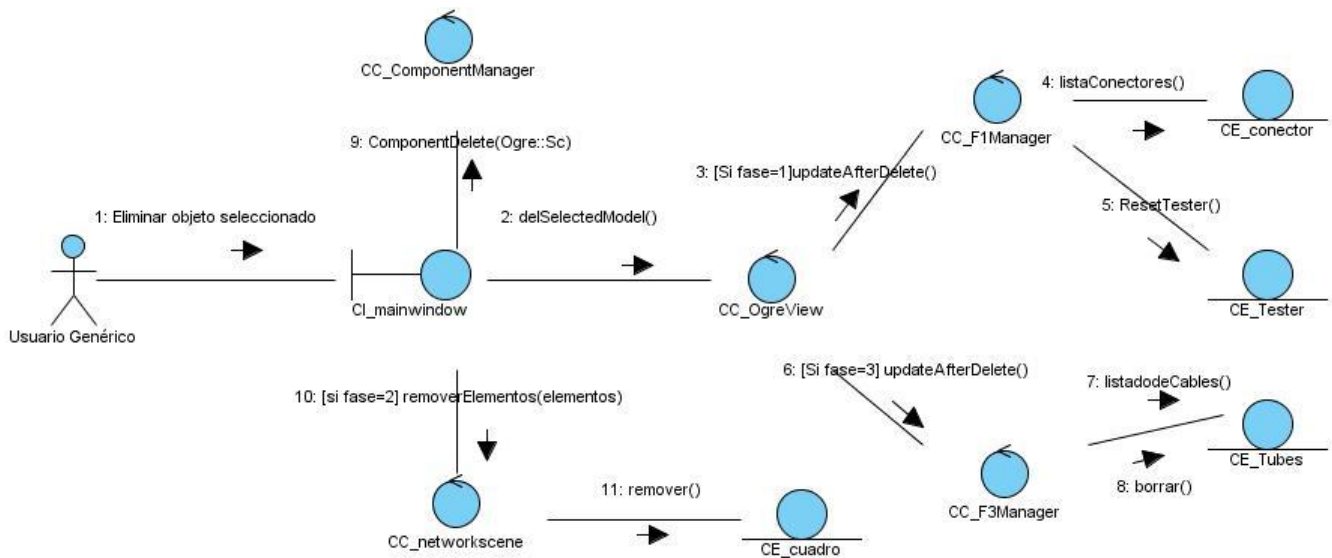
**Figura 66: Diagrama de colaboración del CU Mostrar Objetos de la Escena.**



**Figura 67: Diagrama de colaboración del CU Mostrar Tiempo de Duración de la Práctica.**



**Figura 68: Diagrama de colaboración del CU Salvar Archivo.**



**Figura 69: Diagrama de colaboración del CU Eliminar Objeto Gráfico.**