

Universidad de las Ciencias Informáticas



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

**Título:** Módulo de diseño 2D para el laboratorio Diseño e Instalación de una Red de Área Local.

**Autor:** Hamlet Morales Luis

**Tutor:** Ing. Yaself Machado Tugores

**Co-tutor:** Ing. Gelson Rafael Saurin Ojeda

**Ciudad de la Habana, Cuba**

**Curso 2010-20111**

## DECLARACIÓN DE AUTORÍA

Declaro ser autor de la presente tesis y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

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

Yaself Machado Tugores

---

Firma del Tutor

Gelson Rafael Saurin Ojeda

---

Firma del Co-Tutor

Hamlet Morales Luis

---

Firma del Autor

## DATOS DE CONTACTO

Tutor:

Ing. Yaself Machado Tugores

Universidad de las Ciencias Informáticas, Habana, Cuba.

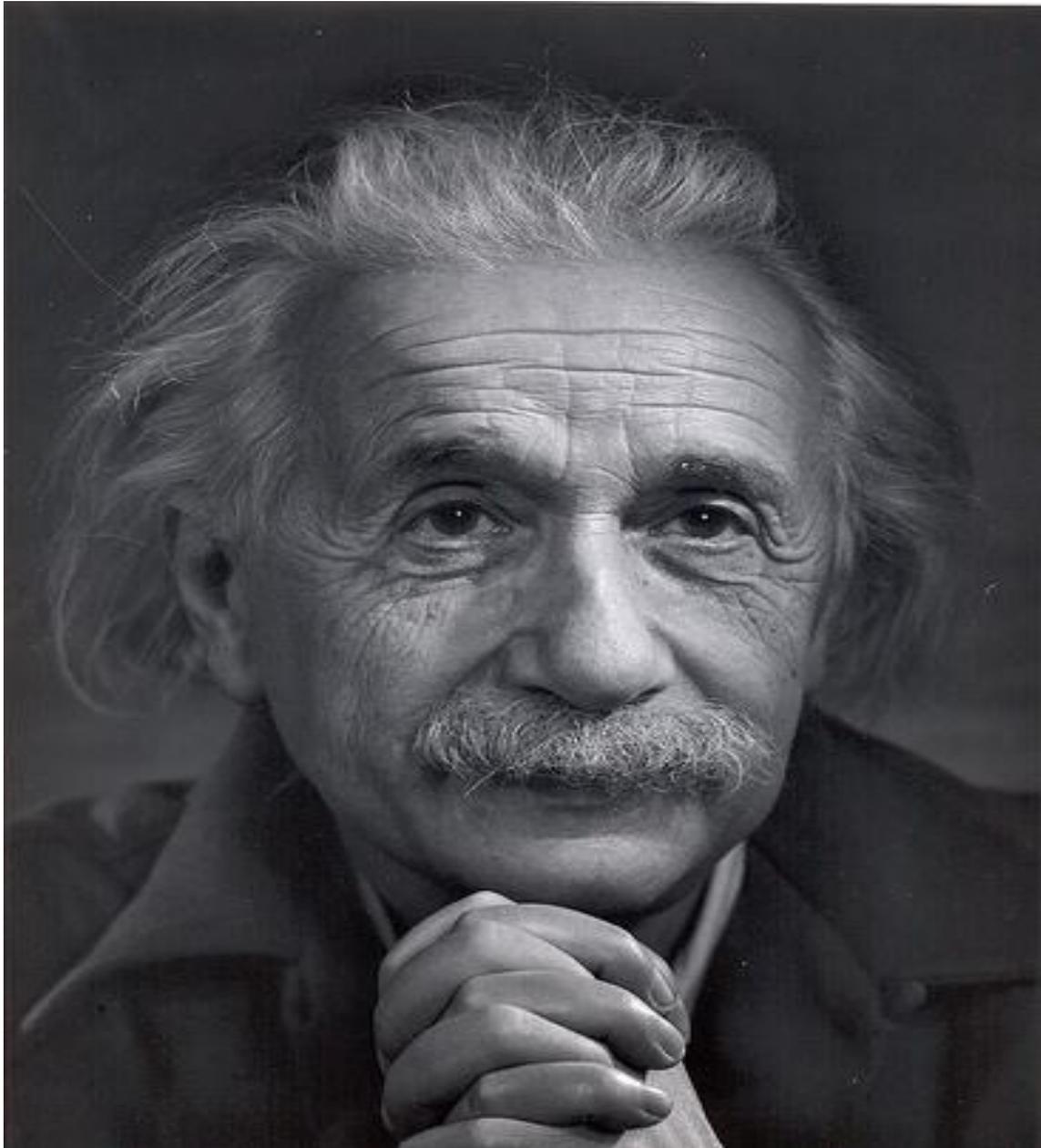
Email: ytugores@uci.cu

Co-Tutor:

Ing. Gelson Rafael Saurin Ojeda

Universidad de las Ciencias Informáticas, Habana, Cuba.

Email: grsaurin @uci.cu



***"En los momentos de crisis sólo la creatividad es más importante que el conocimiento."***

***Albert Einstein***

### AGRADECIMIENTOS

*Son muchas las personas a las que tengo que agradecerles por ser tan importantes en mi vida y les doy las gracias por todo lo que han hecho por mí.*

*Pero antes quiero agradecerle a la Revolución, a Fidel y a la UCI por darme la oportunidad de formarme como profesional...*

*A mis padres por el apoyo y el amor que me han brindado, demostrándome que siempre puedo contar con ellos...*

*A mi hermano por siempre estar presente cuando lo necesito...*

*A mi novia Yusmary por todo el amor y la dedicación que ha sabido entregarme...*

*A mi familia que de una forma u otra han estado pendientes de mis estudios y mi formación como profesional....*

*Al aguerrido colectivo del proyecto Laboratorios Virtuales...*

*A todos mis amigos y compañeros tanto dentro como fuera de la universidad por haber compartido buenos y malos momentos en estos 5 años...*

*A mi tutor Yaself y a mi co-tutor Gelson por su atención y siempre dedicarme un tiempo en su apretada agenda...gracias...*

*A todas aquellas personas que influyeron en mi formación profesional, mil gracias...*

### DEDICATORIA

*A mis padres que siempre han confiado en mí y han hecho todo lo posible por convertirme en un hombre de bien.*

## RESUMEN

Los laboratorios virtuales son medios educativos de gran utilidad, en ellos es posible simular el ambiente de un laboratorio tradicional y desarrollar una práctica de laboratorio sin correr los riesgos ni costear los gastos que conlleva una práctica de laboratorio tradicional.

La enseñanza de temas informáticos como los de redes y conectividad involucra un gasto en material de alto costo en tecnología y hardware para las instituciones académicas. Por lo que en este trabajo se hace la propuesta de desarrollar un módulo que permita crear de manera virtual el diseño de una red de área local.

Este módulo será utilizado en el laboratorio virtual Diseño e Instalación de una Red de Área Local, cuya finalidad es la de brindarle al usuario distintas funcionalidades que le permitan diseñar una red de área local.

A lo largo de este trabajo se expone en detalles cómo fue desarrollado este módulo, cuáles son las funcionalidades que brinda y qué importancia posee el mismo.

## PALABRAS CLAVE:

**Laboratorios virtuales, módulo de diseño 2D, red de área local.**

**ÍNDICE**

INTRODUCCIÓN.....	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.....	5
1.1 Introducción.....	5
1.2Módulo .....	5
1.3 Diseño 2D (2 Dimensiones).....	5
1.4 Módulos de diseño 2D de redes virtuales.....	5
1.5 Laboratorio Virtual.....	6
1.6 Antecedentes de módulos de diseño 2D de redes virtuales. ....	7
1.7 Red de Área local (LAN).....	12
1.8 Topologías de una red de Área local (LAN).....	12
1.8.1 Topología de Bus circular .....	12
1.8.2 Topología de anillo.....	13
1.8.3 Topología Malla .....	14
1.8.4 Topología árbol.....	16
1.8.5 Topología estrella .....	15
1.9 Principales componentes de una red de Área local. ....	17
1.9.1 Puentes .....	17
1.9.2 Computadoras .....	17
1.9.3 Hub.....	18
1.9.4 Repetidor .....	19
1.9.5Router.....	19
1.9.6 Switch.....	20
1.10Metodología del desarrollo del software .....	21
1.10.1 XP (Extreme Programming) .....	21
1.10.2 Scrum .....	23
1.10.3 MSF (Microsoft Solution Framework).....	24

1.10.4 RUP (Rational Unified Procces) .....	25
1.11 Entorno de desarrollo integrado .....	27
1.11.1 Qt Creator .....	27
Principales características de Qt Creator: .....	28
1.11.2 Visual Studio .....	28
1.12 Herramientas Case. ....	29
1.12.1 Visual Paradigm .....	29
1.12.2 Rational Rose .....	30
1.13 Lenguaje Unificado de Modelado (UML).....	31
1.14 Lenguajes de Programación.....	32
1.14.1 C++ .....	32
1.14.2 C# .....	33
1.15 Conclusiones parciales.....	35
<b>CAPÍTULO 2: DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA .....</b>	<b>36</b>
2.1 Introducción.....	36
2.2 Propuesta de las soluciones técnicas.....	36
2.3 Modelo de dominio .....	37
2.3.1 Glosario de términos .....	38
2.4 Requisitos del sistema.....	38
2.4.1 Requisitos Funcionales .....	38
2.4.2 Requisitos No Funcionales .....	39
2.5 Descripción del sistema propuesto.....	40
2.6 Modelo de casos de uso del sistema.....	40
2.6.1 Actores del sistema.....	41
2.6.2 Diagrama de Casos de uso del sistema.....	41
2.7 Descripción de Casos de Uso del Sistema .....	42
2.8 Conclusiones parciales.....	46

---

CAPÍTULO 3: ANÁLISIS Y DISEÑO .....	47
3.1 Introducción.....	47
3.2 Diagramas de Clases de Análisis. ....	47
3.3 Diagramas de Secuencia .....	49
3.4 Diagrama de Clases del Diseño del Sistema.....	51
3.4.1 Diagrama de Clases del Paquete Lógica de Negocio .....	52
3.4.2 Diagrama de Clases del Paquete Presentación .....	53
3.5 Conclusiones parciales.....	54
CAPÍTULO 4: IMPLEMENTACIÓN Y VALIDACIÓN DE LA PROPUESTA .....	55
4.1 Introducción.....	55
4.2 Diagrama de Componentes.....	55
4.3 Diagrama de Componentes del Paquete Lógica de Negocio.....	56
4.4 Diagrama de Componentes del Paquete Presentación .....	57
4.5 Validación.....	57
4.5.1 Validación del Caso de Uso Autenticar .....	57
4.5.2 Validación del Caso de Uso Diseñar red virtual .....	60
4.5.3. Validación del Caso de Uso Genera Reporte.....	62
4.6 Conclusiones parciales.....	65
CONCLUSIONES GENERALES .....	66
RECOMENDACIONES.....	67
REFERENCIAS BIBLIOGRÁFICAS.....	68
ANEXO.....	70

## INTRODUCCIÓN

La realidad virtual es un área de la informática muy utilizada hoy en día, mediante su uso es posible transformar eventos de la realidad en eventos artificiales. Estamos en presencia de realidad virtual cuando contamos con un sistema o un interfaz informático que se encarga de generar entornos sintéticos que suceden en tiempo real, donde se propone una representación virtual de determinados objetos y situaciones del mundo real, esto a través de medios electrónicos, como la computadora [\(1\)](#).

A su vez el tema de Laboratorios Virtuales es un nuevo paradigma dentro de la realidad virtual que tiene como objetivo la integración de modelos, que permiten interacción, que sólo es posible por la tecnología que la soporta, facilitando con ello acceder libremente y sin intermediación a los materiales originales.

Los laboratorios virtuales son sistemas computacionales que pretenden aproximar el ambiente de un laboratorio tradicional. Los experimentos se realizan paso a paso, siguiendo un procedimiento similar al de un laboratorio tradicional ya que se visualizan instrumentos y fenómenos mediante objetos dinámicos, imágenes o animaciones. Se obtienen resultados numéricos y gráficos, tratándose estos matemáticamente para la obtención de los objetivos perseguidos en la planificación docente de las asignaturas.

Cuba a pesar de ser un país subdesarrollado y bloqueado ha dado pasos significativos en la creación de laboratorios virtuales, ejemplo de esto lo encontramos en la facultad 5 de la Universidad de las Ciencias Informáticas (UCI), en el proyecto Laboratorios Virtuales donde se pone de manifiesto toda esta opción válida en el campo de la educación. En este proyecto se crean laboratorios virtuales para suplir la necesidad de medios didácticos interactivos en los institutos educativos, en el cual el estudiante interactúa con el entorno virtual para realizar una serie de ejercicios de laboratorios.

Actualmente el proyecto tiene el compromiso de desarrollar 3 laboratorios virtuales, uno de ellos llamado Diseño e Instalación de una Red de Área Local, el cual actuará como material de apoyo en la asignatura de redes y consiste en diseñar una red de área local de manera virtual. En este laboratorio el usuario debe poder manipular los diversos componentes de una red de área local y realizar el enlace entre estos mediante el cableado correspondiente. Para el desarrollo del laboratorio virtual Diseño e Instalación de una Red de Área Local, el proyecto necesita contar con un módulo robusto y eficiente que permita en un ambiente virtual el diseño de una red de área local. En estos momentos el proyecto no cuenta con un módulo capaz de permitirle al usuario diseñar una red de área local de manera virtual.

En la actualidad existen módulos desarrollados con las características deseadas pero son software privativo y una de las exigencias del cliente es que no se utilice este tipo de software, pues esto conllevaría a un pago extra a las empresas desarrolladoras del software.

Teniendo en cuenta lo anteriormente planteado surge la **Situación Problemática**: Que actualmente debido a que no se cuenta con un módulo que permita el diseño de una red de área local en un entorno virtual, no se puede llevar a cabo la práctica del laboratorio Diseño e Instalación de una Red de Área Local, lo que trae como consecuencia que no se pueda cumplir el compromiso con el cliente, afectando la imagen del proyecto Laboratorios Virtuales.

Dada la situación problemática expuesta anteriormente surge el **Problema Científico**: ¿Cómo eliminar las limitaciones que imposibilitan la realización de la práctica del laboratorio Diseño e Instalación de una Red de Área Local?, problema que guía la realización del trabajo de diploma.

Es definido en esta investigación como **Objeto de Estudio**: Módulos de diseño de una red virtual, derivándose de este el **Objetivo general**: Desarrollar un módulo de diseño 2D para el laboratorio Diseño e Instalación de una Red de Área Local, derivándose el **Campo de Acción**: Módulos de diseño 2D de una red de área local virtual.

**La idea a defender** de esta investigación es: Con la aplicación del módulo desarrollado será posible diseñar una red de área local de modo virtual para el proyecto Diseño e Instalación de una Red de Área Local.

Para lograr los objetivos planteados es necesario realizar algunas tareas investigativas que son definidas a continuación:

1. Elaboración del marco teórico a través del estudio del estado del arte sobre el tema en la actualidad.
2. Selección de las herramientas, lenguajes y tecnologías a utilizar para el desarrollo del módulo.
3. Análisis y diseño del módulo.
4. Elaboración de la propuesta de solución.
5. Implementación de la propuesta de solución.
6. Validación de la propuesta de solución.

Además para todo el proceso de investigación y elaboración de este trabajo se tomará en cuenta la utilización de varios métodos científicos de investigación.

## Teóricos:

- **Histórico – Lógico:** Método teórico mediante el cual se constatará como ha sido la trayectoria histórica real, la evolución y desarrollo de los aspectos principales de los módulos de diseño 2D de redes virtuales.
- **Analítico – Sintético:** Este método teórico será utilizado en la investigación para buscar la esencia de lo que existe en el mundo acerca de los módulos de diseño 2D de redes virtuales para aplicaciones en laboratorios virtuales, los rasgos que los caracterizan y los distinguen.

## Empíricos:

- **Consulta de fuentes de información:** Método que permite tener acceso a la información disponible sobre el tema.
- **Entrevista:** Método empírico para obtener información amplia, abierta y directa de forma oral durante una conversación planificada sobre los módulos de diseño 2D de redes virtuales, Laboratorios virtuales, aplicaciones extensibles, extensiones para aplicaciones, que sirva para guiar el resto de la investigación.
- **Observación:** Método empírico que permite conocer la realidad mediante la percepción directa de los objetos y fenómenos, es una manera de acceder a la información directa e inmediata sobre el proceso, fenómeno u objeto que está siendo investigado.

El trabajo de diploma está estructurado en cuatro capítulos, recogiendo en ellos todo lo referente al trabajo que se realizó:

### **Capítulo 1:** Fundamentación Teórica

Muestra un panorama sobre el diseño 2D de una red de área local, presentando una minuciosa revisión del estado del arte sobre el desarrollo de módulos 2D para el diseño de una red de área local teniendo en cuenta las herramientas, el lenguaje y las tecnologías a utilizar para el desarrollo de estos módulos.

### **Capítulo 2:** Descripción de la propuesta de solución.

Se describe una propuesta para el desarrollo del módulo teniendo en cuenta la selección de las herramientas y los lenguajes, además se mencionan los requisitos y los casos de uso del sistema.

## **Capítulo 3** Análisis y diseño

Se muestran los diagramas de clases del análisis, los diagramas de secuencia correspondientes a los casos de uso y los diagramas de clases del diseño que aparecen en diagramas de paquetes.

## **Capítulo 4:** Implementación y Validación de la propuesta

Se lleva a cabo la implementación del sistema y se aplican los procedimientos para validar la propuesta del desarrollo del módulo de diseño 2D para el laboratorio Diseño e Instalación de una Red de Área Local, mostrando los resultados de estas validaciones.

## CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

### 1.1 Introducción

En el presente capítulo se abordará la parte teórica del trabajo de diploma, pues se elabora el marco teórico a partir del estudio del estado del arte de los módulos de diseño 2D de redes virtuales, esta tarea tiene la finalidad de recopilar la mayor información posible de estos módulos. Se presentará además ejemplos de la utilización de los módulos anteriormente mencionados en Cuba y el resto del mundo.

Con el fin de crear un mayor entendimiento del problema se abordarán algunos conceptos básicos, entre los que se encuentran: laboratorios virtuales, módulo, diseño 2D y módulos de diseño 2D. Teniendo en cuenta que en el módulo a desarrollar se va a diseñar una red de área local, es de interés para la investigación contar con la definición de la misma además conocer acerca de los componentes y las topologías que la conforman

También en este capítulo se exponen una selección de las posibles herramientas y lenguajes para el desarrollo del módulo. Esto posibilita realizar un estudio para la posterior selección teniendo en cuenta las características del módulo y las exigencias del cliente.

### 1.2 Módulo

Un módulo es un componente auto controlado de un sistema, el cual posee una interfaz bien definida hacia otros componentes; algo es modular si es construido de manera tal que se facilite su ensamblaje y acomodamiento flexible.

### 1.3 Diseño 2D (2 Dimensiones) [\(2\)](#)

El diseño 2D es un conjunto de objetos de dibujos que contienen solamente la geometría en dos dimensiones, es decir todos son objetos planos cuentan con largo y ancho pero carecen de profundidad, los colores son planos y no existe perspectiva de ningún tipo.

### 1.4 Módulos de diseño 2D de redes virtuales [\(2\)](#)

Los módulos de diseño 2D de redes virtuales son herramientas de simulación que permiten el diseño, la construcción, y la prueba de una red de comunicaciones en un ambiente simulado, son un espacio electrónico de trabajo cuyo principal objetivo es el de brindar las opciones al usuario para que sea capaz de diseñar una red de manera virtual, estos cuentan con un área que abarca la mayor parte de la interfaz gráfica de la aplicación y es en este espacio donde se sitúan los diferentes componentes 2D

que forman el diseño de la red. En otras áreas de menor dimensión aparecen los componentes de la red de área local junto al cableado para unir a los mismos.

## 1.5 Laboratorio Virtual

El laboratorio virtual es definido como “un espacio electrónico de trabajo concebido para la colaboración y la experimentación a distancia con objeto de investigar o realizar otras actividades creativas, y elaborar y difundir resultados mediante tecnologías difundidas de información y comunicación. Un laboratorio virtual es diferente de un “laboratorio verdadero” o de un “laboratorio tradicional”. Sin embargo, no se considera que el laboratorio virtual vaya a suplantar a los verdaderos laboratorios o competir con ellos. En cambio, los laboratorios virtuales constituyen una posible extensión de los verdaderos laboratorios y abren nuevas perspectivas que no se podían explorar completamente, dentro de un verdadero laboratorio, a un costo asequible. El concepto de laboratorio virtual está implícito en otras nociones tales como las de "colaboratorio", "grupo de trabajo virtual", "empresa virtual", "grupo interinstitucional" y "grupo de colaboración a distancia". En un sentido más amplio, el laboratorio virtual es un tipo de colaboración centrada en el logro de determinados objetivos creativos o de ayuda a la toma de decisiones. Por lo tanto, un laboratorio virtual puede dedicarse prácticamente a todas las esferas de la actividad intelectual humana. [\(3\)](#)

### Ventajas [\(4\)](#)

1. Acerca y facilita a un mayor número de alumnos la realización de experiencias, aunque alumno y laboratorio no coincidan en el espacio. El estudiante accede a los equipos del laboratorio a través de una interfaz, pudiendo experimentar sin riesgo alguno.
2. Reducen el coste del montaje y mantenimiento de los laboratorios tradicionales, siendo una alternativa barata y eficiente, donde el estudiante simula los fenómenos a estudiar como si los observase en el laboratorio tradicional.
3. Los estudiantes aprenden mediante prueba y error, sin miedo a sufrir o provocar un accidente.

### Desventajas [\(4\)](#)

- 1 El laboratorio virtual no puede sustituir la experiencia práctica altamente enriquecedora del laboratorio tradicional.
- 2 En el laboratorio virtual se corre el riesgo de que el alumno se comporte como un mero espectador. Es importante que las actividades en el laboratorio virtual, vengán acompañadas de un guión que explique el concepto a estudiar, así como las ecuaciones del modelo utilizado. Es necesario que el estudiante realice una actividad ordenada y progresiva, conducente a

alcanzar objetivos básicos concretos.

- 3 El alumno no utiliza elementos reales en el laboratorio virtual, lo que provoca una pérdida parcial de la visión de la realidad. Además, no siempre se dispone de la simulación adecuada para el tema que el profesor desea trabajar.

## 1.6 Antecedentes de módulos de diseño 2D de redes virtuales.

En este epígrafe se hace el estudio de los principales módulos de diseño de redes virtuales que existen en el mundo, con el fin de conocer sus principales características tanto en la interfaz de usuario como en las funcionalidades que brindan.

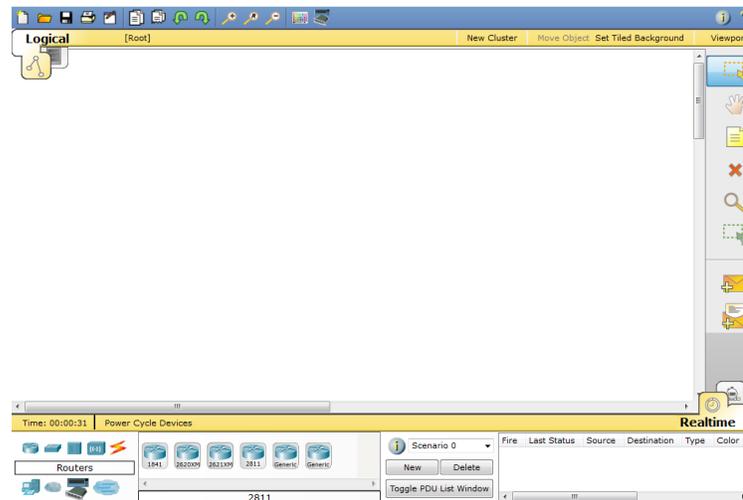
### 1. PACKET TRACER™ [\(5\)](#)

Es un simulador gráfico de redes desarrollado y utilizado por Cisco empresa multinacional con sede en California, Estados Unidos especializada en la construcción y venta de equipos de telecomunicaciones.

Este simulador es un simulador de entorno de redes de comunicaciones de fidelidad media, que permite crear topologías de red mediante la selección de los dispositivos y su respectiva ubicación en un área de trabajo, utilizando una interfaz gráfica.

#### **Características generales.**

Packet Tracer es un simulador que permite realizar el diseño de topologías, la configuración de dispositivos de red, así como la detección y corrección de errores en sistemas de comunicaciones. Ofrece como ventaja adicional el análisis de cada proceso que se ejecuta en el programa de acuerdo a la capa de modelo OSI que interviene en dicho proceso; razón por la cuál es una herramienta de gran ayuda en el estudio y aprendizaje del funcionamiento y configuración de redes de comunicaciones.



**Figura 1.1 Ventana Principal del Packet Tracer**

## Ventajas

- El enfoque pedagógico de este simulador, hace que sea una herramienta muy útil como complemento de los fundamentos teóricos sobre redes de comunicaciones.
- El programa posee una interfaz de usuario muy fácil de manejar, e incluye documentación y tutoriales sobre el manejo del mismo.

## Desventajas

- Es un software propietario, y por ende se debe pagar una licencia para instalarlo.
- Solo permite modelar redes en términos de filtrado y retransmisión de paquetes.
- No permite crear topologías de red que involucren la implementación de tecnologías diferentes a Ethernet.

## 2. FLAN (F- Links And Nodes) [\(5\)](#)

Es un software desarrollado con el lenguaje de programación Java y se distribuye con licencia pública GNU. Se considera que pertenece al grupo de los simuladores de propósito general, ya que por medio de Java se pueden crear y configurar nuevos dispositivos, aplicaciones o protocolos de red, aun si no están incluidos dentro de las librerías del programa, inclusive se pueden realizar modificaciones al código fuente de FLAN.

## Características generales

FLAN es una herramienta de simulación que permite el diseño, la construcción y la prueba de una red de comunicaciones en un ambiente simulado. El programa hace el análisis de las redes asociando su estructura basada en nodos y enlaces, con bloques simples, por medio de los cuales se puede entender el funcionamiento especialmente de los protocolos de enrutamiento que maneja la capa de red.

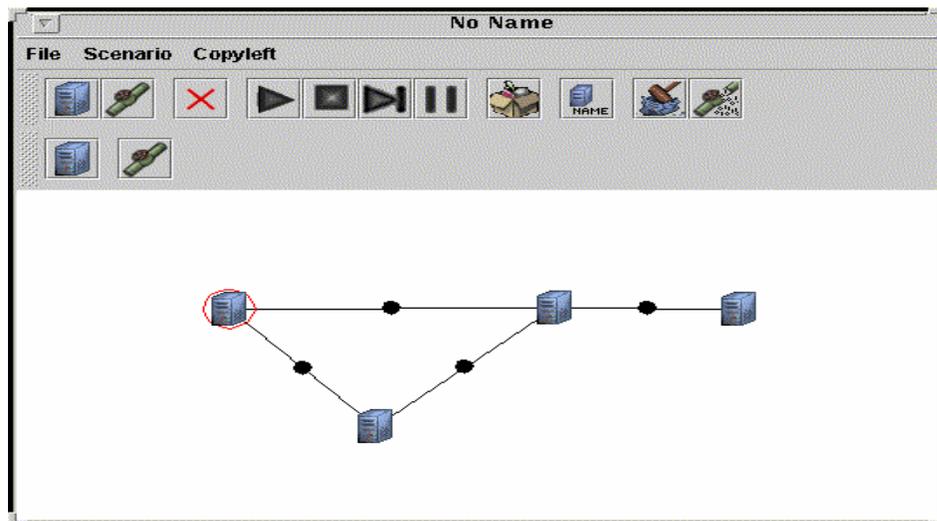


Figura 1.2 Ventana principal de FLAN

## Ventajas

- Este es un software multiplataforma, es decir que puede ser implementado sobre cualquier sistema operativo que soporte la máquina virtual de Java.
- El programa contiene además, unas herramientas llamadas manejadores, que son protocolos específicos que ayudan a determinar cómo es recibida la información, cómo procesarla y además cómo dirigir la simulación.

## Desventajas

- FLAN está diseñado para la prueba de protocolos en redes pequeñas, es decir, que tengan menos de 100 nodos. Aunque el usuario puede tener tantos nodos como desee, el funcionamiento se verá afectado mientras se agreguen más y más nodos.

- El programa permite que se trabaje con la interfaz gráfica, sin embargo es necesario tener conocimientos básicos sobre el lenguaje de programación Java, para poder hacer más configuraciones con el software.

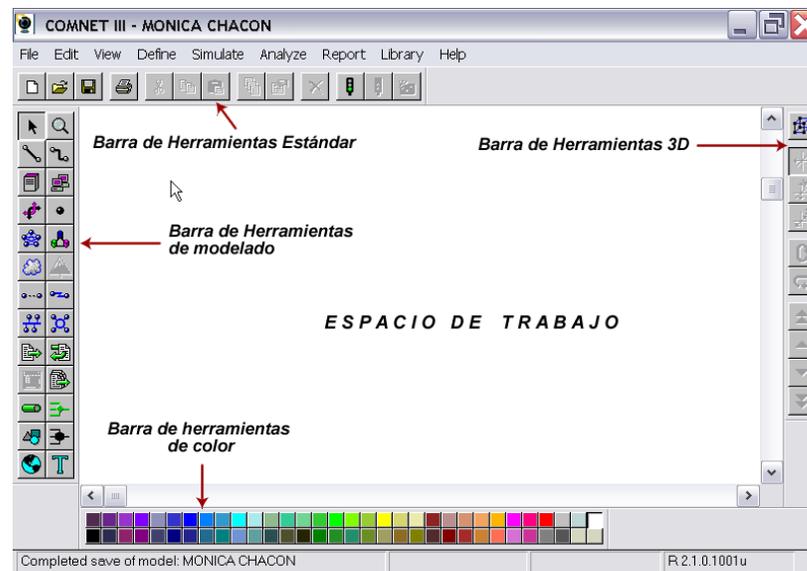
### 3. COMNET III™[\(5\)](#)

COMNET III es una herramienta comercial orientada al diseño, configuración y estudio de las redes de comunicaciones, haciendo uso del lenguaje de programación MODSIM II. Desarrollado por CACI Products Inc empresa situada en California, Estados Unidos que se especializa en desarrollar y dar mantenimientos a software de comunicaciones.

Por medio de este programa es posible crear topologías de redes complejas, configurar varias tecnologías, protocolos y dispositivos de red, para hacer un análisis detallado del funcionamiento y del rendimiento de redes tipo LAN, MAN y WAN, utilizando una interfaz gráfica en un ambiente de ventanas.

#### **Características generales**

Este software gráfico permite analizar y predecir el funcionamiento de redes informáticas, desde topologías básicas de interconexión hasta esquemas mucho más complejos de simulación con múltiples redes interconectadas con diversos protocolos y tecnologías como Ethernet, ATM, Satelitales, Frame Relay, X25, etc. Dentro del área de trabajo del programa, se hace la descripción gráfica del modelo de red, se asocian las fuentes generadoras de tráfico en la red, se configuran los parámetros y las características de los dispositivos de acuerdo a la aplicación que se desea implementar; luego se pone en marcha la simulación y finalmente, se analizan los resultados estadísticos sobre el desempeño de la red, los cuales son programados antes de iniciar la simulación y que se generan automáticamente cuando se concluye la simulación. Algunos de los parámetros que se pueden incluir dentro de los informes de la red esta: la ocupación de enlaces o nodos, la cantidad de mensajes generados, las colisiones, entre otros.



**Figura 1.3 Ventana principal de COMNET III**

## Ventajas

- El programa ofrece la posibilidad de simular una gran cantidad de protocolos y tecnologías de red, y ofrece la posibilidad de crear protocolos a medida que se van necesitando.
- Permite configurar y observar una gran cantidad de parámetros durante la simulación como: colisiones, capacidad de los buffers de entrada y salida de los dispositivos, utilización del canal, anchos de banda, etc.

## Desventajas

- Es un software propietario.
- Por ser una de las herramientas de simulación más completas del mercado, la programación de los parámetros de los dispositivos y enlaces de la red tiende a ser compleja.
- Además de los conocimientos sobre el manejo y el diseño de redes de comunicaciones, se requieren conocimientos en otras áreas como por ejemplo la estadística.

En Cuba debido al nivel informático alcanzado se han desarrollado varios laboratorios virtuales principalmente en asignaturas básicas como física y química. En cuanto a laboratorios virtuales destinados a diseñar redes de manera virtual no aparecen precedentes registrados lo que le da mayor importancia al módulo a desarrollar.

## 1.7 Red de área local (LAN) (6)

LAN es la abreviatura de Network Area Local (red de área local o simplemente red local). Una red local es la interconexión de varios ordenadores y periféricos para intercambiar recursos e información. En definitiva que permite que dos o más máquinas se comuniquen.

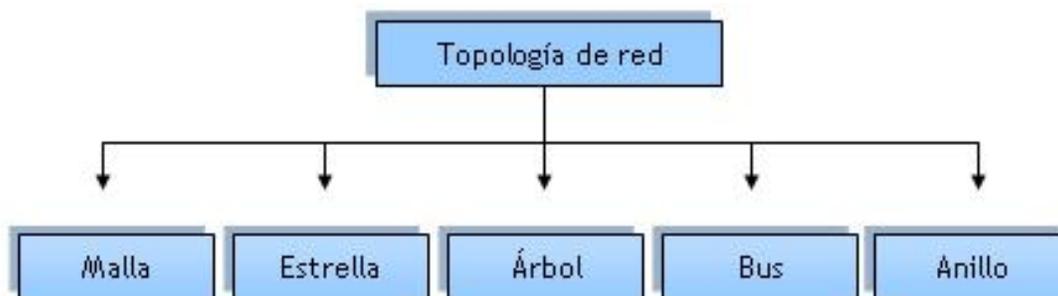
El término red local incluye tanto el hardware como el software necesario para la interconexión de los distintos dispositivos y el tratamiento de la información.

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.

Todos los dispositivos pueden comunicarse con el resto aunque también puede funcionar de forma independiente. Las velocidades de comunicación son elevadas estando en el orden de varios millones de bits por segundo dependiendo del tipo de red que se use. Es un sistema fiable ya que se dispone de sistemas de detección y corrección de errores de transmisión.

## 1.8 Topologías de una red de área local (LAN)

Las topologías describen la red físicamente y también brindan información acerca del método de acceso que se usa. (7)



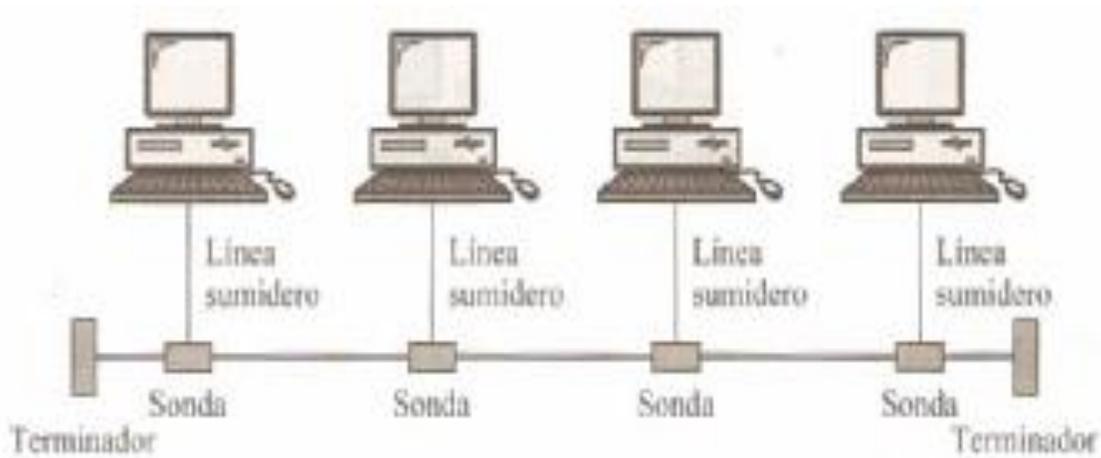
**Figura 1.4 Topología de la red.**

### 1.8.1 Topología de Bus Circular

Una topología de bus es multipunto. Un cable largo actúa como una red troncal que conecta todos los dispositivos en la red.

Los nodos se conectan al bus mediante cables de conexión (latiguillos) y sondas. Un cable de conexión es una conexión que va desde el dispositivo al cable principal. Una sonda es un conector que, o bien se conecta al cable principal, o se pincha en el cable para crear un contacto con el núcleo metálico.

Entre las ventajas de la topología de bus se incluye la sencillez de instalación. El cable troncal puede tenderse por el camino más eficiente y, después, los nodos se pueden conectar al mismo mediante líneas de conexión de longitud variable. De esta forma se puede conseguir que un bus use menos cable que una malla, una estrella o una topología en árbol. (7)



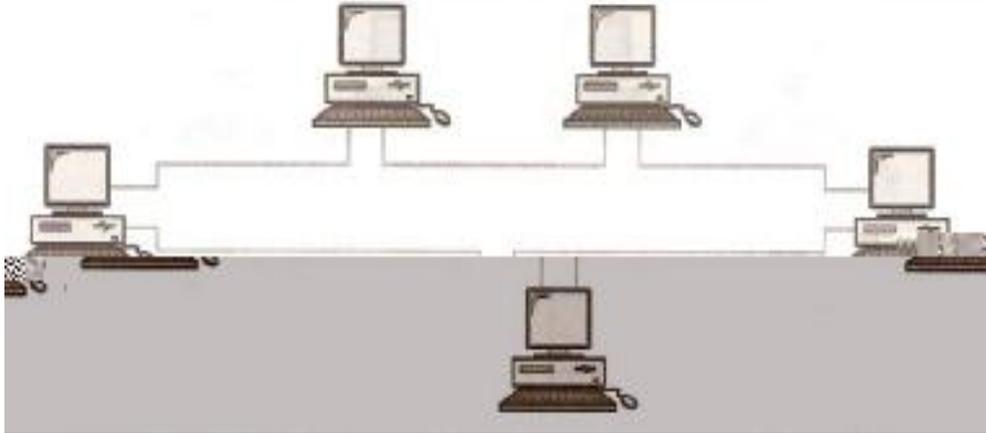
**Figura 1.5 Topología Bus Circular**

## 1.8.2 Topología de Anillo (7)

En una topología en anillo cada dispositivo tiene una línea de conexión dedicada y punto a punto solamente con los dos dispositivos que están a sus lados. La señal pasa a lo largo del anillo en una dirección, o de dispositivo a dispositivo, hasta que alcanza su destino. Cada dispositivo del anillo incorpora un repetidor.

Un anillo es relativamente fácil de instalar y reconfigurar. Cada dispositivo está enlazado solamente a sus vecinos inmediatos (bien físicos o lógicos). Para añadir o quitar dispositivos, solamente hay que mover dos conexiones.

Las únicas restricciones están relacionadas con aspectos del medio físico y el tráfico (máxima longitud del anillo y número de dispositivos). Además, los fallos se pueden aislar de forma sencilla. Generalmente, en un anillo hay una señal en circulación continuamente.



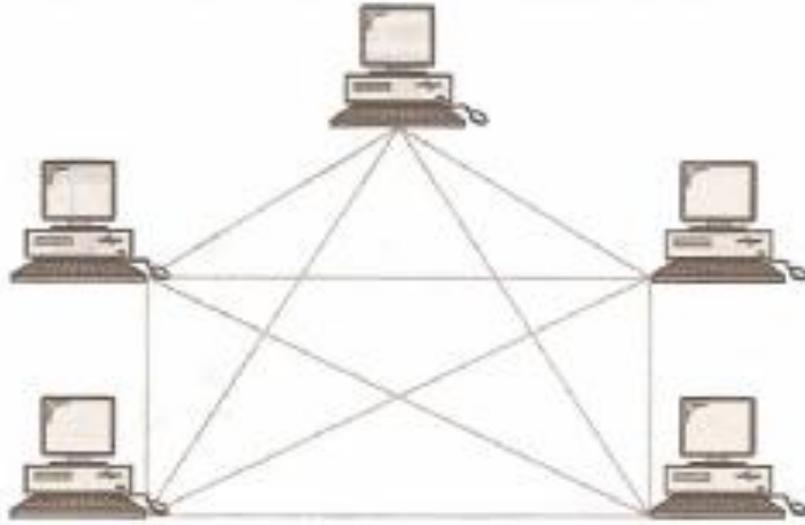
**Figura 1.6 Topología de Anillo**

### 1.8.3 Topología Malla [\(7\)](#)

En una topología en malla, cada dispositivo tiene un enlace punto a punto y dedicado con cualquier otro dispositivo. El término dedicado significa que el enlace conduce el tráfico únicamente entre los dos dispositivos que conecta.

Por tanto, una red en malla completamente conectada necesita  $n(n-1)/2$  canales físicos para enlazar  $n$  dispositivos. Para acomodar tantos enlaces, cada dispositivo de la red debe tener sus puertos de entrada/salida (E/S).

Una malla ofrece varias ventajas sobre otras topologías de red. En primer lugar, el uso de los enlaces dedicados garantiza que cada conexión sólo debe transportar la carga de datos propia de los dispositivos conectados, eliminando el problema que surge cuando los enlaces son compartidos por varios dispositivos. En segundo lugar, una topología en malla es robusta. Si un enlace falla, no inhabilita todo el sistema. Otra ventaja es la privacidad o la seguridad. Cuando un mensaje viaja a través de una línea dedicada, solamente lo ve el receptor adecuado. Las fronteras físicas evitan que otros usuarios puedan tener acceso a los mensajes.



**Figura 1.7 Topología Malla**

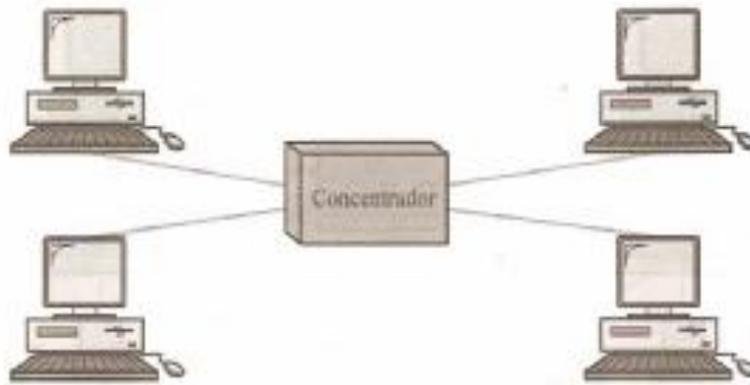
### 1.8.4 Topología Estrella

En la topología en estrella cada dispositivo solamente tiene un enlace punto a punto dedicado con el controlador central, habitualmente llamado concentrador. Los dispositivos no están directamente enlazados entre sí. (7)

A diferencia de la topología en malla, la topología en estrella no permite el tráfico directo de dispositivos. El controlador actúa como un intercambiador: si un dispositivo quiere enviar datos a otro, envía los datos al controlador, que los retransmite al dispositivo final.

Una topología en estrella es más barata que una topología en malla. En una red de estrella, cada dispositivo necesita solamente un enlace y un puerto de entrada/salida para conectarse a cualquier número de dispositivos.

Este factor hace que también sea más fácil de instalar y reconfigurar. Además, es necesario instalar menos cables, y la conexión, desconexión y traslado de dispositivos afecta solamente a una conexión: la que existe entre el dispositivo y el concentrador. (7)

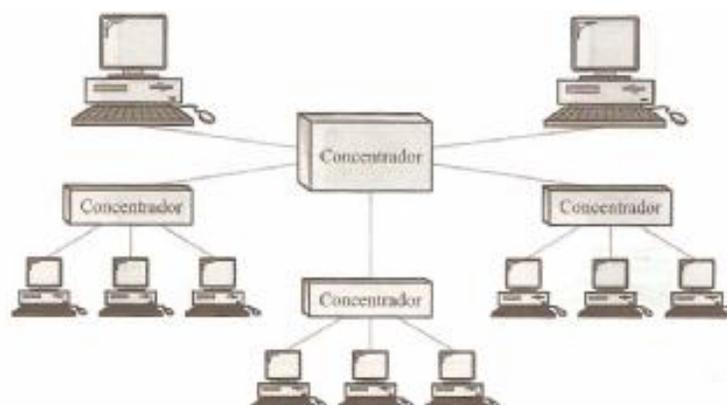


**Figura 1.8 Topología Estrella**

### 1.8.5 Topología Árbol (7)

La topología en árbol es una variante de la de estrella. Como en la estrella, los nodos del árbol están conectados a un concentrador central que controla el tráfico de la red. Sin embargo, no todos los dispositivos se conectan directamente al concentrador central. La mayoría de los dispositivos se conectan a un concentrador secundario que, a su vez, se conecta al concentrador central.

El controlador central del árbol es un concentrador activo. Un concentrador activo contiene un repetidor, es decir, un dispositivo hardware que regenera los patrones de bits recibidos antes de retransmitidos. Retransmitir las señales de esta forma amplifica su potencia e incrementa la distancia a la que puede viajar la señal. Los concentradores secundarios pueden ser activos o pasivos. Un concentrador pasivo proporciona solamente una conexión física entre los dispositivos conectados.



**Figura 1.9 Topología Árbol**

## 1.9 Principales componentes de una red de Área local.

### 1.9.1 Puentes

Un puente es un dispositivo de interconexión de redes de ordenadores que opera en la capa 2 (nivel de enlace de datos) del modelo OSI. Este interconecta dos segmentos de red (o divide una red en segmentos) haciendo el pasaje de datos de una red hacia otra, con base en la dirección física de destino de cada paquete. Un bridge conecta dos segmentos de red como una sola red usando el mismo protocolo de establecimiento de red.

Funciona a través de una tabla de direcciones MAC detectadas en cada segmento a que está conectado. Cuando detecta que un nodo de uno de los segmentos está intentando transmitir datos a un nodo del otro, el bridge copia la trama para la otra subred. Por utilizar este mecanismo de aprendizaje automático, los bridges no necesitan configuración manual. [\(8\)](#)



**Figura 1.10 Puente**

### 1.9.2 Computadoras

Una computadora o un computador, es una máquina electrónica que recibe y procesa datos para convertirlos en información útil. Una computadora es una colección de circuitos integrados y otros componentes relacionados que puede ejecutar con exactitud, rapidez y de acuerdo a lo indicado por un usuario o automáticamente por otro programa, una gran variedad de secuencias o rutinas de instrucciones que son ordenadas, organizadas y sistematizadas en función a una amplia gama de

aplicaciones prácticas y precisamente determinadas, proceso al cual se le ha denominado con el nombre de programación y al que lo realiza se le llama programador.



**Figura 1.11 Computadora**

### 1.9.3 Hub

Concentrador o ethernet hub, un dispositivo para compartir una red de datos o de puertos USB de un ordenador. El hub (concentrador) es el dispositivo de conexión más básico. Es utilizado en redes locales con un número muy limitado de máquinas. No es más que una toma múltiple RJ45 que amplifica la señal de la red (base 10/100).[\(8\)](#)



**Figura 1.12 Hub**

## 1.9.4 Repetidor

Este dispositivo sólo amplifica la señal de la red y es útil en las redes que se extienden grandes distancias. En una línea de transmisión, la señal sufre distorsiones y se vuelve más débil a medida que la distancia entre los dos elementos activos se vuelve más grande. Dos nodos en una red de área local, generalmente, no se encuentran a más de unos cientos de metros de distancia. Es por ello que se necesita equipo adicional para ubicar esos nodos a una distancia mayor. (8)

Un repetidor es un dispositivo sencillo utilizado para regenerar una señal entre dos nodos de una red. De esta manera, se extiende el alcance de la red. El repetidor funciona solamente en el nivel físico (capa 1 del modelo OSI), es decir que sólo actúa sobre la información binaria que viaja en la línea de transmisión y que no puede interpretar los paquetes de información.



**Figura 1.13 Repetidor**

## 1.9.5 Router

Un router es un dispositivo de interconexión de redes informáticas que permite asegurar el enrutamiento de paquetes entre redes o determinar la ruta que debe tomar el paquete de datos. Cuando un usuario accede a una URL, el cliente web (navegador) consulta al servidor de nombre de dominio, el cual le indica la dirección IP del equipo deseado.

La estación de trabajo envía la solicitud al router más cercano, es decir, a la pasarela predeterminada de la red en la que se encuentra. Este router determinará así el siguiente equipo al que se le enviarán los datos para poder escoger la mejor ruta posible. Para hacerlo, el router cuenta con tablas de

enrutamiento actualizadas, que son verdaderos mapas de los itinerarios que pueden seguirse para llegar a la dirección de destino. Existen numerosos protocolos dedicados a esta tarea [\(8\)](#)



**Figura 1.14 Router**

## 1.9.6 Switch [\(8\)](#)

Un conmutador o switch es un dispositivo digital de lógica de interconexión de redes de computadores que opera en la capa 2 (nivel de enlace de datos) del modelo OSI. Su función es interconectar dos o más segmentos de red, de manera similar a los puentes (bridges), pasando datos de un segmento a otro de acuerdo con la dirección MAC de destino de las tramas en la red.

Los conmutadores se utilizan cuando se desea conectar múltiples redes, fusionándolas en una sola. Al igual que los puentes, dado que funcionan como un filtro en la red, mejoran el rendimiento y la seguridad de las LANs (Local Area Network- Red de Área Local).



**Figura 1.15 Switch**

## 1.10 Metodología del desarrollo del software

### Metodologías ágiles

Los procesos ágiles son una buena elección cuando se trabaja con requisitos desconocidos o variables. Si no existen requisitos estables, no existe una gran posibilidad de tener un diseño estable y de seguir un proceso totalmente planificado, que no vaya a variar ni en tiempo ni en dinero. En estas situaciones, un proceso adaptativo será mucho más efectivo que un proceso predictivo. Por otra parte, los procesos de desarrollo adaptativos también facilitan la generación rápida de prototipos y de versiones previas a la entrega final, lo cual agradará al cliente. [\(8\)](#)

Las metodologías ágiles proporcionan una serie de pautas y principios junto a técnicas pragmáticas que puede que no curen todos los males pero harán la entrega del proyecto menos complicada y más satisfactoria tanto para los clientes como para los equipos de entrega

Entre las metodologías ágiles más destacadas hasta el momento se pueden nombrar:

- XP (Extreme Programming)
- Scrum

#### 1.10.1 XP (Extreme Programming)

Se requiere un grupo pequeño de programadores para trabajar con esta metodología entre 2 – 15 personas y estas irán aumentando conforme sea necesario.

Sus programadores pueden ser ordinarios. Combina las que han demostrado ser las mejores prácticas de desarrollo de software, y las lleva al extremo. El desarrollo de software es riesgoso y difícil de controlar. Se rediseñará todo el tiempo (**refactoring**), dejando el código siempre en el estado más simple posible. Se harán pruebas todo el tiempo, no sólo de cada nueva clase (**pruebas unitarias**) sino que también los clientes comprobarán que el proyecto va satisfaciendo los requisitos (**pruebas funcionales**).

Las pruebas de integración se efectuarán siempre, antes de añadir cualquier nueva clase al proyecto, o después de modificar cualquiera existente (**integración continua**), utilizando *frameworks de testing*, como el

*xUnit*. Las iteraciones serán radicalmente más cortas de lo que es usual en otros métodos, esto permite beneficiarse de la retroalimentación tan a menudo como sea posible. [\(9\)](#)

## **XP define 4 variables para el proyecto de software:**

- Coste
- Tiempo
- Calidad
- Alcance.

## **XP tiene como valores lo siguiente:**

- Comunicación
- Simplicidad
- Realimentación
- Coraje.

Este es un conjunto mínimo y consistente de valores que permitirán hacer la vida más fácil del grupo, la gerencia y los clientes. Sirve tanto a los fines humanos como a los comerciales.

XP es un sistema de prácticas mínimas, le suponen utilizarlas todas en el principio de un proyecto y adaptarlas y agregar las adicionales como cuando usted experimenta la necesidad. Se puede ver técnico como caso de RUP, aunque él parece ser algo diferente en cultura. En el hecho, racional incluso proporciona un XP plugin para su software de RUP.

XP intenta minimizar el riesgo de fallo del proceso por medio de la disposición permanente de un representante competente del cliente a disposición del equipo de desarrollo. Este representante debería estar en condiciones de contestar rápida y correctamente a cualquier pregunta del equipo de desarrollo de forma que no se retrase la toma de decisiones.

En XP, la programación se hace en parejas, pero el código pertenece al equipo completo, no a un programador o pareja, de forma que cada programador puede cambiar cualquier parte del código en cualquier momento si así lo necesita, dejándose en todo caso las mejoras orientadas al rendimiento, para el

final. Presenta un diseño evolutivo hace que no se le de apenas importancia al análisis como fase independiente, puesto que se trabaja exclusivamente en función de las necesidades del momento.

## 1.10.2 Scrum

Es una metodología para la gestión y desarrollo de software basada en un proceso iterativo e incremental utilizado comúnmente en entornos basados en el desarrollo ágil de software. [\(9\)](#)

Aunque Scrum estaba enfocado a la gestión de procesos de desarrollo de software, puede ser utilizado en equipos de mantenimiento de software, o en una aproximación de gestión de programas.

Scrum es un modelo de referencia que define un conjunto de prácticas y roles, y que puede tomarse como punto de partida para definir el proceso de desarrollo que se ejecutará durante un proyecto. Los roles principales en Scrum son el ScrumMaster, que mantiene los procesos y trabaja de forma similar al director de proyecto, el ProductOwner, que representa a los stakeholders (clientes externos o internos), y el Team que incluye a los desarrolladores.

Durante cada sprint, un periodo entre 15 y 30 días (la magnitud es definida por el equipo), el equipo crea un incremento de software potencialmente entregable (utilizable). El conjunto de características que forma parte de cada sprint viene del Product Backlog, que es un conjunto de requisitos de alto nivel priorizados que definen el trabajo a realizar. Los elementos del Product Backlog que forman parte del sprint se determinan durante la reunión de Sprint Planning. Durante esta reunión, el Product Owner identifica los elementos del Product Backlog que quiere ver completados y los hace del conocimiento del equipo. Entonces, el equipo determina la cantidad de ese trabajo que puede comprometerse a completar durante el siguiente sprint. Durante el sprint, nadie puede cambiar el Sprint Backlog, lo que significa que los requisitos están congelados durante el sprint.

Scrum permite la creación de equipos auto organizado impulsando la localización de todos los miembros del equipo, y la comunicación verbal entre todos los miembros y disciplinas involucrados en el proyecto.

Un principio clave de Scrum es el reconocimiento de que durante un proyecto los clientes pueden cambiar de idea sobre lo que quieren y necesitan (a menudo llamado requirements churn), y que los desafíos impredecibles no pueden ser fácilmente enfrentados de una forma predictiva y planificada. Por lo tanto, Scrum adopta una aproximación pragmática, aceptando que el problema no puede ser completamente

entendido o definido, y centrándose en maximizar la capacidad del equipo de entregar rápidamente y responder a requisitos emergentes.

## Metodologías tradicionales

Las metodologías tradicionales imponen una disciplina de trabajo sobre el proceso de desarrollo del software, con el fin de conseguir un software más eficiente. Para ello, se hace énfasis en la planificación total de todo el trabajo a realizar y una vez que está todo detallado, comienza el ciclo de desarrollo del producto software. Se centran especialmente en el control del proceso, mediante una rigurosa definición de roles, actividades, artefactos, herramientas y notaciones para el modelado y documentación detallada. Además, las metodologías tradicionales no se adaptan adecuadamente a los cambios, por lo que no son métodos adecuados cuando se trabaja en un entorno, donde los requisitos no pueden predecirse o bien pueden variar. [\(9\)](#)

Entre las metodologías tradicionales o pesadas podemos citar:

- MSF (Microsoft Solution Framework)
- RUP (Rational Unified Procces)

### 1.10.3 MSF (Microsoft Solution Framework)

Esta es una metodología flexible e interrelacionada con una serie de conceptos, modelos y prácticas de uso, que controlan la planificación, el desarrollo y la gestión de proyectos tecnológicos. MSF se centra en los modelos de proceso y de equipo dejando en un segundo plano las elecciones tecnológicas. [\(9\)](#)

MSF tiene las siguientes características:

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

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

*Modelo de Arquitectura del Proyecto:* Diseñado para acortar la planificación del ciclo de vida. Este modelo define las pautas para construir proyectos empresariales a través del lanzamiento de versiones.

*Modelo de Equipo:* Este modelo ha sido diseñado para mejorar el rendimiento del equipo de desarrollo. Proporciona una estructura flexible para organizar los equipos de un proyecto. Puede ser escalado dependiendo del tamaño del proyecto y del equipo de personas disponibles.

*Modelo de Proceso:* Diseñado para mejorar el control del proyecto, minimizando el riesgo, y aumentar la calidad acortando el tiempo de entrega. Proporciona una estructura de pautas a seguir en el ciclo de vida del proyecto, describiendo las fases, las actividades, la liberación de versiones y explicando su relación con el Modelo de equipo.

*Modelo de Gestión del Riesgo:* Diseñado para ayudar al equipo a identificar las prioridades, tomar las decisiones estratégicas correctas y controlar las emergencias que puedan surgir. Este modelo proporciona un entorno estructurado para la toma de decisiones y acciones valorando los riesgos que puedan provocar.

*Modelo de Diseño del Proceso:* Diseñado para distinguir entre los objetivos empresariales y las necesidades del usuario. Proporciona un modelo centrado en el usuario para obtener un diseño eficiente y flexible a través de un enfoque iterativo. Las fases de diseño conceptual, lógico y físico proveen tres perspectivas diferentes para los tres tipos de roles: los usuarios, el equipo y los desarrolladores.

*Modelo de Aplicación:* Diseñado para mejorar el desarrollo, el mantenimiento y el soporte, proporciona un modelo de tres niveles para diseñar y desarrollar aplicaciones software. Los servicios utilizados en este modelo son escalables, y pueden ser usados en un solo ordenador o incluso en varios servidores.

## 1.10.4 RUP (Rational Unified Procces)

**RUP** es un marco del proyecto que describe una clase de los procesos que son iterativos e incrementales. Define un manejo entero de las actividades y de los artefactos que usted necesita elegir. Es el proceso de desarrollo más general de los existentes actualmente.

Los procesos de RUP estiman tareas y horario del plan midiendo la velocidad de iteraciones concerniente a sus estimaciones originales. Las iteraciones tempranas de proyectos conducidos RUP se enfocan fuertemente sobre arquitectura del software; la puesta en práctica rápida de características se retrasa hasta que se ha identificado y se ha probado una arquitectura firme. [\(10\)](#)

- 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

Además de las características propias de los procesos iterativos se caracteriza por:

- 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.

RUP es una metodología muy extensa y en la mayoría de 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 estamos obligados a hacer uso de todas las actividades y entregables definidos, sino que podemos configurar nuestro proceso para adaptarnos únicamente a aquellas partes que se consideran necesarias.

El proceso RUP se repite en una serie de ciclos. Cada ciclo concluye con una versión del producto (release) y cada ciclo está dividido en 4 fases: inicio, elaboración, construcción y transición. Cada una de las fases está a su vez dividida en iteraciones, y en cada iteración se realizan 5 procesos o etapas principales: requerimientos, análisis, diseño, implementación, pruebas y entrega.

RUP define por lo tanto el proceso mediante dos dimensiones:

- Dinámica o en el tiempo. Se expresa en términos de ciclos, fases, iteraciones y echas límite o hitos.

- Estática. Se describe en términos de actividades, entregables, roles y flujos de trabajo.

RUP organiza las actividades del proceso en 9 disciplinas totales que se dividen entre 2 grupos: 6 disciplinas básicas y 3 disciplinas de soporte:

- Disciplinas básicas: Modelo del Negocio, Requerimientos, Análisis y Diseño, Implementación, Prueba y Despliegue (o Entrega).
- Disciplinas de soporte: Configuración y Gestión de Cambios, Gestión de Proyectos y Entorno. [\(10\)](#)

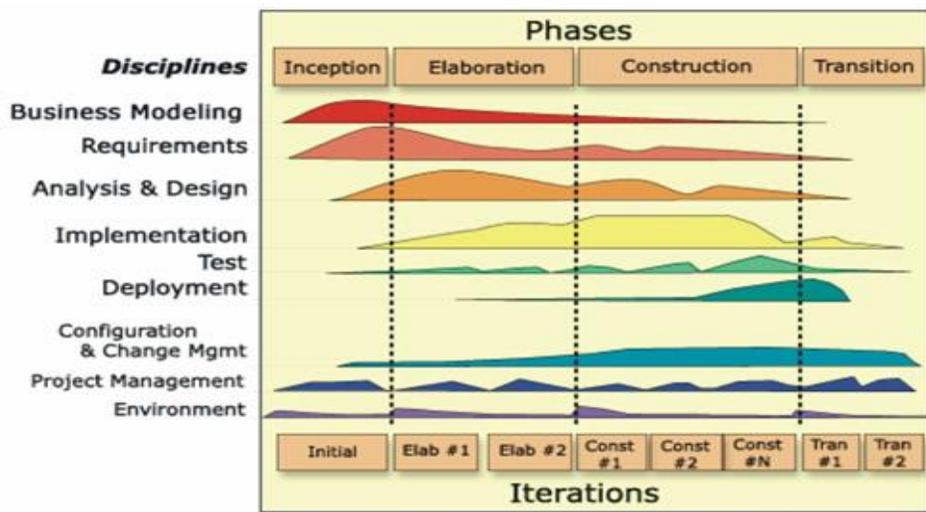


Figura 1.16 Proceso Unificado de Desarrollo de Software

## 1.11 Entorno de desarrollo integrado

### 1.11.1 Qt Creator

Qt Creator es un IDE creado por Trolltech para el desarrollo de aplicaciones con las bibliotecas Qt. Ha sido desarrollado para ser un entorno de desarrollo integrado (IDE) multiplataforma adaptado a las necesidades de los desarrolladores de Qt. Este IDE permite a los desarrolladores crear aplicaciones para múltiples escritorios y plataformas de dispositivos móviles. Es distribuido bajo tres tipos de licencias: Qt Commercial Developer License Qt GNU LGPL v. 2.1 Qt GNU GPL v. 3.0 [\(11\)](#)

Sistemas Operativos.

- GNU/Linux 2.6.x, para versiones de 32 y 64 bits con Qt 4.x instalado. Además hay una versión para Linux con gcc 3.3.
- Mac OS X 10.4 o superior, requiriendo Qt 4.x.
- Windows XP, Vista y Seven, requiriendo el compilador MinGW y Qt 4.4.3 para MinGW.

Principales características de Qt Creator:

- Posee un avanzado editor de código C++.
- Además soporta los lenguajes: C#/.NET Languages (Mono), Python: PyQt y PySide, Ada, Pascal, Perl, PHP y Ruby.
- Posee también una GUI integrada y diseñador de formularios.
- Herramienta para proyectos y administración.
- Ayuda sensible al contexto integrado.
- Depurador visual.
- Resaltado y auto-completado de código.
- Soporte para refactorización de código.

## 1.11.2 Visual Studio [\(12\)](#)

Microsoft Visual Studio es un entorno de desarrollo integrado (IDE, por sus siglas en inglés) para sistemas operativos Windows. Soporta varios lenguajes de programación tales como Visual C++, Visual C#, Visual J#, ASP.NET y Visual Basic .NET, aunque actualmente se han desarrollado las extensiones necesarias para muchos otros.

Visual Studio permite a los desarrolladores crear aplicaciones, sitios y aplicaciones web, así como servicios web en cualquier entorno que soporte la plataforma .NET (a partir de la versión net 2002). Así se pueden crear aplicaciones que se intercomunican entre estaciones de trabajo, páginas web y dispositivos móviles.

## Principales características del Visual Studio

- Provee una serie de herramientas para desarrollo, así como características de debugging, funcionalidad en base de datos y características innovadoras para la creación de aplicaciones en una variedad de plataformas.
- Ofrece a desarrolladores nuevas herramientas para la fácil creación de aplicaciones conectadas en las últimas plataformas incluyendo web, Windows Vista, Office 2007, SQL Server 2008 y Windows Server 2008. Para la web, tenemos ASP.NET, AJAX y otras tecnologías como Silverlight, WPF, etc. que nos dará la posibilidad de crear aplicaciones con rica interfaz de usuario, para poder dar una experiencia de usuario sin precedentes.
- Crea los diagramas de capas a partir de C o C++ codifica y valida las dependencias del código contra los diagramas de capas.
- Genera el código para los diagramas de clases UML.
- Crea la clase UML los diagramas a partir del código.
- Importación la secuencia UML, clase y elementos de diagrama de casos de uso de 2.1 archivos XMI, que puede exportar de otras herramientas.
- Crea los vínculos y vea vincula de los elementos de trabajo para modelar los elementos.
- Proporciona extensibilidad que le permite escribir el código personalizado para modificar los diagramas de capas y validar el código contra ellos.

## 1.12 Herramientas Case.

### 1.12.1 Visual Paradigm

Visual Paradigm es un modelador UML que permite el diseño de sistemas con todo tipo de diagramas UML. También admite el uso extensivo de gestión de casos, el requisito SysML diagrama y diseño de bases de datos (con ERD). Con VP-UML, equipo de desarrollo de software puede realizar el análisis del sistema y diseño de forma eficaz.

Lista de principales características [\(13\)](#)

- Soporte de UML versión 2.1.

- Diagramas de Procesos de Negocio - Proceso, Decisión, Actor de negocio, Documento. Modelado colaborativo con CVS y Subversión (nueva característica). Interoperabilidad con modelos UML2 (metamodelos UML 2.x para plataforma Eclipse) a través de XMI (nueva característica).
- Ingeniería de ida y vuelta.
- Ingeniería inversa - Código a modelo, código a diagrama.
- Editor de Detalles de Casos de Uso - Entorno todo-en-uno para la especificación de los detalles de los casos de uso, incluyendo la especificación del modelo general y de las descripciones de los casos de uso.
- Diagramas de flujo de datos.
- Ingeniería inversa de bases de datos - Desde Sistemas Gestores de Bases de Datos (DBMS) existentes a diagramas de Entidad-Relación.

## 1.12.2 Rational Rose [\(14\)](#)

Rational Rose Enterprise es una buena herramienta para el ambiente de modelado que soporte la generación de código a partir de modelos en Ada, ANSI C++, C++, CORBA, Java™/J2EE™, Visual C++® y Visual Basic®. Como todos los demás productos Rational Rose, proporciona un lenguaje común de modelado para el equipo que facilita la creación de software de calidad más rápidamente.

Características principales:

- Soporte para análisis de patrones ANSI C++, Rose J y Visual C++ basado en "Design Patterns: Elements of Reusable Object-Oriented Software".
- Característica de control por separado de componentes, modelo que permite una administración más granular y el uso de modelos.
- Soporte de ingeniería Forward y/o reversa para algunos de los conceptos más comunes de Java.
- Capacidad de análisis de calidad de código.
- El Add-In para modelado Web provee visualización, modelado y las herramientas para desarrollar aplicaciones de Web.
- Modelado UML para trabajar en diseños de base de datos, con capacidad de representar la integración de los datos y los requerimientos de aplicación a través de diseños lógicos y físicos.

- Publicación web y generación de informes para optimizar la comunicación dentro del equipo.

## 1.13 Lenguaje Unificado de Modelado (UML)

Lenguaje Unificado de Modelado (LUM o UML, por sus siglas en inglés, Unified Modeling Language) es un lenguaje de modelado visual que permite visualizar, especificar, construir y documentar los artefactos que se generan en el proceso de modelado y construcción de un software (14) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad. Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocio y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes reutilizables.

Es importante resaltar que UML es un "lenguaje de modelado" para especificar o para describir métodos o procesos. Se utiliza para definir un sistema, para detallar los artefactos en el sistema y para documentar y construir. En otras palabras, es el lenguaje en el que está descrito el modelo.

Este surge ante la necesidad que tenían los desarrolladores del software de comunicar sus modelos a otros desarrolladores, y no solo entre ellos mismo sino a todos los que estuvieran vinculados con el proyecto e incluso, a desarrolladores de futuras generaciones. Además, también se hacía necesario proporcionar un marco en el que desarrolladores individuales pudieran pensar o analizar y registrar estos modelos en documentos pues no podían retener todo en sus cabezas.

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. [\(15\)](#)

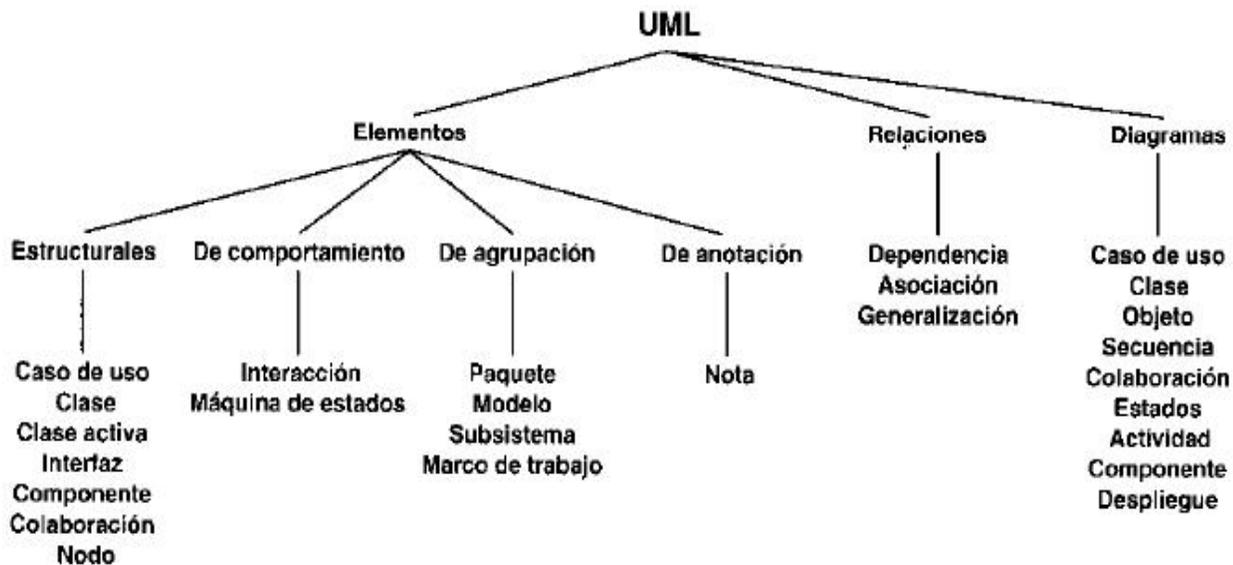


Figura 1.17 Categorías del UML

## 1.14 Lenguajes de Programación

Un lenguaje de programación, es aquel elemento dentro de la informática que permite crear programas mediante un conjunto de instrucciones, operadores y reglas de sintaxis; que pone a disposición del programador para que este pueda comunicarse con los dispositivos de hardware y software existentes.

[\(16\)](#)

### 1.14.1 C++

C++ es un lenguaje de programación diseñado a mediados de los años 1980 por Bjarne Stroustrup. La intención de su creación fue el extender al exitoso lenguaje de programación C con mecanismos que permitan la manipulación de objetos. En ese sentido, desde el punto de vista de los lenguajes orientados a objetos, el C++ es un lenguaje híbrido, se suele decir que es un lenguaje híbrido, ya que permite la programación estructurada. Es un lenguaje de nivel intermedio, pudiéndose utilizar tanto para escribir software de bajo nivel, como drivers y componentes de sistemas operativos, como para el desarrollo rápido de aplicaciones, según el marco de trabajo con el que se disponga, como VCL de Borland C++ Builder.

Los compiladores de C++ generan código nativo con un alto grado de optimización en memoria y velocidad, lo que lo convierte en uno de los lenguajes más eficientes.

Una de las características más interesantes del lenguaje es la sobrecarga de operadores. Esto significa que a los operadores intrínsecos del lenguaje se les puede redefinir la semántica: se pueden escribir funciones que en vez de tener un nombre, se asocian a un operador, que debe tener por lo menos un parámetro de tipo clase.

## 1.14.2 C#

C# es un lenguaje de programación orientado a objetos desarrollado y estandarizado por Microsoft como parte de su plataforma .NET, que después fue aprobado como un estándar por la ECMA e ISO. Su sintaxis básica deriva de C/C++ y utiliza el modelo de objetos de la plataforma.NET, similar al de Java aunque incluye mejoras derivadas de otros lenguajes (entre ellos Delphi).C# forma parte de la plataforma.NET, ésta es una interfaz de programación de aplicaciones (API), mientras que C# es un lenguaje de programación independiente diseñado para generar programas sobre dicha plataforma. Ya existe un compilador implementado que provee el marco de DotGNU - Mono que genera programas para distintas plataformas como Win32, UNIX y Linux.

Principales características del C#

- Sencillez: C# elimina muchos elementos que otros lenguajes incluyen y que son innecesarios en .NET. Por ejemplo:
  - El código escrito en C# es auto contenido, lo que significa que no necesita de ficheros adicionales al propio fuente tales como ficheros de cabecera o ficheros IDL.
  - El tamaño de los tipos de datos básicos es fijo e independiente del compilador, sistema operativo o máquina para quienes se compile (no como en C++), lo que facilita la portabilidad del código.
  - No se incluyen elementos poco útiles de lenguajes como C++ tales como macros, herencia múltiple o la necesidad de un operador diferente del punto (.) acceder a miembros de espacios de nombres (::)
- Modernidad: C# incorpora en el propio lenguaje elementos que a lo largo de los años ha ido demostrándose son muy útiles para el desarrollo de aplicaciones y que en otros lenguajes como Java o C++ hay que simular.

- Orientación a objetos: Como todo lenguaje de programación de propósito general actual, C# es un lenguaje orientado a objetos, aunque eso es más bien una característica del CTS que de C#.

## 1.15 Conclusiones parciales

En este capítulo se dio un paso importante para el desarrollo de la investigación científica pues se realizó el estudio de varios temas de interés, recopilando información necesaria que será utilizada para definir las características del módulo a desarrollar a la hora de hacer una propuesta de la solución.

## CAPÍTULO 2: DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

### 2.1 Introducción

En el presente capítulo se proponen soluciones técnicas para desarrollar el módulo de diseño 2D para el laboratorio Diseño e Instalación de una Red de Área Local, teniendo en cuenta varios aspectos como la metodología, el lenguaje de programación y de modelado a utilizar y el entorno de desarrollo integrado, además se hace mención de las características generales que debe llevar el módulo a desarrollar. Y se definen los requerimientos del sistema tanto los funcionales como los no funcionales.

### 2.2 Propuesta de las soluciones técnicas

- Como metodología se propone RUP, ya que es un proceso de Ingeniería de Software cuyo objetivo es producir software de alta calidad, es decir, que cumpla con los requerimientos de los usuarios dentro de una planificación y presupuesto establecidos, cubriendo el ciclo de vida de desarrollo del software. RUP junto con UML (Unified Modeling Language) constituye el estándar más utilizado para el análisis, implementación y documentación de un software en la industria de hoy en día. El RUP es flexible ya que no es un sistema con pasos firmemente establecidos, sino un conjunto de metodologías adaptables al contexto y necesidades de cada organización.
- Se propone para la solución como lenguaje de programación al C++ debido a que es un lenguaje de programación orientado a objeto lo cual permite la reutilización y extensión del código y agiliza al desarrollo del software, además es uno de los lenguajes más potentes porque nos permite programar a alto y a bajo nivel, este es un lenguaje versátil, general y multiplataforma.
- Como propuesta a la solución se tiene el uso del entorno de desarrollo integrado Qt Creator debido a que este es un software libre, con la ventaja de ser multiplataforma que posee un avanzado editor de código C++ que es el lenguaje de programación seleccionado para el desarrollo del módulo, además cuenta también con una interfaz gráfica de usuario (GUI) integrada y diseñador de formularios. Es una herramienta para proyectos y administración que cuenta con características tales como ayuda sensible al contexto integrada, depurador visual, resaltado y auto-completado de código y soporte para refactorización de código entre otras.
- La propuesta de la herramienta CASE para la solución es el Visual Paradigm para UML ya que es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El software de

modelado UML ayuda a una más rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. La herramienta UML CASE también proporciona abundantes tutoriales de UML, demostraciones interactivas de UML y proyectos UML, además es un software libre.

### 2.3 Modelo de dominio

El modelo de dominio es utilizado para capturar y expresar el entendimiento ganado en un área bajo análisis como paso previo al diseño de un sistema, ya sea de software o de otro tipo .Se hace una representación visual mediante clases conceptuales de los principales conceptos del mundo real a tener en cuenta para el desarrollo del módulo.

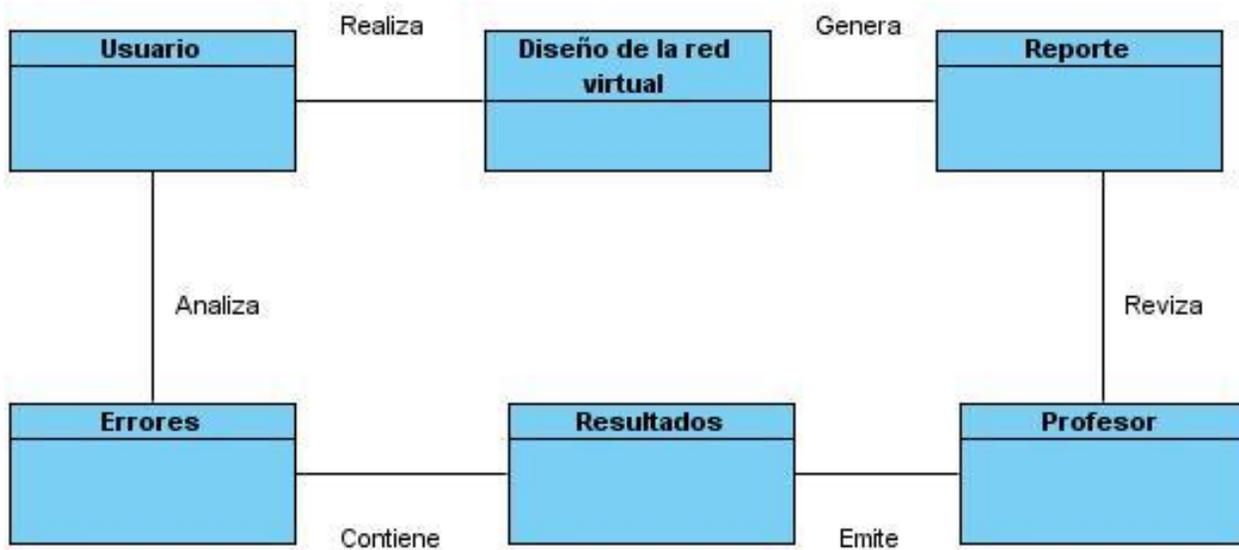


Figura 2.1 Diagrama del Modelo de dominio

### 2.3.1 Glosario de términos

**Usuario:** Es la persona que interactúa con el módulo con el fin de realizar el diseño de una red de área local virtual.

**Diseño de la red virtual:** Es realizado por el usuario y es una representación gráfica de la red virtual creada.

**Reporte:** Informe generado a partir del diseño de la red virtual donde aparecen los datos de la misma.

**Profesor:** Es la persona responsable de revisar el reporte generado.

**Resultados:** Es la evaluación dictada por el profesor después de la revisión del informe.

**Errores:** Aparecen en los resultados emitidos por el profesor con el fin de que el usuario los analice.

### 2.4 Requisitos del sistema

Los requisitos constituyen una necesidad documentada sobre el contenido, forma o funcionalidad de un producto o servicio. Estos facilitan el entendimiento entre usuarios y desarrolladores del sistema a elaborar. Los requisitos del sistema se dividen en funcionales estos definen el comportamiento interno del software y los no funcionales que definen las características de la aplicación.

#### 2.4.1 Requisitos Funcionales

Característica requerida del sistema que expresa una capacidad de acción del mismo, una funcionalidad; generalmente expresada en una declaración en forma verbal. A continuación se muestran los requisitos funcionales que debe cumplir el sistema.

##### R1. Autenticar

El sistema debe posibilitar que el usuario entre sus datos para identificarse.

##### R2.Gestionar componentes

R 2.1 El sistema deber permitir que el usuario adicione nuevos elementos de la red virtual.

R 2.2 El sistema deber permitir que el usuario modifique los elementos de la red virtual.

R 2.3 El sistema deber permitir que el usuario elimine los elementos de la red virtual.

### **R3. Salvar Archivos**

El sistema debe ser capaz de generar un reporte donde se le informe al usuario los datos de la red virtual creada.

### **R4. Enmarcar componentes**

R 4.1 El sistema debe permitir que el usuario agrupe los objetos seleccionados.

R 4.2 El sistema debe permitir que el usuario desagrupe los objetos seleccionados.

### **R5. Limpiar Escena**

El sistema debe permitir que el usuario elimine en una acción todos los componentes de la escena.

#### **2.4.2 Requisitos No Funcionales**

Característica requerida del sistema, del proceso de desarrollo, del servicio prestado o de cualquier otro aspecto del desarrollo, que señala una restricción del mismo. A continuación se muestra los requisitos no funcionales que debe cumplir el sistema.

##### **1. Requerimientos de Usabilidad.**

El sistema deberá ser usado por usuarios capacitados para el uso de la herramienta y que hayan leído previamente el manual de ayuda.

##### **2. Requerimientos de Interfaz.**

El sistema deberá tener una interfaz amigable con una buena utilización de los elementos de diseño, con adecuada combinación de colores.

##### **3. Requerimientos de Soporte.**

El sistema debe tener alguna documentación o ayuda.

#### 4. Requerimientos de Hardware del Sistema.

El sistema para su correcto funcionamiento necesita 256 MB de memoria RAM como mínimo.

#### 5. Requerimientos de Software del Sistema.

El sistema funciona en los siguientes sistemas operativos, Windows XP, Windows Seven ó alguna distribución GNU/Linux.

### 2.5 Descripción del módulo propuesto

Con el fin de darle cumplimiento a los objetivos propuestos al inicio de este trabajo y teniendo en cuenta todos los requerimientos planteados, el módulo que se lleva a proposición debe contar con las siguientes características.

El módulo será utilizado para el modelado de una red de área local de manera virtual.

Interfaz bien definida, debe contar con un área de trabajo, espacio donde se realizará el diseño de la red virtual, un panel de herramientas espacio donde aparecen todos los componentes y los enlaces de la red para ser seleccionados y un menú de tareas que permita ejecutar las funcionalidades del módulo.

Diseño libre teniendo en cuenta que el módulo será utilizado como material de apoyo para una asignatura el usuario podrá realizar el diseño uniendo diferentes tipos de componentes con diferentes tipos de enlaces. Esto posibilita que puedan ser creadas todas las topologías existentes en una red de área local.

Opción de enmarcado, esto permite simular una habitación en el diseño dándole mayor realidad y comodidad en la manipulación de los componentes enmarcados.

Generar reporte, esto permite crear un fichero en donde se refleja el costo de la red virtual creada por el usuario.

Además se cuenta con la existencia de un rol, el del usuario que trabajará con el software.

### 2.6 Modelo de casos de uso del sistema

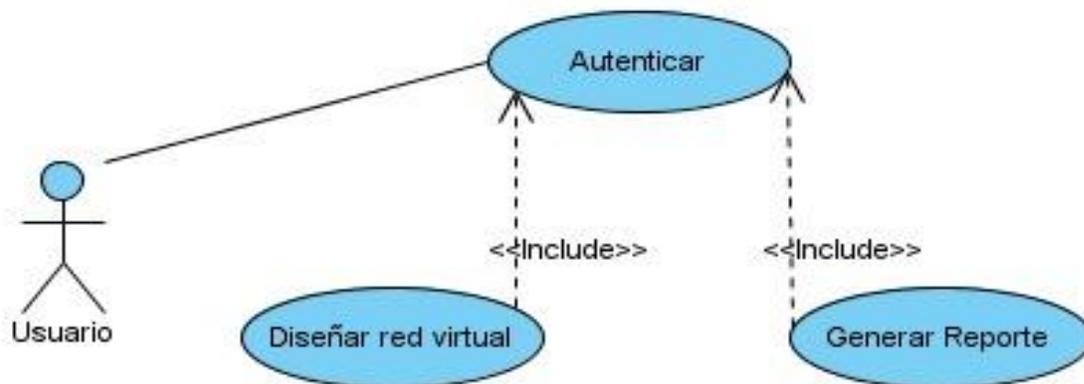
Utilizando el lenguaje de modelado UML se hará una representación de los requisitos funcionales mediante los diagramas de casos de uso además de describir los actores que interactúan con el sistema.

## 2.6.1 Actores del sistema

Actores	Justificación
Usuario	Representa a una persona que tiene la capacidad de interactuar con el módulo y lograr el diseño de una red de área local virtual.

**Tabla 2.1 Actores del sistema**

## 2.6.2 Diagrama de Casos de uso del sistema



**Figura 2.2 Diagrama de Casos de Uso del Sistema.**

## 2.7 Descripción de Casos de Uso del Sistema

Los casos de uso son una secuencia de interacciones que se desarrollarán entre un sistema y sus actores en respuesta a un evento que inicia un actor principal sobre el propio sistema. En las tablas que a continuación se presentan se argumentan los flujos operacionales de cada caso de uso.

<b>CU-1</b>	Autenticar	
<b>Actor</b>	Usuario	
<b>Propósito:</b>	El usuario entra datos al sistema que lo identifiquen.	
<b>Resumen</b>	El caso de uso da inicio cuando el usuario introduce los datos que lo identifican al sistema.	
<b>Precondiciones</b>	El sistema muestra la ventana de autenticarse.	
<b>Referencia</b>	R1.	
<b>Flujo Normal de Eventos</b>		
<b>Acción del Actor</b>	<b>Respuesta del sistema</b>	
1. El usuario introduce los datos que lo identifican y acepta.	1.1 El sistema valida que los datos del usuario estén correctamente. 1.2 Muestra la ventana de la interfaz del módulo terminando así el caso de uso.	
<b>Flujos Alternos</b>		
<b>Acción del Actor</b>	<b>Respuesta del sistema</b>	
	1.2 El sistema muestra un cuadro de dialogo especificando el error en los datos del usuario.	
<b>Prioridad</b>	Crítica	

**Tabla 2.1 Descripción del CU Autenticar**

<b>CU-2</b>	Diseñar red virtual	
<b>Actor</b>	Usuario	
<b>Propósito:</b>	Permitir que el usuario diseñe una red de área local virtual.	
<b>Resumen</b>	El caso de uso da inicio cuando el usuario hace alguna modificación en la escena donde se diseña la red de área local virtual.	
<b>Precondiciones</b>	El sistema debe haber mostrado la interfaz del módulo.	
<b>Referencia</b>	<b>R2.1,R2.2,R2.3,R4.1,R4.2,R5</b>	
<b>Flujo Normal de Eventos</b>		
<b>Acción del Actor</b>	<b>Respuesta del sistema</b>	
2. El usuario puede ejecutar la opción de gestionar componentes, enmarcar componentes o limpiar escena.	1.1 El sistema ejecuta una de las siguientes opciones. <ul style="list-style-type: none"> <li>a) Si el usuario desea gestionar un componente consultar la sección gestionar componente.</li> <li>b) Si el usuario desea enmarcar un componente consultar la sección enmarcar componentes.</li> <li>c) Si el usuario desea limpiar escena consultar la sección limpiar escena.</li> </ul>	
<b>Prioridad</b>	Crítica	

**Tabla 2.2 Descripción del CU Diseñar red virtual**

<b>Sección</b>	Gestionar Componentes	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. Adiciona, modifica o elimina un componente.	1.1 El sistema muestra el cambio en la escena.	
Prioridad	Crítica	

**Tabla 2.3 Descripción de la sección Gestionar Componentes del CU Diseñar red virtual**

Sección	Enmarcar Componentes	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. Selecciona la opción de agrupar o desagrupar los componentes.	1.1 El sistema genera o elimina un marco alrededor de los componentes seleccionados en la escena.	
Prioridad	Secundaria	

**Tabla 2.4 Descripción de la sección Enmarcar Componentes del CU Diseñar red virtual**

<b>Sección</b>	Limpiar Escena	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. Selecciona la opción limpiar escena.	1.1 Elimina todos los componentes de la escena.	
Prioridad	Secundaria	

**Tabla 2.5 Descripción de la sección Limpiar Escena del CU Diseñar red virtual**

<b>CU-3</b>	Generar Reporte	
<b>Actor</b>	Usuario	
<b>Propósito:</b>	El usuario podrá tener acceso al reporte generado por el sistema.	
<b>Resumen</b>	El caso de uso tiene inicio al usuario solicitar un reporte con los datos de la red virtual creada.	
<b>Precondiciones</b>	El sistema debe haber mostrado la interfaz del módulo.	
<b>Referencia</b>	R3.	
<b>Flujo Normal de Eventos</b>		
<b>Acción del Actor</b>	<b>Respuesta del sistema</b>	
1. El usuario presiona el botón salvar.	1.1 El sistema genera un documento en la dirección seleccionada por el usuario con los datos de la red virtual creada.	
<b>Prioridad</b>	Crítica	

**Tabla 2.6 Descripción del CU Generar Reporte**

### 2.8 Conclusiones parciales

Durante este capítulo fueron expuestas las soluciones técnicas para desarrollar el módulo de diseño 2D para el laboratorio Diseño e Instalación de una Red de Área Local, teniendo en cuenta varios aspectos como la metodología, el lenguaje de programación y de modelado a utilizar y el entorno de desarrollo integrado. Se identificaron los requisitos funcionales y no funcionales del sistema. Es realizada también la descripción del módulo propuesto, espacio donde se conocen las características que serán aplicadas en el desarrollo. Se expusieron los casos de uso del sistema, con la correspondiente descripción textual, todo esto con el fin de definir el módulo a desarrollar.

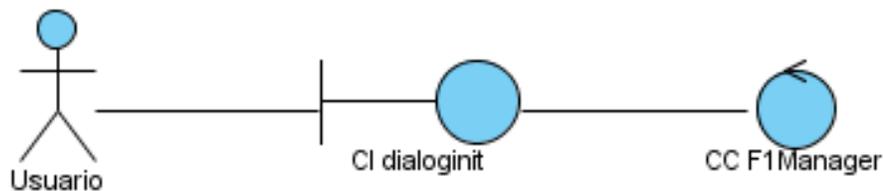
## CAPÍTULO 3: ANÁLISIS Y DISEÑO

### 3.1 Introducción

El siguiente capítulo presenta la concepción general referente al análisis y diseño del sistema propuesto, donde el análisis tiene como objetivo mantener un modelo eficiente de la solución propuesta que sirva de base para el diseño. Teniendo como objetivo que se pueda profundizar más en el desarrollo del sistema, se muestran los diagramas de clases de análisis y diseño. En el análisis se encuentran los diagramas de clases de análisis y los diagramas de secuencias correspondientes a los casos de uso definidos en el capítulo anterior. En el diseño se mostrarán los diagramas de paquetes.

### 3.2 Diagramas de Clases de Análisis.

Para un mayor entendimiento de los procesos efectuados en el dominio del sistema se mostrará un diagrama de secuencia por cada caso de uso.



**Figura 3.1 Diagrama de Clases del Análisis del CU Autenticar**

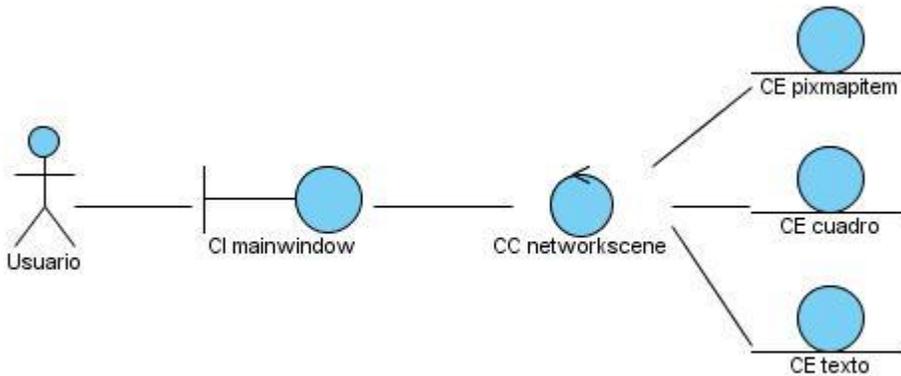


Figura 3.2 Diagrama de Clases del Análisis del CU Diseñar red virtual

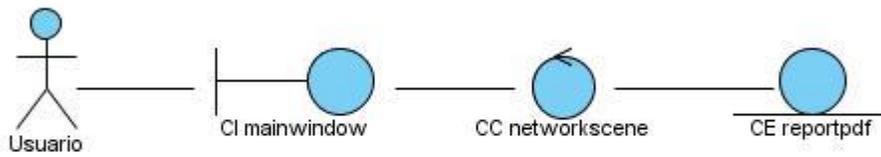


Figura 3.3 Diagrama de Clases del Análisis del CU Generar Reporte

3.3 Diagramas de Secuencia

A continuación se muestran los diagramas de secuencia por cada caso de uso para aclarar la interacción entre los objetos de las clases del análisis.

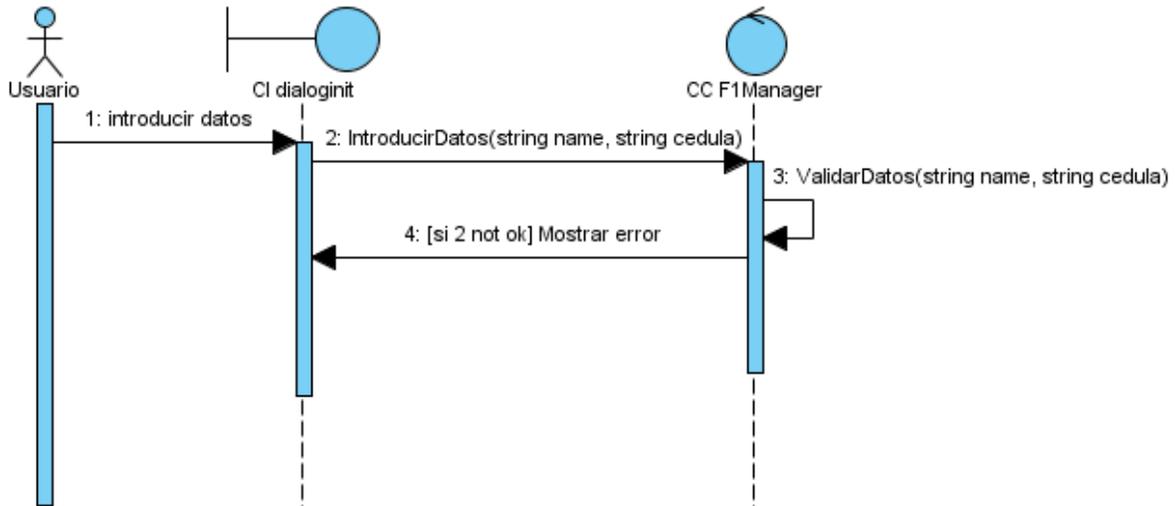


Figura 3.4 Diagrama de Secuencia del CU Autenticar

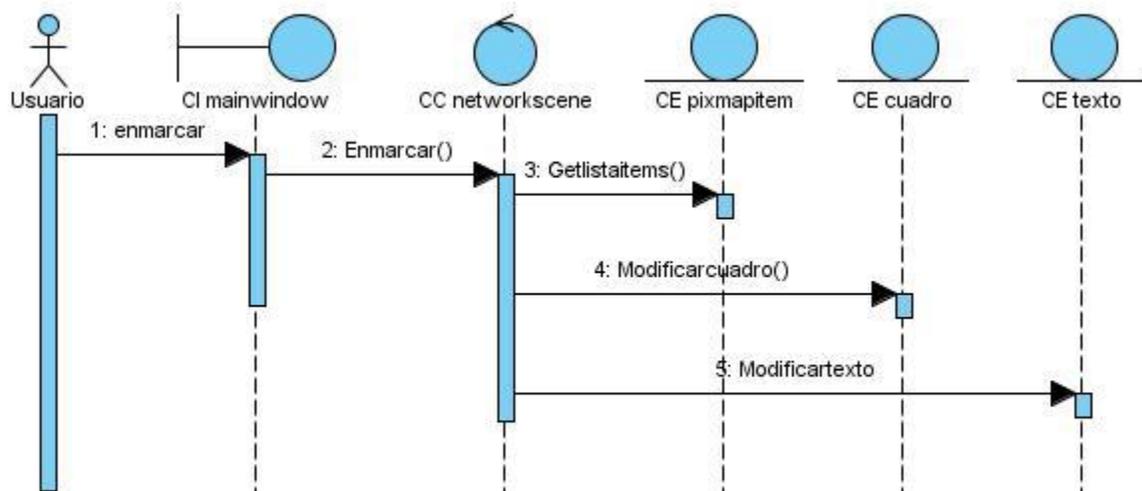


Figura 3.5 Diagrama de Secuencia del CU Diseñar red virtual. Sección Enmarcar

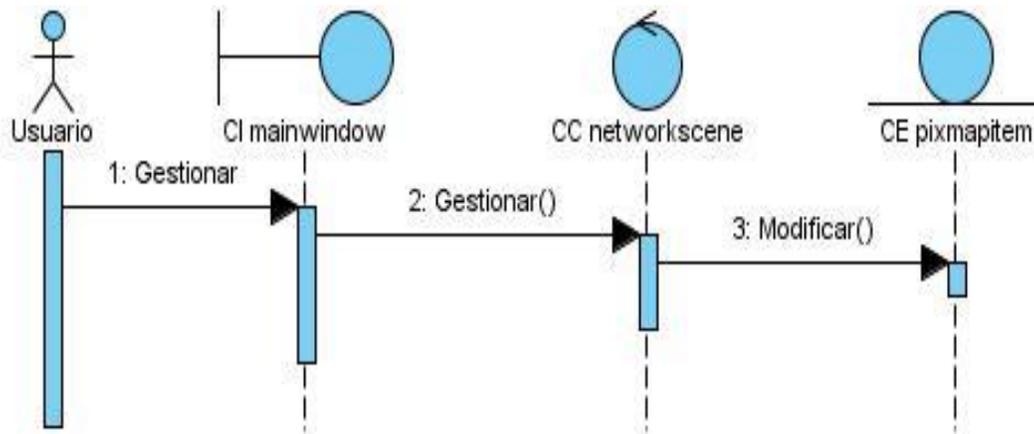


Figura 3.6 Diagrama de Secuencia del CU Diseñar red virtual. Sección Gestionar Componentes

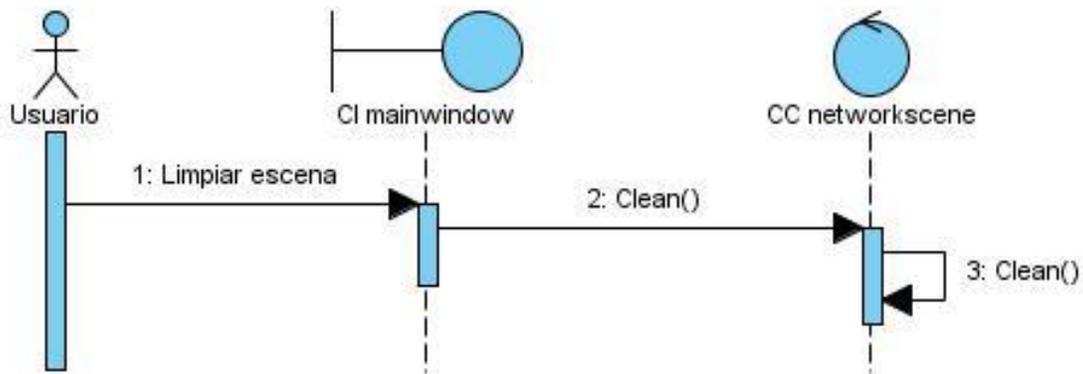


Figura 3.7 Diagrama de Secuencia del CU Diseñar red virtual. Sección Limpiar Escena

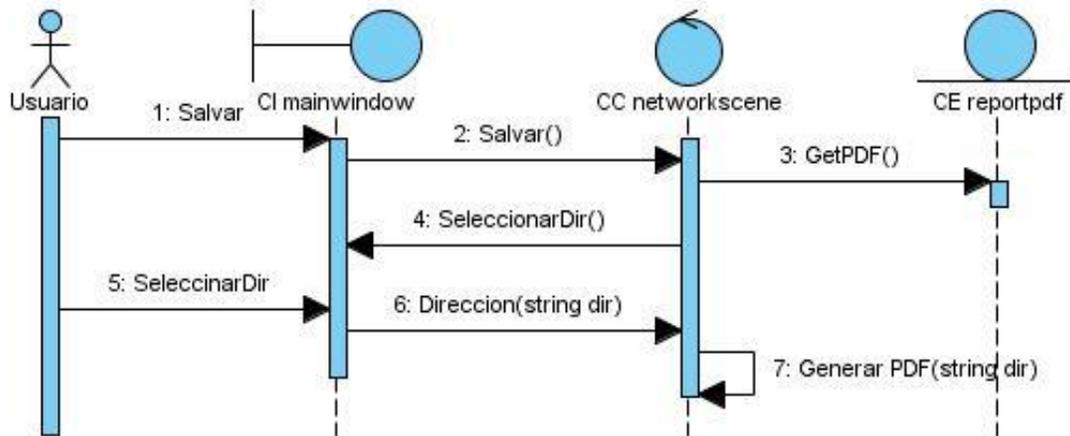


Figura 3.8 Diagrama de Secuencia del CU Generar Reporte

### 3.4 Diagrama de Clases del Diseño del Sistema

A continuación se muestra el diagrama de clases del diseño el cual fue desglosado por paquetes ya que de esta forma es más fácil comprender el mismo. Se desarrollaron 3 paquetes: Lógica de negocio, Presentación y FrameWork Qt. El paquete Lógica de negocio encierra todo lo relacionado con el flujo de datos de la red virtual, datos que son visualizados por el paquete Presentación utilizando todas las funcionalidades del Qt encerradas en el paquete FrameWork Qt.

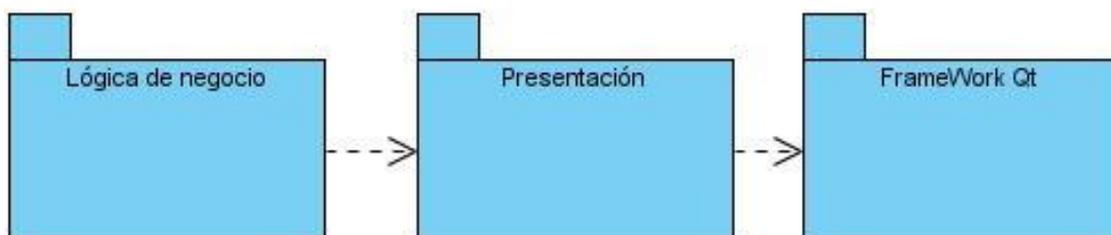


Figura 3.9 Diagrama General de Paquetes de Clase del Diseño

3.4.1 Diagrama de Clases del Paquete Lógica de Negocio

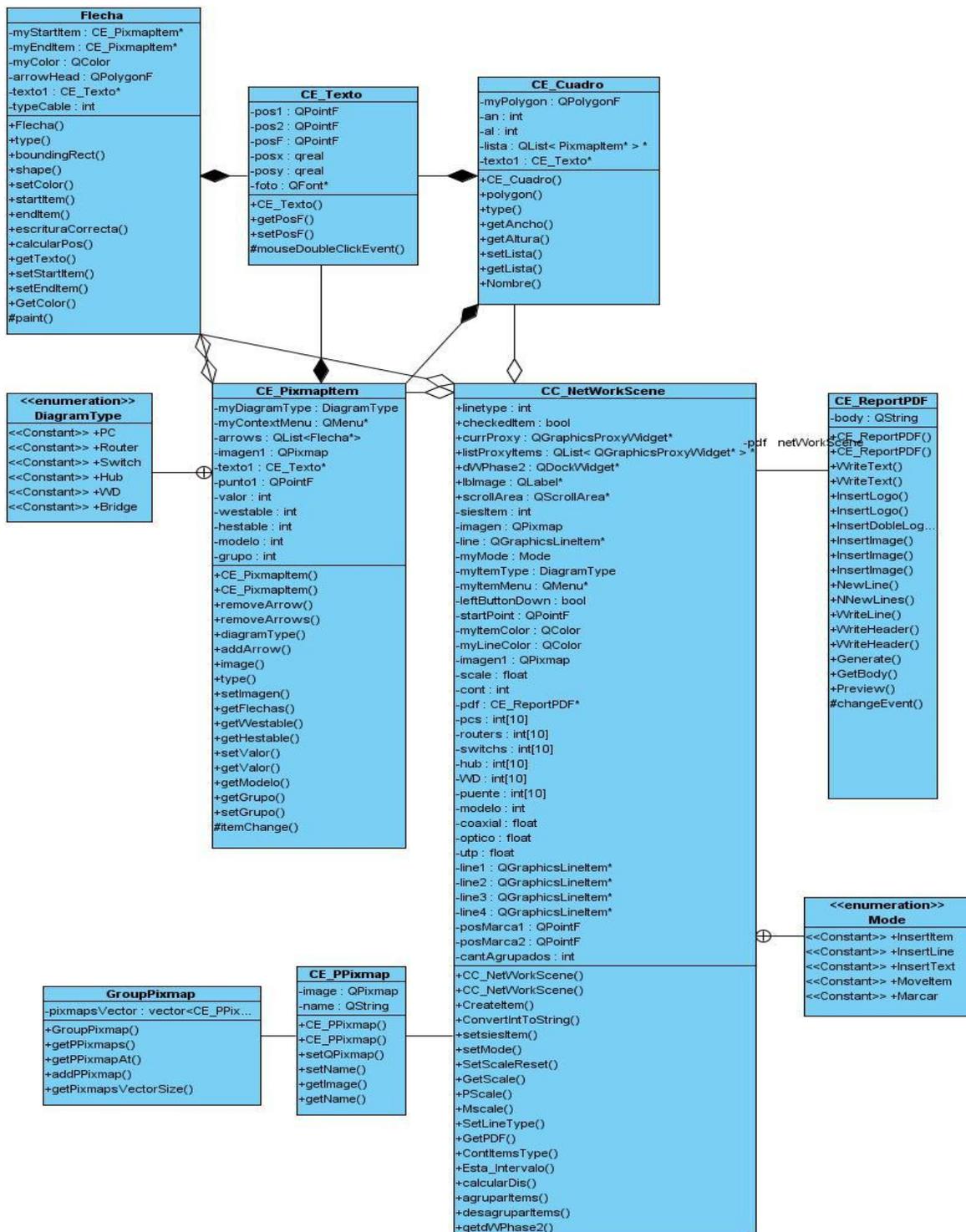


Figura 3.10 Diagrama de Clases de Diseño del Paquete Lógica de Negocio

3.4.2 Diagrama de Clases del Paquete Presentación



Figura 3.11 Diagrama de Clases de Diseño del Paquete Presentación

### 3.5 Conclusiones parciales

En este capítulo se representó la concepción general del análisis y el diseño del sistema, fueron definidos los diagramas tanto del análisis como el diseño, posibilitando esto comprender la lógica del sistema en general.

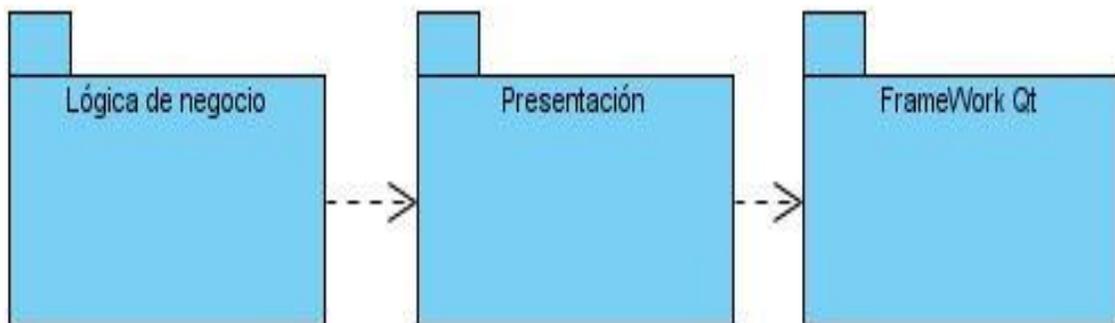
## CAPÍTULO 4: IMPLEMENTACIÓN Y VALIDACIÓN DE LA PROPUESTA

### 4.1 Introducción

En este capítulo se muestra todo lo relacionado con la implementación y la validación de la aplicación a partir de los casos de uso previamente definidos en capítulos anteriores. Se exponen además los diagramas de componentes del sistema.

### 4.2 Diagrama de Componentes

El Diagrama de Componentes define cómo las clases, artefactos y otros elementos de bajo nivel, se unen para formar componentes de alto nivel y las conexiones entre ellos. A continuación se muestra la agrupación en paquetes de todos los componentes.



**Figura 4.1 Diagrama del Paquete de Componentes.**

4.3 Diagrama de Componentes del Paquete Lógica de Negocio

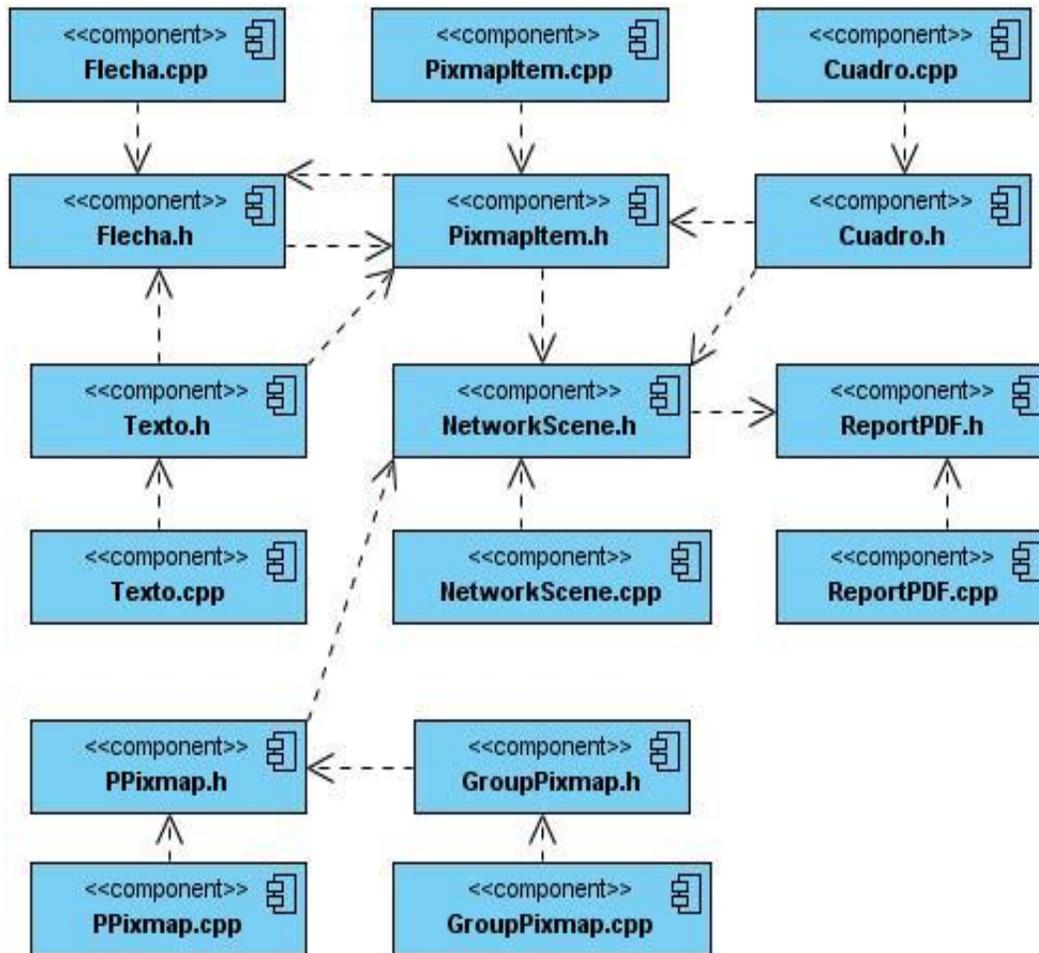


Figura 4.2 Diagrama de Componentes del Paquete Lógica de Negocio.

4.4 Diagrama de Componentes del Paquete Presentación

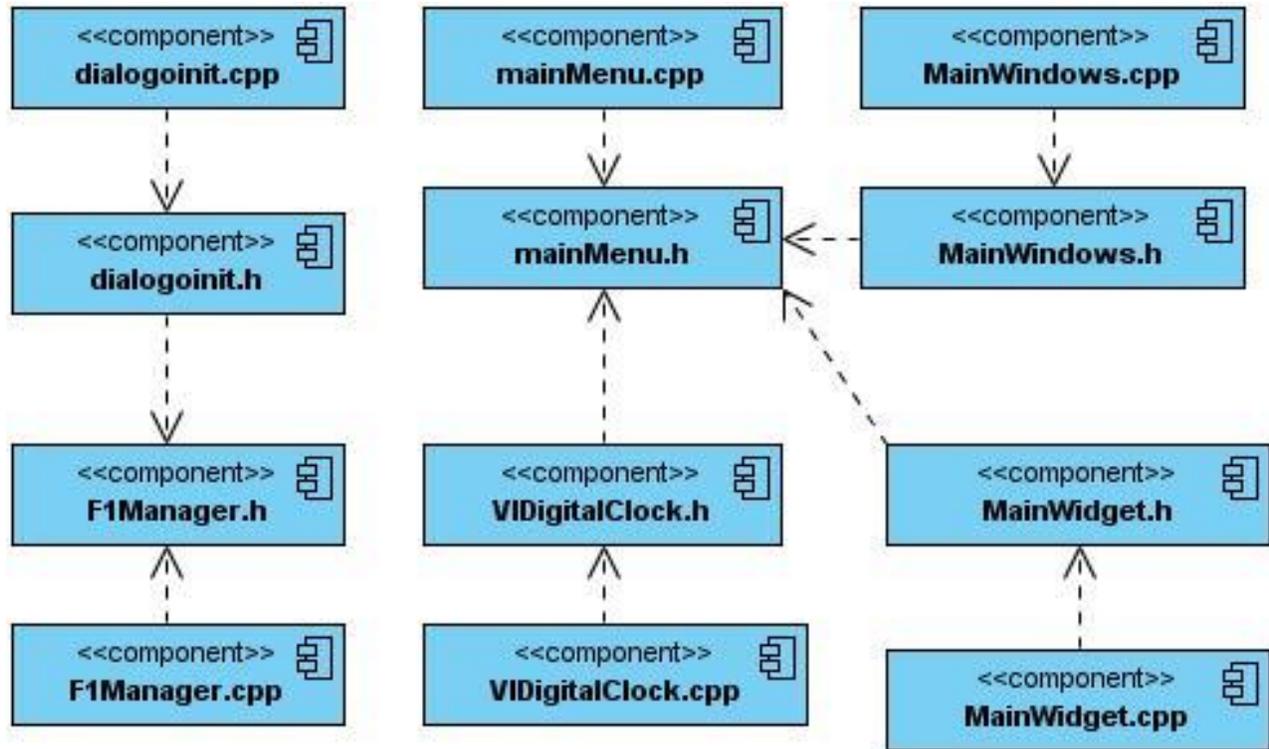


Figura 4.3 Diagrama de Componentes del Paquete Presentación.

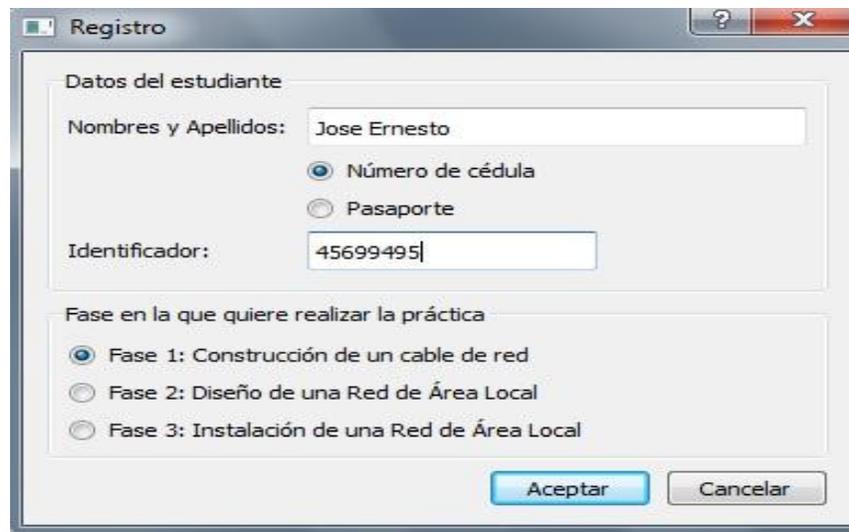
4.5 Validaciones de errores cometidos por el usuario

La validación de las funcionalidades del sistema se realiza por caso de uso, donde por cada caso de uso definido se realiza una descripción de su funcionamiento teniendo en cuenta los valores de: Entrada, Resultado y Descripciones.

4.5.1 Validación del Caso de Uso Autenticar

**Entrada**

El usuario debe introducir los datos: nombre, número de pasaporte o número de cédula.



Registro

Datos del estudiante

Nombres y Apellidos: Jose Ernesto

Número de cédula  
 Pasaporte

Identificador: 45699495

Fase en la que quiere realizar la práctica

Fase 1: Construcción de un cable de red  
 Fase 2: Diseño de una Red de Área Local  
 Fase 3: Instalación de una Red de Área Local

Aceptar Cancelar

Figura 4.4 Interfaz de Autenticación.

### Resultado

Si la autenticación está correcta con los datos introducidos, el usuario ingresa a la práctica del laboratorio, si no se le muestran las ventanas emergentes del error cometido.

### Errores en los datos de autenticación



Registro

Datos del estudiante

Nombres y Apellidos: Jose

Número de cédula

Alerta

El nombre debe de tener un mínimo de 7 caracteres.

Aceptar

Fase 3: Instalación de una Red de Área Local

Aceptar Cancelar

Figura 4.5 Interfaz de Alerta de nombre con menos de 7 caracteres

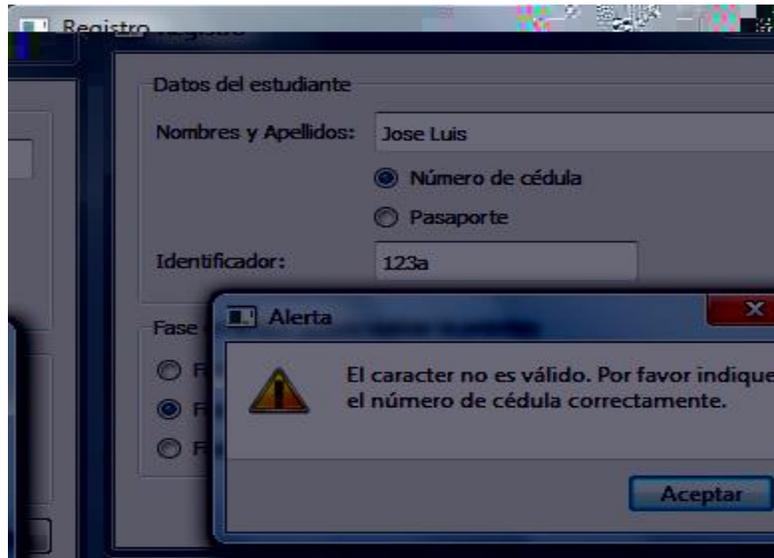


Figura 4.6 Interfaz de Alerta de carácter no válido

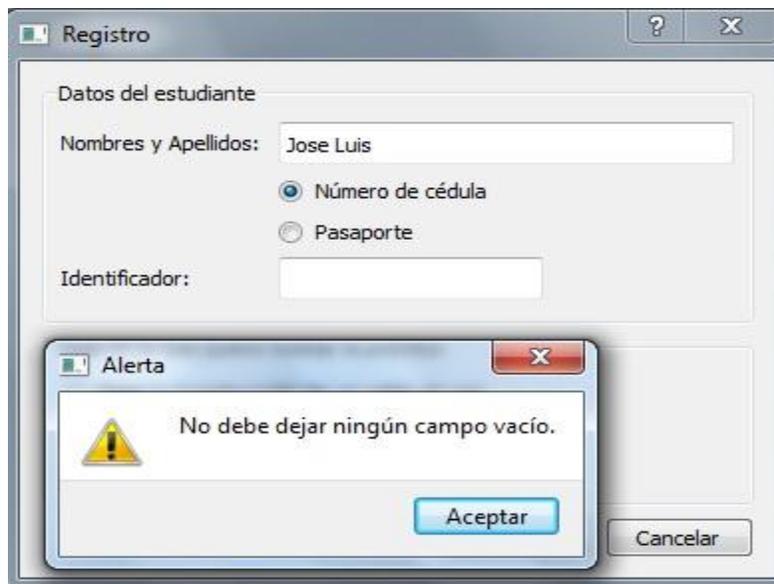
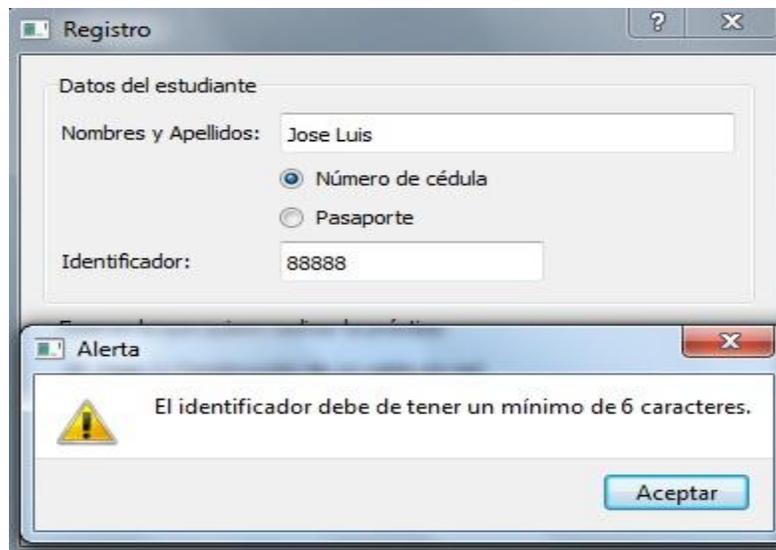


Figura 4.7 Interfaz de Alerta de campo vacío



**Figura 4.8** Interfaz de Alerta de identificador con menos de 6 caracteres.

### Descripción

Se verifica la correcta autenticación del usuario mediante las validaciones correspondientes.

### 4.5.2 Validación del Caso de Uso Diseñar red virtual

#### Entrada

El usuario interactúa con el módulo con el fin de realizar el diseño de la red virtual, utilizando todas las opciones que están a su alcance.

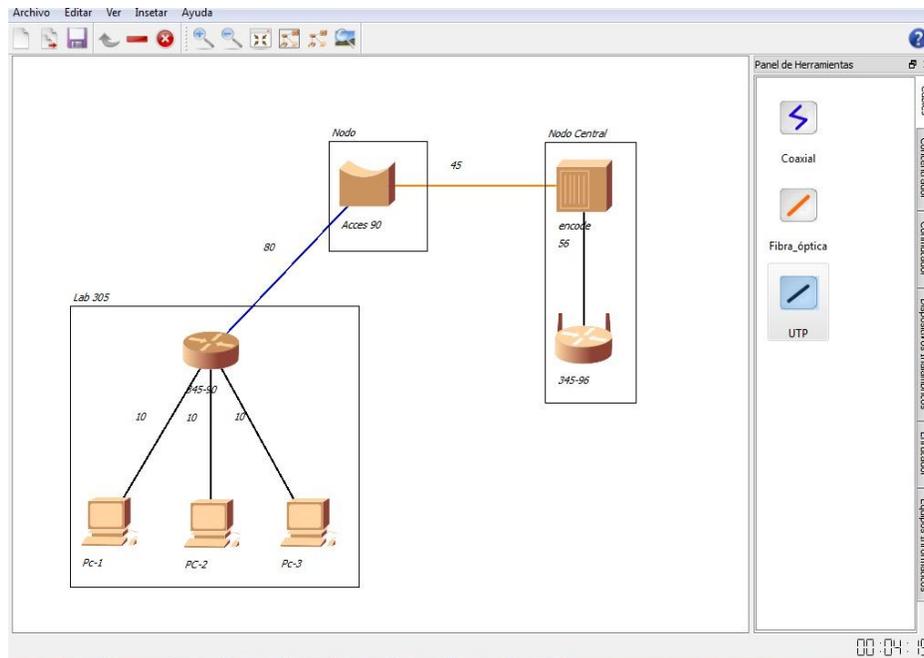


Figura 4.9 Interfaz del módulo

## Resultado

Es graficado el diseño de la red virtual.

## Errores en el diseño de la red virtual

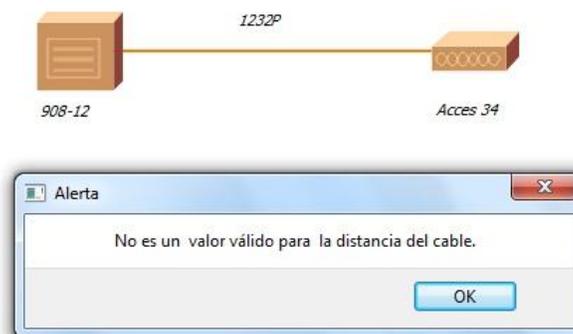


Figura 4.10 Interfaz de Alerta de valor no válido para la distancia del cable

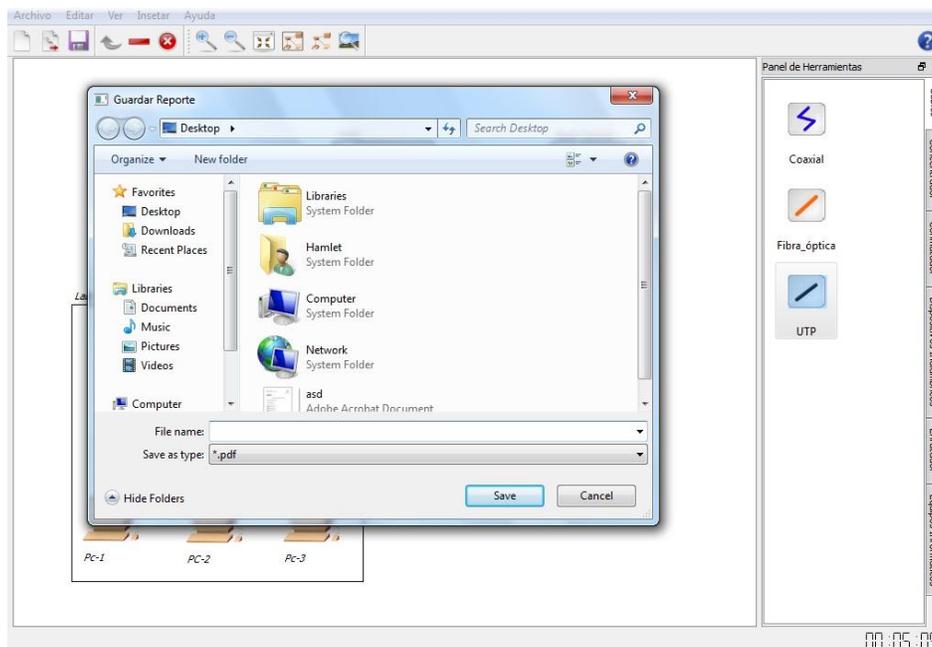
## Descripción

El usuario tiene a su alcance distintas funcionalidades que le permiten diseñar en un ambiente virtual una red de área local.

### 4.5.3. Validación del Caso de Uso Genera Reporte

#### Entrada

Guardar los datos de la red virtual creada por el usuario.



**Figura 4.11 Interfaz de Generar Reporte**

#### Resultado

Es generado un fichero donde se registran los datos de la red virtual creada por el usuario en la práctica del laboratorio.



5/12/2011 12:25:03 PM

Nombre y Apellidos: Hamlet Morales

Identificador: 888888

Código de barra:



## Diseño e Instalación de una Red de Área Local (LAN)

### Diseño de una Red de Área Local (Fase 2)

#### A. Dispositivos seleccionados:

Cantidad de equipos informáticos por modelo:

PC: 3

Cantidad de enrutadores (routers) por modelo:

Enrutador: 1

Cantidad de conmutadores (switches) por modelo:

Puente: 1

Cantidad de concentradores (hubs) por modelo:

Concentrador: 1

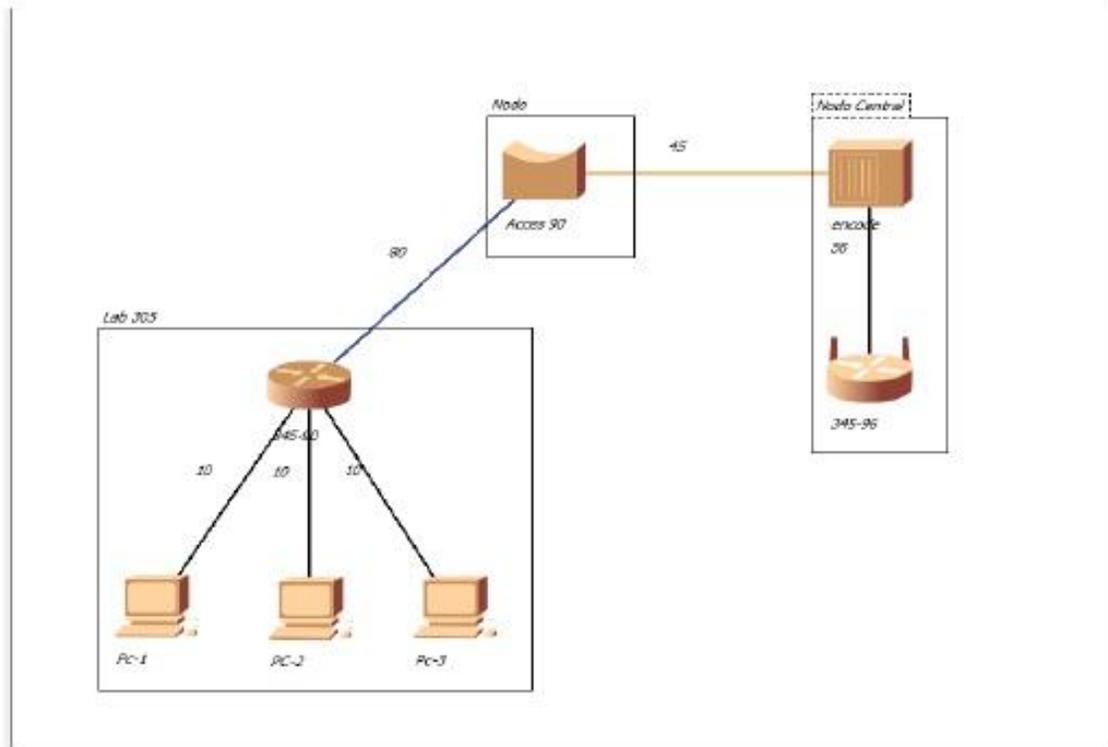
Cantidad de dispositivos inalámbricos por modelo:

Linksys\_WRT300N: 1

Cantidad de cable coaxial utilizado: 80 m

Cantidad de cable UTP utilizado: 86 m

Cantidad de cable fibra óptica utilizado: 45 m



Duración total: 00:05:09 (hh:mm:ss)

**Figura 4.13 Fichero Generado**

## Descripción

Después de haber sido generado el fichero se verifica que contenga todos los datos establecidos en la plantilla.

### 4.6 Conclusiones parciales

En este capítulo se implementó y se validó la propuesta de solución a desarrollar un módulo de diseño 2D para el laboratorio Diseño e Instalación de una Red de Área Local, mostrándose los temas concernientes a la implementación y validación de la aplicación basada en todas las descripciones de los capítulos anteriores.

### CONCLUSIONES GENERALES

La investigación realizada permitió arribar a las siguientes conclusiones:

- ✓ Se elaboró el marco teórico sobre el tema en la actualidad.
- ✓ Se seleccionaron las herramientas, lenguajes y tecnologías a utilizar para el desarrollo del módulo.
- ✓ Se realizó el análisis y diseño del módulo.
- ✓ Se elaboró la propuesta de solución.
- ✓ Se implementó y se validó la propuesta de solución.

Con la culminación del trabajo se logró dar cumplimiento al objetivo general, pues se desarrolló un módulo de diseño 2D para el laboratorio Diseño e Instalación de una Red de Área Local y con la aplicación de este módulo será posible crear un diseño de una red de área local en un entorno virtual, objetivo principal en la práctica del laboratorio. Debido a esto es posible cumplir con el contrato contraído con el cliente y que el proyecto mantenga una buena imagen.

### RECOMENDACIONES

Analizando los resultados del presente trabajo se recomienda:

- ✓ Usar el presente trabajo como material de consulta en proyectos productivos, utilizándolo como apoyo para la capacitación de nuevos programadores y diseñadores.
- ✓ Adicionar nuevas validaciones y funcionalidades al módulo con el fin de convertirlo en una herramienta de diseño y configuración de redes virtuales.

# REFERENCIAS BIBLIOGRÁFICAS

## REFERENCIAS BIBLIOGRÁFICAS

1. Antonio Jiménez, Angélica & Aguilar Vera, Raúl A. *Agentes Pedagógicos Virtuales Inteligentes: Una Estrategia para Entrenamiento de Equipos* .Universidad Autónoma de Yucatán y Universidad Politécnica de Madrid.
2. Blog Informático [Consultado el: 20 de mayo de 2011] Disponible en: <http://www.scribd.com/doc/54444984/conceptos-del-diseno-bidimensional>
3. James P. Instituto Internacional de Física Teórica y Aplicada [Consultado el: 7 de diciembre de 2010] .Disponible en: <http://unesdoc.unesco.org/images/0011/001191/119102s.pdf>
4. Juan del Rosal. Universidad Nacional de Educación a Distancia (UNED), Madrid, España [Consultado el: 12 de diciembre de 2010].Disponible en: <http://www.formatex.org/micte2005>
5. Bonk, C.J. *Herramientas software para la simulación de redes de comunicaciones*. Universidad de España Universidad y Politécnica de Madrid.
6. Berná, J.A., Pérez, M., Crespo, L.M., (2002).Redes de Computadores para Ingenieros en Informática, Publicaciones de la Universidad de Alicante, Alicante
7. Blog Informático [Consultado el: 20 de diciembre de 2010] Disponible en: <http://www.bloginformatico.com/topologia-de-red.php>
8. Tanenbaum, A. S., (1997). Componentes de una red de Área local, Editorial Prentice-Hall, México.
9. Brito, K. Biblioteca virtual de derecho, economía y ciencias sociales [Consultado el: 8 de diciembre de 2010]. Disponible en: <http://www.eumed.net/libros/2009c/584/Methodologias%20tradicionales%20y%20metodologias%20agiles.htm>
10. Días, 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>
11. Blog Informático [Consultado el: 20 de diciembre de 2010] Disponible en: <http://pixelcoblog.com/qt-creator-completo-entorno-de-desarrollo-multiplataforma/>
12. Blog Informático [Consultado el: 16 de abril de 2011] Disponible en: <http://mredison.wordpress.com/2007/12/02/caractersticas-de-visual-studio-2008/>
13. 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/>
14. Rational Rose Data Modeler. [En línea] [Consultado el: 9 de diciembre de 2010]. Disponible en: <http://www.rational.com.ar/herramientas/rosedatamodeler.html>

## REFERENCIAS BIBLIOGRÁFICAS

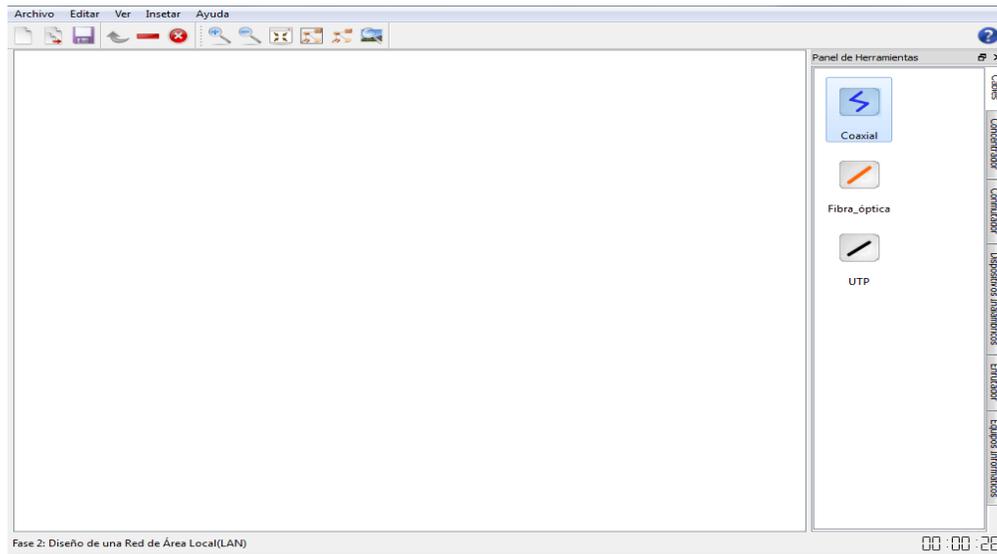
---

15. Parihar, Mridula. ASP.NET. Madrid: ANAYA, 2002. ISBN: 84-4 15-1385-6.

16. Definición de lenguaje de programación. [Consultado el 2 de enero del 2011]. Disponible en: <http://www.definición.org/lenguaje-de-programación>

ANEXO

**Imagen 1:** Interfaz principal del módulo de diseño 2D para el laboratorio Diseño e Instalación de una Red de Área Local.



**Imagen 2:** Diseño de una red de área local.

