

# Universidad de las Ciencias Informáticas

## Facultad 1



Simulador de redes como medio de enseñanza-aprendizaje  
para el desarrollo profesional de los estudiantes de la  
Universidad de las Ciencias Informáticas

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

**Autores:** Antonio Collazo Garcia  
Jorge Fernández Sánchez

**Tutor:** Ing. José Ramón Fernández Pérez

**La Habana, Cuba**

**Junio, 2015**

**DECLARACIÓN DE AUTORÍA**

Declaramos ser autores de la presente tesis que tiene por título: Simulador de redes como medio de enseñanza-aprendizaje para el desarrollo profesional de los estudiantes de la Universidad de las Ciencias Informáticas y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo. Para que así conste firmamos la presente tesis a los \_\_ días del mes de Junio del año 2015.

**Antonio Collazo Garcia**

**Jorge Fernández Sánchez**

\_\_\_\_\_  
Firma del autor

\_\_\_\_\_  
Firma del autor

**Ing. José Ramón Fernández Pérez**

\_\_\_\_\_  
Firma del tutor

**Agradecimientos.**

*A mi madre y mi padre, por su esfuerzo, sacrificio y amor incondicional. Gracias por hacerme sentir tan orgulloso de ustedes.*

*A mi hermano, por guiarme, ayudarme, apoyarme..., en fin, gracias por ser un gran ejemplo para mí y poder contar siempre contigo.*

*A mi novia, por estar presente en todo momento. Gracias por ser tan especial.*

*A Yake y Liuris, muchas gracias por toda su ayuda.*

*A mi familia, por todo su apoyo.*

*A Jorge Fernández, por ser un gran compañero y amigo en estos 5 años.*

*A mis compañeros de aula y apartamento.*

*A todas las personas que me han ayudado durante mi carrera.*

*Antonio Collazo Garcia*

*A mi madre, por siempre creer en mí, por exigirme y mantenerme por el buen camino. Por darme todo su amor, por enseñarme con su ejemplo a ser un luchador y esforzarme por las cosas que quiero. Por estar siempre en las buenas y en las malas. Por ser la mejor madre del mundo.*

*A mi padrastro, a quien incluso puedo llamar mi padre, por educarme y quererme como a su hijo y permitirme a mí, quererlo como a un padre.*

*A toda mi familia, por preocuparse y mantenerse al tanto de mi vida.*

*A todos los nuevos amigos hechos en la UCI, en especial a mis compañeros de aula.*

*A mis hermanos del apartamento, Ángel, Osmel, Manuel y Jorge Ángel.*

*A Antonio Collazo, Marycarmen y Lizandra por ser unas excelentes personas y amigos.*

*A mis amigos, Eddy, José Daniel y Adrian Lamorú.*

*A todos muchas gracias por su apoyo incondicional y su amistad.*

*Jorge Fernández Sánchez*

### **Resumen.**

La introducción de las tecnologías de la información y las comunicaciones en el proceso de enseñanza-aprendizaje de la Universidad de las Ciencias Informáticas ha incorporado nuevos medios de enseñanza-aprendizaje. Un ejemplo es el simulador de redes Cisco Packet Tracer, utilizado para desarrollar la habilidad de configurar equipos de interconexión de redes, pero este simulador además de ser privativo, no se corresponde con las tecnologías presentes en la universidad, Huawei y Allied Telesyn, causando dificultades en el desarrollo profesional de los estudiantes.

El presente trabajo propone el desarrollo de un simulador de redes de las tecnologías Huawei y Allied Telesyn, que pueda ser utilizado como un medio de enseñanza-aprendizaje, para contribuir al desarrollo profesional de los estudiantes de la Universidad de las Ciencias Informáticas en la habilidad de configurar equipos de interconexión de redes. La aplicación proveerá una ayuda proactiva que apoyará el proceso de configuración de los dispositivos de interconexión, así como un mecanismo de evaluación para medir el desempeño del usuario, características que facilitan su integración en las actividades docentes.

**Palabras claves:** medios de enseñanza-aprendizaje, simulador de redes, Huawei y Allied Telesyn.

**Índice de contenido**

Introducción. .... 1

1. Capítulo 1. .... 5

    1.1 Aspectos teóricos. .... 5

    1.2 Análisis de las soluciones existentes. .... 9

    1.3 Entorno tecnológico. .... 12

    1.4 Lenguaje de programación. .... 13

    1.5 Metodología de desarrollo de software. .... 14

        1.5.1 XP (Extreme Programming). .... 15

    1.6 Conclusiones parciales. .... 17

2. Capítulo 2. .... 18

    2.1 Propuesta de solución. .... 18

    2.2 Lista de reserva del producto. .... 18

    2.3 Historias de usuario. .... 20

    2.4 Plan de iteraciones. .... 23

    2.5 Tareas de ingeniería. .... 24

    2.6 Patrón arquitectónico. .... 25

    2.7 Patrones GRASP. .... 26

    2.8 Patrones GoF. .... 29

    2.9 Tarjetas CRC. .... 30

    2.10 Conclusiones parciales. .... 32

3. Capítulo 3. .... 33

    3.1 Estándar de codificación. .... 33

3.2	Pruebas.....	34
3.2.1	Pruebas de aceptación.....	34
3.2.2	Resultado de las pruebas.....	40
3.3	Validación de la propuesta de solución.....	41
3.4	Aplicación del método Delphi.....	43
3.4.1	Selección de los expertos.....	44
3.4.2	Elaboración del cuestionario.....	45
3.4.3	Verificar la concordancia entre los expertos.....	46
3.4.4	Desarrollo práctico y explotación de los resultados.....	47
3.5	Conclusiones parciales.....	48
4.	Conclusiones generales.....	50
5.	Recomendaciones.....	51
6.	Referencias bibliográficas.....	52
7.	Bibliografía.....	54

**Índice de tablas**

Tabla 1: Comparación de herramientas de simulación de equipos de interconexión de redes ..... 11

Tabla 2: Ranking de “agilidad” (Los valores más altos representan una mayor agilidad)..... 15

Tabla 3: Historia de usuario “Diseñar segmentos de redes” .....20

Tabla 4: Historia de usuario “Administrar proyecto” .....21

Tabla 5: Historia de usuario “Configurar PC” .....21

Tabla 6: Historia de usuario “Configurar *switches*” .....21

Tabla 7: Historia de usuario “Mostrar información de los dispositivos” .....22

Tabla 8: Historia de usuario “Crear evaluación” .....22

Tabla 9: Historia de usuario “Administrar proyecto de evaluación” .....23

Tabla 10: Historia de usuario “Realizar evaluación” .....23

Tabla 11: Plan de iteraciones .....24

Tabla 12: Tarea de ingeniería “Crear el núcleo del sistema” .....24

Tabla 13: Tarea de ingeniería “Adicionar y eliminar dispositivos” .....25

Tabla 14: Tarjeta CRC “Device” .....31

Tabla 15: Tarjeta CRC “PC” .....31

Tabla 16: Tarjeta CRC “IP” .....32

Tabla 17: Prueba de aceptación. Insertar dispositivos.....34

Tabla 18: Prueba de aceptación. Conectar dispositivos .....35

Tabla 19: Prueba de aceptación. Nuevo proyecto .....36

Tabla 20: Prueba de aceptación. Guardar proyecto .....36

Tabla 21: Prueba de aceptación. Abrir proyecto .....37

Tabla 22: Prueba de aceptación. Configurar PC .....37

Tabla 23: Prueba de aceptación. Configurar switch.....38

Tabla 24: Prueba de aceptación. Mostrar información de los dispositivos .....39

Tabla 25: Prueba de aceptación. Crear evaluación .....39

Tabla 26: Prueba de aceptación. Realizar evaluación .....40

Tabla 27: Resultados de las pruebas al sistema .....41



**Índice de figuras**

Figura 1: Comparativa entre las diferentes metodologías ..... 16

Figura 2: Patrón arquitectónico: Modelo Vista Controlador ..... 26

Figura 3: Patrón GRASP: experto, clase AT\_VLAN..... 27

Figura 4: Patrón GRASP: creador, clase AT\_Switch ..... 27

Figura 5: Patrón GRASP: polimorfismo, clase Device ..... 28

Figura 6: Patrón GRASP: polimorfismo, clase PC..... 28

Figura 7: Patrón GRASP: polimorfismo, clase AT\_Switch ..... 28

Figura 8: Patrón GRASP: polimorfismo, clase HW\_Switch ..... 28

Figura 9: Patrón GRASP: alta cohesión, clase IP..... 29

Figura 10: Patrón GoF: Singleton, clase MacGenerator ..... 30

**Introducción.**

Una de las misiones de las universidades es lograr una buena formación en sus estudiantes, donde la práctica de los conocimientos teóricos adquiridos juega un papel imprescindible. Dado el desarrollo tecnológico alcanzado por las Tecnologías de la Información y las Comunicaciones (TIC) y su utilización, es necesario que las universidades tengan la capacidad de adaptar estas nuevas tecnologías e incorporarlas en sus Procesos de Enseñanza-Aprendizaje (PEA) y en el desarrollo de sus actividades prácticas. Mundialmente, universidades de alto prestigio como Stanford y Carnegie Mellon, hacen uso de las TIC para la generación de contenidos educativos [1].

Uno de los componentes del PEA en los que ha influido la introducción de las TIC es en los Medios de Enseñanza-Aprendizaje (MEA), que según Heimann<sup>1</sup>, *“son todos aquellos recursos que ponen al alumno en contacto con la realidad de los contenidos”*. [2]

Un ejemplo de esto, es el uso de simuladores como MEA para llevar a la práctica los contenidos adquiridos en las diferentes áreas del conocimiento. Diversas universidades hacen uso de simuladores como parte de su PEA, de las que se citan: la Universidad de Valparaíso con un simulador clínico que permite practicar diversas especialidades médicas a sus estudiantes, con muñecos que reaccionan ante su tratamiento [3]; y la Universidad de Vigo que utiliza un simulador de calorímetro para complementar la práctica de la determinación del calor de neutralización y obtener algunas valoraciones ácido-base en procesos químicos [4].

También, a nivel local la Universidad de las Ciencias Informáticas (UCI) constituye un ejemplo fehaciente en la introducción de las TIC en el PEA, con el propósito de ampliar la disponibilidad de recursos que favorezcan el desarrollo de habilidades profesionales en los estudiantes. Un ejemplo es el uso de simuladores de redes para cubrir el déficit de equipos de interconexión para su utilización en la docencia.

Los simuladores, dentro del mundo de las redes de la información, son programas computacionales diseñados para crear ambientes de comunicaciones con características muy similares a la realidad [5]. El uso de simuladores de redes puede ser un excelente recurso didáctico debido al gran acercamiento a la realidad que presentan, el nivel de confianza que les brindan a los estudiantes al enfrentarse con los dispositivos por primera vez y la posibilidad de implementar y probar complejas redes informáticas.

---

<sup>1</sup> C. Rosales, Didáctica. Núcleos fundamentales, Madrid, 1988.

Actualmente en la UCI se hace uso del simulador de redes CISCO Packet Tracer para desarrollar la habilidad de configurar equipos de interconexión de redes. Este simulador además de ser privativo, permite configurar equipos de interconexión solo de la tecnología CISCO, no correspondiéndose con las tecnologías presentes en la UCI como Huawei y Allied Telesyn.

Como consecuencia de las deficiencias detectadas en el simulador utilizado y la escasez actual de equipos de interconexión de redes en la UCI para su utilización en la docencia, se ve afectado el desarrollo de la habilidad: configurar equipos de interconexión de redes, causando dificultades en el desarrollo profesional de los estudiantes.

Por lo anteriormente planteado se define el siguiente **problema de investigación**: ¿cómo contribuir al desarrollo profesional de los estudiantes de la Universidad de las Ciencias Informáticas en la habilidad de configurar equipos de interconexión de redes?

**Objetivo general:** desarrollar un simulador de redes para contribuir al desarrollo profesional de los estudiantes de la Universidad de las Ciencias Informáticas en la habilidad de configurar equipos de interconexión de redes.

**Objeto de estudio:** simuladores de redes como medios de enseñanza-aprendizaje.

### **Preguntas científicas:**

1. ¿Qué fundamentos teóricos sustentan el diseño de un simulador de redes como medio de enseñanza-aprendizaje?
2. ¿Cómo estructurar una aplicación para la simulación de redes que pueda ser utilizada como medio de enseñanza-aprendizaje?
3. ¿Qué factibilidad traería la implementación de un nuevo simulador de redes, que simule los dispositivos de redes existentes en la universidad, para el desarrollo profesional de los estudiantes en la habilidad de configurar equipos de interconexión de redes?

### **Tareas de investigación:**

1. Determinación de los principales fundamentos teóricos que sustentan el diseño de un simulador de redes como medio de enseñanza-aprendizaje.
2. Análisis de la estructura y funcionalidades de los simuladores que se emplean como medios de

enseñanza-aprendizaje para el desarrollo profesional en la habilidad de configurar equipos de interconexión de redes.

3. Definición de los requisitos funcionales y no funcionales que debe tener un simulador de redes como medio de enseñanza-aprendizaje para el desarrollo profesional de la habilidad: configurar equipos de interconexión de redes.
4. Implementación de un simulador de redes a partir de los requisitos definidos.
5. Aplicación de pruebas al simulador.
6. Validación de la factibilidad del simulador de redes como medio de enseñanza-aprendizaje para el desarrollo profesional de la habilidad: configurar equipos de interconexión de redes.

### **Posibles resultados:**

- Documentación ingenieril asociada a los flujos de trabajo: negocio, requisitos, diseño, implementación y pruebas a partir de la metodología seleccionada.
- Simulador de redes que pueda ser utilizado como un medio de enseñanza-aprendizaje, para contribuir al desarrollo profesional de los estudiantes de la Universidad de las Ciencias Informáticas en la habilidad de configurar equipos de interconexión de redes.

### **Métodos teóricos:**

- Analítico-sintético: para el estudio de las fuentes bibliográficas y conceptos existentes sobre las herramientas de simulación de redes, con el objetivo de poder demostrar la necesidad de la investigación y sintetizar todo el contenido encontrado en las consultas realizadas sobre las fuentes bibliográficas.

### **Métodos empíricos:**

- Observación: utilizado para obtener información sobre los sistemas de simulación de redes existentes y analizar los protocolos, algoritmos y otras características que implementan.
- Revisión documental: se pone de manifiesto al revisar la bibliografía existente y consultar la información en Internet para llevar a cabo las tareas de investigación.
- Entrevista: para obtener información de especialistas en el área de configuración de equipos de

interconexión de redes.

- Encuesta: se pone de manifiesto en las encuestas realizadas a expertos en el área de configuración de equipos de interconexión de redes, para validar la factibilidad de la herramienta de simulación de redes propuesta.

### **Estructura del documento**

El presente trabajo está estructurado en tres capítulos que abarcan todo el proceso de desarrollo de la solución informática propuesta.

**Capítulo 1:** en este capítulo se presentan los elementos teóricos que sirven de base a la investigación del problema planteado. Se describen los conceptos fundamentales asociados al dominio del problema. Además se realiza un análisis de varias herramientas de simulación de redes informáticas. También, se define el entorno tecnológico para el desarrollo de la propuesta de solución.

**Capítulo 2:** en este capítulo se realiza la descripción de la propuesta de solución, sus requisitos funcionales y no funcionales, así como el plan de iteraciones. También se describe la arquitectura del sistema, haciendo énfasis en los patrones de diseño.

**Capítulo 3:** en este capítulo se describe la implementación de la propuesta de solución teniendo en cuenta las técnicas de codificación empleadas. Se presentan las pruebas de aceptación para validar el correcto funcionamiento de la propuesta de solución. También se describe el proceso de validación de la herramienta propuesta como un medio de enseñanza-aprendizaje, haciendo uso del método de expertos Delphi.

## 1. Capítulo 1.

En el presente capítulo se realiza un estudio de los principales conceptos teóricos relacionados con los simuladores de redes, vistos desde la perspectiva que los relaciona con su utilización como medios de enseñanza-aprendizaje. También se hace un análisis de los principales simuladores de redes existentes, definiéndose además el entorno de trabajo utilizado.

### 1.1 Aspectos teóricos.

Con el objetivo de sustentar la investigación se hace necesario definir los aspectos teóricos de los medios de enseñanza-aprendizaje y los simuladores de redes, permitiendo establecer una relación entre ellos; además de los conceptos necesarios asociados al proceso de simulación de redes.

#### Medios de enseñanza-aprendizaje.

Los autores Antonio J. Colom, Jaume Sureda y Jesús Salina plantean que los medios de enseñanza-aprendizaje son: *“aquellos recursos materiales que facilitan la comunicación entre profesores y alumnos. Son recursos instrumentales que inciden en la transmisión educativa, afectan directamente a la comunicación entre profesores y alumnos y tienen sólo sentido cuando se conciben en relación con el aprendizaje.”* [6].

También el Dr. Pere Marqués Graells plantea que los medios de enseñanza-aprendizaje son: *“todos aquellos materiales elaborados con la intención de facilitar los procesos de enseñanza y de aprendizaje”* [7].

Los autores asumen el concepto dado por Araujo y Chadwick que plantea: *“un medio puede definirse como cualquier forma de instrumento o equipamiento que se utiliza normalmente para transmitir información (...) incluyendo las tecnologías desarrolladas en el campo de las comunicaciones”* [8], pues este permite enmarcar a los simuladores de redes como medios de enseñanza-aprendizaje.

#### Simuladores de redes.

Un simulador de redes es un software que predice el comportamiento de una red de computadoras. En los simuladores, las redes de computadoras son típicamente modeladas con dispositivos, enlaces, aplicaciones, etc. [9].

Además según Martínez Zornoza los simuladores, dentro del mundo de las redes de la información, “son programas computacionales diseñados para crear ambientes de comunicaciones con características muy

similares a la realidad” [5].

Para entender mejor el concepto de simulador de redes es necesario aclarar un conjunto de términos relacionados que se exponen a continuación:

### **Redes de computadoras.**

Una red de computadoras es una interconexión de computadoras para compartir información, recursos y servicios. Esta interconexión puede ser a través de un enlace físico (cableado) o inalámbrico [10].

**Dispositivos de Interconexión:** los dispositivos de interconexión son aquellos que hacen posible la comunicación entre las diferentes redes y nodos pertenecientes a una red de computadoras. Estos dispositivos operan en diferentes niveles del modelo de referencia OSI, de forma que cada uno de ellos tiene una funcionalidad diferente [11].

- **Nivel Físico:** repetidores y concentradores de cableado.
- **Nivel de Enlace:** puentes y conmutadores.
- **Nivel de Red:** encaminadores.
- **Niveles Superiores:** pasarelas.

**Repetidores:** los repetidores son dispositivos encargados de regenerar la señal entre los dos segmentos de red que interconectan, extendiendo de esta forma su alcance. Se encarga de tomar la señal que circula por una red y propagarla sin efectuar ningún tipo de traducción o interpretación de dicha señal. Su efecto sobre el retardo de propagación de la señal es mínimo [11].

**Concentradores de cableado:** un concentrador es un dispositivo que actúa como punto de conexión central entre los nodos que componen una red. Los equipos conectados al propio concentrador son miembros de un mismo segmento de red, y comparten el ancho de banda del concentrador para sus comunicaciones.

Los concentradores pueden ser de dos tipos:

- **Activos:** realizan la regeneración de la señal que reciben antes de ser enviada.
- **Pasivos:** en este caso no regeneran la señal, limitándose a interconectar los equipos.

Cuando un equipo envía un mensaje, los datos llegan al concentrador y éste los regenera (si es un concentrador activo) y los retransmite a todos los puestos que estén conectados a cada uno de sus puertos.

**Puentes:** un puente es un dispositivo que conecta dos redes de distintas topologías y protocolos a nivel de enlace [11].

Las funciones de un puente son:

- Dividir una red de área local en dos subredes. Cuando una red de área local se hace demasiado grande, en cuanto a número de nodos, debe ser dividida para que su rendimiento sea mejor.
- Interconectar dos redes de área local, pudiendo tener protocolos de nivel de enlace o medios de transmisión distintos. Como puede ser la interconexión de una red inalámbrica a una cableada.
- Controlar las tramas defectuosas.

**Conmutadores:** un conmutador, también denominado *switch*, es un dispositivo que permite la interconexión de redes de área local a nivel de enlace. A diferencia de los puentes, los conmutadores sólo permiten conectar redes que utilicen los mismos protocolos a nivel físico y de enlace. Su principal función es segmentar una red para aumentar su rendimiento [11].

**Encaminadores:** también denominado *router*, es un dispositivo que permite interconectar redes que operan con una capa de red diferente. Como funciona a nivel de red los protocolos de comunicación en los niveles superiores, a ambos lados del encaminador, deben ser iguales. En una red de área extensa, cualquiera de las estaciones intermedias en la transmisión de un mensaje se considera un encaminador. Por ello, al recibir un paquete, debe extraer de éste la dirección del destinatario y decidir cuál es la mejor ruta, a partir del algoritmo y tabla de encaminamiento que utilice. Además un encaminador dispone de sus propias direcciones a nivel de red [11].

**Pasarelas:** el concepto de pasarela, también denominada *gateway*, es quizás algo abstracto. Básicamente es un sistema de hardware o software que permite interconectar redes que utilizan arquitecturas completamente diferentes con el propósito de que intercambien información. Una pasarela es, por ejemplo, un enrutador que dirige el tráfico desde una estación de trabajo a la red exterior que sirve las páginas Web. En el caso de acceso telefónico, la pasarela sería el ISP (proveedor de servicios de Internet) que conecta el usuario a Internet [11].

**Medios de Transmisión:** el medio de transferencia, es el soporte o forma de conexión que permite la transferencia de datos entre dos dispositivos o nodos en una red. El medio puede ser alambrado (con cables) o inalámbrico. El medio de transmisión es un factor determinante en la velocidad de transmisión de datos



[12].

Algunos medios de transmisión son:

- Radiofrecuencia
- Microondas
- Fibra óptica
- Línea telefónica
- Cable coaxial
- Cable ethernet

#### **Protocolos de red.**

- **IP:** el protocolo que define el mecanismo de entrega sin conexión y no confiable es conocido como Protocolo Internet y, por lo general se le identifica por sus iniciales, IP. El protocolo IP proporciona tres definiciones importantes. Primero, define la unidad básica para la transferencia de datos utilizada a través de una red de redes TCP/IP. Es decir, especifica el formato exacto de todos los datos que pasarán a través de una red de redes TCP/IP. Segundo, el software IP realiza la función de ruteo, seleccionando la ruta por la que los datos serán enviados. Tercero, además de aportar especificaciones formales para el formato de los datos y el ruteo, el IP incluye un conjunto de reglas que le dan forma a la idea de entrega de paquetes no confiable. Las reglas caracterizan la forma en que los anfitriones y ruteadores deben procesar los paquetes, cómo y cuándo se deben generar los mensajes de error y las condiciones bajo las cuales los paquetes pueden ser descartados. El IP es una parte fundamental del diseño de la red de redes TCP/IP, que a veces se conoce como tecnología basada en el IP [13].
- **ICMP:** el Protocolo de Mensajes de Control de Internet (ICMP) permite que los ruteadores envíen mensajes de error o de control hacia otros ruteadores o anfitriones; el ICMP proporciona comunicación entre el software del Protocolo IP en una misma máquina y el mismo software en otra [13].
- **ARP:** (del inglés *Address Resolution Protocol*) protocolo TCP/IP utilizado para asignar una dirección

IP de alto nivel a una dirección de hardware físico de bajo nivel. ARP se utiliza a través de una sola red física y está limitada a redes que soportan difusión de hardware [13].

- **RIP:** (del inglés *Routing Information Protocol*) protocolo utilizado para difundir información de ruteo dentro de un sistema autónomo [13].
- **OSPF:** (del inglés *Open Shortest Path First*) protocolo de ruteo diseñado por la IETF. Utilizado para la actualización de ruteo que implementa el algoritmo de Dijkstra para calcular las rutas más cortas [13].
- **TCP:** (del inglés *Transmission Control Protocol*) protocolo de nivel de transporte TCP/IP estándar que proporciona el servicio de flujo confiable *full duplex*. El TCP/IP permite que el proceso en una máquina envíe un flujo de datos hacia el proceso de otra. Está orientado a la conexión en el sentido de que, antes de transmitir datos, los participantes deben establecer la conexión. Todos los datos viajan en segmentos TCP, donde cada viaje se realiza a través de Internet en un datagrama IP [13].

### **Simulación de redes de computadoras.**

La simulación es una técnica que permite reproducir la esencia de un fenómeno sin reproducir el fenómeno en sí con la ventaja adicional de poder hacerlo en una escala de tiempo relativa, comparada con la que se necesitaría para obtener los mismos resultados utilizando físicamente el objeto a simular. Además, esta técnica permite recrear con “exactitud” los eventos, propiedades, características y funcionamiento de un sistema, permitiendo su análisis, con una inversión menor a la que se necesitaría en la experimentación sobre un modelo real.

Según el comportamiento del modelo, las simulaciones se pueden clasificar en: determinísticas, aquellas en las que la variación de su estado se puede predecir con certeza y estocásticas, que varían su estado de forma aleatoria a través del tiempo, sin seguir algún tipo de patrón [14].

#### **1.2 Análisis de las soluciones existentes.**

El estudio de las herramientas existentes estuvo enfocado en analizar la correspondencia de las mismas con las necesidades definidas por el cliente y determinar características comunes necesarias en las herramientas de simulación de redes. A continuación se presentan algunos de los principales simuladores de redes analizados.

**Enterprise Network Simulator Platform (ENSP)** es una plataforma de simulación de red libre y escalable desarrollada por Huawei. Mediante la simulación de *routers* y *switches* empresariales de Huawei. Permite a los usuarios ejercer las operaciones del dispositivo y aprender las tecnologías de redes sin necesidad de utilizar dispositivos reales. [15]

**FLAN** (del inglés *F- Links And Nodes*), herramienta multiplataforma desarrollada en Java, que permite el diseño, construcción y evaluación de una red de computadores en un ambiente simulado. La sencillez de su interfaz gráfica, facilita el uso en el proceso de enseñanza-aprendizaje, de los protocolos de enrutamiento de las redes [14].

**Kiva** es un software orientado principalmente a simular el comportamiento del protocolo IP, y especialmente para el estudio del tratamiento de los datagramas y el encaminamiento de los mismos por una red. La principal aplicación del programa es en la enseñanza de los fundamentos sobre el funcionamiento de redes de datos. El programa es multiplataforma, dado que todo su entorno fue desarrollado con Java [14].

**VirtualNet** es una herramienta desarrollada en Java por la UCI que permite la simulación de redes de computadoras e integra los contenidos de diseño de redes, direccionamiento IP y de enrutamiento, ejecutándose estos últimos procesos a partir del diseño de una red WAN. Esta herramienta está desarrollada con el objetivo de ser utilizada en la docencia.

**CISCO Packet Tracer** es un simulador gráfico de redes desarrollado y utilizado por Cisco como herramienta de entrenamiento para obtener la certificación CCNA. La herramienta es multiplataforma, permite realizar el diseño de topologías, la configuración de dispositivos de red de la tecnología CISCO, así como la detección y corrección de errores en sistemas de comunicaciones. Posee un mecanismo de evaluación automática orientado a la certificación de niveles.

### **Análisis:**

Para analizar las herramientas de simulación de redes se definieron los siguientes criterios acorde a las necesidades del cliente para su utilización en la docencia:

C1: ¿Es multiplataforma?

C2: ¿Es software libre?

C3: ¿Simula la configuración real de los dispositivos de interconexión?

C4: ¿Posee características que facilitan su uso como medio de enseñanza-aprendizaje?

C5: ¿Configura dispositivos de la tecnología Huawei?

C6: ¿Configura dispositivos de la tecnología Allied Telesyn?

C7: ¿Posee algún mecanismo de evaluación para la comprobación de conocimientos?

C8: ¿Integra tecnologías de distintos fabricantes?

De los cuales se identificaron los criterios 3, 4, 5, 6, 7 y 8 como críticos, debido a que presentan una mayor importancia para el cliente.

**Tabla 1: Comparación de herramientas de simulación de equipos de interconexión de redes**

Simulador	C1	C2	C3	C4	C5	C6	C7	C8
Cisco Packet Tracer	X		X	X			X	
ENSP		X	X		X			
FLAN	X	X	X	X				
Kiva	X	X		X				
VirtualNet	X	X		X				

Tras analizar los resultados del estudio, se determinó que estas herramientas no cumplen con todos los criterios necesarios definidos, presentando una mayor carencia en los criterios 5, 6, 7 y 8.

Durante el análisis se determinaron un conjunto de características comunes presentes en las herramientas analizadas, las cuales se muestran a continuación:

- Capacidad para diseñar segmentos de redes.
- Representación visual de los dispositivos y sus conexiones en forma de grafo.
- Capacidad de simular el comportamiento de distintos dispositivos.

- Mostrar información del estado actual de los dispositivos simulados.

### 1.3 Entorno tecnológico.

A partir de la necesidad de un medio que apoye el proceso de enseñanza-aprendizaje, ajustándose a las tendencias actuales para soluciones de simuladores, a las metodologías para el desarrollo de software y a las tecnologías de software libre, se determinó desarrollar una solución informática bajo el entorno de desarrollo Netbeans, haciendo uso de la herramienta Git para el control de versiones, con el lenguaje de programación Java y guiado a través de la metodología ágil de desarrollo de software XP.

#### Entorno de Desarrollo Integrado: Netbeans 8.0

NetBeans es un proyecto exitoso de código abierto con una gran base de usuarios y una comunidad en constante crecimiento. NetBeans IDE es una herramienta para que los programadores puedan escribir, compilar, depurar y ejecutar programas. Está escrito en Java, pero puede servir para cualquier otro lenguaje de programación. Existe además un número importante de módulos para extender el NetBeans IDE. Este es un producto libre y gratuito sin restricciones de uso [16].

Se decidió utilizar esta herramienta por las siguientes razones:

- El equipo de desarrollo tiene conocimientos y experiencia trabajando con esta herramienta.
- Permite desarrollar interfaces gráficas de usuarios de forma manual, ahorrando considerablemente el tiempo de desarrollo.
- Ofrece herramientas para el autocompletado y refactorización de código Java.
- Detecta errores de codificación sin necesidad de hacer compilación de código.
- Provee mecanismos de sugerencias de código para la corrección de errores.
- Se integra con sistemas de control de versiones.

#### Herramienta CASE: Visual Paradigm for UML

Herramienta CASE que acelera el desarrollo de aplicaciones, sirviendo de intermediario visual entre arquitectos, analistas y diseñadores de software, mediante un ambiente de modelado superior que posibilita un trabajo más fácil y dinámico [17].

Se decidió utilizar esta herramienta por las siguientes razones:

- Es multiplataforma y cuenta con una versión libre.
- Genera la documentación del sistema en los formatos PDF, HTML y el formato de documentos de Microsoft Word.
- Soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción y despliegue.
- Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación.

## **Sistema de control de versiones: Git**

GIT es un sistema de control de versiones, completamente distribuido e increíblemente rápido, con eficiente soporte para proyectos de pequeño, mediano y largo alcance.

Inicialmente fue desarrollado por Linus Torvalds para mantener el control de cambios del núcleo de Linux, pero actualmente es ampliamente utilizado por la comunidad de desarrolladores mundial. Es empleado en muchos sitios de alojamiento de proyectos de software como GitHub <<https://github.com>>, Bitbucket <<https://bitbucket.org>> y en la UCI Codecomunidades <<https://codecomunidades.uci.cu>> donde actualmente se encuentra alojado el código de la aplicación.

Se decidió utilizar esta herramienta porque el equipo de desarrollo tiene conocimientos y experiencia trabajando con ella. Además, existe un creciente auge en la comunidad de desarrollo que impulsa el uso de la misma.

### **1.4 Lenguaje de programación.**

Java es un lenguaje de programación desarrollado por Sun Microsystems a principios de los años 90 que implementa el paradigma orientado a objetos. Su sintaxis es similar a la de lenguajes como C y C++ [18].

Se decidió utilizar Java por las siguientes razones:

- Tiene un modelo de programación orientado a objetos simple que elimina herramientas de bajo nivel, que suelen inducir a muchos errores, como la manipulación directa de punteros o memoria.
- Las aplicaciones Java están típicamente compiladas en un *bytecode*, que permite que las mismas

sean multiplataforma.

- El equipo de desarrollo tiene conocimientos y experiencia trabajando con este lenguaje.

### **1.5 Metodología de desarrollo de software.**

Las metodologías de desarrollo de software tienen como objetivo dar solución a los problemas existentes en la producción de software, que cada vez son más complejos. Éstas engloban procedimientos, técnicas, documentación y herramientas que se utilizan en la creación de un producto de software de gran valor y alta calidad. El éxito del producto depende, en gran medida, de la selección de la metodología escogida, sea tradicional o ágil.

Dada las características del equipo de desarrollo, el cual cuenta con tres integrantes (dos desarrolladores y el cliente), se hace necesario utilizar una metodología de desarrollo de software ágil, por las siguientes razones:

- Están especialmente preparadas para cambios durante el proyecto.
- El cliente es parte del equipo de desarrollo.
- Los equipos de desarrollo son pequeños y trabajan en el mismo sitio.
- Generan pocos artefactos.
- Definen pocos roles.

Tras analizar el nivel de agilidad de las metodologías de desarrollo ágiles, expuesto en la tabla 2 [19], se ha decidido utilizar XP por su alto índice de agilidad, lo cual permite al equipo de desarrollo enfocarse más en la implementación de la aplicación y menos en la generación de artefactos asociados a una metodología, debido a que se necesitan hacer iteraciones cortas que permitan mostrar al cliente una versión funcional de la aplicación.

**Tabla 2: Ranking de “agilidad” (Los valores más altos representan una mayor agilidad)**

	CMM	ASD	Crystal	DSDM	FDD	LD	Scrum	XP
Sistema como algo cambiante	1	5	4	3	3	4	5	5
Colaboración	2	5	5	4	4	4	5	5
Características Metodología (CM)								
- Resultados	2	5	5	4	4	4	5	5
- Simplicidad	1	4	4	3	5	3	5	5
- Adaptabilidad	2	5	5	3	3	4	4	3
- Excelencia técnica	4	3	3	4	4	4	3	4
- Prácticas de colaboración	2	5	5	4	3	3	4	5
<b>Media CM</b>	2.2	4.4	4.4	3.6	3.8	3.6	4.2	4.4
<b>Media Total</b>	1.7	4.8	4.5	3.6	3.6	3.9	4.7	4.8

A continuación se exponen las principales características de esta metodología así como las prácticas básicas que deben seguirse.

### 1.5.1 XP (Extreme Programming).

La programación extrema es una metodología ágil desarrollada por Kent Back para la construcción de software. La misma se basa en la simplicidad, comunicación y reutilización del código desarrollado. Su idea básicamente consiste en desarrollar subversiones del software, realizando entregas con frecuencia (cada dos semanas o menos). Cada subversión, se debe codificar de manera simple para que funcione correctamente. El diseño se hace a medida que se avanzan sobre las subversiones. El código y el diseño se van modificando continuamente, añadiéndoles funcionalidades [20].

A diferencia de las metodologías tradicionales donde se plantea un desarrollo en vertical, con cuatro procesos claramente diferenciados (análisis, diseño, implementación y pruebas) los cuales se realizan de manera secuencial en grandes periodos de tiempo cada uno; o de las metodologías iterativas las cuales proponen un desarrollo en espiral; XP propone una única etapa de análisis en la cual se defina el alcance del proyecto, para después ir resolviendo pequeños problemas planteados en el análisis, realizando paralelamente los procesos de diseño, implementación y pruebas lo que desemboca en pequeñas entregas



que van resolviendo requisitos de usuario y son validadas por el cliente [20].

En la figura 1 [19], puede observarse claramente las características comentadas de cada una de las metodologías:

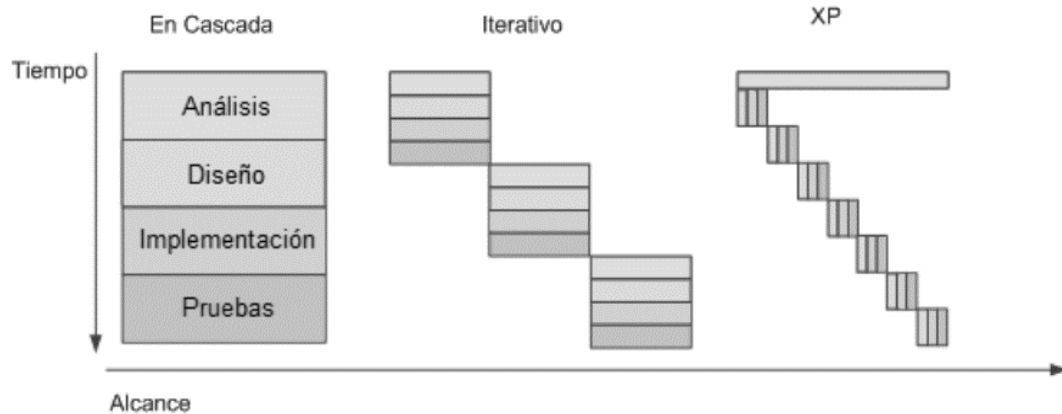


Figura 1: Comparativa entre las diferentes metodologías

A continuación se muestran algunas de las prácticas básicas planteadas por XP [20]:

1. **Equipo completo:** está formado por el equipo de desarrollo, el cliente y el responsable del proyecto.
2. **Planificación:** se observan las historias de usuario y se planifica en qué orden se van a realizar. Puesto que la planificación puede o no respetarse, es necesario revisarla constantemente.
3. **Pruebas del cliente:** el cliente, con la ayuda de los desarrolladores, propone sus propias pruebas.
4. **Versiones pequeñas:** las subversiones deben ser lo suficientemente pequeñas como para poder implantarse y probarse en un periodo de no más de una semana.
5. **Diseño simple:** a medida que se implementa el código se deben de ir dando algunos retoques a la interfaz, analizando cada una de ellas y eliminando partes innecesarias de la misma.
6. **Pareja de programadores:** dos personas aportan más al código que una sola.
7. **Mejorar el código:** mientras se codifica debe mejorarse el código, extraer funcionalidades comunes, eliminar código innecesario, etc.
8. **Integración continua:** cuando se tenga una nueva funcionalidad, debe recompilarse y probarse.

9. **El código es de todos:** cualquier persona dentro del equipo de desarrollo debe conocer el código y poder modificarlo.
10. **Normas de codificación:** se debe definir un estilo común de codificación.
11. **Metáforas:** hay que buscar frases que definan cómo funcionan las distintas partes del programa, así todo el equipo entenderá de qué se está hablando.
12. **0 horas extras y días 0:** se quiere señalar que no deben realizarse horas extras, y no dejar días sin hacer nada.

### 1.6 Conclusiones parciales.

La caracterización de las herramientas, metodología, lenguajes, conceptos teóricos sobre redes de computadoras y procesos de simulación, permitió la familiarización con los elementos del entorno de desarrollo y la obtención de los conocimientos necesarios para de esta forma poder utilizarlos en la construcción de la propuesta de solución.

El estudio de los diferentes sistemas de simulación de redes existentes en la actualidad para los procesos de diseño y simulación de redes, permitió adquirir un conjunto de conocimientos previos sobre las características fundamentales que presentan estos sistemas. A pesar de que los mismos no satisfacen las necesidades y condiciones requeridas, constituyen una base de estudio para llegar a la obtención de la propuesta de solución.

## 2. Capítulo 2.

En el presente capítulo se describe la propuesta de solución teniendo en cuenta los pasos y artefactos definidos para el proceso de desarrollo que propone la metodología XP. Se definen las historias de usuario asociadas a los requisitos del cliente. Además se elaboró el plan de iteraciones que registra las historias de usuario por cada iteración.

### 2.1 Propuesta de solución.

Para dar solución al problema planteado se propone desarrollar una herramienta de simulación de redes, la cual permitirá simular la configuración de dispositivos de interconexión de las tecnologías Huawei (switch capa 3, serie 5600) y Allied Telesyn (switch capa 2, serie 8500). Además permitirá diseñar segmentos de redes de computadoras en un entorno gráfico, posibilitando representar estos dispositivos y sus conexiones. La aplicación constará además con un mecanismo de evaluación que permitirá calificar el desempeño del usuario a partir de una evaluación previamente creada. También permitirá guardar el estado actual del trabajo realizado como un proyecto, que luego podrá ser cargado en la aplicación para su posterior modificación.

Como parte de la configuración de los dispositivos de interconexión, se abarcarán los contenidos relacionados con la administración de VLANs; la definición del modo de acceso de los puertos dentro de las VLANs; así como la implementación de los protocolos: ARP para asignar una dirección IP de alto nivel a una dirección de hardware físico de bajo nivel, RIP para transportar información de ruteo e ICMP para comprobar la comunicación entre dispositivos.

### 2.2 Lista de reserva del producto.

**Requisitos funcionales:** son aquellas acciones o funciones que el sistema será capaz de realizar y describen las transformaciones que el sistema realiza sobre las entradas para producir salidas [21].

- Diseñar segmentos de redes.
  - Adicionar dispositivos (PC, *switch* Huawei y *switch* Allied Telesyn).
  - Conectar dispositivos (conexión serial y conexión *ethernet*).
  - Eliminar conexiones y dispositivos.

- Administrar proyecto.
  - Guardar proyecto.
  - Abrir proyecto.
  - Crear proyecto.
- Configurar PC.
  - Configurar nombre, IP, máscara de red y *gateway* de las PC.
- Configurar *switches*.
  - Establecer parámetros para la conexión.
  - Configurar las propiedades de los *switches* mediante líneas de comando.
  - Mostrar ayuda proactiva de los comandos.
- Mostrar información de los dispositivos.
  - Mostrar estado de los puertos de los *switches*.
  - Mostrar tabla ARP de los *switches*.
  - Mostrar tabla de ruta de los *switches* L3.
- Crear evaluación.
  - Generar lista de acciones.
  - Modificar lista de acciones.
  - Asignar valores de puntuación a cada acción.
- Administrar proyecto de evaluación.
  - Guardar proyecto de evaluación.
  - Abrir proyecto de evaluación.
- Realizar evaluación.
  - Evaluar.

- Generar reporte de evaluación.

**Requisitos no funcionales:** son propiedades o cualidades que el producto debe o necesita cumplir [21].

- **Portabilidad:** el sistema debe poder ser ejecutado sobre los Sistema Operativos: Linux y Windows.
- **Usabilidad:** las vistas de la aplicación deben indicar en cada momento la acción que se está realizando, así como los íconos deben estar representados por una imagen acorde a la acción que realizan.
- **Software:** se debe disponer de la máquina virtual de Java versión 1.7 o superior.
- **Hardware:** se requiere como mínimo, una computadora con 96 Mb de RAM y un procesador a 500MHz, para el correcto funcionamiento de la aplicación (ver Anexo 14).

### 2.3 Historias de usuario.

Las Historias de Usuario (HU) describen las funcionalidades del software y son escritas por el cliente empleando un lenguaje claro. Permiten responder rápidamente a las funcionalidades cambiantes y son la base para las pruebas funcionales del sistema. Se emplean para hacer estimaciones de tiempo [22].

A continuación se detallan las HU (tablas de la 3 a la 10) realizadas que abarcan las funcionalidades del sistema propuesto, donde cada punto estimado representa una semana de trabajo (5 días: lunes, martes, miércoles, jueves, viernes).

**Tabla 3: Historia de usuario “Diseñar segmentos de redes”**

Número: 1	Historia de usuario		
<b>Nombre:</b> Diseñar segmentos de redes.			
<b>Puntos estimados:</b> 2	<b>Iteración asignada:</b> 1	<b>Prioridad:</b> Alta	
<b>Descripción:</b> la aplicación permitirá adicionar dispositivos (PC y <i>switches</i> ) al área de trabajo, así como conectarlos entre sí con los diferentes medios de conexión ( <i>serial</i> y <i>ethernet</i> ).			

**Tabla 4: Historia de usuario “Administrar proyecto”**

Número: 2	Historia de usuario	
<b>Nombre:</b> Administrar proyecto.		
<b>Puntos estimados:</b> 1	<b>Iteración asignada:</b> 1	<b>Prioridad:</b> Media
<b>Descripción:</b> la aplicación permitirá guardar, abrir y crear nuevos proyectos.		
<b>Nota:</b> los proyectos se guardarán con la extensión “.ath”.		

**Tabla 5: Historia de usuario “Configurar PC”**

Número: 3	Historia de usuario	
<b>Nombre:</b> Configurar PC.		
<b>Puntos estimados:</b> 1	<b>Iteración asignada:</b> 2	<b>Prioridad:</b> Alta
<b>Descripción:</b> la aplicación permitirá configurar las propiedades de conexión de las PCs.		
<b>Nota:</b> las propiedades a configurar son: nombre, IP, máscara de red y <i>gateway</i> .		

**Tabla 6: Historia de usuario “Configurar switches”**

Número: 4	Historia de usuario	
<b>Nombre:</b> Configurar <i>switches</i> .		
<b>Puntos estimados:</b> 4	<b>Iteración asignada:</b> 2	<b>Prioridad:</b> Alta
<b>Descripción:</b> la aplicación permitirá configurar las propiedades de los <i>switches</i> Huawei y Allied Telesyn, mediante líneas de comandos.		

**Nota:** se deberá establecer una conexión previa desde una pc con los parámetros de conexión correctos. La aplicación permitirá configurar el estado de los puertos, gestionar VLANs, definir rutas estáticas, configurar RIP y gestionar el acceso a los dispositivos.

**Tabla 7: Historia de usuario “Mostrar información de los dispositivos”**

Número: 5		Historia de usuario	
<b>Nombre:</b> Mostrar información de los dispositivos.			
<b>Puntos estimados:</b> 1	<b>Iteración asignada:</b> 2	<b>Prioridad:</b> Media	
<b>Descripción:</b> la aplicación deberá mostrar información sobre el estado de los dispositivos seleccionados.			
<b>Nota:</b> se mostrará información sobre el estado de los puertos, tablas ARP y tablas de rutas de los dispositivos.			

**Tabla 8: Historia de usuario “Crear evaluación”**

Número: 6		Historia de usuario	
<b>Nombre:</b> Crear evaluación.			
<b>Puntos estimados:</b> 3	<b>Iteración asignada:</b> 3	<b>Prioridad:</b> Alta	
<b>Descripción:</b> la aplicación permitirá crear una evaluación a partir del diseño y configuración de una red, así como modificar la puntuación de cada acción realizada.			

Tabla 9: Historia de usuario “Administrar proyecto de evaluación”

Número: 7	Historia de usuario	
<b>Nombre:</b> Administrar proyecto de evaluación.		
<b>Puntos estimados:</b> 1	<b>Iteración asignada:</b> 3	<b>Prioridad:</b> Alta
<b>Descripción:</b> la aplicación permitirá guardar y abrir un proyecto de evaluación.		
<b>Nota:</b> los proyectos de evaluación se guardarán con la extensión “.eath”.		

Tabla 10: Historia de usuario “Realizar evaluación”

Número: 8	Historia de usuario	
<b>Nombre:</b> Realizar evaluación.		
<b>Puntos estimados:</b> 2	<b>Iteración asignada:</b> 3	<b>Prioridad:</b> Alta
<b>Descripción:</b> a partir de una evaluación creada, el sistema analizará las acciones realizadas por el usuario y de esta forma ofrecerá una evaluación del trabajo realizado, así como un reporte con las incidencias de la evaluación.		

## 2.4 Plan de iteraciones.

Para el desarrollo de la aplicación se dividieron las historias de usuario en las siguientes iteraciones:

**Iteración 1:** esta iteración tiene como objetivo la implementación de todo el ambiente gráfico de la aplicación y las funcionalidades básicas del sistema.

**Iteración 2:** en esta iteración se implementarán las historias de usuario relacionadas con la configuración de los equipos.

**Iteración 3:** en esta iteración se implementarán las funcionalidades de crear y realizar evaluaciones.



Tabla 11: Plan de iteraciones

Iteraciones	Orden de las historias de usuario	Duración (semanas )
Iteración 1	Diseñar segmentos de redes.	3
	Administrar proyecto.	
Iteración 2	Configurar PC.	6
	Configurar <i>switches</i> .	
	Mostrar información de los dispositivos.	
Iteración 3	Crear evaluación.	5
	Realizar evaluación.	

## 2.5 Tareas de ingeniería.

Durante el transcurso de las iteraciones, se realiza la implementación de las historias de usuario definidas por el cliente y descritas por el equipo de desarrollo. Las historias de usuario se descomponen en tareas de ingeniería, las cuales son asignadas a los programadores para ser implementadas durante la iteración correspondiente. Una historia de usuario puede tener asociada varias tareas de ingeniería para poder darle respuesta a las necesidades del cliente.

A continuación se muestran dos de las tareas de ingeniería propuestas para desarrollar el sistema, el resto de las tareas quedan plasmadas en el Anexo 1.

Tabla 12: Tarea de ingeniería “Crear el núcleo del sistema”

Tarea de ingeniería	
Número: 1	Historia de usuario: 1

<b>Nombre:</b> Crear el núcleo del sistema.	
<b>Tipo:</b> desarrollo	<b>Puntos estimados:</b> 0.8
<b>Fecha inicio:</b> 8/12/2014	<b>Fecha fin:</b> 11/12/2014
<b>Programador responsable:</b> Antonio Collazo Garcia y Jorge Fernández Sánchez	
<b>Descripción:</b> crear el sistema de clases referente a los dispositivos y controladores necesarios, así como las interfaces de usuario que permitan definir un diseño inicial del sistema.	

Tabla 13: Tarea de ingeniería “Adicionar y eliminar dispositivos”

Tarea de ingeniería	
<b>Número:</b> 2	<b>Historia de usuario:</b> 1
<b>Nombre:</b> Adicionar y eliminar dispositivos.	
<b>Tipo:</b> desarrollo	<b>Puntos estimados:</b> 1.2
<b>Fecha inicio:</b> 12/12/2014	<b>Fecha fin:</b> 17/12/2014
<b>Programador responsable:</b> Jorge Fernández Sánchez	
<b>Descripción:</b> implementar las funcionalidades necesarias para permitir que el usuario pueda añadir y eliminar elementos al área de trabajo que representen los dispositivos de interconexión y las computadoras.	

## 2.6 Patrón arquitectónico.

**Modelo Vista Controlador (MVC):** el patrón MVC es una guía para el diseño de arquitecturas de aplicaciones que ofrezcan una fuerte interactividad con el usuario. MVC separa el sistema en tres capas

fundamentales, la capa modelo, representa los datos y su representación en entidades relacionadas con la aplicación; la capa vista o presentación encargada de representar los datos del modelo y permitir al usuario interactuar con estos; y una capa controladora encargada de procesar las peticiones del usuario provenientes de la capa presentación y donde está encapsulada la lógica del negocio [23].

En la propuesta de solución se utilizaron las librerías de interfaces gráficas de usuario que vienen en el SDK de Java, llamadas Java Swing, que se basa en el patrón MVC, presentando la particularidad de que el controlador y la vista están implementados como un único elemento denominado “Delegado de Interfaz de Usuario (*UI delegate*)”. A continuación se muestra un diagrama de la relación entre los componentes del MVC utilizado en el desarrollo de la propuesta de solución (ver figura 2).

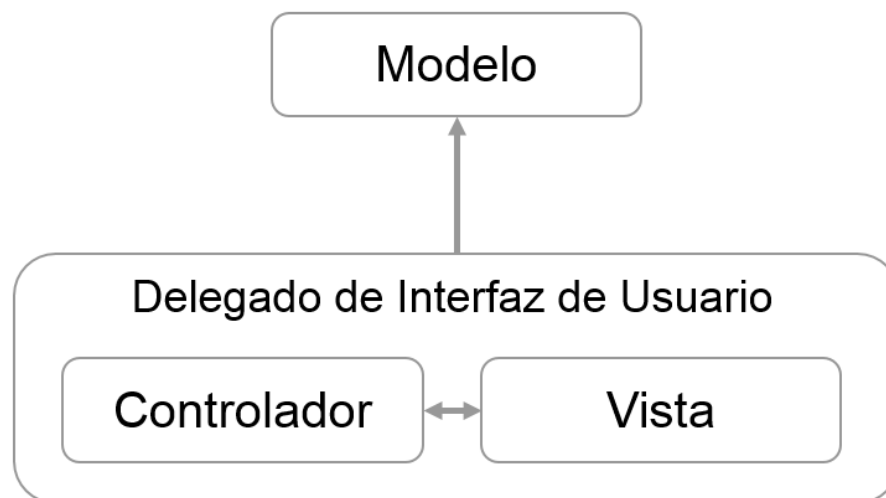


Figura 2: Patrón arquitectónico: Modelo Vista Controlador

## 2.7 Patrones GRASP.

Para la implementación del sistema se tuvo en cuenta el uso de los patrones GRASP (del inglés *General Responsibility Assignment Software Patterns*), también conocidos como los patrones para la asignación de responsabilidades. Los patrones GRASP son parejas de problema-solución con un nombre, que codifican buenos principios y sugerencias relacionados frecuentemente con la asignación de responsabilidades, constituyendo un apoyo que ayuda a entender el diseño de objetos [24].

Para el desarrollo de la aplicación se tuvieron en cuenta los siguientes patrones GRASP:

- **Patrón experto:** las responsabilidades de realizar una labor es de la clase que tiene o puede tener los datos involucrados. Una clase, contiene toda la información necesaria para realizar la labor que tiene encomendada. A continuación se muestra un ejemplo en la figura 3 de su utilización en la clase AT\_VLAN.

```
public class AT_VLAN implements Serializable{
    private String name;
    private int vid;
    private IP ip;
    private IP mask;
    private ArrayList<AT_Port> taggedPorts;
    private ArrayList<AT_Port> untaggedPorts;
```

Figura 3: Patrón GRASP: experto, clase AT\_VLAN

- **Patrón creador:** asigna a la clase B la responsabilidad de crear una instancia de la clase A bajo la condición de que B agrega, contiene, registra o usa objetos de la clase A, o tiene la información inicial que será pasada a la clase A cuando es creada. Se evidencia su utilización en la clase AT\_Switch (ver figura 4).

```
public class AT_Switch extends Device implements Runnable, Serializable {
    private ArrayList<AT_VLAN> vlans;
    private ArrayList<AT_Port> ports;

    public void addVLAN(AT_VLAN newVLAN) {...14 lines }
```

Figura 4: Patrón GRASP: creador, clase AT\_Switch

- **Polimorfismo:** se hace uso del polimorfismo, cuando las alternativas o comportamientos relacionados varían según el tipo (clase), se asigna la responsabilidad para el comportamiento utilizando operaciones polimórficas a los tipos para los que varía el comportamiento. Se muestra su utilización en las clases Device, PC, AT\_Switch y HW\_Switch (ver figuras 5, 6, 7 y 8).

```
public abstract class Device extends JLabel implements Serializable{
    public abstract void recieve(Trama trama, Device aThis);
    public abstract void send(Trama trama, Device aThis);
}
```

Figura 5: Patrón GRASP: polimorfismo, clase Device

```
public class PC extends Device implements Serializable {
    @Override
    public void recieve(Trama trama, Device dev) {...73 lines }

    @Override
    public void send(Trama trama, Device dev) {...37 lines }
}
```

Figura 6: Patrón GRASP: polimorfismo, clase PC

```
public class AT_Switch extends Device implements Runnable, Serializable {

    @Override
    public void recieve(Trama trama, Device dev) {...64 lines }

    @Override
    public void send(Trama trama, Device dev) {...154 lines }
}
```

Figura 7: Patrón GRASP: polimorfismo, clase AT\_Switch

```
public class HW_Switch extends Device implements Serializable {

    @Override
    public void recieve(Trama trama, Device dev) {...176 lines }

    @Override
    public void send(Trama trama, Device dev) {...86 lines }
}
```

Figura 8: Patrón GRASP: polimorfismo, clase HW\_Switch

- **Alta cohesión:** hace referencia a que las responsabilidades de una clase y la información que almacena, deben ser coherente y estar relacionadas con la clase. A continuación se muestra en la figura 9, su utilización en la clase IP.

```

public class IP implements Serializable{

    private int[] IP;

    public static boolean validate(String str_ip){...8 lines }

    public IP(Integer oct1, Integer oct2, Integer oct3, Integer oct4) {...7 lines }

    public IP() {...7 lines }

    public IP(String str_ip) {...7 lines }

    public IP(int[] IP) {...4 lines }

    public void setIP(int oct1, int oct2, int oct3, int oct4) {...6 lines }

    public void setIP(int[] newIP){...3 lines }

    public void setIP(String newIP){...6 lines }

    public String getIP_String() {...9 lines }

    public int[] getIP() {...3 lines }

    public IP AND(IP other){...10 lines }

    public IP clone() {...4 lines }

    public boolean equals(IP other) {...6 lines }

    @Override
    public String toString() {...3 lines }

}

```

Figura 9: Patrón GRASP: alta cohesión, clase IP

- **Bajo acoplamiento:** expone la idea de tener las clases lo menos ligadas entre sí que se pueda. De tal forma que en caso de producirse una modificación en alguna de ellas, se tenga la mínima repercusión posible en el resto. Este patrón fue aplicado en el diseño de las clases del sistema, potenciando la reutilización y disminuyendo la dependencia entre estas.

## 2.8 Patrones GoF.

Los patrones GoF (del inglés *Gang of Four*), describen las formas comunes en que diferentes tipos de objetos pueden ser organizados para trabajar unos con otros. Tratan la relación entre clases, la combinación

de clases y la formación de estructuras de mayor complejidad. Permiten crear grupos de objetos que ayudan a realizar tareas complejas [25].

En el desarrollo de la aplicación se hace uso del patrón *Singleton* en la clase `MacGenerator` (ver figura 10) para garantizar que sólo exista una instancia única de la clase, ya que el mismo está diseñado para restringir la creación de objetos pertenecientes a una clase o el valor de un tipo a un único objeto.

```
public class MacGenerator {
    ArrayList<String> list;
    private static MacGenerator instance = null;

    private MacGenerator(){
        list = new ArrayList<String>();
        for (int i = 16; i <= 254; i++){
            list.add(Integer.toHexString(i));
        }
    }

    public static MacGenerator getInstance() {
        if(instance == null) {
            instance = new MacGenerator();
        }
        return instance;
    }
}
```

Figura 10: Patrón GoF: Singleton, clase `MacGenerator`

## 2.9 Tarjetas CRC.

La metodología no requiere la representación del sistema informático mediante diagramas de clases utilizando notación UML, en su lugar se crean las tarjetas CRC que permiten trabajar con una metodología basada en objetos y posibilitan que el equipo completo contribuya en la tarea del diseño. Una tarjeta CRC está dividida en tres secciones, las cuales contienen la información del nombre de la clase, sus responsabilidades y colaboradores. Una clase describe cualquier objeto o evento, mediante los atributos y los métodos, las responsabilidades son las tareas que realizan o los métodos correspondientes a la clase. Los colaboradores son las demás clases con las que trabaja conjuntamente para cumplir con sus responsabilidades. A continuación se presentan algunas de las tarjetas CRC, el resto se encuentra en el Anexo 2.

**Tabla 14: Tarjeta CRC “Device”**

Device	
Responsabilidades	Colaboradores
Almacenar información genérica de los dispositivos	Connection
Definir imagen que represente al dispositivo	

**Tabla 15: Tarjeta CRC “PC”**

PC	
Responsabilidades	Colaboradores
Almacena información de las computadoras	Device
Definir dirección IP	IP
Definir dirección de la máscara de red	Hconsole
Definir dirección del <i>gateway</i>	AT_Console
Definir dirección MAC	ARP_Table
Definir tabla ARP	MacGenerator
Definir imagen que representa una computadora	PackageG
Enviar paquetes por la red	
Recibir paquetes de la red	
Analizar paquetes de red	



Tabla 16: Tarjeta CRC “IP”

IP	
Responsabilidades	Colaboradores
Almacenar direcciones IP versión cuatro	

### 2.10 Conclusiones parciales.

En el presente capítulo se describió la propuesta del sistema que se desea implementar, obteniéndose el plan de iteraciones como parte de la fase de planificación, dejándose claras las iteraciones por las cuales va a transitar la aplicación, señalando las historias de usuario que serán implementadas en cada una de ellas. Como parte del diseño del sistema se describieron las tarjetas CRC, las cuales representan cada clase del sistema y sus relaciones. Se definió el patrón arquitectónico y los patrones GRASP y GoF utilizados. Estos elementos permitieron al equipo de desarrollo planificar el tiempo necesario para la implementación de la solución propuesta y obtener una especificación más detallada de los requerimientos del cliente.

### 3. Capítulo 3.

En el presente capítulo se define el estándar de codificación empleado, así como las pruebas de aceptación efectuadas para la comprobación del correcto funcionamiento del sistema. También se realiza la validación de la propuesta, mediante el método de expertos Delphi.

#### 3.1 Estándar de codificación.

Un estándar de codificación completo comprende todos los aspectos de la generación de código. Un código fuente completo debe reflejar un estilo armonioso, como si un único programador hubiera escrito todo el código de una sola vez. Al comenzar un software, se establece un estándar de codificación para asegurar que todos los programadores del proyecto trabajen de forma coordinada. Cuando el proyecto de software incorpore código fuente previo, o bien cuando realice el mantenimiento de un sistema de software creado anteriormente, el estándar de codificación debería establecer cómo operar con la base de código existente.

Para el desarrollo de la herramienta se define el siguiente estándar de codificación.

**Clases, Interfaces y Enumeradores:** las iniciales en mayúscula. Las clases pertenecientes al paquete AT (Allied Telesyn) deben comenzar con “AT\_”. Las clases pertenecientes al paquete HW (Huawei) deben comenzar con “HW\_”.

Ejemplos:

```
public class MainWindow extends javax.swing.JFrame
```

```
public class AT_Switch extends Device
```

```
public class HW_Switch extends Device
```

```
public enum AT_EnumLevel
```

**Métodos:** deben ser verbos, la primera letra en minúscula y el resto de las palabras empiezan con mayúscula.

Ejemplo:

```
public void setConsole(AT_Console console)
```

**Variables:** comienzan con minúscula y puede utilizarse el elemento “\_” para concatenar nombres.

Ejemplos:

`private MainPanel panel`

`private AT_EnumLevel cli_level`

### 3.2 Pruebas.

Las pruebas del software son un elemento crítico para la garantía de la calidad del software. El objetivo de la etapa de pruebas es garantizar la calidad del producto desarrollado. Estas son utilizadas para identificar posibles fallos durante el proceso de desarrollo. En esta etapa se ejecutan pruebas a los componentes del software con el objetivo de medir el grado en que cumple con los requerimientos, se diseñan escenarios complejos donde se evalúan funciones específicas del software bajo condiciones controladas y resultados esperados.

#### 3.2.1 Pruebas de aceptación.

El objetivo de estas pruebas es verificar los requisitos, por este motivo, los propios requisitos del sistema son la principal fuente de información a la hora de construir las pruebas de aceptación. Estas son creadas a partir de las historias de usuario, las cuales pueden tener más de una prueba de aceptación, tantas como sean necesarias para garantizar su correcto funcionamiento y no se considera completa hasta que no supera sus pruebas de aceptación. Cada una de ellas representa una salida esperada del sistema. Es responsabilidad del cliente verificar la corrección de las pruebas de aceptación y tomar decisiones acerca de las mismas.

A continuación se muestran las pruebas de aceptación realizadas al sistema (ver tablas de la 17 a la 26).

**Tabla 17: Prueba de aceptación. Insertar dispositivos**

Caso de prueba de aceptación	
<b>Código:</b> HU1_P1	<b>Historia de usuario:</b> 1
<b>Nombre:</b> Insertar dispositivos.	

<b>Descripción:</b> comprobar la inserción de dispositivos.
<b>Condiciones de ejecución:</b>
<b>Entradas/Pasos de ejecución:</b> para añadir un dispositivo se selecciona el tipo de dispositivo a añadir desde la barra de herramientas y se selecciona el lugar donde se desea insertar en el área de trabajo.
<b>Resultado esperado:</b> los dispositivos son añadidos sin generar errores.
<b>Evaluación:</b> prueba satisfactoria.

Tabla 18: Prueba de aceptación. Conectar dispositivos

Caso de prueba de aceptación	
<b>Código:</b> HU1_P2	<b>Historia de usuario:</b> 1
<b>Nombre:</b> Conectar dispositivos.	
<b>Descripción:</b> comprobar la conexión entre dispositivos.	
<b>Condiciones de ejecución:</b> deben haber sido insertados al menos dos dispositivos.	
<b>Entradas/Pasos de ejecución:</b> para conectar dos dispositivos debe seleccionarse el tipo de conexión y al conectar cada dispositivo debe elegirse correctamente el puerto correspondiente al tipo de conexión.	
<b>Resultado esperado:</b> los dispositivos son conectados sin generar errores.	
<b>Evaluación:</b> prueba satisfactoria.	

Tabla 19: Prueba de aceptación. Nuevo proyecto

Caso de prueba de aceptación	
<b>Código:</b> HU2_P3	<b>Historia de usuario:</b> 2
<b>Nombre:</b> Nuevo proyecto.	
<b>Descripción:</b> comprobar la creación de un proyecto de trabajo.	
<b>Condiciones de ejecución:</b>	
<b>Entradas/Pasos de ejecución:</b> para crear un nuevo proyecto se selecciona la opción "Nuevo".	
<b>Resultado esperado:</b> se crea directamente un nuevo proyecto con el área de trabajo en blanco.	
<b>Evaluación:</b> prueba satisfactoria.	

Tabla 20: Prueba de aceptación. Guardar proyecto

Caso de prueba de aceptación	
<b>Código:</b> HU2_P4	<b>Historia de usuario:</b> 2
<b>Nombre:</b> Guardar proyecto.	
<b>Descripción:</b> comprobar la funcionalidad de guardar un proyecto de trabajo.	
<b>Condiciones de ejecución:</b> existen cambios en el proyecto de trabajo.	
<b>Entradas/Pasos de ejecución:</b> para guardar un proyecto se selecciona la opción "Guardar".	

<p><b>Resultado esperado:</b> si el proyecto ha sido guardado anteriormente se guardan los cambios en el archivo correspondiente, en caso contrario se muestra una ventana para seleccionar la dirección y el nombre del archivo donde se guardarán los cambios.</p>
<p><b>Evaluación:</b> prueba satisfactoria.</p>

**Tabla 21: Prueba de aceptación. Abrir proyecto**

Caso de prueba de aceptación	
<b>Código:</b> HU2_P5	<b>Historia de usuario:</b> 2
<b>Nombre:</b> Abrir proyecto.	
<b>Descripción:</b> comprobar la funcionalidad de abrir un proyecto de trabajo.	
<b>Condiciones de ejecución:</b> existen un proyecto previamente guardado.	
<b>Entradas/Pasos de ejecución:</b> para abrir un proyecto se selecciona la opción “Abrir” y luego se selecciona el archivo de proyecto.	
<b>Resultado esperado:</b> si el archivo seleccionado no es un archivo de proyecto, se le informa al usuario mediante una ventana de error, en caso contrario el proyecto es abierto, mostrando la información correspondiente.	
<b>Evaluación:</b> prueba satisfactoria.	

**Tabla 22: Prueba de aceptación. Configurar PC**

Caso de prueba de aceptación	
<b>Código:</b> HU3_P6	<b>Historia de usuario:</b> 3

<b>Nombre:</b> Configurar PC.
<b>Descripción:</b> comprobar la configuración de las propiedades de una PC.
<b>Condiciones de ejecución:</b> existen un dispositivo de tipo PC previamente creado.
<b>Entradas/Pasos de ejecución:</b> para configurar un PC se hace doble <i>click</i> sobre la PC a configurar, luego se añaden o modifican las propiedades de la misma y se da <i>click</i> en el botón aceptar para guardar los cambios.
<b>Resultado esperado:</b> las propiedades de la PC son cambiadas satisfactoriamente.
<b>Evaluación:</b> prueba satisfactoria.

Tabla 23: Prueba de aceptación. Configurar switch

Caso de prueba de aceptación	
<b>Código:</b> HU4_P7	<b>Historia de usuario:</b> 4
<b>Nombre:</b> Configurar <i>switch</i> .	
<b>Descripción:</b> comprobar la configuración de los <i>switches</i> .	
<b>Condiciones de ejecución:</b> existe una PC conectada a un <i>switch</i> mediante una conexión serial.	
<b>Entradas/Pasos de ejecución:</b> para configurar un <i>switch</i> se hace doble <i>click</i> sobre la PC, se establecen los parámetros de conexión, se hace <i>click</i> en el botón “Ok” y posteriormente se realizan las configuraciones mediante líneas de comando.	
<b>Resultado esperado:</b> si los parámetros de conexión establecidos son incorrectos se le informa al usuario mediante una ventana de error. Al cerrar la conexión se guardan	

satisfactoriamente las configuraciones en el <i>switch</i> .
<b>Evaluación:</b> prueba satisfactoria.

Tabla 24: Prueba de aceptación. Mostrar información de los dispositivos

Caso de prueba de aceptación	
<b>Código:</b> HU5_P8	<b>Historia de usuario:</b> 5
<b>Nombre:</b> Mostrar información de los dispositivos.	
<b>Descripción:</b> comprobar la funcionalidad “Mostrar información de los dispositivos”.	
<b>Condiciones de ejecución:</b> existe al menos un dispositivo en el área de trabajo.	
<b>Entradas/Pasos de ejecución:</b> se selecciona el dispositivo del cual se quiere obtener la información.	
<b>Resultado esperado:</b> se muestra la información del dispositivo seleccionado.	
<b>Evaluación:</b> prueba satisfactoria.	

Tabla 25: Prueba de aceptación. Crear evaluación

Caso de prueba de aceptación	
<b>Código:</b> HU6_P9	<b>Historia de usuario:</b> 6
<b>Nombre:</b> Crear evaluación.	
<b>Descripción:</b> comprobar la funcionalidad “Crear evaluación”.	
<b>Condiciones de ejecución:</b>	



<b>Entradas/Pasos de ejecución:</b> se crea una red, se configura y se selecciona la opción “Crear evaluación”; luego se modifican los elementos de la evaluación, se añaden las preguntas y se guarda la evaluación en un fichero de evaluación.
<b>Resultado esperado:</b> se crea correctamente la evaluación.
<b>Evaluación:</b> prueba satisfactoria.

**Tabla 26: Prueba de aceptación. Realizar evaluación**

Caso de prueba de aceptación	
<b>Código:</b> HU7_P10	<b>Historia de usuario:</b> 7
<b>Nombre:</b> Realizar evaluación.	
<b>Descripción:</b> comprobar la funcionalidad “Realizar evaluación”.	
<b>Condiciones de ejecución:</b> tener abierto un archivo de evaluación.	
<b>Entradas/Pasos de ejecución:</b> se selecciona la opción “Evaluar”.	
<b>Resultado esperado:</b> se muestra una ventana con los resultados de la evaluación.	
<b>Evaluación:</b> prueba satisfactoria.	

### 3.2.2 Resultado de las pruebas

Se procede al análisis de los resultados de las pruebas ejecutadas al sistema. En la tabla 27 se exponen el número de No-Conformidades (NC) por iteración, se especifican las que fueron corregidas, las que no procedieron (NP) por estar condicionadas por factores ajenos al desarrollo de la aplicación, así como las válidas.

Tabla 27: Resultados de las pruebas al sistema

Iteración	NC	NP	Válidas	Corregidas
1	27	8	19	13
2	12	0	12	12
3	2	1	1	1
<b>Total</b>	<b>41</b>	<b>9</b>	<b>32</b>	<b>26</b>

Luego se procede a aplicar la métrica de Densidad de Defectos (DD) para determinar si se desarrolló con calidad el sistema que se propone. La calidad del sistema es inversamente proporcional a su densidad de defectos. Esta métrica es utilizada para establecer la proporción de defectos con respecto al tamaño del producto y se calcula de la siguiente manera:

$$DD = TP/DT$$

Siendo DD densidad de defectos, DT defectos totales encontrados en el producto y TP tamaño del producto (dígase en líneas de código).

$$DD = 32/480000$$

$$DD = 0.00007$$

Tras conocer que el resultado de la medición es aproximadamente 0.00007, puede confirmarse cuán cerca de cero se encuentra el indicador y por tanto puede decirse que el software presenta gran calidad en su implementación.

### 3.3 Validación de la propuesta de solución.

Para validar la factibilidad de la herramienta propuesta se decidió utilizar el método de expertos Delphi, debido a que este método permite con su aplicación tener estimaciones razonablemente buenas y las mejores suposiciones a falta de cifras más exactas. El método Delphi es aceptable en la comunidad científica, tiene una elevada madurez, proporciona estimaciones de precisión moderada y no exige que se reúna a los expertos en un lugar determinado [26].

El método Delphi consiste en la selección de un grupo de expertos a los que se les pregunta su opinión sobre cuestiones referidas a acontecimientos del futuro. Este procedimiento extrae y maximiza las ventajas que presentan los métodos basados en grupos de expertos y minimiza sus inconvenientes. Este método consta de cuatro fases: [26]

- **Formulación del problema:** se trata de una etapa fundamental en la realización de un Delphi. En un método de expertos, la importancia de definir con precisión el campo de investigación es muy grande por cuanto que es preciso estar muy seguros de que los expertos reclutados y consultados poseen todas las mismas nociones de este campo. La elaboración del cuestionario debe ser llevada a cabo según ciertas reglas: las preguntas deben ser precisas, cuantificables e independientes.
- **Elección de expertos:** la etapa es importante puesto que el término de experto es ambiguo. Con independencia de sus títulos, su función o su nivel jerárquico, el experto será elegido por su capacidad de encarar el futuro y los conocimientos que posea sobre el tema consultado. La falta de independencia de los expertos puede constituir un inconveniente; por esta razón los expertos son aislados y sus opiniones son recogidas por vía postal o electrónica y de forma anónima; así pues se obtiene la opinión real de cada experto y no la opinión más o menos falseada por un proceso de grupo (se trata de eliminar el efecto de los líderes).
- **Elaboración y lanzamiento de los cuestionarios (en paralelo con la fase de elección de expertos):** los cuestionarios se elaboran de manera que faciliten (en la medida en que una investigación de estas características lo permite) la respuesta por parte de los consultados.
- **Desarrollo práctico y explotación de resultados:** el cuestionario es enviado a cierto número de expertos (hay que tener en cuenta las no-respuestas y abandonos). Naturalmente el cuestionario va acompañado por una nota de presentación que precisa las finalidades, el espíritu del Delphi, así como las condiciones prácticas del desarrollo de la encuesta (plazo de respuesta, garantía de anonimato). Luego se procede al estudio de resultados para llegar a conclusiones.

Este método presenta tres características fundamentales: [26]

- **Anonimato:** durante un Delphi, ningún experto conoce la identidad de los otros que componen el grupo de debate. Esto tiene una serie de aspectos positivos, como son:
  - Impide la posibilidad de que un miembro del grupo sea influenciado por la reputación de otro

de los miembros o por el peso que supone oponerse a la mayoría. La única influencia posible es la de la congruencia de los argumentos.

- Permite que un miembro pueda cambiar sus opiniones sin que eso suponga una pérdida de imagen.
- El experto puede defender sus argumentos con la tranquilidad que da saber que en caso de que sean erróneos, su equivocación no va a ser conocida por los otros expertos.
- Iteración y realimentación controlada: la iteración se consigue al presentar varias veces el mismo cuestionario. Como, además, se van presentando los resultados obtenidos con los cuestionarios anteriores, se consigue que los expertos vayan conociendo los distintos puntos de vista y puedan ir modificando su opinión si los argumentos presentados les parecen más apropiados que los suyos.
- Respuesta del grupo en forma estadística: la información que se presenta a los expertos no es sólo el punto de vista de la mayoría, sino que se presentan todas las opiniones indicando el grado de acuerdo que se ha obtenido.

Sus desventajas más significativas están dadas en que: [26]

- Es muy laborioso y demanda tiempo su aplicación.
- Es costoso en comparación con otros, ya que requiere del empleo de: tiempo de los expertos, hojas, impresoras, teléfono, correo, etcétera.

Por otra parte, este método tiene como ventajas que presenta amplias posibilidades de aplicación a gran número de situaciones o casos y su sencillez de uso cuando se dispone de expertos bien informados y dispuestos a colaborar. Se puede utilizar en situaciones que comprendan tareas y actividades simples, actividades basadas en procedimientos, tareas de diagnóstico y de control de procesos, considerando el impacto de los factores de tipo socio-técnico sobre la conducta. [26]

### **3.4 Aplicación del método Delphi.**

El método Delphi se basa en una secuencia de pasos que a continuación se mencionan:

- Selección de los expertos.
- Cálculo del coeficiente de concordancia de Kendall.

- Cálculo de las frecuencias absolutas.
- Cálculo de las frecuencias absolutas acumuladas.
- Cálculo de las frecuencias relativas acumuladas.
- Cálculo de los puntos de corte.

Todos estos pasos serán descritos a continuación para mostrar los resultados que fueron arrojados luego de llevar a cabo la aplicación del método para validar la factibilidad de la propuesta de solución.

### 3.4.1 Selección de los expertos

Según el método Delphi para que una persona pueda ser etiquetada como experta debe poseer un conocimiento profundo de la tarea o actividad que será objeto de análisis y valoración y tiene que estar familiarizada con el sistema en el que ella se desarrolla. Asimismo, los expertos a los que se recurre tienen que ser capaces de traducir su valoración en términos de probabilidades. [26]

Para la validación de la propuesta, se han escogido ocho expertos, se señala que es necesario escoger a siete como mínimo, debido a que el error disminuye notablemente por cada experto añadido hasta llegar a los siete y como máximo treinta pues la mejora en la previsión es muy pequeña y normalmente el incremento en costo y trabajo de investigación no compensa la mejora. Existen características propias que poseen cada uno de los expertos seleccionados y que se tuvieron en cuenta para conformar el panel.

- Graduado de nivel superior.
- Experiencia en el tema de redes de comunicación.
- Experiencia docente en el tema de redes de comunicación.
- Honestidad.
- Disposición en participar en la encuesta.
- Capacidad de análisis y de pensamiento.

El objetivo de este proceso es validar la factibilidad de la solución propuesta, mediante la aplicación de cuestionarios a la muestra seleccionada. Esta selección se lleva a cabo, teniendo en cuenta la experiencia y capacitación que presenta el experto sobre el tema. Las respuestas del panel seleccionado pueden

contribuir a perfeccionar la herramienta propuesta. Se seleccionan ocho expertos de los nueve encuestados. Para la selección de los expertos es útil emplear la valoración por competencias mediante un formulario de autovaloración (ver Anexo 3). Este método consiste en calcular el coeficiente de competencia (K) a partir de su conocimiento o información sobre el tema ( $K_c$ ) y el coeficiente de argumentación o valoración ( $K_a$ ), mediante la siguiente fórmula:  $K = 1/2 (K_c + K_a)$ . La interpretación de los coeficientes de competencias es la siguiente:

- Si  $0,8 < K \leq 1,0$  coeficiente de competencia alto.
- Si  $0,5 < K \leq 0,8$  coeficiente de competencia medio.
- Si  $K \leq 0,5$  coeficiente de competencia bajo.

Para determinar el coeficiente de conocimiento o información ( $K_c$ ) el experto marcará en la casilla enumerada (ver Anexo 3), según su criterio acerca de la capacidad que él tiene sobre el tema que se la ha sometido a su consideración, en una escala del 0 al 10 y que después para ajustarla a la teoría de las probabilidades se multiplicará por 0,1.

Para determinar el coeficiente de argumentación o valoración ( $K_a$ ), se ofrece una tabla con cierta información (ver Anexo 3). El experto debe marcar, según su criterio, los elementos que le permiten argumentar su evaluación del nivel de conocimiento seleccionado anteriormente. Las marcas de los expertos se traducen a puntos teniendo en cuenta la escala que se muestra en el Anexo 4. El resultado del coeficiente de competencias de todos los encuestados se muestra en el Anexo 5.

De los nueve expertos encuestados uno posee un coeficiente de competencia bajo, cuatro un coeficiente de competencia medio y cuatro lo tienen alto. Esto determinó en la decisión de que sólo ocho hayan sido incluidos en el grupo de expertos para la validación de la propuesta. En la tabla 83 se muestran los nombres de los expertos que colaboraron con la validación de la propuesta de solución (ver Anexo 6).

#### **3.4.2 Elaboración del cuestionario.**

Después de determinar el coeficiente de competencia de los expertos se procede a realizar el cuestionario para validar la propuesta de solución. Para la elaboración del cuestionario se tuvieron en cuenta los principios básicos que debería cumplir la propuesta presentada para su utilización como medio de enseñanza-aprendizaje en la UCI, el mismo se muestra en el Anexo 7.

### 3.4.3 Verificar la concordancia entre los expertos.

El coeficiente de concordancia de Kendall se obtiene aplicando la fórmula:

$$W = \frac{12 \cdot \Delta^2}{m^2(k^3 - k)}$$

W: coeficiente de concordancia.

k: cantidad de características o número de índices a evaluar.

m: número de expertos que emiten criterio.

$$\Delta^2 = \sum_{i=1}^m (A_i - T)^2$$

T: factor de concordancia.

A<sub>i</sub>: ponderación total de todas las características según el experto i.

$$T = \frac{1}{K} \left( \sum_{j=1}^k \sum_{i=1}^m A_{ij} \right); \quad \sum_{i=1}^m A_i \leq T$$

A<sub>ij</sub>: ponderación sobre el criterio de la característica o variable i, según el experto j.

Se le asignan valores a las categorías (Muy Adecuado (5), Bastante Adecuado (4), Adecuado (3), Poco Adecuado (2), Nada Adecuado (1)). En dependencia de la evaluación que el experto de a cada pregunta será el valor asociado que se pondrá en la tabla 84 mostrada en el Anexo 8.

Se tiene:

$$m = 8$$

$$k = 10$$

Obteniendo:

$$T = 36.3$$

$$\Delta^2 = 808.7$$

$$\mathbf{W = 0.153}$$

W expresa el grado de concordancia entre los ocho expertos al dar un orden evaluativo a las preguntas sometidas a valoración. Este coeficiente siempre será positivo y estará comprendido entre los valores de cero y uno.

### Chi cuadrado real

$$X^2_{\text{real}} = m * (k - 1) * W$$

$$X^2_{\text{real}} = 11.03$$

Si  $X^2_{\text{real}} < X^2_{(\alpha, k-1)}$ , entonces existe concordancia en el trabajo de los expertos, sino es necesario repetir el trabajo de expertos.

$$X^2_{(\alpha, k-1)} = 14.6837$$

$\alpha = 0.1$  para un nivel de confianza del 90%.

Como  $11.03 < 14.6837$ , se puede concluir que hay concordancia entre los expertos.

#### 3.4.4 Desarrollo práctico y explotación de los resultados.

Tabulados los datos, se realizan los siguientes pasos para obtener los resultados deseados:

- **Primer paso:** construir la tabla de frecuencias absolutas (ver Anexo 9) y a partir de esta construir la tabla de frecuencias absolutas acumuladas. Esto se hace por fila, excepto el valor de la primera columna de esa fila, las restantes se obtienen sumando el actual y el anterior. En la frecuencia absoluta acumulada desaparece la última columna (ver Anexo 10).
- **Segundo paso:** construir la tabla de frecuencias relativas acumuladas, para lo cual, se divide el valor de cada celda de la tabla anterior entre el número de expertos consultados, en este caso ocho (ver Anexo 11).
- **Tercer paso:** se buscan las imágenes de los elementos de la tabla de frecuencias relativas



acumuladas (ver Anexo 11) por medio de la función Distribución Normal Estándar. A la misma tabla se le adicionan tres columnas y una fila para colocar los resultados que se mencionan a continuación.

- Suma de filas.
- Promedio de las columnas.
- Los promedios de las filas se obtienen dividiendo por cinco (cantidad de categorías) la suma de las filas.
- Para hallar N, se calcula la sumatoria de las sumas de las filas y se divide entre el resultado de multiplicar el número de categorías por el número de preguntas.
- N-P da el valor promedio que otorgan los expertos para cada categoría propuesta.

Los puntos de corte se utilizan para determinar la categoría de cada criterio según la opinión de los expertos consultados. Con ellos se opera como se muestra a continuación:

- Muy adecuado: si  $N-P < 0.12$ .
- Bastante adecuado: si  $0.12 \leq N-P < 1.04$ .
- Adecuado: si  $1.04 \leq N-P < 2.08$ .
- Poco adecuado: si  $2.08 \leq N-P < 2.29$ .
- Nada adecuado: si  $N-P \geq 2.29$ .

Obteniéndose los resultados mostrados en el Anexo 12.

Después de analizar los resultados de las encuestas se comprueba la factibilidad de la propuesta de solución, concluyéndose que el 50% de los expertos está de acuerdo en que el simulador de redes propuesto como medio de enseñanza-aprendizaje, es muy adecuado y el otro 50% coincide en que es bastante adecuado.

### **3.5 Conclusiones parciales.**

En el presente capítulo se especificó el estándar de codificación que abarca la estructura que tendrá el código del sistema, permitiendo una mejor comprensión de este por parte de los desarrolladores del sistema. Por otra parte se realizaron las pruebas de aceptación que garantizan el cumplimiento de las necesidades

del cliente reflejadas en las historias de usuario definidas. También, con la utilización del método de expertos Delphi, se logró validar la factibilidad de la utilización del simulador propuesto como un medio de enseñanza-aprendizaje.

#### 4. Conclusiones generales.

Al término de la presente investigación, sobre el desarrollo de un simulador de redes como medio de enseñanza-aprendizaje, para contribuir al desarrollo profesional de los estudiantes de la UCI en la habilidad de configurar equipos de interconexión de redes, se puede concluir que el objetivo general fue cumplido, alcanzándose los siguientes resultados:

- Con la investigación realizada, se pudo demostrar que un simulador de redes puede ser utilizado como un medio de enseñanza-aprendizaje.
- La metodología de desarrollo de software XP, en conjunto con el entorno tecnológico utilizado, permitieron agilizar y planificar correctamente el proceso de desarrollo de la solución propuesta.
- El resultado satisfactorio de las pruebas de aceptación aseguró la calidad y la aprobación por parte del cliente del simulador de redes desarrollado.
- La herramienta desarrollada simula los dispositivos de interconexión de capa 3 (Huawei serie 5600) y de capa 2 (Allied Telesyn serie 8500); facilita la configuración de estos disponiendo de una ayuda proactiva; diseña segmentos de redes de computadoras en un entorno gráfico; y presenta un mecanismo de evaluación que permite calificar el desempeño del usuario. Aspectos que validan su utilización como un medio de enseñanza-aprendizaje, contribuyendo al desarrollo profesional de los estudiantes de la UCI en la habilidad de configurar equipos de interconexión de redes.

## 5. Recomendaciones.

Para garantizar el perfeccionamiento progresivo de la herramienta de simulación de redes desarrollada, se recomienda:

- Agregar más comandos de configuración a los dispositivos.
- Implementar un monitor de paquetes.
- Agregar dispositivos de capa 2 y capa 3 de ambas tecnologías.
- Permitir crear perfiles de usuarios.
- Diseñar una vista web que permita integrar la herramienta con el EVA.

**6. Referencias bibliográficas.**

- [1] J. I. Santos, J. M. Galán, L. R. Izquierdo y R. del Olmo, «Aplicaciones de las TIC en el nuevo modelo de enseñanza del EEES,» nº 39, 2009.
- [2] C. Rosales, Didáctica. Núcleos fundamentales, Madrid, 1988.
- [3] I. Matte Urrejola, «T13,» 2014. [En línea]. Available: <http://www.t13.cl/videos/t13-central/inauguran-simulador-clinico-en-universidad-de-valparaiso>. [Último acceso: 11 2014].
- [4] I. Pérez Juste, «EXPERIMENTACION EN QUIMICA FISICA,» 29 06 2010. [En línea]. Available: [http://eqf\\_web.webs.uvigo.es/eqf\\_programa.htm](http://eqf_web.webs.uvigo.es/eqf_programa.htm). [Último acceso: 11 2014].
- [5] E. Martínez Zornoza, Aprendizaje con Simuladores. Aplicación a las Redes de Comunicaciones.
- [6] A. J. Colom Cañellas, J. Sureda Negre y J. Salinas Ibáñez, Tecnología y medios educativos, Barcelona: Cincel-Kapelusz, 1988.
- [7] P. Marqués Graells, «LOS MEDIOS DIDÁCTICOS,» 07 08 2011. [En línea]. Available: <http://www.peremarques.net/medios2.htm>. [Último acceso: 2015].
- [8] Colectivo de autores, Preparación Pedagógica Integral para profesores universitarios, Vedado, Ciudad Habana: Felix Varela, 2003.
- [9] S. Asmussen y P. W. Glynn, Stochastic Simulation: Algorithms and Analysis, New York: Springer-Verlag, 2007.
- [10] F. J. Media Robles, Redes de área local 2, Madrid, España: Ra-MA, 2006.
- [11] J. J. Grimaldos, «I.E.S. "Cura Valera". Departamento de Informática,» 2013. [En línea]. Available: <http://informatica.iescuravalera.es/iflica/gtfinal/libro/index.html>. [Último acceso: 1 2015].
- [12] L. Alegsa, «ALEGSA,» 2015. [En línea]. Available: <http://www.alegsa.com.ar/Dic/medio%20de%20transmision%20de%20datos.php>. [Último acceso: 1 2015].
- [13] D. E. Comer, Redes Globales de Informacaión con internet y TCP/IP. Principios Basicos, Protocolos y Arquitectura, Juárez, México: PRENTICE-HALL HISPANOAMERICANA, S. A., 1996.
- [14] W. E. Castellanos Hernández y M. E. Chacón Osorio, «UTILIZACIÓN DE HERRAMIENTAS SOFTWARE PARA EL MODELADO Y LA SIMULACIÓN DE REDES DE COMUNICACIONES,» Colombia.

- [15] «eNSP-Huawei Products,» 2014. [En línea]. Available: <http://enterprise.huawei.com/en/products/network-management/automation-tools/tools/hw-201999.htm>. [Último acceso: 1 2015].
- [16] «Netbeans,» 2015. [En línea]. Available: [https://netbeans.org/index\\_es.html](https://netbeans.org/index_es.html). [Último acceso: 1 2015].
- [17] T. de la Rosa Martín y D. Galpert Cañizares, «Revista Cubana de Ciencias Informáticas,» 2013. [En línea]. Available: [http://scielo.sld.cu/scielo.php?pid=S2227-18992013000300002&script=sci\\_arttext](http://scielo.sld.cu/scielo.php?pid=S2227-18992013000300002&script=sci_arttext). [Último acceso: 1 2015].
- [18] N. Meyeres, Java Programming on Linux, Waite Group Press, 2000.
- [19] P. Letelier y M. C. Penadés, Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP), Valencia.
- [20] J. H. Canós, P. Letelier y M. C. Penadés, «Metodologías Ágiles en el Proceso de Desarrollo de Software.,» 2003.
- [21] Unidad I Requerimientos de Software Unidad Curricular Ingeniería de Software II. Módulo: Fundamentos de Ingeniería de requisitos y análisis.
- [22] J. Joskowicz, Reglas y Prácticas, España, 2008.
- [23] P. Grenyer, «Model View Controller with Java Swing,» 12 2008. [En línea]. Available: <http://accu.org/index.php/journals/1524>. [Último acceso: 4 2015].
- [24] C. Larman, UML y Patrones: Una introducción al análisis y diseño orientado a objetos y el proceso unificado, Texas: Prentice Hall, 1997.
- [25] E. Gamma, R. Helm, R. Johnson y J. Vlissides, Design Patterns: Elements of reusable Object-Oriented Software, Addison-Wesley Profesional Computer Series, 1994.
- [26] C. R. Trujillo, Casos exitosos de forecasting en Colombia. Empaques flexibles y semi rígidos en, 2004.
- [27] H. Facundo Arena, La Biblia de Linux., Manual de uso, instalacion y configuracion., Buenos Aires, Argentina: MP Ediciones, 2002.
- [28] J. Byous, Java technology: The early years., 2005.
- [29] R. Mesa Cabeza, «ClubEnsayos,» 29 11 2013. [En línea]. Available: <http://www.clubensayos.com/Informes-De-Libros/UML-CFE-COMPLETO/1314345.html>. [Último acceso: 1 2015].

## 7. Bibliografía.

- [1] J. I. Santos, J. M. Galán, L. R. Izquierdo y R. del Olmo, «Aplicaciones de las TIC en el nuevo modelo de enseñanza del EEES,» nº 39, 2009.
- [2] C. Rosales, Didáctica. Núcleos fundamentales, Madrid, 1988.
- [3] E. Martínez Zornoza, Aprendizaje con Simuladores. Aplicación a las Redes de Comunicaciones.
- [4] A. J. Colom Cañellas, J. Sureda Negre y J. Salinas Ibáñez, Tecnología y medios educativos, Barcelona: Cincel-Kapelusz, 1988.
- [5] Colectivo de autores, Preparación Pedagógica Integral para profesores universitarios, Vedado, Ciudad Habana: Felix Varela, 2003.
- [6] S. Asmussen y P. W. Glynn, Stochastic Simulation: Algorithms and Analysis, New York: Springer-Verlag, 2007.
- [7] F. J. Media Robles, Redes de área local 2, Madrid, España: Ra-MA, 2006.
- [8] D. E. Comer, Redes Globales de Informacaión con internet y TCP/IP. Principios Basicos, Protocolos y Arquitectura, Juárez, México: PRENTICE-HALL HISPANOAMERICANA, S. A., 1996.
- [9] W. E. Castellanos Hernández y M. E. Chacón Osorio, «UTILIZACIÓN DE HERRAMIENTAS SOFTWARE PARA EL MODELADO Y LA SIMULACIÓN DE REDES DE COMUNICACIONES,» Colombia.
- [10] N. Meyeres, Java Programming on Linux, Waite Group Press, 2000.
- [11] P. Letelier y M. C. Penadés, Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP), Valencia.
- [12] J. H. Canós, P. Letelier y M. C. Penadés, «Metodologías Ágiles en el Proceso de Desarrollo de Software.,» 2003.
- [13] Unidad I Requerimientos de Software Unidad Curricular Ingeniería de Software II. Módulo: Fundamentos de Ingeniería de requisitos y análisis.
- [14] J. Joskowicz, Reglas y Prácticas, España, 2008.
- [15] C. Larman, UML y Patrones: Una introducción al análisis y diseño orientado a objetos y el proceso unificado, Texas: Prentice Hall, 1997.
- [16] E. Gamma, R. Helm, R. Johnson y J. Vlissides, Design Patterns: Elements of reusable Object-

Oriented Software, Addison-Wesley Profesional Computer Series, 1994.

- [17] C. R. Trujillo, Casos exitosos de forecasting en Colombia. Empaques flexibles y semi rígidos en, 2004.
- [18] H. Facundo Arena, La Biblia de Linux., Manual de uso, instalacion y configuracion., Buenos Aires, Argentina: MP Ediciones, 2002.
- [19] J. Byous, Java technology: The early years., 2005.