

**SOFTWARE PARA APOYAR EL PROCESO ENSEÑANZA-APRENDIZAJE DE  
LOS PROTOCOLOS DE COMUNICACIÓN TCP E IP.**

**RICARDO DELGADO MARTINEZ  
DIEGO ALEJANDRO VILLA RENTERIA**

Proyecto de grado presentado como requisito para optar al título de  
Ingeniero de Sistemas

**Director  
Ing. Hernán Porras Díaz M. Sc., Ph. D.**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER  
FACULTAD DE INGENIERIAS FÍSICO MECÁNICAS  
ESCUELA DE INGENIERIA DE SISTEMAS E INFORMÁTICA  
BUCARAMANGA  
2004**

## DEDICATORIA

*Así como grano a grano de arena se forman las playas, como hoja a hoja se forman los árboles, así con esfuerzos e ilusiones culmina este recorrido que me llevo a formarme como persona y ser humano.*

*Doy gracias a Dios por haberme permitido llegar hasta aquí por que sin su compañía nada sería posible, se que en el esfuerzo de mis padres, el amor de mi madre, los consejos de mi padre y en el apoyo de mi hermano esta el espíritu de Dios alentando mi camino.*

*Agradezco a los profesores y compañeros de la universidad, a aquellos que considero como mis amigos por haberme acompañado en este camino que lleno de espinas y piedras pero que nos lleva a un mundo donde la formación y el conocimiento son una puerta abierta al éxito personal y profesional*

*Ricardo Delgado Martínez*

*Primero que todo le doy gracias a Dios, por iluminarme el camino, por darme parte de su infinita sabiduría para tomar las decisiones claves en mi vida y por darme la paciencia necesaria para sobrellevar los momentos en los que pareciera que las metas trazadas no se pudieran lograr.*

*Doy gracias a mis tíos, Manuel, Jairo y Sigifredo Renteria, que sin importar el orden en que los mencione contribuyeron en igual medida a mi formación académica y sin darse cuenta se convirtieron en mi modelo a seguir como hijo, hermano y persona.*

*A mis padres Clara y Antonio y a mis hermanas Sonia y Farides, por brindarme su apoyo, por enseñarme a valorar las cosas simples de la vida y sobre todo por creer en mí.*

*A Manuel Renteria, a su esposa Olga Reyes y sus hijos Adriana, Rosibel y Freddy, gracias por aceptarme en su hogar y permitirme ser miembro activo de su hermosa familia.*

*Resulta difícil no darle crédito a mis compañeros y amigos Milena, Flor, Nini, Yehison y Ricardo los cuales me acompañaron en este largo camino, me brindaron su confianza y me ayudaron a crecer como persona.*

*Gracias a los compañeros de carrera que sin tener lazos fuertes de cercanía en el transcurso de la carrera, estuvieron prestos a brindar un intercambio constructivo de ideas que permitieron mejorar aspectos conceptuales y de diseño de este proyecto.*

*Un agradecimiento muy especial para mi amigo y compañero de proyecto Ricardo Delgado y a su familia, por la paciencia y comprensión en este camino que parecía interminable pero que finalmente logramos finalizar.*

*Diego Alejandro Villa*

*Este proyecto es el resultado del gran esfuerzo de muchas personas, que directa o indirectamente, aportaron un grano de arena para que estuviéramos en este punto.*

*El resultado obtenido esta dedicado a nuestros padres y familiares que con sus sacrificios y su confianza nos permitieron lograr este objetivo; a nuestros amigos que enriquecieron nuestro conocimiento y nos permitieron conocer sus puntos de vista para plasmarlos en este proyecto.*

## CONTENIDO

	Pág.
INTRODUCCIÓN	
PARTE1: FUNDAMENTOS	3
CAPITULO 1: PRESENTACIÓN	3
1.1 PRESENTACIÓN DEL INFORME	3
1.2 DESCRIPCIÓN DEL PROYECTO	4
1.2.1 Objetivo General	4
1.2.2 Objetivos específicos	5
1.3 JUSTIFICACIÓN	6
1.3.1 Antecedentes y descripción del problema	6
1.3.2 Descripción de los objetivos	10
1.3.3 Impacto	11
1.3.4 Viabilidad	11
CAPITULO 2: MARCO TEÓRICO	13
2.1 GENERALIDADES DE INTERNET	13
2.1.1 World Wide Web –WWW	14
2.1.2 Exploradores y servidores web	15
2.2 SOFTWARE EDUCATIVO	16
2.2.1 Tipos de software educativo	16
2.2.2 Elaboración de software educativo	17

2.3	ARQUITECTURA DE INFORMACIÓN	18
2.3.1	Modelo Cliente/Servidor	18
2.4	PÁGINAS WEB ESTÁTICAS	19
2.4.1	HTML(Hiper Tex Markup Language)	20
2.5	PÁGINAS WEB DINÁMICAS EN EL CLIENTE	20
2.5.1	HTML Dinámico (DHTML)	21
2.5.2	Lenguajes Scripts	22
2.5.3	Applets de JAVA	24
2.6	PROTOCOLOS DE INTERNET	25
2.6.1	Modelo de capas TCP/IP	25
2.6.2	SMTP	31
2.6.3	Fragmentación, reensamblaje de datagramas IP	34
2.6.4	Direccionamiento IP	38
2.6.4.1	División En Subredes	40
2.6.5	Transferencia de datos de TCP	41
2.6.6	Encaminamiento de data gramas IP	48
2.6.7	Detección de errores	53
2.6.8	Control de errores	60
2.6.9	Control de flujo	65
	CAPITULO 3: MARCO METODOLÓGICO	72
3.1	CICLOS DE VIDA DE DESARROLLO DE SOFTWARE	72
3.2	PROCESO UNIFICADO	72

3.3 LENGUAJE UNIFICADO DE MODELADO (UML)	75
3.3.1 Modelado de software con UML	75
3.4 SELECCIÓN DEL MODELO DE DESARROLLO	76
PARTE 2: DESARROLLO DEL SISTEMA	78
CAPITULO 4: FASE DE INICIO	78
4.1 ALCANCE DEL PROYECTO	78
4.1.1 Objetivos de la fase de inicio	79
4.2 IDENTIFICACION DE ACTORES	80
4.2.1 Requisitos candidatos del sistema	82
4.3 DESCRIPCIÓN DE MÓDULOS DE LA HERRAMIENTA	82
4.3.1 Descripción del módulo Documentación	82
4.3.2 Descripción del módulo Ayuda	83
4.3.3 Descripción del módulo SEEPINTmail	83
4.3.4 Descripción del módulo Laboratorios	84
4.3.5 Descripción del módulo Buzón	85
4.4 ARQUITECTURA CANDIDATA	86
4.5 RIESGOS DEL PROYECTO	87
4.6 COSTOS DEL PROYECTO	89
4.7 AGENDA DEL PROYECTO	89
4.8 EVALUACIÓN DE LA FASE	90
CAPITULO 5: FASE DE ELABORACIÓN	91
5.1 GENERALIDADES	91
5.1.1 Planeación	91

5.1.2	Flujos, actividades y productos a entregar	92
5.2	CAPTURA DE REQUISITOS	92
5.3	ANÁLISIS	92
5.3.1	Análisis de la arquitectura	93
5.3.2	Análisis de casos de uso	96
5.4	DISEÑO	98
5.4.1	Diseño de la arquitectura	99
5.4.2	Diseño de la interfaz	100
5.5	IMPLEMENTACIÓN	102
5.6	EVALUACIÓN	103
5.6.1	Cumplimiento de criterios	103
	CAPITULO 6: FASE DE CONSTRUCCIÓN	104
6.1	DESARROLLO DEL SOFTWARE	104
6.1.1	Estructura de los programas	105
6.1.2	Entrada y salida de datos	105
6.1.3	Interacción con el usuario	106
6.1.4	Mensajes del sistema	106
6.1.5	Tratamiento de riesgos	107
6.2	SUBSISTEMA DE GESTION DE DOCUMENTOS	108
6.3	SUBSISTEMA DE GESTIÓN DE AYUDA	110
6.4	SUBSISTEMA DE GESTION DE SEEPINT MAIL	111
6.5	SUBSISTEMA DE GESTION DE LABORATORIOS	113
6.6	SUBSISTEMA DE GESTION DEL BUZÓN	121

7	CONCLUSIONES	122
8	RECOMENDACIONES	123
	BIBLIOGRAFÍA	
	ANEXOS	

1. TÍTULO SOFTWARE PARA APOYAR EL PROCESO ENSEÑANZA-APRENDIZAJE DE LOS PROTOCOLOS DE COMUNICACIÓN TCP E IP\*.
2. AUTORES DELGADO MARTINEZ RICARDO, VILLA RENTERIA DIEGO ALEJANDRO\*\*.
3. PALABRAS CLAVES Protocolos de comunicación, Educación virtual, JAVA SWING, Aplicaciones Cliente / Servidor, TCP/IP.
4. DESCRIPCIÓN:

Dada la creciente aceptación del Software educativo como herramienta de apoyo en las diferentes ramas del conocimiento y a la ausencia de material de este tipo, enfocado al área de las telecomunicaciones y los diversos tópicos que estas traen consigo, en la Escuela de Ingeniería de Sistemas e Informática de la Universidad Industrial de Santander; se desarrolló el módulo de los protocolos TCP e IP como parte del conjunto de herramientas encaminadas a facilitar el proceso de enseñanza aprendizaje de los protocolos de Internet.

Para generar este módulo de TCP e IP, se tomó la decisión de usar el Lenguaje Unificado de Modelado (UML), como herramienta para el análisis y diseño del sistema; así como el proceso unificado de desarrollo de software, como metodología que permitió su planeación y desarrollo de una manera clara y ordenada.

Se entrega una herramienta educativa que brinda al estudiante la posibilidad de entender el funcionamiento de los procedimientos realizados por los protocolos TCP e IP en la misión de transferir información, de una manera ágil y sencilla dado que los contenidos se encuentran organizados consecuentemente con cada uno de los procesos, logrando difundir el contenido educativo a toda la comunidad universitaria.

La versión inicial de la herramienta, ha sido desarrollada en ambiente web, con soporte en JavaS

---

\* Trabajo de Grado

\*\* Facultad de Ciencias Físico-Mecánicas, Ingeniería de Sistemas, Ing. Hernán Porras Díaz M. Sc., Ph. D.

1. TITLE SOFTWARE TO SUPPORT THE EDUCATION - LEARNING PROCESS OF THE EDUCATION PROTOCOLS TCP AND IP\*.
2. AUTORES THIN DELGADO MARTINEZ RICARDO, VILLA RENTERIA DIEGO ALEJANDRO\*\*.
3. KEY WORDS communications protocols, virtual education, JAVA SWING, applications to Client / Server, TCP/IP.

4. DESCRIPCIÓN:

Given the increasing acceptance of educative software as a support tool in the different branches of knowledge, and from the material absence of this type, focussing in the area of telecommunications and the diverse topics that they bring themselves, in the computing and engineering system school of the Universidad Industrial de Santander, there was developed a module of the protocols TCP and IP as a part of the set of tools desingned to facilitate the learning process of the IP, to generate this TCP module and IP; a decision to use the modulated unified lenguaje (UML) was taken, as a tool for the analysis and the system lesign; furthermore the unified process of software development, as the methodology which allowed the planning and development in a clear and organized way.

An educative tool that offers the student the possibility to understand the functioning of the procedures developed by the protocols TCP and IP in the mission of transferring information is given, in an agile and easy way because the contents are found organized in a consistent way with cach one of the processes, getting to diffuse the educative content to all the university community.

The initial version of the tool has been developed in a web atmosphere with Java Script and applets of Java 2 support.

---

\* Work of degree

\*\* Faculty of sciences physique -mecanicas, Ingeenery of Systems, Ing. Hernán Porras Díaz M. Sc., Ph. D.

## INTRODUCCIÓN

Las herramientas en Internet e Intranets se han convertido en el medio de difusión y presencia de la actividad virtual de las organizaciones en los últimos años, éstas recopilan información sobre sus objetivos, características y servicios. Pero la funcionalidad del sitio web no está enfocada sólo en páginas de lectura, la interactividad usuario – sistema es su característica más importante, haciendo que sus visitantes, los usuarios inicien procesos que les permita incrementar sus conocimientos y realizar operaciones, mejorando los niveles de aprendizaje en el caso de sitios educativos, que brindan opciones pedagógicas enmarcadas en entornos virtuales de instrucción, que colaboran en la obtención de mejores resultados.

Todo el marco metodológico, pedagógico y tecnológico, debe estar acompañado de elementos humanos tan propios y transparentes como lo son la honestidad, el respeto y la responsabilidad, pues de este modo es como un aprendiz puede sacar provecho de los recursos que brinda este tipo de herramientas y aplicarlas a su vida en pro del mejoramiento social, cultural , técnico y de investigación de la persona misma y de la sociedad a la cual hace parte.

La escuela de Ingeniería de Sistemas e informática de la Universidad Industrial de Santander, de frente a los nuevos retos impuestos por el mundo actual, y la avidéz de conocimiento que demandan sus estudiantes, vincula los trabajos de grado, aprovechando el talento humano calificado de sus estudiantes, para la incorporación de herramientas educativas, que organizadas en lugares estratégicos y permitiendo el acceso libre a la comunidad educativa, se podrá

conseguir el mejor beneficio, logrando construir conocimiento, a partir de los elementos que ofrecen estos materiales.

En el marco de las nuevas tendencias gubernamentales, donde el gobierno, trabaja en la ampliación de la cobertura educativa y con el fin de llevar educación a comunidades distantes, es ahí como se puede sacar el mejor provecho de todo ese material que reposa en servidores de la EISI, mediante la implementación de sistemas Web, que permitan a estudiantes en otros lugares apartados del territorio nacional, acceder a programas técnicos o profesionales, aprovechando las plataformas que operan en torno al Internet y de las cuales la universidad puede hacer uso.

Todos estos cambios que se esperan para el futuro, demandan un esfuerzo continuo y progresivo desde el presente, en el cual se organice una estructura que integre y saque el mejor provecho de las herramientas que se han realizado para estos fines.

La herramienta SEEPINT fue pensada como un conjunto de módulos, que se integraran para cubrir en gran parte los contenidos del área de la telemática y telecomunicaciones, hasta el presente se ha desarrollado un modulo que muestra los procesos de DNS con el protocolo UDP. La herramienta que hace parte de este trabajo de grado, es otro elemento más en esa gran cadena de herramientas que esperamos se sigan construyendo, para facilitar el entendimiento de contenidos tan complejos como estos. TCP e IP son la base teórica de está herramienta, que se desarrollo en tecnología web, y con herramientas de libre acceso, para facilitar y garantizar la propagación de la herramienta.

El desarrollo involucró el uso de nuevas herramientas para el análisis y diseño de sistemas como es UML (Lenguaje Unificado de modelado) y de metodologías de desarrollo de software como el proceso unificado, el cual se centra en la funcionalidad que el sistema debe poseer para satisfacer las necesidades de un usuario que interactúa con él.

**PARTE 1: FUNDAMENTOS**  
**CAPITULO 1 PRESENTACIÓN**

**1.1 PRESENTACIÓN DEL INFORME**

El contenido de este documento se encuentra dividido en capítulos, éstos contienen la base teórica del desarrollo de la herramienta educativa para la enseñanza de los protocolos de Internet para la escuela de Ingeniería de Sistemas e Informática de la Universidad Industrial de Santander.

*Parte I:* presenta los fundamentos del proyecto incluyendo los aspectos generales, el marco teórico y metodológico, que sirvieron de guía para el desarrollo del proyecto.

*Capítulo 1. PRESENTACIÓN*

Presenta los objetivos, antecedentes y la justificación de desarrollo del proyecto.

*Capítulo 2. MARCO TEÓRICO*

Presenta el marco teórico de los lenguajes de programación, arquitectura y fundamentos teóricos, utilizados para la realización de la herramienta.

*Capítulo 3. MARCO METODOLÓGICO*

Presenta una comparación entre las metodologías existentes para el desarrollo del software y explica en detalle la metodología del proceso unificado.

Parte II: presenta el desarrollo del sistema a lo largo de cada una de las fases del proceso unificado.

#### *Capítulo 4. FASE DE INICIO*

Explica la fase de inicio, fase que permite poner en marcha el proyecto y definir el alcance del mismo.

#### *Capítulo 5. FASE DE ELABORACIÓN*

Explica la fase de elaboración, fase que permite definir la arquitectura, el diseño y la interfaz del proyecto.

#### *Capítulo 6. FASE DE CONSTRUCCIÓN*

Presenta la fase de construcción, fase que permite explicar la implementación de la primera versión del producto.

El informe finaliza con las conclusiones y recomendaciones de los autores sobre el proyecto.

## **1.2 DESCRIPCIÓN DEL PROYECTO**

**1.2.1 Objetivo General** Facilitar a los estudiantes de pregrado y postgrado de la UIS, en especial a los estudiantes vinculados a la Escuela de Ingeniería de Sistemas e Informática EISI la comprensión conceptual de los procedimientos, que son llevados a cabo por los protocolos TCP e IP en la arquitectura TCP/IP<sup>1</sup>, que hacen posible la comunicación fiable entre computadoras.

---

<sup>1</sup> Transmission Control Protocol/Internet Protocol. TCP/IP es un conjunto de Protocolos de Comunicación, desarrollados para permitir a computadores heterogéneos, compartir recursos a través de una red.

### 1.2.2 Objetivos Específicos

⊕ Desarrollar una herramienta<sup>2</sup> software para que se desempeñe como laboratorio virtual, la cual permita al estudiante observar parámetros<sup>3</sup> que intervienen en la transferencia de mensajes, de modo que asimile la forma en que la arquitectura de protocolos TCP/IP ofrece sus servicios en el nivel de Interred con el protocolo Internet IP y con el protocolo de control de transmisión TCP en la capa de transporte, a la aplicación mencionada en el objetivo siguiente.

⊕ Construir una aplicación que permita emular el funcionamiento de un sistema de correo electrónico, contemplando particularmente el envío y recepción de mensajes, siendo ésta, la aplicación que demanda los servicios de los protocolos TCP e IP; dicha aplicación servirá de apoyo a la herramienta SEEPINT.

⊕ Formalizar, las guías de laboratorio, de tal forma que permitan al estudiante confrontar los conceptos con los casos implementados en la herramienta SEEPINT, de manera que ésta se convierte en un banco de pruebas que facilite el entendimiento de los siguientes procedimientos llevados a cabo por los protocolos TCP e IP:

<sup>4</sup>Fragmentación, reensamblado, direccionamiento de datagramas.

<sup>5</sup>Encaminamiento de datagramas.

<sup>6</sup>Transferencia de datos entre puertos de origen y destino, a través de las primitivas de respuesta y solicitud de servicio.

Detección de errores, mediante el control por retroalimentación, considerando los esquemas de paridad y verificación de suma de bloque<sup>7</sup>.

---

<sup>2</sup> La herramienta final, de ahora en adelante la identificaremos por el nombre de SEEPINT( Software Educativo para la enseñanza de los protocolos de Internet.)

<sup>3</sup> Se tendrán en cuenta parámetros como: interferencia eléctrica, congestionamiento del receptor, congestionamiento de la red.

<sup>4</sup> Laboratorio de fragmentación, reensamblaje y direccionamiento de datagramas.

<sup>5</sup> Laboratorio de encaminamiento, simulando el funcionamiento del protocolo de pasarela interior OSPF.

<sup>6</sup> Laboratorio de transferencia de datos.

Control de errores, mediante el esquema de control de retransmisión básico que es utilizado por TCP, llamado RQ continua<sup>8</sup> .

Control de flujo, considerando el mecanismo de Ventanas deslizantes<sup>9</sup>.

## 1.3 JUSTIFICACIÓN

**1.3.1 Antecedentes y Descripción del Problema** Desde sus aparición, los computadores han traído consigo grandes aportes para las organizaciones, la comunidad científica y la comunidad en general. Entre muchos de tales avances, están las redes computacionales inicialmente concebidas para el intercambio de información entre las dependencias de la empresa u organización a través de sus computadores y luego ampliada para satisfacer la necesidad de compartir recursos y de comunicación particularmente.

Los protocolos de comunicación indiscutiblemente se convierten en uno de los pilares esenciales para el establecimiento y mantenimiento de las comunicaciones entre computadores, ya que son éstos los que establecen los parámetros de comunicación y son los encargados de proporcionar fiabilidad durante la comunicación, generando confianza por parte de los usuarios.

A un cuando existe un modelo muy completo como el OSI de la ISO, hacemos referencia a la arquitectura de protocolos TCP/IP, que caracterizado por su sencillez, se ha convertido en el conjunto de protocolos de mayor difusión en el mercado por ser capaz de brindar, una comunicación fiable y en cierto modo segura.

---

<sup>7</sup> Laboratorio de detección de errores que comprende los dos métodos utilizados por el protocolo TCP en el proceso de detección de errores.

<sup>8</sup> Laboratorio de control de errores; el método utilizado por TCP, es un esquema de solicitud de repetición automática ARQ; que emplea una estrategia de repetición selectiva o de retroceder N. Utilizaremos el RQ-inactiva como vehículo para explicar muchos de los aspectos mas generales del RQ-continua, esto se debe a que RQ-inactiva es el esquema de control de errores más simple.

<sup>9</sup> Laboratorio de control de flujo con Ventana deslizante.

De forma involuntaria, en nuestro que hacer diario hacemos uso de este grupo de protocolos, y es poco probable que cuando se le pregunte a alguien ¿qué harás mas tarde?, la persona responda: “*voy a comunicarme con tal IP para revisar mi correo, bajar música o películas*”, simplemente comenta: “*voy a revisar mi correo y a navegar un ratito en Internet*”. En algún punto de nuestra carrera universitaria quizá nos hagamos la pregunta ¿Cómo logro comunicarme con un computador conectado al otro lado del mundo?, pregunta a la que podemos responder en la medida que avanzábamos en nuestros programas académicos, de acuerdo a los conceptos adquiridos en asignaturas o tópicos relacionados con redes de computadores o que de una u otra forma hemos adquirido mediante la lectura, pero realmente no apreciamos el proceso como se podría apreciar “el choque de dos cuerpos en un laboratorio de colisiones en el área de física”.

La información efectiva para el desarrollo de una institución es aquella que se encuentre disponible en el momento y en lugar preciso; entendiendo este enunciado la UIS decidió instalar, en 1986, la primera red de datos de área local (LAN) para el campus principal, utilizando varios servidores y estaciones de trabajo UNIX y PRIMOS, todos ellos bajo una plataforma común de comunicaciones de la familia de protocolos TCP/IP.

En 1992, como consecuencia de la corriente de globalización, la universidad a través de una conexión con par telefónico dedicado a la red COLDAPAQ de Telecom, dispuso del primer enlace a la red telemática de alcance mundial BITNET, predecesora de Internet. Con este enlace, se disponía además de los servicios locales, con los servicios de correo electrónico, transferencia bidireccional de archivos e intercambio electrónico de mensajes en línea con cualquier usuario conectado a BITNET.

Con la aparición de nuevas tecnologías de telecomunicaciones y la creciente demanda de servicios de red, en 1995 la UIS decidió rediseñar su red de área local; esto originó el crecimiento acelerado de la red, la cual ahora contaba con

nuevos equipos, sistemas operativos y protocolos de red. En esta época Internet se encontraba en su apogeo y apareció el término Intranet para describir la aplicación de las tecnologías propias de Internet aplicadas en redes corporativas internas. En general, las empresas empezaron a utilizar la Intranets para publicar y compartir información en una forma más efectiva, rápida y económica mediante la aplicación de los paradigmas de vinculación y presentación derivados de Internet.

En 1998 se hace entrega oficial de la primera versión del Sistema Intranet administrativa para la UIS, como un nuevo producto del proyecto OSIRIS<sup>10</sup>, ofreciendo una gama de servicios orientados a las dependencias administrativas de la UIS, tales como la publicación de documentos emitidos por el Consejo Superior, la rectoría y el Consejo Académico; además de la publicación de noticias y eventos relacionados con la institución entre otros.

Aprovechando, la infraestructura de comunicación, que se halla en funcionamiento dentro del claustro universitario, vemos la necesidad que hay por parte de los estudiantes y docentes, de contar con herramientas que relacionadas entre sí brinden a los estudiantes de ingeniería de sistemas, especialización en telecomunicaciones y todo profesional que adscriba a su campo de desempeño el área de las telecomunicaciones, la oportunidad de adquirir conocimientos que en el presente son un tanto difíciles de entender, en sus aspectos conceptuales.

En virtud de la misión de la universidad, somos conscientes de la necesidad que existe en la actualidad, por parte del profesional del futuro, adquirir de manera integral, los conceptos que fundamentan el desempeño de las tecnologías emergentes de última generación, con este propósito, planteamos, cooperar en el desarrollo de un conjunto de aplicaciones que permita a las escuelas de la UIS que dictan este tipo de contenido en sus asignaturas, dotar al estudiante de un material digital educativo claro, sin límite de ejemplares, para que el estudiante e

investigador de nuevas tecnologías pueda entender con claridad su fundamento, de modo que inicie procesos de desarrollo, en los que brinde aportes en el mejoramiento técnico de las tecnologías de redes existentes, y se dé pie para el surgimiento de otras. En nuestro caso, dado que estamos trabajando con algunos de los conceptos que determinan el desempeño de la arquitectura de protocolos TCP/IP, el estudiante investigador tendrá a disposición una herramienta educativa que le dará soporte para iniciar el estudio de la seguridad de redes en Internet, en el ámbito de protocolos, pudiendo aportar en la mejora o diseño de nuevos protocolos de comunicación seguros en el sentido de confiabilidad, integridad y autenticación de los datos, en el desarrollo de aplicaciones de red con grados de seguridad confiables para el usuario que use las redes como medio para la realización de transacciones u operaciones en línea.

El docente contará con una herramienta de apoyo, con la cual podrá mostrar y de esta forma lograr que sus alumnos comprendan los mecanismos de comunicación que operan en el nivel de Interred y transporte con los protocolos IP y TCP respectivamente; herramienta que podrá utilizar también para explicar temas relacionados con algoritmos de búsqueda, teoría de grafos, teoría de decisiones, simulación digital, entre otros, a partir de los diferentes laboratorios mencionados en los numerales 2.2.3.1 a 2.2.3.6; facilitando así, el proceso enseñanza-aprendizaje.

Como puede notarse, el desarrollo de esta herramienta educativa demanda la aplicación de muy variados conocimientos propios de la ingeniería de sistemas, en especial los relacionados con el ciclo de vida de desarrollo de software, telemática, simulación digital y entre otros. Además, el tamaño y la complejidad algorítmica de la aplicación, la utilización y adaptación de diversas tecnologías de información a nuestro ambiente educativo convierte a este proyecto en un trabajo de grado válido para dos estudiantes de Ingeniería de Sistemas.

---

<sup>10</sup> Proyecto de Optimización de los Servicios Informáticos y de las Redes de Información como Soporte de la Gestión Universitaria –OSIRIS-.

### **1.3.2 Descripción de los Objetivos**

Desarrollar una herramienta software para que se desempeñe como laboratorio virtual, la cual permita al estudiante observar parámetros que intervienen en la transferencia de mensajes, de modo que asimile la forma en que la arquitectura de protocolos TCP/IP ofrece sus servicios en el nivel de Interred con el protocolo Internet IP y con el protocolo de control de transmisión TCP en la capa de transporte, a la aplicación mencionada en el objetivo siguiente.

Para poder interactuar con los diferentes laboratorios, la herramienta SEEPINT será el medio a través del cual el (los) estudiante(s) y el (los) docente(s) podrán hacer dicha interacción, aprovechando los elementos computacionales que facilitan el aprendizaje.

Construir una aplicación que permita emular el funcionamiento de un sistema de correo electrónico, contemplando particularmente el envío y recepción de mensajes, siendo ésta, la aplicación que demanda los servicios de los protocolos TCP e IP; dicha aplicación servirá de apoyo a la herramienta SEEPINT.

La arquitectura de protocolos TCP/IP contempla una aplicación en su capa superior, la cual demanda los servicios mencionados en los numerales 2.2.3.1. a 2.2.3.6.

Para mostrar cómo los protocolos TCP e IP ofrecen dichos servicios, apoyaremos la herramienta SEEPINT, en la emulación de un sistema de correo electrónico; dicho sistema de correo se construirá mediante la simulación de los agentes de transferencia de mensajes.

Formalizar, las guías de laboratorio, de tal forma que permitan al estudiante confrontar los conceptos con los casos implementados en la herramienta SEEPINT, de manera que ésta se convierte en un banco de pruebas que facilite el entendimiento de los siguientes procedimientos llevados a cabo por los protocolos TCP e IP.

Los numerales 2.2.3.1 al 2.2.3.6 del objetivo 2.2.3. Comprenden el contenido temático que manejará la herramienta, los cuales serán abordados a partir de la construcción de las guías de los laboratorios respectivos, de esta manera se hace posible cumplir con los requerimientos que se enuncian en el objetivo general.

La nueva revolución educativa del siglo XXI, exige tanto a docentes como a estudiantes, adquirir la cultura del aprendizaje colaborativo apoyado en el autoaprendizaje, por esto se hace útil el empleo de herramientas computacionales que apoyen al estudiante en el proceso de aprendizaje, de tal manera que el docente se convierte en un guía del proceso.

Estas herramientas, permiten que el estudiante aprenda a partir de su propia interacción con la herramienta, teniendo la ventaja de que el error es considerado como una posibilidad de autovaloración de los procesos realizados, permitiendo al mismo tiempo la reflexión del alumno para la mejora de los resultados.

**1.3.3 Impacto** El desarrollo de este proyecto se traducirá en beneficios para la escuela de Ingeniería de Sistemas e Informática de la Universidad Industrial de Santander, ya que servirá como material de soporte en cursos donde manejen contenidos relacionados con el tema de los protocolos de Internet.

De igual forma le permitirá a los estudiantes, ahorro de recursos económicos y materiales que normalmente utilizan en el proceso de aprendizaje, ya que mediante esta herramienta, y aprovechando la estructura de red con que cuenta la universidad, podrán acceder al material de forma gratuita.

**1.3.4 Viabilidad** La herramienta SEEPINT será implantada en el servidor alojado en la Escuela de Ingeniería de Sistemas e Informática.

Además, para la realización de este proyecto se utilizarán herramientas de libre distribución. El uso de este tipo de herramientas poseen muchas ventajas durante su implementación e implantación.

Estas herramientas de licencia gratuita, cuentan con un enorme grupo de personas trabajando en torno a ofrecer en ellas la mayor calidad y seguridad. Permitiendo encontrar diversos grupos de discusión que guían y aportan al desarrollador en como usarlas adquiriendo los mayores beneficios, además de dar la oportunidad de intercambiar conocimientos sobre éstos productos.

## **CAPITULO 2 MARCO TEÓRICO**

### **2.1 GENERALIDADES DE INTERNET**

Internet ha cambiado la forma de comunicación de millones de personas, cosas que años atrás parecían imposibles, son realidades cotidianas. Hoy en día, gracias a Internet, cualquier usuario de ordenador puede transmitir y recibir información de todo tipo a través de la línea telefónica, muy pronto a través de la línea de energía. Internet es una red de redes de ordenadores que pueden compartir todo tipo de información, y se ha convertido en la actualidad en una herramienta vital para el desarrollo de las comunidades humanas.

Todo comenzó hace aproximadamente 35 años en EE.UU. cuando el departamento de defensa creó un sistema de comunicación dedicado a la investigación militar (ARPANET) que pudiese hacer frente a una guerra nuclear. Para ello pusieron en marcha una red de ordenadores interconectados, de tal manera que aunque se produjesen interrupciones en las líneas telefónicas siempre existiesen caminos alternativos para poder continuar el envío de información. Posteriormente, centros de investigación y universidades utilizaron esta infraestructura para comunicarse entre sí, más tarde se conectaron empresas e instituciones diversas. Finalmente fueron los usuarios particulares de decenas de países los que con el tiempo han llegado a constituir lo que se conoce actualmente como Internet.

El crecimiento exponencial de Internet debe su éxito a varios hechos: su universalidad, la posibilidad de expansión continua sin límite aparente y la capacidad de adaptarse rápidamente a las nuevas tecnologías que puede incorporar, de forma que continuamente crece en tamaño, en posibilidades y servicios. Otro aspecto fundamental es el hecho, que la conexión entre dos puntos cualesquiera del mundo se gestiona como una llamada local, lo que supone un

ahorro importante en el costo de la misma. Todo ello hace que Internet sea una herramienta de trabajo fundamental en profesiones como la medicina, la educación, en las que son necesarias la comunicación entre colegas y la adquisición de información actualizada.

Podríamos decir que Internet funciona como una nube o red de ordenadores interconectados entre si mediante líneas de comunicaciones de alta velocidad. Conectados a esta red se encuentran otros ordenadores (nodos), que son los que dan acceso a los usuarios individuales. Los ordenadores de éstos se conectan a los nodos a través de la línea telefónica.

La comunicación en la red se realiza siguiendo el esquema de “Cliente/Servidor”, de tal forma que el ordenador del usuario (Cliente) accede a la información generada por un programa que se ejecuta en un ordenador remoto (Servidor).

Para poder comunicarse entre sí, los clientes y los servidores comparten unas reglas de comunicación (protocolos) que se encargan de compatibilizar las solicitudes de información y las respuestas a dichas peticiones. Internet se basa en un sistema de transporte de información llamado TCP/IP que permite la transmisión correcta de datos.

**2.1.1 World Wide Web – WWW** Es el servicio más popular en Internet, comúnmente conocido como “WWW” o simplemente “Web”. Está basado en tecnología de hipertexto. Con este mecanismo se puede ir navegando de un ordenador a otro, pasando por las reseñas de la información al lugar donde ésta reside, independientemente del servidor con que inicialmente se estableció la conexión. Permite organizar la información de Internet, atravesando documentos. El servidor se limita a transferir archivos a los clientes, y éstos visualizan la información. La importancia de los programas clientes ha hecho que éstos sean más conocidos que el propio servicio.

Los recursos apuntados desde el enlace de la página web pueden ser de cualquier tipo:

- ⊕ Otro documento hipertextual
- ⊕ Ordenadores anfitriones para hacer telnet (directorios, bases de datos)
- ⊕ Ficheros para traer por FTP<sup>11</sup>
- ⊕ Ficheros con imágenes, sonidos, videos digitalizados
- ⊕ Documentos con “formato” (Word, Adobe Acrobat, entre otros.)

**2.1.2 Exploradores y servidores web** El diseño del World Wide Web sigue el modelo cliente / servidor: un paradigma de división del trabajo informático en el que las tareas se reparten entre un número de clientes que efectúan peticiones de servicios de acuerdo con un protocolo, y un número de servidores que las atienden. En el Web, las estaciones de trabajo son clientes que demandan hipertextos a los Servidores. Para el buen funcionamiento de este sistema es necesario del protocolo de Transferencia de Hipertexto (HTTP<sup>12</sup>) y el Lenguaje de Marcado de Hipertextos (HTML<sup>13</sup>).

Un explorador es un programa que recorre el World Wide Web y muestra páginas. El explorador solicita una página de un servidor con base a su dirección de Internet, recupera el documento del servidor y presenta el contenido. A medida que HTML evoluciona, los exploradores evolucionan con él, agregando funcionalidades más avanzadas.

---

<sup>11</sup> FTP (File Transfer Protocol) es un protocolo para hacer posible la transferencia de archivos entre computadoras locales o remotas.

<sup>12</sup> http (HyperText Transfer Protocol). Protocolo que permite realizar saltos hipertextuales (nodo de origen a nodo de destino) recuperando en forma rápida y efectiva documentos hipermedia de la WWW.

<sup>13</sup> HTML (Hyper Text Markup Lenguaje) permite representar hipertextos incluyendo información sobre la estructura y el formato de representación y, especialmente, indicando el origen y destino de saltos hipertextuales.

## **2.2 SOFTWARE EDUCATIVO**

Podemos situar la producción de programas informáticos para el sector educativo en los inicios de los años sesenta. En aquella época, en Norteamérica, a pesar de no existir los ordenadores personales, se comenzaron a desarrollar programas informáticos con finalidad educativa. Evidentemente, la producción de software ha cambiado notablemente debido a los espectaculares avances técnicos. Sin embargo, los problemas pedagógicos de fondo siguen presentes y las discusiones sobre las estrategias y métodos de enseñanza que debe adoptar un programa informático para la enseñanza sigue siendo materia de total actualidad.

La complejidad de la producción de software educativo estriba en el hecho de que deben efectuarse decisiones en torno a los contenidos (selección, organización, adaptación a los usuarios, etc), a las estrategias de enseñanza de dichos contenidos y a la forma de presentación (diseño de pantallas) más adecuadas con el objeto de facilitar el proceso de aprendizaje del usuario.

**2.2.1 Tipos de Software Educativo** Existen en el mercado muchos tipos de productos a los que se califica como educativos. Aquí nos referiremos al software educativo como cualquier producto realizado con una finalidad educativa. Bajo la amplia definición podemos encontrar programas de lo más variado. Adaptaremos una clasificación bastante estándar dividiendo el formato de los programas en cinco tipos distintos: . (Ver tabla 1) tutoriales, práctica y ejercitación, simulación, hipertextos e hipermedias.

Tabla 1. Clasificación de programas educativos.

Tipo de programa	Propósito del programa	Decisiones sobre el diseño instructivo
Tutorial	Programa de enseñanza	Contenido en función del nivel de los usuarios, estructuración del contenido, estrategia didáctica.
Práctica y ejercitación	Programa de ejercicios, ayuda a la adquisición de una destreza	Nivel, contenido y estructura de los ejercicios
Simulación	Proporcionar entornos de aprendizaje basados en situaciones reales	Modelo de simulación, tipos de feedback.
Hipertextos Hipermedias	Proporciona un entorno de aprendizaje no lineal	Organización de contenido, Determinación de los enlaces, selección de medios (hipermedia)

Hay que tener presente que la división descrita en la tabla 1, es bastante teórica ya que en la actualidad, en un mismo programa así como en la herramienta SEEPINT, podemos encontrar formatos diferentes. Por ejemplo, en SEEPIN podemos encontrar una parte Tutorial, complementada por una simulación y unos ejercicios para evaluar conocimientos adquiridos. Aunque existen programas que incorporan más de un formato, pensamos que la clasificación sigue siendo útil ya que cada uno de estos formatos conlleva decisiones diferentes sobre el diseño instructivo.

**2.2.2 Elaboración del software educativo.** Existen metodologías para la creación de productos informáticos. La denominada ingeniería del software trata precisamente del estudio de los métodos para la optimización de la producción de programas informáticos. Éstos se aplican en la elaboración de programas que resultan altamente complejos en cuanto a tamaño y estructura. Para la elaboración de programas educativos no suele utilizarse este tipo de métodos ya que los programas son de menor tamaño aunque pueden tener una estructura organizativa más compleja. La diferencia fundamental estriba en que un producto

educativo no es un programa de control de una actividad específica sino que se supone que el producto provocara un cambio en el usuario, le proporcionará un determinado aprendizaje.

Es nuestro, el caso de productos de enseñanza no comerciales producidos por profesores y estudiantes. Son productos diseñados a medida para un determinado curso. No suelen ser comerciales y normalmente se desarrollan en universidades, organizaciones publicas, departamentos de formación, etc.

## **2.3 ARQUITECTURA DE INFORMACIÓN**

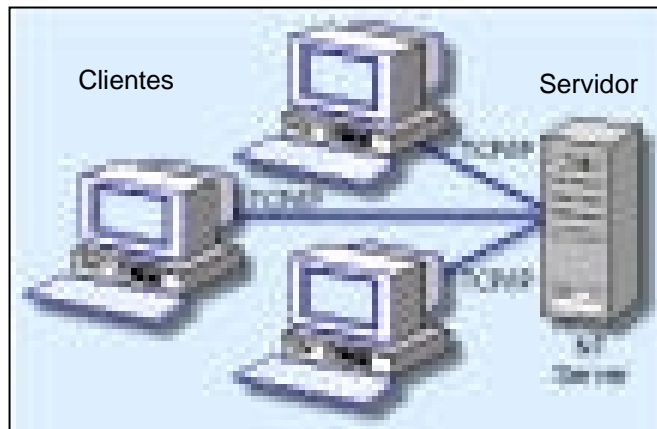
**2.3.1 Modelos Cliente / Servidor** Cliente/Servidor es una frase que se usa para describir un modelo de interacción entre dos procesos, que se ejecutan en forma simultánea. Este modelo es una comunicación basada en una serie de preguntas y respuestas, que asegura que si dos aplicaciones intentan comunicarse, una comienza la ejecución y espera indefinidamente que la otra le responda y luego continua con el proceso.

La mayoría de las comunicaciones extremo a extremo en las redes, están basadas en el modelo Cliente/Servidor.

- ⊕ Cliente: aplicación que inicia la comunicación, es dirigida por el usuario.
- ⊕ Servidor: es quien responde a los requerimientos de los clientes, son procesos que se están ejecutando indefinidamente.

Los procesos clientes son más sencillos que los procesos de los servidores, los primeros no requieren de privilegios de sistemas para operar, en cambio los procesos servidores sí.

Figura 1. Proceso Cliente/Servidor



Los usuarios cuando quieren acceder a un servicio de red, ejecutan un software cliente.

Esta arquitectura está compuesta de 2 capas:

- ⊕ La capa de presentación: en ella se encuentra HTML, DHTML y Scripting, componentes, programas ejecutables y paginas web.
- ⊕ La capa de aplicación o negocio que consume más recursos: en ella se encuentra entre otros, el servidor de componentes, servidor de mensajería y servidores web.

Ambas capas se implementan y ejecutan en el cliente, de modo que el servidor funciona como una base de datos dedicada solamente al almacenamiento de datos.

## 2.4 PÁGINAS WEB ESTÁTICAS

Las páginas web estáticas son textos ASCII escritos en lenguaje HTML. Que se transfieren entre los servidores de WWW y los navegadores de los clientes mediante el protocolo, su único proceso realizado es la visualización de sus contenidos (escritos en lenguaje HTML) por parte del explorador del cliente. Las

páginas estáticas forman la base necesaria para la presentación de datos en muchos tipos de situaciones. También influye decisivamente la sencillez con que se pueden crear, instalar y mantener. Uno de los lenguajes que presenta páginas web estáticas es HTML, a continuación se presentan algunas características de este lenguaje.

**2.4.1 HTML (*HyperText Markup Language*)** Es un lenguaje muy sencillo que permite escribir hipertexto, es decir, texto presentado en forma estructurada y agradable, con enlaces que conducen a otros documentos o fuentes de información relacionadas, y con inserciones multimedia (gráficos, sonidos, etc). La descripción se basa en especificar en el texto la estructura lógica del contenido (títulos, párrafos de texto normal, enumeraciones, definiciones, citas, etc.) así como los diferentes efectos que se quieran dar (especificar los lugares del documento donde se debe poner cursiva, negrita, o un gráfico determinado) y dejar que luego la presentación final de dicho hipertexto se realice por un programa especializado del cliente, como Internet Explorer o Netscape.

Adicionalmente, los documentos HTML pueden ser vistos en toda clase de sistemas como son Macintosh, PC o UNIX.

## **2.5 PÁGINAS WEB DINAMICAS EN EL CLIENTE**

En estas páginas toda la carga de procesamiento de los efectos y las funcionalidades las soporta el navegador. Los usos típicos de las páginas de cliente son efectos especiales para webs como el control de ventanas, presentaciones en las que se pueden mover objetos por la página, control de formularios, cálculos, etc.

El código necesario para crear los efectos y funcionalidades se incluye dentro del mismo archivo HTML y es llamado SCRIPT. Cuando una página HTML contiene scripts de cliente, el navegador se encarga de interpretarlos y ejecutarlos para realizar los efectos y funcionalidades.

Las páginas del cliente son muy dependientes del sistema donde se están ejecutando y esa es su principal desventaja, ya que cada navegador tiene sus propias características, incluso cada versión, y lo que puede funcionar en un navegador puede no funcionar en otro.

Como ventaja se puede decir que estas páginas descargan al servidor algunos trabajos, ofrecen respuestas inmediatas a las acciones del usuario y permiten la utilización de algunos recursos de la máquina local.

Entre los lenguajes y herramientas que colaboran a la presentación de páginas web dinámicas en el cliente se tienen los enunciados a continuación.

**2.5.1 HTML dinámico (DHTML)** El HTML dinámico es una forma que tienen las páginas de aportar interactividad a las mismas. DHTML es una característica de Netscape Communicator 4.0 y Microsoft Explorer 4.0 y posteriores versiones de ambos navegadores, y está orientada al usuario. Es tarea del navegador mostrar y manipular las páginas web.

El DHTML tiene la ventaja de que es una herramienta con la que se pueden crear efectos que requieren poco ancho de banda, a la hora de bajarlos de Internet y, son estos efectos los que aumentan la funcionalidad de las páginas. Se puede utilizar para crear animaciones, juegos, aplicaciones, para introducir nuevas formas de navegar a través de los sitios web, y para crear un auténtico entramado en capas que con sólo el HTML sería imposible abordar. Aunque muchas de las características del DHTML se podrían duplicar con otras herramientas como Java

o Flash, el DHTML ofrece la ventaja de que no requiere ningún tipo de plug-in<sup>14</sup> para poder utilizarlo.

Aunque las tecnologías en las que se basa el DHTML (HTML, CSS<sup>15</sup>, JavaScript) están estandarizadas, la forma en que Netscape y Microsoft las implementan difieren entre sí.

Por este motivo, la creación de páginas web que usen esta tecnología, puede llegar a convertirse en una tarea muy compleja, puesto que hay que conseguir que la página se visualice perfectamente en ambos navegadores.

**2.5.2 Lenguajes scripts** Un lenguaje script permite embeber código fuente para la programación del lado del cliente, directamente en la página HTML, y el plug-in que interpreta ese lenguaje se activará automáticamente cuando se cargue el navegador. Estos lenguajes tienden a ser muy simples y sencillos, además se cargan muy rápidamente porque van incluidos en la página que envía el servidor. Estos lenguajes se utilizan fundamentalmente para hacer más atractiva la interfaz de las páginas, por que disponen de elementos gráficos y pueden llegar a resolver parte de los problemas que se plantean en la programación de la parte cliente del sistema Cliente/Servidor; siendo lenguajes mucho más sencillos de aprender e implementar soluciones con ellos que recurrir a Java o ActiveX. Entre los lenguajes de guiones (scripts) se encuentran JavaScript.

Las aplicaciones más habituales de estos lenguajes son:

- ⊕ Validar datos en el cliente y comprobar la consistencia de los valores antes de enviar un formulario.

---

<sup>14</sup> Un plug-in es un software que se instala en el ordenador del cliente y que, en unión al navegador, permite visualizar algunos contenidos de páginas web en formatos diferentes al HTML.

<sup>15</sup> CSS. (Cascading Style Sheets) Hojas de estilo en cascada. Determinan el estilo para cada elemento y medio de presentación. Permiten separar físicamente el estilo del contenido. Pueden ir definidas en cascada, donde una hoja de estilo puede dar modificaciones sobre otra ya definida, además, al modificar una hoja de estilo, se modifica la presentación de todo el sitio.

- ⊕ Actualizar campos relacionados en formularios (por ejemplo, establecer las opciones de una lista desplegable en función del valor seleccionado en unos botones de radio).
- ⊕ Realizar procesamientos que no requieran la utilización de información centralizada.
- ⊕ Servir de base para la utilización de otras tecnologías (HTML Dinámico, XML, ActiveX).

**2.5.2.1 JavaScripts** Se trata de un lenguaje de programación del lado del cliente, debido a que el navegador soporta la carga de procesamiento. Gracias a su compatibilidad con la mayoría de los navegadores modernos, es un lenguaje de programación del lado del cliente más utilizado.

Con JavaScript se crean efectos especiales en las páginas y se definen interactividades con el usuario. El navegador del cliente es el encargado de interpretar las instrucciones Javascript y ejecutarlas para realizar estos efectos e interactividades, de modo que el mayor recurso, y tal vez el único, con que cuenta este lenguaje es el propio navegador.

Entre las acciones típicas que se pueden realizar en JavaScript se conocen dos vertientes. Por un lado los efectos especiales sobre páginas web, para crear contenidos dinámicos y elementos de la página que tengan movimiento, cambien de color o cualquier otro dinamismo. Por el otro, Javascript permite ejecutar instrucciones como respuesta a las acciones del usuario, permitiendo crear páginas interactivas con programas como calculadoras, agendas, o tablas de cálculo.

Javascript es un lenguaje con muchas posibilidades, permite la programación de pequeños scripts, pero también de programas más grandes, orientados a objetos, con funciones, estructuras de datos complejas, etc.

Javascript no es una versión limitada de otro lenguaje, y no es una simplificación de cualquier otro. Es por lo tanto, limitado. No se pueden escribir aplicaciones autónomas, por ejemplo, y tiene una capacidad muy limitada para leer y escribir

archivos. Es más, los script (o guiones) de JavaScript no se pueden ejecutar sin la presencia de un interprete, bien sea un servidor web o en un navegador (o browser).

JavaScript es un lenguaje sin tipos de datos. Es decir, no se necesita declarar el tipo de datos de las variables explícitamente cuando sea necesario. Por ejemplo, si se quiere añadir un número a un campo o ítem que es una cadena de caracteres (o string), él número se convierte a texto.

**2.6.3 Applets de java** Un Applet de Java es una aplicación Java que presenta unas ciertas restricciones y que puede ser vinculada a una página web y enviada por Internet desde un servidor a un cliente.

La principal ventaja de los applets de Java reside en su sistema de seguridad, que permite ejecutar aplicaciones en los equipos de los clientes de forma segura. El inconveniente viene de la mano de la necesidad de mayores recursos en los equipos de los clientes, así como de un sistema de ejecución más lento basado en la interpretación de código intermedio.

*Características de los applets:*

Los archivos de Java compilados (\*.class) se descargan a través de la red desde un servidor hasta el browser en cuya JVM (Java Virtual Machine<sup>16</sup>) se ejecutan. Pueden traer también a través de la red ficheros de imágenes y sonido.

Los applets no tienen ventana propia: se ejecutan en la ventana del browser (en un panel incrustado en la página HTML, puesto que la clase Applet descende de panel)

---

<sup>16</sup> JVM es un programa que traduce el código Java para una plataforma específica de hardware, software que hace que Java sea portátil.

Tienen importantes restricciones de seguridad, que se chequean al llegar al browser: sólo pueden acceder a una limitada información sobre el ordenador en el que se ejecutan, sólo pueden leer y escribir archivos en el servidor del que han venido, etc.

Las Applets de confianza (trusted) pueden pasar por encima de estas restricciones.

## **2.7 PROTOCOLOS DE INTERNET**

Los protocolos están presentes en todas las etapas necesarias para establecer una comunicación entre equipos de cómputo, desde aquellas de más bajo nivel (e.g. la transmisión de flujos de bits a un medio físico) hasta aquellas de más alto nivel (e.g. el compartir o transferir información desde una computadora a otra en la red).

Tomando al modelo OSI (Open Systems Interconnection) como referencia podemos afirmar que para cada capa o nivel que él define existen uno o más protocolos interactuando. Los protocolos son entre pares (peer-to-peer), es decir, un protocolo de algún nivel dialoga con el protocolo del mismo nivel en la computadora remota.

### **2.6.1 Modelo de capas TCP/IP**

**2.6.1.1 Evolución Histórica De TCP/IP** Remontándonos en la historia, en 1969 la *Defense Advanced Research Projects Agency* (DARPA) financió un proyecto de investigación y desarrollo para crear una red de conmutación de paquetes experimental. Esta red, llamada ARPANET, fue construida para estudiar técnicas para proporcionar robustez, fiabilidad y comunicación independiente del fabricante. Múltiples técnicas de comunicación modernas fueron desarrolladas inicialmente en ARPANET.

La red experimental ARPANET tuvo tanto éxito que muchas de las organizaciones agregadas a ella empezaron a utilizarla para las comunicaciones diarias. En 1975 ARPANET se convirtió en una red operativa y la responsabilidad de administrar la red fue concedida a la *Defense Communications Agency* (DCA, ahora llamada *Defense Information Systems Agency-DISA*). Sin embargo, el desarrollo de ARPANET no se detuvo. Las bases de los protocolos TCP/IP se desarrollaron después de que ésta fuese operativa. De hecho, la incorporación de nuevas redes provocó un cambio en la filosofía de trabajo de algunos de los protocolos de ARPANET debido a que creció la diversidad de situaciones de trabajo al tener interconectadas entre sí redes de distintos fabricantes, calidades, equipo, *software*, usos.

Los protocolos TCP/IP fueron adoptados como estándares militares (MIL STD) en 1983 y todos los *hosts* conectados a la red debían convertirse a estos protocolos. Para facilitar esta conversión, DARPA fundó Bolt, Beranek y Newman (BBN) para que TCP/IP estuviese implementado en los sistemas UNIX Berkeley (de ahí viene la asociación UNIX-TCP/IP).

Al tiempo que TCP/IP se tomaba como estándar, el término *Internet* era cada vez más usado. En 1983, ARPANET se dividió en MILNET, una parte sin clasificar del *Defense Data Network* (DDN), y una nueva y pequeña ARPANET. Este mismo año se añadió al conjunto CSNET, y en 1986 NSFNET engrosó las filas del tratamiento distribuido en red que dio origen a Internet.

El término "Internet" se usaba inicialmente para referirse al conjunto MILNETARPANET. En 1990 ARPANET pasó formalmente a la historia, pero hoy en día *Internet* se usa más que nunca y en redes de todo el mundo. El desarrollo de Internet ha llevado consigo el crecimiento de distintas organizaciones dentro de la red.

El término **Internet** ha llevado en ocasiones a confusión: *internet* (i minúscula) – interred– es un nombre común que hace referencia a la interconexión de distintas redes físicas bajo un protocolo común, formando una única red lógica. Internet (I

mayúscula) es un nombre propio que se refiere al conjunto de redes mundiales que usan el *Internet Protocol* (IP).

Tras este breve resumen de los orígenes de ARPANET y TCP/IP pasaremos en la siguiente sección a estudiar sus características.

**2.6.1.2 Que es TCP/IP?** TCP/IP es el protocolo común utilizado por todos los ordenadores conectados a Internet, de manera que éstos puedan comunicarse entre sí. Hay que tener en cuenta que en Internet se encuentran conectados ordenadores de clases muy diferentes y con hardware y software incompatibles en muchos casos, además de todos los medios y formas posibles de conexión. Aquí se encuentra una de las grandes ventajas del TCP/IP, pues este protocolo se encargará de que la comunicación entre todos sea posible. TCP/IP es compatible con cualquier sistema operativo y con cualquier tipo de hardware.

TCP/IP no es un único protocolo, sino que es en realidad lo que se conoce con este nombre es un conjunto de protocolos que cubren los distintos niveles del modelo OSI. Los dos protocolos más importantes son el TCP (Transmission Control Protocol) y el IP (Internet Protocol), que son los que dan nombre al conjunto y son el objeto de nuestro estudio. Aunque no existe una visión universal de cómo está TCP/IP dispuesto en capas, normalmente se lo describe como un sistema compuesto por menos capas que el modelo OSI. La mayoría de las descripciones que se hacen de TCP/IP lo definen como una arquitectura de tres a cinco capas. El modelo de cuatro capas de la Figura 2 es la descripción más estándar para TCP/IP.

Figura 2. Descripción del modelo de capas TCP/IP

Aplicación	<b>SMTP,</b> http, ftp, bgp, ssl, DNS, telnet, finger, POP3/MAP	DNS, RIP, SNMP, traceroute, tftp	Ping		
Transporte	<b>TCP</b>	UDP	ICMP	<b>OSPF</b>	
Interred	<b>IP</b>				ARP
Enlace	Ethernet 802.3, Token Ring/ 802.5, FDDI SDN Frame Relay ATM Wireless/802.11 SONET/SDH PPP, HDLC, XDCI, Cable Modem				

⊕ **Aplicación:** Se corresponde con los niveles OSI de aplicación, presentación y sesión. Aquí se incluyen protocolos destinados a proporcionar servicios, tales como correo electrónico (SMTP), transferencia de ficheros (FTP), conexión remota (TELNET) y otros más recientes como el protocolo HTTP (Hypertext Transfer Protocol).

⊕ **Transporte:** Coincide con el nivel de transporte del modelo OSI. Los protocolos de este nivel, tales como **TCP** y UDP, se encargan de manejar los datos y proporcionar la fiabilidad necesaria en el transporte de los mismos.

⊕ **Interned:** Es el nivel de red del modelo OSI. Incluye al protocolo **IP**, que se encarga de enviar los paquetes de información a sus destinos correspondientes. Es utilizado con esta finalidad por los protocolos del nivel de transporte.

⊕ **Acceso a la Red ó Enlace:** Los niveles OSI correspondientes son el de enlace y el nivel físico. Los protocolos que pertenecen a este nivel son los encargados de la transmisión a través del medio físico al que se encuentra conectado cada host, como puede ser una línea punto a punto o una red Ethernet.

El TCP/IP necesita funcionar sobre algún tipo de red o de medio físico que proporcione sus propios protocolos para el nivel de enlace de Internet. Por este motivo hay que tener en cuenta que los protocolos utilizados en este nivel pueden ser muy diversos y no forman parte del conjunto TCP/IP. Sin embargo, esto no debe ser problemático puesto que una de las funciones y ventajas principales del TCP/IP es proporcionar una abstracción del medio de forma que sea posible el intercambio de información entre medios diferentes y tecnologías que inicialmente son incompatibles.

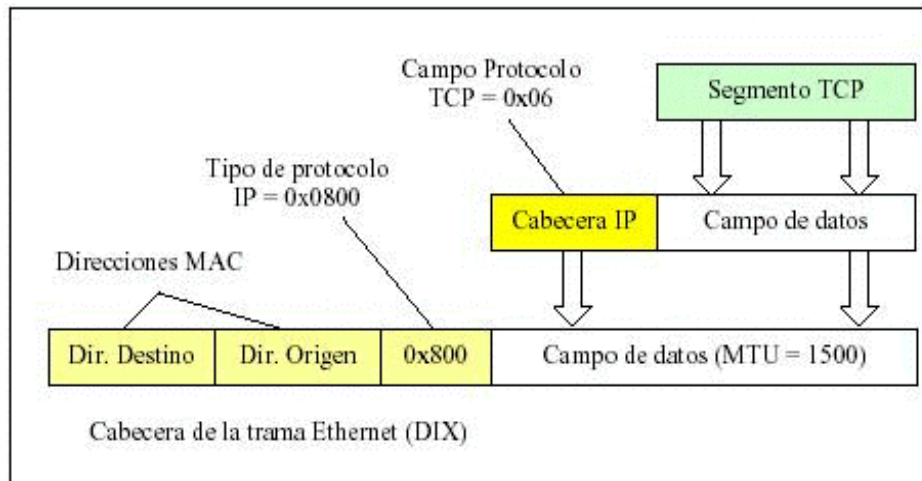
Para transmitir información a través de TCP/IP, ésta debe ser dividida en unidades de menor tamaño. Esto proporciona grandes ventajas en el manejo de los datos que se transfieren y, por otro lado, esto es algo común en cualquier protocolo de comunicaciones. En TCP/IP cada una de estas unidades de información recibe el nombre de "datagrama" (datagram), y son conjuntos de datos que se envían como mensajes independientes.

Como en el modelo OSI, los datos pasan hacia abajo en la pila cuando se envían por la red y hacia arriba cuando se reciben. Cada capa de la pila añade una información de control para asegurarse del reparto seguro de los datos. A esta información de control se le conoce como *cabecera* porque se coloca delante de los datos que se van a transmitir. Cada capa trata toda la información recibida de la capa superior como datos y le pone su cabecera delante de esta información.

Al hecho de añadir información de control adicional en cada capa se le conoce como *encapsulamiento*. Cuando los datos se reciben ocurre el caso contrario, cada capa retira la cabecera correspondiente antes de que la información le llegue a la capa superior. Por tanto, la información que se recibe de una capa inferior se trata como una cabecera y unos datos. Observamos en la Figura 3 este proceso.

Al igual que en el caso de UDP e ICMP, los segmentos TCP viajan dentro de tramas IP

Figura 3. Encapsulado de un segmento TCP en un datagrama.



Cada capa tiene su propia estructura de paquete o PDU (*Protocol Data Unit*). Conceptualmente una capa no conoce nada sobre las estructuras de datos usadas en capas superiores o inferiores; en realidad las estructuras de datos se diseñan para ser compatibles con las capas colindantes de forma que resulte una transmisión más eficiente. Aún así, cada capa tiene su propia estructura de datos y su propia terminología para describir esta estructura. La Figura 4 muestra los términos usados por cada capa de TCP/IP para referirse a los datos que son transmitidos. Las aplicaciones en TCP se refieren al conjunto de datos como el *flujo (stream)*, mientras que en UDP se le llama *mensaje*. En TCP a los datos se les llama *segmentos* mientras que en UDP se les llama *datagramas*. La mayoría de las redes hacen referencia a los conjuntos de información que transmiten como *paquetes* o *bloques*.

Figura 4. Terminología de TCP/IP



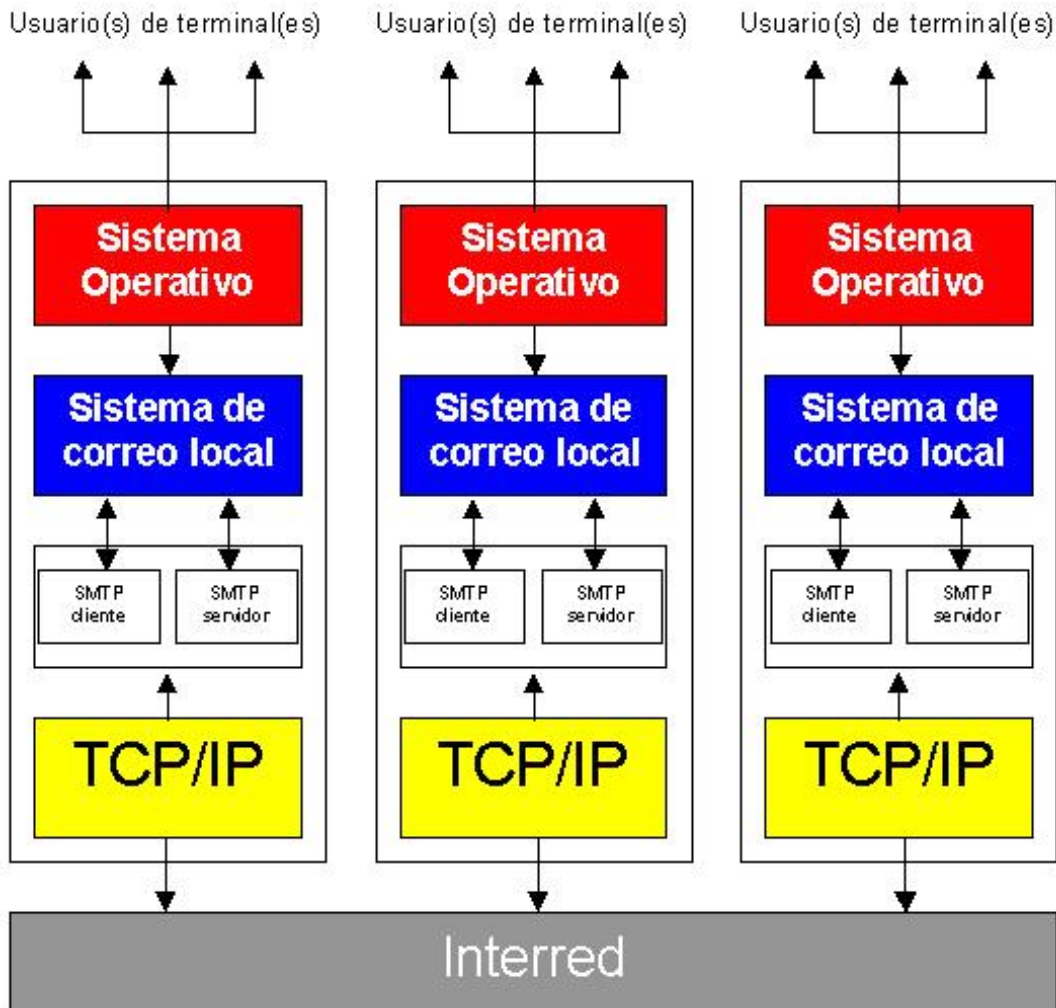
## 2.6.2 PROTOCOLO SIMPLE DE TRASFERENCIA DE MENSAJES SMTP

Es probable que el correo electrónico -el conocidísimo e-mail- sea el servicio que más se usa en términos de redes de computadores. Desde hace muchos años la mayor parte de los sistemas de cómputo interactivos grandes ha contado con sistemas de correo locales; por tanto, la extensión de este servicio a las redes fue una evolución natural cuando estos sistemas se conectaron entre sí.

SMTP (*simple mail transfer protocol*) gestiona la transferencia de correo del sistema de correo de un computador anfitrión a otro. El SMTP no se encarga de aceptar correo de los usuarios locales ni de distribuir el correo recibido a su(s) destinatarios(s); éstas son funciones del sistema de correo local. La interrelación del SMTP y el sistema de correo local se ilustra en la Figura 5.

Puesto que el SMTP interactúa con el sistema de correo local y no con el usuario, está oculto para cualesquiera transferencias de correo locales de esa máquina; sólo se programa su ejecución cuando es necesario enviar correo a una máquina distinta o se recibe correo de una máquina remota. Lo normal es mantener una cola de entrada y una de salida en la interfaz entre el sistema de correo local y los componentes de cliente y servidor del SMTP. El cliente se ocupa de iniciar la transferencia de correo a otro sistema y el servidor se encarga de recibirlo. Aunque en la figura se muestran múltiples usuarios, cabe la posibilidad de que sólo intervenga un usuario que podría estar en un computador personal.

Figura 5. Diagrama SMTP - sistema local de correo



El sistema de correo local mantiene un buzón por cada usuario, en el que este último puede depositar o recibir correo. Cada buzón tiene un nombre único que comprende dos partes: una parte local y una parte global. La primera es el nombre del usuario que es único sólo dentro del sistema de correo local; la segunda, la identidad del sistema anfitrión, debe ser única dentro de toda la Interred. Por lo regular, esta parte consta de varios campos cuyo formato preciso será distinto en los diferentes tipos de establecimiento: educativos, gubernamentales, militares, etc.

Quedan por considerar dos puntos en torno a la transferencia de correo: el formato del correo -para asegurar que se interprete del mismo modo en todos los

sistemas- y el SMTP con que es transferido de una máquina a otra. El formato del correo consiste en una cabecera y un cuerpo que a su vez comprenden varias líneas de texto ASCII; una línea en blanco.

cada línea de la cabecera consta de una palabra clave seguida de un signo de dos puntos y una cadena de texto. Algunas palabras clave son obligatorias, mientras que otras son opcionales, La cabecera mínima es:

TO:

nombre del destinatario

FROM:

nombre del remitente

Una vez creado un mensaje de correo en el formato estándar, el sistema de correo local determina, a partir del nombre del destinatario, si debe depositarlo en un buzón local o en la cola de salida para su reenvío. El SMTP cliente que transferirá el correo determina primero la dirección de IP del sistema anfitrión de destino a partir del servicio de directorio -llamado sistema de nombres de dominio- y utiliza esa dirección, junto con la dirección de puerto bien conocida de SMTP (25) para iniciar el establecimiento de una conexión de transporte con el SMTP servidor del sistema anfitrión de destino. Una vez establecida la conexión, el cliente iniciará la transferencia del correo al servidor.

La transferencia de correo implica el intercambio de las PDU de SMTP denominadas órdenes. Todas las órdenes se codifican como cadenas de caracteres ASCII y comprenden un número de tres dígitos o una orden textual, o ambas cosas. Las órdenes se transfieren por la conexión de transporte establecida mediante las primitivas de usuario SEND/DELIVER (enviar/entregar) de TCP.

Una vez establecida una conexión de TCP, el SMTP servidor devuelve la orden 220 al cliente, con lo que indica que está listo para recibir el correo. El cliente responde enviando una orden Hello junto con la identidad de la máquina cliente. Al recibir esto, el servidor responde con la identidad de la máquina servidora en la

que se registra la transacción de correo. El cliente inicia el envío de la cabecera de correo, que también se confirma con una orden 250; el resto de las líneas de cabecera se envían del mismo modo.

Cuando el cliente envía una orden DATA, queda indicado el inicio del contenido del cuerpo del correo. El servidor responde con una orden 354 y el cliente envía entonces el contenido del correo como una secuencia de líneas terminada con una línea que sólo contiene un punto. El servidor confirma la recepción de esta última con una orden 250. La fase de transferencia de correo termina cuando el cliente envía una orden QUIT (salir) y el servidor devuelve una orden 221, después de lo cual se libera la conexión de transporte.

Las anteriores son las funciones básicas asociadas al SMTP, pero algunas implementaciones disponen de funciones adicionales. Por ejemplo, si el destinatario ya no tiene un buzón en el servidor, el SMTP servidor puede devolver una dirección de reenvío. También, una vez que un SMTP cliente ha enviado su correo, puede invitar al servidor a que envíe correo en el sentido opuesto -si hay correo pendiente- antes de liberar la conexión de transporte.

### **2.6.3 FRAGMENTACIÓN Y REENSAMBLADO DE DATAGRAMAS DE IP**

El tamaño de los datos del usuario a lo que suele llamarse una NSDU asociados a una solicitud de NS\_User (Protocolo de Aplicación) puede ser de hasta 64 K = 65536 octeto (bytes). Los tamaños de paquete máximos(MTU) asociado a los diferentes tipos de red son mucho menores, pues van desde 128 octetos en algunas redes de conmutación de paquetes X25, hasta más de 8000 octetos en algunas LAN. En la tabla 2 encontraremos algunos valores de MTU, para protocolos comunes de nivel de enlace.

Tabla 2. Tamaño de MTU para protocolos comunes de nivel de enlace

Protocolo al nivel de enlace	MTU (bytes)
PPP(valor por defecto)	1500
PPP(bajo retardo)	296
SLIP	1006 límite original
X.25	1600 (RFC 1356)
Frame Relay	1600 (Depende de la red)
SMDS	9235
Ethernet DIX	1500
Ethernet LLC-SNAP	1492
IEEE 802.4/802.2	8166
Token Ring 16Mbps	17940
Token Ring 4Mbps	4440
FDDI	4352
Hyperchannel	65535
Clasical IP aver ATM	9180

Las funciones de fragmentación y reensamblado asociadas al IP dividen los datos del usuario asociada a una solicitud NS\_User en fragmentos más pequeños (segmento en la terminología de la ISO), de modo que puedan ser transferidos a través de una red específica en datagramas del tamaño apropiado. Al recibir los fragmentos de datos relacionados con la misma NSDU y contenidos en cada datagrama de IP, el IP reensambla la NSDU antes de pasarla al NS\_User de destino.

Se puede adoptar una de dos posibles estrategias, ya que el tamaño máximo de paquete puede variar de una red a otra:

Las funciones de fragmentación y reensamblado se pueden efectuar en cada red *fragmentación Intrared* o bien de extremo a extremo *fragmentación Interred*.

En general, el IP de un sistema anfitrión sólo conoce el tamaño máximo de paquete asociado a subred local. De manera similar, el IP de cada pasarela solo conoce los tamaños máximos de paquetes asociados a las dos redes a las que está conectada. Con la fragmentación Intrared, el IP del sistema anfitrión de origen fragmenta los datos de NS\_User (la NSDU) en varios datagramas direccionados individualmente según lo prescrito por la red a la que está

conectado el sistema anfitrión, y luego inicia la transmisión de estos datagramas, ya sea al sistema anfitrión de destino o al primer Gateway (pasarela de la ruta) mediante el SNDCCP, el cual obtiene la dirección de NPA del sistema anfitrión o pasarela.

Al recibir cada datagrama, el IP del sistema anfitrión o pasarela reensambla la NSDU y luego la refragmenta para producir un conjunto de datagramas direccionados en forma individual y posiblemente distintos, según lo señalado por el tamaño máximo de paquetes de la siguiente red.

Este procedimiento se repite en cada una de las pasarelas hasta que el datagrama llegue al IP del sistema anfitrión de destino, donde la NSDU se volverá a reensamblar y se pasará al NS\_User de destino.

Con la fragmentación Interred, el IP del sistema anfitrión de origen realiza el mismo procedimiento anterior de fragmentación y envía los datagramas resultantes al IP de la primera pasarela. No obstante, esta vez el IP no reensambla la NSDU; en vez de ello, modifica los campos apropiados y envía los datagramas recibidos directamente por la segunda red (si es que ésta puede manejar datagramas con este tamaño) o bien refragmenta el datagrama en fragmentos (datagramas más pequeños).

Este procedimiento se repite en la pasarela siguiente. Si embargo, si en la última red/subred puede manejar paquetes de tamaño mayor que el de los datagramas que recibe, éstos se reenviarán directamente después de realizarse ciertas modificaciones en algunos campos de la cabecera. Al igual que con la fragmentación Intra-red, el IP del sistema anfitrión de destino reensambla los datos de usuario de cada datagrama que recibe y pasa la NSDU resultante al NS\_User de destino.

Podemos deducir, sobre todo a partir de los flujos de paquetes asociados a la tercera red, que la fragmentación Intrared permite usar el tamaño de paquetes máximo de cada red, ya que los fragmentos individuales se reensamblan en cada pasarela de la ruta. Lo que no ocurre con la fragmentación Interred, pero se tiene la ventaja de que no es preciso reensamblar en cada pasarela.

*De hecho, en el IP se usa la fragmentación Interred.* A primera vista quizá esto parezca sorprendente pero se emplea a causa del problema de la pérdida de datagramas. Algunas redes operan con un protocolo sin conexión del mejor intento con la posibilidad de que uno o más datagramas relacionados con una sola NSDU se alteren durante la transmisión. Si se usa fragmentación Intrared, el IP receptor de cada pasarela reensamblará la NSDU completa antes de reenviarla a la siguiente red. Si faltan fragmentos, en el caso de que un datagrama sea desechado por alguna alteración, el IP receptor deberá decidir en qué punto cancelará la función de reensamblado.

El IP del sistema anfitrión de origen determina esto definiendo un límite máximo para el tiempo que una pasarela puede esperar datagramas relacionados con una NSDU durante cada operación de reensamblado. Éste límite, el tiempo de vida, se lleva en la cabecera de todos los datagramas relacionados con la NSDU; lo establece el IP del sistema anfitrión de origen y posteriormente cada IP que procesa el datagrama lo decrementa. Si un datagrama se fragmenta, el valor actual se copiará en la cabecera de los nuevos datagramas, si el valor llega a cero en cualquier punto del proceso durante el reensamblado en una pasarela (o sistema anfitrión), se interrumpirá la función de reensamblado y todos los fragmentos relacionados con esa NSDU serán desechados.

El campo de tiempo de vida de cada datagrama contiene múltiplos de un segundo, así que la cantidad en la que cada IP lo decrementa variará dependiendo del retardo de tránsito medio conocido de la red correspondiente. En el caso de la fragmentación Interred, el IP de cada pasarela disminuirá también el campo de tiempo de vida de cada datagrama que reciba y se desearán todos los datagramas en los que el valor llegue a cero.

El IP del sistema anfitrión de destino abortará su operación de reensamblado basado en el mismo criterio. En ambos casos, si faltan fragmentos y la operación de reensamblado se aborta, se generará un mensaje de error de *tiempo excedido* y se devolverá al IP del sistema anfitrión de origen.

#### **2.6.4 DIRECCIONAMIENTO IP**

Cada interfaz de red de cada host o router en un red IP se identifica mediante, por lo menos una, dirección única -dentro de dicha red- de 32 bits; representada por cuatro número decimales separados por punto y comprendido entre 0 y 255, que equivalen al valor de cada uno de los cuatro bytes que componen la dirección. Por ejemplo una dirección IP válida sería 10.220.0.20.

Si un nodo dispone de varias interfaces físicas, como por ejemplo un router, cada una deberá tener necesariamente una dirección IP distinta si desea ser accedido de forma diferenciada por este protocolo

Las direcciones IP tienen una estructura jerárquica, una parte de la dirección corresponde a la red (*identificador de red -netid-*) y la otra al host (*identificador del sistema anfitrión - hostid-*) dentro de la red. Así, cuando un router recibe un paquete por una de sus interfaces compara la parte de la dirección que identifica la red a la que va dirigido el paquete con las entradas contenidas en sus tablas y reenvía el paquete por la interfaz correspondiente.

La siguiente tabla muestra la clasificación de las redes según el rango de direcciones IP comprendidos por estas.

Tabla 3. clasificación de las redes según el rango de direcciones IP comprendidos por estas.

CLASE	PRIMERA IP	ULTIMA IP	Nº HOST POR RED
A	1.0.0.0	126..255.255.255	167,777,216
B	128.0.0.0	191.255.255.255	65,536
C	192.0.0.0	223.255.255.255	256
D	224.0.0.0	239.255.255.255	N/A
E	240.0.0.0	247.255.255.255	N/A

Las direcciones de clase D se utilizan para definir grupos multicast, estas direcciones no deben aparecer como dirección de origen de un paquete.

Las direcciones de clase E no se utilizan de momento, y están reservadas para usos posteriores.

Para indicar qué parte de la dirección corresponde a la red y qué parte al host, se utiliza una notación denominada *máscara*, consiste en poner a unos (255) los bits que corresponden a la red y a 0 los que corresponden a la parte del host. Así por ejemplo, una red clase A tiene una máscara 255.0.0.0, lo que quiere decir que los ocho primeros bits especifican la red y los 24 restantes el host. De igual modo, una red clase B tiene una máscara de 255.255.0.0 y una red de clase C una máscara de 255.255.255.0. Las máscaras permiten extraer de forma sencilla la parte de la red o de host de una dirección.

Existen algunas reglas y convenios que asignan significados especiales a determinadas direcciones IP:

La dirección 255.255.255.255 se utiliza para indicar broadcast en la propia red.

La dirección 0.0.0.0 identifica al host actual. Sólo se puede utilizar como dirección de origen.

Las direcciones con el campo de host en cero, identifican redes y por tanto no se utilizan para ningún host.

Una dirección con todos los campos de host en uno se utiliza como dirección de broadcast dentro de la red.

Una dirección con el campo de red con todos los bits en cero identifica un host en la propia red.

Las redes 10.0.0.0 (clase A), 172.16.0.0 a 172.31.0.0 (clase B) y 192.168.0.0 a 192.168.255.0 (clase C) están reservadas para redes privadas o intranets.

Como consecuencia de las reglas 3 y 4, siempre hay dos direcciones inútiles en una red, la primera y la última. Por ejemplo, en la red 152.74.21.0 (clase B) se tiene que reservar la dirección 152.74.21.0 para identificar la red y la dirección 152.74.21.255 para envíos broadcast a toda la red. Por lo tanto se dispondrá de 254 direcciones para host no 256.

#### **2.6.4.1 DIVISIÓN EN SUBREDES**

Existe un mecanismo que permite que permite dividir una red IP en trozos de subred, de forma que, por ejemplo, una empresa puede solicitar un clase B y asignar fragmentos de dicha red a cada oficina a medida que se fueran incorporando a la red. Esto equivale a crear un nivel jerárquico intermedio entre la red y el hos, permitiendo grados de agrupación.

Para dividir la red en subredes, se define una nueva máscara. Los bits en uno (255) de la máscara, identifican la parte de la red y subred y los bits en cero corresponden al host.

Cuando se crean subredes hay dos direcciones en cada subred que quedan reservadas (inutilizadas): las que corresponden al campo hos todos en ceros - identifica la subred- y todo en uno -broadcast dentro de la red-.En la siguiente tabla se muestran los valores de máscaras permitidas y el número de host que permiten estas máscaras.

Tabla 4. valores de máscaras permitidas y el número de host que permiten estas máscaras.

Valor de máscara	Direcciones Permitidas
0	254
128	126
192	62
224	30
240	14
248	6
252	2
254	0
255	0

Así, la dirección IP 152.74.21.0 con máscara 255.255.128.0 quiere decir lo siguiente:

Identificador de red 152.74.0.0

Primera dirección válida: 152.74.0.1

Última dirección válida: 152.74.127.254

Dirección de broadcast: 152.74.127.255

Número máximo de host:  $(128*256)-2 = 32766$

### 2.6.5 TRANSFERENCIA DE DATOS CON TCP

Generalidades, El UDP se usa cuando no se requiere corrección de errores o bien para un solo intercambio de mensajes cortos de solicitud/respuesta entre dos protocolos de aplicación. Sin embargo, en casi todas las aplicaciones distribuidas abiertas se necesita un servicio de transporte de mensajes confiable (orientado a conexión). Un ejemplo es la transferencia del contenido de un archivo que contiene información confidencial de un usuario en una aplicación de correo

dentro del dominio de una empresa de investigación, o un registro bancario. Queda claro que en tales aplicaciones, incluso la alteración de un solo bit binario tiene una importancia primordial.

En el conjunto TCP/IP, el protocolo de transporte orientado a conexión es el protocolo de control de transmisión (**TCP**), y el servicio que ofrece a los usuarios - a través de los protocolos de aplicación- se denomina servicio de flujo confiable.

TCP es un protocolo de nivel de transporte completo que proporciona un servicio de transferencia fiable de datos y un método para trasladar datos encapsulados con TCP a un protocolo de nivel de aplicación.

TCP tiene las siguientes características.

*Orientado a conexión.* Antes de transferir los datos, dos procesos de nivel de aplicación deben negociar formalmente una conexión TCP utilizando el proceso de establecimiento de conexión adecuado. Las conexiones de TCP se cierran formalmente empleando el proceso de desconexión TCP.

*Full Duplex.* Para cada extremo de una conexión TCP, la conexión consta de dos enlaces lógicos, uno de salida y otro de entrada. Con la tecnología apropiada en el nivel de red los datos pueden fluir simultáneamente en ambos sentidos. La cabecera TCP contiene tanto el número de secuencia de los datos de salida como el reconocimiento de los datos de entrada.

*Fiable.* Los datos que se envían por una conexión TCP se numeran en secuencia y se espera un reconocimiento positivo por parte del receptor. Si no se recibe este reconocimiento, el segmento se retransmite. En el receptor, los segmentos duplicados se descartan y los segmentos que llegan fuera de secuencia se colocan en su posición dentro de la secuencia. Todo segmento transmitido va protegido frente a errores mediante un código detector (CRC), que verifica la integridad de la información recibida.

*Flujo de bytes.* Para TCP los datos que se envían por los enlaces lógicos de entrada y salida se consideran un flujo continuo de bytes. El número de secuencia

y de reconocimiento que se envían en cada cabecera TCP definen puntos concretos de este flujo de bytes. TCP no tiene en consideración otras divisiones dentro del flujo de datos, siendo el protocolo de aplicación el que establezca las divisiones lógicas adecuadas (por ejemplo, fin de registro ó de campo en bases de datos, fin de orden, etc.)

*Control de flujo en ambos extremos.* Para evitar la transmisión de excesivos datos simultáneos, que podría causar problemas de congestión en los routers, TCP implementa un control de flujo en el emisor que regula la cantidad de datos que se envían. Para evitar que el emisor transmita datos que el receptor no es capaz de almacenar, TCP también implementa control de flujo en el receptor, indicando cuánto espacio se encuentra disponible en los buffers del receptor.

*Segmentación de datos de aplicación.* TCP segmentará los datos obtenidos del proceso de aplicación para que se ajusten al tamaño de los paquetes IP. Ambos extremos TCP pueden negociar el tamaño máximo de segmento, existiendo además la posibilidad de ejecutar un algoritmo de descubrimiento del tamaño máximo en la ruta (PMTU).

*Transmisión uno a uno.* Las conexiones TCP son un circuito lógico punto a punto entre dos procesos de nivel de aplicación. TCP no proporciona servicios de difusión. TCP emplea el mecanismo de los **puertos** para el acceso a diferentes destinos dentro de un Host. Al igual que UDP, la utilización de los puertos permite que un cliente seleccione el servidor correspondiente a la aplicación deseada dentro del computador destino.

*Fases de TCP [Modelo de operación]* Para identificar los distintos flujos de datos que un módulo TCP puede manejar simultáneamente, TCP proporciona un identificador de puerto. Como los identificadores de puertos son seleccionados de forma independiente por cada TCP, puede que no sean únicos. Para disponer de direcciones únicas dentro de cada TCP, se concatena la dirección de Internet que identifica al módulo de TCP con el identificador de puerto para así conformar una dirección de conector ('socket') que será única a largo de todo el conjunto de redes interconectadas.

Una conexión queda completamente especificada por el par de conectores de sus extremos. Un conector local puede participar en muchas conexiones con diferentes conectores foráneos. Una conexión puede ser utilizada para transportar datos en los dos sentidos, es decir, es bidireccional ('full duplex').

Los módulos de TCP pueden asociar libremente los puertos a procesos. Sin embargo, algunos conceptos básicos son necesarios en cualquier implementación. Debe haber un conjunto de conectores públicos que el módulo de TCP asocie sólo con los procesos "apropiados" por algún medio. Se supone la posibilidad de que los procesos puedan "poseer" puertos, y que los procesos puedan iniciar conexiones sólo con los puertos que ellos posean.

Para garantizar que no se pierde ningún bloque (segmento) de datos, una sesión TCP consta de tres fases.

**2.6.5.1 Establecimiento De Conexión:** Durante esta fase los hosts origen y destino determinan unos parámetros necesarios para el intercambio de datos, como son el número inicial de secuencia, los tamaños de buffer necesarios, etc., creando un marco para el intercambio fiable de información.

Puede establecerse una conexión entre dos máquinas sólo si no existe una conexión entre los dos sockets, ambas máquinas están de acuerdo en la conexión y ambas máquinas tienen recursos TCP adecuados, para servir la conexión. La aceptación de conexiones la puede desencadenar una aplicación o una rutina de administración del sistema.

Cuando se establece una conexión se le dan ciertas propiedades que son válidas hasta que la conexión se cierra. De manera común, éstas son un valor de precedencia y un valor de seguridad. Estos parámetros los acuerdan las dos aplicaciones cuando la conexión está en proceso de establecerse.

En la mayoría de los casos, una conexión la esperan dos aplicaciones, así que emiten solicitudes abiertas ya sea activas o pasivas. Aunque en una interacción

cliente servidor el cliente siempre inicia el establecimiento de una conexión, en aplicaciones más generales que no se basan en el modelo cliente servidor es posible que ambas partes traten de establecer una conexión al mismo tiempo. En prevención de esto, las conexiones se establecen mediante un intercambio de mensajes (segmentos) de tres pasos denominado procedimiento de *saludo de tres pasos*. Además, el flujo de datos en cada sentido de una conexión se controla en forma independiente. A fin de evitar cualquier ambigüedad con los valores de número de secuencia inicial en ambos lados de una conexión, cada lado informa al otro el número de secuencia inicial que se propone usar, y estos números se confirman como parte del procedimiento de saludo.

*Cliente servidor:* El lado iniciador establece una conexión enviando un segmento con la bandera SEQ en 1 y el número de secuencia inicial propuesto en el campo de número de secuencia,  $seq = X$ . En un caso real el valor de  $X$  es un número que elige cada entidad TCP para iniciar sus valores de secuencia. En cuanto recibe este segmento, el lado que responde toma nota del valor del número de secuencia para el sentido entrante y luego devuelve un segmento con las banderas SEQ y ACK en 1, su propio valor asignado para el sentido opuesto ( $seq = Y$ ) en el campo de número de secuencia, y el valor  $X+1$  ( $ack = X+1$ ) en el campo de confirmación para indicar que ya tomó nota del valor inicial para su sentido entrante. Al recibir esto, el lado iniciador toma nota de  $Y$  y devuelve un segmento con únicamente la bandera ACK puesta en 1 y el valor  $Y+1$  en el campo de confirmación.

*Posibilidad de colisión:* En el caso que ambos lados enviaran un segmento SEQ al mismo tiempo, cada lado se limitaría a devolver un segmento ACK confirmando el número de secuencia apropiado. así quedan establecidos ambos lados de la conexión y pueden comenzar a transmitir datos en forma independiente.

**2.6.5.2 Transferencia de Datos:** La información viaja en ambos sentidos fraccionada en *segmentos* de datos. En esta fase entran en juego los mecanismos de detección y corrección de errores, control de flujo y control de la congestión.

Los procedimientos de control de errores y de flujo asociados a la fase de transferencia de datos se basan en una estrategia de control de errores de retroceder  $N$  y en un mecanismo de control de flujo de ventana deslizante. Consideraremos el ejemplo de un intercambio de segmentos típico relacionado con un intercambio de mensajes de solicitud/respuesta cliente servidor. Suponemos que el cliente emite un mensaje de solicitud corto de  $N$  octetos y, para forzar su entrega, pone en 1 el parámetro de bandera de empujar. El TCP transmisor envía este mensaje directamente en un segmento con las banderas SEQ y PSH en 1 y el valor actual del número de secuencia, que suponemos es  $X$ . Al recibir el segmento, el lado receptor se da cuenta de que la bandera PSH es 1 y, por tanto, entrega el mensaje directamente y devuelve un segmento con la bandera ACK en 1, un número de confirmación de  $X+N+1$  *que indica el siguiente octeto que espera recibir* y un valor de ventana de  $N$  para volver la ventana a su valor original.

El servidor devuelve un mensaje de respuesta que suponemos para el ejemplo que tiene tres veces el tamaño máximo de segmento  $W$ , de modo que el TCP local debe enviar el mensaje en tres segmentos. A fin de ilustrar el mecanismo de ventana suponemos que la ventana de transmisión es igual a  $2W$ , o sea, que es suficiente para enviar dos segmentos de tamaño máximo. Después de enviar dos segmentos, el servidor deberá esperar hasta recibir una confirmación con una asignación de crédito adicional en su campo de ventana.

Cuando el TCP cliente recibe cada uno de los segmentos, devuelve un segmento con la bandera ACK en 1; al recibir el segundo segmento, el TCP entrega el contenido actual de su buffer (los dos segmentos) al AP cliente. En el lado del servidor, con la recepción del primer segmento ACK, el TCP servidor incrementa su valor de crédito y procede a enviar el tercer segmento. Se supone que el TCP receptor entrega este tercer segmento directamente a la aplicación cliente.

Aunque esto no se muestra, si se recibe un segmento con un campo de secuencia o de confirmación incorrecto, se devuelve un segmento con las banderas RST y SEQ/ACK en 1 y el o los valores esperados en el o los campos apropiados.

*Nota:* TCP soporta una función de envío (push) de los protocolos de capa superior. Un envío se usa cuando una aplicación desea enviar datos de inmediato y confirmar que el mensaje pasado al TCP se ha transmitido con éxito. Para hacer esto se establece una bandera de envío en la conexión entre el protocolo de aplicación y TCP, instruyendo al TCP para adelantar cualquier información en el búfer de la aplicación al destino lo más pronto posible (lo contrario de conservarla en el búfer hasta que esté lista para transmitirse).

**2.6.5.3 Cierre de Conexión:** Tras el intercambio de información, y a propuesta de uno de los extremos, se intenta una desconexión negociada, donde no queden datos sin entregar por ninguna de las dos partes.

Para cerrar una conexión, uno de los TCP recibe un primitivo de cierre del protocolo de la capa de aplicación y emite un mensaje con la bandera FIN activada.

A continuación se presentan las acciones de cada entidad de TCP en respuesta a los dos métodos de terminación de conexión alternativos.

*Desconexión elegante:* Es este caso suponemos que el protocolo cliente ya terminó de enviar todos sus datos y simplemente desea terminar la conexión. Entonces, al recibir la primitiva CLOSE, el cliente envía un segmento con la bandera FIN en 1. Al recibir este segmento, el servidor emite una primitiva CLOSING al protocolo servidor y devuelve un segmento ACK al cliente para confirmar la recepción del segmento FIN. Suponemos que el protocolo servidor también terminó de presentar datos y por tanto emite una primitiva CLOSE en respuesta a la primitiva CLOSING. Sin embargo, en el ejemplo suponemos que el TCP servidor todavía tiene datos pendientes por transmitir, que envía en un

segmento con las banderas SEQ y FIN en 1. Al recibir este segmento, el TCP cliente emite una primitiva TERMINATE y devuelve un ACK por los datos que acaba de recibir. Al recibir este ACK, el TCP servidor emite su propia TERMINATE al protocolo servidor. Como alternativa, si el TCP servidor no recibe un ACK dentro de un lapso igual al doble del periodo de tiempo de vida, (TTL) supone que el ACK fue alterado y emite una TERMINATE al servidor.

*Abortar:* En la secuencia de aborto, el lado del cliente termina de inmediato ambos lados de la conexión y envía un segmento con la bandera RST en 1. Al recibir este segmento, el servidor termina ambos lados de la conexión abruptamente y emite una primitiva TERMINATE con el código de razón correspondiente a conexión abortada.

## 2.6.6 ENCAMINAMIENTO DE DATAGRAMAS IP

Como cada red (o subred) de una interred puede usar diferentes tipos de direcciones de punto de conexión, un sistema - sistema anfitrión o pasarela-conectado a una red sólo puede enviar un datagrama directamente a otro sistema si está conectado a la misma red. Para enrutar datagramas a través de múltiples redes, el IP de cada pasarela de interred deberá conocer la dirección de punto de conexión del sistema anfitrión de destino -si está conectado a una red a la que está conectada la pasarela- o bien la siguiente pasarela a lo largo de la ruta hacia la red de destino deseada, en caso contrario. De nuevo, la siguiente pasarela debe estar conectada a una red a la que esté conectada esta pasarela. La principal dificultad del enrutamiento consiste en cómo los anfitriones y las pasarelas de la Interred consiguen y mantienen su información de enrutamiento.

Son dos los enfoques básicos para el enrutamiento de una Interred: el centralizado y **el distribuido**. Con un esquema de *enrutamiento centralizado*, la información de enrutamiento asociada a cada pasarela se recibe de un sitio central por medio de la red y de mensajes de gestión de red especiales. El sistema de gestión de red procura mantener actualizado su contenido a medida que se agregan y eliminan

redes y anfitriones, y a medida que se diagnostican y reparan fallos. En general, salvo en las interredes más pequeñas, ésta sólo será una solución viable si cada red individual tiene su propio sistema de gestión de red que cuente con procedimientos avanzados de configuración y control de fallos.

Con un esquema de *enrutamiento distribuido*, todos los anfitriones y pasarelas cooperan en forma distribuida para asegurar que la información de enrutamiento que cada sistema -anfitriones y pasarelas- tiene sea actual y consistente. Cada sistema mantiene información de enrutamiento en forma de una *tabla de enrutamiento* que contiene la dirección del punto de conexión a la red (NPA) con que se reenviará cada datagrama. La Internet se vale de un esquema de este tipo. El procedimiento de enrutamiento asociado al IP lee primero la dirección del IP del punto de conexión a la subred (SNAP) de destino contenida en el datagrama y con ella obtiene la dirección de punto de conexión correspondiente -de un sistema anfitrión o una pasarela- de la tabla de enrutamiento. Además, con una serie de protocolos de enrutamiento se crea y mantiene el contenido de cada tabla de enrutamiento en forma distribuida.

*Sistemas autónomos:* Para reflejar el hecho de que la Internet se compone de varias interredes gestionadas y operadas en forma independiente, cada interred se trata como un sistema autónomo con sus propios algoritmos de enrutamiento y autoridad de gestión interna. La Internet combinada se considera una red fundamentalmente central a la cual están conectadas varios sistemas autónomos. Con los términos de pasarela interior y pasarela exterior distinguimos, respectivamente, las pasarelas que usan dentro de un sistema autónomo y las que sirven para conectar un sistema autónomo a la red central. Los protocolos de enrutamiento correspondientes son el protocolo de pasarela interior (IGP) y el protocolo de pasarela exterior (EGP). Como la Internet consiste en un conjunto interconectado de interredes, cada sistema autónomo tiene su propio IGP . No obstante, el EGP de Internet es, y debe ser una norma válida para toda la Internet. Si todas las pasarelas y sistemas anfitriones de una interred tuvieran una entrada individual en su tabla de enrutamiento para todos los demás sistemas, el tamaño

de las tablas y la capacidad de procesamiento y transmisión necesaria para mantener las tablas sería excesivo y, en el caso de la Internet, imposible de manejar. En vez de ello la información de enrutamiento se organiza jerárquicamente como sigue:

- ❖ Los anfitriones mantienen suficiente información de enrutamiento para reenviar datagramas a otros anfitriones o a una o más pasarelas internas conectados a la misma red.
- ❖ Las pasarelas internas mantienen suficiente información de enrutamiento para reenviar datagramas a anfitriones u otras pasarelas internas dentro del mismo sistema autónomo.
- ❖ Las puertas externas mantienen suficiente información de enrutamiento para reenviar datagramas a una pasarela interior, si el datagrama es para el mismo sistema autónomo, o para otra pasarela exterior, en el caso contrario.

Se han desarrollado varios protocolos de enrutamiento con miras a implementar este esquema. Entre ellos están el protocolo intrared denominado protocolo de resolución de direcciones (ARP), varios protocolos de puerta interior (IGP) y un protocolo de pasarela exterior (EGP).

*Protocolo de Pasarela Interior:* Este protocolo puede variar de un sistema autónomo a otro. El protocolo más utilizado es el protocolo de información de enrutamiento (RIP). Se trata de un protocolo de enrutamiento distribuido basado en la técnica llamada algoritmo de vector de distancia (DVA). Un protocolo de introducción más reciente se basa en dos algoritmos conocidos como algoritmo de estado de enlace (LS) y algoritmo de camino más corto primero (SPF). Se ha adoptado el protocolo de abrir primero el camino más corto de estado de enlace (OSPF de estado de enlace) como norma internacional para el CLNP de la ISO. Puesto que el DVA es específico de TCP/IP, analizaremos aquí el OSPF.

El objetivo del protocolo es determinar el contenido de la base de información de enrutamiento de cada GW, es decir, los GW vecinos (adyacencias) con que se

enrutarán los PDU (una unidad de datos de protocolo). Sin embargo, antes de que pueda ponerse en marcha el algoritmo de búsqueda de rutas, es necesario que cada GW establezca el par NET/SNPA (la identidad del GW y de la red/subred) de cada uno de sus vecinos. Si las redes son de difusión, cada GW difunde una PDU de GW a GW por cada una de sus subredes a través de la dirección de destino a todos los GW. Esta PDU contiene el identificador del GW de origen, y cada SNDAP (protocolo de acceso dependiente de la subred) que la recibe pasa la dirección de SNPA correspondiente (en esta subred) al protocolo GW a GW como parámetro con el SNDCP (protocolo de convergencia dependiente de la subred) intermedio. En el caso de subredes sin difusión, el gestor de la red debe introducir la PDU GW a GW. En cualquier caso esta información se almacenará en una base de datos aparte, denominada *base de datos de adyacencias*. Una vez hecho esto el protocolo GW a GW puede participar en el procedimiento de deducción del costo del camino más corto.

Recordemos que el gestor de red introduce las identidades de las subredes conectadas a un GW, junto con sus métricas de costo, en una tabla (la base de datos de circuitos). A intervalos regulares, cada GW (en la práctica) envía a cada uno de sus vecinos una PDU de estado de enlace (LS: link state) que contiene el NET del GW y una lista de los identificadores de las subredes a las que está conectado el GW, junto con los valores de costo (uno para cada métrica) asociados a cada subred. De esta forma, un GW recibirá una PDU de estado de enlace de cada uno de sus vecinos informándole de las subredes (enlaces) conectados a él y de sus valores de costo. Esta información se almacenará en la base de datos de estados de enlace mediante un proceso de actualización del protocolo de Interred (SNICP).

Una vez que ha hecho, el proceso de actualización de cada GW enviará una copia de la PDU de estado de enlace a cada uno de sus vecinos (excepto al que envió la PDU). El resultado es que cada GW recibirá un conjunto adicional de PDU de estado de enlace que, de hecho se originaron en los vecinos de sus vecinos. Este procedimiento continúa, de modo que durante cierto lapso de tiempo, cada GW recibirá un conjunto completo de PDU de estado de enlace que contienen las

identidades de las subredes -y sus valores de costo de camino (uno por cada métrica)- conectados a todos los demás GW de la Interred. Este tipo de procedimiento de enrutamiento se denominará **inundación**.

Cada vez que se introduce un nuevo conjunto de PDU de estado de enlace en la base de datos de estados de enlace, se ejecuta un procedimiento de decisión cuya función es, en primer término, aplicar el algoritmo SPF a la base de datos de estados de enlace y, en segundo término, determinar a partir de todas las entradas de las distintas bases de datos cuál GW vecino debe usarse para llegar a cada uno de los otros GW, con base en los valores de costo de camino correspondientes. Esto se hace por turno para cada métrica de enrutamiento.

*Algoritmo de enrutamiento:* Describiremos el algoritmo de enrutamiento conocido como *algoritmo del camino más corto primero*. Con que los GW enrutan las PDU por la Interred. Existen varios algoritmos de este tipo, pero el que utiliza la ISO es el algoritmo de **Dijkstra**, que se aplica ya en varias redes grandes, entre ellas la ARPANET.

Puesto que los GW se encargaran de todo el enrutamiento, para fines de enrutamiento una Interred puede verse simplemente como una comunidad distribuida de GW interconectados mediante enlaces (lógicos). En la práctica, los enlaces pueden ser un camino a través de una LAN o WAN o un enlace directo punto a punto. Cada enlace utiliza varias métricas de enrutamiento, cada una de las cuales tiene asociado un valor de costo. Estas métricas son:

- *Capacidad:* Una medida de volumen de transmisión del circuito en bits por segundo; un valor alto indica una capacidad más baja. En la práctica, ésta es la métrica que se usa por omisión.
- *Retardo:* Este se relaciona con el retardo de tránsito medio asociado a un enlace (subred) y, por tanto, incluye los retardos de encolado en puentes y

centrales, una vez más, un valor alto indica un retardo de tránsito más largo.

- *Gasto*: Una medida del costo en dinero implicado en un enlace; un valor alto indica un mayor costo monetario. Desde luego, si se puede utilizar una subred privada, esto será preferible a usar, digamos, un enlace a través de una sub red conmutada de portadora pública.
- *Error*: Una medida de la probabilidad residual de errores asociada al circuito, los valores altos indican una mayor probabilidad de errores no detectados.

Debemos saber que todas estas métricas se relacionan con parámetros de la QoS, así que ésta nos servirá para seleccionar la métrica de enrutamiento.

Con el término 'costo de camino' indicamos el costo total acumulativo de los enlaces que se usan en un camino específico entre un par dado de host a través de la Interred. Por tanto, el costo de camino asociado a una ruta puede diferir dependiendo de la métrica empleada, así que el término 'costo del camino más corto' se referirá a un camino empleando una sola métrica de enrutamiento.

La mejor manera de describir el algoritmo del camino más corto primero es considerar la topología de la Interred que presentamos a continuación, para apreciar como se desarrolla el algoritmo de Dijkstra.

## **2.6.7 DETECCIÓN DE ERRORES**

Durante la transmisión de datos entre dos equipos terminales de datos (DTE) es muy común, en especial si las líneas de transmisión están en un entorno eléctricamente ruidoso, que las señales eléctricas que representa el flujo de bits transmitido sufran cambios a causa de la interferencia electromagnética que los dispositivos eléctricos cercanos inducen en las líneas de transmisión. Como

resultado de lo anterior, el receptor podría interpretar señales que representan un 1 binario como señales de 0 binario y viceversa. Para asegurar que la información recibida por un DTE destino tenga una alta probabilidad de ser idéntica a la que envió el DTE transmisor, el receptor debe contar con un mecanismo, que le permita deducir con suficiente probabilidad, cuando la información recibida contiene errores, para lograr esto se emplean dos estrategias:

⊕ **El control de errores hacia adelante**, en el que cada carácter o trama transmitido contiene información adicional (redundante) que permite al receptor, no sólo determinar la presencia de errores, sino también detectar en qué punto del flujo están dichos errores. Así los datos correctos se obtienen invirtiendo los bits en cuestión.

⊕ **El control de errores por retroalimentación (retrospectivo)**, en el que cada carácter o trama incluye sólo la información adicional para que el receptor pueda saber si se presentan errores, pero no su ubicación exacta. Se utiliza un esquema de control de retransmisión para solicitar el envío de otra copia de la información que se espera, será correcta.

El número de bits requeridos para lograr un control de errores hacia adelante, crece conforme aumenta el número de bits de información. Por esta razón, el control de errores por retroalimentación es el método predominante en los sistemas y redes de comunicación de datos.

El control de errores por retroalimentación se puede dividir en dos partes:

- ⊕ Las técnicas que se emplean para lograr una detección de errores confiable y
- ⊕ los algoritmos de control con que se realizan los esquemas de control de retransmisión asociados.

En éste laboratorio se centra únicamente en las técnicas de detección de errores más comunes.

Los dos factores que determinan que determinan el tipo de esquema para detección de errores son la *la tasa de errores de bit (BER: bit error rate)* de la línea o circuito y el tipo de errores, es decir, si los errores ocurren aleatoriamente en bits aislados o en grupos contiguos de bits (a esto se le conoce como ráfagas de errores). La BER es la probabilidad P de que un solo bit este alterado en un intervalo de tiempo definido, a sí, una BER de  $10^{-3}$  significa que, en promedio, un bit de  $10^3$  se alterará durante un periodo de tiempo definido.

Si se están transmitiendo caracteres individuales empleando transmisión asíncrona (ocho bits por carácter más uno o dos bits de inicio y otro de paro), la probabilidad de que un carácter esté alterado es aproximadamente  $10^{-2}$ . O bien, si se están transmitiendo bloques de caracteres por transmisión síncrona (digamos 125 caracteres por bloque, con ocho bits por carácter), la probabilidad de que un bloque (trama) contenga un error es aproximadamente 1, lo que significa que, en promedio cada bloque contendrá un error y habrá que retransmitirlo. Esta longitud de trama es demasiado grande para y deberá ser reducida para que el rendimiento sea aceptable.

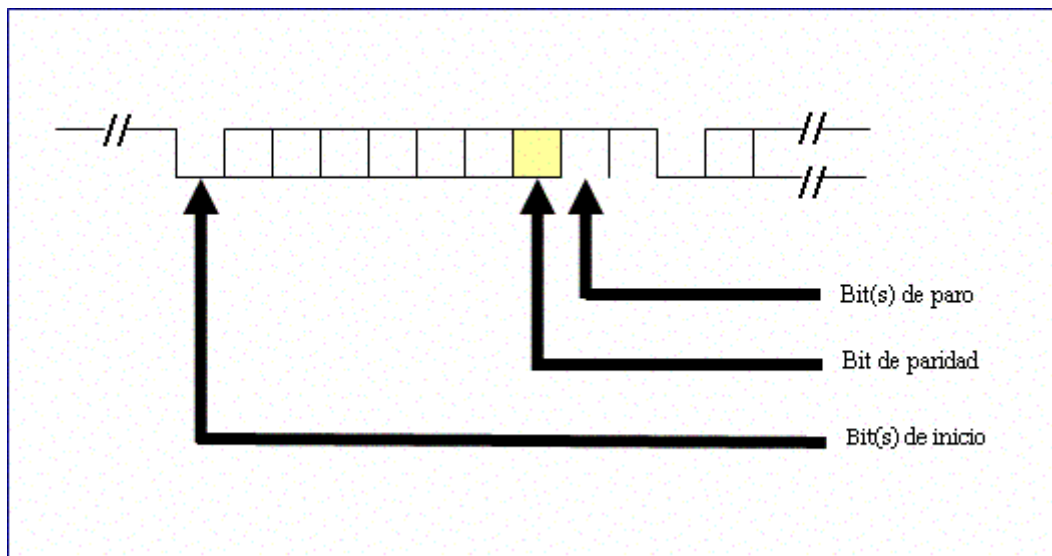
Los tres esquemas más utilizados son los de paridad, verificación de suma de bloque y verificación de redundancia cíclica. En esta herramienta son presentados los dos primeros métodos de detección.

**2.6.7.1 PARIDAD** El método de bit de paridad es el más común para detectar errores de bit con transmisión síncrona y asíncrona orientada a caracteres. En este esquema el transmisor agrega un bit adicional - el de paridad- a cada carácter antes de transmitirlo. El bit de paridad se calcula en base los bits que constituyen el carácter transmitido. Al recibir el carácter, el receptor realiza la misma función con el carácter recibido y compara el bit de paridad recalculado con el el recibido.

Si son iguales, se supone que no hubo ningún error, en caso contrario, se supone que hubo un error de transmisión.

**Cálculo del bit de paridad** Para obtener el bit de paridad de un carácter, se suman (en módulo dos) todos los bits 1 del código correspondiente al carácter, y se escoge el bit de paridad de modo que el número total de bits 1 (incluido el de paridad) sea par -paridad par-o impar -paridad impar-. A continuación se muestra el principio de este esquema en las figuras siguientes.

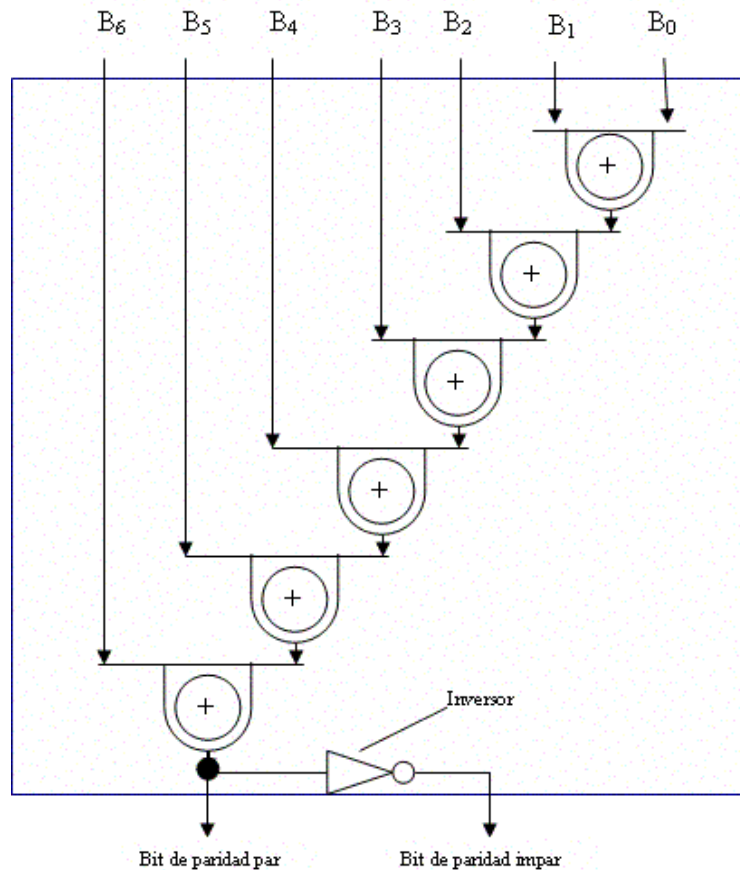
Figura 6. Posición del bit de paridad en el carácter transmitido



El circuito con que se genera el bit de paridad de cada carácter consiste en un conjunto de compuertas **OR exclusiva (XOR)** -llamadas también sumador módulo 2, debido a que la salida de la operación OR Exclusiva entre dos dígitos binarios equivale a la suma de esos dígitos sin el bit de acarreo -conectadas como se ilustra en la figura (b). Primero se aplica una OR exclusiva entre el par de dígitos menos significativos, luego se aplica otra OR exclusiva a la salida anterior y al siguiente bit y a si sucesivamente. La salida de la última compuerta es el bit de paridad requerido, el cual se carga en el registro piso del transmisor antes de

enviar el carácter. De manera similar, en el extremo receptor el bit de paridad recalculado se compara con el recibido; si son distinto, se habrá detectado un error de transmisión.

Figura 7. Circuito generador del bit de paridad



Una deficiencia de este esquema es que no detecta errores en números pares de bits, pues el patrón de bits resultante (alterado) será distinto pero válido. Veamos:

<b>1001001</b>	<b>1</b>	Carácter transmitido (Paridad par)
<b>1111111</b>	<b>1</b>	Carácter recibido (alterado pero válido)

Sin embargo si se generan alteraciones en un número impar de bits, será detectado el error de transmisión.

**2.6.7.2 VERIFICACIÓN DE SUMA DE BLOQUE** Cuando se están transmitiendo bloques de caracteres, es mayor la probabilidad de que un carácter (y por tanto un bloque) contenga un error de bit. La probabilidad de que un bloque contenga un error se denomina *tasa de errores de bloque*. Al estar transmitiendo bloques de caracteres (tramas) se puede lograr una extensión en la capacidad de detección de errores obtenida con un solo bit de paridad por carácter mediante un conjunto adicional de bits de paridad calculado a partir del bloque completo de caracteres de la trama. Con este método a cada uno de los caracteres (de la trama) se le asigna un bit de paridad *-paridad transversal o de fila-*. Adicionalmente se calcula un bit de paridad por cada posición de bit *-paridad longitudinal o de columna-* de la trama completa. El conjunto resultante de bits de paridad por cada columna se denomina *carácter de verificación (de suma) de bloque* por que cada bit que compone este carácter es la suma módulo 2 de todos los bits de la columna correspondiente. En el ejemplo de la figura (c) se emplea paridad impar para los bits de paridad de fila y paridad par para los bits de paridad de columna, y se supone que la trama contiene caracteres imprimibles.

Figura 8 Ejemplo de verificación de suma de bloque

	Pr	B6	B5	B4	B3	B2	B1	B0	
Bits de paridad (impar) transversal (de fila)	0	0	0	0	0	0	1	0	= STX
	1	0	1	0	1	0	0	0	Contenido de la trama
	0	1	0	0	0	1	1	0	
	0	0	1	0	0	0	0	0	
	1	0	1	0	1	1	0	1	
	0	1	0	0	0	0	0	0	
	1	1	1	0	0	0	1	1	
	1	0	0	0	0	0	1	1	= ETX
	1	1	0	0	0	0	0	1	= BCC
		Bits de paridad (par) longitudinal (de columna)							

STX = Carácter de inicio de trama.

ETX = Carácter de fin de trama.

Pr = Bits de paridad de fila.

A partir de este ejemplo se puede deducir que, si bien dos errores de bit en un carácter no serán detectados por la verificación de paridad de fila, si serán detectados por la verificación de paridad de columna correspondiente; esto se cumple sólo si no ocurren dos errores de bit en la misma columna simultáneamente, pero la probabilidad de que esto ocurra es mucho menor que la probabilidad de que sucedan dos errores de bit en un solo carácter.

Con la verificación de suma de bloque mejora de manera significativa la capacidad de detección de errores, por esta razón, es una de las alternativas que utiliza el TCP.

## 2.6.8 CONTROL DE ERRORES

Cuando introducimos datos en un computador por medio de un teclado, al pulsar cada tecla normalmente se transmite al computador la palabra de código resultante como una serie de bits a través de un UART y con transmisión asíncrona. Luego el programa que controla el proceso de entrada del computador lee y almacena el carácter recibido e inicia el proceso de salida hacia la pantalla. Si el carácter que se exhibe no es el mismo que se introdujo o que se deseaba, el usuario sólo tiene que introducir un carácter de control apropiado: el de suprimir o el de retroceso, por ejemplo. Cuando el programa de control recibe este carácter de control, desecha el carácter previamente introducido y lo borra de la pantalla. De esta suerte, el usuario efectúa una forma de control manual de errores.

Cuando una terminal está conectada a un computador remoto por medio de digamos, la PSTN analógica y un módem se utiliza un procedimiento similar. En vez de que cada carácter teclado se exhiba directamente en la pantalla de la terminal, primero es transmitido al computador remoto, el cual lee y almacena el carácter y lo retransmite de vuelta a la terminal, que entonces lo exhibe. Si este carácter es diferente del que se introdujo o del que se deseaba, el usuario puede iniciar la transmisión de un código de eliminación apropiado. Este modo de control de errores se denomina *verificación de eco*.

Por otro lado, cuando un computador está transfiriendo bloques de caracteres (tramas) a través de un enlace de datos en serie a otro computador, el programa del computador receptor que controla el proceso de recepción deberá emprender automáticamente el procedimiento de control de errores sin la intervención del usuario. Por lo regular, el computador receptor examina la trama recibida en busca de posibles errores de transmisión y luego devuelve un mensaje (trama) de control corto ya sea para indicar que recibió la trama correctamente o para solicitar el envío de otra copia. a este tipo de control de errores se le llama solicitud de repetición selectiva (*ARQ automatic repeat request*).

Son los dos tipos básicos de ARQ: RQ inactiva, que se usa con los esquemas de transmisión de datos orientados a caracteres (o bytes), y RQ continua, que emplea estrategias de repetición selectiva o bien de retroceder N. La RQ continua se utiliza principalmente con los esquemas de transmisión orientados a bits. Aunque en muchas aplicaciones la RQ inactiva está siendo reemplazada por el más eficiente esquema de RQ continua.

**2.6.8.1 RQ inactiva:** El esquema de control de errores de RQ inactiva fue definida para lograr la transferencia confiable -es decir, con una alta probabilidad de no incluir errores ni repeticiones y de apearse a la secuencia original- de bloques (tramas) de caracteres imprimibles y de control de formato a través de un enlace de datos en serie, entre un DTE de origen y un DTE de destino. Para distinguir entre el transmisor (origen) y el receptor (destino) de las tramas de datos -más bien conocidas como tramas de información o tramas I - suelen utilizarse los términos primario(P) y secundario(S) respectivamente. Así, el esquema de control de errores de RQ inactiva se ocupa de la transferencia fiable de tramas I entre un primario y un secundario a través de un enlace de datos en serie.

El protocolo de RQ inactiva opera en el modo semidúplex, ya que el primario, después de enviar una trama I, debe esperar hasta que el secundario le envíe la indicación de que recibió correctamente la trama o no. A continuación, el primario envía la siguiente trama o, si la trama anterior no se recibió correctamente, retransmite una copia del mismo.

Hay dos formas de implementar este esquema. Con la *retransmisión implícita*, S sólo confirma las tramas recibidas correctamente y P interpreta la ausencia de un mensaje de confirmación como indicación de que la trama anterior fue alterada. Como alternativa, cuando S detecta una trama alterada devuelve una confirmación negativa para solicitar el envío de otra copia de la trama; esto es *solicitud explícita*.

**2.6.8.2 RQ continua:** Con un esquema de control de errores de RQ continua, el enlace es mucho mejor aprovechado, a expensas de que aumentan los requerimientos de almacenamiento temporal. Como veremos, se necesita un enlace dúplex para su implementación.

Debemos tener claro, que en todos los esquemas de RQ continua las tramas alteradas son desechadas y que las solicitudes de retransmisión se inician sólo después de que fue recibida la siguiente trama sin errores.

**2.6.8.2.1 Repetición selectiva:** Al igual que con el esquema de control de errores de RQ inactiva, la repetición selectiva puede implementarse de dos maneras: S confirma las tramas que recibe correctamente y P determina, a partir de la secuencia de tramas ACK recibidas que se perdió una trama (retransmisión implícita), o bien S devuelve una confirmación negativa explícita para toda trama faltante en la secuencia (solicitud explícita). En ambos esquemas, si una trama se recibe fuera de secuencia, S la conservará en la lista de recepción del enlace hasta haber recibido la trama que seguía en la secuencia.

*Retransmisión Implícita:* El funcionamiento de este esquema se basa en la recepción de tramas ACK asociadas a tramas posteriores para iniciar la retransmisión de una trama previa.

*Solicitud Explícita:* Mecanismo por el cual el secundario le informa al primario mediante una trama NAK que ocurrió una falla, y por lo cual el primario retransmite la trama alterada. Consiste en que una trama de confirmación negativa explícita solicite la retransmisión de una trama específica. A la confirmación negativa se le denomina rechazo selectivo.

La repetición selectiva se emplea primordialmente cuando las tramas transmitidas son en esencia entidades independientes (es decir, cuando el orden de recepción carece de importancia), o bien cuando todas las tramas tienen que ver con el mismo mensaje (o con una trama mayor) y los mensajes (tramas) son reensamblados por el secundario. Un ejemplo de esta última situación es cuando se trata de un enlace con una alta tasa de errores de bit (como un enlace de radio) y el tamaño máximo de trama asociado al enlace es pequeño. La estrategia normal consiste en aplicar la repetición selectiva para los fragmentos de trama más pequeños relacionados con cada una de las tramas grandes y en reensamblar dichos fragmentos para formar una trama grande antes de entregar esta última. Así, para solicitar la retransmisión de cualesquiera fragmentos que se hayan perdido basta una sola NAK.

**2.6.9.2.2 Retroceder N (Mecanismo utilizado por TCP):** No obstante, en muchas aplicaciones las tramas tienen que entregarse en el mismo orden en que se presentaron, de modo que S debe retener (en almacenamiento temporal) las tramas recibidas fuera de secuencia hasta haber recibido la o las tramas faltantes. Como a menudo las tramas son grandes, y el número de buffers de trama requeridos también puede hacerse alto, la capacidad de almacenamiento temporal que requiere el subsistema de comunicaciones puede ser inaceptablemente elevada. Es por esta razón que la mayor parte de las aplicaciones del tipo mencionado y de las redes terrestres emplean el esquema de control de retransmisión de retroceder N.

Con este esquema, cuando el secundario detecta una trama fuera de secuencia, informa al primario que debe comenzar a retransmitir todas las tramas a partir de un número de trama especificado. Esto lo hace devolviendo una trama de confirmación negativa especial llamada *rechazo*.

Concluimos, concerniente al RQ continua, que la perdida de una trama I se detecta sólo después de haberse recibido correctamente la siguiente trama I. Esto requiere un flujo continuo de tramas I listas para ser transmitidas, pues de lo contrario habrá retardos inaceptables mientras S espera la transmisión de la siguiente trama I. Para contemplar esta posibilidad, P suele valerse de un mecanismo de tiempo máximo adicional, similar al que describimos para el esquema de control de RQ inactiva. Son varios los enfoques posibles, pero uno sencillo supone que se activa un cronómetro independiente cada vez que P transmite una trama I, y se para (restablece) cuando se recibe una confirmación que indica su recepción correcta. Si no se recibe la confirmación de una trama antes de que expire su intervalo de tiempo máximo, se retransmitirá esa trama.

El intervalo de tiempo máximo que se escoja debe ser mayor que el retardo de propagación del peor caso entre la transmisión de una trama y la recepción de la confirmación asociada. Además, con un mecanismo de tiempo máximo S puede recibir copias duplicadas de una trama que haya recibido correctamente, pues es posible que la trama de confirmación correspondiente se haya alterado en su camino hacia P. Al no recibir confirmación, P supone que la trama original fue alterada y equivocadamente retransmite otra copia.

Si el esquema es de retroceder N, ello no implica ningún problema, ya que el  $N(S)$  de la trama duplicada no será igual al  $V(R)$  actual que tiene S y será desechada automáticamente; pero si el esquema es de retransmisión selectiva se permite la posibilidad de que P envíe una o más tramas duplicadas si S mantiene una lista ordenada (la lista de recepción) de las últimas N tramas I que recibió correctamente. De este modo, S puede comprobar si una trama recibida es un duplicado de una trama que ya había recibido correctamente (y por ende confirmado) o si es una nueva trama.

Por claridad, suponemos que las tramas de información fluyen en una dirección y que el camino de retorno se destina sólo a las confirmaciones. Sin embargo, casi todos los enlaces de comunicación que usan RQ continua son enlaces dúplex y transportan información en ambas direcciones. Para ello cada uno de los extremos

de estos enlaces contiene tanto un primario como un secundario: el primario controla la secuencia de tramas I transmitidas y el segundo hace lo propio con las recibidas. Así, ambos lados del enlace cuentan con un V(S), controlada por el primario, y un V(R) controlada por el secundario. Aunque se emplean tramas ACK y NAK independientes, como es posible que haya una trama I a la espera de ser transmitida en la dirección opuesta cuando se debe devolver una ACK o una NAK, algunos protocolos aprovechan las tramas I que fluyen en la dirección de retorno para llevar información de confirmación de tramas I transmitidas en la dirección de ida, a fin de mejorar el grado de utilización del enlace. En tal caso, cada trama I transmitida contiene tanto un N(S) - para indicar el número de secuencia de transmisión- como un N(R) - que contiene información de confirmación para la dirección opuesta. A este esquema se le denomina confirmación superpuesta. Un protocolo que utiliza este mecanismo es el protocolo de enlace de datos de alto nivel (HDLC).

### **2.6.9 CONTROL DE FLUJO**

El control de flujo es un mecanismo por el cual modem y ordenador gestionan los intercambios de información. Estos mecanismos permiten detener el flujo cuando uno de los elementos no puede procesar mas información y reanudar el proceso cuando las condiciones así lo ameriten. Los métodos más comunes de control de flujo son:

- Control de flujo hardware.
- Control de flujo software.

**2.6.9.1 Control de flujo de software:** Este se ocupa de controlar la tasa de transmisión de caracteres o tramas por un enlace, de modo que el receptor siempre tenga suficientes recursos de almacenamiento temporal para aceptarlos antes de ser procesados. Además es empleado para la retransmisión selectiva orientada a tramas, el receptor podría agotar su capacidad de almacenamiento temporal si está tratando de retener un número indeterminado de tramas. Sus dos esquemas más comunes son:

**2.6.9.1.1 X-ON/X-OFF:** La verificación de Eco además de efectuar el control manual de errores, realiza implícitamente funciones de control de flujo ya que, si el computador remoto agota su capacidad de almacenamiento temporal, dejará de enviar eco de los caracteres a la pantalla de la terminal. Esto hará que el usuario automáticamente deje de teclear más caracteres. No obstante, por lo regular la falta de eco de caracteres por parte del computador se debe a una sobrecarga temporal de éste. En tal caso, si el usuario no deja de transmitir nuevos caracteres, el computador incurrirá en gastos extra de procesamiento innecesarios por el mero hecho de leerlos y desecharlos.

Comúnmente se invoca a un recurso automático de control de flujo adicional para asegurar que una terminal no envíe más caracteres mientras no se haya resuelto una situación de sobrecarga. Para que esto se pueda lograr, el computador devuelve un carácter de control especial, X-OFF, al dispositivo controlador dentro de la terminal, ordenándole que suspenda la transmisión. Cuando se recibe el carácter X-OFF, la terminal ignorará los caracteres adicionales introducidos por el teclado, o bien los almacenará temporalmente en un buffer local hasta que se despeje la sobrecarga. Para que el computador no tenga gastos extra de procesamiento innecesarios y pueda controlar el flujo de caracteres.

**2.6.9.1.2 Ventana Deslizante (Mecanismo utilizado por TCP):** El término ventana de transmisión se refiere a un buffer en el cual se almacenan copias de las tramas enviadas, en espera de recibir el ACK correspondiente; si no llegan en el tiempo previsto, se realiza una nueva copia y se retransmite la trama. El número

de secuencia de transmisión,  $N(S)$ , es la posición que ocupa la trama enviada en el buffer. El número de secuencia viaja en la cabecera de la trama, dentro del campo de control.

Por ventana de recepción se entiende el buffer donde se almacenan las tramas que llegan a una máquina por alguno de sus enlaces. En este buffer esperan a ser procesadas, y a que se devuelva el acuse de recibo correspondiente a cada una de ellas, para que la máquina origen sepa que la transmisión ha llegado sin problemas a su destino. El número de secuencia de recepción,  $N(R)$ , es la posición que ocupa la trama recibida en el buffer de recepción. El tamaño de la ventana puede estar preestablecido, o puede negociarse durante el establecimiento de la conexión.

Tengamos en cuenta que un esquema de control de errores de RQ Inactiva hace un aprovechamiento ineficiente del ancho de banda de transmisión, requiere un mínimo de capacidad de almacenamiento temporal ya que, una vez que P (el primario o emisor) ha transmitido una trama, debe esperar la confirmación de S (el secundario o receptor) antes de transmitir la siguiente. Así, el flujo de tramas I por el enlace está automáticamente bajo un control estricto.

Con un esquema de control de errores de RQ continua, P puede enviar continuamente tramas I antes de recibir una confirmación. Con este tipo de esquema es posible que el destino agote su capacidad de almacenamiento temporal si, por ejemplo no puede pasar a la capa superior las tramas que recibe. S dejará de devolver tramas de confirmación, la lista de disminución de P crecerá y esto, a su vez, podrá interpretarse como una señal para que P deje de transmitir más tramas hasta que se reinicie el flujo de confirmaciones.

Para implementar este esquema, se fija un límite máximo al número de tramas I que puede estar en espera de confirmación y que, por tanto, están pendientes en la lista de disminución. Este límite es la ventana de transmisión,  $K$ , del enlace. Si  $K=1$ , el esquema de control de flujo se convierte en RQ Inactivo, con la consecuente disminución en la eficiencia de transmisión. Por lo regular se escoge un límite tal que, siempre que el destino puede pasar a la capa superior o absorber

todas las tramas que reciba, la ventana de transmisión no obstaculice el flujo de tramas  $I$  por el enlace. Al seleccionar la ventana de transmisión hay que tomar en cuenta factores como el tamaño máximo de trama, el almacenamiento temporal disponible, el retardo de propagación del enlace y la tasa de transmisión de bits.

*Ventajas del control de flujo:*

Técnica para que el transmisor no sature al receptor

Receptor reserva memoria temporal para el almacenamiento de datos, los procesa y los envía a niveles superiores

Control de flujo evita que se sature esta memoria

*Desventajas del control de flujo:*

Se ha intentado utilizarlo para resolver congestión

No es apropiado para tráfico en ráfagas

Se restringe al usuario al tráfico promedio pero no funciona para picos de gráfico.

**2.6.8.2 Grado de utilización del enlace** Cuando hablamos del protocolo de RQ inactiva, podemos decir con bastante aproximación, que el grado de utilización de un enlace,  $U$ , es una función del tiempo  $T_{ix}$  requerido para transmitir una trama  $I$  y del retardo de propagación del enlace,  $T_p$ . Sin embargo, en enlaces que  $T_p$  es mayor que  $T_{ix}$ , la ventana de transmisión  $K$  también influye sobre el aprovechamiento del enlace.

En general, el grado de utilización del enlace  $U$  para una ventana de transmisión  $k$  está dada por:  $U = 1$ , si  $K$  es mayor o igual que  $1 + 2a$ .

$U = K / (1 + 2a)$  si  $K$  es menor que  $1 + 2a$ .

Para observar como  $K$  influye sobre el aprovechamiento del enlace, analizaremos el siguiente caso:

Se va a transmitir una serie de tramas de 1000 bits a través de un protocolo de RQ continua. Determinar la eficiencia del enlace para los siguientes tipos de enlace de datos si la velocidad de propagación es de  $2 \times 10^8 \text{ m s}^{-1}$  y las tasa de errores de bit de los enlaces son tan bajas que pueden ser ignoradas:

- ⊕ Un enlace de 1 Km de 1 Mbps y una ventana de transmisión  $K = 2$ .
- ⊕ Un enlace de 10 Km de 200 Mbps y una ventana de transmisión  $K = 7$ .
- ⊕ Un enlace por satélite de 50.000 Km de 2 Mbps y una ventana de transmisión  $K = 127$ .

A continuación desarrollaremos, aplicando las formulas  $T_p = (S / V)$ ,  $T_{ix} = (N_i / R)$ ,  $a = (T_p / T_{ix})$  y  $N_i = \text{Número de bits de información}$ , que nos permitan ver los grados de utilización de los enlaces mencionados.

$$\oplus T_p = (10^3 / 2 \times 10^8) = 5 \times 10^{-6} \text{ s}$$

$$T_{ix} = (1000 / 1 \times 10^6) = 10^{-3} \text{ s.}$$

Por tanto:

$$a = (5 \times 10^{-6} \text{ s} / 10^{-3} \text{ s}) = 5 \times 10^{-3}$$

Como  $K = 2$  y es mayor que  $1 + 2a$ , en consecuencia  $U = 1$ .

$$\oplus T_p = (10 \times 10^3 / 2 \times 10^8) = 5 \times 10^{-5} \text{ s}$$

$$T_{ix} = (1000 / 200 \times 10^6) = 5 \times 10^{-6} \text{ s.}$$

Por tanto:

$$a = (5 \times 10^{-5} \text{ s} / 5 \times 10^{-6} \text{ s}) = 10$$

Como  $K = 7$  y es menor que  $1 + 2a$ , en consecuencia  $U = K / (1+2a) = 7 / (1+20) = 0,33$

$$\oplus T_p = ( 50 \times 10^6 / 2 \times 10^8 ) = 0,25 \text{ s}$$

$$T_{ix} = ( 1000 / 2 \times 10^6 ) = 5 \times 10^{-4} \text{ s.}$$

Por tanto:

$$a = ( 0,25 \text{ s} / 5 \times 10^{-4} \text{ s} ) = 500$$

Como  $K = 127$  y es menor que  $1 + 2a$ , en consecuencia  $U = K / ( 1+2a ) = 127 / (1+1000) = 0,127$

A partir de los resultados podemos deducir que en ciertos casos la elección de  $K$  influye en gran medida en el grado de utilización del enlace. Como puede apreciarse, incluso con  $K = 127$  el aprovechamiento de un enlace de satélite sigue siendo muy bajo, Por ello, se usa una  $K$  mucho más grande. Esto también se aplica a enlaces terrestres que operan con altas tasas de bits.

En los resultados observados anteriormente, supusimos que no había errores de transmisión. Los errores reducen aún más el grado de utilización del enlace porque hay que retransmitir algunas tramas. Sin embargo, el efecto difiere un poco en los dos esquemas de retransmisión. Por ejemplo, si se usa un esquema de repetición selectiva (Ver Guía V), tan sólo se reduce  $U$  en el número de intentos de transmitir cada trama,  $N_r$ , ya que sólo se retransmite la trama alterada. En consecuencia, si  $P_f$  es la tasa de errores de trama del enlace y suponemos errores aleatorios:

$$N_r = 1 / ( 1 - P_f ) \quad \text{El valor modificado de } U \text{ es entonces:}$$

$$U = K / N_r ( 1 + 2a ) = K ( 1 - P_f ) / ( 1 + 2a ) ,$$

Para una  $K$  menor que  $1 + 2a$ . Si  $K$  es mayor o igual que  $1 + 2a$ , obtenemos  $U$  con sólo sustituir  $K = 1 + 2a$  en esta expresión, ya que  $1 + 2a$  es el número máximo de tramas que es posible retransmitir antes de recibir una confirmación en un tiempo  $T_{ix} + 2T_p$ . Así:

$$U = ( 1 + 2a ) ( 1 - P_f ) / ( 1 + 2a ) = 1 - P_f,$$

para  $K$  igual o mayor que  $1 + 2a$ .

Con un esquema de retroceder N, el aprovechamiento del enlace se reduce todavía más porque, de alterarse una trama, será necesario retransmitir más de una trama. Una vez más, el número de tramas adicionales por transmitir está determinado por la magnitud de K relativa a  $1 + 2a$ .

Si K es menor que  $1 + 2a$ , el número de veces que es necesario retransmitir  $(K - 1)$  tramas es simplemente  $P_f (K - 1)$ . En cada ocurrencia, habrá un retardo adicional de  $1 + 2a$ . La expresión modificada es entonces:

$$U = K (1 - P_f) / [(1 + 2a) + (1 + 2a)P_f(K - 1)]$$

$$U = K (1 - P_f) / (1 + 2a)(1 + P_f(K - 1)) ,$$

para K menor que  $1 + 2a$ , y:

$$U = (1 + 2a)(1 - P_f) / (1 + 2a)(1 + P_f(K - 1))$$

$$U = (1 - P_f) / (1 + P_f(K - 1)) ,$$

para K igual o mayor que  $1 + 2a$

Debemos tener en cuenta que estas dos fórmulas son aproximaciones y sólo producirán resultados útiles si se cumplen las aproximaciones previas. No obstante, constituyen un buen indicio tanto del nivel de rendimiento que cabe esperarse como del rendimiento relativo de cada uno de los métodos.

## CAPITULO 3 MARCO METODOLÓGICO

### 3.1 CICLOS DE VIDA DE DESARROLLO DE SOFTWARE

Existen diversos paradigmas<sup>17</sup> en el desarrollo de software (ciclo de vida clásico, construcción de prototipos, modelo en espiral, Proceso unificado, entre otros). La metodología adoptada es el Proceso Unificado, dadas las características y objetivos del proyecto. Aquí se requiere un marco de trabajo que permita generar sistemas altamente fiables, con una buena gestión de los riesgos, con posibilidad de realizar modificaciones a mitad de camino y poder ofrecer a los usuarios y directores signos visibles de progreso, entre otros aspectos.

### 3.2 PROCESO UNIFICADO

El Proceso Unificado proporciona normas para el desarrollo eficiente de software de calidad en parte porque desde las primeras fases del desarrollo se centra en la detección y corrección de riesgos críticos para el proyecto, además que se basa en tres conceptos fundamentales descritos a continuación:

*Dirigido por Casos de Uso:* Un caso de uso es una descripción de un conjunto de secuencias de acciones que un sistema lleva a cabo y que proporcionan un resultado de interés para un usuario determinado; dicho usuario no es necesariamente una persona, puede ser algún tipo de sistema externo que interactúe con el sistema en cuestión, por ejemplo sensores, temporizadores, etc.

En el Proceso Unificado, el desarrollo del software se centra en la funcionalidad requerida por el usuario por lo que los casos de uso sirven como una representación de los requisitos funcionales del proyecto en desarrollo; pero los

---

<sup>17</sup> Pasos que abarcan los métodos, las herramientas y los procedimientos que hacen parte de la construcción de un sistema bajo la Ingeniería del Software.

casos de uso son más que una herramienta para la captura de requisitos, ya que ellos dirigen el proceso de desarrollo de software en su totalidad.

*Centrado en la Arquitectura:* La arquitectura de un sistema software se describe mediante diferentes vistas del sistema en construcción, dichas vistas deben contener los elementos del sistema más importantes para los desarrolladores, tales como nodos e interfaces, entre otros. Además la arquitectura debe incluir tanto los aspectos estáticos como dinámicos más significativos del sistema. La arquitectura interactúa dinámicamente con los casos de uso, esto es, los casos de uso pueden llegar a alterar la arquitectura y esta a su vez puede hacer necesaria la inclusión de nuevos casos de uso.

*Iterativo e Incremental:* Las iteraciones hacen referencia a diversos pasos en el flujo de trabajo, y los pasos, al incremento del producto. Las iteraciones deben estar controladas, es decir, deben ejecutarse en forma planificada. En cada iteración, se identifican y especifican los casos de uso relevantes, se crea un diseño utilizando la arquitectura seleccionada como guía, se implementa el diseño mediante componentes, y se verifican que los componentes satisfacen los casos de uso. Si una iteración cumple sus objetivos el desarrollo continúa con la siguiente iteración, en caso contrario, se revisa las decisiones previas y se prueba un nuevo enfoque. Esto hace que el desarrollo a través de las iteraciones sea incremental.

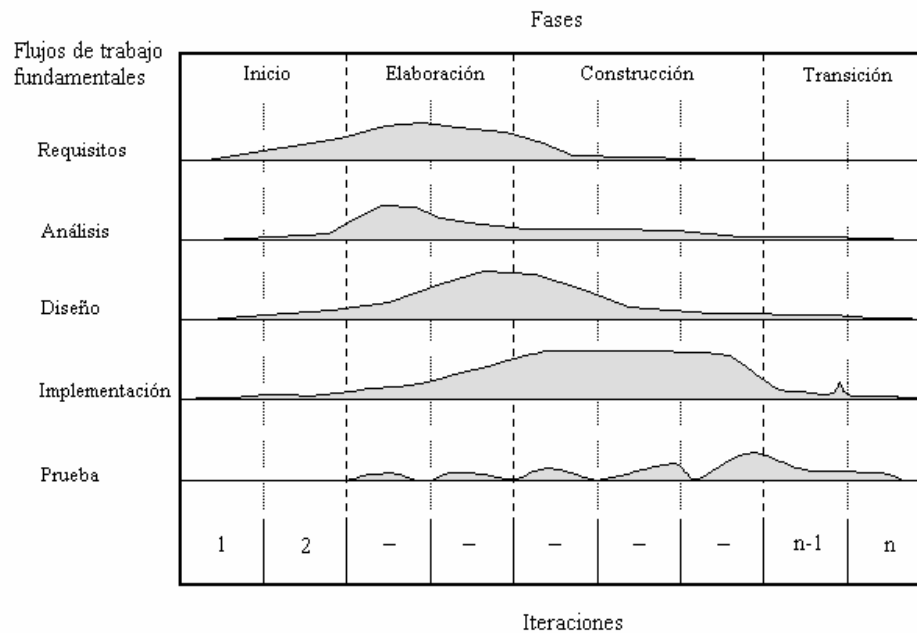
La arquitectura proporciona la estructura sobre la cual guiar las iteraciones mientras que los casos de uso definen los objetivos y dirigen el trabajo de cada iteración. De esta manera el proceso reduce el riesgo de grandes retrasos en la entrega de un producto, se fijan metas más inmediatas por lo cual se pueden controlar mejor el avance del proyecto.

**Fases de un ciclo** Cada ciclo de desarrollo se divide en cuatro fases como se muestra en la figura 9.

El proceso unificado lleva a cabo el proceso de desarrollo en ciclos, donde se obtiene una nueva versión del producto al final de cada ciclo. Cada ciclo se divide

a su vez en cuatro fases: *inicio*, *elaboración*, *construcción*, y *transición*. Cada una de estas fases concluye con un hito, cada hito es determinado por la disponibilidad de un conjunto de artefactos. Además dichas fases se divide a su vez en iteraciones. Cada una de las iteraciones se asemeja a un pequeño ciclo de vida en cascada, pasando a través de los cinco flujos de trabajo fundamentales: requisitos, análisis, diseño, implementación y prueba.<sup>18</sup> cada iteración además incluye una planificación que precede a los flujos de trabajo y una evaluación de ella misma.

Figura 9. “Fases De Un Ciclo En El Proceso Unificado”<sup>19</sup>



Nota: La fase de transición, el software se despliega en la comunidad de usuarios. Una vez que el sistema ha sido puesto en manos de los usuarios finales, a menudo aparecen cuestiones que requieren un desarrollo adicional para ajustar el sistema, corregir algunos problemas no detectados o finalizar algunas

18 Tomado del libro “El Proceso Unificado de Desarrollo de Software” de Jacobson, Booch y Rumbaugh.

19 Adaptación libre de la figura 5.6. tomada del libro “El Proceso Unificado de Desarrollo de Software” de Jacobson, Booch y Rumbaugh. pág 100.

características que habían sido pospuestas. Esta fase demanda una considerable cantidad de tiempo, razón por la cual se omite para los propósitos del presente proyecto de grado.

### **3.3 LENGUAJE UNIFICADO DE MODELADO (UML)**

El lenguaje unificado de modelado es una notación estándar para el modelado de sistemas software, resultado de una propuesta promovida por el consorcio OMG (Object Management Group), del cual forman parte las empresas más importantes que se dedican al desarrollo de software.

UML consiste en una serie de reglas y recomendaciones para representar modelos. Debe tenerse en claro que no se trata de un proceso de desarrollo, es decir, no describe los pasos sistemáticos a seguir para desarrollar software, UML sólo permite documentar y especificar los elementos creados mediante un lenguaje común describiendo modelos.

**3.3.1 Modelado de software con UML.** Para la construcción de modelos, hay que centrarse en los detalles relevantes mientras se ignoran los demás, por lo cual con un único modelo no es suficiente. Varios modelos aportan diferentes vistas de un sistema los cuales nos ayudan a comprenderlo desde varios frentes. Así, UML recomienda la utilización de nueve diagramas para representar las distintas vistas de un sistema. Estos diagramas son los siguientes:

- ⊕ *Diagrama de casos de uso.* Modela la funcionalidad del sistema agrupándola en descripciones de acciones ejecutadas por un sistema para obtener un resultado.
- ⊕ *Diagrama de clases.* Muestra las clases (descripciones de objetos que comparten características comunes).

- ⊕ *Diagrama de objetos.* Muestra una serie de objetos (instancias de clases) y sus relaciones.
- ⊕ *Diagrama de secuencia.* Enfatiza la interacción entre los objetos resaltando la organización estructural de los objetos en lugar del orden de los mensajes intercambiados.
- ⊕ *Diagrama de estados.* Modela el comportamiento de acuerdo con los eventos.
- ⊕ *Diagrama de actividades.* Simplifica el diagrama de estados modelando el comportamiento mediante flujos de actividades.
- ⊕ *Diagrama de componentes.* Muestra la organización y las dependencias entre un conjunto de componentes.
- ⊕ *Diagrama de despliegue.* Muestra los dispositivos que se encuentran en un sistema y su distribución en el mismo.

### 3.5 SELECCIÓN DEL MODELO DE DESARROLLO

Las Descripciones que siguen a continuación son los criterios que se tiene en cuenta para la valoración de cada uno de los modelos de ciclo de vida descritos anteriormente, éstos permitieron escoger el método más recomendable para el desarrollo del proyecto.

⊕ *Poca identificación de los requerimientos.* Hace notación al óptimo funcionamiento del modelo de ciclo de vida cuando el cliente o el desarrollador identifican erradamente los requerimientos del sistema o cuando el cliente es propenso a modificar los requerimientos. De esta manera se indica si el modelo es el adecuado para el desarrollo exploratorio del software.

⊕ *Poca comprensión sobre la arquitectura.* Hace notación al óptimo funcionamiento del modelo de ciclo de vida cuando se está desarrollando en un área de aplicación nueva o en áreas familiares con capacidad de desarrollo desconocidas.

- ⊕ *Genera un sistema altamente fiable.* Hace notación a los defectos que pueden aparecer en un sistema desarrollado con un modelo de ciclo de vida cuando éste comienza a funcionar.
- ⊕ *Genera un sistema con amplio desarrollo.* Hace notación a la facilidad con que se puede modificar el sistema de información en tamaño y diversidad durante su tiempo de vida.
- ⊕ *Gestiona riesgos.* Hace notación a la capacidad que posee el modelo para identificar y controlar riesgos durante la aplicación, al igual que los riesgos para el producto.
- ⊕ *Requiere poco tiempo de gestión.* Hace notación a la cantidad de tiempo de gestión y técnico requerido para la utilización efectiva del modelo. Este tiempo incluye planificación, seguimiento del estado, producción de la documentación, adquisición de paquetes y otras actividades que no se incluyen directamente en la producción del software.
- ⊕ *Modificaciones a medio camino.* Hace notación a la capacidad para modificar aspectos significativos del software a mitad de camino de la planificación del desarrollo. Esto no incluye la modificación de la función principal del producto.
- ⊕ *Requiere poca sofisticación para la directiva y desarrolladores.* Se refiere a especificar el nivel de educación y formación que se necesita para utilizar el modelo con éxito.

Basados en las características enunciadas anteriormente para cada una de las metodologías de desarrollo de software y tomando en cuenta las ventajas e inconvenientes de los modelos de ciclo de vida detallados en este documento, se llego a la conclusión que el Proceso unificado de desarrollo de software es el modelo mas indicado para seguir la ejecución del proyecto.

## **PARTE 2: DESARROLLO DEL SISTEMA**

### **CAPÍTULO 4: FASE DE INICIO**

El objetivo general de la fase de inicio es poner en marcha el proyecto. En esta fase se define el alcance del sistema propuesto, es decir, se limita el área que va a cubrir el proyecto. Además, se definen las principales funciones del sistema.

El proceso unificado pretende en esta fase el esbozo de una arquitectura candidata que pueda soportar el ámbito del sistema. Por último, se realizan las estimaciones iniciales de costo y agenda que permitan desarrollar el análisis inicial del negocio.

#### **4.1 ALCANCE DEL PROYECTO**

Para la realización de la herramienta educativa es necesario limitar el área que cubrirá el proyecto, en este caso, operará sobre una plataforma Web enfocada a apoyar las actividades pedagógicas específicas del área de telecomunicaciones en la temática de protocolos de Internet, enriqueciendo las posibilidades de acceder a información representativa en pro de una formación integral.

En la búsqueda de métodos y herramientas, que permitieran llegar a la comunidad educativa interesada en mejorar y aclarar sus conocimientos mediante procesos de educación auto asistida, la escuela de Ingeniería de Sistemas ha encontrado en la Internet el medio de acercar al aula de clases elementos que permiten acceder al conocimiento sin implicar trasladarse o contar con nutridos presupuestos para adquirir materiales y ponerlos al alcance de los alumnos. Esta herramienta ofrecerá interactividad, comunicación, dinamismo en la presentación de contenidos y elementos que permiten atender a los usuarios con distintos estilos de aprendizaje, todo en un mismo sitio: Su computador.

El criterio que se ha tenido en cuenta para decidir el ámbito de la herramienta es la identificación de actores para obtener un entorno en el que se va a desarrollar la herramienta.

**4.1.1 Objetivos de la fase de inicio.** A continuación se plantean los objetivos que deben cubrirse al final de éste capítulo.

- ✦ Identificar los procesos que deben tener los módulos del sistema logrando un esquema funcional, evitando la redundancia de procedimientos y buscando colmar las expectativas que se tienen para cada uno de los temas abordados.
- ✦ Identificar los actores<sup>20</sup> que participaran en la herramienta de acuerdo su interacción de forma directa con los procesos que en ella se dan.
- ✦ Identificar los casos de uso<sup>21</sup> correspondientes para cada uno de los actores, permitiendo establecer los posibles procedimientos a seguir en la operación de los módulos que conforman la herramienta.
- ✦ Identificar los estados que hacen parte de cada uno de los procesos tenidos en cuenta para cada módulo y con ellos realizar los respectivos diagramas, los cuales permitirán establecer la ruta de ejecución de cada procedimiento.

---

<sup>20</sup> Un actor es un rol que un usuario juega con respecto al sistema, no necesariamente representa a una persona en particular, si no la labor que realiza frente a la herramienta.

<sup>21</sup> Se emplean para visualizar el comportamiento de la herramienta, una parte de ella o una sola clase. De forma que se pueda conocer como responde esa parte de la herramienta.

## 4.2 IDENTIFICACIÓN DE ACTORES

Los actores que participan en el sistema son Visitante y Receptor.

Tabla 5. Descripción de los actores que participarán en la herramienta.

Actor	Descripción	Rol (Papel que desempeña)	Necesidades (Para qué utiliza el sistema)
Visitante	Representa a la persona que quiera acceder a la información general y laboratorios de la herramienta	<ul style="list-style-type: none"> <li>⊕ Explorar la información disponible en la herramienta.</li> <li>⊕ Interactuar con Los applets disponibles para cada laboratorio.</li> <li>⊕ Enviar mensajes de aportes y sugerencias.</li> </ul>	Utiliza el sistema para: <ul style="list-style-type: none"> <li>⊕ Enriquecer sus bases teóricas en el tema de los protocolos de comunicación TCP e IP.</li> <li>⊕ Aportar ideas que favorezcan la evolución de la herramienta en versiones futuras.</li> </ul>
Receptor	Representa a la(s) persona(s) encargada(s) de recopilar la información y velar por que las opinión de los visitantes sean tenidas en cuenta.	<ul style="list-style-type: none"> <li>⊕ Analizar, depurar y la información proveniente del módulo buzón.</li> </ul>	<ul style="list-style-type: none"> <li>⊕ Llevar un seguimiento que permita establecer la aceptación de la herramienta.</li> <li>⊕ Qué temas relacionados con el ámbito de la herramienta no han sido abordados por ésta</li> </ul>

En la tabla 6 se presentan los casos de uso identificados en esta fase, teniendo en cuenta los actores que intervienen en cada uno de ellos.

Tabla 6. Casos de uso por actores.

	<b>Caso de Uso</b>	<b>Actores</b>
	<b>Módulo documentación</b>	
1	Leer documentación	Visitante
	<b>Módulo de Ayuda</b>	
2	Solicitar ayuda	Visitante
	<b>Módulo SEEPINTmail</b>	
3	Solicitar SEEPINTmail	Visitante
4	Establecer conexión	Visitante
5	Enviar mensaje	Visitante
6	Cerrar sesión	Visitante
	<b>Módulo Laboratorios</b>	
7	Solicitar Guía I	Visitante
8	Solicitar Guía II	Visitante
9	Solicitar Guía III	Visitante
10	Solicitar Guía IV	Visitante
11	Solicitar Guía V	Visitante
12	Solicitar Guía VI	Visitante
	<b>Módulo Buzón</b>	
13	Enviar Aporte /Solicitud	Visitante
14	Revisar Aportes /Solicitudes	Todos

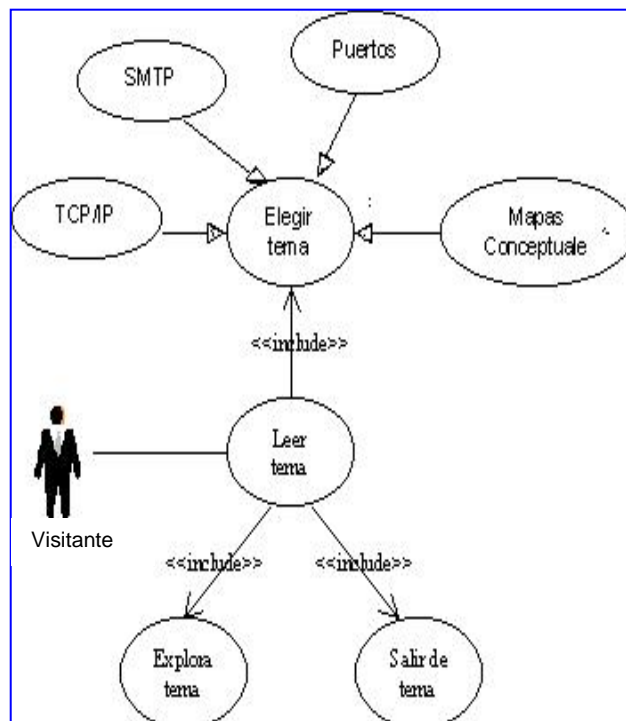
**4.2.1 Requisitos candidatos del sistema.** Antes de realizar una descripción de los módulos de la herramienta, se determinaron una serie de requisitos funcionales, evaluando características provenientes tanto de diversas fuentes de información como de la selección de aquellos temas que revisten mayor importancia y que son de interés general.

### 4.3 DESCRIPCIÓN DE MÓDULOS DE LA HERRAMIENTA

**4.3.1 Descripción del módulo Documentación.** Este módulo permite acceder a información complementaria que le permitirá al actor involucrado, ubicarse teóricamente en los diferentes conceptos que giran en torno a los protocolos TCP e IP.

**4.3.1.1 Casos de uso.** El diagrama de casos de uso que se muestra en la figura 10 nos permite ver la operación que realiza el actor involucrado para éste módulo.

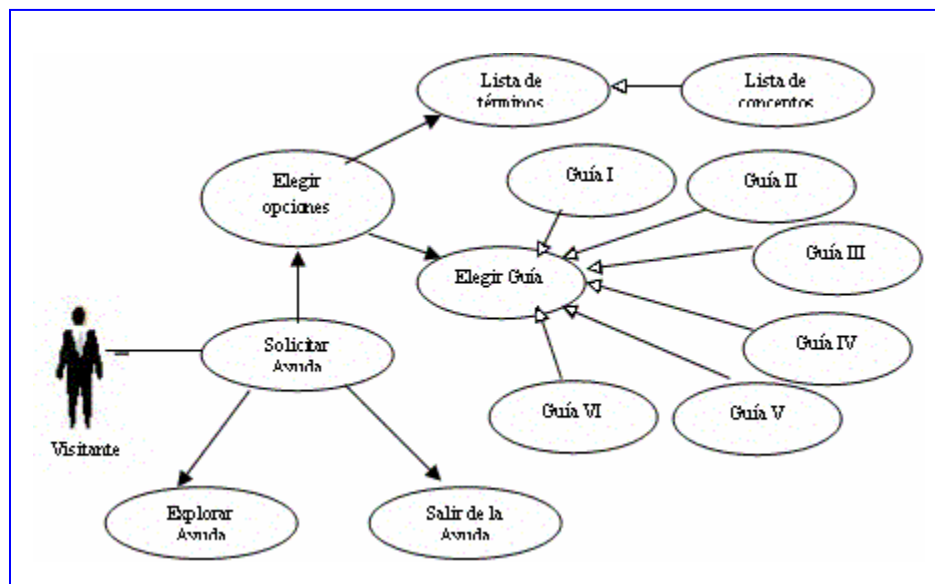
**Figura 10. Diagrama de Caso de uso del módulo Documentación**



**4.3.2 Descripción del módulo Ayuda.** Este módulo facilita al actor involucrado ilustración sobre el manejo de la herramienta y acceso a un diccionario de términos relacionados con la temática de la herramienta.

**4.3.2.1 Caso de uso.** El caso de uso que se presenta en la figura 11 permite comprender el ámbito del módulo Ayuda para el actor Visitante.

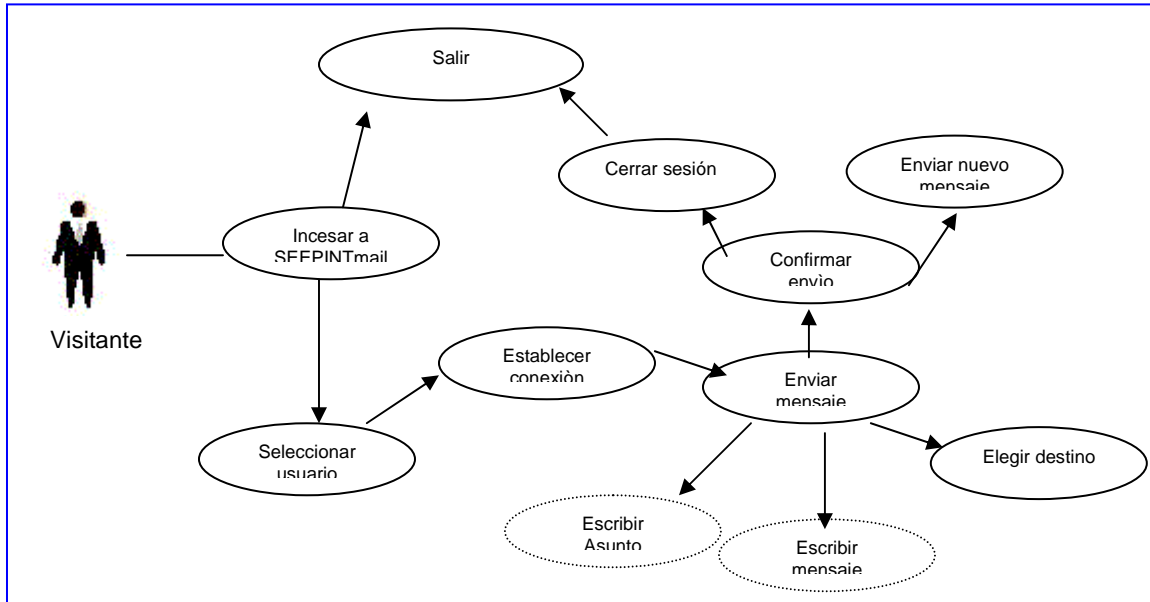
**Figura 11. Diagrama de Caso de uso del módulo Ayuda**



**4.3.3 Descripción del módulo SEEPINTmail.** Este módulo es un preámbulo al conjunto de laboratorios conforman la herramienta. Imita la etapas en que opera una aplicación de correo electrónico real. Se emplea para mostrar el correo electrónico cómo un caso de uso que solicita los servicios de TCP e IP.

**4.3.3.1 Caso de uso.** El caso de uso que se presenta en la figura 12. permite comprender el ámbito del módulo SEEPINTmail para el actor Visitante.

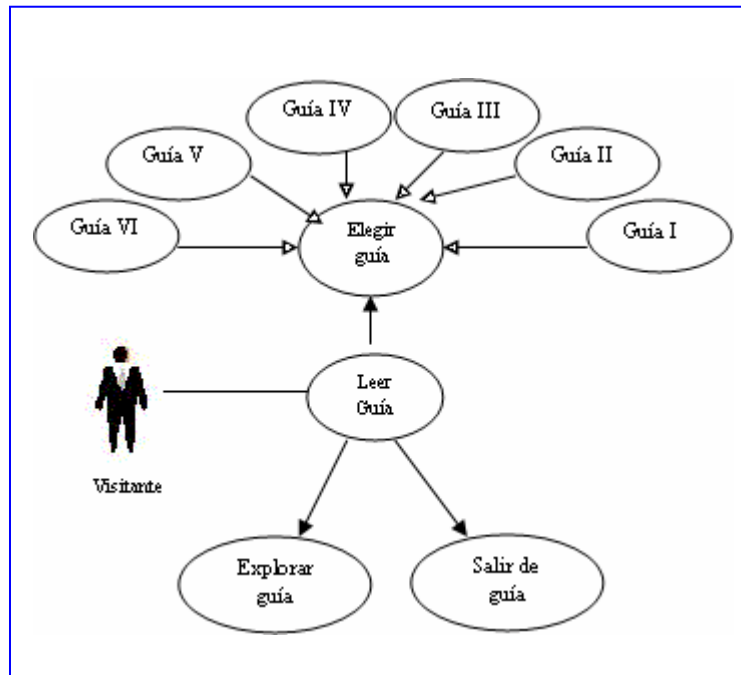
**Figura 12. Diagrama de Caso de uso del módulo SEEPINTmail**



**4.3.4 Descripción del módulo Laboratorios.** Este módulo es el componente principal de la herramienta, por que en él se han implementado un conjunto de s para mostrar el funcionamiento de los procesos que hacen de TCP e IP protocolos útiles en la transferencia de información.

**4.3.4.1 Casos de uso.** El diagrama de casos de uso que se muestra en la figura 13 nos permite ver la operación que realiza el actor involucrado en éste módulo.

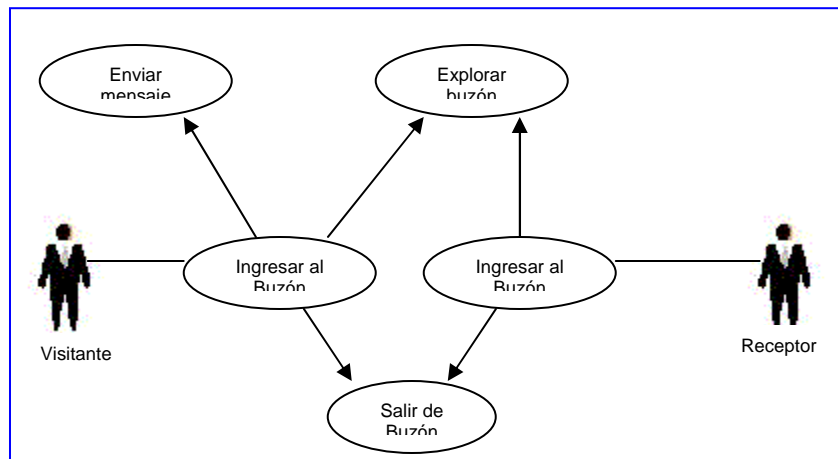
Figura 13. Diagrama de Caso de uso del módulo Laboratorios



**4.3.5 Descripción del módulo Buzón.** Este módulo es un valor agregado de la herramienta que permitirá recopilar la información suministrada por el actor Visitante con el propósito de que ésta información sea tenida en cuenta para la evolución de la herramienta en versiones posteriores.

**4.3.5.1 Casos de uso.** El diagrama de casos de uso que se muestra en la figura 14 nos permite ver la operación que realizan los actores involucrados en éste módulo.

Figura 14. Diagrama de Caso de uso del módulo Buzón



#### 4.4 ARQUITECTURA CANDIDATA

El desarrollo de la herramienta, estará basado en applets de Java (UI-SWING), que es un lenguaje de alto nivel orientado a objetos, que permite visualizar el resultado de sus operaciones, funciones, procedimientos y clases en documentos HTML los cuales son interpretados por la máquina virtual de Java.

Para el manejo de la información que el actor Visitante puede suministrar a la herramienta, se ha utilizará PHP, que es un lenguaje de alto nivel interpretado de alto nivel, embebido en páginas HTML y ejecutado en el servidor, que permitirá visualizar la información al actores involucrado con esta (Visitante y Receptor).

Para el almacenamiento de la información, teniendo en cuenta que ésta es poca , se ha recurrido a la utilización de archivos planos lo que permitirá a la herramienta ser independiente del servidor de Base de Datos que se haya instalado en la máquina en que resida.

Esta combinación - Java, Java Script, HTML, PHP- convierten a la herramienta en un material rico en contenido, dinamismo y altamente interactivo.

## 4.5 RIESGOS DEL PROYECTO

Pueden existir algunos riesgos que de no ser mitigados pondrían en peligro el éxito del proyecto, aunque no es necesaria su eliminación, se puede realizar un plan de contingencia con el fin de reducir su posibilidad de ocurrencia o gravedad de la misma en caso de que ocurran.

En la lista de los riesgos críticos mostrados en la tabla 7 se incluyen:

- ⊕ Descripción: Comienza con una corta descripción a la cual se le van añadiendo detalles.
- ⊕ Prioridad: Se le asigna una prioridad al riesgo, la cual se puede clasificar como crítica, significativa o rutinaria.
- ⊕ Impacto: Indica que parte del proyecto o del sistema se verán afectadas por el riesgo.
- ⊕ Monitor: Indica quien es responsable del seguimiento de un riesgo persistente.
- ⊕ Responsabilidad: Indica qué individuo o unidad de la organización es responsable de eliminar el riesgo.
- ⊕ Contingencia: Indica lo que ha de hacerse en caso de que el riesgo se materialice.

Tabla 7: Listado de riesgos posibles.

Descripción	Permanencia del servidor Apache activo	Capacidad de usuarios concurrentes del servidor Web
Prioridad	Crítico	Crítico
Impacto	Se vera afectado completamente el desempeño de las páginas , ya que no habrá respuesta ante la petición de recursos al servidor por parte de los usuarios	Se afecta notablemente la funcionalidad de los módulos dinámicos del sistema, haciendo que los procesos sean mas lentos y en el peor de los casos niegue el acceso a los usuarios por falta de capacidad del servidor.
Monitor	Administrador Web, EISI-UIS	Administrador Web, EISI-UIS
Responsabilidad	EISI-UIS	EISI-UIS
Contingencia	El servidor diariamente y a manera de seguridad, está en prueba por parte de los miembros de la EISI, lo cual garantiza un tiempo mínimo en la caída del servidor. El administrador de la Web subirá el servidor las veces que sea necesario.	Si se llegase a pasar el tope de usuarios concurrentes, el usuario nota una leve demora del sistema, en este caso debe procederse a

#### 4.6 COSTOS DEL PROYECTO

Luego de la evaluación de los diferentes conceptos y requerimientos que involucran costos en la elaboración del proyecto se estimo la suma de **17'730.000** como el valor necesario para su elaboración. Este valor es justificado en el presupuesto de la tabla 8.

Tabla 8: Presupuesto timado para el desarrollo del proyecto

	Concepto	Valor/Und. [\$/mes]	Dedicación	Cantidad [meses]	Subtotal [€]
<b>Personal</b>	Dirección				
	Director	4'500.000	0.1	06	2'700.000
	Ingeniería				
	De desarrollo (2)	1'000.000	1.0	06	12'000.000
<b>Equipo</b>	Computadores (1)	250.000		06	1'500.000
	Software	<sup>22</sup> 100.000		06	600.000
<b>Suministros</b>	Papelería <sup>23</sup>	30.000		12	360.000
	Libros	25.000		12	300.000
<b>Otros</b>	Internet	20.000		12	240.000
	Dispositivos de almacenamiento	30.000			30.000
<b>TOTAL</b>					<b>17'730.000</b>

#### 4.7 AGENDA DEL PROYECTO

Después de definir los requerimientos candidatos del sistema se realizó un cronograma que permite ver en detalle el tiempo requerido para la elaboración del proyecto. En el anexo A se presenta el cronograma del proyecto.

<sup>22</sup> Incluye Microsoft Windows 2000 Server, Magic Draw UML 5.0. No se aplica el costo total de las licencias sino el valor de su depreciación mensual.

<sup>23</sup> Incluye papel, tinta para impresora y fotocopias.

## **4.8 EVALUACIÓN DE LA FASE**

En cumplimiento de los objetivos para el desarrollo de esta fase, fue necesaria la realización de una investigación preliminar para cumplir con los objetivos de cada servicio ofrecido en la herramienta, de acuerdo a esto, se listaron los casos de uso candidatos para este fin, se identificaron los actores para cada uno de estos casos. Una vez identificados los casos de uso, se realizaron los diagramas globales correspondientes a estos, permitiendo crear una imagen más clara de la descripción individual de los casos correspondientes para cada uno de los módulos.

Finalmente en esta fase se evaluó la arquitectura candidata del sistema, se identificaron los posibles riesgos que pueden presentarse, aunque estos no atentan contra la finalidad del sistema en todo momento, es importante identificarlos a fin de hallar posibles estrategias que mitiguen su presencia. También se presenta en esta fase la agenda de desarrollo del proyecto y su costo aproximado.

## CAPITULO 5 FASE DE ELABORACIÓN

### 5.1 GENERALIDADES

El objetivo principal de esta fase es construir una arquitectura estable, que guíe el desarrollo del sistema a lo largo del ciclo. Además lleva al estudio del sistema propuesto al punto de planificar la fase de construcción con gran precisión. Los objetivos específicos planteados para esta fase son los siguientes:

- ⊕ Establecer la base de una arquitectura sólida para guiar el flujo de trabajo durante la fase de construcción.
  
- ⊕ Definir la mayor parte de los requisitos.

**5.1.1 Planeación.** Al comienzo de la fase de elaboración, se recibe un modelo de casos de uso parcialmente completo y una descripción general de la arquitectura candidata del sistema. Estos artefactos serán la guía y punto de partida para iniciar la búsqueda de una arquitectura estable que sea consistente durante la construcción del sistema.

Para cumplir con esta meta se adoptará un punto de vista general del sistema. En algunos casos, en los que los riesgos técnicos predominen, o sean los más significativos, se necesitará profundizar para establecer una arquitectura sólida. Por tanto se tomarán decisiones de la arquitectura con base en la comprensión del sistema en su totalidad: su ámbito, sus requisitos funcionales y no funcionales. También al final de esta fase, se habrá acumulado la información necesaria para planificar la fase de construcción.

**5.1.2 Flujos, actividades y productos a entregar.** En la fase de elaboración se busca una arquitectura estable que soporte el desarrollo posterior en la fase de construcción.

Para lograr éste objetivo se deben perfeccionar los casos de uso ya identificados, identificar nuevos si se requieren y detallar los necesarios para el entendimiento de la arquitectura. Además, se debe realizar un análisis de los paquetes que componen la herramienta y el posterior diseño de sus subsistemas. Luego, hacia el final de la fase se debe definir el diseño de la interfaz.

## **5.2 CAPTURA DE REQUISITOS**

En este flujo de trabajo se establecerá la prioridad y estructura de los casos de uso, en especial aquellos que sean de importancia para la definición de la arquitectura. Para ello se evaluarán los casos de uso descritos con anterioridad, prestando total atención a cada detallan cuanto a su descripción, flujo e interacción con los actores del sistema con el fin de evitar que los casos de uso importantes no se tomen en cuenta.

Debido a que en la fase de inicio la labor de investigación en torno a los procesos que debían vincularse en cada uno de los módulos de la herramienta, se realizó teniendo en cuenta el más mínimo detalle; para la fase de elaboración no se hace necesaria la descripción de nuevos casos de uso, permitiendo realizar la identificación de los paquetes de análisis en torno a estos casos de uso ya descritos.

## **5.3 ANÁLISIS**

En la fase de elaboración se presenta un modelo refinado de análisis, teniendo en cuenta los casos de uso significativos desde el punto de vista de la arquitectura y se trabajara con los casos de uso más complejos que necesiten ser refinados a fin de entender los detalles de la herramienta.

**5.3.1 Análisis de la arquitectura** En la fase de inicio se identifica una arquitectura candidata de la herramienta, evaluando que esta fuera factible. Ahora, se extenderá este análisis hasta que pueda servir como soporte de una arquitectura ejecutable viable para la herramienta.

A continuación se identifican los paquetes de análisis más relevantes para la herramienta a partir de los casos de uso identificados y que ofrecen la mejor funcionalidad de esta.

Figura 15. Paquete de análisis de Gestión de Documentos

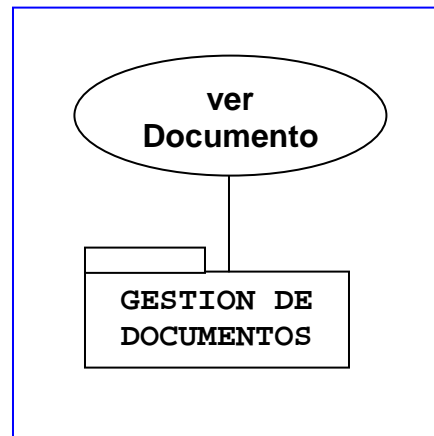


Figura 16. Paquete de análisis de Gestión de Ayuda

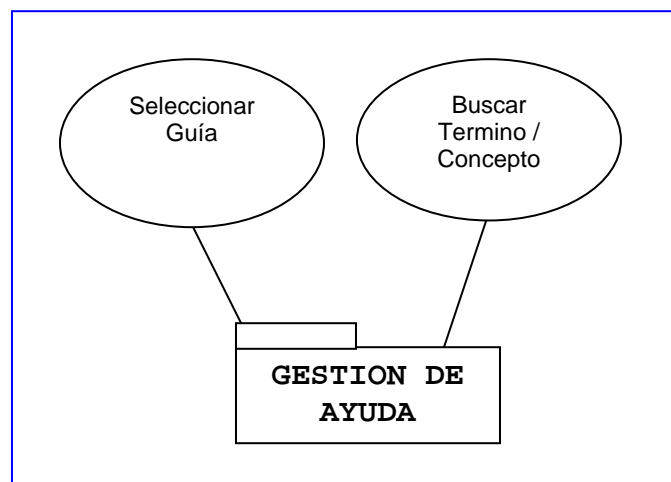


Figura 17. Paquete de análisis de Gestión de SEEPINT mail

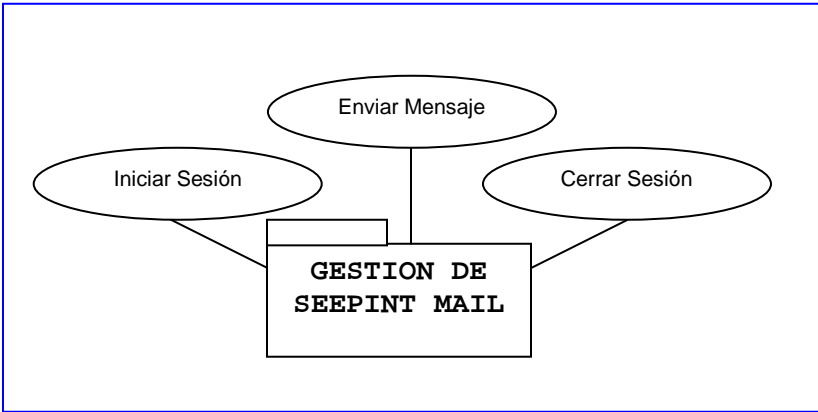
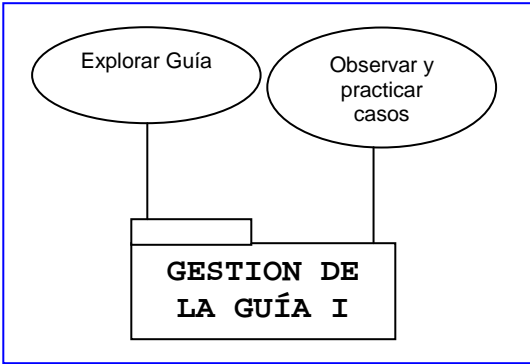


Figura 18. Paquete de análisis de Gestión de Laboratorios, Guía I , Guía II, Guía III, Guía IIV, Guía V, Guía VI. Estos seis últimos contenidos dentro del primero.

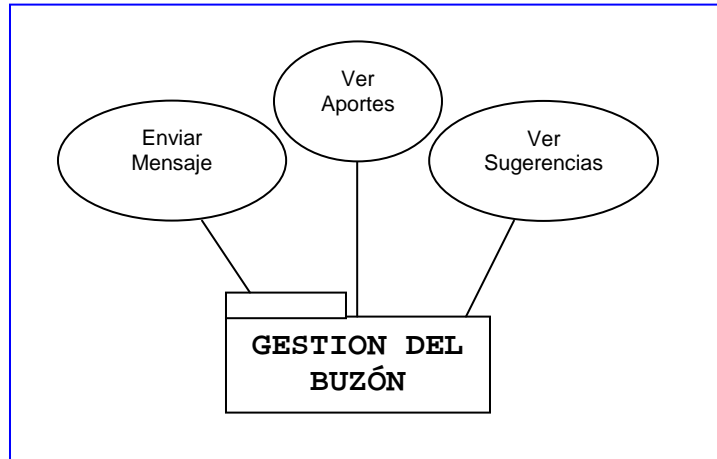


Nota, el paquete para las guías II, III, IV, V y VI es igual que el de la guía I, pero en la figura 19 se muestra una vista del paquete completo

Figura 19. Paquete Gestión de Laboratorio Completo.



Figura 20. Paquete de análisis de Gestión de BUZÓN



**5.3.2 Análisis de casos de uso.** La descripción de algunos casos de uso serán refinados de acuerdo a los modelos ya elaborados, teniendo en cuenta aquellos casos de uso que sean importantes para la definición de la arquitectura o para la comprensión de los requisitos. Para los demás casos de uso sólo se necesita tener una comprensión de lo que son y que el impacto que generan, para poder ser implementados en la etapa de construcción, sea mínimo.

A continuación se detallara el diagrama de colaboración para las guías I y II del paquete de análisis Gestión de laboratorios, estos es suficiente, para la definición de la arquitectura, es por esta razón que no se detallan la totalidad de las guías..

### 5.3.2.1 Paquete de Gestión de Laboratorios

En la figura 21 y 22 se detalla el caso de uso Solicitar Guía I y II mediante diagramas de colaboración.

Figura 21. Detalle de caso de uso Solicitar Guía I

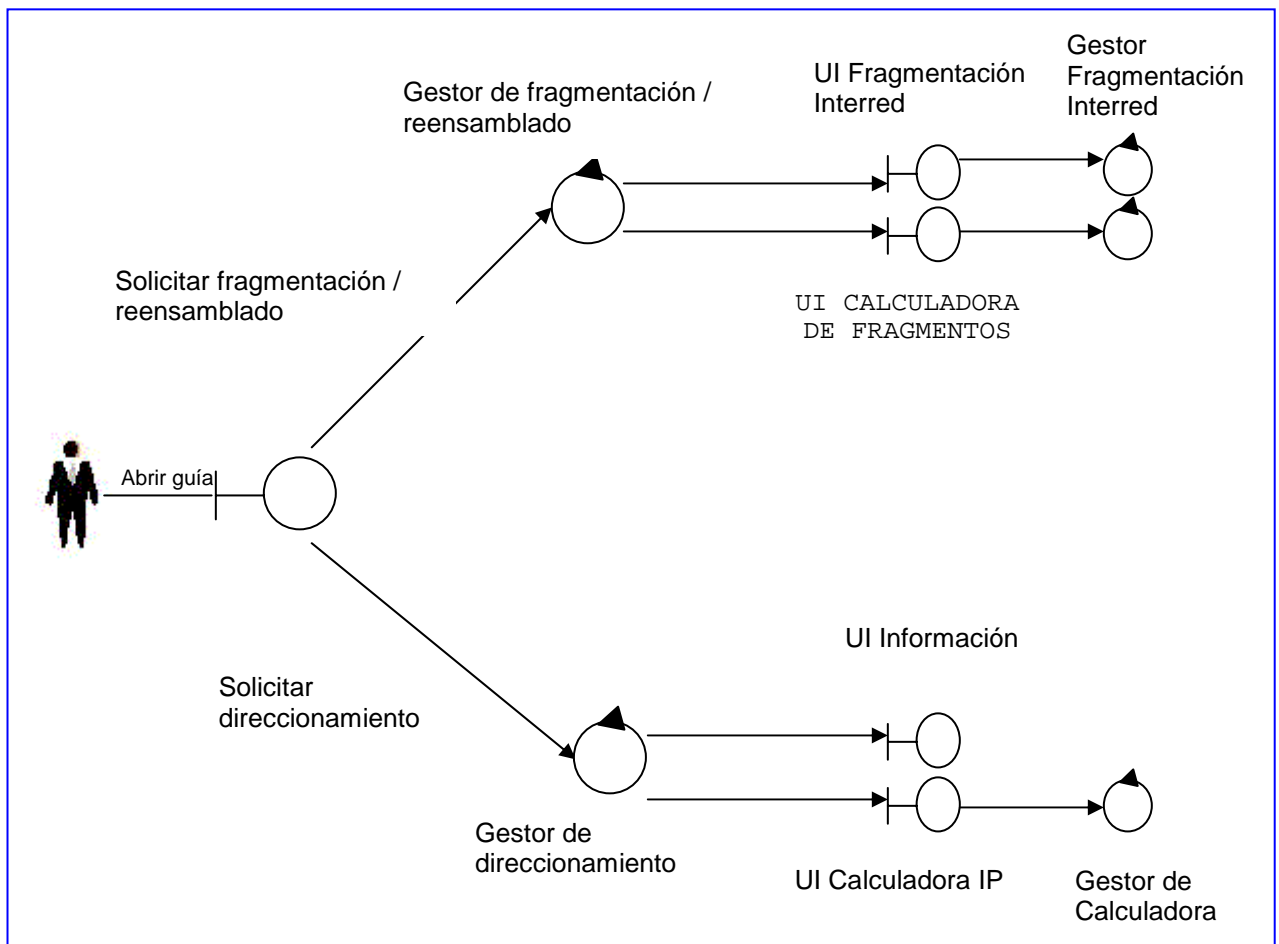
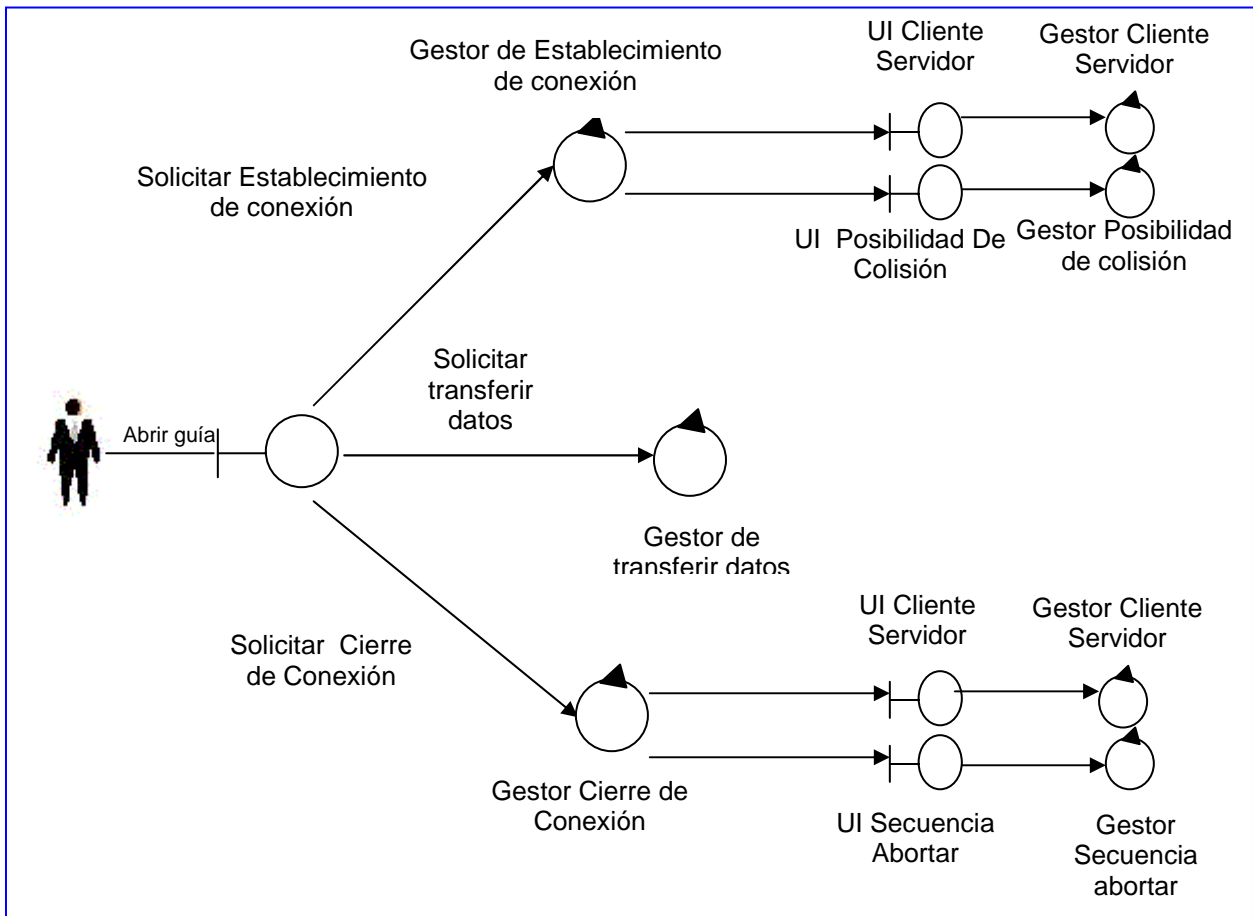


Figura 22. Detalle de caso de uso Solicitar Guía II



## 5.4 DISEÑO

En el diseño busca modelar el sistema y lograr que este soporte los requisitos funcionales, no funcionales y otras restricciones. Esta etapa es el resultado del modelo de análisis, el cual proporciona una comprensión más detallada de los requisitos y suministra la estructura del sistema que se debe conservar cuando se forma a éste.

El diseño tiene como propósito comprender con detenimiento aspectos relacionados con los requisitos no funcionales y restricciones relacionadas con los lenguajes de programación, los componentes, sistemas operativos entre otros. Esto contribuye a crear una arquitectura sólida y estable.

### **5.4.1 Diseño de la arquitectura**

El objetivo del diseño de la arquitectura es identificar y plasmar los modelos de diseño y despliegue y su arquitectura mediante la identificación de los nodos, configuraciones de red, subsistemas, interfases y software del sistema.

#### **5.4.1.1 Nodos y configuraciones de Red**

Para esta herramienta se identifican dos clases de nodos: un nodo servidor en el que reposa la capa lógica de aplicaciones y el nodo cliente que se encargara de la capa de presentación en un modelo de arquitectura cliente servidor, siendo este esquema la arquitectura a seguir.

Dado que el sistema será desarrollado con tecnología web se hace evidente de los protocolos http y TCP/IP para la comunicación entre los dos nodos anteriormente mencionados.

#### **5.4.1.2 Subsistemas**

Los subsistemas se identifican como un medio para organizar el modelo de diseño en piezas manejables, a demás de facilitar el trabajo de diseño y su posterior implementación. Un subsistema debe ser cohesivo, es decir sus contenidos deben estar fuertemente asociados y débilmente acoplados, es decir, tiene una dependencia mínima con los demás subsistemas.

A continuación se identificaran los subsistemas a partir de los paquetes de análisis. Durante esta fase se realizo una descomposición adecuada del sistema en paquetes, por esto es posible utilizarlos como base para los subsistemas de diseño. Contamos entonces con los subsistemas de: Documentación, Ayuda, SEEPIN mail, Laboratorios y Buzón.

## 5.4.2 Diseño de Interfaz de Usuario

La interfaz de usuario es la que determina que tan utilizable y funcional es un sistema definiendo de este modo el uso o deserción por parte de los usuarios. Teniendo en cuenta lo anterior, para el desarrollo de la interfaz se siguieron los siguientes lineamientos.

- ⊕ **Distribución consistente:** Las pantallas de la aplicación web deben caracterizarse por una distribución consistente en texto, imágenes y controles gráficos. Se deben prever todas las acciones erróneas de usuario en su interacción con el sistema (No elegir un elemento de una lista de selección, digitar texto donde se requiera un valor numérico, dejar un campo en blanco donde no es permitido, entre otras), el sistema debe mostrar mensajes de aviso para estos errores con el fin de mitigar la posible inserción de datos incorrectos.
- ⊕ **Flujo de navegación:** El paso de una pantalla a otra debe ser coherente con el trabajo que el usuario intente realizar.
- ⊕ **Evitar la saturación de las pantallas:** La comprensión que tiene el usuario de la tarea que realiza en el sistema es fundamental como parámetro de diseño de la interfaz, por ejemplo si para realizar una determinada tarea se requiere de la obtención de mucha información es conveniente dividir en diferentes pantallas la captura de esta o establecer opciones.

Es de gran importancia tener en cuenta los lineamientos que existen en torno al desarrollo de aplicaciones web, se deben evitar los posibles errores de diseño de la interfaz, ya que estos seguramente harán que el usuario del sistema se confunda y abandone su tarea.

#### **5.4.2.1 Errores que deben evitarse en el diseño de una buena interfaz<sup>24</sup>**

Inconsistencia en la navegación; debe mostrarse consistencia en torno a como el usuario se presentan al usuario las barras de navegación, los botones de selección y formularios de interacción con el sistema entre otros, evitando que el usuario pierda tiempo ubicándose al navegar, ya que estará en capacidad de identificar que parámetros caracterizan las herramientas del sistema a las que dará mayor uso.

Existencia de vínculos rotos; es importante evaluar mediante pruebas de buena operación de todos los vínculos presentados en el sistema.

El uso de marcos; conocidos también como frames, permiten que el sistema se muestre más ordenado, evitando la inserción de imágenes o barras de navegación.

Símbolos de en construcción; El sistema debe presentar la únicamente la información que tiene validez para los usuarios.

#### **5.4.2.2 Distribución de la información de acuerdo a los contenidos mostrados.**

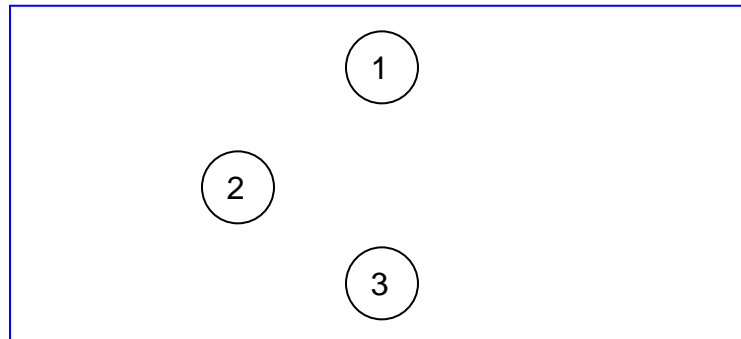
En la herramienta la distribución de la información, se centra en el modulo Laboratorios, teniendo cuidado de mantener los espacios adecuados para los otros módulos, ya que si no son el centro de la herramienta, contribuyen de buena forma en el éxito de los laboratorios, por ejemplo si existe un termino que no se entiende, se puede buscar su concepto en el modulo Ayuda.

A continuación se muestra y describe la distribución de la información que se plantea con la interfaz diseñada.

---

<sup>24</sup> Referido de “ los siete pecados mortales de un sitio web (porque deben evitarse a toda costa)”, Jesse Berst; -Seven Deadly Website Sins-. [http://www.zdnet.com/anchordesk/story\\_1716.html](http://www.zdnet.com/anchordesk/story_1716.html)

Figura 22. Distribución de la pantalla para la página principal de la Herramienta Web.



La pantalla principal de la herramienta se encuentra distribuida en tres áreas, la N° 1 se denomina encabezado y contiene los vínculos que permiten al usuario navegar y acceder a los diferentes módulos de la herramienta. El área N° 2 denominada Información, permite visualizar los contenidos que se despliegan una vez el usuario halla elegido una opción de la zona N° 1. y la zona N° 3 denominada pie, contiene el nombre completo de la Herramienta y los vínculos que permiten al usuario retornar y acceder a la pagina principal aplicadas sobre la zona N° 2

## 5.5 IMPLEMENTACIÓN

Este flujo de trabajo implementa y prueba los componentes arquitectónicamente significativos a partir de los elementos de diseño obtenidos. El resultado que se obtiene se toma como la línea base de la arquitectura.

La mayor parte de la arquitectura del sistema e capturada durante el diseño. La implementación tiene como propósito desarrollar la arquitectura y el sistema como un todo. Dado el enfoque incremental del proceso unificado, la implementación se dará a través de una sucesión de pasos pequeños y manejables.

## 5.6 EVALUACIÓN

Se deberá determinar si la arquitectura escogida es sólida y estable cómo para guiar el trabajo a lo largo de la fase de construcción y durante toda la vida útil del sistema. Esta arquitectura deba ser adaptativa en cuanto a cambios a demás de permitir el fácil crecimiento del sistema.

Para lograr éste arquitectura se deben analizar aquellos casos de uso que tengan impacto sobre la arquitectura. También, debe continuarse el control de los riesgos críticos surgidos desde la fase anterior y aquellos que puedan surgir durante el desarrollo de esta fase.

**5.6.1 Cumplimiento de criterios** En la búsqueda del principal objetivo de esta fase, el cual es encontrar la arquitectura estable que guíe el desarrollo del sistema, se han enfocado a las tareas de los diferentes flujos de trabajo, obteniendo la vista de la arquitectura estable. Durante esta fase se identificaron la mayor parte de los requisitos del sistema, se verificó la importancia de los actores encontrados en la fase anterior y se detalló los casos de uso significativos, donde se notó que la mayoría de estos no tienen complicaciones relevantes en su desarrollo, siendo suficiente con presentar una descripción de los mismos para ser comprendidos en su totalidad e implementarlos en la fase de construcción.

Las opciones que debería soportar el sistema y la manera como estas se organizarían para brindar al usuario una fácil visualización de las mismas, fueron de fácil obtención gracias a los casos de uso y a los subsistemas de diseño obtenidos a partir de los paquetes de análisis.

## **CAPÍTULO 6 FASE DE CONSTRUCCIÓN**

El objetivo general de esta fase es crear un producto en su versión operativa inicial, también conocida como “versión beta”, lista para ser usada, con todos los casos de uso implementados y probados. Durante esta fase la atención y el trabajo se centran principalmente en los flujos de diseño e implementación. Al final de esta fase el producto deberá cumplir con todos los requisitos planteados en las fases anteriores.

Los objetivos específicos de esta fase son:

- Identificar, describir, analizar, diseñar e implementar la totalidad de los casos de uso.
- El mantenimiento de la integridad de la arquitectura, modificándola sólo cuando sea necesario.
- Monitorizar los riesgos críticos y significativos tratados en las primeras fases.
- Realizar la planeación preliminar de la fase de transición (fase a realizar en un proyecto de grado posterior).

### **6.1 DESARROLLO DEL SOFTWARE**

A continuación se explica con detalle el desarrollo de la herramienta y la interfaz definida teniendo en cuenta los aspectos tratados y evaluados en las fases anteriores, además de la presentación de formatos entrada/salida, menús entre otros.

**6.1.1 Estructura de los programas** Para generar los programas realizados en el desarrollo de la herramienta se tuvo en cuenta la estructura de directorios que se presenta en la tabla 9.

Tabla 9. Estructura de directorios en el servidor que aloja la herramienta

DIRECTORIOS	MODULOS
Archivos F	Archivos Fuente .java
Ayuda	Modulo Ayuda
Buzon	Modulo Buzón
Documentacion	Modulo Documentación
Imágenes	Interfaz
Index.html	Interfaz
Laboratorios	Modulo Laboratorios
Rfc	Modulo Documentación

**6.1.2 Entrada y salida de datos** Para apreciar el funcionamiento correcto de los applets de java es importante validar la información que suministra el usuario por teclado, para lograr esto se tuvo en cuenta lo siguiente:

- ⊕ Antes de realizar cualquier ejecución de los applets, se verifica que los datos suministrados se encuentren completos.
- ⊕ Se valida la correspondencia entre el tipo de dato solicitado y el tipo de dato suministrado.
- ⊕ A nivel de código fuente se deben realizar comentarios descriptivos con el fin de facilitar su comprensión.

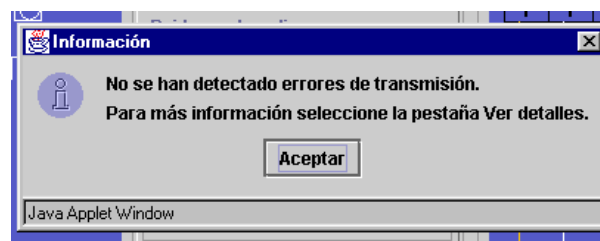
**6.1.3 Interacción con el usuario** Para facilitar la interacción de los usuarios con la herramienta se hace necesario tener en cuenta aspectos como:

- ⊕ Evitar ambigüedades, es decir, que exista correspondencia entre la acción solicitada por el usuario y la respuesta de la herramienta.
- ⊕ Dar sensación de avance de haber pasado de una etapa a otra cuando se realiza alguna acción.
- ⊕ Minimizar la cantidad de información que el usuario deba digitar.
- ⊕ Permitir acciones reversibles.
- ⊕ Emplear la misma secuencia de acciones cuando se trate de situaciones similares, esto permite que el usuario se familiarice con la herramienta.
- ⊕ Utilizar menús desplegables para el acceso a diferentes opciones.
- ⊕ Manejar los errores a través de mensajes que describan el problema en un lenguaje de fácil comprensión.
- ⊕ Mantener características visuales de la pantalla como son el tipo de letra, los colores, entre otros.

**6.1.4 Mensajes del sistema** El manejo de mensajes contribuye a mejorar la calidad de la herramienta y reduce significativamente la frustración de los usuarios cuando aparecen errores. Estos mensajes son: tipo informativo, advertencia y confirmación de eventos

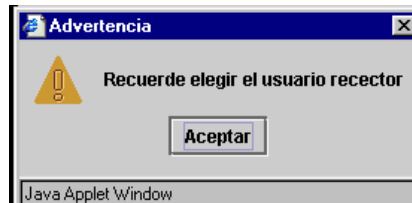
- ⊕ Mensajes tipo informativo: Suministran al usuario información sobre la ocurrencia de un evento desencadenado por una acción del usuario.

Figura 23. Mensaje tipo advertencia del sistema.



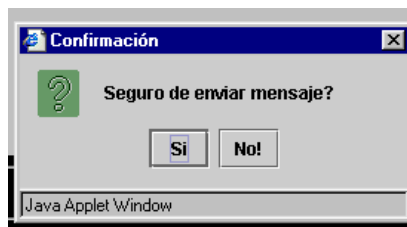
⊕ Mensajes tipo Advertencia: Suministran al usuario información sobre la ocurrencia de acciones no validas desencadenadas por una omisión del usuario o en respuesta a una modificación no válida. Estos mensajes no permiten que el usuario avance hasta no corregir error u omisión.

Figura 24. Mensaje tipo advertencia del sistema.



⊕ Mensaje de confirmación: Permite al usuario tomar una decisión, en cuanto a si debe o no continuar hacia la ocurrencia de un evento.

Figura 25. Mensaje tipo confirmación del sistema.



**6.1.5 tratamiento de riesgos** Todo proyecto tiene implícito riesgos los cuales pueden ser identificados a fin de buscar mecanismos que permitan ser mitigados. Durante el desarrollo se detectaron riesgos que pueden alterar su correcto funcionamiento. Estos riesgos así como los controles establecidos para mitigarlos se pueden apreciar en la tabla 10

Tabla 10. Controles implementados para mitigar los riesgos de la herramienta

<b>Ingreso y modificación de datos</b>
La herramienta enfrenta este riesgo validando la información que digitan los usuarios, verificando que los campos considerados como requeridos tengan la información solicitada antes de realizar alguna ejecutar la acción solicitada.
<b>Estado de los procesos</b>
Para evitar que el usuario pierda la secuencia de los estados de los diferentes procesos, se han implementado en los applets de la herramienta secuencias informativas cuya finalidad es mantener al usuario al tanto de lo que acontece en algunos procesos en los que resulta un tanto complicado llevar un seguimiento visual únicamente. En otros casos se muestra un cuadro resumen en el que se visualiza un detalle del proceso realizado.
<b>Evolución de la herramienta</b>
Para facilitar la comprensión de los algoritmos que permiten la realización de cada uno de los applets, el código fuente ha sido detallado con comentarios que facilitan esta labor. De este modo se deja abierta la posibilidad de hacer modificaciones, en los applets, que permitan mejorarlos o ampliar su contenido.

## 6.2 SUBSISTEMA DE GESTION DE DOCUMENTOS

En las fases anteriores se ha hecho mención de subsistema de gestión de documentos, en esta fase se mostrará la composición de este subsistema, la cual será detallada en tabla 11. así como es aspecto visual de algunos de los documentos contenidos en este subsistema.

Tabla 11. Documentos que conforman el subsistema gestión de documentos.

DOCUMENTO	DESCRIPCIÓN
Introduccion.html	Es indispensable, aterrizar la temática de la herramienta con el entorno en el cual se disponen sus temas.
TCP_IP.html	Es conveniente tenga una idea a cerca de la familia de protocolos TCP/IP y que papel desempeñan los protocolos TCP e IP en ésta familia.
SMTP.html	Dado que la aplicación elegida como preámbulo del conjunto de laboratorios de esta herramienta, es el correo electrónico, se considera necesario que los usuarios tengan conocimiento del protocolo de aplicación de correo electrónico dentro de la familia TCP/IP.
Puertos.html	A un que no son un tema abordado en los laboratorios de esta herramienta, se considera apropiado conocer el concepto de puerto y su labor dentro de la pila TCP/IP.
Mapas.html	Es un documento que permite el acceso a esquemas conceptuales que permiten interpretar la composición y el funcionamiento de los protocolos TCP e IP y a también acceder a los documentos RFC793 y RFC791.

Las figuras que se muestran a continuación son una vista de la información contenida en los documentos descritos en el subsistema a que se hace mención.

Figura 26. Vista del contenido del modulo de documentos

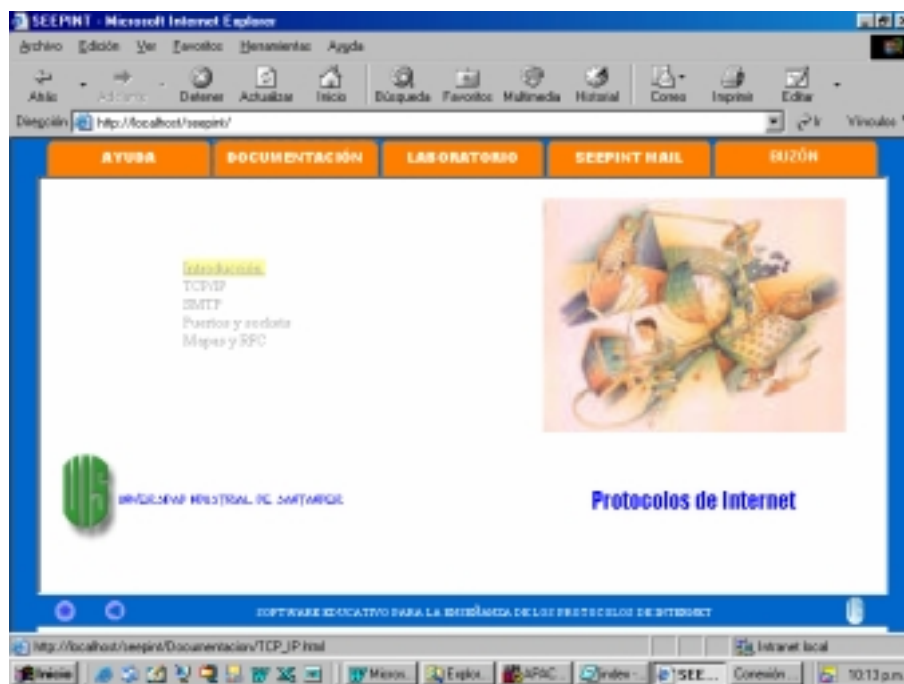
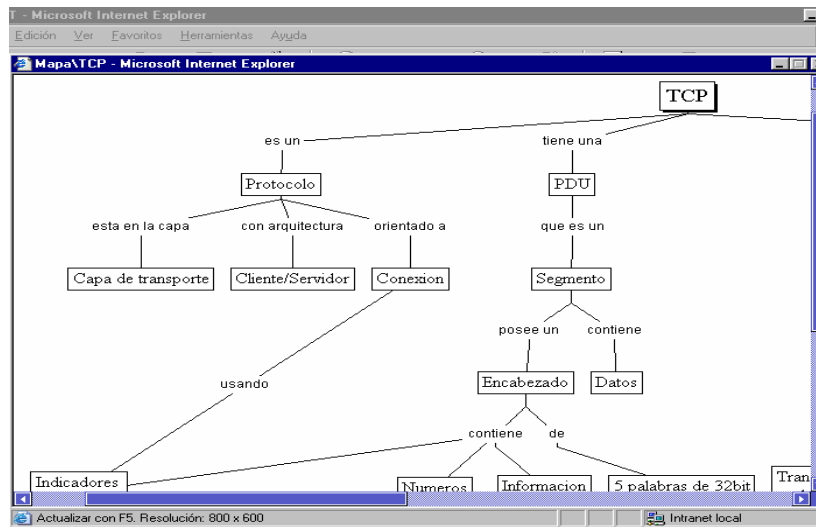


Figura 27. Vista al mapa conceptual del protocolo TCP



### 6.3 SUBSISTEMA DE GESTION DE AYUDA

El subsistema de ayuda sirve al visitante como soporte en la exploración de la del módulo laboratorios y a demás contiene un diccionario, con algunas palabras y su concepto, muchas de las cuales han sido utilizadas en contenido de la herramienta y otras tantas relacionadas con los temas tratados.

Las siguientes figuras son una vista a la interfaz del módulo ayuda.

Figura 28. Vista del módulo ayuda – Opción guías.

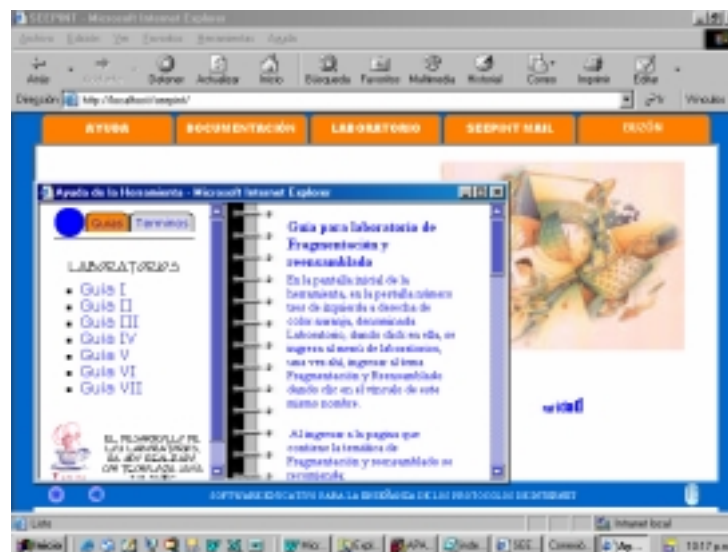
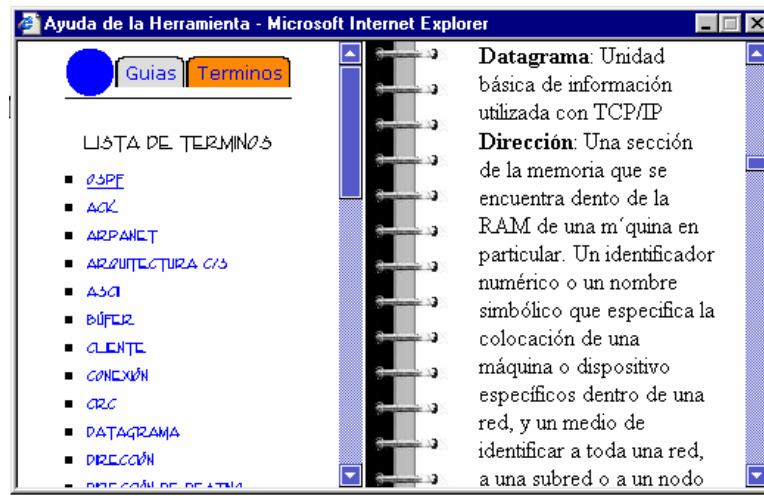


Figura 29. Vista del módulo ayuda – Opción términos

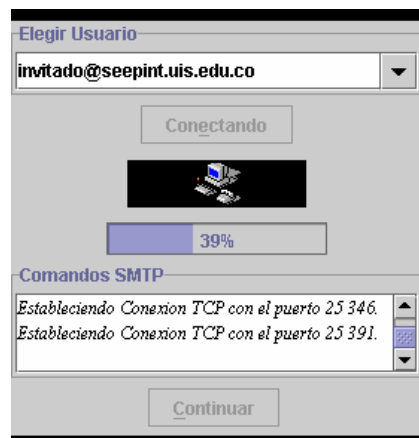


#### 6.4 SUBSISTEMA DE GESTIÓN DE SEEPINTmail

SEEPINT mail es un subsistema construido con la finalidad de mostrar una de las aplicaciones que solicita y hace uso de los protocolos TCP e IP. Este subsistema, ha sido concebido en cuatro etapas, denominadas así:

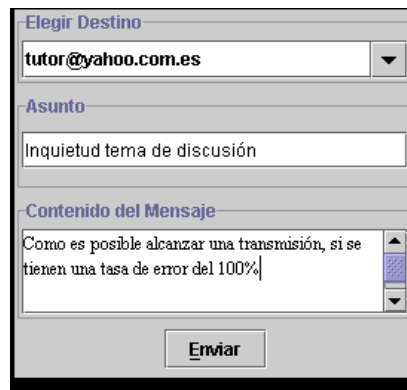
Conexión: Interfaz que emula la conexión de un usuario al servidor de correo SEEPINT MAIL, a continuación en la figura 31 se observa una vista de esta etapa.

Figura 30 Vista Etapa de Conexión



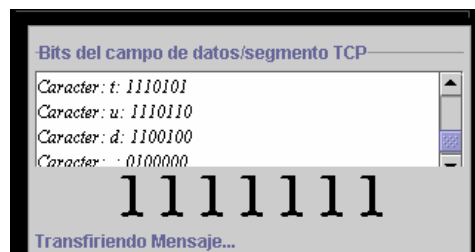
Cabecera y contenido del mensaje: Se toma como referencia la cabecera mínima del protocolo SMTP, en esta etapa se elige el destinatario del mensaje y se ingresa el asunto y el contenido de este, en la figura 32 observamos una vista de esta etapa.

Figura 31. Vista Etapa de Cabecera y contenido del mensaje



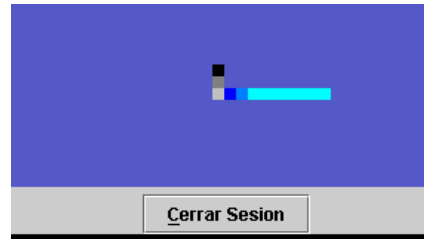
Envío: Se muestra el contenido del mensaje (Asunto y Mensaje) convertido a formato binario según el código ASCII, mientras es emulado el envío de estos caracteres. En la figura 33 se muestra esta etapa.

Figura 32. Etapa de Envío



Cerrar sesión: Se permite emular la desconexión del usuario conectado con el servidor de correo SEPINT MAIL.

Figura 33. Etapa Cerrar Sesión



## 6.5 SUBSISTEMA DE GESTIÓN DE LABORATORIOS

Dado que en las anteriores etapas, se ha hecho una descripción, muy sencilla acerca del contenido de los laboratorios, mostraremos cada uno de los Applets que hacen parte de los laboratorios de la herramienta, ya que estos son el principal componente de la herramienta.

6.5.1 Laboratorio de Fragmentación reensamblado y direccionamiento. Al igual que los siguientes laboratorios, la guía esta constituida, por un documento que contiene información documental y un marco que incluye los applets, en este caso el applet *fragmentacion.java* y *Direccionamiento.java*. En las siguientes figuras, se mostrara una vista de los applets mencionados.

Figura 34. Applet Fragmentación y Reensamblado



Figura 35. Applet Direccionamiento

Calculadora

Dirección IP: 35 . 128 . 127 . 6

Máscara de Subred: 255 . 255 . 128 . 0

Calcular

Red clase:	A
Dirección de red:	35.128.96.0
Número máximo de host:	32766
Dirección de broadcast:	35.128.223.255
Primera IP válida:	35.128.64.1
Última IP válida:	35.128.223.254

6.5.2 Laboratorio de Transferencia de datos entre puertos de origen y destino. Se encuentra constituido por una guía que contiene a su vez tres applets que muestran y describen las distintas alternativas contempladas dentro de TCP para las fases de establecimiento de una conexión, transferencia de datos y terminación de la conexión.

Figura 36. Applet de establecimiento de una conexión

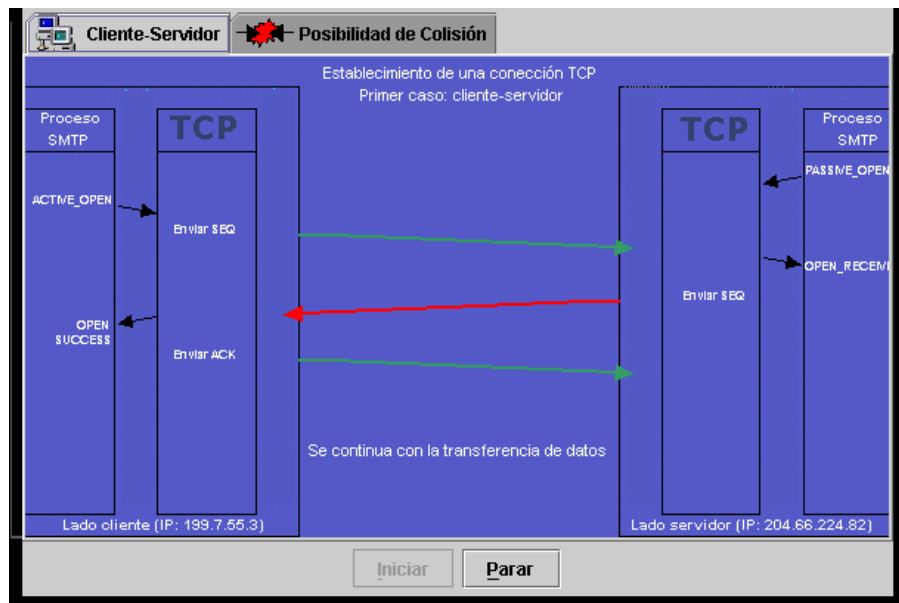


Figura 37. Applet de Transferencia de data gramas

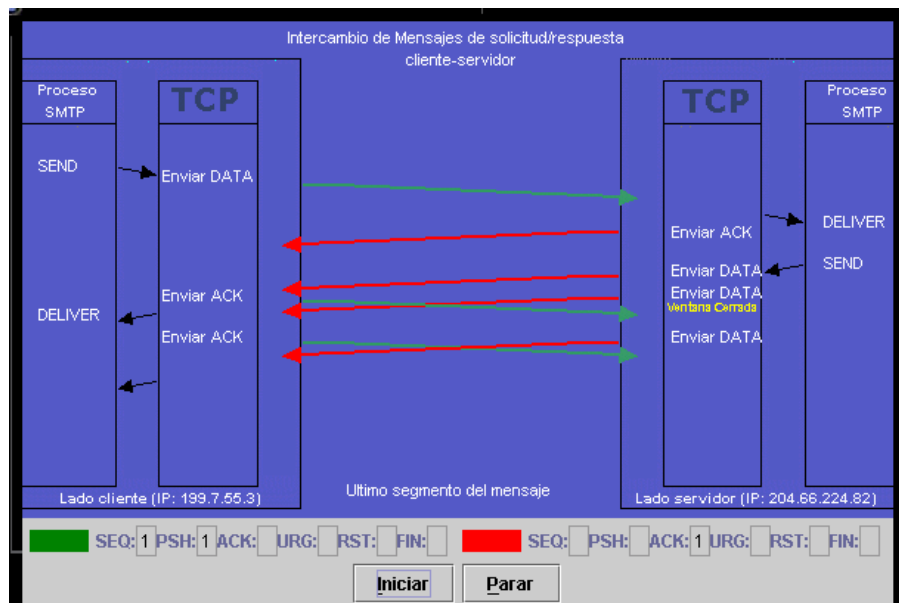
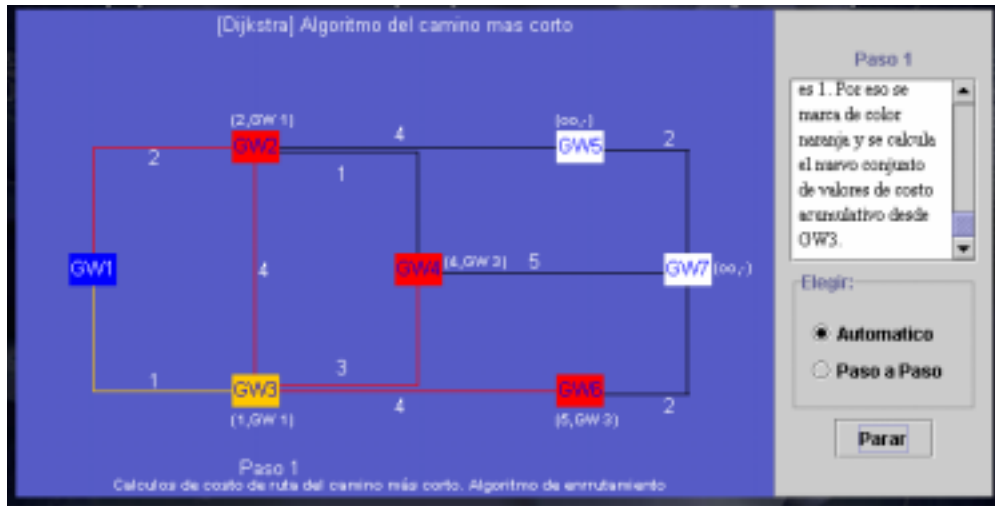




Figura 40 Applet algoritmo Dijkstra (Auxiliar)



6.5.4 Laboratorio de Detección de errores. Esta guía se encuentra compuesta por dos applets, que muestran el mecanismo de detección de errores que podrían estar incluidas en las diferentes implementaciones de TCP como son Bit de paridad y Suma de Bloque.

Figura 41 Applet Detección de errores por Bit de Paridad

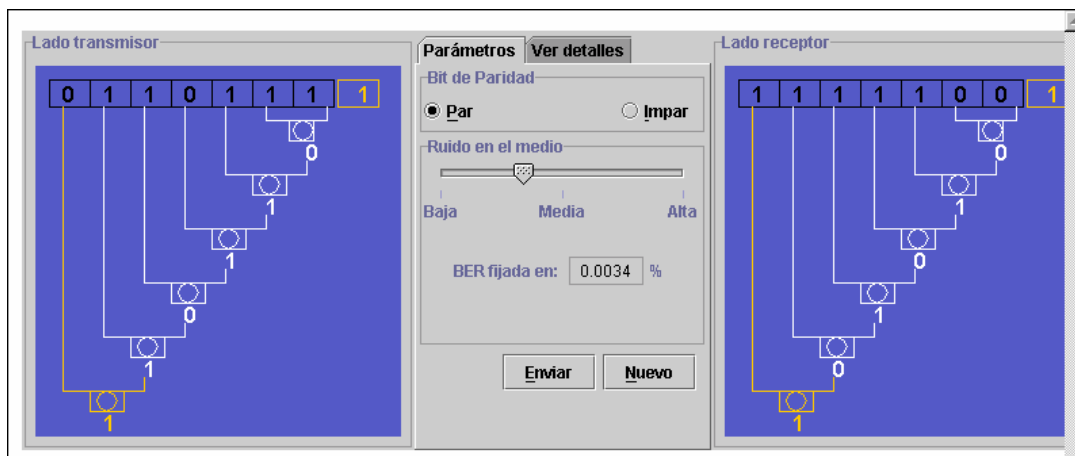


Figura 42 Applet Detección de Errores Por Suma De Bloque



6.5.5 Laboratorio de Control de Errores. Esta guía se encuentra compuesta por cuatro Applet, que describen, el principio de funcionamiento del RQ continua en sus mecanismos de repetición selectiva y retroceder n.

Figura 43. Applet Control de Errores RQ continua



Figura 44. Applet Control de Errores RQ continua – Repetición Selectiva Retransmisión Implícita

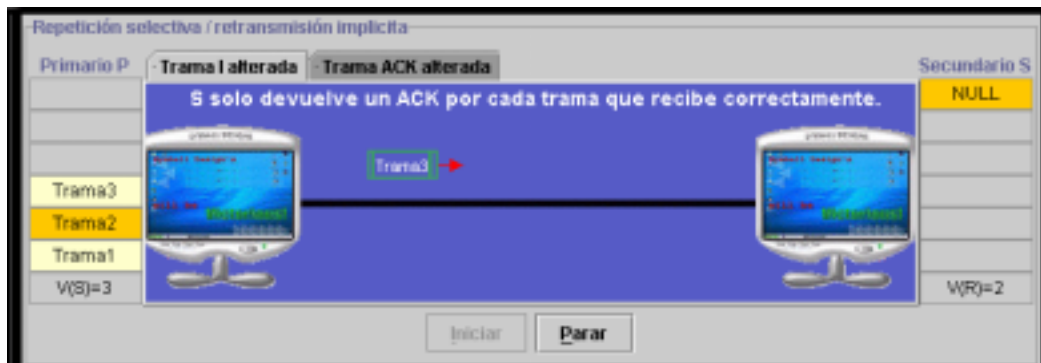


Figura 45. Applet Control de Errores RQ continua – Repetición Selectiva Solicitud Explicita

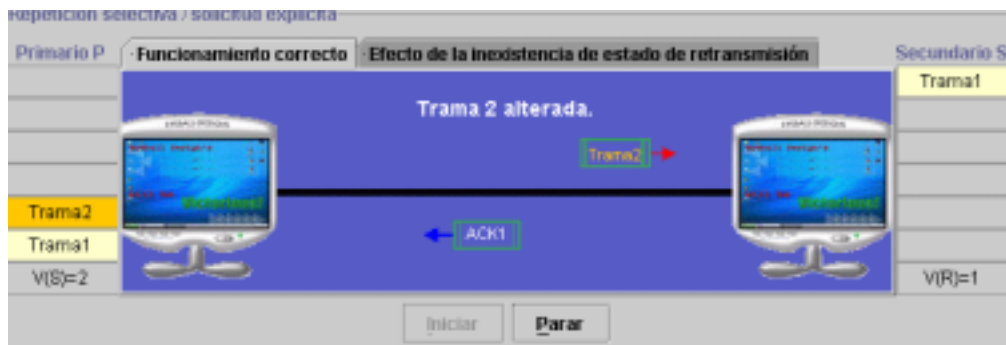
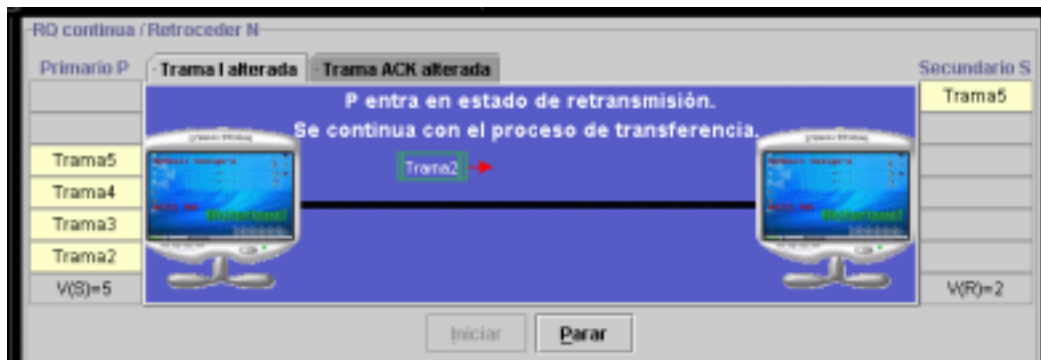


Figura 46. Applet Control de Errores RQ continua – Retroceder N

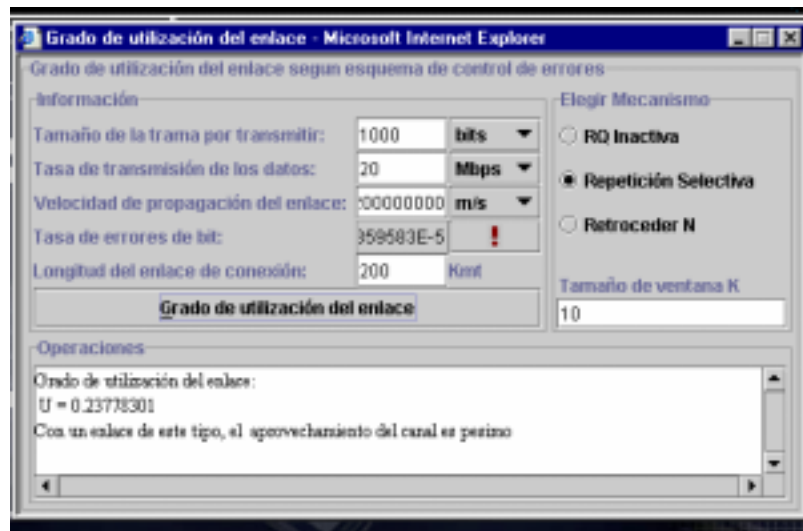


6.5.6 Laboratorio de Control de Flujo. Esta guía está compuesta por dos Applets, el primero muestra el mecanismo de una ventana deslizante de tamaños 1, 2 y 3. y el segundo permite que el usuario pueda optimizar el grado de utilización de un enlace, mediante la sensibilización de la ventana K.

Figura 47. Applet Control de flujo Ventana Deslizante



Figura 48. Applet Grado de Utilización del enlace



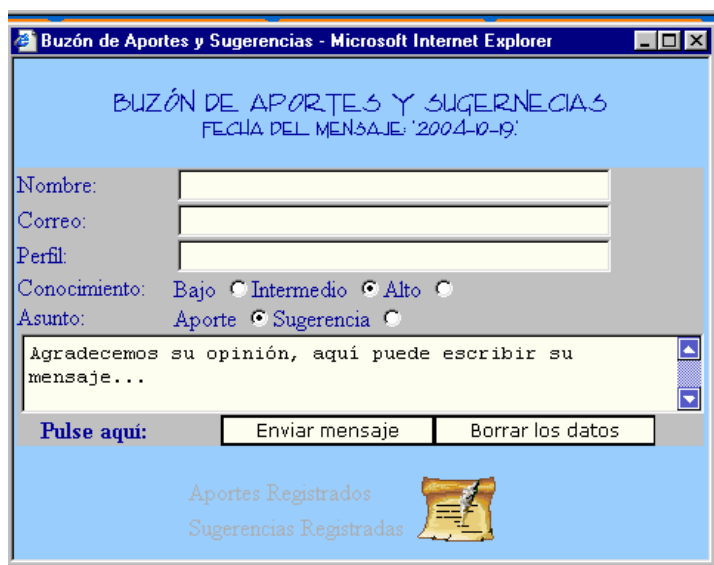
## 6.6 SUBSISTEMA DE GESTIÓN DE BUZÓN

Este subsistema descrito en las etapas anteriores, fue realizado, teniendo en cuenta que esta herramienta ha sido pensada no solo para que contenga el material actual, si no para que pueda irse desarrollando a medida que los usuarios, la comunidad académica baya realizando aportes y sugerencia, que sirvan de soporte a nuevas versiones de la herramienta, en pro del mejoramiento y depuración de contenidos.

El modulo de buzón es la herramienta que permite recolectar las inquietudes (aportes y sugerencias) formuladas por el actor visitante. Se encuentra implementada en PHP, que registran la información en documentos de texto, y que a la vez pueden ser visualizadas por todos los actores desde el mismo modulo, para que el actor visitante pueda conocer las inquietudes formuladas por otras personas y pueda contactarlas.

A continuación se muestra el formulario que permite la utilización del modulo buzón.

Figura 49. Formulario de Buzón de Aportes y Sugerencias.




The image shows a screenshot of a web browser window titled "Buzón de Aportes y Sugerencias - Microsoft Internet Explorer". The page content is as follows:

BUZÓN DE APORTE Y SUGERENCIAS  
FECHA DEL MENSAJE: '2004-10-19'

Nombre:   
Correo:   
Perfil:   
Conocimiento: Bajo  Intermedio  Alto   
Asunto: Aporte  Sugerencia

Agradecemos su opinión, aquí puede escribir su mensaje...

Pulse aquí:

Aportes Registrados   
Sugerencias Registradas

## 7. CONCLUSIONES

⊕ El adaptar una metodología para el desarrollo de un proyecto, como es el proceso unificado, es importante en el propósito de obtener un software de calidad, desarrollado e implementado cumpliendo con el tiempo y presupuesto determinados durante su planeación. Sin embargo, es importante tener en cuenta que el aplicar una secuencia de pasos única en seguimiento de una metodología para el desarrollo no nos garantiza el éxito del proyecto. La metodología en sí, aporta un marco de trabajo que debe ser adaptado al proyecto y a las circunstancias bajo las cuales se trabaja.

⊕ La utilización de herramientas de libre licencia como Java y PHP permitió a los autores aclarar las ventajas que trae consigo el uso de estas herramientas, tales como la baja inversión de pequeñas sumas de dinero en software y beneficios de seguridad, estabilidad y desempeño de las aplicaciones, gracias a que este tipo de herramientas, se encuentran respaldadas por grandes organizaciones que trabajan para que sean tan estables como las herramientas comerciales. Para el caso de entidades publicas como la universidad, y más aun para el desarrollo de herramientas educativas, es una ventaja de fácil aplicación, que permite estar a la vanguardia en tecnología, sin quedar atados a las restricciones de las licencias o contratos impuestos por los creadores de software propietario.

⊕ Este proyecto fue desarrollado como un eslabón, ante la expectativa de que más estudiantes, adapten nuevos contenidos a este tipo de herramientas, que le permita a la escuela no contar con unos pocos, sino con la mayoría de contenidos involucrados en el área de las telecomunicaciones, con el fin de hacer del aprendizaje una bonita experiencia de compartir el conocimiento con futuras generaciones de estudiantes.

## 8. RECOMENDACIONES

⊕ La herramienta realizada, esta construida para funcionar sobre una plataforma web, dado que posee un contenido, ligeramente pesado, seria aconsejable, que el servidor en el cual se alojara para su funcionamiento futuro, tuviera capacidades de procesamiento elevadas acorde con las circunstancias actuales, 2600 Mhz o superior. De tal manera que se alcance un rendimiento optimo en los clientes.

⊕ Para garantizar, un aprovechamiento adecuado del contenido de la herramienta, se insiste en que se haga una exploración secuencial de la herramienta de acuerdo a la numeración por guías que se maneja, además es indispensable que los usuarios tengan cierto conocimiento sobre la temática, de no ser así, se recomienda leer detenidamente todos los documentos del modulo de documentación.

## **BIBLIOGRAFIA**

JACOBSON, Ivar, BOOCH, Grady, RUMBAUGH, James. El Proceso Unificado de Desarrollo de Software. Primera edición. Addison Wesley. España 2000.

GROS, Begoña. Pautas Pedagógicas para la Elaboración de Software. Primera edición. Ariel S.A. Barcelona 1997.

TIMOTHY, Parker. Aprendiendo TCP/IP en 14 Días. Segunda edición. Prentice Hall. Mexico 1997.

HALSALL, Fred. Comunicación de Datos, Redes y Sistemas Abiertos. Cuarta Edición. Addison Wesley Longman 1997.

SCHILDT, Herbert. Manual de Referencia Java 2. Cuarta edición. Mc Graw Hill. Madrid 2001.

BYSHOP, Judy. Java Fundamentos de Programación. Segunda edición. Alfa. Madrid 1999.

## REFERENCIAS EN INTERNET

<http://java.sun.com/books/Series/Tutorial/index.html>. ADAPTACIÓN DEL TUTOR DE JAVA DE SUN POR JUAN ANTONIO CON ACTUALIZACIÓN 7/12/2000.

<http://java.sun.com>. Página de Sun dedicada a la tecnología java.

<http://www.jdance.com/>. Configuración e implementación de applets de java.

<http://www.desarrolloweb.com/>. Sitio, dedicado a la publicación de contenidos adecuados para el trabajo en web con JavaScript.

<http://members.tripod.com/~MoisesRBB/>. Sitio, dedicado a la publicación de material referente a los protocolos de comunicaciones.

<http://neo.lcc.uma.es/evirtual/cdd/applets/distancia%20corta/>. Sitio, dedicado a la publicación de contenidos acerca de los métodos de rutas críticas.

# ANEXOS

### ANEXO A. CRONOGRAMA DE ACTIVIDADES

ETAPAS Y TAREAS	MCS/SEMANAS																							
	MCS 1				MCS 2				MCS 3				MCS 4				MCS 5				MCS 6			
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
<b>FASE DE INICIO</b>																								
Requisitos																								
Análisis																								
Diseño																								
Implementación																								
<b>FASE DE ELABORACIÓN</b>																								
Requisitos																								
Análisis																								
Diseño																								
Implementación																								
Prueba																								
<b>CONSTRUCCIÓN Y ENTREGA</b>																								
Requisitos																								
Análisis																								
Diseño																								
Implementación																								
Prueba																								
<b>DOCUMENTACIÓN</b>																								

## ANEXO B ESCENARIOS DE PRUEBA

**DURANTE LA FASE DE CONSTRUCCIÓN, SE REALIZARON, DOS ESCENARIOS DE PRUEBA DESCRITOS ASÍ:**

*Escenarios de Pruebas 1*, denominadas pruebas de formato, con un grupo de usuarios seleccionados, entre los cuales se eligieron estudiantes con conocimientos básicos del tema, y profesionales con dominio del tema.

Acción, se explora la herramienta, permitiendo visualizar todos los módulos que la componen, recopilando la apreciación de las personas vinculadas a las pruebas.

El formato general de pruebas fue el siguiente:

El objeto de este formato es reunir las impresiones generales del usuario durante su contacto con la herramienta SEEPINT. Está estructurado en diferentes ítems que le permiten seleccionar la opción que considere más adecuada de acuerdo a su punto de vista.

Tenga en cuenta los siguientes criterios de evaluación:

- M: *Malo.*** El ítem evaluado no se encuentra acorde con sus expectativas.  
**R: *Regular.*** El ítem evaluado no satisface totalmente sus expectativas pero puede ser mejorado.  
**B: *Bueno.*** El ítem evaluado satisface sus expectativas, pero existen aspectos que podrían ser ajustados y que no influyen en el desempeño actual de la herramienta.  
**E: *Excelente.*** El ítem evaluado satisface totalmente sus expectativas.

### ERRORES DE CONTENIDO Y DE NAVEGACION

Califique el contenido del sitio teniendo en cuenta:

DESCRIPCION ITEM	M	R	B	E
Ortografía y gramática.				
La información es presentada en forma estructurada y comprensible al usuario.				
Los recursos gráficos contribuyen a la fácil asimilación del tema				
Distribución del espacio.				
Tamaño de las fuentes.				
Manejo de colores.				

Califique la facilidad de exploración del sitio teniendo en cuenta:

DESCRIPCION ITEM	M	R	B	E
Las opciones presentadas en el menú principal permiten acceder al sitio totalmente.				
Mientras explora el sitio le resulta fácil retornar a un punto determinado de éste.				
Los íconos de navegación se identifican fácilmente y se presentan en forma estándar.				
Existe coherencia entre el icono de direccionamiento y la página mostrada.				

### INTERESES EN USABILIDAD

DESCRIPCION ITEM	M	R	B	E
El tema presentado es de utilidad para complementar sus conocimientos profesionales.				
La herramienta le ofrece ventajas de aprendizaje en comparación a los métodos tradicionales de estudio en lo referente al tema.				
La herramienta es adecuada para ser utilizada como medio didáctico en la ilustración del tema.				
Considera que éste tipo de herramientas podría vincularse dentro del contenido de una materia que contemple este tema como medio para lograr los objetivos trazados.				

El resultado obtenido de diez personas que realizaron las pruebas, se plasma a continuación.

### ERRORES DE CONTENIDO Y DE NAVEGACION

Califique el contenido del sitio teniendo en cuenta:

DESCRIPCION ITEM	M	R	B	E
Ortografía y gramática.	2	4	3	1
La información es presentada en forma estructurada y comprensible al usuario.	0	4	6	0
Los recursos gráficos contribuyen a la fácil asimilación del tema	1	7	3	0
Distribución del espacio.	6	2	2	0
Tamaño de las fuentes.	8	0	2	0
Manejo de colores.	5	3	2	0

**CALIFIQUE LA FACILIDAD DE EXPLORACIÓN DEL SITIO TENIENDO EN CUENTA:**

<b>DESCRIPCION ITEM</b>	<b>M</b>	<b>R</b>	<b>B</b>	<b>E</b>
Las opciones presentadas en el menú principal permiten acceder al sitio totalmente.	0	3	6	1
Mientras explora el sitio le resulta fácil retornar a un punto determinado de éste.	1	4	5	0
Los íconos de navegación se identifican fácilmente y se presentan en forma estándar.	2	2	6	0
Existe coherencia entre el icono de direccionamiento y la página mostrada.	0	0	1	9

**INTERESES EN USABILIDAD**

<b>DESCRIPCION ITEM</b>	<b>M</b>	<b>R</b>	<b>B</b>	<b>E</b>
El tema presentado es de utilidad para complementar sus conocimientos profesionales.	1	5	4	0
La herramienta le ofrece ventajas de aprendizaje en comparación a los métodos tradicionales de estudio en lo referente al tema.	5	3	1	1
La herramienta es adecuada para ser utilizada como medio didáctico en la ilustración del tema.	2	3	4	1
Considera que éste tipo de herramientas podría vincularse dentro del contenido de una materia que contemple este tema como medio para lograr los objetivos trazados.	1	2	8	0

Como complemento al formato general de pruebas, se recopiló información abierta, de los usuarios vinculados a la prueba mediante la utilización del módulo buzón. En la siguiente tabla, describiremos algunas de las sugerencias más destacadas, y que nos hacen conscientes de las debilidades de la herramienta hasta la fecha en que se realizaron las pruebas.

Apreciaciones de los usuarios registradas en el buzón.

<p>Fecha y hora: Jueves_14-10-2004[01:34:29]          Nombre: Yehison Ortega          Correo: yortega_e@hotmail.com          Perfil: Estudiante Ingenieria de sistemas de ultimo nivel Nivel de conocimiento: Bajo          Mensaje: -El aspecto negro de la herramienta provoca fatiga visual, lo que puede conllevar a cierto desinterés, factor primordial en un sw educativo.          -Existen unos errores ortográficos en las palabras "conexión". . En los laboratorios existen algunos links que no son visibles. Deberían manejar algún</p>
---

tipo de patron que acostumbre al usuario a reconocer los links, como una imagen.

Fecha y hora: Jueves\_14-10-2004[01:45:09]

Nombre: Milena Pimentel Sanchez

Correo: milenillas1 hotmail.com Perfil: Estudiante Nivel de conocimiento:

Intermedio Mensaje: Bueno jovenes lo primero es el color, cambiar ese tono del fondo de las paginas, con un tono mas claro se pueden apreciar mas los hipervinulos y los iconos de navegacion, tambien hay que tener en cuenta la ortografia (conexion y receptor), y la fuente.

Fecha y hora: Jueves\_14-10-2004[03:50:20]

Nombre: Nini Johanna Arévalo O. Correo: ninijarevalo@hotmail.com

Perfil: Estudiante de Ing. Sistemas

Nivel de conocimiento: Bajo Mensaje: Comparto la opinion de mi compañero yehison que el color negro no es el mas adecuado (a pesar de que me guste el color)les sugiero investigar acerca de los parametros que ya estan establecidos (acercade letra y color, etc)para el diseño de paginas de acuerdo al publico al cual van dirigidas.Les aconsejo tambien agregar iconos que ayuden a identificar la ubicacion de los applets;y leer detenidamente la informacion ya que encuentre unos errores de digitación.Pero a pesar de esto LOS FELICITO NIÑOS HAN HECHO UN MUY BUEN TRABAJO!

Fecha y hora: Jueves\_14-10-2004[04:13:04]

Nombre: Ana Carolina Daza Rodriguez Correo: anacarodaza@hotmail.com

Perfil: Estudiante Ing de Sistemas Nivel de conocimiento: Bajo

Mensaje: Primero que todo los felicito por el trabajo y por la forma que manejan las sugerencias y aportes de nosotros. Como sugerencia les digo lo siguiente: 1. Con respecto al color seria recomendable que se pensara en cambiarlo ya que no resultaria agradable si se necesitara estar mucho tiempo frente a la herramienta. 2. En la parte de documentacion les recomiendo que deberian mostrar, lo que contiene cada documento para evitar recorrer la pagina, cuando se necesita algo muy especifico, y si se pudiera en el mejor de los casos colocar unos links que nos llevaran a los puntos del contenido. 3. En el laboratorio de encamamiento de datagrama en el applet de OSPF se deberia tener la opcion de DESHACER para facilitar el diseno de la topologia ya que solo existe la opcion de Reiniciar todo el diseno. Espero que de algo les sirva estas sugerencia

Fecha y hora: Miercoles\_20-10-2004[10:33:43]

Nombre: Ivan Dario Rueda Correo: idrueda@hotmail.com

Perfil: Estudiante de Ing Industrial y de Sistemas Nivel de conocimiento: Bajo

Mensaje: Creo que los vinculos deberian marcarse con algo que los identifique mas como vinculos, para que las personas puedan llegar con claridad a estos.

<p>Fecha y hora: Jueves_20-10-2004[10:03:59]  Nombre: Felix Andres Cepeda Gomez Correo: facg662@hotmail.com  Perfil: Estudiante de ing de sistemas e industrial Nivel de conocimiento: Intermedio  Mensaje: la herramienta esta muy bien, quizas los cambios que le haria serian de forma, algunos de los gif animados no se ven muy bien, tambien cambiaria si es posible el tipo y color de letra utilizado en la explicacion de los laboratorios para dare un poco mas de estilo.</p>
<p>Fecha y hora: Jueves_21-10-2004[14:03:59]  Nombre: Oscar Jaimes Correo: ojaimes@uis.edu.co  Perfil: Empleado UIS  Mensaje: 1. El tipo de letra que se esta usando, debe ser más llamativa y parrafos justificados.  2. Antes de empezar hablar de TCP/IP, deben de aclarar que es una red, las clases que hay ,y definir que son protofolos de comunicación.  3. En la presentación inicial, deben de hacer énfasis hacía que alumnos va dirigido el curso y con las capacidades que debe de contar para el pleno desarrollo del curso.</p>

Nota: Los errores de redacción y de ortografía, se debe a que los usuarios, a medida que van procesando una idea, escriben y lo hacen rápido para no perder el hilo de la prueba.

*Escenario de Pruebas 2*, denominadas pruebas de validación de parámetros, que fue realizado por los desarrolladores de la herramienta, con el fin de comprobar que los Applets validen la información que un usuario ingresa. Dado que la información que se ingresa a algunos Applets son solo campos de valores numéricos enteros y que tienen rangos definidos, en la siguiente tabla se describe las acciones que toma el Applet frente a un determinado ingreso.

<b>Applet</b>	<b>Nombre del Campo</b>	<b>Parámetro</b>	<b>Evento</b>
Applet Calculadora en Guía de Fragmentación y reensamblado.	NSDU	65537	Mensaje de Error
Applet Ventana deslizante	Valor de ventana	3	Incremento en la salida de tramas.
Applet Fragmentación	Congestión del GW2	Alto	Detención del proceso al llegar la trama al GW2
Applet Grado de utilización del enlace en Guía de Control de flujo.	Tamaño de la trama por transmitir	34r	El sistema anula el carácter invalido

Al realizar estas pruebas, queda claro, que la herramienta cumple con las expectativas planteadas, no obstante existen detalles que serán corregidos para el momento de la entrega de la herramienta a la Escuela de Ingeniería de Sistemas de la Universidad.

## ANEXO C. MANUAL DE INSTALACIÓN

**PARA PODER DAR USO A LA HERRAMIENTA, ES INDISPENSABLE, QUE CUALQUIER USUARIO, CONOZCA LA INFORMACIÓN NECESARIA Y SUFICIENTE, PARA PONER EN MARCHA EL FUNCIONAMIENTO DE LA MISMA, PARA TAL EFECTO DESCRIBIMOS CONTINUACIÓN LOS PASOS PARA ALCANZAR UNA INSTALACIÓN ADECUADA DEL PRODUCTO.**

Requisitos:

El equipo servidor, donde será alojado el directorio de archivos que conforman la herramienta, deberá tener instalado lo siguiente.

- ⊕ Un servidor web Apache actualización no inferior a la versión 1.3.23
- ⊕ Modulo de PHP superior a la versión 4.1.1.
- ⊕ Plug – in de Java 1.3.1
- ⊕ Browser Actualizado, Netscape Communicator 4.0 y Microsoft Explorer 4.0 o posteriores versiones de ambos navegadores.

El equipo cliente, donde será visualizada la herramienta por los usuarios, deberá tener instalado lo siguiente.

- ⊕ Plug – in de Java 1.3.1
- ⊕ Browser Actualizado, Netscape Communicator 4.0 y Microsoft Explorer 4.0 o posteriores versiones de ambos navegadores.

Nota: Para obtener un mejor aprovechamiento visual, se recomienda que el equipo cliente este posea una configuración de pantalla de 800 x 600 píxeles.

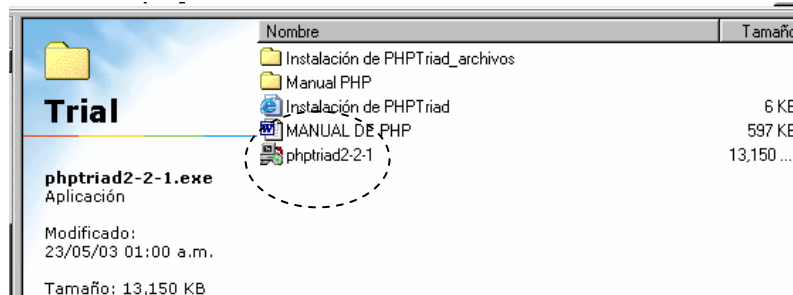
*Proceso de Instalación:*

Para preparar la máquina que hará el papel de servidor, el proceso inicia con la instalación del paquete phpriad2-2-1.exe que se encuentra adjunto a los archivos en el CD-ROM de la herramienta.

Pasos para instalar phpriad2-2-1.exe.

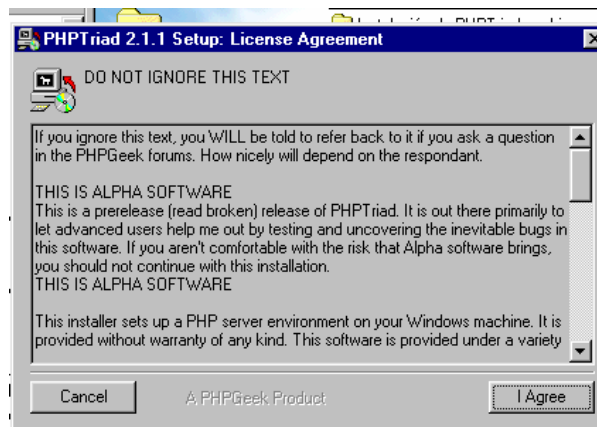
- ⊕ Paso 1: dar doble clic sobre el icono phpriad2-2-1.exe. que se muestra en la figura C –01.

Figura C-01 Icono de phpriad2-2-1.exe



⊕ Paso 2: En la ventana que se muestra en la figura C-02, una vez leído el mensaje de notificación, dar clic sobre el botón I Agree

Figura C-02 Iniciar el proceso de Instalación



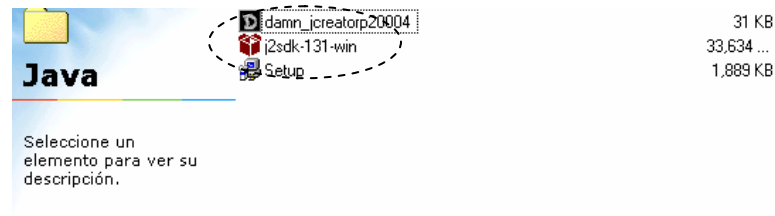
⊕ Paso 3: Esperar a que el paquete de instalación, guarde los archivos y configure el modulo de PHP. Una vez terminado el proceso, el usuario podrá cerrar la ventana, dando clic sobre el botón Close.

**UNA VEZ REALIZADO EL PROCESO ANTERIOR, SE CONTINUA CON LA INSTALACIÓN DEL PLUG – IN DE JAVA VERSIÓN 1.3.1. , QUE SE ENCUENTRA ADJUNTO A LOS ARCHIVOS DEL CD-ROM DE LA HERRAMIENTA.**

## Pasos para instalar j2sdk-131-win.exe

⊕ Paso 1: Para iniciar, debemos ejecutar la aplicación j2sdk-131-win.exe, mostramos el icono en la figura C-03

Figura A-03 Icono de j2sdk-131-win.exe



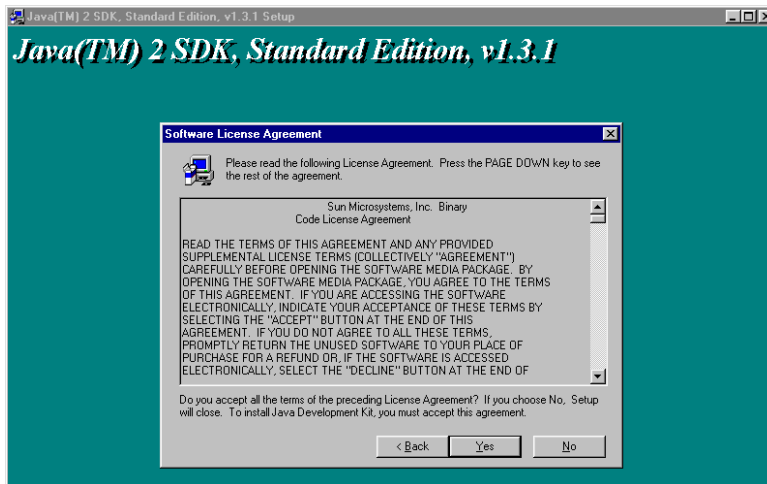
⊕ Paso 2: Una vez abierto el paquete, dar clic en el botón siguiente (Next) que se observa en la figura C-04.

Figura C-04 Ventana de Continuación Proceso de Instalación



⊕ Paso 3: Se continua el proceso con la lectura de la licencia y click sobre el botón Yes que se aprecia en la figura C-05.

Figura C-05 Ventana de Licencia Proceso de Instalación



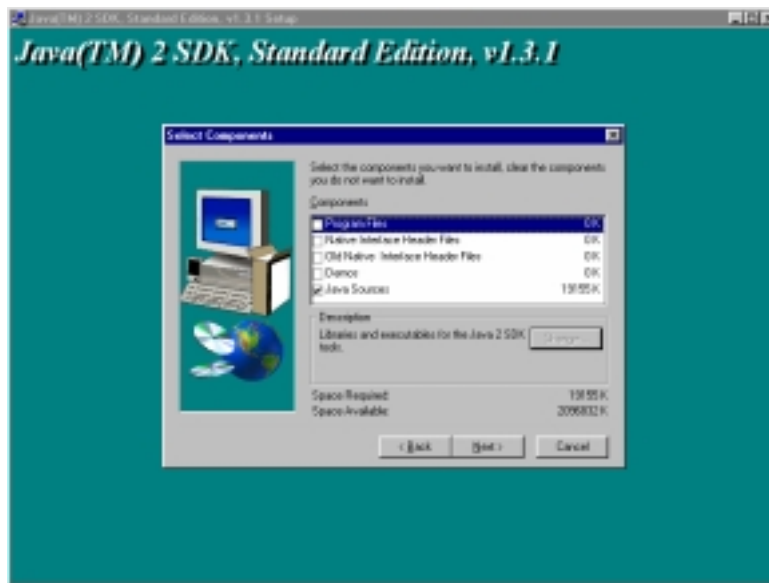
⊕ Paso 4: Una vez pasado a este paso, se debe confirmar si el Plug – In deberá ser instalado en la unidad principal, de lo contrario se debe elegir otra dando click sobre el botón Browser. En la figura C-06 se observa la ventana de confirmación.

Figura A-06. Ventana de confirmación de Ubicación de la instalación.



⊕ Paso 5: Para comenzar con la carga de archivos al sistema, debemos indicar al paquete en la ventana que se muestra en la figura C-07 que solo deba instalarse Java Sources, de modo que solo instale el Plug – In.

Figura C-07. Ventana de Elegir Opción de Instalación



⊕ Paso 7: una vez que se da clic en el botón Next (siguiente de la ventana de la figura C -07), el paquete instala el programa en la máquina y termina el proceso.

Una vez terminado el proceso anterior, se cuenta con los elementos necesarios en la máquina, para poder explorar y visualizar la herramienta, es entonces cuando se procede a instalar la herramienta en la máquina servidor.

La instalación de la herramienta, es muy sencilla, solo basta con copiar el directorio seepint, en el directorio htdocs que se encuentra en la ruta C:/Apache/htdocs, la letra C de la ruta, no es absoluta, esta puede cambiar dependiendo la unidad donde se encuentre instalado el servidor web.

Una vez que se ha almacenado la herramienta, en la máquina servidor, la herramienta se encuentra lista para visualizarla desde cualquier cliente, para tener éxito en la misión, se requiere repetir el proceso de instalación del Plu- In de Java 1.3.1 en cada una de las máquinas cliente<sup>25</sup>, de donde se desea explorar la herramienta, además iniciar el servidor Apache que está en la máquina servidor e iniciar el browser en el cliente y por medio de la dirección IP del servidor se explora la herramienta.

<sup>25</sup> El cliente, deberá tener los permisos adecuados y conocer la dirección IP para poder acceder a la máquina servidor.

Para obtener el IP de la máquina servidor, se debe buscar, en las ventanas que aparecen al examinar propiedades de TCP/IP en la ventana de configuración de Red de Windows.

## ANEXO D. MANUAL DE USUARIO

PARA FACILITAR EL ACCESO, A TODOS LOS RECURSOS QUE OFRECE LA HERRAMIENTA, SE RECOMIENDA UTILIZAR EL SIGUIENTE ORDEN DE NAVEGACIÓN.

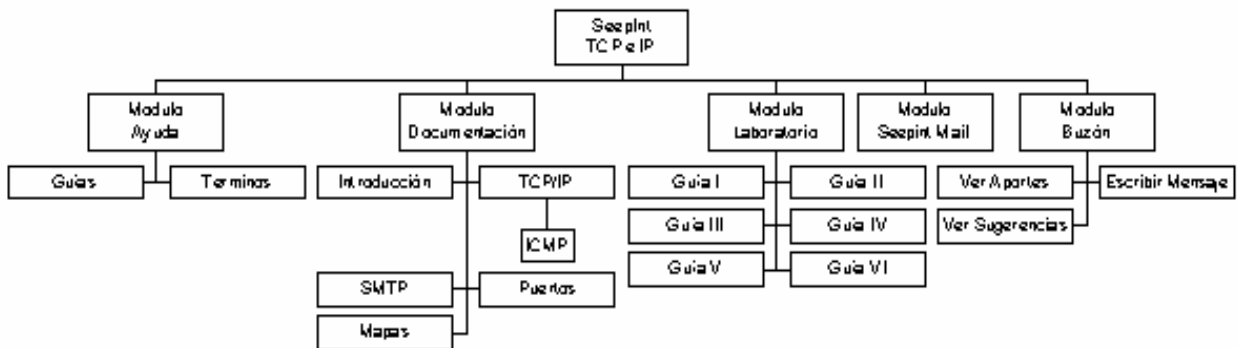
En la pantalla principal que se observa en la figura D – 01, se encuentra en la parte superior, el menú principal, que le permite al usuario acceder a cada uno de los módulos de la herramienta.

Figura D – 01. Pantalla principal de la herramienta.



Una vez que se está en el menú principal de la herramienta, el usuario puede tomar la decisión de ingresar a la herramienta, según las opciones que muestra el siguiente árbol de navegación que podemos apreciar en la figura D –02.

Figura D – 02. Árbol de despliegue de la Herramienta.



Conociendo el despliegue que puede tener la herramienta, describiremos cada uno de los módulos, a fin de detallar, la interacción completa entre usuario y herramienta.

#### ⊕ Modulo de Ayuda:

*Descripción:* Este modulo, como su nombre lo indica, ayuda al usuario, en dos aspectos, el primero, es que contiene un seguimiento por cada una de las guías que conforma el modulo laboratorio, de tal manera que el usuario, podrá enterarse en cualquier momento, que contiene cada una de las guías, y lo segundo, es que contiene la opción términos, que corresponde a un listado de términos que aparecen en la herramienta, con su correspondiente significado, de tal manera que si un usuario no entiende o recuerda la definición de un termino, se dirige a la ayuda y en la opción Términos, podrá encontrar el concepto o definición del termino

*Modo de Ingreso:* Para acceder a este modulo, basta con posicionarse con el ratón sobre la solapa de color naranja de nombre Ayuda, que se encuentra en el menú principal, ver figura D-01 parte superior y dar clic, desplegándose la ventana de ayuda.

#### ⊕ Modulo Documentación:

*Descripción:* Este modulo, contiene un grupo de documentos, que son esenciales en la contextualización de la temática de TCP e IP, de tal manera que el usuario puede leer los documentos en cualquier instante, de tal forma que pueda empaparse del tema o complementar.

*Modo de Ingreso:* Para acceder a este modulo, basta con posicionarse con el ratón sobre la solapa de color naranja de nombre Documentación, que se encuentra en el menú principal, ver figura D-01 parte superior y dar clic, para desplegar la pagina principal de documentación sobre el marco principal de la herramienta, para poder ingresar a cualquiera de los temas dando clic sobre el vinculo.

#### ⊕ Modulo Laboratorio:

*Descripción:* Este modulo, siendo el componente principal de la herramienta, es el mas extenso, ya que contiene todas las guías de laboratorio, con sus Applets correspondientes, de tal manera que lo detallaremos por guías.

*Modo de Ingreso:* Para acceder a este modulo, basta con posicionarse con el ratón sobre la solapa de color naranja de nombre Laboratorio, que se encuentra en el menú principal, ver figura D-01 parte superior y dar clic, para desplegar la

pagina principal de laboratorio sobre el marco principal de la herramienta, para poder ingresar a cualquiera de las guías dando clic sobre cada vinculo.

Al dar clic sobre el vinculo, Fragmentación y reensamblado, el usuario, podrá leer el contenido de la guía y al final conseguirá acceder al Applet Fragmentación y reensamblado, este Applet presenta dos vista, la primera ubicada en la primera solapa, contiene la representación dinámica de la fragmentación Interred. En la segunda solapa, encontrara una aplicación de nombre Calculadora de Fragmentos, que le permitirá calcular el número de fragmentos que una trama debe ser dividida para poder ser enviada por un canal de comunicación.

Al dar clic sobre el vinculo, Direccionamiento, el usuario, podrá leer el contenido de la guía y al final conseguirá acceder al Applet Direccionamiento, este Applet, permite al usuario, escribir una dirección IP cualquiera, conformada por cuatro campos, y una mascara de sub red, una vez digitada, el Applet valida la información digitada y arroja información correspondiente, además en la segunda solapa, de nombre información, el usuario podrá verificar los rangos de direcciones IP.

Al dar clic sobre el vinculo, Transferencia de Datos, el usuario, podrá leer el contenido de la guía y al final conseguirá acceder a los Applets Establecimiento de conexión, Transferir Datos y Cierre de conexión. Se recomienda observarlos, en el mismo orden que han sido mencionados, dado que el proceso en la realidad ocurre en este orden.

Al dar clic sobre el vinculo Encaminamiento de Datagramas IP, el usuario, podrá leer el contenido de la guía y al final conseguirá acceder a los Applets Algoritmo de Dijkstra, con el que el usuario podrá observar el mecanismo con el que el protocolo OSPF toma las decisiones de encaminamiento, además en el Applet encaminamiento, Protocolo OSPF, el usuario, podrá diseñar una topología de red en la solapa Topología, además insertar los costos que cuesta transferir un paquete desde el origen hasta el final, y observar los resultados en la tercera solapa de nombre Enrutamiento, que corresponden a las rutas que debe tomar un paquete para ser enrutado desde el origen hasta su destino.

Al dar clic sobre el vinculo Detección de errores de TCP, el usuario, podrá leer el contenido de la Guía y al final conseguirá acceder a los Applets Método de Paridad y Método Suma de Bloque, con los que el usuario podrá comparar la capacidad que cada método tiene para detectar errores.

Al dar clic sobre el vinculo Control de Errores, el usuario, podrá leer el contenido de la Guía y al final conseguirá acceder a los Applets que le permitirán observar los métodos de control de errores de Repetición Selectiva y Retroceder N.

Al dar clic sobre el vinculo Control de Flujo de TCP, el usuario, podrá leer el contenido de la Guía y al final conseguirá acceder a los Applets que le permitirán, observar el mecanismo de ventana deslizante, y luego, comprobar, como el tamaño elegido para la ventana deslizante, puede influir en el nivel de desempeño de un canal de comunicación.

#### ⊕ Modulo SEEPIN MAIL

*Descripción:* Este modulo, tiene como misión, mostrar al usuario, que las aplicaciones de Internet, en este caso particular el correo electrónico, son las que requieren en algún momento los servicios de los protocolos TCP e IP y de sus procesos correspondientes.

*Modo de Ingreso:* Para acceder a este modulo, basta con posicionarse con el ratón sobre la solapa de color naranja de nombre SEEPINT MAIL, que se encuentra en el menú principal, ver figura D-01 parte superior y dar clic, luego, seguir la secuencia que van indicando los Applets.

#### ⊕ Modulo BUZÓN

*Descripción:* Este modulo, tiene como función particular, recolectar o almacenar información proveniente de los usuarios, para que posiblemente en futuras versiones de la herramienta, estas sean tenidas en cuenta y puedan ayudar a mejorar la herramienta.

*Modo de Ingreso:* : Para acceder a este modulo, basta con posicionarse con el ratón sobre la solapa de color naranja de nombre BUZÓN, que se encuentra en el menú principal, ver figura D-01 parte superior y dar clic, luego, el usuario, deberá llenar los campos que solicita la herramienta como son el nombre, el correo electrónico, el perfil y el contenido del mensaje.

Descripción de Campos.

*Perfil:* El perfil, es una descripción corta de lo que es el usuario.

*Aporte:* Un aporte, es una idea puntual, y que muestra la forma en que una falencia debe ser corregida.

*Sugerencia:* Una sugerencia, debe ser una idea global sobre algo que no esta cumpliendo con las expectativas del usuario.