PONTIFICIA UNIVERSIDAD CATOLICA DEL ECUADOR

SEDE AMBATO

ESCUELA DE INGENIERIA DE SISTEMAS

DISERTACIÓN DE GRADO PREVIA LA OBTENCIÓN DEL TITULO DE INGENIERIA DE SISTEMAS

"Creación de un WebMail para la PUCESA"

Ruth Eugenia Cruz Paz

Verónica del Carmen Jerez Chachapoya

DIRECTOR DE LA DISERTACIÓN: Ing. Janio Jadán

PONTIFICIA UNIVERSIDAD CATOLICA DEL ECUADOR PROGRAMAS ACADEMICOS AMBATO
BIBLIOTECA

PONTIFICIA UNIVERSIDAD CATOLICA DEL ECUADOR
AMBATO
INGENIERIA DE SISTEMAS
SECRETARIA

AMBATO, 2003.

PONTIFICIA UNIVERSIDAD CATOLICA DEL ECUADOR

SEDE AMBATO

ESCUELA DE INGENIERIA DE SISTEMAS

DISERTACIÓN DE GRADO PREVIA LA OBTENCIÓN DEL TITULO DE INGENIERIA DE SISTEMAS

"Creación de un WebMail para la PUCESA"

DIRECTOR: _____

Ing. Janio Jadán.

Ruth Eugenia Cruz Paz

Verónica del Carmen Jerez Chachapoya

AMBATO, 2003.

DEDICATORIA

El presente trabajo está dedicado a Dios por guiar mis pasos con sabiduría e inteligencia.

De una manera muy especial a la memoria de mi madre que con su sacrificio y abnegación se convirtió en mi ejemplo de superación, admiración, y la perfecta entrega de amor; se que hubiera compartido este momento con orgullo y alegría pero a pesar de su ausencia siempre estará presente.

Ami tía Agélica, quién supo motivarme moralmente, confió en mí y me apoyó para que una meta más en mi vida haya culminado.

Ruth.

AGRADECIMIENTO

Agradezco a la Pontificia Universidad Católica del Ecuador Sede Ambato, por haberme acogido en sus aulas y formarme tanto en el campo científico como en el aspecto humano.

A todos los maestros que a lo largo de esta carrera supieron impartir sus conocimientos de la mejor manera.

De manera especial al Ing. Janio Jadan director de esta tesis, por haberme guiado con paciencia y esmero para la elaboración de la misma, más que un maestro ha sido un amigo, por lo que se a ganado mi respeto y admiración

Ruth.

AGRADECIMIENTO

Agradezco de manera especial a mi director de tesis
Ing. Janio Jadán
que más allá de ser un profesor ha sido un gran amigo, brindándome sus conocimientos y a la vez guiándome para salir adelante, ganándose mi admiración y respeto por todo eso y mucho mas. Mil Gracias.

A mi compañera y amiga Ruth que es como mi hermana quién me apoyo en los momentos que más necesite.

De igual manera agradezco a todos los profesores que de una u otra manera me apoyaron en esta tesis.

Verónica.

INDICE

INDICE

CAPITULO I 1
1. ASPECTOS GENERALES 1
1.1. INTRODUCCIÓN 1
1.2. ANTECEDENTES Y JUSTIFICACIÓN 2
1.3. DESCRIPCIÓN DEL PROYECTO 3
1.4. ESTADO ACTUAL DEL CENTRO DE COMPUTO DE LA PUCESA 4
1.4.1. POLÍTICAS DE CONTROL 5
1.4.1.1. POLÍTICAS DE INGRESO AL LABORATORIO 5
1.4.1.2. MONITOREO Y SANCIONES 6
1.4.2. SERVICIOS DEL CENTRO DE CÓMPUTO 7
1.4.3. SOFTWARE EXISTENTE EN EL CETRO DE CÓMPUTO 7
1.4.4. HARDWARE EXISTENTE EN EL CENTRO DE CÓMPUTO DE LA PUCESA 9
1.4.4.1. CENTRO DE CÓMPUTO 9
1.4.4.2. LABORATORIO IBM 11
1.4.4.3. LABORATORIO COMPAQ 12
1.5. ORGANIZACIÓN DEL DOCUMENTO 14
CAPITULO II 15
2. PLATAFORMA LINUX 15
2.1.INTRODUCCIÓN 15
2.2. ESTUDIO DEL SISTEMA OPERATIVO LINUX 15
2.3. HISTORIA 17
2.4. LAS DISTRIBUCIONES DE LINUX 19
2.5. REQUERIMIENTOS PARA LA INSTALACIÓN 20
2.6. INSTALACIÓN Y CONFIGURACIÓN 21
2.6.1. USO DE FIPS PARA REALIZAR PARTICIONES 22
2.6.2. USO DE RESTORRB 23
2.6.3. LIMITACIONES DE FIPS 24
2.6.4. INSTALACIÓN DE LINUX 25
2.7. CARACTERÍSTICAS Y FUNCIONAMIENTO 26
2.7.1. LINUX Y SUS SHELLS 26

2.7.2. LINUX ES MULTITAREA 27
2.7.3. LINUX ES MULTIUSUARIO 27
2.7.4. LINUX ES SEGURO 28
2.7.5. LINUX Y SU CONTROL DE DISPOSITIVOS 28
2.7.6. LINUX Y LAS REDES DE ORDENADORES 29
2.8. VENTAJAS DE LINUX 29
2.9. DESVENTAJAS DE LINUX 30
CAPITULO III 32
3. INTERNET 32
3.1. INTRODUCCIÓN 32
3.1.2. CONCEPTO 33
3.1.3. ESTRUCTURA CLIENTE/SERVIDOR 33
3.1.4. VENTAJAS Y DESVENTAJAS DEL USO DE INTERNET 34
3.2. INTRANET 35
3.2.1. CONCEPTO 35
3.2.2. VENTAJAS DE UNA INTRANET 36
3.2.3. CARACTERÍSTICAS DE LA INTRANET 36
3.3. EXTRANET 37
3.3.1. CONCEPTO 37
3.3.2. CARACTERÍSTICAS 38
3.4. PROTOCOLOS DE COMUNICACIÓN 38
3.4.1. FTP (FILE TRANSFER PROTOCOL) 39
3.4.2. HTTP (HYPER TEXT TRANSFER PROTOCOL 40
3.4.3. NFS (NETWORK FILE SYSTEM) 41
3.4.4. POP3 POST OFFICE PROTOCOL VERSION 3 42
3.4.5. TCP/IP TRANSFER COMMUNICATION PROTOCOL / INTERNET PROTOCOL 43
3.4.5.1. PRINCIPALES CARACTERÍSTICAS DE TCP 44
3.4.5.2. NIVELES DE LA ARQUITECTURA DEL TCP/IP 47
3.5. COMUNICACIONES EN RED 48
3.5.1. SOCKETS STREAM *(TCP)* 48
3.5.2. SOCKETS DATAGRAMA *(UDP)* 49

PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR
PROGRAMAS ACADÉMICOS AMBATO
BIBLIOTECA

3.5.3. SOCKETS RAW 49
3.6. SEGURIDADES 51
3.6.1 AUTENTIFICACIÓN 52
3.6.2. ENCRIPTACIÓN 53
3.7. VIRUS EN LA WEB 55
3.7.1. DAÑOS DE LOS VIRUS 56
CAPITULO IV 58
4. SISTEMA DE CORREO ELECTRONICO WEBMAIL 58
4.1. INTRODUCCIÓN 58
4.2. BENEFICIOS DE WEBMAIL 58
4.3. INTERFAZ DE UN WEBMAIL EN EL LADO CLIENTE 59
4.3.1. ACCESO A LA CUENTA 59
4.3.2. INTRODUCCIÓN A LA PÁGINA PRINCIPAL 60
4.3.3.ESTABLECIMIENTO DE PREFERENCIAS 61
4.3.4. LÍMITE DEL TIEMPO DE CONEXIÓN 62
4.3.5.CIERRE DE LA SESIÓN DE WEBMAIL 62
4.3.6. IDENTIFICACIÓN DE USUARIOS 62
4.4. ESTRUCTURA DE UN WEBMAIL EN EL LADO SERVIDOR 63
4.4.1. INTRODUCCIÓN 63
4.4.2.SENDMAIL 64
4.4.2.1. CARACTERÍSTICAS DE SENDMAIL 64
4.4.2.2. FUNCIONES DE SENDMAIL 67
CAPITULO V 69
5. ESTUDIO DE LOS LENGUAJES DE PROGRAMACIÓN PARA EL DESARROLLO DEL WEBMAIL 69
5.1. HTML (HIPERTEXT MARKUP LANGUAGE) 69
5.1.1. QUÉ ES EL HTML 69
5.2. JSP (JAVA SERVLET PAGE) 70

5.2.1. QUÉ ES JAVA 70
5.3. CGI (COMMON GATEWAY INTERFACE) 71
5.3.1. ESPECIFICACIONES 72
5.3.2. DIRECTORIO CGI-BIN 73
5.4. INTRODUCCIÓN A PHP (HYPERTEXT PREPROCESSOR) 74
5.4.1. CONCEPTO 74
5.4.2. UNA BREVE HISTORIA DE PHP 75
5.4.3. VENTAJAS DE PHP 76
5.4.4 SEGURIDAD EN PHP 80
5.4.4.1. COOKIES 81
5.4.4.2. POSIBLES ATAQUES 83
5.4.4.3. AUTENTICACIÓN 84
5.5. LENGUAJE DE PROGRAMACIÓN EN PHP 86
5.5.1.VARIABLES 86
5.5.2. OBJETOS 87
5.5.3. CONTROL DE FLUJO 88
5.5.4. FUNCIONES 90
5.5.5. VARIABLES DE LA WEB 91
5.5.6. INCLUIR FICHEROS 91
5.5.6.1. OBTENER LA DIRECCIÓN IP Y EL NAVEGADOR USADO POR EL VISITANTE 92
5.5.6.2. ENVIAR UN E-MAIL 93
5.6. CONEXIÓN A BASES DE DATOS 93
CAPITULO VI 98
6. DESARROLLO E IMPLENTACION DEL WEBMAIL EN LA PUCESA 98
6.1. DESCRIPCIÓN TOTAL DEL PROYECTO 98
6.2. METODOLOGÍA PARA LA ELABORACIÓN DEL SISTEMA 98
6.3. PROCESO DE PRE-DESARROLLO 100
6.4. ESCOGITAMIENTO DE LAS HERRAMIENTAS DE DESARROLLO 103
6.5. PROCESO DE DESARROLLO 107
6.5.1. FLUJO DE LA INFORMACIÓN 107
6.5.2. REQUERIMIENTOS DEL SISTEMA 109

INTRODUCCIÓN

Con el avance de la tecnología la manera de comunicarse ha ido cambiando, es así que hace un tiempo atrás, para comunicarse con otros países era un poco problemático, sin embargo hoy tenemos a nuestro alcance el Internet, que es la red más grande de computadoras, por medio de la cual nos podemos comunicar de una forma inmediata y a bajo costo.

El presente trabajo esta orientado a la elaboración de un WebMail para la Pontificia Universidad Católica del Ecuador Sede Ambato, el proyecto surgió por la necesidad de que quienes conforman esta institución puedan, leer, escribir y enviar e-mails(correo electrónico), ya sea dentro o fuera de la Universidad, en una forma más rápida y desde un ámbito más confiable y amigable.

La Universidad cuenta con la Plataforma Linux, que poco a poco está siendo más conocida, es por eso que el presente trabajo es desarrollado en este sistema operativo. Linux está siendo usado por cientos de miles de personas alrededor del mundo, está disponible para quién lo desee utilizar; es un sistema operativo de código abierto, es decir que se distribuye de forma gratuita bajo una licencia pública general de GNU (Organización de Software Libre (Free Software)), esta licencia tiene por objeto asegurar que Linux siga siendo gratuito y a la vez estándar. Al instalar el sistema operativo, también se tiene la posibilidad de instalar varios programas, tales como hojas de cálculo, bases de datos, procesadores de texto, varios lenguajes de programación, paquetes de telecomunicaciones y juegos.

Para que los programas se comuniquen en Internet se usan los Protocolos de comunicación, que proporcionan un vehículo de entrega para las imágenes, gráficos, video, hipertexto u otros datos en la Web. El objetivo principal de un protocolo de comunicación es promover la forma de compartir archivos entre computadoras

(programas y datos), alentar al uso remoto de las computadoras, y transferir datos de una forma segura y optima por computadora.

EL WebMail es un conjunto de scripts (pequeños programas) a través del cual se puede acceder al correo electrónico directamente desde cualquier navegador (Internet Explorer o Netscape). Esto es de gran utilidad, puesto que no importa en qué lugar un usuario se encuentre, podrá acceder a leer y responder sus mensajes de correo electrónico con total facilidad. Algunas de las muchas características del WebMail comprenden: lista de contactos, programación diaria, corrector ortográfico y marcadores web.

Se puede tener acceso a WebMail desde cualquier lugar con acceso a Internet y un explorador web. Muchos Webmails poseen antivirus para analizar mensajes de correo entrantes. Un Webmail típico aparece como una página web normal y amigable al usuario final o cliente, transparente a la tecnología, procesos y plataforma que residen en la estructura del webmail en el servidor.

Para que exista un envío de datos el proyecto usa Sendmail que es un conocido agente de transferencia de correo MTA (Mail Transport Agent) usado en Internet que maneja un amplio porcentaje de todos los correos ruteados en Internet. El deber principal de Sendmail, como otros MTAs, es el de trasladar con seguridad e-mails entre hosts, habitualmente usando protocolos de comunicación. No obstante, Sendmail es altamente configurable, potente, estable y compatible con los estándares de Internet. Sendmail es el agente de transporte de correo más común de Internet (en los sistemas UNIX).

Para la realización de este trabajo se utilizó el paquete PHP4, acrónimo de "PHP: Hypertext Preprocessor", es un lenguaje interpretado de alto nivel embebido en páginas HTML. PHP es un lenguaje de script similar al HTML(Lenguaje de marcación de Hipertexto), multiplataforma y que se ejecuta del lado del servidor, por eso nos permite acceder a los recursos que tenga el servidor como por ejemplo podría ser una base de datos. El programa PHP es ejecutado en el servidor y el resultado enviado al navegador. El resultado es normalmente una página HTML, el lenguaje PHP es un lenguaje de programación de estilo clásico, esto quiere decir que es un lenguaje de programación con variables, sentencias condicionales, bucles, y funciones.

Al ser PHP un lenguaje que se ejecuta en el servidor no es necesario que el navegador lo soporte, es independiente del navegador, pero sin embargo para que las páginas PHP funcionen, el servidor donde están alojadas debe soportar PHP.

Actualmente en Internet existe un sinnúmero de WebMails desde los gratuitos que son utilizados para que muchas personas visiten estos sitios e indirectamente se empapen de publicidad, hasta el personal de una Empresa u organización.

La creación de un **WebMail** para la Pontificia Universidad Católica Sede-Ambato (PUCESA) surge de la necesidad de contar con un sistema que permita a los estudiantes, profesores y personal administrativo poder rescatar, enviar y redactar correos ya sea dentro o fuera de la Institución, de una forma más natural y amigable.

Un Webmail puede ser desarrollado en diferentes lenguajes de programación y para diferentes plataformas. Una de las plataformas que poco a poco va ganando terreno en el campo del Internet es Linux y al tener la PUCESA un servidor de correo en esta plataforma, este proyecto pretende desarrollar un Webmail bajo este sistema operativo. Además porque Linux es un sistema operativo con una serie de características que lo hacen totalmente diferente a los más comunes como MS-DOS o Windows95.

Con el Desarrollo e implantación de un WebMail propio en la PUCESA se podría reducir costos, ya que como es conocido existen empresas que desarrollan este tipo de aplicaciones con costos de licencias de uso. Además quedaría abierta la posibilidad de crear un WebMail gratuito para el público en general, con el fin de que se pueda aprovechar espacios publicitarios que generen ingresos económicos adicionales a la Universidad.

1.2. Antecedentes y Justificación.

Sabiendo que un WebMail es un conjunto de scripts (pequeños programas) a través del cual se puede acceder a un correo electrónico directamente desde cualquier navegador (Internet Explorer, Netscape o Linux). Esto es de gran utilidad puesto que no importa; en qué lugar un estudiante o personal docente se encuentre, podrá acceder a leer y responder sus mensajes de correo electrónico con total facilidad.

La aplicación del WebMail estará a disposición de los estudiantes que necesiten comunicarse de manera inmediata; ya sea enviando trabajos referentes a sus estudios o con personas particulares sin costo alguno y en menor tiempo y desde cualquier estación de trabajo por medio de la Web (browser) independientemente de la plataforma.

La meta del WebMail es permitir el mejoramiento de la circulación de la información y la comunicación entre los estudiantes dentro y fuera de Universidad. Además se pretende a diferencia de WebMail tradicionales, complementar el WebMail de la PUCESA con reportes estadísticos y poder conocer cuentas de correo con demasiado volumen de información, frecuencia de uso y otros parámetros que podrían nacer de la investigación. Una característica importante con la que va a contar nuestro Web Mail también será el análisis de los correos antes de ponerlos en el WebMail para determinar si tiene VIRUS, con el fin de no contaminar la red de la PUCESA.

1.4. Estado Actual del Centro de Computo de la PUCESA.

El centro de Cómputo de una Institución es una pieza fundamental de soporte, para hacer que determinada institución llegue a cumplir sus objetivos. En la PUCESA, la misión del Centro de Computo es la de fortalecer y mantener los servicios de computación para satisfacer las necesidades de información y equipamiento de clientes internos y convertirse al mismo tiempo en un ente de apoyo a la docencia. Esta misión se ha venido realizando con cabalidad, ya que continuamente se están estructurando políticas de control, aplicando nuevos servicios, software y mantenimiento a los equipos con los que cuenta.

A continuación se describirá el estado actual del centro de cómputo en los aspectos antes mencionados.

1.4.1. Políticas de Control.

Actualmente se están aplicando nuevas políticas para el ingreso a los laboratorios y se está realizando un monitoreo de las actividades de los estudiantes en el computador con el fin de que los recursos sean aprovechados en el aprendizaje de temas referentes al área.

1.4.1.1. Políticas de Ingreso al Laboratorio.

Existen dos tipos de políticas, la primera cuando los estudiantes tienen clases, estas son:

- Primero debe registrarse el Docente que recibe el Laboratorio.

- Luego de registrarse el Docente, ingresan los alumnos que serán totalmente responsables de la computadora que utilicen.

Con lo anterior queda totalmente prohibido el ingreso a los laboratorios si no se encuentra el Docente.

Por otro lado cuando los laboratorios no están siendo utilizados en clases, se los usa para alquilarlos con las siguientes políticas:

- Los alumnos deben registrarse en la oficina del centro de computo y serán totalmente responsables del equipo mientras el carnet permanezca en ésta dependencia.

- Los alumnos tienen la obligación de mantener el aseo, el orden y el respeto en el interior de los laboratorios.

- Durante el uso de los laboratorios los alumnos están sujetos a las obligaciones y sanciones que contempla el reglamento interno.

1.4.1.2. Monitoreo y Sanciones.

Por antecedentes registrados en la vida del centro de Cómputo, hoy se está realizando un monitoreo de las estaciones de trabajo para garantizar que el personal y estudiantado de la PUCESA siga con los fines que se quiere lograr.

Las politicas aplicadas en el monitoreo son las siguientes:

Desde el 1 de Junio del 2001 todas las computadoras de los laboratorios son Monitoreadas desde la oficina del centro de Cómputo.

Se prohibe el ingreso a páginas de contenido ofensivo y que pueda saturar el canal de Internet.

Se prohibe también la descarga de archivos de más de 2 Megabytes (download), archivos midi o mp3.

Se permite el acceso a CHAT ROOMS siempre y cuando el contenido sea eminentemente científico- técnico (Foros).

El uso del ESCÁNER deberá ser únicamente con fines académicos, no personales.

Obviamente si no se cumplen las politicas antes mencionadas se podrían tener las siguientes sanciones:

1. Primera falta, llamada de atención Verbal o Escrita (mensaje en el terminal).

2. Segunda falta, llamada de atención verbal o Escrita (mensaje en el terminal).

3. Tercera falta, prohibición de uso de laboratorio durante 5 días.

Se mantendrá un registro histórico de las sanciones por cada estudiante que podrá ser adjuntada a su hoja de vida estudiantil.

1.4.2. Servicios del Centro de Cómputo.

El trabajo continuo y conjunto del personal encargado del centro de cómputo, estudiantado, extensionistas y practicantes han permitido que actualmente la PUCESA por medio de su centro de cómputo brinde un sin número de servicios, entre los principales se mencionan:

1. Laboratorio IBM con 10Pcs.
2. Laboratorio Compaq con 6Pcs.
3. Servicio de Hosting para páginas web.
4. Proveedor Servicios Internet.
5. Correo electrónico para estudiantes, personal administrativo y docente.
6. Biblioteca Virtual.
7. Mantenimiento de los Equipos.

1.4.3. Software existente en el Cetro de Cómputo.

La Universidad Católica como una Institución que promulga las buenas costumbres y ética moral, mantiene en orden y de acuerdo a la ley licencias de los siguientes productos informáticos mostrados en la **tabla 1.1.**

Categoría	Paquete	Versión	Fabricante	Costo	Cantidad Licencias
Sistemas Operativos	Windows NT	4.0	Microsoft	$ 350	1
	Linux Red hat	6.2	GNU	Gratuito	1
	Sun Solaris	2.6	Sun Micro system	$100	1
	Windows 98	98	Microsoft	$130	13
Herramientas de Desarrollo	Oracle	8.1	Oracle Corporation	Donación	1
	Power Builder	5.0	Power Soft	$3000	
	Microsoft Office 97	97	Microsoft	$500	
		97	Microsoft	$650	10
	Visual Studio	6.0	Microsoft	$500	1
Bases de Datos	SQL Server	7.0	Microsoft	$400	10
	DB2	8.0	IBM	Donación	1
Gráficos	Autocad	2000	Autocad		
	Flash	4.0	Macromedia	30 días	1
	Fireworks	3.0	Macromedia	30 días	1
Diseño de Páginas Web	Dream Weaver	4.0			
	Font Page	98	Microsoft	$200	1
Lenguajes de Programación	Pascal para Windows	1.5	Microsoft		1
		6.0	Microsoft	$200	Académica
	Visual C++	6.0	Microsoft	$120	1
	Visual J++				10
Browser	Internet Explorer	5.0	Microsoft	Gratuito	

Tabla 1.1: Software existente en el Centro de Cómputo de la PUCESA

1.4.4. Hardware Existente en el Centro de Cómputo de la PUCESA.

Últimamente el Centro de Cómputo no adquirido mucho equipo informático, pero ha podido desempeñar bien sus funciones con los recursos de hardware con los que cuenta. A continuación se describirá el Hardware del Centro de Cómputo y la disposición de su red.

1.4.4.1. Centro de Cómputo.

El centro de Cómputo de la PUCESA es usado para todas las labores que realiza el encargado del mismo, entre ellas monitorear a los laboratorios de la Universidad, aquí se encuentra el servidor de la red, usado para las conexiones a Internet entre otras labores, los equipos existentes son los siguientes como se los detalla en la **Tabla1.2.**

Equipo	Marca	Características
Computador	SUN	SUN Microsystem 250 e Procesador Ultra Spark 400 Mhz, 256 RAM, CD-ROM, 180 Gb de DDH, Tape backup todo tecnología SCSI.
Computador	CLON	Clon AMD 1.26, 256 RAM, disco 306 1Fdd, 1 CD- ROM, video sonido, fax integrado, teclado mouse tipo PS2, tarjeta de red, 2 puertos Usb.
SWITCH	Centre Com	3726XL, 10 BaseT, 1 puerto Fast Ethernet.
HUB	Centre Com	3624 TRS, 24 Puertos, 10 Base T

Tabla1.2: Equipos existentes en el Centro de Computo de la Universidad.

En la **Fig.# 1.1** se indica la distribución de los equipos en el Centro de Cómputo de la Universidad.

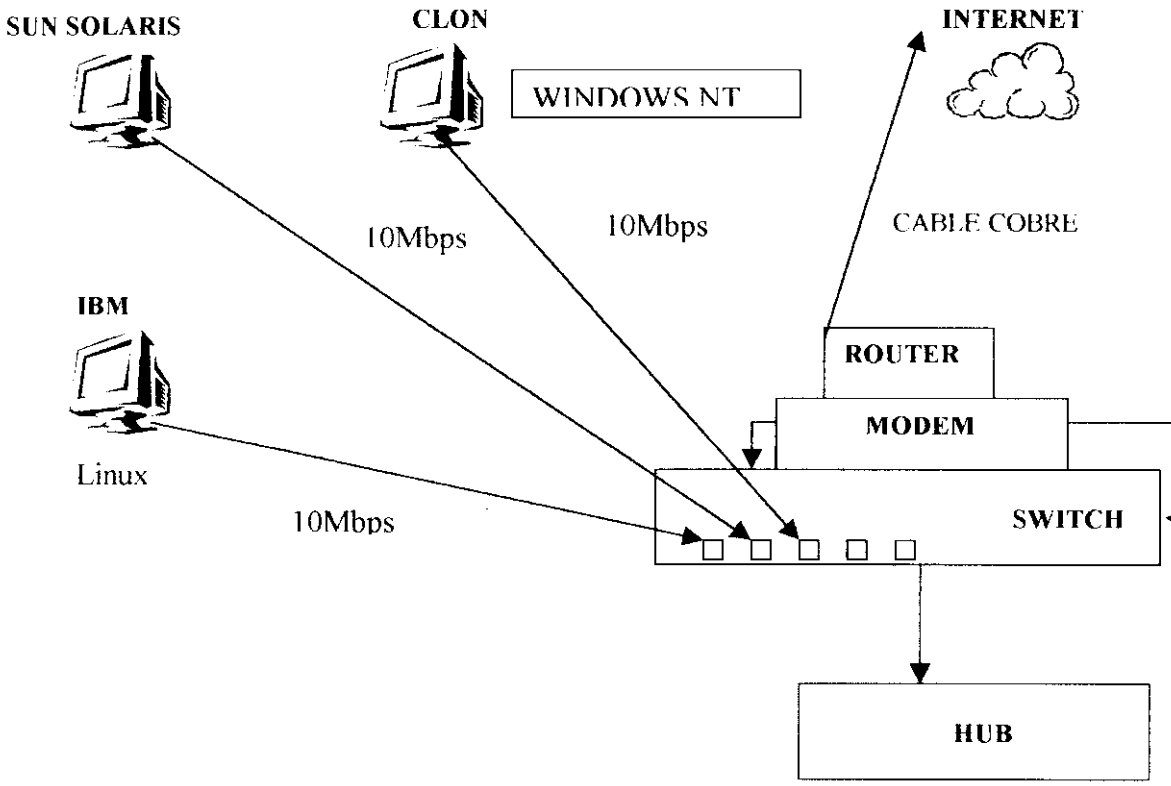


Fig.# 1.1: Distribución de los equipos en el centro de Cómputo.

1.4.4.2. Laboratorio IBM.

El laboratorio IBM es usado por los estudiantes y profesores para recibir y dictar clases, también se puede hacer uso de Internet, en horas de clases o fuera de ellas, previa la autorización correspondiente. Los equipos existentes se los indica en la **Tabla 1.3.**

Equipo	Marca	Características
Computador	IBM	Intel Celeron 200Mhz, 32 RAM, Dims 72 pines, Disco duro Quantum 2.16 Gb, video, sonido incorporado,mainboard Intel, CD-ROM, floppy.
HUB	Centre Com	3624 TRS, 24 Puertos, 10 Base T

Tabla 1.3: Equipos del Laboratorio IBM.

En la **Fig.# 1.2** se indica la distribución de los equipos del Laboratorios IBM de la Universidad, todos los equipos trabajan con el sistema operativo Windows 98 a excepción del equipo IBM 02, que utiliza Windows NT.

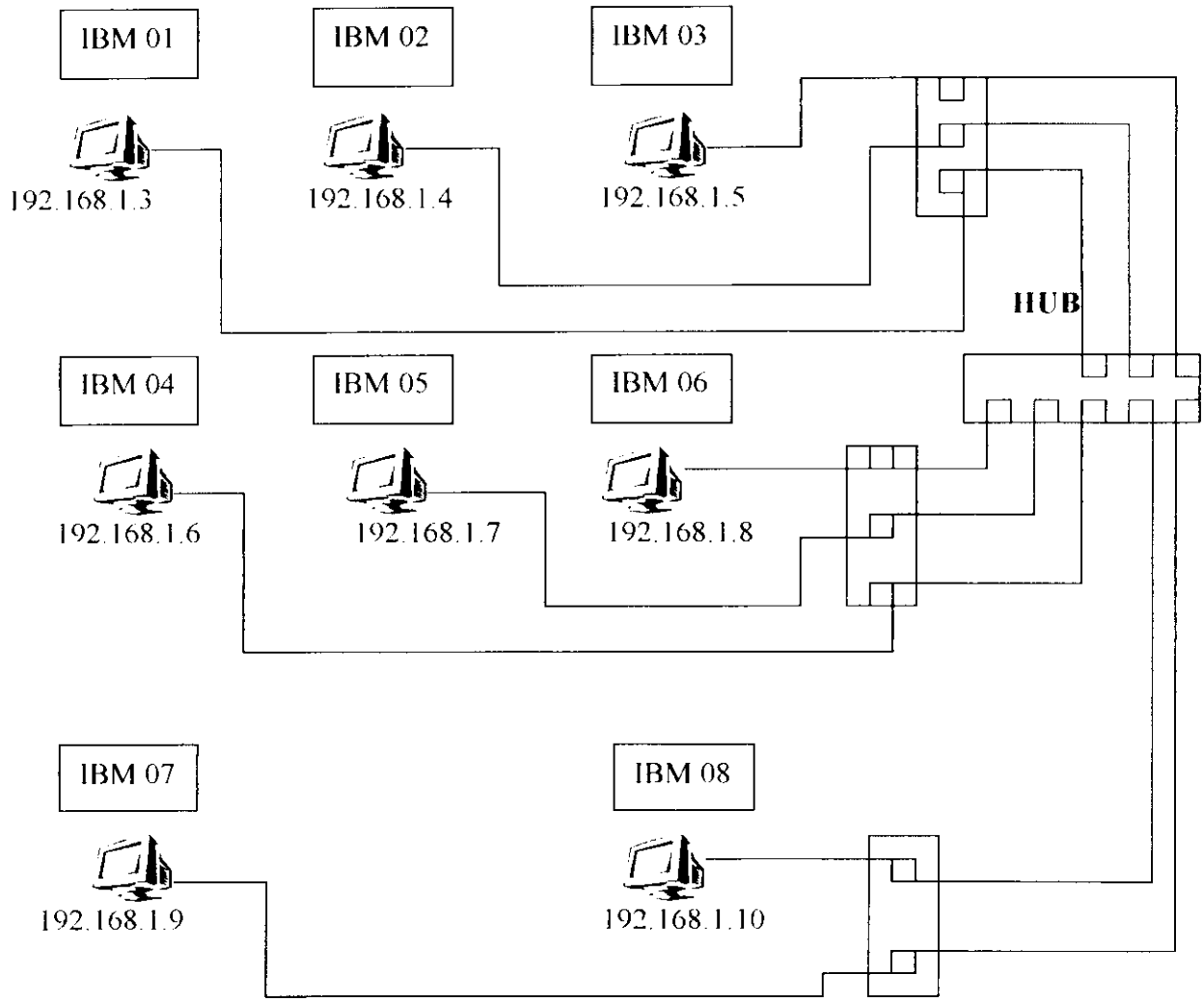


Fig.# 1.2: Distribución de los equipos del Laboratorio IBM.

1.4.4.3. Laboratorio Compaq.

El Laboratorio Compaq tiene un uso similar al Laboratorio IBM, en la **Tabla 1.4.** se detalla los equipos existentes con sus respectivas características.

Equipo	Marca	Características
Computador	Compaq	Procesadores Intel Pentium I, 16 RAM, video Sonido Incorporado, CD-ROM 4x, Disco duros de 1.2 Gb, tarjeta de red de 10 Mpbs, drive, monitor, teclado.
HUB	Centre Com	3624 TRS, 24 Puertos, 10 Base T

Tabla 1.4: Equipos del Laboratorio Compaq.

En la **Fig.# 1.3** se indica la distribución de los equipos del Laboratorio COMPAQ de la Universidad, cabe indicar que estos equipos funcionan con sistema operativo Windows 98.

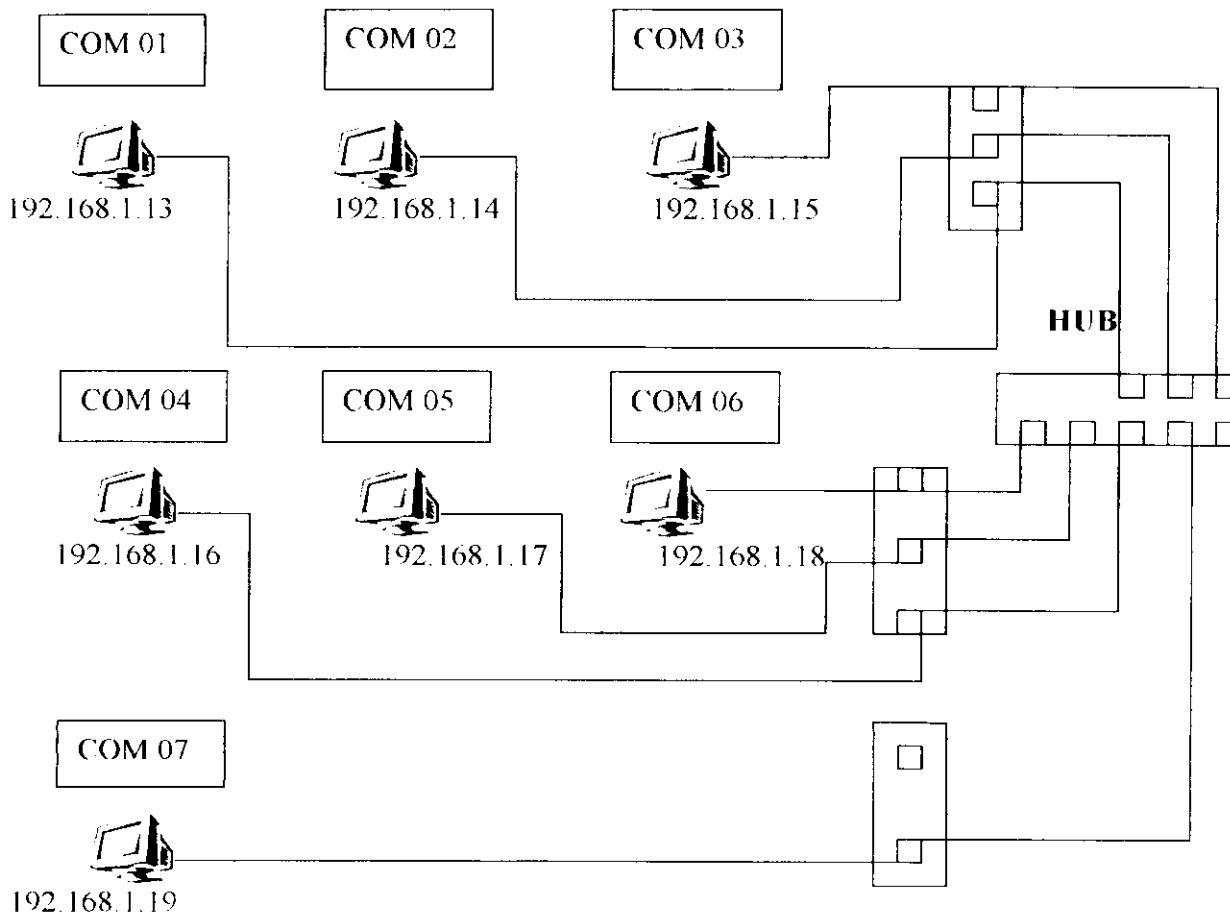


Fig.# 1.3: Distribución de los equipos del Laboratorio COMPAQ.

1.5. Organización del documento.

En él capitulo I se describió brevemente sobre la presente investigación, y los aspectos generales del laboratorio de la PUCESA como la distribución de los equipos, y características de los mismos; puesto que este proyecto quedará al servicio de la Universidad.

En el Capitulo II se describe la plataforma Linux que será utilizado en nuestro proyecto comenzando desde su historia, versiones existentes, requerimientos y formas de instalación.

En el Capitulo III se hablará sobre los conceptos generales de Internet, Intranet, los protocolos de comunicación en la red, y las seguridades en la red.

En el Capitulo IV se hablará de la característica de un WebMail en detalle, el Correo Electrónico, Comandos Mail, Ventajas y desventajas sobre lo mencionado y se analizará WebMails existentes en el mercado.

El Capítulo V se realizará un estudio de los lenguajes de programación para el desarrollo del WebMail, entre ellos HTML, JAVA, JAVA SCRIPT, PHP3.

El Capítulo VI dentro de este proyecto será el más importante puesto que se hablará sobre el desarrollo y la implementación del WebMail realizado en Linux, el estudio del proyecto en sus diferentes etapas, la configuración y por último la Instalación del mismo para uso de nuestra Universidad

CAPITULO II

2. PLATAFORMA LINUX.

2.1. Introducción.

Linux es un sistema operativo con una serie de características que lo hacen totalmente diferente a los más usados como MS-DOS o Windows. La desventaja de estos es que están realizados con tecnología propietaria, dicho de otra forma no existe posibilidad de realizar modificaciones ni de saber como se realizó dicho sistema.

Pero esto no es una dificultad para aquellos que no les importe sobre qué plataforma trabajan pero una gran frustración para los inquietos programadores, que desean conocer de cerca la Arquitectura del Sistema Operativo, y esto último es lo que permiten las plataformas Linux.

2.2. Estudio del Sistema Operativo Linux.

LINUX Es un Sistema Operativo para PCs que usan procesadores 386, 486 y Pentium, es un sistema operativo de código abierto, es decir que se distribuye de forma gratuita bajo una licencia pública general de GNU (Organización de Software Libre (Free Software)), poniéndolo a disposición de cualquiera que lo desee utilizar. Aún cuando Linux tenga registro de Copyright, y no sea estrictamente de dominio público. La licencia tiene por objeto asegurar que Linux siga siendo gratuito y a la vez estándar.

El hecho de que Linux sea gratis da a la gente a veces, la impresión equivocada que de algún modo es inferior a un sistema operativo profesional, pero al contrario este provee una plataforma sólida para uso en sus redes.

Linux es una implementación independiente de POSIX, la cual incluye multitarea real, conectividad de 32-bits, TCP/IP (Transfer Control Protocol / Internet Protocol), manejo de memoria, etc. Se puede usar como una estación de trabajo UNIX ó como servidor para tareas que abarcan desde servidor de Web altamente eficiente hasta estaciones de trabajo de bajo costo para redes cliente/servidor. La lista de características de Linux es impresionante. Sin embargo, los factores que aseguran el éxito a largo plazo de Linux tienen poco que ver con la lista de características, sino más bien con su licencia.

Linux es una versión de Unix libremente distribuible e independiente, para plataformas con maquinas x86, Motorola 68k, Digital Alpha, Sparc, Mips y Motorola Power PC. Es una implantación de la especificación POSIX con la cual cumplen todas las verdaderas versiones de Unix. El núcleo de Linux no usa código de AT&T o de cualquier otra fuente propietaria, la mayoría de los programas disponible para Linux es desarrollado por el proyecto GNU de la Free Software Foundation (Fundación de programas Libres).

Este soporta un amplio espectro de aplicaciones o paquetes de programación tales como X Window, Emacs, redes de datos bajo protocolos TCP/IP (incluyendo PPP(Protocolo de punto a punto (Point to Point Protocol)), ISDN(Integrate Services Digital Network)).

Mucha gente ha corrido benchmarks en sistemas Linux 80486 consiguiendo a este comparable con estaciones de rango-medio de Sun y Digital.

Linux está disponible en Internet en cientos de servidores FTP(File Transfer Protocol) y de varios distribuidores en discos flexibles o CD-Rom, alguno de estos son Caldera, Debian, Slakware, Red Hat, Suse, entre otros.

El núcleo de Linux esta legalmente protegido por la Licencia Publica GNU (GPL (License Publishes General)), es usualmente empaquetado con varios ejecutables que comprenden un sistema operativo Unix funcional. Estos paquetes de programas son llamados "distribuciones" y vienen en muchos tamaños y arreglos.

Linux esta siendo usado por cientos de miles de personas alrededor del mundo. Es usado para desarrollo de programas, redes (Intranet e Internet), y como plataforma para

usuarios finales. Linux se ha convertido en una alternativa efectiva en costo comparándolo con costosos sistemas Unix.

Linux no es de dominio publico, pero bajo la Licencia Publica de GNU, el código fuente para Linux debe siempre permanecer libremente disponible. La gente puede cobrar dinero por Linux si desea, en tanto que no intente limitar la redistribución de Linux.

Mucha gente alrededor del mundo ha trabajado en equipo para crear Linux, bajo los lineamientos de Linus Torvald, el autor original, y cada uno mantiene los derechos de autor del código que él o ella haya escrito.

2.3. Historia.

La historia de Linux esta muy ligada a la de uno de los sistemas operativos más populares: el UNIX; por eso podemos comenzar la historia con los primeros pasos de este sistema. Inicialmente (1950) los ordenadores eran ordenadores monousuarios. Unos años después (1960) aparecieron los sistemas de tiempo compartido. Gracias a él varios usuarios podían estar conectados al mismo tiempo a un mismo ordenador. En 1965 un conjunto de empresas se agrupó para crear un sistema operativo de gran potencia al que denominaron Multics pero este fracasó. La persistencia de uno de los programadores dio su fruto generando: un sistema operativo que inicialmente se le había denominado Unics en tono jocoso, UNIX había nacido.

Un artículo que describía el Sistema Operativo UNIX, atrajo la atención de las universidades que solicitaron el código fuente para estudiarlo y explicarlo en sus aulas. Pero surge un problema y es que en las Facultades no entienden el código fuente de UNIX por lo que el sistema operativo sigue considerándose complejo. Con este problema surge el MINIX que es un UNIX abreviado partiendo del Multics y ofreciendo las mismas llamadas al sistema que Unix, con un código inteligible para los estudiantes ya que su creador era un profesor universitario.

En 1990, un estudiante de 23 años de la Universidad de Helsinki, Linus Torvalds, comenzó a desarrollar, como hobby, un proyecto basado en el Minix. Quería llevar a cabo un sistema operativo que aprovechase la arquitectura de 32 bits para multitarea y eliminar las barreras del direccionamiento de memoria.

Torvalds empezó escribiendo el núcleo del proyecto en ensamblador, y luego comenzó a añadir código en C, lo cual incrementó la velocidad de desarrollo, e hizo que empezara a tomarse en serio su idea.

La primera versión no la dio a conocer porque ni siquiera tenía drivers de disquete, además de llevar un sistema de almacenamiento de archivos muy defectuoso.

Definitivamente saca una nueva versión consiguiendo así que el sistema Linux empezara a crearse, y con el añadido de otras versiones como FreeBSD conseguirían llegar un sistema operativo actual capaz de competir con los más innovadores de la época. El apoyo más grande, sin duda, ha sido Internet ya que a través de ella se ha podido demostrar que se puede crear un sistema operativo para todos los usuarios sin la necesidad de fines lucrativos.

Todo esto que hemos relatado anteriormente ocurrió, apenas, hace 8 años pero si contamos en tiempo informático estamos hablando de hace una eternidad. Básicamente podemos decir que hoy Linux es un sistema muy completo. El proyecto de Linus Torvalds aún no ha terminado, y se piensa que nunca se terminará por ésta continua evolución de la Informática.

El 5 de octubre de 1991, Linus anunció su primera versión "oficial" de Linux, versión 0.02. Desde entonces, muchos programadores han respondido a su llamada, y han ayudado a construir Linux como el sistema operativo completamente funcional que es hoy.

2.4. Las distribuciones de Linux.

Linux en si no estaba disponible de una forma sencilla de instalar. Ante esa necesidad aparecieron diversas distribuciones cada una de ellas con una determinadas características, ventajas y desventajas. Una distribución de Linux cuenta con una determinada cantidad de programas incluidos, una determinada forma de instalarse, una determinada forma de organizar los paquetes, entre otras cosas. Veamos las más conocidas:

- **Mini-Linux:** Está distribución instala una versión bastante reducida de Linux, aunque con X(Interfaz Gráfica). Se instala sobre la misma partición DOS del ordenador, lo que simplifica el proceso de instalación al máximo pero reduce la potencia y eficiencia del Linux. Es la distribución más sencilla pero la menos recomendable debido a que reduce la flexibilidad que el Linux es capaz de proporcionar.

- **Slakware:** Según algunos administradores esta es una de las distribuciones más completas. Se puede considerar como la distribución estancada de Linux.

- **Red-hat:** Esta alcanzado bastante popularidad últimamente debido a la gran cantidad de programas que contiene, en cambio tiene una extraña forma de organizar los directorios, de tal manera que resulta muy confusa lo que le quita puntos.

- **Debian:** Una distribución Linux sobria, sin florituras. Eso se hizo con el objetivo de hacer Linux lo más parecido a un Unix. Esta es una versión que inicialmente fue gestionada como un proyecto de la Free Software Foundation para evitar la comercialización de Linux, siendo la filosofía de éste la del dominio público, posteriormente la FSF (Free Software Foundation) abandonó el proyecto Debian, dejando a sus gestores solos, dedicándose exclusivamente al proyecto HURP (GNU Unix) el cual se supone que será un Unix capaz de ejecutar en cualquier plataforma y ejecutar cualquier binario de cualquier

plataforma a otra si necesitar compilación. Según los criterios adoptados por los gestores de Debian todos los paquetes que entren en esta distribución deben estar exentos de bugs (errores), en caso de que los tenga dispone de una dirección Internet para mandar bugs y dispone también de una dirección de servicio técnico en caso de problemas. Como se puede apreciar es un intento de convertir Linux en un sistema operativo totalmente formal.

2.5. Requerimientos para la Instalación.

En general para que el Linux pueda funcionar de una forma precisa como mínimo debe tener lo siguiente: al menos un computador con las siguientes características:

- 386 SX
- 1Mb RAM
- 28 Mb Disco Duro libres.
- Unidad de Floppy de 3" y1/2 de Alta densidad (1´44 Mb) o de 5" y 1/4 de (1'2 Mb)

Para que el Linux vaya a una velocidad y potencia aceptables se precisan las siguientes características:

- 486 DX
- 8 Mb RAM
- 50 Mb Disco Duro Libres.
- Unidad Floppy 3" y 1/2 de alta densidad
- Lector de CD-Rom de 8V

Si se desea usar las X-Windows de forma aceptable se precisaran las siguientes características:

- 486 DX

- 16 MB RAM
- 100 Mb Disco Duro Libres

La necesidad de disco duro irá en proporción a la cantidad de programas que se instalen. Como mínimo son precisos 20 Mb y una instalación completa puede aproximarse a los 130 Mb. Como se ve las estaciones inteligentes realizan un alto consumo de disco.

2.6. Instalación y Configuración.

Linux se puede instalar, usualmente, en 2 tipos de particiones. Aquí trataremos la distribución Slakware.

En una MSDOS/WINDOWS o en una LINUX. Si se instala sobre una MSDOS/WINDOWS el sistema ira bastante más lento que en una típica LINUX.

Cuando se instala en MSDOS/WINDOWS se precisará el paquete de utilidades UMSDOS.

Las particiones del disco duro permiten, como su nombre dice, partir este en diversas partes para que puedan ser utilizadas por diversos sistemas operativos. La información donde se encuentra la cantidad de disco que le corresponde a cada partición está en la tabla de particiones.

Para poder realizar una partición se precisa modificar los datos contenidos en ella. Pero el proceso de "partir" el disco duro es delicado y precisa muchas precauciones.

Anteriormente se precisaba usar el programa fdisk del MSDOS. Pero este programa borraba toda la información y se necesitaba realizar un Back-up de este en decenas de disquetes, lo que lo hacia pesado.

Afortunadamente un usuario de Linux decidió solventar este problema creando un programa que no formatease el disco al cambiar la Tabla de Particiones. Este es el FIPS,

- 16 MB RAM
- 100 Mb Disco Duro Libres

La necesidad de disco duro irá en proporción a la cantidad de programas que se instalen. Como mínimo son precisos 20 Mb y una instalación completa puede aproximarse a los 130 Mb. Como se ve las estaciones inteligentes realizan un alto consumo de disco.

2.6. Instalación y Configuración.

Linux se puede instalar, usualmente, en 2 tipos de particiones. Aquí trataremos la distribución Slakware.

En una MSDOS/WINDOWS o en una LINUX. Si se instala sobre una MSDOS/WINDOWS el sistema ira bastante más lento que en una típica LINUX.

Cuando se instala en MSDOS/WINDOWS se precisará el paquete de utilidades UMSDOS.

Las particiones del disco duro permiten, como su nombre dice, partir este en diversas partes para que puedan ser utilizadas por diversos sistemas operativos. La información donde se encuentra la cantidad de disco que le corresponde a cada partición está en la tabla de particiones.

Para poder realizar una partición se precisa modificar los datos contenidos en ella. Pero el proceso de "partir" el disco duro es delicado y precisa muchas precauciones.

Anteriormente se precisaba usar el programa fdisk del MSDOS. Pero este programa borraba toda la información y se necesitaba realizar un Back-up de este en decenas de disquetes, lo que lo hacia pesado.

Afortunadamente un usuario de Linux decidió solventar este problema creando un programa que no formatease el disco al cambiar la Tabla de Particiones. Este es el FIPS,

disponible en sunsite. FIPS cambia la información de la tabla escribiendo directamente sobre ella.

Para poder utilizar FIPS se precisan los siguientes pasos:

2.6.1. Uso de FIPS para realizar Particiones.

Generalmente FIPS es un programa que se asegura bastante antes de realizar operaciones. No se han dado conocidos, pero en caso de que se produzcan un error fatal se tendrá que proceder con fdisk, intentando previamente salvar toda la información previamente almacenada. Para realizar las prácticas se siguen los siguientes pasos:

1. Chequear el disco con alguna utilidad de escaneo como Norton Disk Doctor o Scan Disk de MSDOS 6.0.

2. Defragmentar el disco duro en forma de optimización completa, de forma que todos los ficheros se almacenen al principio del disco. Esto permitirá que FIPS pueda utilizar el espacio libre restante evitando que se pierdan o sobrescriban ficheros de MSDOS/WINDOWS a la hora de hacer la partición. Esto se realiza con SPEEDISK, una herramienta de Norton Utilities o DEFRAG de MSDOS 6.0. Hay programas de uso público que realizan esta labor.

3. Desactivar El IMAGE o MIRROR, de las PCTOOLS o de las NORTON, debido a que este fichero se almacena al final del disco duro impide al FIPS que encuentre el primer sector del disco duro que se encuentre libre hasta el final del disco. Después de esto borrar el fichero que contiene la información, mirrorsav.fil en el caso de las PCTOOLS.

4. Arranca con el disquete FIPS

5. Activa FIPS

6. En la versión 1.5 de este programa preguntará si se desea realizar un BACKUP de la tabla de particiones. Se debe responder "Sí" a la pregunta. Si algo va mal se podrá rectificar de posibles errores con el programa RESTORRB.EXE en caso de fallo.

7. Elegir la cantidad de disco que le corresponde a la partición antigua y a la nueva.

8. Salir confirmando los cambios.

9. Volver a iniciar la computadora con el disquete FIPS dentro. Generalmente se suele realizar una partición adicional para que el Linux en el que éste realiza sus swapping. Esta es una partición especial. Para crearla se deben repetir los pasos del 5 al 8, pero dándole a esta partición un tamaño mayor a 8Mb como mínimo.

10. Volver a iniciar la computadora con el disquete FIPS dentro.

11. Acceder al c: Y comprobar con el NDD o SCANDISK el disco rectificar todos los posibles errores que se produzcan. Comprobar si la unidad antigua contiene el tamaño asignado por FIPS.

12. Volver a iniciar la computadora sin el disquete FIPS.

13. Todo correcto.

En caso de que haya habido graves problemas al comprobar c: en el paso 11 o 13. Deberemos proceder a restituir la tabla de particiones. Afortunadamente este proceso es sencillo gracias al programa de utilidad RESTORRB.EXE que acompaña a FIPS.

2.6.2. Uso de RESTORRB.

Arranca el ordenador con el disco de FIPS

1. Activa RESTORRB

2. El programa permitirá elegir entre diversos ficheros backup creados en el punto 6 del uso de fips. Elige el más reciente, el de numeración más alta.

3. Volver a reiniciar el ordenador con el disquete de FIPS

4. Accede a c: Y comprueba.

5. Si se han realizado varias particiones repetiremos el proceso pero substituyendo el fichero por el inmediatamente anterior.

2.6.3. Limitaciones de FIPS.

FIPS realiza muchos chequeos de seguridad antes de comenzar su trabajo. FIPS no esta comprobado para todo tipo de disco duro. Veamos sus principales limitaciones:

- No puede trabajar sobre discos duros que su BIOS no use la interrupción 13 para realizar accesos a bajo nivel al disco.

- Tampoco trabaja con discos duros con tamaño de sector 512 Bytes.

- No funciona FIPS sobre tablas de particiones de 12 bits. No realiza particiones sobre discos formateados con sistema operativo MSDOS 3.0 o inferior.

- No puede trabajar sobre particiones extendidas del DOS.

- No puede partir con discos que contengan más de 4 particiones.

- No es capaz partir con discos con más de 4085 clusters.

2.6.4. Instalación de Linux.

Después de haberse realizado las particiones para Linux y para swap comienza el proceso de instalación del Linux es decir la instalación del kernel en la partición de Linux.

Si se ha traído el Linux por vía disquetes se debe verificar que existan los programas rawrite.exe, un bootdisk y un rootdisk. Estos disquetes deben contener el kernel del Linux compilados para computadoras con procesador Intel.

Pero si se dispone de un Compact Disc con los programas de instalación de Linux se puede reiniciar el equipo con el CD para que automáticamente se ejecute el programa de instalación, o de igual forma se puede generar un disquete de inicio con el programa RAWRITE: EXE.

Tras tener todo preparado se inicia el ordenador con el bootdisk.

Después de pulsar [ENTER] el Bootdisk verificará la computadora para ver cuál es el hardware. Si este no es muy raro el proceso ira sobre ruedas. Un posible fallo en este proceso es que el disco duro supere los 500 MB de tamaño. En este caso la tabla de particiones supera los 16 bits cosa que en antiguas versiones del kernel resultaba insuperable, en este caso es necesario utilizar una versión más reciente del bootdisk.

Durante el proceso de instalación de Linux aparece una pantalla para la administración de particiones en el que se debe especificar por lo menos dos de ellas, una para los archivos nativos de Linux que generalmente por omisión es la raíz "/ ", y otra para la memoria de intercambio swap, que por lo menos debe ser tamaño de la memoria RAM existente en la máquina

Generalmente /dev/hda1 corresponde a la partición inicial del disco duro. Si existe otra partición el sistema automáticamente asigna un identificador, por ejemplo /dev/hda5.

Al finalizar la instalación se puede reiniciar el equipo, una vez reiniciado aparecerá la palabra LILO, que significa Linux Loader, en el cual se ingresa con la cuenta del administrador (root) y la respectiva contraseña, que se especificó en la instalación.

2.7.Características y Funcionamiento.

En términos generales podemos decir que disponemos de varios tipos de sistema de archivos para poder acceder a archivos en otras plataformas. Incluye un entorno gráfico X Window, que nada tiene que envidiar a los modernos y caros entornos comerciales. Está orientado al trabajo en red, con todo tipo de facilidades como correo electrónico por ejemplo. Posee cada vez más software de libre distribución, que desarrollan miles de personas a lo largo y ancho del planeta. Linux se está convirtiendo en el sistema operativo preferido por la mayoría de los informáticos.

Un ejemplo de la popularidad que ha alcanzado este sistema y la confianza que se puede depositar en él es que incluso la NASA ha encomendado misiones espaciales de control de experimentos a la seguridad y la eficacia de Linux.

2.7.1. Linux y sus Shells.

Cada usuario de un sistema Linux tiene su propia interfaz de usuario o Shell. Los usuarios pueden personalizar sus shells adecuándolos a sus propias necesidades específicas. En este sentido, el Shell de un usuario funciona más como un entorno operativo que el usuario puede controlar.

Linux permite la utilización de distintos tipos de shell programables. Para aquellos que se pregunten qué es un shell es como el command.com de ms-dos, es decir, un intérprete de comandos. Es básicamente la interfaz, el modo de comunicación, entre el usuario y el sistema. Cada shell tiene sus características propias. La principal diferencia que existe entre los distintos tipos de shell radica en la sintaxis de la línea de comandos.

No es necesario aprender a programar con todos los tipos de shell ya que sabiendo uno, los conocemos todos, así que es mucho más sencillo de lo que parece.

2.7.2. Linux es Multitarea.

La multitarea no consiste en hacer que el procesador realice más de un trabajo al mismo tiempo (un solo procesador no tiene esa capacidad), lo único que realiza es presentar las tareas de forma intercalada para que se ejecuten varias simultáneamente, a menos que sea un computador que tenga más de un procesador, para lo cual Linux soporta esta característica.

2.7.3. Linux es Multiusuario.

Para que pueda desarrollar esta labor (de compartir los recursos de un ordenador) es necesario un sistema operativo que permita a varios usuarios acceder al mismo tiempo a través de terminales, y que distribuya los recursos disponibles entre todos. Así mismo, el sistema debería proporcionar la posibilidad de que más de un usuario pudiera trabajar con la misma versión de un mismo programa al mismo tiempo, y actualizar inmediatamente cualquier cambio que se produjese en la base de datos, quedando reflejado para todos.

Pues bien, este sistema operativo no lo tenemos que inventar puesto que ya esta inventado. Pero no todo es tan bonito como se pinta ya que el hecho de que se conecten al servidor más usuarios significa que es más difícil mantener la seguridad. Otra de las características referentes a esta tema es que Linux es multiplataforma. Fue diseñada para plataforma Intel pero ha sido fácilmente exportado a diversos tipos de sistema.

2.7.4. Linux es Seguro.

El concepto de seguridad en redes de ordenadores es siempre relativo. Un sistema puede ser seguro para un determinado tipo de actividades e inseguro para otras. Por ejemplo, nosotros no guardaríamos secretos de estado en un sistema Linux al que pudiera acceder mucha gente y careciese de un administrador dedicado absolutamente a la tarea, ya que según todos los hackers (intrusos), no hay sistema cuya seguridad sea perfecta. El sistema de contraseñas que protege el acceso al sistema se basa en algoritmos de seguridad. Pero claro, por muy bueno que sea el algoritmo, si después permitimos a sus usuarios poner como contraseña su nombre de usuario, de nada servirá la contraseña y todos sus esfuerzos.

Si se quiere que el sistema sea seguro, se debe administrar de tal forma que se tengan controlados a los usuarios en todo momento, para poder aconsejarles e incluso regañarles, en caso de que cometan alguna imprudencia, todo ello con el fin de mantener la propia seguridad de sus datos y de los nuestros. Para ayudarse a mantener la seguridad surgen nuevas herramientas constantemente, tanto para detectar intrusos como para encontrar fallos en el sistema y evitar así ataques desde el exterior.

2.7.5. Linux y su Control de Dispositivos.

Una vez instalado Linux se podrá acceder a un directorio llamado /dev. Dentro de él se observa un montón de archivos con nombres tan dispares como hda1(Disco Duro IDE) o mouse. Estos son los controladores de dispositivos del sistema. La mayoría de los sistemas operativos para ordenadores personales, como MS-DOS/WINDOWS, llevaban parcialmente implementadas en el núcleo las facilidades de acceso a los distintos dispositivos, como el disco duro o el ratón, de tal modo que a no ser que sé reescriba el núcleo, difícilmente se podrá tener el control sobre nuevos tipos de dispositivos.

Los controladores son tratados de forma independiente al núcleo del sistema, y por lo tanto se podrá añadir tantos controladores como dispositivos nuevos se vayan añadiendo

al ordenador. Por otra parte todos los dispositivos son tratados de igual forma, y gracias a ello se podrá redirigir datos de la misma manera al disco duro o a la impresora.

2.7.6. Linux y las Redes de Ordenadores.

Cuando se trabaja con Linux se está ante un sistema operativo orientado al trabajo de redes de ordenadores. Se dice esto porque cuando se trabaja con un sistema como MS-DOS/WINDOWS se sabe que todas las operaciones que conllevan las órdenes ejecutadas se llevan a cabo dentro de la carcasa del ordenador mientras que en Linux no se puede garantizar esta afirmación.

Linux dispone de varios protocolos como PPP, SLIP, TCP/IP, PLIP, etc., para la transferencia de archivos entre plataformas. Tiene a su disposición multitud de aplicaciones de libre distribución que permiten navegar a través de Internet y enviar y recibir correo electrónico. Posee gran variedad de comandos para comunicación interna entre usuarios que se encuentren ubicados en plataformas distintas (gracias a utilidades como telnet). En fin, un universo de posibilidades de comunicación a recopilar las distintas aplicaciones escritas para Linux y ponerlas en uno u otro formato, con diferentes facilidades de instalación, mantenimiento y configuración.

La licencia garantiza la libre distribución de las aplicaciones, pero las empresas pueden cobrar por el trabajo de agrupar un determinado conjunto de esas aplicaciones y hacer más sencilla su instalación. Lo único que no varía para nadie es el núcleo del sistema, que se desarrolla de forma coordinada y con actualizaciones sistemáticas. Es por ello que antes de instalar Linux hemos de elegir qué distribución nos interesa más.

2.8.Ventajas de Linux.

Entre las ventajas más importantes de Linux se tienen las siguientes:

- Multiusuario

- Multitarea

- Soporta acceso remoto.

- Soporte nativo de TCP/IP (Fácil conexión a Internet y otras redes).

- Contiene xFree86, que es una interfaz gráfica de usuario basada en los estándares de X- Window, y también es gratuita.

- Al instalar el sistema operativo, también se tiene la posibilidad de instalar varios programas, tales como hojas de cálculo, bases de datos, procesadores de texto, varios lenguajes de programación, paquetes de telecomunicaciones y juegos.

- Cumple los estándares POSIX y de Sistemas Abiertos, esto es que tiene la capacidad de comunicarse con sistemas distintos a él.

- Existe mucha documentación sobre éste.

2.9. Desventajas de Linux.

Entre las desventajas de Linux se puede mencionar:

- Carencia de soporte técnico.

- Debido a que el código del Kernel está disponible, es posible que el sistema de vulnerable a ataques de intrusos.

- Problemas de hardware, no soportan las plataformas y no es compatible con algunas marcas específicas.

- No existe un control de calidad al momento de elaborar software para Linux, pues muchas veces las aplicaciones se hacen y se liberan sin control alguno.

- Es poco probable que aplicaciones para DOS y OS/2, se ejecuten correctamente bajo Linux.

- No hay forma segura de instalarlo sin reparticionar el disco duro.

- El reparticionar el disco duro, implica borrar toda la información del mismo y después restablecerla.

- Se requiere experiencia y conocimiento del sistema para administrarlo, pues como es un sistema por línea de comandos, estos poseen muchas opciones y en ocasiones es difícil realizar algunas tareas, que en otros sistemas operativos de red son triviales.

CAPITULO III.

3. INTERNET.

3.1.Introducción.

Con más de 200 millones de usuarios en todo el mundo, Internet se ha convertido en el medio de comunicación más extendido en toda la historia de la humanidad. Esta, bien llamada "Red de redes", permite establecer la cooperación y colaboración entre gran número de comunidades y grupos de interés por temas específicos, distribuidos por todo el planeta.

A través de ella es posible encontrar toda clase de software para una gran variedad de computadoras y sistemas operativos, pueden consultarse los catálogos de las bibliotecas más importantes del mundo, acceder a bases de datos con los temas más diversos y transferir copias de los documentos encontrados, es posible visualizar y copiar archivos de imágenes con fotografías de todo tipo o reproducciones de cuadros, pueden hacerse cosas como conversar a tiempo real dos o más personas, separadas por miles de kilómetros de distancia.

Internet nació de manos de ARPAnet en tiempos de la guerra fría. Esta empresa, estaba afiliada al ejército norteamericano de forma que todos sus nuevos descubrimientos, estaban destinados a la seguridad de Norteamérica.

ARPAnet consiguió la forma de aunar estas redes informáticas, mediante un protocolo al que denominaron TCP/IP (Transfer Control Protocol / Internet Protocol).

Mediante este protocolo, consiguieron que los datos pudiesen viajar libremente entre las distintas redes. Este protocolo, funciona enviando una série de paquetes binarios de un

sistema a otro. Existen varios tipos de paquetes, de forma que podemos hablar de nuevos protocolos de envío de datos.

3.1.2. Concepto.

Internet es un conjunto de redes, redes de ordenadores de equipos físicamente unidos mediante cables que conectan puntos de todo el mundo.

Esta gigantesca red se difumina en ocasiones porque los datos pueden transmitirse vía satélite, o a través de servicios como la telefonía celular o porque a veces no se sabe muy bien a donde esta conectada. El acceso a Internet suele ser libre y gratuito para estudiantes y profesores (debido a su filosofía de origen) y también lo es el uso de la infraestructura para todo el mundo, aunque los particulares y proveedores de acceso han de pagar para tener acceso y realizar actividades comerciales o privadas.

3.1.3. Estructura Cliente/Servidor.

El término alude a un procesamiento colaborativo de datos entre dos o más ordenadores conectados en una red. Si el entorno es multimedia, el cliente es el dispositivo que visualiza el video, cuadros y texto, o reproduce el audio distribuido por el servidor. El cliente puede ser un ordenador personal o un asistente digital personal, incluso una televisión inteligente que pueda comprender datos digitales. El servidor es el depositario del video digital, audio, fotografías digitales y texto, y los distribuye bajo demanda; debe ser una máquina capaz de almacenar los datos y ejecutar todo el software que suministra éstos al cliente.

Una red multimedia puede tener un servidor que distribuya datos a múltiples clientes a la vez. Un cliente también podría tener múltiples servidores enviando datos simultáneamente.

En el entorno de computación actual, un ordenador Macintosh, Windows, UNIX o una computadora grande, puede ser un cliente. Cualquiera de estas plataformas puede actuar como servidor e incluso puede actuar como cliente y servidor simultáneamente. Esta doble función es posible debido a las capacidades multitarea de los modernos sistemas operativos.

3.1.4. Ventajas y Desventajas del Uso de Internet.

Entre las principales ventajas que ofrece el uso de Internet como fuente de información tenemos:

- El ahorro de tiempo al no tener que trasladarse hacia una biblioteca. Esto siempre y cuando posea el acceso a Internet en su casa.

- El ahorro de tiempo al no tener que transcribir gran cantidad de caracteres, utilizando el método de "copiar y pegar" texto.

- El ahorro de costos al no tener que pagar por copias fotostáticas o por la compra de libros o revistas.

- El acceso a una cantidad mayor de fuentes de información, con motores de búsqueda especializados que ahorran el tiempo de búsqueda de los datos.

- El acceso a herramientas informáticas para el intercambio de la información, tales como: e-mail, charlas en línea (chats), forum de discusión, etc.

- El ahorro de tiempo permite culminar con más rapidez los trabajos de investigación.

- El acceso a grandes bases de datos ofrece la oportunidad de realizar un trabajo de mayor calidad.

- El acceso directo a la tecnología moderna obliga al usuario a mantenerse al día en los cambios tecnológicos. Esto incide positivamente en el desarrollo cultural del individuo.

- Sirve como complemento del aprendizaje a la par del uso de las bibliotecas tradicionales.

Quizá la posible gran desventaja que podríamos encontrar en un uso muy frecuente de Internet para buscar información es la de que ciertos estudiantes se dedican a copiar textualmente monografías que se encuentran en la red, sin hacer uso de su creatividad y de su capacidad de análisis. Esto evidentemente afecta negativamente el desarrollo intelectual de estos estudiantes.

3.2. INTRANET.

3.2.1.Concepto.

Se entiende como Intranet el sistema de información interna, basado en tecnología Internet, utilizado por una empresa. Aunque su uso se limita habitualmente al personal de la misma, puede convertirse en una Extranet y hacerse extensivo a los clientes, proveedores y colaboradores de dicha organización.

Una Intranet representa la parte "inteligente" de la organización y su objetivo es organizar el trabajo de cada una de las personas de la empresa a un mínimo coste, tiempo y esfuerzo con el único fin de llegar a ser más productiva, más rentable y más competitiva.

Al igual que en Internet, la pieza clave de la Intranet es el World Wide Web, es el servicio más utilizado actualmente dentro de Internet, junto con el correo electrónico es

una herramienta basada en hipertexto que permite recuperar y mostrar información en diferentes tipos.

En una Intranet también se utiliza correo electrónico, aunque este es interno, es decir, sin necesidad de tener acceso a Internet. Igualmente se utilizan el resto de herramientas de Internet: listas de distribución, boletines de noticias, transferencia de ficheros, acceso remoto, charlas interactivas, videoconferencia, etc.

3.2.2. Ventajas de una Intranet.

Dentro de la ventajas más importantes de una Intranet tenemos:

- Mejora la competitividad.
- Ahorra costes y tiempo.
- Mejora la productividad.
- Mejora las comunicaciones y promueve la participación.
- Permite una relación directa con el cliente.
- Contribuye a la implantación y mantenimiento de la calidad.
- Permite desarrollar el tele trabajo.

3.2.3. Características de la Intranet.

Existen diferentes características de una Intranet entre las más importantes tenemos las siguientes:

- Sencillez de funcionamiento para el usuario.
- Flexibilidad y permanentemente actualización.
- Incorpora la tecnología multimedia.
- Es interactivo.
- Es una tecnología cliente / servidor.
- Compatibilidad con cualquier tipo de plataforma informática.

EXTRANET

3.3.1. Concepto.

Una definición aproximada de extranet sería "tecnología para transferencia segura de datos entre intranets protegidas por cortafuegos".
La idea actual con las extranet pareciera ser vincular intranets, que en principio deberían estar fuera del dominio público. Entonces sucede algo que pudiera entenderse como el regreso al punto de partida ya que, en efecto, la propia Internet es una red de redes internas.

Una **extranet** es una intranet extendida fuera de los límites de la empresa o corporación. Cuando el acceso selectivo es proporcionado a entidades fuera de la organización, la intranet se convierte en extranet. Por ejemplo, si tu compañía permite a sus proveedores, clientes, etc., tener acceso a tu intranet, estaremos hablando de una extranet. Ese acceso puede ser restringido por mecanismos de seguridad como identificadores de usuario y contraseñas (passwords).

Una extranet podría calificarse como una extensión de la Intranet en la que se permite el acceso a proveedores y clientes de la compañía.

Seguimos sin hablar de acceso a internet. Algunas de las razones de las ventajas de habilitar este acceso serian: mejorar las comunicaciones con los clientes dado que en este tipo de acceso podemos habilitar un canal de comunicación en el que puedan consultar el estado de sus pedidos, enviar sugerencias y quejas, porque no, también podrían hacer sus pedidos; otro agente que puede beneficiarse de la extranet son los proveedores, con los que podemos habilitar un nuevo sistema de enlace que permite mejorar la comunicación, agilizar los tramites y abaratar costes .

3.3.2. Características.

Entre las principales características tenemos las siguientes:

- Permite acomodar nuevos miembros con flexibilidad.
- Conecta con proveedores y clientes. Permite análisis integrado del estado del medioambiente y el cumplimiento de las directivas de "reporting".
- Conecta a los proveedores de información.
- Conecta a los clientes. Permite acceso a bases de datos.

3.4. Protocolos de Comunicación.

En la actualidad contamos con muchos protocolos de comunicación comerciales con los cuales muchas veces aun sin darnos cuenta, los utilizamos, nos ayudan a hacer tareas como lo son el Internet, una transferencia por módem o una simple comunicación a un servicio en línea inteligente.

A continuación mencionaremos y explicaremos varios de estos protocolos, estos son los más importantes:

3.4.1. FTP (File Transfer Protocol), Protocolo de Transferencia de Archivos.

El objetivo principal de este protocolo son varios puntos, promover el compartir archivos entre computadoras (programas y datos), alentar al uso remoto de las computadoras, y transferir datos de una forma segura y óptima por computadora. FTP mas que para ser usado por un usuario directamente es para que los programas lo usen entre ellos para comunicarse.

FTP ha ido evolucionando demasiado en todos estos años desde que se creo, este empezó en 1971 con un modelo de transferencia llamado RFC 141 en M.I.T. Fue hasta después de muchas revisiones que llegó a RFC 265 cuando ya se le consideró un protocolo de transferencia de archivos completa entre HOSTs (servidores de archivos) de ARPHANET.

El tipo de datos ASCII, es el mas común en el protocolo FTP, este se usa cuando se transfieren archivos de texto, la computadora que envía (sender), debe convertir cualquiera que sea su estructura de archivos interna, debe convertir sus datos al formato genérico de 8 bits, y el que recibe (receiver) lo debe convertir de nuevo a su formato propio.

El tipo EBCDIC es el más eficiente cuando ambos el que recibe y el que envía lo usan como formato propio, este tipo se representa también en 8 bits pero de forma EBCDIC.

Lo único en lo que cambian es en la forma de reconocer los códigos de los caracteres.

El formato de IMAGEN es cuando se empaca todo lo que se quiere enviar en cadenas seguidas de paquetes de 8 bits, esto es no importa el formato en que internamente se maneje la información, cuando se va a enviar se tiene que hacer una conversión de 8

bits en 8 bits y cuando el que recibe tiene todo el paquete, el mismo debe codificarlos de nuevo para que la transmisión sea completada.

En la estructura de datos en FTP se consideran tres tipos diferentes de archivos:

1. File – structure, donde no hay estructuras internas y el archivo es considerado una secuencia continua de bytes

2. Record – structure, donde los archivos contienen puros registros iguales en estructura.

3. Page – structure, donde los archivos contienen páginas enteras indexadas separadas.

Al establecer una conexión por FTP se debe tomar en cuenta que el mecanismo de transferencia consiste en colocar bien la transferencia de datos en los puertos adecuados y al concluir la conexión estos puertos deben ser cerrados adecuadamente. El tamaño de transferencia es de 8 bits, en ambos. El que va a transferir, debe escuchar desde el puerto hasta que el comando enviado sea recibido y este será el que dé la dirección de la transferencia. Una vez recibido el comando y establecido una transferencia del servidor a que solicita se inicializa la comunicación de la transferencia para verificar la conexión, esta es una cabecera con un formato específico, después de esto se comienza a enviar las tramas de 8 bits sin importar el tipo de datos que sea (antes mencionado), y al finalizar se envía otra trama cabecera ya establecida confirmando la transferencia completada.

3.4.2. HTTP (Hyper Text Transfer Protocol), Protocolo para la transferencia de hipertextos.

El Protocolo que controla la transferencia de datos en la World Wide web (WWW) es el HTTP, que proporciona un vehículo de entrega para las imágenes, gráficos, video, hipertexto u otros datos en la Web.

Mientras se recorre la web, el explorador intercambia mensajes con los servidores Web gracias al HTTP.

Cada vez que hace un clic en un hipervínculos o pasa de un recurso a otro, el explorador utiliza HTTP para acceder al recurso seleccionado.

HTTP es un protocolo sin estado, en otras palabras el explorador y el servidor deben crear y después cortar una conexión de red por cada operación HTTP, a este conjunto de peticiones y respuestas se denomina transacción.

Mediante HTTP los clientes y los servidores determinan de forma dinámica el formato de los documentos, lo que permiten que utilicen formato de datos no estándar para el intercambio de datos. Si el receptor no tiene un modo de ver o acceder a los datos, puede descargar un programa complemento que le permita recibir el contenido.

Las cabeceras de HTTP pueden contener información acerca de los objetos que transmite la aplicación a través de la Web. Con la información de las cabeceras, las aplicaciones Cliente-Servidor negocian formatos que pueden utilizar para transferir los objetos. Si no reconocen la información de la cabecera, la ignoran. Por tanto, puede probar nuevos protocolos en la Web sin comprometer la integridad del HTTP.

Además el protocolo esta basado en texto por lo cual es legible y no necesita decodificación

3.4.3. NFS (Network file system), Sistema de archivos de RED.

NFS es un sistema distribuido para archivos, este es utilizado en redes heterogéneas, con éste protocolo, el usuario solo ve un directorio cuando esta dentro de la red, claro que tiene ramas dentro pero no puede ver más arriba del nivel en el que se entra, tal vez los archivos dentro de esta estructura del directorio ni siquiera esta en la misma computadora.

3.4.4. POP3 Post office Protocol version 3.

Este es netamente un protocolo para la administración de correo en Internet.

En algunos nodos menores de Internet normalmente es poco práctico mantener un sistema de transporte de mensajes (MTS). Por ejemplo, es posible que una estación de trabajo no tenga recursos suficientes (espacio en disco, entre otros) para permitir que un servidor de SMTP(Simple Mail Transfer Protocol) [RFC821] y un sistema local asociado de entrega de correo esté residente y continuamente en ejecución.

A menudo es muy útil poder administrar correo sobre estos nodos, y frecuentemente soportan un user agent (UA) (agente de usuario) para ayudar en las tareas de manejo de correo. Para resolver el problema, un nodo que sí sea capaz de soportar un MTS ofrecerá a estos nodos menos dotados un servicio de maildrop. Se entiende por maildrop, el "lugar" en el sistema con el MTS donde el correo es almacenado para que los otros nodos puedan trabajar con él sin necesidad de mantener su propio MTS. El Protocolo de oficina de correos - Versión 3 (POP3) está destinado a permitir que una estación de trabajo acceda dinámicamente a un maildrop en un host servidor de forma útil y eficiente. Esto significa que el protocolo POP3 se usa para permitir a una estación de trabajo recobrar correo que el servidor tiene almacenado.

POP3 no está destinado a proveer de extensas operaciones de manipulación de correo sobre el servidor; normalmente, el correo es transmitido y entonces borrado. IMAP4 es un protocolo más avanzado y complejo y es tratado en [RFC1730] y revisado en [RFC 2060].

De aquí en adelante él termino (host) cliente se refiere a un host haciendo uso del servicio POP3 y host servidor al que ofrece este servicio. Inicialmente, el host servidor comienza el servicio POP3 leyendo el puerto 110 TCP. Cuando un host cliente desea de hacer uso del servicio, establece una conexión TCP con el host servidor. Cuando la conexión se establece, el servidor POP3 envía un saludo. Entonces, el cliente y el servidor de POP3 intercambian comandos y respuestas respectivamente hasta que la conexión se cierra o es abortada.

3.4.5. TCP/IP Transfer Communication Protocol / Internet Protocol.

El TCP/IP es un conjunto de protocolos de comunicación, es decir de convenciones particulares, creadas para permitir la colaboración y la partición de recursos entre más computadoras conectadas entre sí en la que está definida como red o network. Internet es en absoluto la más grande entre todas las redes existentes, debido a que logra conectar entre sí ordenadores personales y redes de menor amplitud en todo el mundo. Sobre Internet, de hecho, puede usted encontrar en conexión los ordenadores de instituciones del gobierno, militares, universidades y empresas privadas. Lo que permite a máquinas tan distintas por hardware y por prestaciones, comunicar entre sí de manera casi transparente, es el, TCP/IP, el cual constituye un tipo de 'lenguaje universal' comprendido y utilizado por todas las máquinas que cooperan en red.

El nombre más apropiado para indicar este conjunto de protocolos, es Internet Protocol Suite, es decir colección de protocolos de Internet. El TCP y el IP son dos protocolos que pertenecen a esta colección.

Puesto que éstos son también los protocolos más conocidos, ha entrado en el uso común llamar TCP/IP a toda la familia, aunque en algunas ocasiones una generalización parecida pueda resultar un error. Como quiera que se llame, el TCP/IP representa una familia de protocolos, proveen a la gestión de las funciones de bajo nivel, que son necesarias para la mayoría de las aplicaciones. El TCP y el IP pertenecen a los protocolos de bajo nivel. Sobre esta base, se desarrollan otros protocolos que gestionan funciones particulares, como la transferencia de ficheros, el envío del correo electrónico, la conexión remota, el control de los usuarios que se han conectado a la red en un momento específico, condividir impresoras y de programas aplicativos, y algo más.

Todo esto está generalmente simplificado en un modelo cliente/servidor, en el cual el servidor se identifica con el ordenador que proporciona un servicio específico, a través de la red (por ejemplo el (sitio FTP de VOL FTP es un servidor de ficheros y de informaciones sobre cómo utilizarlos de la manera mejor) y en el cual el término cliente se identifica con el ordenador que explota este servicio, aunque con la palabra cliente

incluya también aquellos programas que uno utiliza para tener acceso a estos mismos servicios (por ejemplo el Tiber y el Netscape son dos clientes típicos para tener acceso a las páginas del WWW).

3.4.5.1. Principales características de TCP.

- TCP es un protocolo orientado a conexión. Esto quiere decir que TCP mantiene información del estado de cada cadena de datos de usuario que circula por él. El término utilizado en este contexto significa también que TCP es responsable de la transferencia de datos entre extremos por la red o redes hasta la aplicación de usuario receptora (o el protocolo de nivel superior). TCP debe asegurar que los datos se transmiten y reciben correctamente por las computadoras atravesando las correspondientes redes.

- Como TCP es un protocolo orientado a conexión, es responsable de la transferencia fiable de cada uno de los caracteres (bytes u octetos) que reciben del nivel superior correspondiente. En consecuencia, utiliza números de secuencia y aceptaciones/rechazos.

- El término asociado con estos aspectos de los protocolos orientados a conexión es el de circuito virtual. Cada octeto transmitido lleva asignado un número de secuencia. El módulo TCP receptor utiliza una rutina de checksum para comprobar la posible existencia de daños en los datos producidos en el proceso de transmisión. Si los datos son aceptables, TCP envía una acepción positiva al módulo TCP remitente. Si los datos han resultado dañados, el TCP receptor los descarta y utiliza un número de secuencia para informar al TCP remitente del problema. Como muchos otros protocolos orientados a conexión, TCP emplea temporizadores para garantizar que no transcurre un lapso de tiempo demasiado grande antes de la transmisión de aceptaciones desde el nodo receptor y de la transmisión de datos desde el nodo transmisor.

- TCP recibe datos de un protocolo de nivel superior de forma orientada a cadenas. Esto es diferente a muchos otros protocolos empleados en la industria. Los protocolos orientados a cadenas se diseñan para enviar caracteres separados y no bloques, tramas, datagramas, etc. Los datos son enviados por un protocolo de nivel superior en forma de cadenas, byte a byte. Cuando llegan al nivel TCP, los bytes son agrupados para formar segmentos TCP. Dichos segmentos se transfieren a IP (o a otro protocolo de nivel inferior) para su transmisión al siguiente destino. Por tanto, las aplicaciones que trabajan normalmente con bloques de datos de longitud fija (una aplicación de gestión de personal que envíe registros de empleados de longitud fija o una aplicación de gestión de nóminas con registros de pago también longitud fija) no pueden utilizar TCP para transmitir bloques fijos al receptor. El nivel de aplicación debe ocuparse de configurar los bloques dentro de las cadenas de TCP.

- TCP comprueba también la duplicidad de los datos. En el caso de que el TCP remitente decida retransmitir los datos, el TCP descarta los datos redundantes. Estos datos redundantes podrían aparecer en la interred, por ejemplo cuando el TCP receptor no acepta el tráfico de manera temporizada, en cuyo caso el TCP remitente decidirá retransmitir los datos.

- Además de la capacidad de transmisión de cadenas, TCP soporta también el concepto de función push. Esta función se utiliza cuando una aplicación desea asegurarse de que todos los datos que han pasado al nivel inferior se han transmitido. Para hacer eso, gobierna la gestión del buffer de TCP. Para obtener esta función, el protocolo de nivel superior envía una orden a TCP con un identificador de parámetro de push a 1. Esta operación implica que TCP envía todo el tráfico almacenado en forma de segmento o segmentos hacia su destino. A más de utilizar los números de secuencia para aceptaciones, TCP los utiliza para la reordenación de los segmentos que llegan a su destino fuera de orden. Como TCP descansa sobre un protocolo no orientado a conexión, es bastante posible que en la interred se creen datagramas duplicados. TCP también elimina los segmentos duplicados.

- TCP emplea un esquema de aceptación inclusiva. El número de aceptación acepta todos los octetos hasta (e incluyendo) el del número de aceptación menos uno. Este esquema es un método muy sencillo y eficiente de aceptar tráfico, pero presenta una desventaja. Por ejemplo, supongamos que se han transmitido diez segmentos y debido a las operaciones realizadas durante el proceso de encaminamiento llegan desordenados. TCP está obligado aceptar sólo el mayor número de bytes contiguos recibidos sin error. No está permitido aceptar el byte de mayor número recibido hasta que hayan llegado todos los bytes intermedios. Por tanto, como en cualquier otro protocolo orientado a conexión, podría transcurrir el periodo de temporización de aceptaciones y TCP transmisora retransmitiría el tráfico no aceptado todavía. Esas retransmisiones podrían introducir una considerable sobrecarga en la red.

- El módulo TCP receptor se ocupa también de controlar el flujo de los datos del transmisor, lo que es muy útil para evitar el desbordamiento de los dispositivos de almacenamiento y la saturación de la máquina receptora. La idea que utiliza TCP es algo poco usual en protocolos de comunicación. Se basa en enviar el dispositivo transmisor un valor de "ventana". Se permite que el transmisor envíe un número máximo de bytes igual al valor de su ventana. Cuando se ha llegado a ese valor, la ventana se cierra y el transmisor debe interrumpir el envío de datos.Además, TCP posee una facilidad muy útil que permite multiplexar varias sesiones de usuario en un mismo computador. Esta operación se realiza definiendo algunas convenciones para compartir puertos y sockets entre usuarios.

- TCP proporciona transmisión en modo dúplex integral entre las entidades que se comunican. De esta forma, la transmisión se puede efectuar en ambos sentidos sin necesidad de esperar a la señal de indicación de cambio de sentido, necesaria en las transmisiones semidúplex. Además, TCP permite a los usuarios especificar niveles de seguridad y prioridades de las conexiones. Aunque esas opciones no están incluidas en todos los protocolos TCP, están definidas en el estándar TCP.

3.4.5.2. Niveles de la Arquitectura del TCP/IP.

La arquitectura del TCP/IP consta de cinco niveles o capas en las que se agrupan los protocolos, y que se relacionan con los niveles OSI de la siguiente manera:

- Aplicación: Corresponde con los niveles OSI de aplicación, presentación y sesión. Aquí se incluye protocolos destinados a proporcionar servicios, tales como correo electrónico, transferencia de ficheros, conexión remota y otros más recientes como el protocolo http(Hipertext Transfer Protocol).

- Transporte: Coincide con el nivel de transporte del modelo OSI. Los protocolos de este nivel, tales como el TCP, UDP, se encargan de manejar los datos y proporcionar la fiabilidad necesaria en el transporte de los mismos. La capa de transporte es la responsable de proveer la comunicación entre hosts. En UDP y TCP, los puntos finales(end points)en la comunicación son denominados ports.

- Internet: 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. En esta capa se direccionan y preparan los paquetes a ser ruteados, si no se encuentra la ruta, el paquete es enviado al geteway por default del host origen, también llamado router.

- Red: Es la interfaz de la red real. TCP/IP no especifica ningún protocolo concreto, así es que corre por las interfaces conocidas, emite al medio físico los flujos de bits y recibe los que de él provienen, consiste en los manejadores de dispositivos que se conectan en el medio de transmisión.

A continuación se indicará la relación existente entre las capas del Modelo OSI con las del protocolo TCP/IP, como se indica en la **tabla 3.1.**

OSI	TCP/IP	Protocolos del TCP/IP
Aplicación	Aplicación	telnet
Presentación		FTP
Sesión		SMTP
Transporte		HTTP
		SNMP
Red	Transporte	Red
Liga de Datos	Internet	TCP/IP IP/ICMP
Físico	Interfase de Red	Interfase de Red

Tabla 3.1: Relación entre las capas del modelo OSI con las del TCP/IP.

3.4. Comunicaciones en Red.

Los sockets son puntos finales de enlaces de comunicaciones entre procesos. Los procesos los tratan como descriptores de ficheros, de forma que se pueden intercambiar datos con otros procesos transmitiendo y recibiendo a través de sockets. El tipo de sockets describe la forma en la que se transfiere información a través de ese socket.

3.5.1. Sockets Stream *(TCP)*.

Son un servicio orientado a conexión, donde los datos se transfieren sin encuadrarlos en registros o bloques. Si se rompe la conexión entre los procesos, éstos serán informados de tal suceso para que tomen las medidas oportunas.

El protocolo de comunicaciones con streams es un protocolo orientado a conexión, ya que para establecer una comunicación utilizando el protocolo TCP, hay que establecer en primer lugar una conexión entre un par de sockets. Mientras uno de los sockets atiende peticiones de conexión (*servidor*), el otro solicita una conexión (*cliente*). Una

vez que los dos sockets estén conectados, se pueden utilizar para transmitir datos en ambas direcciones.

3.5.2. Sockets Datagrama *(UDP)*.

Es un servicio de transporte sin conexión. Son más eficientes que TCP, pero en su utilización no está garantizada la fiabilidad. Los datos se envían y reciben en paquetes, cuya entrega no está garantizada. Los paquetes pueden ser duplicados, perdidos o llegar en un orden diferente al que se envió.

El protocolo de comunicaciones con datagramas es un protocolo sin conexión, es decir, cada vez que se envíen datagramas es necesario enviar el descriptor del socket local y la dirección del socket que debe recibir el datagrama. Como se puede ver, hay que enviar datos adicionales cada vez que se realice una comunicación, aunque tiene la ventaja de que se pueden indicar direcciones globales y el mismo mensaje llegará a un muchas máquinas a la vez.

3.5.3. Sockets Raw.

Son sockets que dan acceso directo a la capa de software de red subyacente o a protocolos de más bajo nivel. Se utilizan sobre todo para la depuración del código de los protocolos.

Diferencias entre Sockets Stream y Datagrama.

Ahora se presenta un problema, al haber varias opciones, ¿qué protocolo, o tipo de sockets, se debe emplear - UDP o TCP? La decisión depende de la aplicación cliente/servidor que se esté escribiendo; aunque hay algunas diferencias entre los protocolos que sirven para ayudar en la decisión y decantar la utilización de sockets de un tipo.

En UDP, cada vez que se envía un datagrama, hay que enviar también el descriptor del socket local y la dirección del socket que va a recibir el datagrama, luego los mensajes son más grandes que los TCP. Como el protocolo TCP está orientado a conexión, hay que establecer esta conexión entre los dos sockets antes de nada, lo que implica un cierto tiempo empleado en el establecimiento de la conexión, que no es necesario emplear en UDP.

En UDP hay un límite de tamaño de los datagramas, establecido en 64 kilobytes, que se pueden enviar a una localización determinada, mientras que TCP no tiene límite; una vez que se ha establecido la conexión, el par de sockets funciona como los streams: todos los datos se leen inmediatamente, en el mismo orden en que se van recibiendo.

UDP es un protocolo *desordenado*, no garantiza que los datagramas que se hayan enviado sean recibidos en el mismo orden por el socket de recepción. Al contrario, TCP es un protocolo *ordenado*, garantiza que todos los paquetes que se envíen serán recibidos en el socket destino en el mismo orden en que se han enviado.

Los datagramas son bloques de información del tipo *lanzar y olvidar*. Para la mayoría de los programas que utilicen la red, el usar un flujo TCP en vez de un datagrama UDP es más sencillo y hay menos posibilidades de tener problemas. Sin embargo, cuando se requiere un rendimiento óptimo, y está justificado el tiempo adicional que supone realizar la verificación de los datos, la comunicación a través de sockets TCP es un mecanismo realmente útil.

En resumen, TCP parece más indicado para la implementación de servicios de red como un control remoto (*rlogin*, *telnet*) y transmisión de ficheros (*ftp*); que necesitan transmitir datos de longitud indefinida. UDP es menos complejo y tiene una menor sobrecarga sobre la conexión; esto hace que sea el indicado en la implementación de aplicaciones cliente/servidor en sistemas distribuidos montados sobre redes de área local.

3.6. Seguridades.

Uno de los puntos más vulnerables de las redes frente a ataques de intrusos, es la captura de información durante su transferencia. Aunque cada sistema que forma parte de una red se proteja internamente a sí mismo y a la información que tiene almacenada, cuando la información se transfiere de un sistema a otro no se conoce a priori el encaminamiento que va a seguir ni las medidas de seguridad que poseen los sistemas por los que atraviesa y los medios por los que se transmite. Por este motivo la transferencia segura de información a través de las redes es en la actualidad, el principal problema que los investigadores intentan solucionar.

Poco a poco en el Ecuador se está produciendo ya la explosión Internet, las compañías ya tienen a Internet como un proyecto implementado o a implementarse en el corto plazo, es decir se están abriendo al mundo. Pero cuidado, esa apertura es también la forma en que bases de datos en donde se guarda información se encuentran expuestas a cualquier persona que deambule en Internet.

El factor seguridad deja de ser ciencia-ficción en nuestro medio para convertirse en punto fundamental a la hora de implementar un sistema, ya que los tan mencionados hackers ya se encuentran aquí, así como en todo el mundo.

Se ha incrementado la variedad y cantidad de usuarios que usan la red para fines tan diversos como el aprendizaje, la docencia, la investigación, la búsqueda de socios o mercados, la cooperación altruista, la práctica política o, simplemente, el juego. En medio de esta variedad han ido aumentando las acciones poco respetuosas con la privacidad y con la propiedad de recursos y sistemas. *Hackers, frackers, crakers* ... y demás familias han hecho aparición en el vocabulario ordinario de los usuarios y de los administradores de las redes.

La propia complejidad de la red es una dificultad para la detección y corrección de los múltiples y variados problemas de seguridad que van apareciendo.

La preocupación del tema de seguridad no termina con la instalación de un Firewall o "muro de contención", sólo es un primer paso. Muchos ataques a servidores pueden pasar a través de las políticas de seguridad de estos aparatos y por lo tanto pulverizar nuestra tranquilidad.

Una vez extendidas las redes, no cabe duda de que en algunas aplicaciones telemáticas resultan imprescindible incorporar servicios de seguridad para que cubran los objetivos para los que fueron previstas. En aplicaciones de nuevo desarrollo sería conveniente que se tengan en cuenta los requerimientos de seguridad antes de su implantación.

En la actualidad la tecnología existente permite incluir seguridad en las aplicaciones con unos costes razonables. Las interfaces de usuario deben ser suficientemente amigables para conseguir que los servicios de seguridad añadidos dificulten lo menos posible el manejo de las aplicaciones.

3.6.1 Autentificación.

Llamamos autentificación a la comprobación de la identidad de una persona o de un objeto. Existen diversos sistemas que pueden servir para la autentificación de servidores, de mensajes y de remitentes y destinatarios de mensajes. Las claves privadas suelen estar alojadas en máquinas clientes y cualquiera que tenga acceso a estas máquinas puede utilizar las claves que tenga instaladas y suplantar la identidad de su legítimo usuario.

Por tanto es necesario que los usuarios adopten medidas de seguridad y utilicen los medios de autentificación de usuario de los que disponen sus ordenadores personales.

Hay tres sistemas de identificación de usuario, mediante contraseña, mediante dispositivo y mediante dispositivo biométrico.

La autentificación mediante contraseña es el sistema más común ya que viene incorporado en los sistemas operativos modernos de todos las computadoras. Las

computadoras que estén preparados para la autentificación mediante dispositivo sólo reconoce al usuario mientras mantenga introducida una "llave", normalmente una tarjeta con chip. Hay sistemas de generación de claves asimétricas que introducen la clave privada en el chip de una tarjeta inteligente.

Los dispositivos biométricos son un caso especial del anterior, en los que la "llave" es una parte del cuerpo del usuario, huella dactilar, voz, pupila o iris. Existen ya en el mercado a precios relativamente económicos ratones que llevan incorporado un lector de huellas dactilares.

3.6.2. Encriptación.

Las organizaciones de poder, a lo largo de la historia, han hecho del secreto de sus comunicaciones un principio fundamental de su actividad. Dicho secreto se intentó proteger mediante la encriptación, es decir, la codificación del lenguaje mediante una clave secreta sólo conocida por la organización emisora del mensaje y el destinatario del mensaje determinado por dicha organización. El anecdotario histórico abunda con ejemplos de batallas e, incluso, guerras supuestamente perdidas o ganadas mediante la interceptación y desencriptación de mensajes decisivos entre los centros de poder.

Por tanto, en cierto modo, no es de extrañar en la era de la información, basada en la comunicación de todo tipo de mensajes, que el poder (y, por tanto, la libertad) tenga una relación cada vez más estrecha con la capacidad de encriptar y descifrar. He aquí que lo que era una arcana tecnología matemática relegada a los dispositivos secretos de los servicios de inteligencia de los Estados se haya convertido, en el espacio de dos décadas, en la tecnología clave para el desarrollo del comercio electrónico, para la protección de la privacidad, para el ejercicio de la libertad en la red y, también, paradójicamente, para nuevas formas de control en la red. La encriptación es el principal campo de batalla tecnológico-social para la preservación de la libertad en Internet.

Una de las claves es *pública* en el sentido de que es asignada al originario/destinatario de un mensaje y que se conoce, mediante un listado, qué clave corresponde a quién. Pero, sin el conocimiento de la clave privada, es muy difícil, si no imposible, descifrar el mensaje. Esa otra clave es específica a cada individuo u organización, sólo quien la detenta la puede utilizar, pero sólo sirve con relación a su clave pública en la que recibe el mensaje.

Mediante este ingenioso sistema matemático, se garantiza a la vez la generalidad del cifrado y la individualidad de su desciframiento.

Como en otros temas de la historia de Internet, el poder de encriptación descentralizado recibió dos usos. Por un lado, fue comercializado. Por otro, sirvió como instrumento de construcción de autonomía de redes de comunicación.

La encriptación es la base de las firmas digitales que permiten el desarrollo del comercio electrónico en condiciones de relativa seguridad. En efecto, si la gente pudiera encriptar sus mensajes en lugar de enviar un mensaje por correo electrónico con su número de tarjeta de crédito abierto a todo el mundo, no tendrían por qué temer su interceptación y mal uso. Esto es, en realidad, lo que hacen las grandes empresas con capacidad de encriptación para transferir fondos y comunicarse mensajes confidenciales. Pero la tecnología de autentificación y firma digital se está difundiendo bajo el control de las empresas e instituciones, sin transmitir la capacidad autónoma de encriptación a los usuarios. Ello es así, por un lado, porque la comercialización de la tecnología creó un sistema de patentes que la hacen costosa en su uso comercial.

Pero, más importante todavía, las administraciones de casi todos los países han puesto enormes cortapisas a la difusión de la tecnología de encriptación por lo que ello representa de posible autonomía para los individuos y organizaciones contestatarias con respecto a los gobiernos y a las grandes empresas.

3.7. Virus en la Web.

El Virus es un pequeño programa escrito intencionalmente para instalarse en la computadora de un usuario sin el conocimiento o el permiso de este. Decimos que es un programa parásito porque el programa ataca a los archivos o sector es de "booteo" y se replica a sí mismo para continuar su esparcimiento.

Algunos se limitan solamente a replicarse, mientras que otros pueden producir serios daños que pueden afectar a los sistemas. Se ha llegado a un punto tal, que un nuevo virus llamado W95/CIH-10xx. o también como CIH. Spacefiller (puede aparecer el 26 de cada mes, especialmente 26 de Junio y 26 de Abril) ataca al BIOS de la PC huésped y cambiar su configuración de tal forma que se requiere cambiarlo. Nunca se puede asumir que un virus es inofensivo y dejarlo "flotando" en el sistema.

Existen ciertas analogías entre los virus biológicos y los informáticos: mientras los primeros son agentes externos que invaden células para alterar su información genética y reproducirse, los segundos son programas-rutinas, en un sentido más estricto, capaces de infectar archivos de computadoras, reproduciéndose una y otra vez cuando se accede a dichos archivos, dañando la información existente en la memoria o alguno de los dispositivos de almacenamiento de la computadora.

Tienen diferentes finalidades: Algunos sólo 'infectan', otros alteran datos, otros los eliminan, algunos sólo muestran mensajes. Pero el fin último de todos ellos es el mismo: propagándose.

Los virus se clasifican por el modo en que actúan infectando la computadora:

- Programa: Infectan archivos ejecutables tales como .com / .exe / .ovl / .drv / .sys / .bin

- Boot: Infectan los sectores Boot Record, Master Boot, FAT y la Tabla de Partición.

- Múltiples: Infectan programas y sectores de "booteo".

- Bios: Atacan al Bios para desde allí reescribir los discos duros.

- Hoax: Se distribuyen por e-mail y la única forma de eliminarlos es el uso del sentido común.

3.7.1. Daños de los Virus.

Definiremos daño como acción una indeseada, y los clasificaremos según la cantidad de tiempo necesaria para reparar dichos daños. Existen seis categorías de daños hechos por los virus, de acuerdo a la gravedad.

1. **Daños Triviales.-** Sirva como ejemplo la forma de trabajo del virus FORM (el más común): En el día 18 de cada mes cualquier tecla que presionemos hace sonar el beep. Deshacerse del virus implica, generalmente, segundos o minutos.

2. **Daños Menores.-** Un buen ejemplo de este tipo de daño es el JERUSALEM. Este virus borra, los viernes 13, todos los programas que uno trate de usar después de que el virus haya infectado la memoria residente. En el peor de los casos, tendremos que reinstalar los programas perdidos. Esto nos llevará alrededor de 30 minutos.

3. **Daños Moderados.-** Cuando un virus formatea el disco rígido, mezcla los componentes de la **FAT** (File Allocation Table, Tabla de Ubicación de Archivos), o sobreescribe el disco rígido. En este caso, sabremos inmediatamente qué es lo que está sucediendo, y podremos reinstalar el sistema operativo y utilizar el último backup. Esto quizás nos lleve una hora.

4. **Daños Mayores.-**Algunos virus, dada su lenta velocidad de infección y su alta capacidad de pasar desapercibidos, pueden lograr que ni aún restaurando un

backup volvamos al último estado de los datos. Un ejemplo de esto es el virus DARK AVENGER, que infecta archivos y acumula la cantidad de infecciones que realizó. Cuando este contador llega a 16, elige un sector del disco al azar y en él escribe la frase: "Eddie lives ... somewhere in time" (Eddie vive ... en algún lugar del tiempo).Esto puede haber estado pasando por un largo tiempo sin que lo notemos, pero el día en que detectemos la presencia del virus y queramos restaurar el último backup notaremos que también él contiene sectores con la frase, y también los backups anteriores a ese. Puede que lleguemos a encontrar un backup limpio, pero será tan viejo que muy probablemente hayamos perdido una gran cantidad de archivos que fueron creados con posterioridad a ese backup.

5. **Daños Severos.-** Los daños severos son hechos cuando un virus realiza cambios mínimos, graduales y progresivos. No sabemos cuándo los datos son correctos o han cambiado, pues no hay pistas obvias como en el caso del DARK AVENGER (es decir, no podemos buscar la frase Eddie lives).

6. **Daños Ilimitados.-** Algunos programas como CHEEBA, VACSINA.44.LOGIN y GP1 entre otros, obtienen la clave del administrador del sistema y la pasan a un tercero. Cabe aclarar que estos no son virus sino troyanos. En el caso de CHEEBA, crea un nuevo usuario con los privilegios máximos, fijando el nombre del usuario y la clave. El daño es entonces realizado por la tercera persona, quien ingresará al sistema y haría lo que quisiera.

CAPITULO IV

4. SISTEMA DE CORREO ELECTRONICO WEBMAIL.

4.1. Introducción.

El WebMail es un sistema de e-mail que opera bajo un explorador web, de múltiples características y personalizable. Algunas de las muchas características del WebMail comprenden: lista de contactos, programación diaria, corrector ortográfico y marcadores web. Se puede tener acceso a WebMail desde cualquier lugar con acceso a Internet y un explorador web.

El WebMail es un conjunto de scripts (pequeños programas) a través del cual se puede acceder al correo electrónico directamente desde cualquier navegador (Internet Explorer o Netscape). Esto es de gran utilidad, puesto que no importa en qué lugar un usuario se encuentre, podrá acceder a leer y responder sus mensajes de correo electrónico con total facilidad.

El webmail se ha convertido en la aplicación estrella para Internet porque no requiere la instalación de ningún software especial y permite a los usuarios acceder a su buzón de correo desde cualquier ordenador que tenga conexión, con sólo introducir su nombre de usuario y su contraseña.

4.2. Beneficios de Webmail.

Se puede acceder en cualquier lugar del mundo y desde cualquier computadora (Ej. cibercafé). Permite recolectar e-mails de otro servidor de correo. Es decir, si se tiene

una casilla en otro servidor se puede recibir los mensajes en la misma bandeja de entrada.

Un Webmail al operar sobre un explorador web puede ser accedido desde diferentes plataformas. Muchos Webmails poseen antivirus para analizar mensajes de correo entrantes.

Al igual que los programas desarrollados para administrar correo electrónico, un webmail es factible adjuntar archivos de cualquier formato y realizar cualquiera de las operaciones normales con los correos.

4.3. Interfaz de un Webmail en el lado Cliente.

Un Webmail típico aparece como una página web normal y amigable al usuario final o cliente, transparente a la tecnología, procesos y plataforma que residen en la estructura del webmail en el servidor.

Para entender el funcionamiento de un webmail se comenzará explicando la interfaz desde el punto de vista cliente y luego se analizará la parte técnica en el servidor.

4.3.1. Acceso a la cuenta.

Para acceder a una cuenta WebMail se conecta el explorador web a un URL específico.

Se escribe el login del usuario en el campo del nombre del usuario en la página para el acceso. Se escribe la contraseña de la cuenta o login en el campo para la contraseña. Se hace clic en el botón "Iniciar la sesión" para comenzar el proceso de autenticación. La contraseña no se muestra para propósitos de seguridad. Si la información es incorrecta se vuelve a presentar la página para el acceso.

Muchos Webmails además permiten al usuario escoger el idioma con el que desean trabajar. En otros casos el servidor. En otros casos el servidor de correo más cercano físicamente.

En la **fig.# 4.1** se muestra un ejemplo de ingreso a la cuenta de un WebMail.



Fig.# 4.1: Ejemplo de una página inicial de un WebMail.

4.3.2. Introducción a la Página Principal.

La página principal WebMail permite el acceso a las funciones de WebMail, tanto como a la observación de los recientes mensajes y a la programación del día. En la página principal están disponibles enlaces a las páginas de Correo, Contactos, Programación, Pendientes, Marcadores y páginas de cierre de sesión. También son accesibles las opciones de Grupo y Preferencias. Para acceder a cualquiera de las anteriores funciones se hace clic en el enlace apropiado. Para leer un mensaje se hace clic en el asunto del mensaje y aparecerá la página del mensaje. Si a este tópico se accede desde una página con ayuda sensible al contexto, se podrá obtener de mayor información mediante la lengüeta Contenido en: La página principal - perspectiva general de la cuenta.

En la **fig.# 4.2** se muestra un ejemplo de la página principal de un WebMail.

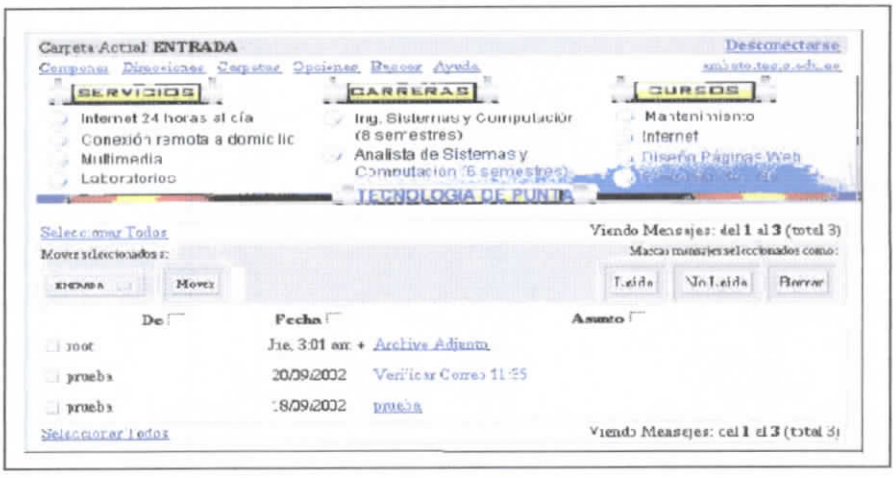


Fig.# 4.2: Ejemplo de la página principal de un WebMail.

4.3.3.Establecimiento de Preferencias.

Muchas funciones WebMail pueden ser personalizadas desde las páginas de preferencias. Para lograr acceso a las páginas de preferencia se hace clic en el enlace "Preferencias" de la página principal WebMail. Un menú desplegable en la página de preferencias contiene una lista de todas las funciones personalizables. Se muestran las opciones seleccionadas actualmente. Para observar las opciones de las funciones se selecciona uno de los menús desplegables y se hace clic en el botón "Ir".

Las preferencias se actualizan con un clic en el botón de "Actualizar preferencias". Las preferencias específicas de las páginas se pueden ver con un clic en el enlace "Prefs" localizado en la página apropiada. Si a este tópico se accede desde una página con ayuda sensible al contexto, se podrá obtener mayor información mediante la lengüeta Contenido en: Configuración de una cuenta WebMail.

En la **fig.# 4.3** se indica un ejemplo de esta página de preferencia u opciones.



Fig.# 4.3: Ejemplo de una página de referencias u opciones de un WebMail.

4.3.4. Límite del Tiempo de Conexión.

No hay límite para el tiempo en el que se puede estar conectado a la cuenta propia. WebMail no corta la sesión, independientemente del tiempo que esté inactiva. La sesión solamente termina cuando se salga voluntariamente o se cierre el explorador web.

4.3.5.Cierre de la Sesión de WebMail.

Para cerrar la sesión de WebMail se hace clic en el enlace "Cerrar sesión" de cualquier página. El salir de la sesión de WebMail aumenta la seguridad de la cuenta. Después que se abandone la sesión, cerrándola, ningún otro usuario puede utilizar el botón Back (atrás) del explorador para acceder a la cuenta.

4.3.6. Identificación de usuarios.

Muchos sitios Web que brindan el servicio de WebMail identifican a un usuario que ha ingresado antes desde el mismo computador a este sitio, esta identificación lo hacen por medio de pequeños programas que guardan una estructura de datos con la información

del usuario. Estos programas se llaman " cookies " y se almacenan en el computador cliente en forma transparente al usuario.

De tal manera que la próxima vez que un usuario conecte desde el mismo computador, el sitio Web buscará si existe un cookie en el cliente, si es encontrado lee la información almacenada en éste y le da la bienvenida con su nombre. login.

De la misma manera un WebMail utiliza cookies para mantener al corriente de la información del usuario, y por lo tanto, deben ser permitidos los cookies antes de conectarse al WebMail. La mayoría de los exploradores web modernos aceptan cookies e incluyen una opción para permitirlos o no.

4.4. Estructura de un WebMail en el lado servidor.

4.4.1. Introducción.

Un Webmail funciona en un espacio similar al de una página web, y de hecho está constituido por un conjunto de páginas web dinámicas. Tomando en cuenta este antecedente un Webmail trabaja con los mismos protocolos de comunicación tratados en el capítulo III (http, TCP/IP,etc).

Un Webmail además está formado por un conjunto de agentes de correo electrónico que permiten leer, extraer y enviar e-mails a través de la interfaz gráfica de un explorador Internet.

Dentro de estos, el agente más importante es el Sendmail.

4.4.2.Sendmail.

Es un conocido agente de transferencia de correo MTA (Mail Transport Agent) usado en Internet que maneja un amplio porcentaje de todos los correos ruteados en Internet. Este agente se traslada de un host a otro buscando correos ruteados. Existen otros agentes de transferencia de correo (que pueden ser utilizados con Red Hat Linux), pero la mayoria de administradores eligen utilizar Sendmail como MTA debido a su potencia, escalabilidad y compatibilidad con los estándares de Internet.

El deber principal de Sendmail, como otros MTAs, es el de trasladar con seguridad e-mails entre hosts, habitualmente usando el Simple Mail Transfer Protocol (SMTP). No obstante, Sendmail es altamente configurable, permitiendo controlar cada aspecto de cómo se gestiona un correo.

4.4.2.1. Características de Sendmail.

Existe una gran variedad de programas de correo electrónico que proveen al usuario de una aplicación para la creación y envío de e-mails. Estos programas son los llamados Agentes de Usuario o MUA (Mail User Agent), y su propósito es el aislar al usuario de los Agentes de Transporte o MTA (Mail Transport Agent), que son los encargados de transferir los e-mails a su correcto destino.

Sendmail es el agente de transporte de correo más común de Internet (en los sistemas UNIX). Aunque actúa principalmente como MTA, también puede ser utilizado como MUA (aunque no posee interfaz de usuario). Las misiones básicas de Sendmail son las siguientes:

- Recoger e-mails provenientes de un Mail User Agent (MUA) como pueden ser Eudora o Pine; o provinientes de un Mail Transport Agent (MTA) como puede ser el propio sendmail.

- Elegir la estrategia de reparto de los mails, basándose en la información de la dirección del destinatario contenida en la cabecera:

- Si el mail es local en nuestro sistema, enviará el mail al programa de reparto local de mails.

- Si el mail no es local, sendmail utilizará el DNS (Domain Name Solver) de nuestro sistema para determinar el host al que debe ser enviado el mail. Para transferir el mensaje, iniciará una sesión SMTP con el MTA de dicho host.

- Si no es posible mandar el mail a su destino (porque la maquina receptora esta desconectada, o va muy lenta), sendmail almacenará los mails en una cola de correo, y volverá a intentar el envío del mail un tiempo después. Si el mail no puede ser enviado tras un tiempo razonable, el mail será devuelto a su autor con un mensaje de error. Sendmail debe garantizar que cada mensaje llegue correctamente a su destino, o si hay error este debe ser notificado (ningún mail debe perderse completamente).

- Reformatear el mail antes de pasarlo a la siguiente máquina, según unas reglas de reescritura. Según el tipo de conexión que poseamos con una determinada máquina, o según el agente de transporte al que vaya dirigido el mail, necesitaremos cambiar los formatos de las direcciones del remitente y del destinatario, algunas líneas de la cabecera del mail, o incluso puede que necesitemos añadir alguna línea a la cabecera. sendmail debe realizar todas estas tareas para conseguir la máxima compatibilidad entre usuarios distintos.

- Otra función muy importante de Sendmail es permitir el uso de "alias" entre los usuarios del sistema; lo que nos permitirá (entre otras funciones) crear y mantener listas de correo entre grupos.

- Ejecución como agente de usuario (MUA). Aunque no posee interfaz de usuario, sendmail también permite el envío directo de mails a través de su ejecutable.

4.4.2.2. Funciones de Sendmail.

Las principales funciones de Sendmail son:

a) Sendmail como agente de usuario.

Como agente de usuario, Sendmail lee por defecto de su entrada estándar hasta encontrar un end-of-file (EOF) o una línea que contenga únicamente un punto (".") . En ese momento manda una copia de ese mensaje a cada una las direcciones destino (que se le habrán pasado previamente en la línea de comandos). El mismo determinará la ruta a seguir (consultando el DNS si fuera necesario), basándose en el contenido de la dirección destinataria.

Sendmail determina cómo enrutar un mensaje (cuyo destino sea la dirección o direcciones que le habremos establecido), de acuerdo a la información que posee en su fichero de configuración (por defecto /usr/lib/sendmail.cf).

Ejemplo 1:

Este comando manda un mail a "alonso@xyz.com", sin título y cuyo cuerpo contiene únicamente "hola que tal".

```
$ sendmail alonso@xyz.com
  hola que tal
  ..
$
```

b) Sendmail como agente de transporte.

La utilización de Sendmail como demonio en el sistema nos permite enviar y recibir correo SMTP (Simple Mail Transfer Protocol). Para ello, Sendmail se queda como proceso residente escuchando el puerto 25, admitiendo y realizando conexiones SMTP

Todas estas características y muchas otras que posee el sendmail deben ser configuradas y variarán de unos sistemas a otros. Para configurarlas hacemos uso del fichero de configuración de Sendmail.

La revisión y modificación de este fichero es bastante complicada y necesita de una serie de conocimientos previos.

1. Situación y descripción de los programas y ficheros de soporte utilizados por sendmail.

2. . /usr/sbin/sendmail (/usr/lib/sendmail en algunos sistemas)
   Ejecutable del sendmail.

3. /etc/sendmail.cf (/usr/lib/sendmail.cf)
   Fichero de configuración de sendmail.

4. . /etc/aliases
   Fichero donde se almacenan los alias del sistema.

5. . /usr/bin/newaliases

Reconstruye la base de datos de alias usada por Sendmail, a partir del fichero aliases. Es un enlace simbólico a Sendmail.

6. . /var/spool/mqueue
   Es el directorio donde se almacena la cola de correo.

7. . /usr/bin/mailq

Saca por pantalla el contenido de la cola de correo. Es un enlace simbólico a sendmail en el modo -bp.

cuando sea necesario (es decir, actúa como un Agente de Transporte de Correo (MTA)). Cuando reciba o una petición de conexión, creará un proceso hijo que se encargará de ello, mientras el proceso padre seguirá escuchando el puerto 25. Solo puede ser ejecutado en este modo por el superusuario del sistema.

CAPITULO V

5. ESTUDIO DE LOS LENGUAJES DE PROGRAMACIÓN PARA EL DESARROLLO DEL WEBMAIL

5.1. HTML (HiperText Markup Language).

El HTML no es más que una aplicación del SGML (Standard Generalized Markup Language), un sistema para definir tipos de documentos estructurados y lenguajes de marcas para representar esos mismos documentos. El término HTML se suele referir a ambas cosas, tanto al tipo de documento como al lenguaje de marcas.

A medida que nos afianzamos en el manejo de Internet cada uno de nosotros pasa por tres etapas diferentes: Al principio solamente conocemos unas pocas páginas, luego nos damos cuenta que existen buscadores lo cual lo hace más interesante y por último nos damos cuenta que en Internet no solamente se puede ver la información sino que también se puede publicar.

Para que varias personas se comuniquen es necesarios que éstas hablen un mismo idioma. El lenguaje que utilizan las computadoras que están conectadas a Internet es HTML.

5.1.1. Qué es el HTML.

El HTML, Hyper Text Markup Language (Lenguaje de marcación de Hipertexto) es el lenguaje de marcas de texto utilizado normalmente en la www (World Wide Web). Fue creado en 1986 por el físico nuclear Tim Berners-Lee; el cual tomo dos herramientas preexistentes: El concepto de Hipertexto (Conocido también como link o ancla) el cual

permite conectar dos elementos entre sí y el SGML (Lenguaje Estándar de Marcación General) el cual sirve para colocar etiquetas o marcas en un texto que indique como debe verse. HTML no es propiamente un lenguaje de programación como C++, Visual Basic, etc., sino un sistema de etiquetas. HTML no presenta ningún compilador, por lo tanto algún error de sintaxis que se presente éste no lo detectará y se visualizara en la forma como éste lo entienda.

El entorno para trabajar HTML es simplemente un procesador de texto, como el que ofrecen los sistemas operativos Windows (Bloc de notas), UNIX (el editor vi o ed) o el que ofrece MS Office (Word). El conjunto de etiquetas que se creen, se deben guardar con la extensión .htm o .html

Estos documentos pueden ser mostrados por los visores o "browsers" de paginas Web en Internet, como Netscape Navigator, Mosaic, Opera y Microsoft Internet Explorer.

5.2. JSP (Java Servlet Page).

El JSP (Java Servlet Pages) nació como respuesta al ASP de Microsoft y no nos vamos a detener demasiado en él porque es prácticamente lo mismo sólo con la diferencia de que lo que hay entre <% %> es código Java.

5.2.1. Qué es Java.

Java es un lenguaje originalmente desarrollado por un grupo de ingenieros de Sun, utilizado por Netscape posteriormente como base para Javascript. Si bien su uso se destaca en el Web, sirve para crear todo tipo de aplicaciones (locales, intranet o internet).

Java es un lenguaje:

- De objetos

- Independiente de la plataforma

Algunas caracteristicas notables:

- Robusto
- Gestiona la memoria automáticamente
- No permite el uso de técnicas de programación Inadecuadas
- Multithreading
- Cliente-servidor
- Mecanismos de seguridad incorporados
- Herramientas de documentación incorporadas

En Java, inclusive, ni siquiera existen las variables globales,(Aunque parezca dificil de aceptar, esto es una gran ventaja desde el punto de vista del desarrollo).

5.3. CGI (Common Gateway Interface).

El CGI (Common Gateway Interface) es un estándar para comunicar aplicaciones externas con los servidores de información, tales como servidores HTTP o *Web*. Un documento en HTML que el *daemon* del *Web* se trae es estático, es decir, se mantiene constante: un fichero de texto que no cambia. Un programa CGI, por otro lado es ejecutado en tiempo real, así que puede generar información dinámica.

Por ejemplo, supongamos que quieres enganchar tu base de datos de Unix al WWW, para permitir a gente de todo el mundo consultarla. Básicamente se necesitará un

programa CGI que el *daemon* del *Web* ejecutará para transmitir la información al gestor de base de datos, y recibir los resultados para presentárselos al cliente. Este es un ejemplo de pasarela (*gateway*), y es lo que CGI, actualmente en su versión 1.1, tiene en sus orígenes.

El ejemplo de la base de datos es una idea sencilla, pero la mayoría de las veces difícil de implementar. Realmente no hay límite con lo que tu quieras enganchar al Web. La única cosa que se debe recordar es que haga lo que haga el programa CGI, no deberá tardar mucho tiempo en procesar. De otra manera, el usuario estaría esperando con su navegador a que algo pase.

5.3.1. Especificaciones.

Como un programa CGI es un ejecutable, es equivalente a dejar al mundo ejecutar un programa en el sistema, que no es lo más seguro de hacer. Por ello existen una serie de precauciones de seguridad que son necesarias de implementar cuando se usan programas CGI. Así el servidor sabe que tiene que ejecutarlo, en vez de simplemente mostrarlo por pantalla. Este directorio está generalmente bajo el control del *webmaster*, prohibiendo al usuario medio crear programas CGI. Hay otros métodos para permitir el accesos a scripts CGI, pero depende del *webmaster* que se de esta posibilidad. Así que deberá contactar con su *webmaster* para consultar la factibilidad de permitirte un acceso a los CGI.

Si se dispone de una versión del servidor *HTTP NCSA*, se verá un directorio denominado *cgi-bin*. Este es el directorio especial antes mencionado, donde todos los programas CGI residen. Un programa CGI se puede escribir en cualquier lenguaje que permita ser ejecutado en el sistema, como:

C/C++
Fortran
PERL
TCL

Algún Shell de Unix
Visual Basic
AppleScript

Simplemente depende de lo que tenga en su sistema. Si se usa un lenguaje de programación como C o *Fortran*, como ya es conocido, se debe compilar el programa antes de poder ejecutarlo. Si se mira en el directorio *cgi-src*, se encontrará el código fuente de algunos programas CGI del directorio *cgi-bin*. Pero, si usa alguno de los lenguajes interpretados, como *PERL*, *TCL*, o un shell de *Unix*, el script simplemente necesita residir en el directorio /cgi-bin, ya que no tiene un código fuente asociado.

Mucha gente prefiere escribir scripts CGI en vez de programas, ya que son más fáciles de depurar, modificar y mantener que un programa típico compilado.

5.3.2. Directorio cgi-bin .

Este es un directorio especial, que contiene los scripts, configurado dentro del servidor http. El servidor conoce que este directorio contiene ejecutables que deberán ser ejecutados y su salida deberá ser enviada al navegador del cliente. No se puede simplemente crear un directorio *cgi-bin*, el administrador del servidor deberá configurarlo para su uso. Si no está configurado, los scripst serán cargados como simples ficheros de texto.

Algunos servidores están configurados de tal manera que los ficheros con una determinada extensión (generalmente *.cgi*) son reconocidos como scripts y serán ejecutados como si estuvieran en un directorio *cgi-bin*.

La configuración de los directorios, o de la extensión mencionada antes, depende únicamente del servidor.

funciones. No es un lenguaje de marcas como podría ser HTML, XML o WML. Está más cercano a JavaScript o a C, para aquellos que conocen estos lenguajes.

Pero a diferencia de Java o JavaScript que se ejecutan en el navegador, PHP se ejecuta en el servidor, por eso nos permite acceder a los recursos que tenga el servidor como por ejemplo podría ser una base de datos. El programa PHP es ejecutado en el servidor y el resultado enviado al navegador. El resultado es normalmente una página HTML pero igualmente podría ser una pagina WML.

Al ser PHP un lenguaje que se ejecuta en el servidor no es necesario que el navegador lo soporte, es independiente del navegador, pero sin embargo para que las páginas PHP funcionen, el servidor donde están alojadas debe soportar PHP.

5.4.2. Una breve historia de PHP.

PHP fue concebido alguna vez por Rasmus Lerdorf. Las versiones no-lanzadas tempranas fueron utilizadas en su Home Page para no perder de vista quién miraba su curriculum vitae en linea. La primera versión usada por otras personas estaba disponible alguna vez a principios de 1995 y era conocida como las herramientas personales del Home Page. Consistió en un motor muy simplista del programa de análisis que entendía solamente algunas macros especiales y un número de utilidades que entonces estaban en uso común en las páginas caseras traseras. Un guestbook (libro de visitas), un contador y un poco de otra materia. El programa de análisis fue reescrito a mediados de 1995 y se la nombró como la versión 2 de PHP/FI. El FI vino de otro paquete que Rasmus había escrito cuál interpretó datos de la forma del HTML. Él combinó las escrituras personales de las herramientas del Home Page con el intérprete de la forma y agregó la ayuda del mySQL.

Es duro dar cualquier estadística, pero se estima que a finales de 1996 PHP/FI estaba en uso en por lo menos 15.000 sitios de la tela alrededor del mundo. A mediados de 1997 este número había crecido sobre los 50.000. PHP para esas fechas cambió de ser un proyecto del animal doméstico de Rasmus al cual un puñado de gente había contribuido,

a ser un esfuerzo muy organizado de equipo. El programa de análisis fue reescrito por Zeev Suraski y Andi Gutmans y este nuevo programa de análisis formó la base para la versión 3 de PHP. Mucho código para uso general de PHP/FI fue cambiado hacia el lado de PHP3 y mucho de este reescrito totalmente.

A mediados de 1999 PHP/FI o el nuevo PHP3 incursionó con un número de productos comerciales tales como web server, RedHat Linux y una estimación conservadora de la plaza fuerte del C2 basada en una extrapolación de los números proporcionados por NetCraft sería que PHP estaba en uso encendido sobre 150.000 sitios alrededor del mundo. Poner eso en perspectiva, ése es más sitios que el servidor de la empresa del buque insignia de Netscape funcionado en el Internet.

5.4.3. Ventajas de PHP.

Aunque la sintaxis del lenguaje tiene cosas heredadas del C y el primer prototipo de PHP estuvo echo en Perl, actualmente tienen casi nada que ver, el PHP fue diseñado específicamente para web.

Si tuviera que ponerlo en pocas líneas "reglas" las ventajas de este lenguaje de programación, diríamos que:

- Al ejecutarse del lado del servidor, el usuario no requiere de ningún tipo de software especial.

- Es multiplataforma, por lo que puede ser instalado en servidores Unix (incluyendo Linux) y Windows (incluyendo 2000).

- Al manejar una sintaxis estandarizada, los programas que uno desarrolle para un servidor pueden correr transparentemente en otro, sin cambiarle una sólo línea de código.

- Es muy amigable con los usuarios novatos. Tiene varios años desde su primer desarrollo, por lo que es bastante maduro.

- Es muy rápida su ejecución, además de poderse conectar con la mayoría de las bases de datos importantes, como Oracle, Access, Informix, PostgreSQL, MySQL y Mini SQL.

- Es software libre, por lo que además es gratuito.

- La integración de PHP con páginas HTML es muy fácil y directa. Se puede "incluir" PHP en una página HTML ya terminada sin ningún problema. Esto significa que hay una clara separación de código y datos.

- El PHP debería ser comparado con el mod_perl, ya que es interpretado por un módulo del apache y no por un intérprete externo, que genera mas sobrecarga (*overhead*) al necesitar un fork/exec para ejecutar el intérprete. Programar con el mod_perl es algo más complicado que programar con Perl como CGI externo directamente, sobre todo por la persistencia y visibilidad (*scope*) de las variables que se use. Por ejemplo, las variables globales pueden ser accedidos por todos los programas que usen el mod_perl, lo que obliga a un control muy estricto de las variables globales. Hace un par de años, con programas hechos para Perl existían bastantes problemas de que se ejecuten correctamente con el mod_perl, al final los programadores terminaban cambiándose de lenguaje porque los programas eran complejos y era mucho trabajo pasarlos el mod_perl.

- El PHP, desde el punto de vista de *script* para webs, tiene prácticamente las mismas posibilidades y potencia que el Perl. Inclusive (y muy importante) es mucho más sencillo en la definición de interfaces (bases de datos en PHP) a costa de perder algo de flexibilidad.

- Desde el punto de vista del trabajo "cooperativo" con los diseñadores, es mucho más cómodo el PHP, ya que los diseñadores/maquetadores pueden hacer el

HTML normal y luego los programadores meten el código PHP necesario. Se considera esto como una de las mayores ventajas del PHP, ya que para hacer sitios web necesitamos siempre de diseñadores.

- El PHP es relativamente nuevo y cada día salen módulos y extensiones (por ejemplo,

- PHP-GTK), está creciendo bastante y hay muchos ejemplos de código, quizás no tanto como el perl todavía, pero está creciendo mucho.

- EL PHP3 (a diferencia del PHP) tiene una muy buena gestión de *thrashing* (liberación de variables/datos que ya no son referenciados). Este era un problema en PHP con los accesos a la base de datos que retornan muchos datos. El PHP4 verifica que los datos ya no son referenciados y libera memoria inmediatamente.

- El Zend optimizer incluido en el PHP4 es muy rápido, e inclusive hay módulos adicionales para optimizar la compilación que es muy buena cuando los scripts son largos y complejos. Ahora también se puede instalar módulos de caché para evitar tener que interpretar cada vez el código fuente.

- La integración del PHP es muy buena con el Apache, por ejemplo, las cabeceras, campos de formularios o argumentos en los URLs, se pueden acceder directamente como variables globales, sin necesidad de interpretarlos. También permite trabajar con funciones POSIX tal como se hace en C sobre UNIX, por ejemplo: semáforos, memoria compartida, acceso a ficheros, funciones de tiempo, etc.

- Los módulos de acceso basado en datos para Postgres, MySQL, Oracle, ODBC genérico, DB2 e Interbase funcionan muy bien. En general no hay problemas con las interfaces a bases de datos.

- Por último,el esfuerzo de aprendizaje de PHP no es muy complicado e incluso menor que en otros lenguajes de programación.

En general la mayoría de programadores usan PHP para embeber código en HTML y si tuvieran que hacer cosas más complejas que no pudiera con el PHP, usarían el Perl.
Al ser PHP, un lenguaje de tipo Script que se lo puede embeber dentro de código HTML, un usuario común y corriente lo puede hacer funcionar en las siguientes plataformas:

- Windows 9x/NT/2000. En 9x necesitamos el Personal Web Server; en NT y 2000 se requiere el Internet Information Server.
- Linux con kernel 2.0 o superior.
- Solaris.
- Irix.
- AIX.
- La mayoría de los sabores de Unix.

En cada caso el procedimiento de instalación es diferente. Si se habla de Windows, necesitamos descompactar el ZIP, modificar el archivo de registros (Regedit) y personalizar el archivo de configuración de PHP (php.ini) para que logre funcionar. Si se trata de Linux, PHP ya viene como un módulo integrado en distribuciones como Red Hat 6.0.x, también será necesario personalizar el php.ini. En el caso de otros Unix es incluso posible que se requiera la recompilación del kernel para poderlo instalar, puesto que las descargas vienen en formato .TAR.GZIP, conteniendo el código fuente.

También, en caso de querer darle acceso a bases de datos, debe estar previamente instalada ésta (Oracle, PostgreSQL, MySQL) para que la encuentre y compile la información pertinente.

En el nivel más básico, PHP puede hacer cualquier cosa que cualquier otro programa del cgi puede hacer, por ejemplo recoger los datos de un formulario, genera el contenido dinámico de la página, o envía y recibe las galletas (cookies).

Lo más significativo de PHP es su soporte para una amplia gama de bases de datos. Escribir un Web page la conexión a Bases de Datos es increíblemente simple. Las bases de datos siguientes se apoyan actualmente: **Tabla 5.1.** Bases de datos.

Adabas D	InterBase	Sólido
dBase	mSQL	Sybase
Empress	MySQL	Velocis
FilePro	Oráculo	Dbm de Unix
Informix	PostgreSQL	

Tabla 5.1: Bases de datos.

PHP también tiene soporte para acceder a otros servicios usando protocolos tales como IMAP, SNMP, NNTP, POP3, o aún HTTP. Se puede también abrir los zócalos crudos de la red y obrar recíprocamente con otros protocolos.

5.4.4 Seguridad en PHP.

La seguridad en lo que respecta a accesos a los comandos del sistema está totalmente limitada a los privilegios del usuario que ejcuta el servidor apache (Se puede ver con que usuario se ejecuta mediante ps aux o en el httpd.conf). Así, si el script se encuentra en un directorio con permisos de escritura al usuario de apache, con php se podrán crear

ficheros y directorios. Además es importante indicar los siguientes puntos entre los más usados para dar seguridad en este lenguaje de programación:

- No pasan los parámetros con GET sino con POST.
- Para mayor seguridad, crear una variable de sesión de 32 bits que pase como parámetro obligatoriamente en cada petición, y se debe comprobar. Así se garantiza el seguimiento exacto del diagrama de flujo del programa.
- No colocan un interprete CGI-BIN de PHP.

Por otro lado .htaccess limita el acceso por clave, pero esto no tiene nada que ver con PHP. Es una autenticación que controla apache, no PHP.

Otra forma de limitar el acceso es colocando la información en páginas dinámicas que solo se exponen cuando los parámetros login y password son correctos al confrontarlos con una tabla de PHP o una tabla SQL.

Desde PHP se puede realizar llamadas al sistema con la función exec. Por ejemplo, exec('ping www.yahoo.com -c4', $array_output); realizaría un ping y devolvería el resultado en un array. Los privilegios de los que se dispone vienen restringidos por el usuario que ejecuta el servidor web en el servidor.

5.4.4.1.Cookies.

La principal utilidad de las cookies (galletas) es la de solventar el problema de la falta de estado en la navegación a través de las paginas web.

Con las cookies, pequeñas porciones de información se quedan registradas en el navegador permitiendo identificar a éste a través de diferentes páginas de un mismo sitio e incluso durante visitas entre distintos días.

Realmente las cookies no son más que cadenas de texto que son enviadas desde el servidor al cliente (navegador) y almacenadas en este, luego el navegador envía estas cookies al servidor permitiendo así la identificación del cliente en el servidor. La cookie es enviada al navegador desde el servidor y si este la acepta permanece en él.

Como se muestra en la **Fig.# 5.1 y 5.2**.

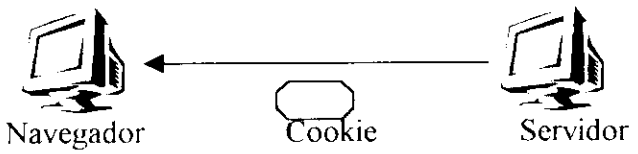


Fig.# 5.1: Las páginas piden la cookie al navegador

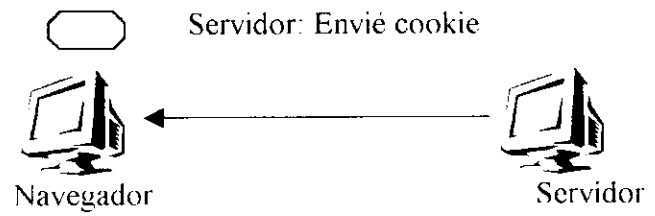


Fig.# 5.2: El navegador las envía, permitiendo la identificación del usuario por parte del servidor.

PHP en forma transparente se apoya en los cookies del HTTP. Los cookies son un mecanismo para almacenar datos en el browser alejado y así seguir o identificar a usuarios de vuelta. Se puede fijar los cookies usando la función de SetCookie(). Los cookies son parte del jefe del HTTP, así que la función de SetCookie debe ser llamada antes de que cualquier salida se envíe al browser. Ésta es la misma restricción que para la función de Header(). Cualquier cookie enviado del cliente será dada vuelta automáticamente en una variable de PHP justa como consigue y fija datos del método.

Si se desea asignar valores múltiples a un cookie, apenas se debe agregar / / al nombre del cookie. Por ejemplo:

SetCookie ("MyCookie[ ] ", "Probando", time()+3600);

Un cookie substituirá un cookie anterior por el mismo nombre en el browser a menos que la trayectoria o el dominio sea diferente

5.4.4.2. Posibles Ataques.

Una opción muy segura es poner el programa de análisis de PHP binario en alguna parte fuera del árbol de la tela de archivos. En /usr/local/bin, por ejemplo. El único downside verdadero a esta opción es que ahora se tendrá que poner una línea similar a:

#!/usr/local/bin/php

Como la primera línea de cualquier archivo que contiene PHP marca con etiqueta. También se necesitará hacer el archivo ejecutable. Es decir, tratar exactamente pues se trataria cualquier otra escritura del cgi escrita en el Perl o lenguajes scripting sh o cualquier otro shell en el que pueda ejecutarse.

Usar PHP como cgi binaria es una opción para las disposiciones que por una cierta razón no desean integrar PHP como módulo en software del servidor (como Apache), o utilizará PHP con diversas clases de envolturas del cgi para crear los ambientes seguros del chroot y del setuid para las escrituras. Incluso si el PHP binario se puede utilizar como intérprete independiente, PHP se diseña para prevenir los ataques.

Ficheros del sistema que tienen acceso: http://my.host/cgi-bin/php?/etc/passwd .

Cuando está invocado como cgi binario, PHP rechaza interpretar las discusiones de la línea de comando.

Generalmente algunos directorios de la configuración del web server se utiliza para volver a dirigir peticiones a los documentos comhttp://my.host/secret/script.php3 al intérprete de PHP. Con esta disposición, del web server se chequea primero los

permisos de acceso al directorio /al secreto. y ése crea después la petición vuelta a dirigir http://my.host/cgi-bin/php/secret/script.php3. Desafortunadamente, si la petición se da originalmente en esta forma, no se hace ningún chequeo del acceso por el web server para el archivo /secret/script.php3, sino solamente para el archivo / cgi-bin/php. De esta manera cualquier acceso capaz/cgi-bin/php del usuario puede tener acceso a cualquier documento protegido en el web server.

En PHP, la opción de tiempo de compilación de la configuración de los directorios se puede utilizar para prevenir este ataque, si el árbol del documento del servidor tiene cualesquiera directorios con restricciones del acceso.

Si un servidor no tiene ningún contenido que no sea restringido por contraseña o control de acceso basado en IP, no hay necesidad de estas opciones de la configuración. Se debe cerciorarse de que las escrituras de PHP no confíen en una u otra manera de llamar la escritura, ni por directamente http://my.host/cgi-bin/php/dir/script.php3 ni por el cambio de dirección http://my.host/dir/script.php3 .

El cambio de dirección se puede configurar por ejemplo en apache por los directorios AddHandler y acción.

5.4.4.3. Autenticación.

Esta opción de tiempo de compilación evita que cualquier persona llame PHP directamente con un URL como http://my.host/cgi-bin/php/secretdir/script.php3 . En lugar, PHP analizará solamente en este modo si ha pasado con un web server vuelve a dirigir.

El cambio de dirección en la configuración de Apache se hace generalmente con los directorios siguientes:

Acción php3-script/php3 de cgi-bin/php AddHandler php3-script

Esta opción se ha probado solamente con el web server de Apache, y confía en Apache para fijar la variable de entorno no estándar del cgi REDIRECT_STATUS en peticiones vueltas a dirigir. Si el web server no soporta ninguna manera, es decir si la petición es directa o vuelta a dirigir, no se puede utilizar esta opción y se debe utilizar una de las otras maneras de funcionar la versión del cgi.

Incluir el contenido activo, como las escrituras y los ejecutables, en los directorios de documento del web server es considera a veces una práctica insegura. Si para un cierto error de la configuración las escrituras no se ejecutan sino se exhiben como documentos generalmente del HTML, ésta puede dar lugar a salida de la información de la característica intelectual o de la seguridad como contraseñas. Por lo tanto muchos administradores preferirán el setting-up de otra estructura del directorio para las escrituras que sea solamente accesible a través del cgi de PHP, y por lo tanto interpretado siempre y no exhibido como tal.

También si el método para cerciorarse de las peticiones no se vuelve a dirigir según lo descrito en la sección anterior, no está disponible, es necesario instalar un doc_root de la escritura que sea diferente de raíz del documento de la tesis.

Se puede fijar la raiz del documento de la escritura de PHP por él doc_root directivo de la configuración en el archivo de php3.ini, o se puede fijar la variable de entorno PHP_DOCUMENT_ROOT. Si se fija, la versión del cgi de PHP construirá siempre el nombre del archivo para abrirse con este *doc_root* y la información de la trayectoria en la petición, así que se puede estar seguro que no se ejecuta ninguna escritura fuera de este directorio (a excepción *de user_dir*).

Otra opción utilizable aquí es user_dir. Cuando es el user_dir unset, sólo la cosa que controla el nombre del archivo abierto es *doc_root*. Abrir un URL como http://my.host/~user/doc.php3 no da lugar a abrir un archivo bajo directorio casero de los usuarios.

Si el user_dir se fija por ejemplo al public_php, una petición como http://my.host/~user/doc.php3 abrirá un archivo llamado doc.php3 bajo directorio

nombrado public php bajo directorio casero del usuario. Si el hogar del es/ home/user del usuario, es ejecutado archivo /home/user/public_php/doc.php3 .La extensión del user_dir sucede sin importar *el* ajuste del doc_root, así que se puede controlar el acceso del directorio de la raíz y de usuario del documento por separado.

5.5. Lenguaje de programación en PHP.

La ventaja que tiene PHP sobre otros lenguajes de programación que se ejecutan en el servidor (como podrían ser los script CGI Perl), es que nos permite intercalar las sentencias PHP en las paginas HTML.

Cumplidos los requisitos de instalación de la base de datos y creación del directorio virtual, se puede tratar la aplicación en forma local.

5.5.1. Variables.

Todos los nombres de variables comienzan con el signo ($) seguido de una letra o "_".

Las variables en PHP distinguen entre mayúsculas y minúsculas, por lo que $A y $a son variables distintas. Aquí se indica algunos ejemplos de variables:
$i
$contador
$nombre
$_TMP

No se tiene que declarar las variables explícitamente. PHP automáticamente las declara cuando se les asigna un valor por primera vez.
Los tipos de datos simples pueden ser enteros, coma flotante, y cadenas de caracteres.

*** Enteros**
$numero = 16;

*** Como flotante**

$pi = 3.1415;

*** Cadenas de caracteres**

$cadena = "Hola Mundo";

Los tipos de datos compuestos son vectores y objetos.

*** Vectores**

$var[0] = "Hola";

$var[1]= "Mundo";

Para añadir un elemento al final de la vector:

$var[]="Prueba";

Se puede acceder a los caracteres de las cadenas de forma separada, por ejemplo si:

$cadena = "Leo";

$cadena[3] ;Devuelve "o"

También se puede indexar los vectores usando cadenas de caracteres:

$mes["Enero"] =1;

$mes["Febrero"]=2;

5.5.2. Objetos.

Un objeto es un tipo de dato compuesto que contiene tanto variables como funciones.

Ejemplo:

```
class prueba {
    var $str = "Hola Mundo";
    function INIT($str) {
        $this->str = $str;
    }
}
$test = new prueba; / crea el Nuevo objeto $test de la clase prueba
print $test->str; / el resultado es "Hola Mundo"
```

```
$test->init("Hola");
print $test->str; / el resultado es "Hola"
```

La versión 3 de PHP no soporta herencia múltiple, protección de datos (o encapsulación) ni destructores, auque si implementa herencia y constructores. Si se tiene algunas nociones de programación orientada a objetos se puede aplicar parte de los conocimientos cuando se programe en este lenguaje.

5.5.3. Control de flujo.

Las estructuras de control en PHP son muy parecidas a las usadas por el lenguaje C. Se usan para controlar el flujo lógico de un programa PHP.

A continuación se enumera las distintas estructuras con su sintaxis. Su utilización es clara si se conoce algún otro lenguaje de programación.

*** If**

```
if (expresión) {
    código
}
elseif(expresión) {
    código
}
else {
    código
}
```

También puede expresarse de la forma:

```
If (expresion):
    código
elseif(expresión):
    código
else :
```

```
    código
endif;
* switch
switch(expresion) {
    case expresion:
        código
        break;
    default:
        código
        break;
}
```

También puede expresarse de esta forma:

```
switch(expresion):
    case expresion:
        código
        break;
    default:
        código
        break;
endswitch;
```

* **while**

```
while(expresion) {
    código
}
```

También puede expresarse de esta forma:

```
while (expresion):
    código
endwhile;
```

* **do/while**

```
do {
    código;
} while (expresión);
```

*** for**

```
for (expresión_de_inicio, expresión_de_condición; expresión_de_iteración) {
    código
}
```

También puede expresarse de esta forma:

```
for (expresión_de_inicio, expresión_de_condición; expresión_de_iteración) :
    código
endfor;
```

ejemplo:

```
for ( i=0; i<10; i++ )
{
  // código
}
```

5.5.4. Funciones.

Se puede crear las funciones de forma sencilla. Aquí se puede ver la sintaxis:

```
function testdesonido($a, $b, $c) {
    return "Probando, $a, $b, $c";
}
```

Así se crea la función testdesonido que acepta tres parámetros y se puede llamarla de esta forma:

```
echo testdesonido(1, 2, 3); / mostrará "Probando, 1, 2, 3"
```

Se puede crear funciones con parámetros opcionales si se le da un valor por defecto a cada parámetro opcional. La función anterior, si los tres parámetros son opcionales, resultaría:

```
function testdesonido($a=1, $b=2, $c=3) {
    return "Probando, $a, $b, $c";
}
```

5.5.5. Variables de la Web.

PHP crea variables de forma automática para toda la información que recibe en una petición HTTP. Esto incluye el método GET, POST, los cookies o las variables de estado. Existen dos formas:

La primera de ellas es cuando el fichero con PHP es el ejecutable de un formulario, por ejemplo:

```
<FORM ACTION="test.php3" METHOD="POST">
<INPUT TYPE=text NAME=var>
</FORM>
```

Cuando el formulario se envía al fichero test.php3, la variable $var dentro de ese fichero contendrá el valor que se introdujo en el campo de texto.

Otra forma de que la variable tome un valor, es llamar a un URL de esta forma:
http://localhost/test.php3?var=Hola+Mundo

Si se quiere ver una lista de todas las variables de estado, se puede incluir el siguiente código en una página y cargarla en el navegador:

```
<?php phpinfo() ?>
```

5.5.6. Incluir ficheros.

Una de las aplicaciones más importantes de PHP es la habilidad de incluir ficheros, los cuales pueden contener instrucciones PHP adicionales. Esto es útil porque permite tener el código común en un único archivo y que al modificarlo, afecte a todas las páginas.

Para incluir un fichero, se debe usar el comando keyword.

```
<?php
$titulo="Mi titulo";
include "cabecera.inc";
?>
```

"cabecera.inc" puede contener tanto HTML como código PHP. En el caso del ejemplo:

```
<HTML><HEAD>
<TITLE><?php $titulo?></TITLE>
</HEAD>
```

Lo que realiza PHP, es incluir el fichero antes de ejecutar el código PHP, por eso la variable $titulo estará disponible dentro de "cabecera.inc".

Por tanto el código anterior sería equivalente a :

```
<?php
$titulo="Mi titulo";
?>
<HTML><HEAD>
<TITLE><?php $titulo?></TITLE>
</HEAD>
```

5.5.6.1. Obtener la dirección IP y el navegador usado por el visitante.

Código

```
<HTML>
<BODY>
Usar el navegador <? echo $HTTP_USER_AGENT ?><BR>
y la dirección IP es <? echo $REMOTE_ADDR ?>
</BODY>
</HTML>
```

La función "echo", envía el texto al navegador. $HTTP_USER_AGENT es una variable web del servidor que contiene el navegador usado. $REMOTE_ADDR es otra variable web del servidor, pero esta contiene la dirección IP.

5.5.6.2. Enviar un e-mail.

En Unix o Linux, probablemente se tenga instalado el programa SendMail., es un programa para envío de correo. Lo ejecuta, se le pasa los parámetros del mensaje y el se encarga de enviarlo.

Hablar con el programa, es como escribir a un fichero. En vez de abrir un fichero, "popen()" abre una comunicación con el programa para que se pueda "enviar" mensajes o comandos. Esto se lo realiza con "puts()". Cuando se acaba la comunicación con el programa, la comunicación se cierra con la función "pclose()".

Puede usar la función "mail()" de PHP. Su sintaxis es la siguiente:
bool mail(string Destinario, string Titulo, string Cuerpo, string Cabeceras_Extra);

5.6. Conexión a Bases de Datos.

Cuando se pide una conexión persistente, PHP comprueba si hay ya una conexión persistente idéntica (que permanecía abierta desde antes) - y si existe, la usa. Si no existe, crea un enlace. Una conexión es una conexión que se abrió hacia el mismo "host", con el mismo nombre de usuario y la misma contraseña (donde sea aplicable).

La gente que no está familiarizada con el modo como trabajan y distribuyen la carga los servidores "web" puede confundir que persistente significa lo que no es. En particular, ellas *no* dan la habilidad de abrir 'sesiones de usuario' en el mismo enlace SQL, no dan la habilidad de construir una transacción de forma eficiente, y no hacen un montón de otras cosas. Las conexiones persistentes no dan ninguna funcionalidad que no fuera posible con sus hermanas no-persistentes.

Esto tiene que ver con el modo como funcionan los servidores "web". Hay tres modos en que un servidor "web" puede utilizar PHP para generar páginas web.

El primer método es usar PHP como una capa CGI. Cuando corre de este modo, se crea y destruye una instancia del intérprete PHP por cada página solicitada (para una página PHP) al servidor. Debido a que se destruye después de cada petición, cualquier recurso que adquiera (como un enlace a un servidor de base de datos) se cierra cuando es destruido. En este caso, no se gana nada si se intentan usar conexiones persistentes.

El segundo, y más popular, método es correr PHP como un módulo en un servidor web multiproceso, lo cual actualmente sólo incluye Apache. Un servidor multiproceso tiene tipicamente un proceso (el padre) que coordina un conjunto de procesos (sus hijos) que realmente hacen el trabajo de servir las páginas web. Cuando entra cada petición de un cliente, es entregada a uno de los hijos que no esté ya sirviendo a otro cliente. Esto significa que cuando el mismo cliente hace una segunda petición al servidor, puede ser atendido por un proceso hijo distinto al de la primera vez. Lo que una conexión persistente hace en este caso es hacerlo de tal modo que cada proceso hijo sólo necesita conectar a la base de datos la primera vez que sirve una página que hace uso de una conexión asi. Cuando otra página solicita una conexión a la base de datos, puede reutilizar la conexión que el hijo estableció previamente.

El último método es usar PHP como un "plug-in" para un servidor web multihilo. En la actualidad es solamente teórico, PHP no funciona aún como "plug-in" para ningún servidor web multihilo. Hay trabajo en progreso para soportar ISAPI, WSAPI y NSAPI (en Windows), lo cual permitirá a PHP ser utilizado como "plug-in" para servidores web multihilo como Netscape FastTrack, Internet Information Server (IIS) de Microsoft, y O'Reilly's WebSite Pro. Cuando esto ocurra, el comportamiento será exactamente el mismo que para el modelo de multiprocesador descrito anteriormente.

Una vez que se tiene creada la base de datos en el servidor, el siguiente paso es conectarse a la misma desde una página PHP.

```
<!-- ejemplo -->
<html>
```

```
<head>
  <title>Ejemplo de PHP</title>
</head>
<body>
<?php
function Conectarse()
{
  if (!($link=mysql_connect("localhost","usuario","Password")))
  {
    echo "Error conectando a la base de datos.";
    exit();
  }
  if (!mysql_select_db("base_datos",$link))
  {
    echo "Error seleccionando la base de datos.";
    exit();
  }
  return $link;
}

$link = Conectarse();
echo "Conexión con la base de datos conseguida.<br>";
mysql_close($link); //cierra la conexion
?>
</body>
</html>
```

Al ejecutar la instrucción mysql_connect creamos un vínculo entre la base de datos y la pagina PHP, este vínculo será usado posteriormente en las consultas que se haga a la base de datos.

Finalmente, una vez que se termina de usar el vínculo con la base de datos, se libera con la instrucción mysql_close para que la conexión no quede ocupada.

Una vez que se conecta con el servidor de bases de datos, ya se puede realizar consultas a las tablas de la base de datos.

```
<!-- ejemplo.com -->
<html>
<head>
  <title>Ejemplo de PHP</title>
</head>
<body>
<H1>Ejemplo de uso de bases de datos con PHP y MySQL</H1>
<?php
  include("conex.phtml");
  $link=Conectarse();
  $result  mysql_query("select * from prueba",$link);
?>
  <TABLE BORDER=1 CELLSPACING=1 CELLPADDING=1>
    <TR><TD> Nombre</TD><TD> Apellidos </TD></TR>
<?php

  while($row = mysql_fetch_array($result)) {
    printf("<tr><td> %s</td><td> %s </td></tr>",
$row["Nombre"],$row["Apellidos"]);
  }
  mysql_free_result($result);
  mysql_close($link);
?>
</table>
</body>
</html>
```

En este ejemplo se utilizó 3 instrucciones nuevas: mysql_query, mysql_fetch_array y mysql_free_result. Con la instrucción mysql_query se hizo una consulta a la base de datos en el lenguaje de consultas SQL, con la instrucción mysql_fetch_array extraemos

los datos de la consulta a un array y con mysql _free result liberamos la memoria usada en la consulta.

El lenguaje PHP proporciona una manera sencilla de manejar formularios, permitiendo de esta manera procesar la información que el usuario ha introducido.

Al diseñar un formulario se debe indicar la página PHP que procesará el formulario, así como el método por el que se le pasará la información a la página.

```
<!-- Ejemplo.com -->
<html>
<head>
  <title>Ejemplo de PHP</title>
</head>
<body>
<H1>Ejemplo de procesado de formularios</H1>
Introduzca su nombre:
<FORM ACTION="procesa.phtml" METHOD="GET">
<INPUT TYPE="text" NAME="nombre"><BR>
<INPUT TYPE="submit" VALUE="Enviar">
</FORM>
</body>
</html>
```

Al pulsar el botón Enviar el contenido de cuadro de texto es enviado a la página que indica el atributo ACTION de la etiqueta FORM.

PHP crea una variable por cada elemento del FORM, esta variable creada tiene el mismo nombre que el cuadro de texto de la página anterior y el valor que se haya introducido. En este ejemplo se ha creado una variable llamada $nombre con el valor que haya introducido el navegante.

CAPITULO VI.

6. DESARROLLO E IMPLENTACION DEL WEBMAIL EN LA PUCESA

6.1. Descripción total del proyecto.

Luego de haber realizado las investigaciones necesarias se llega a la parte más importante que es la implantación del sistema, el mismo que acompaña a la documentación de la investigación, puesto que aquí se definen las experiencias que se han adquirido en el desarrollo de este sistema cuyo objetivo es el de poseer un sistema completo funcionando, que servirá para el envió y recepción de Mails a través de un WebMail propio de todos quienes integran la Pontificia Universidad Católica del Ecuador Sede Ambato.

A continuación hablaremos sobre la metodología que se ha utilizado para la elaboración del sistema.

6.2. Metodología para la Elaboración del Sistema.

La metodología utilizada para el desarrollo del WebMail de la PUCESA fue la de CASCADA, en la que se sigue el ciclo de vida del software, que a continuación se describe.

1) **Proceso de Pre-Desarrollo:** En esta Etapa se realizó la recopilación de los datos en el laboratorio de la Universidad y de manera especial se hizo un estudio profundo del WebMail con el que cuenta nuestra Institución, detectando algunas opciones por incrementarse para el mejor funcionamiento del mismo, de esta manera se optó por este proyecto como una alternativa más completa.

2) **Escogitamiento de las Herramientas de Desarrollo:** Antes de realizar el desarrollo del sistema se estudió algunos posibles lenguajes en los cuales se podría realizar este proyecto, se optó por el que mayor facilidad presentaba, fácil manejo y de fácil entendimiento, tanto para el desarrollo como también para el uso del sistema.

3) **Proceso de Desarrollo:** En esta Etapa se toma en cuenta el objetivo de la realización del proyecto, los datos que se van a utilizar, la estructuración de los mismos, desarrollo del sistema en si, implantación del proyecto y la interfaz a utilizarse. Esta etapa está basada en algunos subprocesos:

   a) **Requerimientos del Sistema:** En esta fase se indican las características con las que debe contar el sistema para su correcto funcionamiento y cumplimiento de las necesidades de nuestra Institución.

   b) **Análisis y Diseño del Sistema:** En fase se estudia de manera especial las necesidades de la institución para la cual se va a realizar el sistema, y por otra parte el diseño de la estructura del sistema.

   c) **Implementación del Sistema:** Aquí se realiza un estudio de los cambios que se realizan tanto en el hardware como en el software del servidor en donde va a funcionar el sistema, puesto que la programación no se realizó en el mismo y para su mejor funcionamiento requiere de algunos cambios, como las versiones de paquetes ya existentes.

   d) **Actualización del Sistema:** En esta fase se recogen todos los problemas encontrados en la implementación y prueba del sistema, con el fin de corregirlos.

6.3. Proceso De Pre-Desarrollo.

Antes de la realización del sistema para la Creación de un WebMail para la PUCESA, se realizaron algunas entrevista y encuestas al Administrador del Centro de Cómputo (Ver anexos). En donde se recabó información de los laboratorios de la Universidad, funcionamiento, distribución de red, etc.

Un resumen de los resultados procesados determinó que el centro de Cómputo de una Institución es una pieza fundamental de soporte, para hacer que determinada institución llegue a cumplir sus objetivos. En la PUCESA, la misión del Centro de Computo es la de fortalecer y mantener los servicios de computación para satisfacer las necesidades de información y equipamiento de clientes internos y convertirse al mismo tiempo en un ente de apoyo a la docencia.

El trabajo continuo y conjunto del personal encargado del centro de cómputo, estudiantado, extensionistas y practicantes han permitido que actualmente la PUCESA por medio de su Centro de cómputo brinde un sin número de servicios, entre los principales se mencionan:

1. Laboratorio IBM con aproximadamente 12 Pcs.
2. Laboratorio Compaq con aproximadamente 10 Pcs.
3. Servicio de Hosting para páginas web.
4. Proveedor de Servicios Internet para estudiantes y personal de la PUCESA.
5. Correo electrónico para estudiantes, personal administrativo y docente.
6. Biblioteca Virtual.
7. Mantenimiento de los Equipos.

El Centro de Cómputo de la PUCESA es usado para todas las labores que realiza el encargado del mismo, entre ellas monitorear a los laboratorios de la Universidad, aquí se encuentra el servidor de la red, usado para las conexiones a Internet entre otras labores, los equipos existentes son:

- Computador SUN Microsystem 250 y Procesador Ultra Spark 400 Mhz, 256 RAM, CD-ROM, 180 Gb de DDH y Tape backup, todo de tecnología SCSI.

- Computador CLON AMD 1.26, 256 RAM, disco 306 IFdd, 1 CD- ROM, video sonido, fax integrado, teclado, mouse tipo PS2, tarjeta de red, 2 puertos Usb.

- SWITCH Centre Com 3726XL, 10 BaseT, 1 puerto Fast Ethernet.

- HUB Centre Com 3624 TRS, 24 Puertos, 10 Base T.

El sistema operativo que se usa en el centro de computo de la PUCESA mostrado en la **Tabla 6.1.**

Paquete	Versión	Fabricante
Sun Solaris	2.6	Sun Micro system
Linux Red Hat	6.2 4.0	GNU Microsoft
Windows NT		

Tabla 6.1: Sistemas Operativos usados en el Centro de Computo.

El laboratorio IBM conocido así por la marca de las computadoras existentes en el mismo, es usado por los estudiantes y profesores para recibir y dictar clases, también se puede hacer uso de Internet, en horas de clases o fuera de ellas, previa la autorización

correspondiente. A continuación indicaremos las características de las máquinas con las que cuenta este laboratorio:

- Computadores IBM Intel Celeron 200Mhz, 32 RAM, Dims 72 pines, Disco duro Quantum 2.16 Gb, video, sonido incorporado, mainboard Intel, CD-ROM, floppy.

- HUB Centre Com 3624 TRS, 24 Puertos, 10 Base T.

En el laboratorio IBM se usa el siguiente Sistema Operativo como se indica en la **Tabla 6.2.**

Paquete	Versión	Fabricante
Windows 98	98	Microsoft

Tabla 6.2: Sistema Operativo utilizado en el Laboratorio IBM.

El Laboratorio Compaq llamado así por la marca de los equipos existentes, tiene un uso similar al Laboratorio IBM, a continuación se detalla los equipos existentes con sus respectivas características:

- Computador Compaq Procesadores Intel Pentium I, 16 RAM, video Sonido Incorporado, CD-ROM 4x, Disco duros de 1.2 Gb, tarjeta de red de 10 Mpbs, drive, monitor, teclado.

- HUB Centre Com 3624 TRS, 24 Puertos, 10 Base T.

El sistema operativo utilizado en el laboratorio Compaq es el que se indica en la **Tabla 6.3**

Paquete	Versión	Fabricante
Windows 98	98	Microsoft

Tabla 6.3: Sistema Operativo utilizado en el Laboratorio Compaq.

Es importante indicar que estos equipos de cada uno de los laboratorios están conectados en red usando la tipología estrella.

Como se mencionó anteriormente la Universidad presta el servicio de Internet a los estudiantes, docentes y administrativos de la misma, es por eso que se vio la necesidad de realizar un proyecto para la creación de un WebMail propio de la Universidad, de fácil uso y sobre todo incrementando opciones que satisfagan las necesidades de la institución.

6.4. Escogitamiento de las herramientas de desarrollo.

El principal Objetivo de este proyecto fue la realización de un WebMail para la PUCESA, para esto se realizaron algunas investigaciones del tipo de programación que se usa para ello, encontramos varios como:

- HTML, Hyper Text Markup Language (Lenguaje de marcación de Hipertexto) es el lenguaje de marcas de texto utilizado normalmente en la www (World Wide Web). El entorno para trabajar HTML es simplemente un procesador de texto, como el que ofrecen los sistemas operativos Windows (Bloc de notas), UNIX (el editor vi o ed) o el que ofrece MS Office (Word). El conjunto de etiquetas que se creen, se deben guardar con la extensión .htm o .html

- El JSP (Java Servlet Pages) Si bien su uso se destaca en el Web, sirve para crear todo tipo de aplicaciones (locales, intranet o internet).

- El CGI (*Common Gateway Interface*) es un estándar para comunicar aplicaciones externas con los servidores de información, tales como servidores HTTP o *Web*.

Luego de realizar un análisis optamos por la plataforma Linux y el Lenguaje de programación PHP4, por la facilidad con la que maneja él envió y recepción de datos, por otra la obtención del mismo no tiene ningún costo y es recomendado para el manejo de multiusuarios y multitareas. Tiene algunas distribuciones, se escogió Red Hat versión 7.1 puesto que la Universidad posee su servidor de correo electrónico con esta plataforma. Sus principales características son:

- Posee un conjunto de aplicaciones útiles para llevar acabo cualquier tarea que podría necesitarse, procesador de textos, utilidades para gráficos, agenda y muchas otras que vienen añadidas cuando se obtiene el CD.

- La Instalación de Red Hat es muy sencilla y puede realizarse de manera rápida con una pequeña preparación avanzada. En una computadora moderna con una unidad de CD- ROM, la instalación puede tardarse entre 15 y 20 minutos desde el inicio hasta su finalización, mientras que otros métodos de instalación, como los de las redes FTP o HTTP, pueden tardar más tiempo debido a la necesidad de transferir información a través de la red.

- Utiliza además Apache que es el servidor web que viene predeterminado y que es el que utilizaremos en nuestra aplicación. Actualmente Apache se está convirtiendo en el servidor web más popular de Internet, y funciona en muchas arquitecturas, tales como, Linux, Solaris, SunOS4.x y Windows NT. Linux es una de las plataformas más usadas para ejecutar Apache debido a su estabilidad y bajo coste

- El proyecto Apache, al igual que Linux, es una historia más de éxito del software gratuito, su objetivo es crear un servidor web completo, robusto y capaz de competir con cualquier producto comercial. Hace años que mucha

gente contribuye con código, ideas y dinero al proyecto Apache, haciendo posible su éxito.

- Apache es el servidor de web por excelencia, una de las principales características de apache es su extensibilidad basada en una gran modularidad de código fuente, lo que ha facilitado la aparición de módulos de extensión como PHP, facilitando enormemente la programación de aplicaciones en el lado del servidor.

Más del 50% de los servidores existentes para la publicación de sitios web tienen instalado servidor web apache (según el libro Administración de Red Hat Linux de Thomas Schenk), existen versiones tanto para Linux como para Windows, mediante algunas investigaciones realizadas se ha demostrado que el comportamiento es adecuado tanto para los diferentes tipos de aplicaciones como para sistemas con bases de datos o motores independientes.

Las propiedades y características del servidor Apache son muy amplias por lo que es muy usado en Internet.

- Apache es uno de los primeros servidores que trabajo con el protocolo http, por lo que está preparado para trabajar con el nuevo estándar y las versiones anteriores.

- Puede trabajar con direcciones IP y con nombres virtuales también admite servidores virtuales.

- Es posible transformar el Servidor apache en un servidor Proxy.

- Permite el trabajo con páginas SSL (Secure Socket Layer, Capa de Socket Seguro), gracias al módulo mod_include mediante este módulo se desarrollan las páginas HTML.

- Es flexible a la hora de registrar y visualizar el estado del servidor, ya que integra registros personales e información sobre el estado del servidor.

- Se puede trabajar en modo multitarea gracias al módulo mod_jserv, por medio de él se ejecuta aplicaciones en Java propias del servidor.

- Tiene tres archivos de configuración, en texto plano, que sirven para establecer los parámetros del servidor. Si no dispone de Interfaz gráfica no es un problema puesto que con tener un editor de texto es suficiente para llevar a cabo la configuración.

- Utiliza el módulo mod-cgi con lo que hace posible la personalización de las variables de entorno y búsqueda de errores.

Para la programación misma se utilizó el paquete PHP4, puesto que el lenguaje PHP4 es un lenguaje de programación de estilo clásico, es decir que es un lenguaje de programación con variables, sentencias condicionales, bucles y funciones. No es un lenguaje de marcas como podría ser HTML, XML o WML. Está más cercano a JavaScript o a C, para aquellos que conocen estos lenguajes.

PHP puede ser embebido dentro de una página HTML ya terminada sin ningún problema. Esto significa que hay una clara separación de código y datos. Desde el punto de vista del trabajo "cooperativo" con los diseñadores, es mucho mas cómodo el PHP, ya que los diseñadores/maquetadores pueden hacer el HTML normal y luego los programadores meten el código PHP necesario. Es considerado como una de las mayores ventajas del PHP, ya que para hacer sitios web se necesita siempre de diseñadores.

Pero a diferencia de Java o JavaScript que se ejecutan en el navegador, PHP se ejecuta en el servidor, por eso nos permite acceder a los recursos que tenga el servidor como por ejemplo una base de datos. El programa PHP es ejecutado en el servidor y el resultado enviado al navegador. El resultado es normalmente una página HTML.

Al ser PHP un lenguaje que se ejecuta en el servidor no es necesario que su navegador lo soporte, es independiente del navegador, pero sin embargo para que sus páginas PHP funcionen, el servidor donde están alojadas debe soportar PHP.

El control interno del correo se basa en Sendmail, es decir utiliza su sistema de control de datos. Es importante indicar que se trabajó con el agente Sendmail que se traslada de un host a otro buscando correos ruteados. Existen otros agentes de transferencia de correo (que pueden ser utilizados con Red Hat Linux), pero la mayoría de administradores eligen utilizar Sendmail debido a su potencia, escalabilidad y compatibilidad con los estándares de Internet.

El deber principal de Sendmail, como otros MTAs (Mail Transport Agent, Agente de Transporte de correo), es el de trasladar con seguridad e-mails entre hosts, habitualmente usando el Simple Mail Transfer Protocol (SMTP). No obstante, Sendmail es altamente configurable, permitiendo controlar cada aspecto de cómo se gestiona un correo. Es por estas razones que nosotros utilizamos este agente en nuestra aplicación.

6.5. Proceso De Desarrollo.

Dentro de esta etapa se analizaron las partes importantes que intervienen para el desarrollo del nuevo sistema tales como el flujo de la información, el requerimiento del sistema, análisis y diseño y la implantación del nuevo software. A continuación se detallará cada una de estas subetapas.

6.5.1. Flujo De La Información.

El flujo de La Información se realiza tomando en cuenta los requerimientos del nuevo sistema a utilizarse, es importante, puesto que permite realizar la correcta instalación, configuración y uso del Web Mail para la PUCESA.

inmediatamente de la conexión, pero si el acceso es permitido, prácticamente ingresa al sistema. el mismo que le pide su nombre y contraseña para el funcionamiento, si los datos han sido correctos, el sistema está listo para ser utilizado, presenta algunas opciones:

Permite revisar el correo, crear nuevo mensaje, enviar, presenta opciones para señalar si los correos han sido leidos o no, carpetas o buzones tanto para basura si desea borrar los correos, carpeta de leidos, no leidos, carpeta de pendientes, carpeta que contiene correos pendientes para ser enviados, finalmente presenta una página de ayuda en la que se explicará brevemente el uso del Web Mail.

Si el usuario ya no desea realizar otra actividad dentro del sistema, prácticamente se cierra el acceso y la conexión al servidor de esta manera se termina la tarea.

6.5.2. Requerimientos Del Sistema.

Los Requerimientos básicos para la Creación de un WebMail para la PUCESA son los siguientes:

- El sistema operativo abierto GNU/Linux Distribución Red Hat 7.1 para Intel(tipo de Procesador).
- Apache versión 2.4.2-2. como servidor de Web.
- PHP versión 4.0. para la programación del sistema.
- Configuración de direcciones IP y dominios
- Servicios POP(Protocolos de Oficina de correos) e IMAP(Protocolo de acceso interactivo al Correo) .

Como ya se indicó anteriormente todas las herramientas enunciadas se las puede obtener sin ningún coste puesto que la distribución GNU es de distribución, por lo que la elaboración del sistema, implantación y el uso del mismo conllevan a una inversión mínima.

6.5.3. Análisis y Diseño.

A continuación indicaremos el diseño del Web Mail, los métodos que utiliza, las funciones realizadas para cada acción, se indicará en forma general lo que contiene cada una de las pantallas del sistema.

1. login	
Constructor	Login()
Métodos	Descripción
public conectar(string usuario, clave)	Método para conectar al servidor de correo HOLAMAIL
public estado (int conecto)	Valor de retorno, si no conecta es falso para indicar que no corresponde datos enviados

Tabla 6.4: Descripción Función Login

2. Componer	
Constructor	Crear_Mensaje()
Métodos	Descripción
public void init()	Inicializa variables y objetos de la interfaz
private void CargaInterfaz()	Método para activar la interfaz para ingresar contenido de correo
Private void Prioridad	Método para seleccionar la forma en que el correo será gestionado por el servidor Mail
Private int Enviar	Método para enviar el contenido del mensaje y sus adjuntos
Private void Examinar	Método para buscar archivos adjuntos en disco
Private void Agregar	Método para indicar que un archivo se adjunta al mensaje a enviar

Tabla 6.5: Descripción Función Componer.

3. Visualizar	
Constructor	Visualizar_Mensaje()
Métodos	Descripción
public void init()	Inicializa variables y objetos de la interfaz
private void CargaInterfaz()	Método para activar la interfaz para visualizar contenido de mensaje
Private void Reenviar	Método de reenvío de mensaje, éste activa la clase componer
Private void Responder	Método para responder el mensaje al usuario original, activa clase componer
Private void ResponderaTodos	Método responder a todos los usuarios que enviaron mensajes
Private void Borrar	Método para eliminar mensaje de correo

Tabla 6.6: Descripción Función Visualizar

4. Entrada	
Constructor	Buzón Entrada ()
Métodos	Descripción
public void init()	Inicializa variables y objetos de la interfaz
private void CargaInterfaz()	Método para activar la interfaz para visualizar contenido de mensaje
Private void Mostrar	Método para mostrar la lista de mensajes recibidos
Private void Leido	Método para cambiar el atributo del mensaje a leído
Private void NoLeido	Método para cambiar atributo a no leído
Private void Borrar	Método para eliminar mensaje

Tabla 6.7: Descripción de la función Entrada

5. Basura	
Constructor	Buzón basura ()
Métodos	Descripción
public void init()	Inicializa variables y objetos de la interfaz
private void CargaInterfaz()	Método para activar la interfaz para visualizar contenido de mensaje
Private void Mover	Método para mover la lista de mensajes hacia otra carpeta
Private void Leido	Método para cambiar el atributo del mensaje a leído
Private void NoLeido	Método para cambiar atributo a no leído
Private void Borrar	Método para eliminar mensaje que indicado.

Tabla 6.8: Descripción de la función del Buzón de Basura.

6. Pendientes	
Constructor	Buzón pendientes ()
Métodos	Descripción
public void init()	Inicializa variables y objetos de la interfaz
private void CargaInterfaz()	Método para activar la interfaz para visualizar contenido de mensaje
Private void Mover	Método para mostrar la lista de mensajes que van a ser cambiados o llevados a otra carpeta
Private void Leido	Método para cambiar el atributo del mensaje a leído
Private void NoLeido	Método para cambiar atributo a no leído
Private void Borrar	Método para eliminar mensaje

Tabla 6.9: Descripción de la función del Buzón de Pendientes.

7.Salida	
Constructor	Buzón Salida ()
Métodos	Descripción
public void init()	Inicializa variables y objetos de la interfaz
private void CargaInterfaz()	Método para activar la interfaz para visualizar contenido de mensaje
Private void Mover	Método para indicar los mensajes que serán trasladados a otra carpeta.
Private void Leido	Método para cambiar el atributo del mensaje a leído
Private void NoLeido	Método para cambiar atributo a no leído
Private void Borrar	Método para eliminar mensaje

Tabla 6.10: Descripción de la función del Buzón de Salida.

8. direcciones	
Constructor	Control Direcciones ()
Métodos	Descripción
public void init()	Inicializa variables y objetos de la interfaz
private void CargaInterfaz()	Método para activar la interfaz para visualizar contenido de mensaje
Private void Añadir Dirección	Método para añadir los datos a la lista de direcciones
Private void Editar Seleccionado	Método para modificar datos de dirección
Private void Borrar Seleccionado	Método para que elimina una dirección selección

Tabla 6.11: Descripción clase Direcciones

9. opciones	
Constructor	ConfigurarOpciones ()
Métodos	Descripción
public void init()	Inicializa variables y objetos de la interfaz
private void CargaInterfaz()	Método para activar la interfaz para modificar opciones personales de usuario
Private void Información Personal	Método para actualizar información de usuario
Private void Resaltado de mensajes	Método para resaltar mensajes
Private void Orden de Indice	Método para indicar la ubicación de las columnas de un mensaje
Private void Preferencias	Método para cambiar el aspecto y la forma de mostrar información
Private void Preferencias Carpetas	Método para cambiar la forma en la que se muestra y manipula las carpetas

Tabla 6.12: Descripción de la función de Opciones.

10. Buscar	
Constructor	Control Buscar ()
Métodos	Descripción
public void init()	Inicializa variables y objetos de la interfaz
private void Buscar	Método para buscar un correo ya sea por criterio, asunto, bcc o por el cuerpo.

Tabla 6.13: Descripción de la función Buscar.

11. Ayuda	
Constructor	Control Ayuda ()
Métodos	Descripción
public void init()	Inicializa variables y objetos de la interfaz
private void CargaInterfaz()	Método para activar la interfaz para visualizar contenido de mensaje

Tabla 6.14: Descripción de la función Ayuda

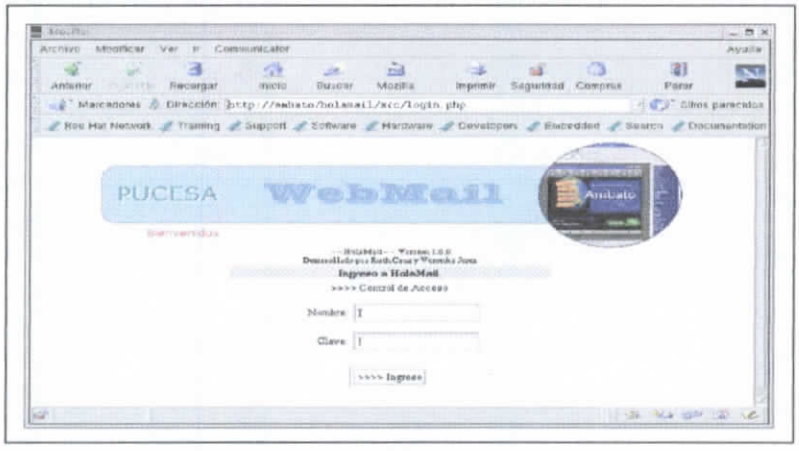
6.5.4 Diseño de Interfaz..

En esta sección se indica cada una de las pantallas que conforman el sistema para el manejo de un WebMail para la Pontificia Universidad Católica del Ecuador Sede Ambato.



Fig# 6.2: Pantalla de Ingreso para la Conexión al WebMail.

En la Fig#6.2. se indica una pantalla con una breve publicidad de la Universidad, además consta de un botón para el ingreso al sistema, está ubicado en la parte derecha de la pantalla, al dar clic sobre el botón comenzamos a dar uso al sistema.



Fig# 6.3: Pantalla de Ingreso al Sistema

En la Fig.# 6.3. se indica la pantalla en la cual pide que el usuario ingrese el nombre y su clave, tiene datos de los autores del sistema, versión del mismo, al dar clic sobre el botón ingresar, prácticamente el usuario está dentro del WebMail.



Fig.# 6.4: Pantalla que indica la carpeta de Entrada.

En la Fig.# 6.4. se puede observar todos los mensajes recibidos por el usuario, los mensajes que aún no han sido leídos se mostrará con negritas, se tiene la opción de mover el mensaje para otra carpeta que el usuario puede escoger, le indica además el origen del mensaje, la fecha que le han enviado, y el asunto. El sistema le indica cuántos mensajes tiene en ese momento, además se tiene las opciones de borrar el mensaje, señalarlo como leído o como no leído.

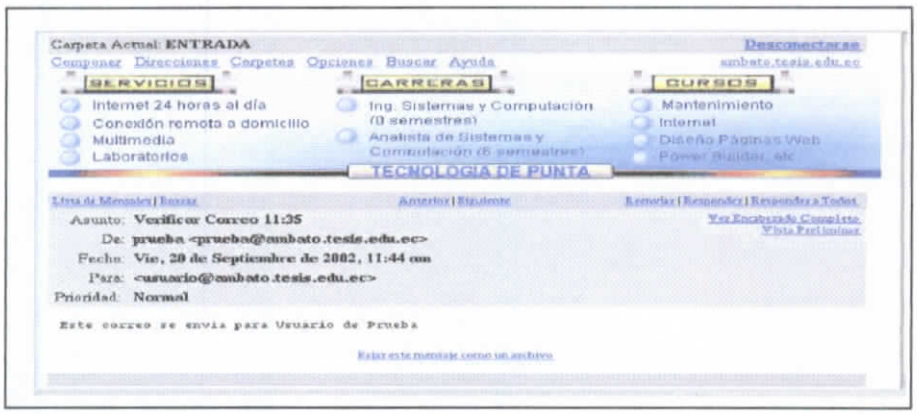


Fig.# 6.5: Pantalla de lectura de un mensaje.

En la Fig.# 6.5. se muestra una pantalla en la cual el mensaje es abierto por el usuario, le indica el asunto del envió, quién lo envía, la fecha, presenta otras opciones como el de reenviar, responder, responder a todos. Se puede leer el cuerpo mismo de mensaje, además presenta otras opciones como borrar el mensaje, ir al mensaje anterior o al siguiente, cada una de estas opciones funcionan al dar clic sobre cada uno de los botones que nos presenta la pantalla.

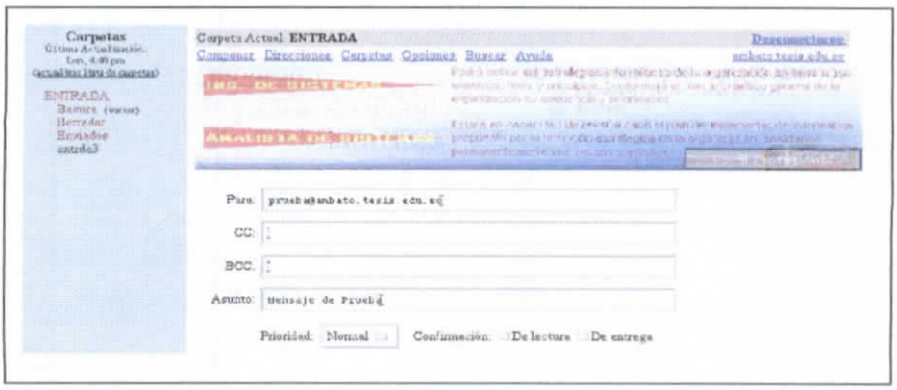


Fig.# 6.6: Pantalla para responder el mensaje.

En la Fig.# 6.6. se muestra la pantalla para que un usuario pueda responder un mensaje o simplemente para escribir un mensaje a otra persona, debe indicar para quien va el mensaje, el asunto, si desea una copia del mismo y para quien va dirigida dicha copia, luego escribirá el texto del cuerpo del mensaje, a continuación existen algunos botones para el envío, si existe un documento adjunto, si desea la confirmación del envió, ya sea de lectura o de entrega. Esta es la misma pantalla que se presenta si damos clic en el botón componer del menú principal o Responder a uno si estamos dentro del mensaje.

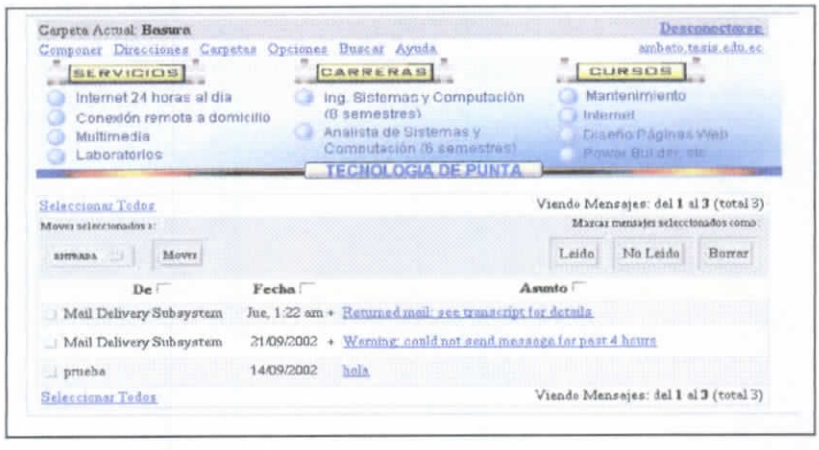


Fig.# 6.7: Pantalla que muestra el contenido de la Carpeta Basura.

En la Fig.# 6.7. se presenta la pantalla sobre la Carpeta de basura, en la misma que están los mensajes que han sido borrados o enviados a esta carpeta por el usuario, cada uno con su encabezado, es decir Asunto, Fecha, Quién lo envía, presenta las opciones de mover el mensaje a otra carpeta, Borrarlo definitivamente, nos indica además el número de mensajes que se encuentran en la carpeta.

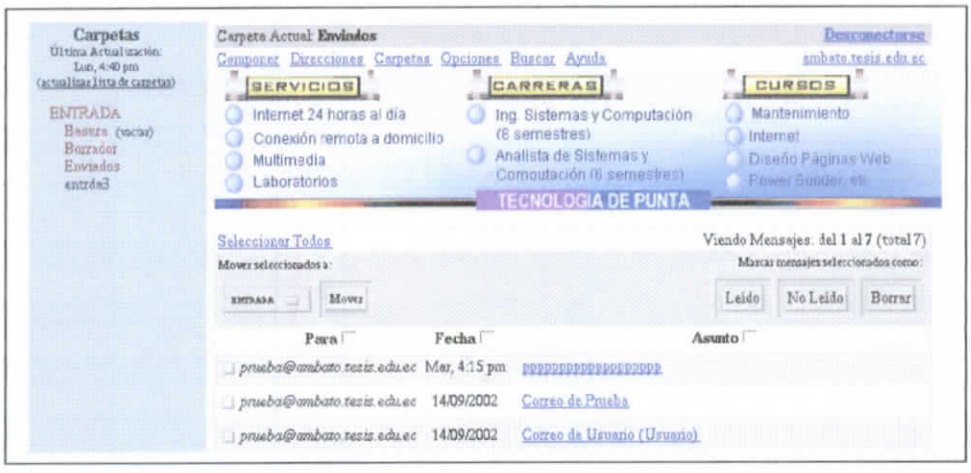


Fig.# 6.8: Pantalla que indica carpeta de Mensajes Enviados.

En la Fig.# 6.8. se muestra la carpeta de mensajes enviados, nos indica el usuario para quien ha sido enviado el mensaje, el asunto y la fecha de envío de cada uno de los mensajes, estos mensajes se pueden mover hacia otras carpetas escogiendo en el botón de la parte superior, presenta además las opciones de borrado, leído o no leído, indica la cantidad de mensajes que tiene dicha carpeta.

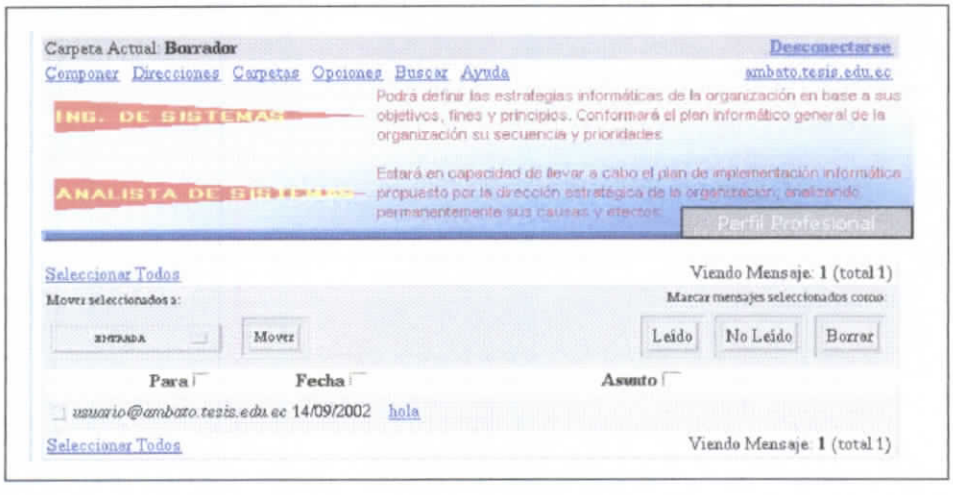


Fig.# 6.9: Pantalla que indica la Carpeta Borrador.

En la Fig.# 6.9. se presentan los mensajes que se encuentran en la carpeta de borrador, cada uno de ellos tiene la fecha en la que han ingresado a esta carpeta, el asunto, e indica para quien fue enviado el mensaje, como en las carpetas anteriores presenta la opción de mover hacia otra carpeta, borrar, indica además la cantidad de mensajes almacenados en esta carpeta.

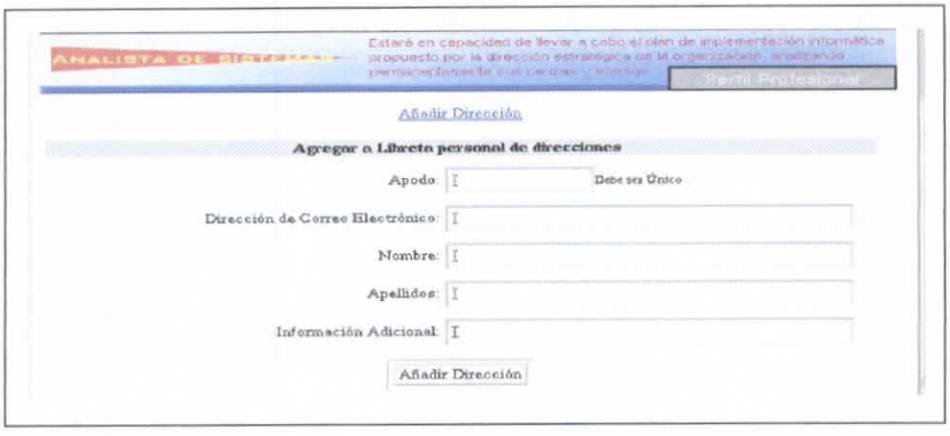


Fig.#6.10: Pantalla para Agregar a libreta de direcciones.

En la fig.#6.10. se presenta una pantalla en la que se ingresan datos de una nueva persona en la libreta de direcciones. Es una característica ideal para ahorrar tiempo, las direcciones que se utilizan con frecuencia pueden ser almacenadas aquí.

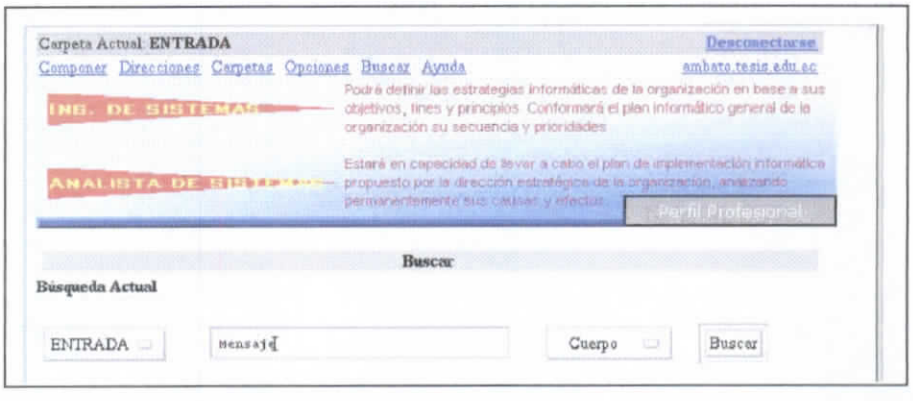


Fig.# 6.11: Pantalla que indica la Opción Buscar.

En la Fig.# 6.11 se muestra la pantalla que contiene la opción Buscar del menú principal, se puede ingresar cualquier palabra que será buscada en los mensajes que

tenga el usuario, puede escoger la carpeta en la que va a realizar la búsqueda, la parte del mensaje ya sea el encabezado, el cuerpo del mensaje o en todo el mensaje, al dar clic sobre el botón Buscar inicia la búsqueda para dar el resultado que el usuario espera.

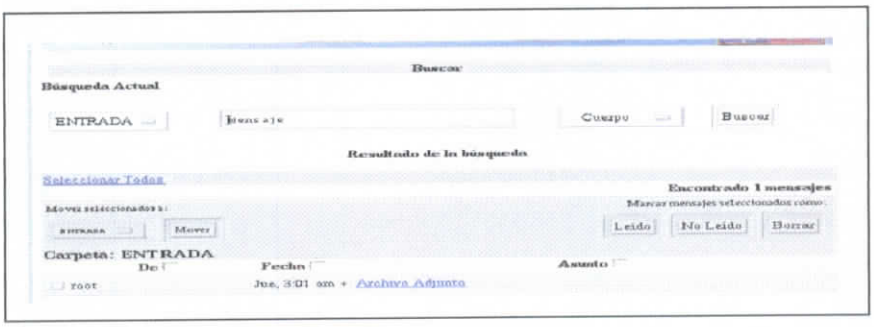


Fig.# 6.12. Pantalla que indica el Resultado de la Búsqueda.

Como se muestra en la Fig.# 6.12 el sistema le indica en qué mensaje se ha podido encontrar la palabra que el usuario mando a buscar, ya sea en el cuerpo o en el encabezado, dependiendo de lo que haya escogido.

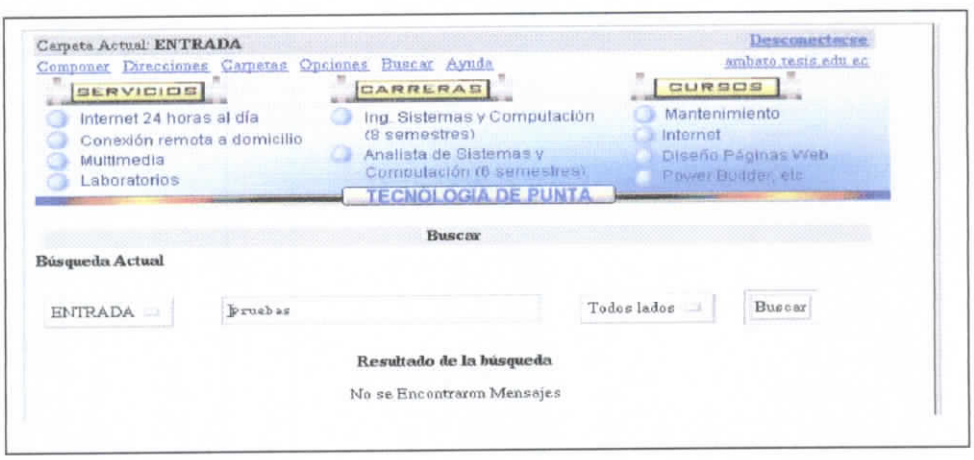


Fig. # 6.13: Resultado de la Búsqueda (no Encontrado).

En la fig.# 6.13. se muestra una pantalla que devuelve el resultado de una búsqueda en la carpeta de Enviados, la búsqueda se realizó en todo el mensaje pero no se encontró la palabra que se ingresó para que realice la búsqueda, se puede volver a realizar la búsqueda con otra palabra o en otra carpeta.



Fig.# 6.14: Pantalla que indica la Opción de carpetas.

En la Fig.# 6.14. se muestra la pantalla en la que se despliegan las carpetas que presenta el sistema, estas son: Entrada, Basura, Enviados, Borrador. Está ubicada en la parte izquierda de la pantalla, es una de las opciones del menú principal.

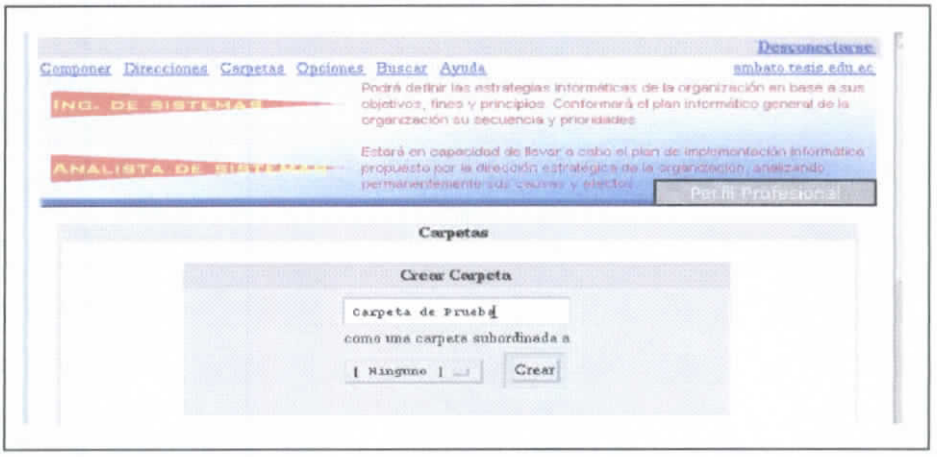


Fig.# 6.15: Pantalla que indica como crear una Nueva Carpeta.

En la Fig.# 6.15. se muestra la forma de crear una nueva carpeta con el nombre y según la necesidad del usuario, puede crear carpetas subordinadas a una de las carpetas principales que como ya se indicó anteriormente se encuentran en la parte izquierda de la pantalla. la nueva carpeta aparecerá al dar clic sobre el botón .

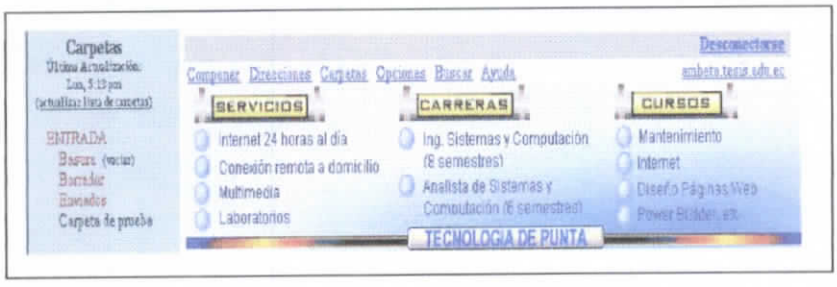


Fig.# 6.16: Pantalla que presenta la Creación de una Nueva Carpeta.

En la Fig.# 6.16. se muestra la creación de la nueva carpeta y aparece junto con las carpetas principales a la izquierda de la pantalla.

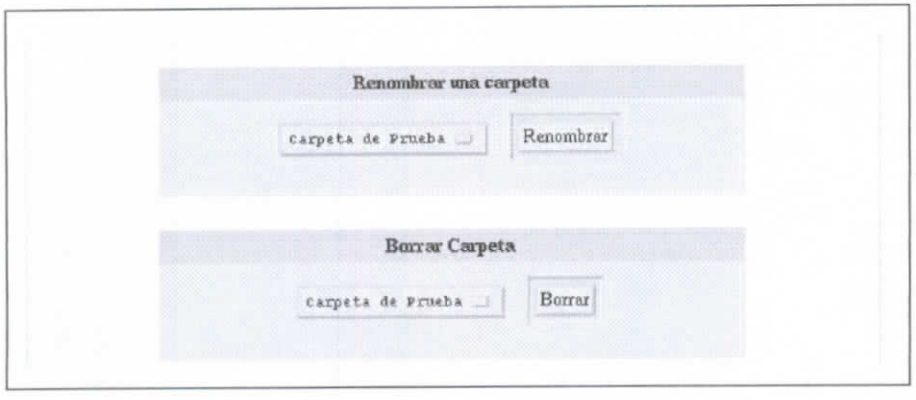


Fig.# 6.17: Pantalla que muestra como Renombrar una carpeta.

Como se observa en la Fig.# 6.17. el sistema también puede renombrar una carpeta, es importante indicar la carpeta a ser renombrada y dar clic sobre el botón renombrar, en la misma pantalla se presenta la opción de borrar la carpeta, según la necesidad del usuario, al dar clic aparece la nueva pantalla para que el usuario digite el nuevo nombre.

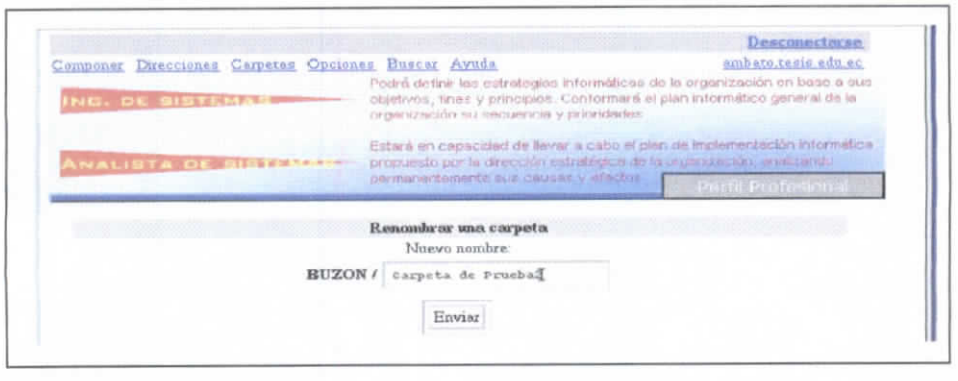


Fig.# 6.18: Pantalla que indica como Renombrar una carpeta.

En la Fig.#6.18. se muestra la manera de renombrar una carpeta creada por el usuario, el sistema pide al usuario el nuevo nombre de la carpeta y prácticamente la cambia de nombre.

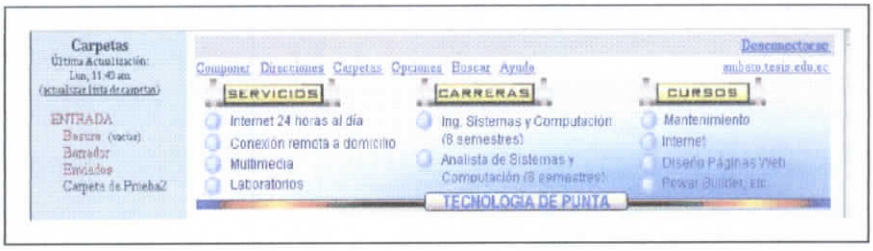


Fig.# 6.19: Pantalla que muestra una carpeta Renombrada.

En la Fig.# 6.19. se muestra como se renombró una carpeta creada por el usuario por otro nombre, esto lo puede hacer el usuario según su necesidad.

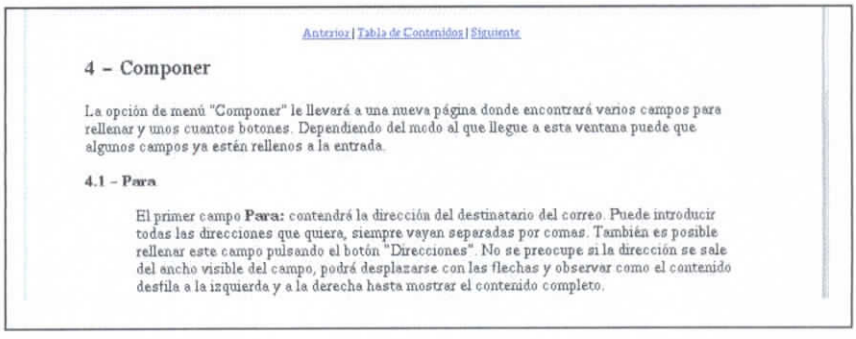


Fig.# 6.20: Pantalla que muestra la Opción Ayuda.

La Fig. # 6.20. muestra que el sistema presenta una opción de ayuda para el usuario que ingresa por primera vez y no conoce del manejo del mismo, pueda consultar por cada una de las opciones del sistema, en el ejemplo anterior la ayuda está relacionada con la opción componer, es decir escribir un Mail, está ayuda aparecerá dando clic sobre el botón "Ayuda" del menú principal.

6.6. Implementación.

Luego de haber terminado el análisis y desarrollo del WebMail se procede a implementarlo, para lo cual procederemos a explicar en forma general la instalación de las herramientas que se utilizaron.

6.6.1. Instalación del sistema.

En cuanto se refiere a la plataforma o sistema operativo sobre el cual operaría nuestro WebMail, se mencionó en capítulos anteriores, que se trabajaría con Linux, debido a que se lo implementaría en el servidor Sun de la PUCESA, el cual al comienzo de nuestra investigación poseía Red Hat Linux 6.0. Tomando en cuenta que nuestra investigación duraría aproximadamente una año, decidimos seleccionar la plataforma Red Hat Linux versión 7.1(la última versión lanzada para esa fecha), con el fin de que no ocurra como en algunas tesis anteriores que cuando están listas para implementarse, no funcionan debido a que fueron desarrolladas para una plataforma antigua.

Debido a que nuestro WebMail fue desarrollado en el lenguaje para páginas Web PHP, fue necesario instalar y levantar los servicios de Apache (Servidor de páginas Web en Linux) y CGI (Interfaz de entrada de comandos). Esto normalmente se recomienda activar al momento de instalar Red Hat, dentro del proceso de instalación existen opciones que permiten activar o desactivar estos servicios.

Luego de la Instalación de Linux se utilizó el comando "setup" al nivel de indicador de comandos, en nuestro caso "root@ambato/root# setup"; se ingresa a la opción System Services y se levanta el servicio httpd para páginas Web, además los servicios: imap (manejo de correos), imaps (auxiliar para manejo de correos), ipop2 – ipop3 (para correo entrante y saliente) y sendmail (como agente de correo). Generalmente estos servicios quedan activados al instalar Red Hat, pero siempre es necesario revisar, puesto que puede suceder que no sean levantados, httpd.conf es el archivo de configuración del sistema, se abre este archivo para indicar en que parte va a ser instalado.

El servidor está configurado como Server Name ambato, Server Admin

root @ambato.tesis.edu.ec, el puerto utilizado para el servicio de páginas web es el 80; para realizar una verificación de esta configuración se utilizóun utilitario que se indicará más adelante. Es importante indicar que este servicio va a ser administrado por User Apache; Group Apache. El camino de direccionamiento de ejecución de páginas web configurado fue:: Document Root/var/www/html.

6.6.2. Configuración de Apache.

Una de las herramientas importantes para que el WebMail pueda ser utilizado es Apache, ya que se está convirtiendo en el servidor web más popular de Internet, y funciona en muchas arquitecturas, tales como, Linux, Solaris, SunOS4.x y Windows NT. Linux es una de las plataformas más usadas para ejecutar Apache debido a su estabilidad y bajo coste, en nuestro caso en particular se configuró de la siguiente manera:

En el Menú inicio, Programas, Sistema, buscamos la opción "Apache configuration", se presenta una pantalla con varias pestañas:

- Main, aqui indicamos el nombre del servidor
  Server Name: ambato
  Webmaster: email address root @ambato.tesis.edu.ec
  Port : 80

- En la pestaña Virtual Host, no es necesario cambiar nada, puesto que por defecto indicará un puerto virtual que es el localhost.

- En al pestaña Server se indicará la herramienta con que corre el servicio:
  User Apache
  Group Apache

- En la última pestaña que presenta la pantalla Perfomance Tuning, se indica cuantos usuarios pueden conectarse simultáneamente, en este caso 150, se acepta y se graba el archivo httpd.conf.

6.6.3. Instalación del Software.

Para que el nuevo software quede instalado en la plataforma que se utilizó es importante encontrar la carpeta html, siguiendo el camino var/www/html, cuando esta carpeta esté abierta se descomprime el software, y aparecerá el archivo index.html y la carpeta holamail, la misma que guarda al software.

Para que el sistema se pueda configurar se debe ingresar a la carpeta holamail y ejecutar el archivo config.php utilizando ./ configure desde el root, aparecerá un menú de configuración del sistema, presenta varias opciones que pueden ser utilizadas según la necesidad o el gusto del usuario, cada vez que se realice un cambio se debe grabar y volver a recargar el sistema, entonces se observará los cambios.

6.7. Análisis y Evaluación de la Implementación.

Luego de haber realizado la respectiva investigación y la realización del sistema se procedió a la implementación del mismo en el servidor de la PUCESA, en este proceso nos encontramos con ciertos problemas porque para este caso en particular no se había actualizado la plataforma Linux a una versión como suponíamos que lo haría al principio de nuestra investigación, pero es importante acotar que por ser un sistema muy versátil y de fácil manejo no presenta complejidad para la instalación en otra máquina que tenga una plataforma que pueda soportar el sistema, siguiendo los pasos que se mencionaron anteriormente.

CONCLUSIONES Y RECOMENDACIONES.

Por medio del estudio que se ha realizado durante este proyecto, se ha podido llegar a conclusiones importantes, para la creación de un WebMail que funcionará en la Pontificia Universidad Católica del Ecuador Sede Ambato, esperamos que sea de utilidad para la realización de nuevos proyectos.

CONCLUSIONES.

- Los objetivos planteados al inicio del presente proyecto de tesis, fueron cumplidos, obteniendo como objetivo final la creación de un WebMail para la PUCESA.

- Como resultado del trabajo llevado a cabo en este proyecto, se ha realizado y documentado un estudio detallado acerca de los diferentes lenguajes de programación existentes para la creación de un WebMail, cabe indicar que se ha profundizado, en el lenguaje PHP, también nos hemos referido a la Plataforma Linux, en la cual se ha realizado este proyecto. El resultado de esta búsqueda se ha plasmado en las páginas del presente documento, sirviendo de punto de partida para futuras investigaciones.

- Para la creación del WebMail, es importante analizar las herramientas que se van a utilizar, tanto la plataforma en la que se va a trabajar como también el lenguaje de programación, debiendo ser fácil de entender tanto para quienes lo programen como para los usuarios.

- A partir de los resultados del estudio bibliográfico, se ha trabajado en el diseño e implementación de este nuevo software, que esperamos que sea de gran utilidad para nuestra Institución. La información que se proporciona en este trabajo es producto de la investigación tanto en el Internet, libros, revistas,

folletos y otros documentos relacionados, esta información se pone a disposición para facilitar algunos datos que puedan servir como punto de partida para nuevas investigaciones.

- Es importante indicar que el proyecto cuenta con una corta publicidad de la Universidad, sin embargo sirve al usuario para orientarlo e informarlo de los servicios que presta la PUCESA, con un alto nivel Académico y Tecnológico.

- Una de las herramientas importantes para la creación de este WebMail es el uso de los protocolos de Comunicación, los mismos que permiten la transferencia de datos de una forma segura y óptima, para que las computadoras puedan compartir archivos (programas y datos), mejorando así la transferencia de los mismos.

- En la elaboración de este proyecto se ha utilizado como lenguaje de Programación PHP4, el mismo que presenta ciertas facilidades para la programación puesto que permite intercalar las sentencias PHP en las paginas HTML. El lenguaje PHP es un lenguaje de programación de estilo clásico, esto quiere decir que es un lenguaje de programación con variables, sentencias condicionales, bucles, y funciones.

- Para la transferencia de correo hemos utilizado Sendmail ya que es un conocido agente de transferencia de correo MTA (Mail Transport Agent) usado en Internet que maneja un amplio porcentaje de todos los correos ruteados en Internet. Este agente se traslada de un host a otro buscando correos ruteados. Existen otros agentes de transferencia de correo (que pueden ser utilizados con Red Hat Linux), pero la mayoría de administradores eligen utilizar Sendmail debido a su potencia, escalabilidad y compatibilidad con los estándares de Internet. El deber principal de Sendmail, como otros MTAs, es el de trasladar con seguridad e-mails entre hosts, habitualmente usando el Simple Mail Transfer Protocol (SMTP).

- La utilización de la Plataforma Linux se debió a que es un sistema operativo con una serie de características que lo hacen totalmente diferente a los más usados como MS-DOS o Windows que están realizados con tecnología propietaria, dicho de otra forma no existe posibilidad de realizar modificaciones ni de saber como se realizó dicho sistema. Es un sistema operativo de código abierto, es decir que se distribuye de forma gratuita bajo una licencia pública general de GNU (Organización de Software Libre (Free Software), soporta un amplio espectro de aplicaciones o paquetes de programación. Por otra parte el servidor de la PUCESA tiene instalado este tipo de plataforma.

- La versión del sistema operativo que se utilizó fue Linux Red Hat 7.1, es importante indicar que existió algunos inconvenientes para la instalación del sistema en el servidor de la Universidad, puesto que la plataforma que utiliza es Linux versión 6.0 para Solaris.

RECOMENDACIONES.

- Al final de este proyecto es importante recalcar el avance de la tecnología y por ende el uso de Internet con sus respectivos servicios, uno de ellos y el más usado es la comunicación por medio del correo electrónico, el WebMail es un conjunto de scripts (pequeños programas) a través del cual se puede acceder al correo electrónico directamente desde cualquier navegador (Internet Explorer o Netscape). Esto es de gran utilidad, puesto que no importa en qué lugar un usuario se encuentre, podrá acceder a leer y responder sus mensajes de correo electrónico con total facilidad.

- Otra recomendación que emana la realización de este proyecto esta relacionada con el aprendizaje de los estudiantes de nuestra institución los mismos que deben profundizar más en el estudio de otros sistemas operativos que no sean los convencionales, como sugerencia está Linux, por todo lo expuesto en este documento, investigar sobre otros lenguajes de programación, que son fáciles de aprender y entender, para la creación de proyectos de una forma más rápida, a menor costo y de fácil manejo.

- Como la Computación están en un constante cambio se recomienda que los estudiantes, profesores y demás integrantes de la institución estén siempre actualizados, puesto que la tecnología siempre tendrá cosas novedosas las cuales debemos aprender.

- Otra Recomendación importante para la Universidad es que los proyectos presentados por los alumnos sean para uso de la Institución, y no queden archivados por falta de compatibilidad entre las plataformas utilizadas y el sistema operativo con el que cuenta el laboratorio de la Universidad.

MANUAL DE USUARIO

El sistema que se ha desarrollado se trata de una interface, o cliente, de correo escrito en PHP4 se ha diseñado para permitir el acceso al correo a través de un servidor desde cualquier parte del mundo empleando la web.

1. Lo Básico.

El WebMail está estructurado en dos segmentos principales llamados frames o marcos, el marco de la izquierda muestra la lista de carpetas suscritas, en el marco de la derecha se indica la mayor parte de opciones: en la parte superior está la barra de menú, la opción desconectarse, la misma que permitirá la salida del programa en forma segura, indica la carpeta actual que se encuentra activada, por defecto la carpeta que se encuentra habilitada al ingresar al sistema es la carpeta de Entrada.

En cada una de las pantallas que presenta el sistema existirá una publicidad de la PUCESA, se encuentra en la parte superior de la pantalla, además cuenta con un antivirus que funciona internamente, si el mensaje que recibe el servidor tiene virus, lo devuelve a su lugar de origen, caso contrario ingresa a la bandeja de entrada del usuario.

En la Barra de Menú bajo la opción de desconexión hay una fila de opciones:

Componer: Permite redactar y enviar mensajes que pueden contener ficheros adjuntos.

Direcciones: Contiene una lista de direcciones según la necesidad del usuario.

Carpetas: Permite manipular carpetas, desde aquí se puede borrar, crear renombrar, suscribirse y darse de baja de las distintas carpetas de correo.

Opciones: Para ajustar las distintas opciones del WebMail.

Buscar: Con esta herramienta se pude localizar un criterio o palabra determinada en un correo.



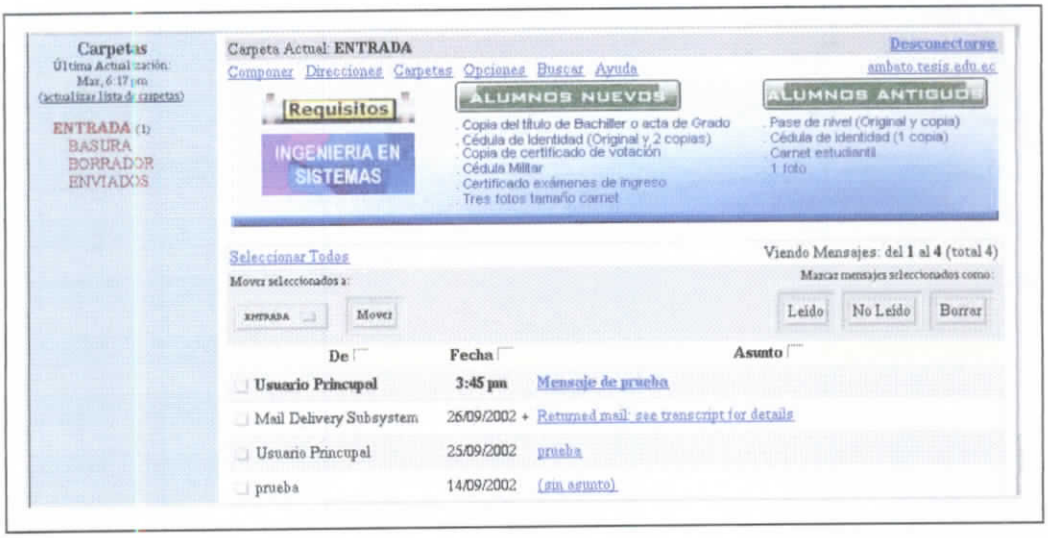
2. El Indice De Mensajes.

Después de seleccionar una carpeta, en el marco de la derecha aparecerá una lista de los mensajes que contiene la carpeta seleccionada, bajo la barra superior del menú, se indica el número y posición de los mensajes que está visualizando. Por ejemplo: viendo mensajes 20 a 30 (45total).

A continuación se encontrará una barra con un desplegable y dos botones, el desplegable permite indicar una carpeta de destino para todos los mensajes que se encuentren marcados antes de pulsar el botón "Mover", de manera similar funciona el botón "Borrar" que eliminaría los mensajes marcados.

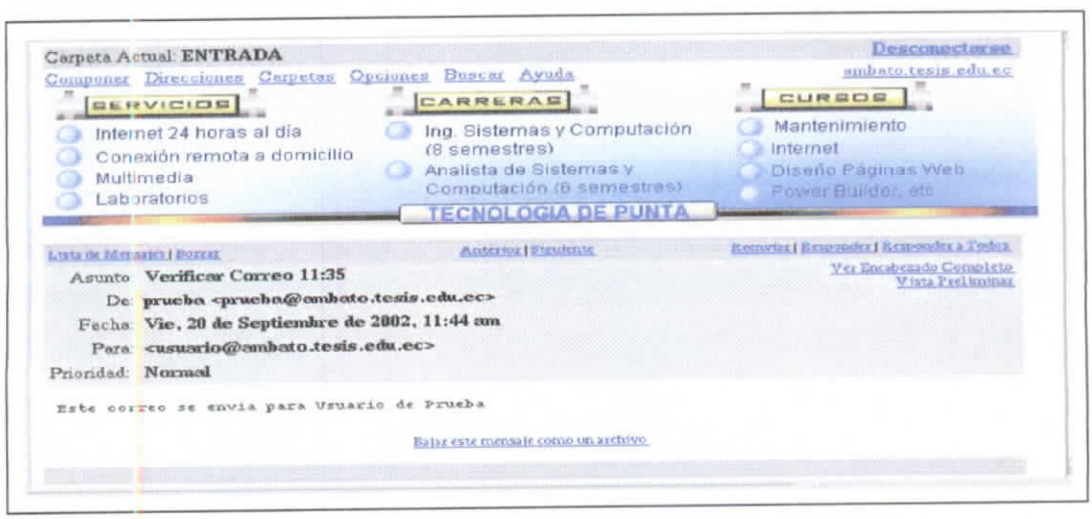
Se muestra una barra con tres campos: De, Fecha y Asunto, estos encabezados separa el índice en tres columnas, De contiene el remitente del mensaje, Fecha, indica la fecha en la que se envió y Asunto, muestra el tema del mensaje. Si entre las columnas de Fecha y Asunto aparece un símbolo por ejemplo +,"A",!, indica que el mensaje contiene ficheros adjuntos, que el usuario ya contestó el mensaje o que el mensaje es urgente.

Por último se mostrará la lista de mensajes en sí, notará que los mensajes no leídos aparecen en **negrita,** mientras que los demás están en letra normal, para cambiar el orden de los mensajes, se puede pulsar sobre el símbolo de la derecha del encabezado por el que quiera ordenar de forma natural, una segunda vez para orden inverso, se podrá observar como el símbolo cambia en función del orden.



3. Leer un correo electrónico.

Al hacer clic en el tema de un mensaje de correo, se podrá observar que las direcciones web y de correo aparecen como enlaces, por lo que el usuario puede pulsarlos para abrir una página o escribir un correo, Aparece además una barra de opciones separada en tres secciones, la primera a la izquierda para volver a la lista o borrar el mensaje, la del medio para avanzar o retroceder en la lista y la de la derecha para reenviar o responder el correo.



3.1. Lista de Mensajes.

Al dar clic en este enlace el mensaje presentado volverá a la lista de mensajes originales.

3.2.Borrar.

Al dar clic sobre este enlace el mensaje en curso será borrado. También se borrarán todos los ficheros adjuntos que pudiera tener el e-mail. Si no se quiere perder estos ficheros es recomendable que se los descargue primero.

3.3. Navegación

Los botones de navegación se encuentran en el medio del menú. Tanto Anterior como Siguiente, solo aparecerán subrayados si es posible moverse en la dirección indicada.

3.4. Siguiente.

Al pulsar este enlace se puede pasar al siguiente correo. Si el enlace no estuviera activo, significa que no hay más mensajes a continuación.

3.5. Reenviar.

A la derecha, en el enlace de "Reenviar" abre la página para componer mensajes, con el mensaje previamente visualizado en el texto del mensaje precedido por el texto Mensaje Original, y los datos de cabecera del mensaje. El asunto original precedido por las letras Fwd. También se representarán los ficheros adjuntos si lo tuviera.

3.6. Responder.

Al pulsar este enlace se puede contestar al remitente del mensaje que esté visualizando. El asunto original se precederá de las letras Re: y el texto original se precederá por el símbolo ">", para poder respetar el contexto del mensaje a la hora de responder. Si se observa líneas del mensaje original que no estuvieran precedidas por el mencionado símbolo esto se debe a que la longitud del corte de línea es superior al ancho de edición y el texto a sobresalido por la derecha.

3.7.Responder a todos.

Hace basicamente lo mismo que responder, solo que todas las direcciones que aparecen en la cabecera del mensaje recibirán una copia de la respuesta.

3.8.Ver encabezado Completo.

Permite visualizar la cabecera completa del mensaje, incluye el camino recorrido por el mensaje así como múltiple información sobre el mensaje.

3.9. Bajar este mensaje como un archivo.

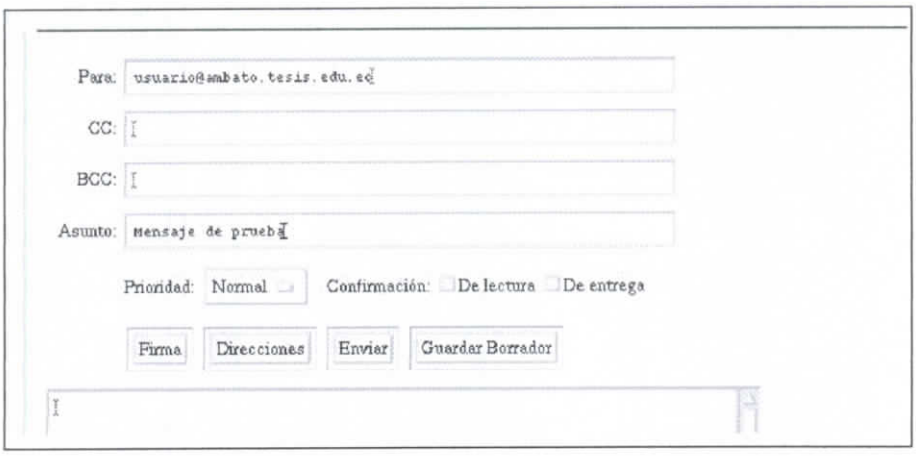
En la parte de abajo, justo por encima de la barra inferior se encuentra este enlace, que permite almacenar una copia del mensaje en formato texto en el disco duro del computador.

3.10.Ficheros Adjuntos.

Todos los ficheros que hayan sido adjuntados al mensaje se mostrarán al final del mismo mediante una serie de enlaces con el nombre de los ficheros. A la derecha del enlace puede verse el tamaño, el tipo del fichero y un nuevo enlace llamado descargar, mientras que el enlace de la izquierda se destina a la visualización del contenido de fichero, el de la derecha sirve para guardar una copia del archivo en el disco duro.

4.Componer.

La opción de menú "Componer" llevará al usuario a una nueva página donde encontrará varios campos para rellenar y unos cuántos botones. Dependiendo del modo al que llegue a esta ventana puede que algunos campos ya estén rellenos de entrada.



4.1. Para.

Contendrá la dirección del destino del correo. Se puede introducir todas las direcciones que quiera, van separadas por una coma. También es posible rellenar este campo pulsando el botón "Direcciones", si la dirección se sale del campo visible se podrá desplazar con las flechas.

4.2. Cc.

Este capo es una abreviatura de Con Copia para, contiene las direcciones de las personas para las que se desea enviar una copia del mensaje, funciona de una manera similar al acampo "Para".

4.3. Bcc.

BCC permite indicar copias "ciegas" , las personas indicadas en este campo recibirán una copia del mensaje sin que las personas indicadas en los dos campos anteriores lo sepan.

Asunto.

Es importante elegir un asunto clarificador para su mensaje, esto le ahorrará tiempo a todos.

4.5.Botón "Direcciones".

Este botón abre la libreta de direcciones precedida por un pequeño buscador. Puede introducir algún texto en el campo "Buscar" para obtener direcciones relacionadas.

4.6.Cuerpo de Mensaje.

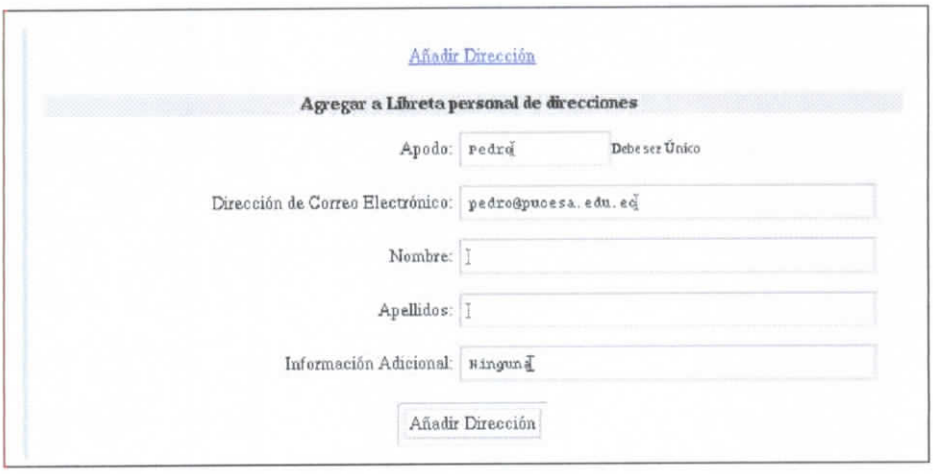
El rectángulo grande debe contener el cuerpo del mensaje si se ha elegido una firma en las opciones esta aparecerá dentro de manera predeterminada. Este apartado es el que contiene el mensaje propiamente dicho.

4.7.Adjunto.

Al final de la página está el campo "Adjunto", que permitirá adjuntar un fichero a su correo, el fichero deberá ser accesible desde la máquina que el usuario este usando, el botón de examinar le permitirá buscar más fácilmente el fichero, aunque también puede escribir a mano la ubicación y el nombre de fichero, una ves que el capo este lleno se debe pulsar el botón de agregar para que el fichero quede vinculado al email. Este proceso puede tardar un poco en función del tamaño y la velocidad de conexión.

5.Direcciones.

La agenda de direcciones es una característica ideal para ahorrar tiempo, las direcciones que se emplean con frecuencia pueden almacenarse aquí.



5.1. Apodo

Aquí se debe poner un nombre con el que se refiere familiarmente a una persona, a la que lo permita asociarlo rápidamente con el propietario del correo al que pertenezca, debe ser único.

5.2. Correo Electrónico.

Debe contener la dirección de correo completa de la persona, el contenido no puede improvisarse.

5.3. Información.

Este campo también puede emplearse para introducir algún dato que permita recordar quien es la persona a la que pertenece.

5.4.Agregar Dirección.

Se debe rellenar los campos listados, los tres primeros son obligatorios (Apodo, Dirección y Nombre) los demás pueden omitirse.

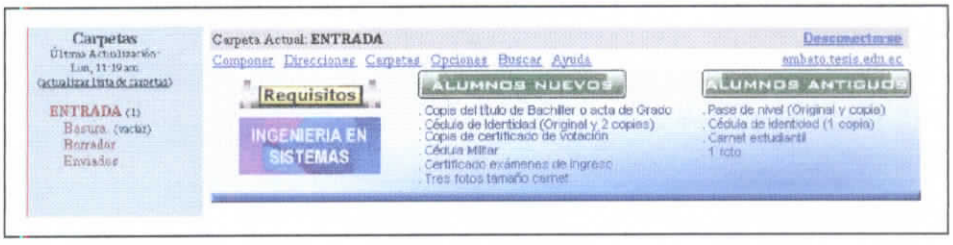
6.Carpetas.

El usuario puede almacenar sus mensajes en distintas carpetas para organizarse mejor.

6.1. Carpetas Suscritas y Panel Izquierdo.

Las carpetas de correo a las que está suscrito están listadas en el lado izquierdo, este panel puede refrescarse automáticamente a intervalos predeterminados en la página de opciones aunque puede refrescarse manualmente sobre el enlace que aparece debajo del título del panel izquierdo.

La primera carpeta de Entrada contiene los mensajes recibidos, y a la derecha entre paréntesis, si los hay, el número de mensajes que aún no se han leído la última vez que se refrescó este panel, bajo esta carpeta pueden haber una o más carpetas, según la configuración de cada usuario.

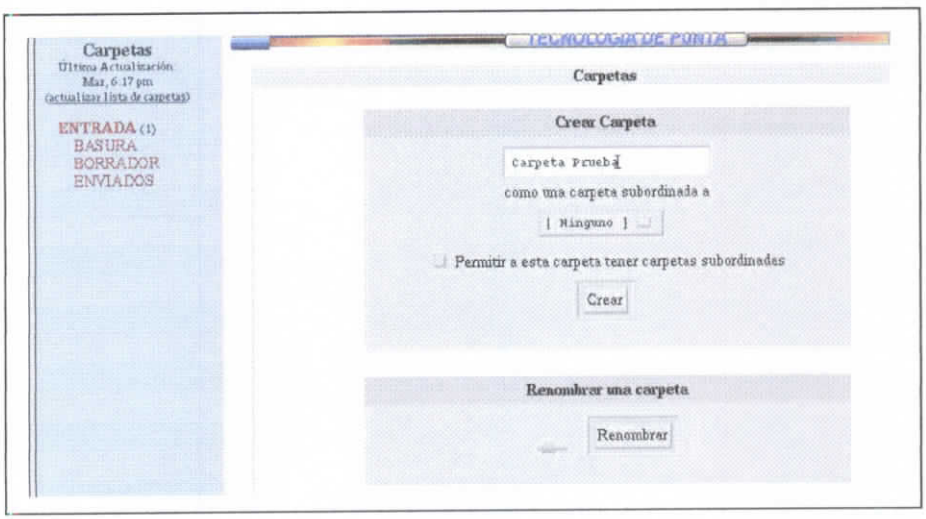


6.2. Borrar.

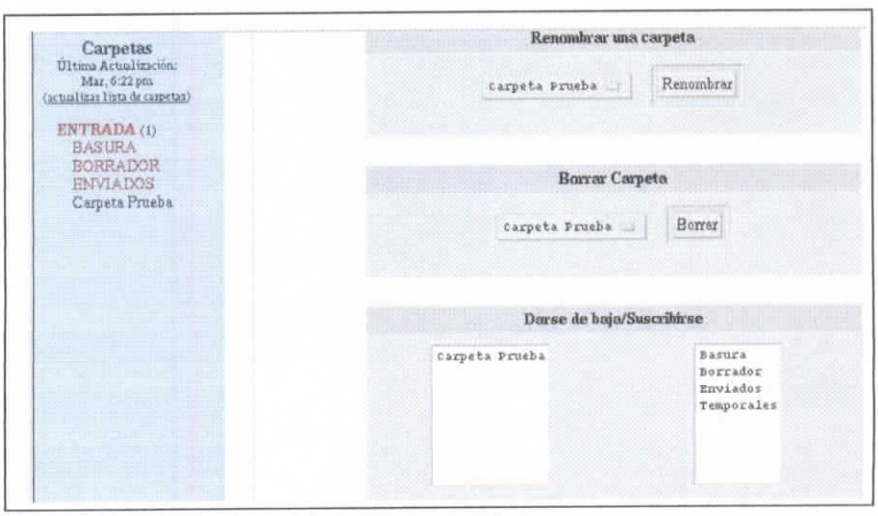
Puede borrar cualquiera de las carpetas que se muestra a la izquierda del botón borra, esta lista no tiene porque contener todas las carpetas existentes, ya que las carpetas especiales no pueden borrarse.

6.3. Crear.

Se puede crear carpetas solo con indicar el nombre de la misma y pulsando el botón crear, pude colocar una carpeta por debajo de la otra, en este caso se debe especificar la carpeta "madre" que ha su vez habrá sido creada activando la opción "permitir a esta carpeta tener subcarpetas".



Luego de refrescarse ya sea automáticamente o en forma manual el lado izquierdo de la pantalla se presenta de la siguiente manera:

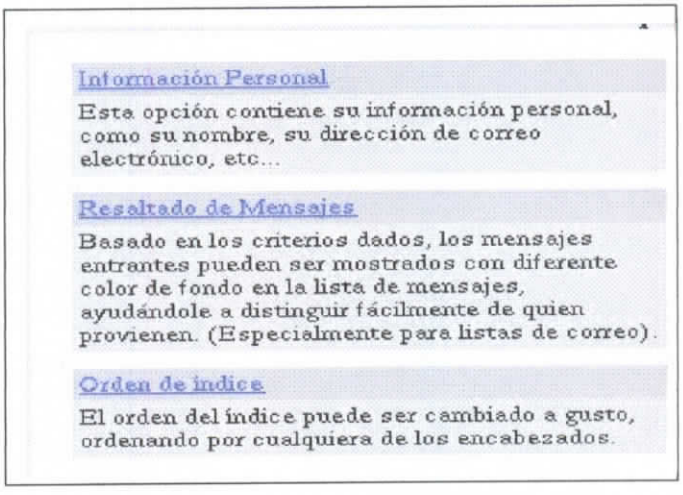


6.4.Renombrar.

Puede renombrar todas las carpetas mostradas en el desplegable, Es importante indicar que las carpetas originales o especiales no se las puede renombrar.

7.Opciones.

Una de la grandes ventajas que presenta este Webmail es el nivel de personalización del que dispone, dependiendo de su configuración podrá disponer de una gran variedad de colores sin alterar a los usuarios del sistema, como Información personal, Preferencias de pantalla, Resaltado de Mensajes, Preferencia de Carpetas.



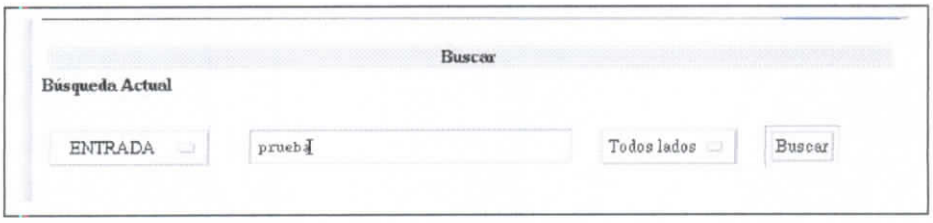
8.Buscar.

Esta es una útil herramienta que permite buscar en una carpeta determinada un criterio dado que se compara con los distintos campos de cabecera de los mensajes.

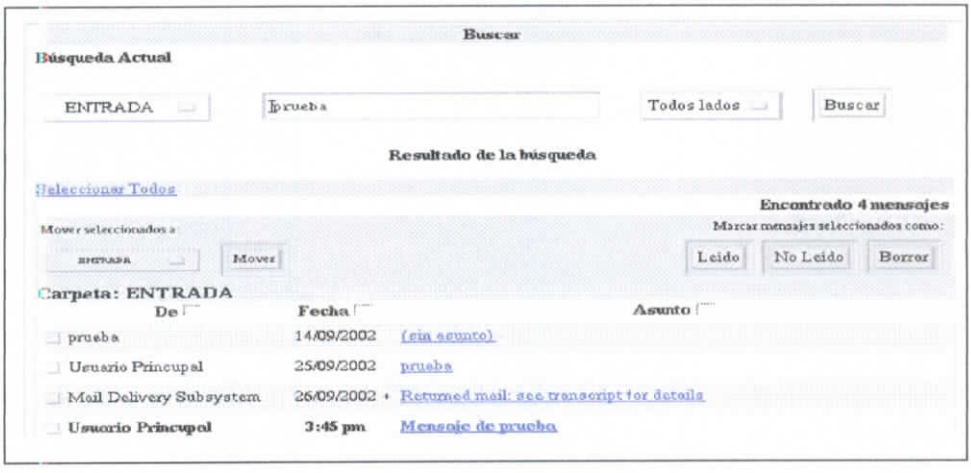
Simplemente elija la carpeta en la que quiere buscar, teclee el criterio de búsqueda y la parte del mensaje que quiere buscar, cuando pulse buscar, la lista de mensajes que

coincidan con su criterio se listará debajo de los campos de búsqueda, puede elegir el mensaje que desee ver y leerlo como si se tratase de una carpeta normal y corriente.

Puede buscar en el Cuerpo, por el Asunto, Todos los lados que incluye todas las partes del mensaje incluyendo la cabecera, Para, Cc y De.



Luego de haber pulsado el botón "Buscar", aparecerá los resultados de la búsqueda:



Es importante mencionar que el sistema presenta una opción de ayuda, la misma que explica lo que se ha expuesto en este manual.

ANEXOS

GLOSARIO E ÍNDICE DE TÉRMINOS

A

ARPANET **Advanced Research Projects Agency Network** Red de la agencia de investigación avanzada de proyectos.

API **Application Program Interface** Conjunto de convenciones internacionales que definen cómo debe invocarse una determinada función de un programa desde una aplicación.

APLICACIÓN Programa informático que proporciona servicios de alto nivel al usuario, generalmente usando otros programas más bajos que se sitúan por debajo.

APPLETS Pequeñas aplicaciones escritas en Java y que se difunden a través de la red para ejecutarse en el visualizador cliente.

ASCII **American Standard Code of Information Exchange**.
Estándar que recoge 128 caracteres, letras, números y símbolos utilizados en procesadores de texto y algunos programas de comunicaciones.

ASP **Active Server Pages.** Páginas activas de servidor. Como su nombre lo indica son páginas activas las cuales son analizadas por el servidor modificando el código HTML según las necesidades.

AUTENTIFICACIÓN Procedimiento de seguridad por el que se verifica el reconocimiento y validación de un usuario de Internet.

B

BUG Termino que expresa el error de programación cualquier defecto de funcionamiento en los equipos de un sistema de computación.

BPS Es el acrónimo de bits por segundo, medida estándar de velocidad con que se trasmiten datos mediante un módem.

BROWSER Los buscadores son sitios Webs, especializados en localizar información a través de Internet.

C

CGI **Common Gateway Interface.** Interfaz de puerta de acceso común que permite a un usuario de Internet interactuar con la World Wide Web.

COOKIE **Galleta.** Archivo de texto en la memoria del cliente del que se valen los servidores Web para guardar información acerca del visitante de un sitio.

CRIPTOGRAFÍA Ciencia que estudia algoritmos informáticos que se codifican la información para posibilitar la realización de transmisiones seguras en la red.

CLIENTE Aplicación que se encarga de formar una solicitud válida para enviarla al servidor.

CLIENTE/SERVIDOR Modelo lógico de una forma de proceso cooperativo, independiente de plataformas hardware y sistemas operativos.

CODIGO Se refiere a las instrucciones contenidas en programa, y entendibles por el ordenador.

D

DNS Es el acrónimo de **Domain Name System** (Sistema de Nombres de Dominio), que se emplea para identificar a un servidor.

DIRECCIÓN IP Dirección de protocolo de Internet. Número especial que se asigna a cada ordenador que forma parte de Internet, aunque sea de manera temporal.

E

EXTRANET Es la tecnología para transferencia segura de datos entre intranets protegidas por cortafuegos.

ENCRIPTAR Técnica por la que la información se hace ilegible para terceras personas. Para poder acceder a ella es necesaria una clave que sólo conocer el emisor y el receptor.

E-MAIL Intercambio de mensajes entre personas conectadas a una red de manera similar al correo tradicional.

F

FIPS Se utiliza para realizar Particiones. Generalmente FIPS es un programa que se asegura bastante antes de realizar operaciones.

FAT **File Allocation Table**, Tabla de Ubicación de Archivos.

FRAME Capacidad de un documento HTML para subdividirse en diferentes ventanas independientes.

FSF **Free software foundation** Fundación de software gratuito

FTP **File Transfer Protocol.** Protocolo de transferencia de archivo que se utiliza para transferir un archivo de un ordenador desde un lugar a otro.

G

GUI **Interfaz Gráfica de Usuario**. Término aplicado a la representación gráfica de la manipulación de los comandos de un programa.

H

HIPERTEXTO Texto vinculado electrónicamente con otros documentos, a los que se puede acceder haciendo clic en la palabra que sirve de vínculo.

HIPERVÍNCULO Enlace. Son referencias entre varias páginas que se encuentran en la red Internet o Intranet.

HOST Utilizado a veces como sinónimo de mainframe, en realidad identifica al ordenador central en un sistema informático complejo.

HOSTING Servicio mediante el cual una empresa se encarga de gestionar las páginas web de otras empresas, incluyendo el hardware, el software y las comunicaciones.

HTML **Hyper Text Markup Language.** Formato de texto electrónico caracterizado por poder vincular otros textos a través de enlaces.

HOME PAGE Define lo que se conoce como página de inicio o principal (la primera que aparece cuando se accede a un servidor de páginas Web.

HTTP Protocolo de transporte de hipertexto **Hipertext Transport Protocol**. Conjunto de reglas pactadas entre sistemas para la transmisión de hipertextos.

I

INFORMACIÓN Elemento fundamental que maneja los ordenadores en forma de datos binarios.

INTERFAZ Se trata del intermedio natural entre la computadora y sus periféricos o sea es la conexión e interacción entre hardware, software y el usuario.

INTERNET Red electrónica mundial que interconecta a miles de redes y millones de usuarios de ordenadores.

INTRANET Es el sistema de información interna, basado en tecnología Internet, utilizado por una empresa; representa la parte "inteligente" de la organización.

ISO **International Standard Organization**. Organismo internacional de normalización.

INTEL El fabricante de microprocesadores y otros semiconductores más

K

KERNEL Rutinas del sistema operativo responsable del manejo de las funciones básicas del sistema

L

LINUX Es un sistema operativo de código abierto, es decir que se distribuye de forma gratuita bajo una licencia pública general de GNU.

M

MTA **Mail Transport Agent** usado en Internet que maneja un amplio porcentaje de todos los correos ruteados en Internet.

MAILTO El protocolo de Internet que utiliza para enviar correo electrónico.

N

NFS **Network File System,** Sistema de archivos de RED. Es un sistema distribuido para archivos, este es utilizado en redes heterogéneas.

P

PASSWORD Conjunto de caracteres mediante los cuales un operador es identificado al entrar en comunicación con una terminal, un sistema o al pretender ejecutar un programa.

PHP3 "Hypertext Preprocessor", es un lenguaje interpretado de alto nivel embebido en páginas HTML.

POP3 Es un protocolo que permitir a una estación de trabajo recobrar correo que el servidor tiene almacenado.

PGP	Programa conocido por las siglas **Pretty Good Privacy** diseñado para asegurar la privacidad del correo electrónico.
PROCESADOR	Es el microchip encargado de ejecutar las instrucciones y procesar los datos que son necesarios para todas las funciones del computador.
PROCESO	Operación o conjunto combinado de operaciones con datos, o bien una secuencia de acontecimientos definida única y delimitada, que obedece a una intención operacional en condiciones predeterminadas. También se denomina proceso a una función que se está ejecutando.
PROTOCOLO	Lista de comandos a la que responde de manera estándar un servidor.
PROXI	Servidor que sirve en las organizaciones como caché o como cortafuegos (firewall).
P.U.C.E.S.A.	Pontificia Universidad Católica del Ecuador Sede Ambato.

R

REGISTRO Es una pequeña unidad de almacenamiento destinada a contener cierto tipo de datos.

RED Grupo de computadores y otros dispositivos periféricos conectados entre sí que pueden comunicarse y trasmitir datos entre ellos.

S

SLIP/PPP Una abreviatura común para los dos tipos de acceso por marcación a Internet que integran a su computadora en forma directa con Internet: el Protocolo Internet de Línea Serial(SLIP) y el Protocolo Punto a Punto (PPP).

SHELL Es básicamente la interfaz, el modo de comunicación, entre el usuario y el sistema. Cada shell tiene sus características propias.

SOCKETS Son puntos finales de enlaces de comunicaciones entre procesos.

SMTP **Simple Mail Transfer Protocol** es un sistema local asociado de entrega de correo residente y continuamente en ejecución.

SERVIDOR Aplicación que se encarga de procesar la petición que recibe del cliente, quién la procesa a fin de mostrarla en un formato comprensible para el usuario.

SQL **Structured query Language**. Es un estándar en el lenguaje de acceso a partir de datos.

SENDMAIL Es el agente de transporte de correo más común de Internet (en los sistemas UNIX).

SGML **Standard Generalized Markup Language,** es un sistema para definir tipos de documentos estructurados y lenguajes de marcas para representar esos mismos documentos.

T

TABLA Una o más filas de celdas de una página que se utilizan para organizar el diseño de una página Web o para ordenar el diseño de una página Web o para ordenar datos sistemáticamente

TELNET Protocolo de emulación de terminales para la entrada remota en internet, forma parte del conjunto TCP/IP. Programa de UNIX que utiliza este protocolo.

TCP/IP **Transmission Control Protocol /Internet Protocol.** Protocolos para la transmisión en Internet a través de un conjunto de reglas estandarizadas que permite a ordenadores de distintas redes comunicarse entre sí.

U

UNIX Sistema Operativo desarrollado por los laboratorios Bell, para la multiprogramación con una estructura de ficheros jerarquizados adoptado luego por la mayoría de fábricas de computadoras.

URL **Universal Resource Locator**. Localizador Universal de Recursos. Sistema unificado de identificación de recursos en la red.

W

WWW **World Wide Web**. Sistema de información con mecanismos de

hipertexto

WEBMAIL Es un agente de correo electrónico que permite leer, extraer y enviar mensajes a través de la interfaz gráfica de un navegador de Internet (browser).

BIBLIOGRAFIA

LIBROS:

- Redes de Computadoras de Andrew Tanenbaum, 1991, edición Pretice – Hall Hispanoamérica.
- Computer Networks Andrew Tanenbaum, 1998, edición Pretice Hall.
- Intranet Creación de un Website Bondensiek, Paúl edición Pretice – Hall.
- Manual de Linux.
- Redes Globales de Información con Internet y TCP/IP Douglas E. Comer.
- Redes de computadoras Uyless Black.
- La Magia de Internet Allen L. Wyatt, McGraw.
- Seguridad para Internet, Edward Amoroso - Ronald Shanp.
- Enciclopedia Informática y Computación Teleinformática de Cettico, Edición Cultural España 1999.

INTERNET:

- http://www.sardis.upeu.edu.pe/lucas/index.htm

- http://www.lucas.hispalinux.es/htmls/manuales.html
- http://www.sololinux.com
- http://www.007mundo.com/tour/webmail/hmi_welcome.htm
- http://www.tecnova.cl/Productos/webmail/descwebmail.html
- http://correo.unirioja.es:8000/manual/
- http://www. intranets.com
- http://www.ciberteca.net.
- http://www.yonau.com/referencia/
- http://www.apache.org
- http://www.openssl.org
- http://www.maestrosdelweb.org
- http://www.desarrolloweb.com
- http://bulmalug.net/
- http://webestilo.com/php/
- http://www.php-es.com/
- http://www.nusphere.com/
- http://rpms.arvin.dk/

- http://www.navegalia.com/personal/gginard/mispag.htm
- http://www.javasoft.com/
- http://www.perl.org
- http://www.home.netscape.com/
- http://www.mx.php.net/support.php
- http://www.php.net/manual/es/function.fsockopen.php
- http://www.htdig.org/
- http://www.zend.com
- http://www.perso.wanadoo.es/antoni.reus/
- http://www.jmk.simplenet.com/prod/download.htm
- http://www.php.unam.mx/
- http://www.php.net/manual/es/security.php
- http://www . monografias.com

PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR
PROGRAMAS ACADEMICOS AMBATO
BIBLIOTECA