

UNIVERSIDAD NACIONAL AUTÓNOMA DE NICARAGUA  
UNAN – LEÓN

Facultad de Ciencias  
Departamento de Computación



*Configuración de Servidor Web  
y de  
Correo Electrónico en Red Hat.*

Autores:

- *Lic. Liuva Pastrán Paniagua.*
- *Lic. Karen Adalís Solís Salinas.*

Monografía para Optar al Título de  
Ingeniería en Sistemas de Información.

León, Nicaragua.  
Diciembre del 2007.

**ESTA MONOGRAFÍA HA SIDO APROBADA POR:**

**Universidad Nacional Autónoma De Nicaragua  
UNAN – LEÓN**

**Facultad de Ciencias  
Departamento de Computación**

**MSc. Ricardo Espinoza.  
Director Departamento Computación**

**MSc. Ernesto Espinoza.  
Coordinador del Curso Sabatino de Nivelación ISI.**

**Lic. Rina del Pilar Aráuz.  
Tutora de la Tesis**

**Tribunal:**

**Presidente: MSc. Martin Ibarra.**

**Secretario: MSc. Maximo Guído.**

**Vocal: MSc. Ernesto Espinoza.**

**Fecha de defensa  
Jueves 13 de diciembre del 2007.**

## ACLARACIÓN AL LECTOR

Como aclaración a nuestros lectores, nos permitimos informar que al momento de realizar la defensa de este documento, ya teníamos el título de **Licenciatura en Computación** que ofreció la facultad de CIENCIAS de la universidad UNAN-LEÓN, y que luego retornamos a dicha facultad para recibir el **Curso Sabatino de Nivelación** ofrecido por el departamento de computación de la facultad de CIENCIAS, donde cursamos y aprobamos satisfactoriamente todas las asignaturas correspondientes a la carrera de Sistemas de Información y así optar al título de **Ingeniero en Sistemas de Información** de la UNAN-LEÓN.

## RESUMEN

### *Configuración de Servidor Web y de Correo Electrónico en Red Hat*

El desarrollo en los nuevos medios de comunicación electrónica permite el crecimiento de la información almacenada en los sistemas, haciendo necesario recurrir a métodos fiables para su manipulación. Siendo el propósito principal de este manuscrito explicar la forma de configurar un servidor web y un servidor de correo electrónico, basados en herramientas propuestas en el terreno tecnológico para servidores que garanticen la seguridad de la información, utilizando así a Red Hat Linux 8.0 como plataforma para este propósito. Dando como resultado un nuevo servidor llamado LIUKAR, el que tiene como objetivo principal el servicio de correo electrónico local y remoto.

## DESCRIPCIÓN DEL DOCUMENTO

Actualmente la información en el ámbito tecnológico es muy amplia y no se puede abordar un tema sin hacer énfasis en otro ya que estos siempre se conectan entre si. Debido a ello, nuestro trabajo monográfico esta compuesto por dos temas, donde cada configuración ha sido desarrollada y explicada de manera individual.

Por tal razón y para una mejor interpretación hemos dividido el documento en tres partes:

En la primera parte, mostramos las consideraciones previas que tomamos en cuenta antes de realizar las configuraciones, ya que ambos servidores fueron ensamblados en la plataforma Red Hat Linux versión 8.0.

En la segunda parte explicamos como configuramos el Servidor Web HTTP de Apache, a través del archivo http.conf.

La tercer parte, detallamos como realizamos la configuración para un Servidor de Correo Electrónico utilizando como agente de transferencia de correo a qmail.

## DEDICATORIA

*A mis padres*

**Félix y Clementina** por sus  
*juiciosos consejos y paciencia.*

*Pero especialmente a*

**Francisca Paniagua (q.e.p.d, 2006)**

*mi abuelita, quien me inculco la fe y el amor a DIOS,  
el valor y respeto a cualquier ser vivo y sobretodo por enseñarme a  
perseverar en la vida y así llegar a obtener mis metas a como se debe,  
con esfuerzo propio.*

*Lilva Pastrán Paniagua.*

## DEDICATORIA

*Dedico este logro primeramente a Dios:*

- *Por ser mi amigo fiel y constante.*
- *Por permitirme llegar a la meta propuesta.*
- *Por concederme la fuerza de voluntad, la sabiduría y la paciencia para lograr terminar este proyecto.*

*A mis Padres:*

*Por su amor incondicional y por todo el apoyo moral y económico que me han brindado en todo este tiempo, en especial a mi Madre, **Martha de Jesús Salinas Flores**, quien con sus sabios consejos me ha enseñado a ser una persona de bien.*

*Y por último, a todas aquellas personas que siempre están presentes en mi vida, brindándome todo su cariño incondicional y por animarme a seguir luchando por todos mis objetivos.*

*Karen Adalís Solís Salinas.*

## AGRADECIMIENTO

*A Dios nuestro Padre Celestial le damos gracias por darnos la fortaleza espiritual y física para ser capaces de avasallar cada dificultad que encontramos al realizar este trabajo.*

*Para lograr nuestro objetivo hubieron algunas personas a quienes agradecemos por su ayuda incondicional.*

*Principalmente a nuestras familias que con su apoyo emocional y económico nos animaron a finalizar un logro más en nuestras vidas.*

*Especialmente queremos agradecer a nuestra Tutora, **Lic. Rina del Pilar Aráuz**, a quien nos permitimos considerar una buena amiga, gracias a su experiencia de docente y sobre todo por compartir con nosotras sus conocimientos, nos permitió disipar dudas, orientándonos como resolver algunos problemas que se nos presentaron en la creación de este proyecto, nos brindó un gran apoyo con su valioso tiempo, paciencia y dedicación, motivándonos y enseñándonos a no desistir de alcanzar nuestro propósito del cual también ella es parte.*

*Por último agradecemos a cada uno de los docentes del departamento de Computación, quienes con su carisma, buen humor, oportuna ayuda y sabios consejos, en muchas ocasiones nos alentaron a seguir adelante.*

**A todos ustedes, muchas gracias.**

# INDICE

<b>OBJETIVOS.....</b>	<b>1</b>
➤ Objetivos Generales	
➤ Objetivos Específicos	

---

## **PRIMER PARTE: CONSIDERACIONES PREVIAS A LAS CONFIGURACIONES**

---

<b>I.</b>	<b>CONSIDERACIONES PREVIAS A LA CONFIGURACIÓN.....</b>	<b>2</b>
<b>1.</b>	<b>Selección del sistema operativo.....</b>	<b>3</b>
	Sistema Operativo Red Hat Linux 8.0.....	3
	Características, Ventajas y Desventajas de Linux.....	4
	<b>a.</b> Características de Linux.....	4
	<b>b.</b> Ventajas de Linux.....	4
	<b>c.</b> Desventajas de Linux.....	5
<b>2.</b>	<b>Selección de la máquina a usar.....</b>	<b>5</b>
	Requerimientos de la computadora.....	6
	➤ Requerimientos mínimos de instalación.....	6
	➤ Requerimientos mínimos recomendados.....	6
	➤ Características de la máquina utilizada para la configuración.....	6
<b>3.</b>	<b>Particiones del disco duro.....</b>	<b>7</b>
	<b>a.</b> swap.....	7
	<b>b.</b> /boot.....	7
	<b>c.</b> root (/).....	7
	<b>d.</b> /home.....	7
	<b>e.</b> /var.....	7
	<b>f.</b> Resultado de las particiones.....	8

---

## **SEGUNDA PARTE: CONFIGURACIÓN DE SERVIDOR WEB.**

---

<b>II.</b>	<b>INTRODUCCIÓN.....</b>	<b>10</b>
<b>III.</b>	<b>MARCO TEÓRICO.....</b>	<b>11</b>
	<b>3.1 Servidores Web.....</b>	<b>11</b>
	Modelo Cliente – Servidor.....	12
	<b>3.2 Servidor HTTP de Apache.....</b>	<b>13</b>

	a. Ficheros principales.....	14
	b. Secciones del archivo httpd.conf.....	15
	c. Características principales de Apache.....	15
<b>IV.</b>	<b>DISEÑO METODOLÓGICO.....</b>	<b>17</b>
	<b>4.1 IP asignada.....</b>	<b>17</b>
	<b>4.2 Configuración del servidor Http de Apache.....</b>	<b>17</b>
	1. Usando la herramienta de configuración de http.....	17
	2. Usando las directivas del fichero de Apache de http.....	17
	a. Pasos para configurar el fichero de las Directivas.....	18
	b. Archivo de configuración de Apache (http.conf).....	19
<hr/>		
<b>TERCERA PARTE:</b>		
<b>CONFIGURACIÓN DE SERVIDOR DE CORREO.</b>		
<b>V.</b>	<b>INTRODUCCIÓN.....</b>	<b>46</b>
<b>VI.</b>	<b>MARCO TEÓRICO.....</b>	<b>48</b>
	<b>6.1 Correo electrónico.....</b>	<b>48</b>
	<b>6.1.1</b> Atributos más característicos del correo electrónico.....	<b>48</b>
	a. Medio electrónico.....	48
	b. Medio asíncrono.....	48
	c. Medio ubicuo.....	48
	d. Medio digital.....	48
	e. Medio informático.....	49
	<b>6.1.2</b> Ventajas del Correo Electrónico.....	<b>49</b>
	<b>6.1.3</b> Dirección de Correo Electrónico.....	<b>49</b>
	<b>6.1.4</b> Composición de un Mensaje.....	<b>50</b>
	<b>6.1.5</b> Lectura del Correo Electrónico.....	<b>51</b>
	a. Correo Electrónico Remoto.....	51
	b. Correo Electrónico Local.....	52
	<b>6.1.6</b> Agentes para el funcionamiento del servicio de E-Mail.....	<b>52</b>
	a. MUA (Agente de Usuario de Correo).....	53
	b. MTA (Agente de Transferencia de Correo).....	53
	c. MDA (Agente de Entrega de Correo).....	53
	d. MSA (Agente de Registro de Correo).....	54
	e. MAA (Agente de Acceso al Correo).....	54
	<b>6.1.7</b> Protocolos del Correo Electrónico.....	<b>54</b>
	a. IMAP.....	55

b.	POP.....	55
c.	SMTP vs. X.400.....	57
d.	SMTP.....	58
<b>6.1.8</b>	Re-transmisión (Relaying).....	59
	Ejemplo de Retransmisión.....	60
<b>6.1.9</b>	Cómo llega un mensaje desde el punto A al punto B.....	62
<b>6.2</b>	<b>qmail</b> .....	63
<b>6.2.1</b>	Razones para utilizar qmail.....	64
a.	Seguridad.....	64
b.	Confiabilidad.....	65
c.	Eficiencia.....	65
d.	Simplicidad.....	65
<b>6.2.2</b>	Características de qmail.....	65
<b>6.2.3</b>	Cómo trabaja qmail.....	66
<b>6.2.4</b>	Comparación de qmail con otros MTA (Agentes de Transporte de Correo).....	66
<b>6.2.5</b>	Arquitectura de qmail.....	67
<b>6.2.6</b>	Requisitos del sistema para instalar qmail.....	68
<b>6.2.7</b>	Requisitos para usar qmail.....	69
<b>6.2.8</b>	Buzones usados por qmail.....	69
a.	Formato de buzón mbox.....	70
b.	Formato de buzón maildir.....	70
<b>6.3</b>	<b>Paquetes relacionados</b> .....	71
<b>6.3.1</b>	ucspi-tcp Server.....	71
<b>6.3.2</b>	PHP4.....	72
<b>6.3.3</b>	Squirrelmail.....	73
<b>6.3.4</b>	Daemontools.....	75
<b>6.3.5</b>	Dot-forward.....	76
<b>6.3.6</b>	Quotas en el Sistema.....	76
<b>6.3.7</b>	AntiSpam.....	78
a.	SpamAssassin.....	78
b.	Perl.....	80
c.	Digest-SHA1.....	81
d.	HTML-Parser.....	81
e.	ExtUtils-MakeMaker.....	82
f.	HTML::Tagset.....	82
<b>6.3.8</b>	Antivirus.....	82
a.	Clamav.....	83

	b. TNEF (Formato de Encapsulación Neutral de Transporte.).....	84
	c. Maildrop.....	85
<b>VII.</b>	<b>DISEÑO METODOLÓGICO.....</b>	<b>86</b>
	Obtención del código fuente.....	86
<b>7.1</b>	<b>Mailserver de la organización.....</b>	<b>87</b>
<b>7.2</b>	<b>Instalación de qmail.....</b>	<b>88</b>
	<b>7.2.1</b> Verificación del entorno de compilación.....	88
	<b>7.2.2</b> Crear directorio qmail.....	89
	<b>7.2.3</b> Grupos y Usuarios para qmail.....	89
	<b>7.2.4</b> Compilar y configurar el directorio de trabajo.....	89
	<b>7.2.5</b> Especificar el nombre del host.....	89
	<b>7.2.6</b> Establecer los "alias" del Sistema.....	90
	<b>7.2.7</b> Especificar el agente de procesamiento de correo.....	90
	<b>7.2.8</b> Eliminar Sendmail.....	90
	<b>7.2.9</b> Instalar el "reemplazo" de Sendmail.....	91
	<b>7.2.10</b> Instalar los Manuales.....	91
<b>7.3</b>	<b>Instalación de ucspi-tcp (tcpserver).....</b>	<b>92</b>
	<b>7.3.1</b> Descomprimir.....	92
	<b>7.3.2</b> Compilar e instalar tcpserver.....	92
	a. Compilar.....	92
	b. Copiar ejecutables.....	92
<b>7.4</b>	<b>Configurar el inicio automático de qmail.....</b>	<b>92</b>
	<b>7.4.1</b> Averiguar el UID y el GID del usuario "qmaild" y del grupo "nofiles".....	93
	<b>7.4.2</b> Añadir al final del archivo /etc/rc.d/rc.local.....	93
<b>7.5</b>	<b>Probando qmail.....</b>	<b>93</b>
	<b>7.5.1</b> Iniciando qmail.....	93
	<b>7.5.2</b> Probar el delivery.....	94
	<b>7.5.3</b> Probar la recepción.....	95
<b>7.6</b>	<b>Aceptar y remitir mensajes.....</b>	<b>95</b>
	<b>7.6.1</b> Qmail debe aceptar correos para el dominio.....	95
	<b>7.6.2</b> Enviar correo desde las estaciones.....	96
<b>7.7</b>	<b>Instalación del webmail (squirrelmail).....</b>	<b>96</b>
	<b>7.7.1</b> Instalar PHP4.....	96
	<b>7.7.2</b> Instalar Squirrelmail.....	97
<b>7.8</b>	<b>Instalación de daemontools.....</b>	<b>98</b>

	<b>7.9</b>	<b>Instalación de dot-forward.....</b>	<b>98</b>
	<b>7.10</b>	<b>Formato maildir.....</b>	<b>98</b>
	<b>7.11</b>	<b>Asignación de cuotas a los usuarios.....</b>	<b>100</b>
	<b>7.12</b>	<b>Instalación de Spamassassin (Antispam).....</b>	<b>102</b>
	<b>7.12.1</b>	Instalando Perl-5.7.8.....	102
	<b>7.12.2</b>	Instalar los Módulos.....	102
	<b>7.12.3</b>	Instalación de Spamassassin.....	103
	<b>7.12.4</b>	Instalar el script de Spamassassin.....	103
	<b>7.12.5</b>	Inicializar Spamassassin.....	104
	<b>7.12.6</b>	Crear el fichero .cf de Spamassassin.....	104
	<b>7.12.7</b>	Comprobar la instalación de Spamassassin.....	105
	<b>7.13</b>	<b>Instalación de Clamav (Antivirus).....</b>	<b>105</b>
	<b>7.13.1</b>	Instalar tnef.....	105
	<b>7.13.2</b>	Instalar maildrop.....	105
	<b>7.13.3</b>	Instalar clamav.....	105
		a. Descomprimirlo desde el explorador en /usr/local/src.	105
		b. Crear los grupos.....	105
		c. Instalar clamav.....	106
		d. Copiar el ejecutable.....	106
		e. Modificar los archivos.....	106
		f. Activar el archivo.....	107
		g. Verificar la instalación.....	107
<b>VIII.</b>			
<b>IX.</b>		<b>CONCLUSIÓN.....</b>	<b>108</b>
<b>X.</b>		<b>RECOMENDACIONES.....</b>	<b>109</b>
<b>XI.</b>		<b>GLOSARIO.....</b>	<b>110</b>
		<b>ANEXOS.....</b>	<b>124</b>
		<b>ANEXO 1: Directivas de Configuración del Fichero http.conf....</b>	<b>125</b>
<b>XII.</b>		<b>ANEXO 2: Interfaz del servidor de correo LIUKAR.....</b>	<b>151</b>
		<b>BIBLIOGRFÍA.....</b>	<b>161</b>

## INDICE DE TABLAS Y FIGURAS

### Lista de Tablas

<b>Tabla</b>	<b>Título</b>	<b>Página</b>
<b>1.1</b>	Requerimientos mínimos de instalación.....	6
<b>1.2</b>	Requerimientos mínimos recomendados.....	6
<b>1.3</b>	Características de la máquina utilizada en la configuración del servidor LIUKAR.....	6
<b>1.4</b>	Información detallada del disco duro.....	8
<b>3.1</b>	Tipos de servidores.....	12
<b>4.1</b>	Referencias específicas del servidor Web.....	44
<b>6.1</b>	qmail con otros MTA.....	66
<b>6.2</b>	Características de Squirrelmail.....	74
<b>6.3</b>	Características generales de SpamAssassin.....	78
<b>7.1</b>	Paquetes a utilizar en el servidor LIUKAR.....	86

## Lista de Figuras

<b>Figura</b>	<b>Título</b>	<b>Página</b>
<b>1.1</b>	Visión general del las particiones del disco duro.....	8
<b>3.1</b>	Arquitectura Cliente/Servidor.....	13
<b>6.1</b>	ISPs del mail Server.....	56
<b>6.2</b>	SMTP trabajando con el IMAP.....	59
<b>6.3</b>	El procedimiento SMTP establece y libera una conexión satisfactoria.....	61
<b>6.4</b>	Arquitectura de los archivos de qmail.....	67
<b>6.5</b>	Interfaz del webmail de Squirrelmail.....	73
<b>7.1</b>	Jerarquía del servidor LIUKAR.....	87
<b>7.2</b>	Fragmento del Archivo para obtener permiso de entradas/salidas remotas de los correos.....	88

# OBJETIVOS

## Objetivos Generales.

Para el correcto cumplimiento de la monografía se plantean dos objetivos generales que nos permiten establecer el límite de nuestro trabajo, siendo estos objetivos los siguientes:

- 1) Orientar un método de configuración para la implementación de un Servidor Web utilizando HTTP de Apache.
- 2) Indicar como configurar un Servidor de Correo Electrónico utilizando el agente de transporte de mensaje qmail.

## Objetivos Específicos.

Para complementar los objetivos generales es necesario individualizar los mismos para tener una visión más enfocada, originando los siguientes objetivos específicos:

- 1) Evaluar y seleccionar un Sistema Operativo adecuado para utilizar herramientas que permitan la implementación de servicios Web y el servidor de correo electrónico.
- 2) Explicar como configurar un servidor web utilizando el archivo de configuración ***/etc/httpd/conf/httpd.conf***.
- 3) Detallar los pasos para la configuración del servidor de correo electrónico usando qmail.
- 4) Poder enviar/recibir correo desde una máquina local o remota.
- 5) Facilitar herramientas para evitar el spam y virus en los correos que recibe nuestro servidor.

CONSIDERACIONES  
previas a las  
Configuraciones

## I. CONSIDERACIONES PREVIAS A LAS CONFIGURACIONES

Antes de configurar el servidor Web y el servidor de Correo, tuvimos en cuenta una serie de consideraciones previas en común tanto a nivel hardware como de software, las que detallaremos a continuación:

1. Selección del sistema operativo.
2. Selección de la máquina a usar.
3. Particiones del disco duro.

Pasamos a detallar cada una de dichas consideraciones.

### 1. Selección del sistema operativo.

La decisión del sistema operativo a utilizar es una elección importante en la etapa de configuración de un servidor web y de correo, para obtener resultados satisfactorios seleccionamos de entre tantos sistemas operativos que existen actualmente, un sistema operativo adecuado en el que se puede desarrollar este tipo de tecnología, eligiendo finalmente el sistema operativo **Red Hat Linux 8.0**.

#### Sistema Operativo Red Hat Linux 8.0

Linux es un sistema operativo de distribución libre desarrollado inicialmente por Linux Torvalds en la Universidad de Helsinki (Finlandia). Una comunidad de programadores expertos en UNÍX, distribuidos por Internet, han ayudado en el desarrollo, distribución y depuración de este sistema operativo. El núcleo de Linux no contiene código desarrollado por AT&T, ni por ninguna otra fuente propietaria.

La mayoría del software disponible en Linux ha sido desarrollado por el proyecto GNU de la free software foundation de Cambridge (Massachussets). Sin embargo, es toda la comunidad de programadores la que ha contribuido al desarrollo de aplicaciones para este sistema operativo. Con la aparición de los ordenadores personales potentes aparece Linux. Desde su origen hasta la actualidad han existido múltiples versiones.

Red Hat puso en el mercado la versión 8.0 de su sistema operativo de fuente abierta que ofrece una interfaz gráfica más agradable. La nueva interfaz de



Bluecurve, basado en Gnome 2,0, ofrece temas, barras, menús y muchas más nuevas opciones gráficas.

La nueva versión también contiene un buen número de aplicaciones actualizadas incluyendo la suite de fuente abierta para la oficina, Open Office, así como el cliente de E-mail Evolution, y el browser Mozilla 1.0.1.

Las nuevas utilidades y características del servidor de la versión Red Hat Linux 8.0 aportan versatilidad a redes pequeñas. Desde la selección de paquetes avanzados durante la instalación hasta una interfaz gráfica de usuario mejorada. También incluye documentación completa y servicios para ayudar, y sobre todo por que es un sistema *Gratuito*.

Red Hat es famoso en todo el mundo por los diferentes esfuerzos orientados a apoyar el movimiento del software libre. No sólo trabajan en el desarrollo de una de las distribuciones más populares de Linux, sino también en la comercialización de diferentes productos y servicios basados en software de código abierto.

## **Características, Ventajas y Desventajas de Linux**

LINUX nos proporciona una serie de características y ventajas que lo hacen diferenciarse del resto de los sistemas operativos, colocándolo como el favorito para muchos administradores de servidores.

### **a) Características.**

1. Es un clon del sistema operativo UNIX por tanto es Multitarea y Multiusuario.
2. Se puede correr la mayoría del software popular para UNIX, incluyendo el Sistema X-Window.
3. Cumple los estándares POSIX y de Sistemas Abiertos, esto es que tiene la capacidad de comunicarse con sistemas distintos a él.
4. Compatible con casi todos los sistemas UNIX.
5. Esta bajo la protección de la Licencia Pública General GNU o GPL.
6. Puede coexistir en entornos con otros sistemas operativos instalados.

### **b) Ventajas de Linux**

1. Precio. Es una implementación de UNIX sin costo.
2. Estabilidad excelente.
3. No pertenece a ningún propietario.
4. Soporta acceso remoto.
5. Soporte nativo de TCP/IP (Fácil conexión a Internet y otras redes).



6. Libre de virus, es muy difícil que sea infectado por virus.
7. Seguridad, es mucho más seguro que otros servidores.
8. Compatibilidad, reconoce la mayoría de los otros sistemas operativos en una red.
9. Velocidad, es mucho más veloz para realizar las tareas.
10. Posee el apoyo de miles de programadores a nivel mundial.
11. El paquete incluye el código fuente, lo que permite modificarlo de acuerdo a las necesidades del usuario.
12. Se puede usar en casi cualquier computadora, desde una 386.
13. Puede manejar múltiples procesadores. Incluso hasta 16 procesadores.
14. Maneja discos duros de hasta 16 TeraBytes.
15. Confiabilidad muy alta.

**c) Desventajas de Linux.**

1. Carencia de soporte técnico.
2. Inconvenientes de hardware, no soporta todas las plataformas, y no es compatible con algunas marcas específicas.

## **2. Selección de la máquina a usar.**

El análisis preciso respecto al hardware fue la parte principal, siendo el soporte del que dependen los tipos de instalaciones a usar para un buen resultado. Por tanto se tuvo en cuenta lo siguiente:

- ✓ Disco duro que cuente con suficiente espacio.
- ✓ Interfaces de red. (Obtención de una IP estática).
- ✓ Sistema de almacenamiento SCSI con soporte RAID.
- ✓ Memoria RAM de al menos 256 MB.
- ✓ Procesador dependiendo de si el contenido del sitio Web es mas bien dinámico o estático.

**Requerimientos de la Computadora.****Tabla 1.1. Requerimientos mínimos de instalación.**

Datos	Descripción
Procesador:	Pentium-class.
RAM:	32MB para modo texto; 128MB para modo gráfico.
Disco Duro:	650MB de espacio.
Monitor:	SVGA (1024x768) para ambiente gráfico.
CD-ROM:	12x o superior que soporte auto inicialización.

**Tabla 1.2. Requerimientos mínimos recomendados.**

Datos	Descripción
Procesador:	Pentium-class 200 MHz o superior.
RAM:	192MB para modo gráfico.
Disco Duro:	2.5GB de espacio; 4.5GB para instalación completa.
Monitor:	SVGA (1028x1024) para ambiente gráfico.
CD-ROM:	32x con auto inicialización.

**Tabla 1.3. Características de la máquina utilizada en la configuración del servidor LIUKAR.**

Datos	Descripción
Procesador	Intel Pentium 4, 1600 MHz (4 x 400)
Disco Duro	ST320410A (20 GB, 5400 RPM, Ultra-ATA/100)
Memoria del Sistema	128 MB (PC133 SDRAM)
Tarjeta de Red	Intel(R) PRO/100 VE Network Connection (172.16.0.223)
Monitor	HP D8899 (MX14450592)
Tarjeta Gráfica	RAGE 128 PRO GL (16 MB)
Tarjeta de Sonido	Intel 82801BA(M) ICH2X/M - AC'97 Audio Controller [C-0]
Puerto de Comunicación	COM1, COM2, LPT1
CD-ROM (Lector óptico)	LITEON CD-ROM LTN486S (48x CD-ROM)
Ratón	Mouse compatible PS/2

### 3. Particiones del disco duro.

En el proceso de preparación para la instalación de Red Hat Linux, tuvimos que considerar el número y el tamaño de las particiones que serán utilizadas por el nuevo sistema operativo. Creamos las particiones siguientes:

#### **A) Partición swap:**

Las particiones swap son utilizadas para soportar la memoria virtual. En otras palabras, los datos son escritos en la swap cuando no hay bastante memoria disponible para contener los datos que su ordenador está procesando. Si su ordenador tiene 16 Megas de RAM o incluso menos, tiene que crear una partición swap. También si tiene más memoria, se recomienda la utilización de una partición swap. El tamaño mínimo para una partición de swap tendría que ser igual a la cantidad de memoria RAM presente en su ordenador, o por lo menos 16MB (entre las dos se aconseja elegir la cantidad mayor).

#### **B) Partición /boot:**

La partición que se crea bajo /boot contiene el kernel del sistema operativo (que permite el arranque de su sistema con Red Hat Linux), junto con algunos ficheros utilizados durante el proceso de arranque. Debido a las limitaciones de la mayoría de BIOSes de PCs, es una buena idea crear una partición pequeña para estos ficheros. Esta partición no debería superar los 32 MB.

#### **C) Partición root (/):**

La partición root es donde se encuentra / (el directorio de root). En esta configuración de las particiones, todos los ficheros (excepto los que residen en /boot) están en la partición de root. Por ello sería una buena elección hacer lo más grande posible el tamaño de su partición de root. Una partición root de 1.2 GB es equivalente a la que es ubicada por una instalación de clase estación de trabajo (con poquísimo espacio libre), mientras que una partición root de 2.4 GB le permitirá instalar todos los paquetes. Es obvio que cuanto más espacio pueda darle a la partición root mejor.

#### **D) Partición /home:**

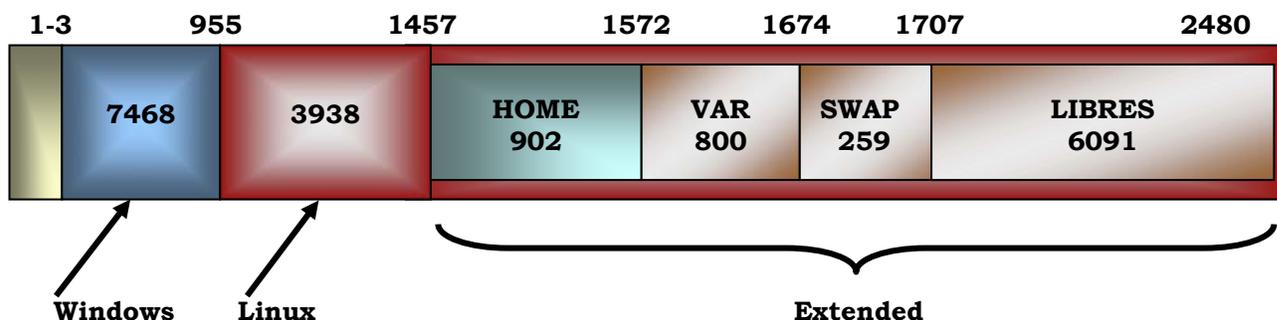
Directorio de todos los usuarios.

#### **E) Partición /var:**

Datos volátil y directorios spooling.

## F) Resultado de las particiones.

En vista que nuestro proyecto fue desarrollado en una de las máquinas del laboratorio No. 1 del departamento de computación, se tuvo que dividir el disco en dos particiones principales. La primera destinada para el sistema operativo Windows y la segunda parte para el sistema operativo Red Hat Linux 8.0, en esta segunda partición se realizaron otras particiones, como muestra la figura 1.1.



**Figura 1.1** Visión general de las particiones del disco duro.

**Tabla 1.4.** Información detallada del disco duro.

Disco Duro	Inicio	Fin	Size (MB)	Tipo Ubicación	Punto de Montaje
/dev/hda					
hda1	1	2	16	Fat	
hda2	3	954	7468	ntfs	Windows
hda3	955	1456	3938	Ext3	Linux /
hda4	1457	2480	8032	Extended	
hda5	1457	1571	902	Ext3	Home
hda6	1572	1673	800	Ext3	Var
hda7	1674	1706	259	Linux-Swap	Swap
	1707	2480	6091	Free Space	

**NOTA:** Muestra desde el entorno de Red Hat Linux la información detallada de cada partición.



# configuración de servidor web

## II. INTRODUCCIÓN

El desarrollo tecnológico ha evolucionado de una manera increíble hasta el punto que tanto empresas (públicas y privadas) u otras instituciones se ven obligadas a recurrir a la microinformática y a las redes de ámbito mundial, que interconectan recursos informáticos de todo tipo, con el objetivo de agilizar su trabajo y satisfacer sus necesidades.

La implementación de servidores Web es la tecnología que mejor cumple con estas demandas. Estos son designados para recibir solicitudes anónimas desde auténticos hosts en la Internet y a liberar las solicitudes de información en una manera rápida y eficiente. De tal forma, ellos proveen un portal que puede ser utilizado para beneficio propio y para ofrecer un servicio al cliente.

Considerando que este tema es de gran alcance, decidimos configurar un servidor web teniendo a **Red Hat Linux 8.0** como sistema operativo, ya que es estable, seguro y permite conseguir fácilmente buenos resultados con máquinas poco potentes como la nuestra y superar a otros sistemas operativos en ordenadores similares, siendo fácil de usar y adaptable a nuestros propósitos.

El software que elegimos para nuestras prácticas y para configurar el Servidor de Web HTTP es el servidor de libre distribución **Apache 2.0.4.18-14**, ya que disponemos del código fuente, configurando así el fichero ***/etc/httpd/conf/httpd.conf*** para brindar atención a las peticiones que realizaran los usuarios desde nuestra página web.

### III. MARCO TEÓRICO

Los servidores web han sido el eje principal para dar origen a muchos otros servidores, ya que permiten la comunicación de una manera más fluida y directa a los usuarios que recurren a él.

#### 3.1 Servidores web.

Los Servidores Web son aquellos que permiten a los clientes compartir datos, documentos y multimedia en formato Web.

Los Servidores Web suministran páginas Web a los navegadores (como por ejemplo, Netscape Navigator, Internet Explorer de Microsoft) que lo solicitan. En términos más técnicos, los servidores Web soportan el Protocolo de Transferencia de Hipertexto conocido como HTTP (HyperText Transfer Protocol), el estándar de Internet para comunicaciones Web. Usando HTTP, un servidor Web envía páginas Web en HTML y CGI, así como otros tipos de scripts a los navegadores o browsers cuando éstos lo requieren.

Cuando un usuario hace clic sobre un enlace (link) a una página Web, se envía una solicitud al servidor Web para localizar los datos nombrados por ese enlace.

El servidor Web recibe esta solicitud y suministra los datos que le han sido solicitados (una página HTML, un script interactivo, una página Web generada dinámicamente desde una base de datos, etc.) o bien devuelve un mensaje de error.

Los servidores web son parte de la tecnología Cliente-Servidor, el servidor Web aporta algunas ventajas adicionales; como acceso más simple a la información (con un simple clic).

En la tabla 3.1 se muestran los servidores que existen en la actualidad y que son utilizados para satisfacer las necesidades de los diferentes usuarios.

**Tabla 3.1. Tipos de Servidores.**

<b>Servidores</b>	<b>Descripción</b>
Archivos	Almacena archivos y aplicaciones de productividad como por ejemplo procesadores de texto, hojas de cálculo.
Bases de datos	Las bases de datos, tablas, índices. Es uno de los servidores que más carga tiene.
Transacciones	Cumple o procesa todas las transacciones. Valida primero y recién genera un pedido al servidor de bases de datos.
Groupware	Utilizado para el seguimiento de operaciones dentro de la red.
Objetos	Contienen objetos que deben estar fuera del servidor de bases de datos. Estos objetos pueden ser videos, imágenes, objetos multimedia en general.
Web	Se usan como una forma inteligente para comunicación entre empresas a través de Internet. Este servidor permite transacciones con el acondicionamiento de un browser específico.

**Modelo Cliente – Servidor.**

En el sentido más estricto, el término **cliente/servidor** describe un sistema en el que una máquina cliente solicita a una segunda máquina llamada servidor que se encarga de darle respuesta (cliente y servidor pueden coincidir físicamente en el mismo ordenador).

El Cliente cumple dos funciones:

1. Gestiona la comunicación con el servidor, solicita un servicio y recibe los datos enviados por aquél.
2. Maneja la interfaz con el usuario: presenta los datos en el formato adecuado y brinda las herramientas y comandos necesarios para que el usuario pueda utilizar las prestaciones del servidor de forma sencilla.

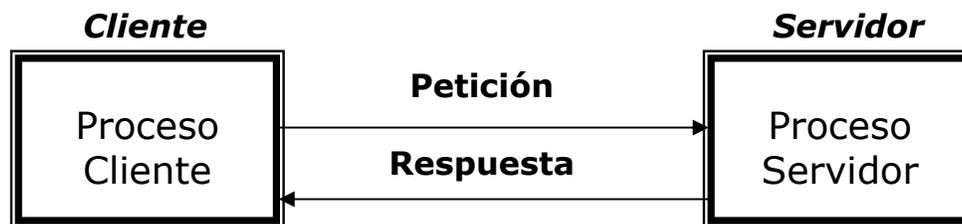
El programa Servidor cumple con la siguiente función:

Sólo tiene que encargarse de transmitir la información de forma eficiente. No tiene que atender al usuario. De esta forma un mismo servidor puede atender a varios clientes al mismo tiempo. Pero un servidor puede solicitar servicio a otro servidor convirtiéndose así en cliente.

El servidor tiene dos características:

1. Un Servidor utiliza el protocolo TCP, tenemos una comunicación orientada a conexión.
2. Un servidor puede a su vez actuar como cliente para atender una petición.

En la figura 3.1 se aprecia la arquitectura cliente/servidor que es el modelo más utilizado tanto para aplicaciones web como para proporcionar servicios de alto nivel o páginas web dinámica.



**Figura 3.1 Arquitectura Cliente/Servidor.**

Este tipo de arquitectura consta de tres componentes:

1. El proceso el cual inicia el dialogo o la solicitud de un servicio se le denomina cliente.
2. El proceso que atiende las solicitudes realizadas por los clientes se le denomina Servidor, las cuales son mas poderosas que los clientes.
3. El medio físico mediante el cual se comunican los clientes con el servidor, estos generalmente son redes de área local (LAN) o redes de área amplia (WAN).

#### **EJEMPLO:**

La mayoría de servidores añaden algún nivel de seguridad a sus tareas. Por ejemplo, al ir a una página y el navegador presenta una ventana de diálogo que pregunta su nombre de usuario y contraseña, ha encontrado una página protegida por contraseña. El dueño o el administrador del servidor maneja una lista de nombres y contraseñas de las personas que se encuentran registradas en dicho servidor, el cual permite que sólo esas personas tengan acceso a él.

### **3.2 Servidor HTTP de apache.**

Linux Red Hat 8.0 también incluye una suite de herramientas de configuración para configurar diversos servicios del sistema incluyendo los servidores de Apache, Samba, ajustes de la red, firewall y los periféricos.

La compañía también ha incluido versiones mejoradas de su compilador de C y del kernel del sistema operativo.

Apache surgió a partir del servidor de HTTP más famoso y difundido en su época: NCSA. Desde entonces se convirtió en un poderoso rival de todos los servidores Unix utilizados hasta la fecha por su eficiencia, funcionalidad y rapidez. Es por ello que se conoce como el "**rey de los servidores Web**".

El paquete de la distribución Red Hat que contiene la implementación del servidor Apache para Linux se nombra *Apache*. Este se ejecuta a través de un *daemon* llamado **httpd** que se manipula utilizando el *script* de inicio del mismo nombre en `/etc/rc.d/init.d/`

El servidor Web Apache está diseñado de forma modular; consiste en muchas porciones de código que hacen referencia a diferentes aspectos o funcionalidades del servidor Web. Esta modularidad es intencionada, con lo cual, cada desarrollador puede escribir su propia porción de código para cubrir una necesidad en particular. Su código, llamado módulo, puede ser integrado en el servidor Web con relativa facilidad.

### **a) Ficheros principales.**

El servidor lee la información de tres ficheros con directivas de configuración. Cualquier directiva puede aparecer en cualquiera de estos ficheros. Los nombres de estos ficheros son relativos a la raíz de instalación del servidor, la cual es definida por la directiva `ServerRoot`.

#### **conf/httpd.conf:**

Contiene las directivas que controlan el funcionamiento del demonio servidor (`httpd`).

#### **conf/srm.conf:**

Contiene las directivas que controlan la especificación de documentos que el servidor proporciona a los clientes.

#### **conf/access.conf:**

Contiene directivas que controlan el acceso a los documentos.

El servidor también leerá un fichero conteniendo los tipos mime de los documentos. Por defecto el fichero será `conf/mime.types`.

### **b) Secciones del archivo httpd.conf.**

Pero solo se utilizará como fichero principal del servidor Web a `conf/httpd.conf`, que esta agrupado con las directivas de configuración. El cual se divide en tres secciones con fines puramente organizativos:



---

- **Sección 1:**

Reúne los aspectos globales del servidor. Por ejemplo: el número máximo de clientes concurrentes, los timeouts, el directorio raíz del servidor, etc.

- **Sección 2:**

Agrupar las directivas que definen la forma de responder a todos los pedidos del servidor principal, o sea aquellos que no son para los *hosts* virtuales, de existir alguno definido. También reúne los aspectos por defecto de todos los *hosts* virtuales que se configuren más adelante.

- **Sección 3:**

Agrupar las directivas relacionadas con los *hosts* virtuales que se definan.

### c) Características principales de Apache.

1. Apache significa "A PAtCHy sErver", o sea se basa en un código y un conjunto de ficheros "parches". Otros desarrolladores relacionan su nombre con el de las tribus nativas americanas de Apaches.
2. Es un servidor *Web* potente, flexible y ajustado al HTTP/1.1
3. Es altamente configurable y extensible.
4. Puede ser ajustado a través de la definición de módulos empleando su propio API (*Application Programming Interface*).
5. Provee todo su código fuente de forma libre y se distribuye bajo una licencia no restrictiva.
6. Se ejecuta en diversas plataformas operativas tales como: Windows 9x/NT, Macintosh, Novell NetWare, OS/2, Linux y la mayoría de los Unix existentes: IRIX, Solaris, HPUX, SCO, FreeBSD, NetBSD, AIX, Digital Unix, etc.
7. Se desarrolla de forma acelerada estimulando la retroalimentación desde sus usuarios a través de nuevas ideas, reportes de errores y parches.
8. Implementa muchas posibilidades frecuentemente demandadas, tales como:
  - 8.1 Bases de datos DBM para autenticación. Permiten establecer fácilmente la protección de documentos a través de *passwords* para una gran cantidad de usuarios sin dañar el funcionamiento del servidor.



- 8.2** Respuestas adaptables a los errores o problemas. Se pueden definir ficheros o *scripts* de tipo CGI que respondan ante la ocurrencia de errores internos o en las solicitudes realizadas.
- 8.3** Directiva para definir múltiples índices. Se utiliza cuando se solicitan directorios por parte de los clientes a partir de lo cual se puede buscar en estos y devolver un documento índice cuyo nombre puede ser por ejemplo: `index.html`, `index.cgi` o `default.html`.
- 8.4** Ilimitadas y flexibles posibilidades de redireccionamiento y definición de alias para los URLs. Apache no tiene un límite establecido para definir alias y redireccionamientos que pueden ser declarados en sus ficheros de configuración.
- 8.5** Negociación del contenido de las respuestas. Apache es capaz de ofrecer la mejor representación de la información accedida de acuerdo con las capacidades del cliente solicitante.
- 8.6** Soporte de *hosts* virtuales. Es la habilidad del servidor de distinguir entre los pedidos hechos a diferentes direcciones IP o nombres de dominio definidos en la misma máquina.
- 8.7** Configuración flexible de las trazas generadas. Es posible adaptar el formato de las trazas obtenidas así como redireccionarlas a través de tuberías (Unix) en aras de filtrarlas. De esta forma se puede lograr por ejemplo dividir dinámicamente las trazas de los *hosts* virtuales en distintos ficheros.

## IV. DISEÑO METODOLÓGICO

Luego de tener el sistema operativo instalado, procedemos con la respectiva configuración del servidor web el cual lleva por nombre LIUKAR, logrando así integrarnos a la red de Internet a través de nuestro propio sitio web, para ello realizamos lo siguiente:

### 4.1 IP asignada.

Como paso inicial para lograr la configuración del Server Web fue necesario tener asignada una IP pública, la IP con la que trabajaremos a lo largo de nuestra configuración sera:

IP Externa: 192.107.104.133

IP Interna : 172.16.4.244

### 4.2 Configuración del servidor HTTP de Apache.

Configuramos el Servidor HTTP de Apache de la versión 8.0, actualmente este paquete es llamado HTTP. Las opciones de configuración han cambiado, hoy en día existen dos maneras para configurar el HTTP. Nosotras optamos por la segunda.

#### 1. Usando la herramienta de configuración de HTTP.

Aquí la configuración se hace mas fácil ya que solo se siguen las instrucciones que se presentan en la interfaz.

#### 2. Usando las directivas del fichero de Apache de HTTP.

En la actualidad el fichero de configuración principal de Apache, y el único que se debe modificar de los existentes, se nombra `httpd.conf`. Los ficheros de configuración agrupados en este directorio están formados por un conjunto de directivas que regulan el comportamiento del servidor.

El fichero de configuración del servidor Web Apache es `/etc/httpd/conf/httpd.conf` para su servidor "Servidor HTTP de Apache". El fichero `httpd.conf` está bien comentado y es bastante autoexplicativo. La configuración predeterminada del Red Hat Linux Advanced Server funciona para los ordenadores de la mayoría de los usuarios, así que no cambiamos ninguna de las directivas en el fichero `httpd.conf`.

Los ficheros vacíos **srm.conf** y **access.conf** se encuentran en el directorio **/etc/httpd/conf**. Precisamente estos ficheros junto con el fichero **httpd.conf** se utilizaron anteriormente como ficheros de configuración de Apache.

Además de los ficheros de configuración mencionados, el acceso a cada directorio del servidor se puede regular en un fichero, contenido en el mismo directorio o en sus antecesores, cuyo nombre por defecto es **.htaccess**, y que también puede ser modificado mediante la configuración.

### **A. Pasos para configurar el fichero de las Directivas:**

1. Tipo de instalación "Servidor", muy adecuada para nuestro caso.
2. Buscamos el archivo `/etc/httpd/conf/httpd.conf`.
3. Copiamos el fichero original dándole por ejemplo, el nombre `httpd.confold` u otro cualquiera, con el fin de no perder el original en caso de posibles errores.
4. Modificamos el fichero `httpd.conf`.
5. Reiniciamos o apagamos y arrancamos el proceso del comando `httpd`.
6. Probamos si el fichero funciona desde la shell con la siguiente orden:  
`# httpd start`

Durante el proceso de configuración nos encontramos con algunos errores, lo que impidió que nuestro servidor web funcionara correctamente. Para resolver esta situación tuvimos que realizar lo siguiente:

1. Revisamos lo que se acababa de modificar en el archivo `httpd.conf`. para tener la certeza de no haber cometido ningún error.
2. Se consultó el fichero de conexión de error (`/var/log/httpd/error_log`). Este puede ser difícil de interpretar, todo depende del nivel de experiencia.
3. Al final encontramos el problema, por lo que revisamos las últimas entradas que nos ayudaron a saber lo que había ocurrido, resolviendo estos inconvenientes de forma satisfactoria.



Si crea una copia de seguridad, podrá recuperar el sistema de posibles errores cometidos antes de editar el nuevo fichero de configuración.

## B. Archivo de configuración de Apache (httpd.conf).

En nuestro caso decidimos configurar el fichero donde se encuentran las directivas, las que nos permiten adaptar el servidor a nuestras necesidades quedando el archivo de la siguiente manera:

```
## httpd.conf - configuration for the Apache web server
#
# Generated automatically... if you edit manually, the changes will be lost
# the next time you run "apacheconfig".
#

### Section 1: Global Environment
#
# The directives in this section affect the overall operation of Apache,
# such as the number of concurrent requests it can handle or where it
# can find its configuration files.
#

# Don't give away too much information about all the subcomponents
# we are running. Comment out this line if you don't mind remote sites
# finding out what major optional modules you are running

ServerTokens OS

#
# ServerRoot: The top of the directory tree under which the server's
# configuration, error, and log files are kept.
#
# NOTE! If you intend to place this on an NFS (or otherwise network)
# mounted filesystem then please read the LockFile documentation
# (available at <URL:http://httpd.apache.org/docs-
# 2.0/mod/core.html#lockfile>);
# you will save yourself a lot of trouble.
#
# Do NOT add a slash at the end of the directory path.
#

ServerRoot "/etc/httpd"

#
```



```
# ScoreBoardFile: File used to store internal server process information.
# If unspecified (the default), the scoreboard will be stored in an
# anonymous shared memory segment, and will be unavailable to third-
#party applications.
# If specified, ensure that no two invocations of Apache share the same
# scoreboard file. The scoreboard file MUST BE STORED ON A LOCAL DISK.
# ScoreBoardFile "/var/run/httpd.scoreboard"

# PidFile: The file in which the server should record its process
# identification number when it starts.
#
PidFile run/httpd.pid

#
# Timeout: The number of seconds before receives and sends time out.
#
TimeOut 300

#
# KeepAlive: Whether or not to allow persistent connections (more than
# one request per connection). Set to "Off" to deactivate.
#
KeepAlive Off

#
# MaxKeepAliveRequests: The maximum number of requests to allow
# during a persistent connection. Set to 0 to allow an unlimited amount.
# We recommend you leave this number high, for maximum performance.
#
MaxKeepAliveRequests 100

#
# KeepAliveTimeout: Number of seconds to wait for the next request from
#the same client on the same connection.
#
KeepAliveTimeout 15

##
## Server-Pool Size Regulation (MPM specific)
##
```

```
# prefork MPM
# StartServers: number of server processes to start
# MinSpareServers: minimum number of server processes which are kept
# spare
# MaxSpareServers: maximum number of server processes which are kept
# spare
# MaxClients: maximum number of server processes allowed to start
# MaxRequestsPerChild: maximum number of requests a server process
# serves

<IfModule prefork.c>
    StartServers 8
    MinSpareServers 5
    MaxSpareServers 20
    MaxClients 150
    MaxRequestsPerChild 100
</IfModule>

# worker MPM
# StartServers: initial number of server processes to start
# MaxClients: maximum number of simultaneous client connections
# MinSpareThreads: minimum number of worker threads which are kept
# spare
# MaxSpareThreads: maximum number of worker threads which are kept
# spare
# ThreadsPerChild: constant number of worker threads in each server
# process
# MaxRequestsPerChild: maximum number of requests a server process
# serves

<IfModule worker.c>
    StartServers      2
    MaxClients        150
    MinSpareThreads   25
    MaxSpareThreads   75
    ThreadsPerChild   25
    MaxRequestsPerChild 0
</IfModule>

# perchild MPM
# NumServers: constant number of server processes
# StartThreads: initial number of worker threads in each server process
# MinSpareThreads: minimum number of worker threads which are kept
# spare
```

```
# MaxSpareThreads: maximum number of worker threads which are kept
# spare
# MaxThreadsPerChild: maximum number of worker threads in each server
# process
# MaxRequestsPerChild: maximum number of connections per server
# process

<IfModule perchild.c>
    NumServers      5
    StartThreads    5
    MinSpareThreads 5
    MaxSpareThreads 10
    MaxThreadsPerChild 20
    MaxRequestsPerChild 0
</IfModule>

#
# Listen: Allows you to bind Apache to specific IP addresses and/or
# ports, in addition to the default. See also the <VirtualHost>
# directive.
#
# Change this to Listen on specific IP addresses as shown below to
# prevent Apache from glomming onto all bound IP addresses (0.0.0.0)
# Listen 12.34.56.78:80

Listen 172.16.4.244:80

#
# Load config files from the config directory "/etc/httpd/conf.d".
#

Include conf.d/*.conf

#
# Dynamic Shared Object (DSO) Support
#
# To be able use the functionality of a module which was built as a DSO you
# have to place corresponding `LoadModule' lines at this location so the
# directives contained in it are actually available _before_ they are used.
# Statically compiled modules (those listed by `httpd -l') do not need
# to be loaded here.
#
# Example:
# LoadModule foo_module modules/mod_foo.so
#
```

```
LoadModule access_module modules/mod_access.so
LoadModule auth_module modules/mod_auth.so
LoadModule auth_anon_module modules/mod_auth_anon.so
LoadModule auth_dbm_module modules/mod_auth_dbm.so
LoadModule auth_digest_module modules/mod_auth_digest.so
LoadModule include_module modules/mod_include.so
LoadModule log_config_module modules/mod_log_config.so
LoadModule env_module modules/mod_env.so
LoadModule mime_magic_module modules/mod_mime_magic.so
LoadModule cern_meta_module modules/mod_cern_meta.so
LoadModule expires_module modules/mod_expires.so
LoadModule headers_module modules/mod_headers.so
LoadModule usertrack_module modules/mod_usertrack.so
LoadModule unique_id_module modules/mod_unique_id.so
LoadModule setenvif_module modules/mod_setenvif.so
LoadModule mime_module modules/mod_mime.so
LoadModule dav_module modules/mod_dav.so
LoadModule status_module modules/mod_status.so
LoadModule autoindex_module modules/mod_autoindex.so
LoadModule asis_module modules/mod_asis.so
LoadModule info_module modules/mod_info.so
LoadModule cgi_module modules/mod_cgi.so
LoadModule dav_fs_module modules/mod_dav_fs.so
LoadModule vhost_alias_module modules/mod_vhost_alias.so
LoadModule negotiation_module modules/mod_negotiation.so
LoadModule dir_module modules/mod_dir.so
LoadModule imap_module modules/mod_imap.so
LoadModule actions_module modules/mod_actions.so
LoadModule speling_module modules/mod_speling.so
LoadModule userdir_module modules/mod_userdir.so
LoadModule alias_module modules/mod_alias.so
LoadModule rewrite_module modules/mod_rewrite.so
LoadModule proxy_module modules/mod_proxy.so
LoadModule proxy_ftp_module modules/mod_proxy_ftp.so
LoadModule proxy_http_module modules/mod_proxy_http.so
LoadModule proxy_connect_module modules/mod_proxy_connect.so

#
# ExtendedStatus controls whether Apache will generate "full" status
# information (ExtendedStatus On) or just basic information
(ExtendedStatus
# Off) when the "server-status" handler is called. The default is Off.
#
#ExtendedStatus On
```



```
### Section 2: 'Main' server configuration
#
# The directives in this section set up the values used by the 'main'
# server, which responds to any requests that aren't handled by a
# <VirtualHost> definition.  These values also provide defaults for
# any <VirtualHost> containers you may define later in the file.
#
# All of these directives may appear inside <VirtualHost> containers,
# in which case these default settings will be overridden for the
# virtual host being defined.
#
# If you wish httpd to run as a different user or group, you must run
# httpd as root initially and it will switch.
#
# User/Group: The name (or #number) of the user/group to run httpd as.
# . On SCO (ODT 3) use "User nouser" and "Group nogroup".
# . On HPUX you may not be able to use shared memory as nobody, and
the
#   suggested workaround is to create a user www and use that user.
# NOTE that some kernels refuse to setgid(Group) or semctl(IPC_SET)
# when the value of (unsigned)Group is above 60000;
# don't use Group #-1 on these systems!
#
User apache
Group apache
#
# ServerAdmin: Your address, where problems with the server should be
# e-mailed.  This address appears on some server-generated pages, such
# as error documents.  e.g. admin@your-domain.com
#
ServerAdmin rina@unanleon.edu.ni
#
# ServerName gives the name and port that the server uses to identify itself.
# This can often be determined automatically, but we recommend you
specify
# it explicitly to prevent problems during startup.
#
# If this is not set to valid DNS name for your host, server-generated
# redirections will not work.  See also the UseCanonicalName directive.
```

```
#
# If your host doesn't have a registered DNS name, enter its IP address
# here.
# You will have to access it by its address anyway, and this will make
# redirections work in a sensible way.
#
ServerName isi.unanleon.edu.ni:80

#
# UseCanonicalName: Determines how Apache constructs self-referencing
# URLs and the SERVER_NAME and SERVER_PORT variables.
# When set "Off", Apache will use the Hostname and Port supplied
# by the client. When set "On", Apache will use the value of the
# ServerName directive.
#
UseCanonicalName Off

#
# DocumentRoot: The directory out of which you will serve your
# documents. By default, all requests are taken from this directory, but
# symbolic links and aliases may be used to point to other locations.
#
DocumentRoot "/var/www/html/mail/squirrelmail-1.5.0"

#
# Disable autoindex for the root directory, and present a
# default Welcome page if no other index page is present.
#
<Directory />
    Options FollowSymLinks
    AllowOverride None
</Directory>

<Directory "/var/www/html">
    Options Indexes FollowSymLinks
    Order allow,deny
    Allow from all
</Directory>

<LocationMatch "^/$">
```

```
Options -Indexes
ErrorDocument 403 /error/noindex.html
</LocationMatch>

#
# UserDir: The name of the directory that is appended onto a user's home
# directory if a ~user request is received.
#
# The path to the end user account 'public_html' directory must be
# accessible to the webserver userid. This usually means that ~userid
# must have permissions of 711, ~userid/public_html must have
permissions
# of 755, and documents contained therein must be world-readable.
# Otherwise, the client will only receive a "403 Forbidden" message.
#
# See also: http://httpd.apache.org/docs/misc/FAQ.html#forbidden
#
<IfModule mod_userdir.c>
#
# UserDir is disabled by default since it can confirm the presence
# of a username on the system (depending on home directory
# permissions).
#
#UserDir "disable"

#
# To enable requests to ~/user/ to serve the user's public_html
# directory, use this directive instead of "UserDir disable":
#
UserDir pub_html

</IfModule>

#
# DirectoryIndex: sets the file that Apache will serve if a directory
# is requested.
#
# The index.html.var file (a type-map) is used to deliver content-
# negotiated documents. The MultiViews Option can be used for the
# same purpose, but it is much slower.
#

<Directory /home/*/pub_html>
AllowOverride FileInfo AuthConfig Limit
Options MultiViews Indexes SymLinksIfOwnerMatch IncludesNoExec
```

```
<Limit GET POST OPTIONS>
    Order allow,deny
    Allow from all
</Limit>
<LimitExcept GET POST OPTIONS>
    Order deny,allow
    Deny from all
</LimitExcept>
</Directory>
```

DirectoryIndex index.htm index.html.var index.php index.html

```
#
# AccessFileName: The name of the file to look for in each directory
# for access control information. See also the AllowOverride directive.
#
AccessFileName .htaccess
```

```
#
# The following lines prevent .htaccess and .htpasswd files from being
# viewed by Web clients.
#
```

```
<Files ~ "\.ht">
    Order allow,deny
    Deny from all
</Files>
```

```
#
# TypesConfig describes where the mime.types file (or equivalent) is
# to be found.
#
TypesConfig /etc/mime.types
```

```
#
# DefaultType is the default MIME type the server will use for a document
# if it cannot otherwise determine one, such as from filename extensions.
# If your server contains mostly text or HTML documents, "text/plain" is
# a good value. If most of your content is binary, such as applications
# or images, you may want to use "application/octet-stream" instead to
# keep browsers from trying to display binary files as though they are
# text.
#
```

```
DefaultType text/plain
```

```
#  
# The mod_mime_magic module allows the server to use various hints from  
the  
# contents of the file itself to determine its type. The MIMEMagicFile  
# directive tells the module where the hint definitions are located.  
#
```

```
<IfModule mod_mime_magic.c>  
# MIMEMagicFile /usr/share/magic.mime  
  MIMEMagicFile conf/magic  
</IfModule>
```

```
#  
# HostnameLookups: Log the names of clients or just their IP addresses  
# e.g., www.apache.org (on) or 204.62.129.132 (off).  
# The default is off because it'd be overall better for the net if people  
# had to knowingly turn this feature on, since enabling it means that  
# each client request will result in AT LEAST one lookup request to the  
# nameserver.  
#
```

```
HostNameLookups Off
```

```
#  
# ErrorLog: The location of the error log file.  
# If you do not specify an ErrorLog directive within a <VirtualHost>  
# container, error messages relating to that virtual host will be  
# logged here. If you *do* define an error logfile for a <VirtualHost>  
# container, that host's errors will be logged there and not here.  
#
```

```
ErrorLog logs/error_log
```

```
#  
# LogLevel: Control the number of messages logged to the error_log.  
# Possible values include: debug, info, notice, warn, error, crit,  
# alert, emerg.  
#
```

```
LogLevel warn
```

```
#  
# The following directives define some format nicknames for use with
```

```
# a CustomLog directive (see below).
#

LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\"" combined
LogFormat "%h %l %u %t \"%r\" %>s %b" common
LogFormat "%{Referer}i -> %U" referer
LogFormat "%{User-agent}i" agent

#
# The location and format of the access logfile (Common Logfile Format).
# If you do not define any access logfiles within a <VirtualHost>
# container, they will be logged here. Contrariwise, if you *do*
# define per-<VirtualHost> access logfiles, transactions will be
# logged therein and *not* in this file.
#
# CustomLog logs/access_log common

CustomLog logs/access_log combined

#
# If you would like to have agent and referer logfiles, uncomment the
# following directives.
#
#CustomLog logs/referer_log referer
#CustomLog logs/agent_log agent

#
# If you prefer a single logfile with access, agent, and referer information
# (Combined Logfile Format) you can use the following directive.
#
#CustomLog logs/access_log combined

#
# Optionally add a line containing the server version and virtual host
# name to server-generated pages (error documents, FTP directory listings,
# mod_status and mod_info output etc., but not CGI generated documents).
# Set to "EMail" to also include a mailto: link to the ServerAdmin.
# Set to one of: On | Off | EMail
#

ServerSignature On

#
# Aliases: Add here as many aliases as you need (with no limit). The format
```

```
is
# Alias fakename realname
#
# Note that if you include a trailing / on fakename then the server will
# require it to be present in the URL. So "/icons" isn't aliased in this
# example, only "/icons/". If the fakename is slash-terminated, then the
# realname must also be slash terminated, and if the fakename omits the
# trailing slash, the realname must also omit it.
#
# We include the /icons/ alias for FancyIndexed directory listings. If you
# do not use FancyIndexing, you may comment this out.
#
Alias /icons/ "/var/www/icons/"

<Directory />
    Options Indexes MultiViews
    AllowOverride None
    Order allow,deny
    Allow from all
</Directory>

#
# This should be changed to the ServerRoot/manual/. The alias provides
# the manual, even if you choose to move your DocumentRoot. You may
comment
# this out if you do not care for the documentation.
#
#Alias /manual "/var/www/manual"

<IfModule mod_dav_fs.c>
    # Location of the WebDAV lock database.
    DAVLockDB /var/lib/dav/lockdb
</IfModule>

#
# ScriptAlias: This controls which directories contain server scripts.
# ScriptAliases are essentially the same as Aliases, except that
# documents in the realname directory are treated as applications and
# run by the server when requested rather than as documents sent to the
client.
# The same rules about trailing "/" apply to ScriptAlias directives as to
# Alias.
#ScriptAlias /cgi-bin/ "/var/www/cgi-bin/"

#<IfModule mod_cgid.c>
```

```
#
# Additional to mod_cgid.c settings, mod_cgid has Scriptsock <path>
# for setting UNIX socket for communicating with cgid.
#
#Scriptsock          logs/cgisock
#</IfModule>

#
# Redirect allows you to tell clients about documents which used to exist in
# your server's namespace, but do not anymore. This allows you to tell the
# clients where to look for the relocated document.
# Example:
# Redirect permanent /foo http://www.example.com/bar

#
# Directives controlling the display of server-generated directory listings.
#
#
# FancyIndexing is whether you want fancy directory indexing or standard.
# VersionSort is whether files containing version numbers should be
# compared in the natural way, so that `apache-1.3.9.tar' is placed before
# `apache-1.3.12.tar'.
#
IndexOptions FancyIndexing VersionSort NameWidth=*

#
# AddIcon* directives tell the server which icon to show for different
# files or filename extensions. These are only displayed for
# FancyIndexed directories.
#
AddIconByEncoding (CMP,/icons/compressed.gif) x-compress x-gzip

AddIconByType (TXT,/icons/text.gif) text/*
AddIconByType (IMG,/icons/image2.gif) image/*
AddIconByType (SND,/icons/sound2.gif) audio/*
AddIconByType (VID,/icons/movie.gif) video/*

AddIcon /icons/binary.gif .bin .exe
AddIcon /icons/binhex.gif .hqx
AddIcon /icons/tar.gif .tar
AddIcon /icons/world2.gif .wrl .wrl.gz .vrml .vrm .iv
AddIcon /icons/compressed.gif .Z .z .tgz .gz .zip
AddIcon /icons/a.gif .ps .ai .eps
```

```
AddIcon /icons/layout.gif .html .shtml .htm .pdf
AddIcon /icons/text.gif .txt
AddIcon /icons/c.gif .c
AddIcon /icons/p.gif .pl .py
AddIcon /icons/f.gif .for
AddIcon /icons/dvi.gif .dvi
AddIcon /icons/uuencoded.gif .uu
AddIcon /icons/script.gif .conf .sh .shar .csh .ksh .tcl
AddIcon /icons/tex.gif .tex
AddIcon /icons/bomb.gif core

AddIcon /icons/back.gif ..
AddIcon /icons/hand.right.gif README
AddIcon /icons/folder.gif ^^DIRECTORY^^
AddIcon /icons/blank.gif ^^BLANKICON^^

#
# DefaultIcon is which icon to show for files which do not have an icon
# explicitly set.
#

DefaultIcon /icons/unknown.gif

#
# AddDescription allows you to place a short description after a file in
# server-generated indexes. These are only displayed for FancyIndexed
# directories.
# Format: AddDescription "description" filename
#
#AddDescription "GZIP compressed document" .gz
#AddDescription "tar archive" .tar
#AddDescription "GZIP compressed tar archive" .tgz

#
# ReadmeName is the name of the README file the server will look for by
# default, and append to directory listings.
#
# HeaderName is the name of a file which should be prepended to
# directory indexes.

ReadmeName README.html
HeaderName HEADER.html

#
# IndexIgnore is a set of filenames which directory indexing should ignore
```

```
# and not include in the listing. Shell-style wildcarding is permitted.
#
IndexIgnore .??* *~ *# HEADER* README* RCS CVS *,v *,t

#
# AddEncoding allows you to have certain browsers (Mosaic/X 2.1+)
uncompress
# information on the fly. Note: Not all browsers support this.
# Despite the name similarity, the following Add* directives have nothing
# to do with the FancyIndexing customization directives above.
#
AddEncoding x-compress Z
AddEncoding x-gzip gz tgz

#
# DefaultLanguage and AddLanguage allows you to specify the language of
# a document. You can then use content negotiation to give a browser a
# file in a language the user can understand.
#
# Specify a default language. This means that all data
# going out without a specific language tag (see below) will
# be marked with this one. You probably do NOT want to set
# this unless you are sure it is correct for all cases.
#
# * It is generally better to not mark a page as
# * being a certain language than marking it with the wrong
# * language!
#
# DefaultLanguage nl
#
# Note 1: The suffix does not have to be the same as the language
# keyword --- those with documents in Polish (whose net-standard
# language code is pl) may wish to use "AddLanguage pl .po" to
# avoid the ambiguity with the common suffix for perl scripts.
#
# Note 2: The example entries below illustrate that in some cases
# the two character 'Language' abbreviation is not identical to
# the two character 'Country' code for its country,
# E.g. 'Danmark/dk' versus 'Danish/da'.
#
# Note 3: In the case of 'ltz' we violate the RFC by using a three char
# specifier. There is 'work in progress' to fix this and get
# the reference data for rfc1766 cleaned up.
#
# Danish (da) - Dutch (nl) - English (en) - Estonian (et)
```

```
# French (fr) - German (de) - Greek-Modern (el)
# Italian (it) - Norwegian (no) - Norwegian Nynorsk (nn) - Korean (kr)
# Portugese (pt) - Luxembourggeois* (ltz)
# Spanish (es) - Swedish (sv) - Catalan (ca) - Czech(cz)
# Polish (pl) - Brazilian Portuguese (pt-br) - Japanese (ja)
# Russian (ru) - Croatian (hr)
#
AddLanguage da .dk
AddLanguage nl .nl
AddLanguage en .en
AddLanguage et .et
AddLanguage fr .fr
AddLanguage de .de
AddLanguage he .he
AddLanguage el .el
AddLanguage it .it
AddLanguage ja .ja
AddLanguage pl .po
AddLanguage kr .kr
AddLanguage pt .pt
AddLanguage nn .nn
AddLanguage no .no
AddLanguage pt-br .pt-br
AddLanguage ltz .ltz
AddLanguage ca .ca
AddLanguage es .es
AddLanguage sv .se
AddLanguage cz .cz
AddLanguage ru .ru
AddLanguage tw .tw
AddLanguage zh-tw .tw
AddLanguage hr .hr

#
# LanguagePriority allows you to give precedence to some languages
# in case of a tie during content negotiation.
#
# Just list the languages in decreasing order of preference. We have
# more or less alphabetized them here. You probably want to change this.
#
LanguagePriority en da nl et fr de el it ja kr no pl pt pt-br ltz ca es sv tw

#
# ForceLanguagePriority allows you to serve a result page rather than
# MULTIPLE CHOICES (Prefer) [in case of a tie] or NOT ACCEPTABLE
```

```
(Fallback)
# [in case no accepted languages matched the available variants]
#
ForceLanguagePriority Prefer Fallback

#
# Specify a default charset for all pages sent out. This is
# always a good idea and opens the door for future internationalisation
# of your web site, should you ever want it. Specifying it as
# a default does little harm; as the standard dictates that a page
# is in iso-8859-1 (latin1) unless specified otherwise i.e. you
# are merely stating the obvious. There are also some security
# reasons in browsers, related to javascript and URL parsing
# which encourage you to always set a default char set.
#
AddDefaultCharset ISO-8859-1

#
# Commonly used filename extensions to character sets. You probably
# want to avoid clashes with the language extensions, unless you
# are good at carefully testing your setup after each change.
# See ftp://ftp.isi.edu/in-notes/iana/assignments/character-sets for
# the official list of charset names and their respective RFCs
#
AddCharset ISO-8859-1 .iso8859-1 .latin1
AddCharset ISO-8859-2 .iso8859-2 .latin2 .cen
AddCharset ISO-8859-3 .iso8859-3 .latin3
AddCharset ISO-8859-4 .iso8859-4 .latin4
AddCharset ISO-8859-5 .iso8859-5 .latin5 .cyr .iso-ru
AddCharset ISO-8859-6 .iso8859-6 .latin6 .arb
AddCharset ISO-8859-7 .iso8859-7 .latin7 .grk
AddCharset ISO-8859-8 .iso8859-8 .latin8 .heb
AddCharset ISO-8859-9 .iso8859-9 .latin9 .trk
AddCharset ISO-2022-JP .iso2022-jp .jis
AddCharset ISO-2022-KR .iso2022-kr .kis
AddCharset ISO-2022-CN .iso2022-cn .cis
AddCharset Big5 .Big5 .big5
# For russian, more than one charset is used (depends on client, mostly):
AddCharset WINDOWS-1251 .cp-1251 .win-1251
AddCharset CP866 .cp866
AddCharset KOI8-r .koi8-r .koi8-ru
AddCharset KOI8-ru .koi8-uk .ua
AddCharset ISO-10646-UCS-2 .ucs2
AddCharset ISO-10646-UCS-4 .ucs4
AddCharset UTF-8 .utf8
```

```
# The set below does not map to a specific (iso) standard
# but works on a fairly wide range of browsers. Note that
# capitalization actually matters (it should not, but it
# does for some browsers).
#
# See ftp://ftp.isi.edu/in-notes/iana/assignments/character-sets
# for a list of sorts. But browsers support few.
#
AddCharset GB2312      .gb2312 .gb
AddCharset utf-7      .utf7
AddCharset utf-8      .utf8
AddCharset big5       .big5 .b5
AddCharset EUC-TW     .euc-tw
AddCharset EUC-JP     .euc-jp
AddCharset EUC-KR     .euc-kr
AddCharset shift_jis  .sjis

#
# AddType allows you to add to or override the MIME configuration
# file mime.types for specific file types.
#
AddType application/x-tar .tgz

#
# AddHandler allows you to map certain file extensions to "handlers":
# actions unrelated to filetype. These can be either built into the server
# or added with the Action directive (see below)
#
# To use CGI scripts outside of ScriptAliased directories:
# (You will also need to add "ExecCGI" to the "Options" directive.)
#
#AddHandler cgi-script .cgi

#
# For files that include their own HTTP headers:
#
#AddHandler send-as-is asis

#
# For server-parsed imagemap files:
#
AddHandler imap-file map
```

```
#
# For type maps (negotiated resources):
# (This is enabled by default to allow the Apache "It Worked" page
# to be distributed in multiple languages.)
#
AddHandler type-map var

# Filters allow you to process content before it is sent to the client.
#
# To parse .shtml files for server-side includes (SSI):
# (You will also need to add "Includes" to the "Options" directive.)
#
AddOutputFilter INCLUDES .shtml

#
# Action lets you define media types that will execute a script whenever
# a matching file is called. This eliminates the need for repeated URL
# pathnames for oft-used CGI file processors.
# Format: Action media/type /cgi-script/location
# Format: Action handler-name /cgi-script/location
#

#
# Customizable error responses come in three flavors:
# 1) plain text 2) local redirects 3) external redirects
#
# Some examples:
#ErrorDocument 500 "The server made a boo boo."
#ErrorDocument 404 /missing.html
#ErrorDocument 404 "/cgi-bin/missing_handler.pl"
#ErrorDocument 402 http://www.example.com/subscription_info.html
#

Alias /error/ "/var/www/error/"
<IfModule mod_negotiation.c>
<IfModule mod_include.c>
    <Directory "/var/www/error/">
        AllowOverride None
        Options IncludesNoExec
        AddOutputFilter Includes html
        AddHandler type-map var
        Order allow,deny
        Allow from all
        LanguagePriority en es de fr
```

```
ForceLanguagePriority Prefer Fallback
</Directory>

ErrorDocument 400 /error/HTTP_BAD_REQUEST.html.var
ErrorDocument 401 /error/HTTP_UNAUTHORIZED.html.var
ErrorDocument 403 /error/HTTP_FORBIDDEN.html.var
ErrorDocument 404 /error/HTTP_NOT_FOUND.html.var
ErrorDocument 405 /error/HTTP_METHOD_NOT_ALLOWED.html.var
ErrorDocument 408 /error/HTTP_REQUEST_TIME_OUT.html.var
ErrorDocument 410 /error/HTTP_GONE.html.var
ErrorDocument 411 /error/HTTP_LENGTH_REQUIRED.html.var
ErrorDocument 412 /error/HTTP_PRECONDITION_FAILED.html.var
ErrorDocument 413 /error/HTTP_REQUEST_ENTITY_TOO_LARGE.html.var
ErrorDocument 414 /error/HTTP_REQUEST_URI_TOO_LARGE.html.var
ErrorDocument 415 /error/HTTP_SERVICE_UNAVAILABLE.html.var
ErrorDocument 500 /error/HTTP_INTERNAL_SERVER_ERROR.html.var
ErrorDocument 501 /error/HTTP_NOT_IMPLEMENTED.html.var
ErrorDocument 502 /error/HTTP_BAD_GATEWAY.html.var
ErrorDocument 503 /error/HTTP_SERVICE_UNAVAILABLE.html.var
ErrorDocument 506 /error/HTTP_VARIANT_ALSO_VARIES.html.var
</IfModule>
</IfModule>

#
# The following directives modify normal HTTP response behavior to
# handle known problems with browser implementations.
#
BrowserMatch "Mozilla/2" nokeepalive
BrowserMatch "MSIE 4\.0b2;" nokeepalive downgrade-1.0 force-response-1.0
BrowserMatch "RealPlayer 4\.0" force-response-1.0
BrowserMatch "Java/1\.0" force-response-1.0
BrowserMatch "JDK/1\.0" force-response-1.0

#
# The following directive disables redirects on non-GET requests for
# a directory that does not include the trailing slash. This fixes a
# problem with Microsoft WebFolders which does not appropriately handle
# redirects for folders with DAV methods.
#
BrowserMatch "Microsoft Data Access Internet Publishing Provider" redirect-carefully
BrowserMatch "^WebDrive" redirect-carefully
```

```
#
# Allow server status reports, with the URL of http://servername/server-
status
# Change the ".your-domain.com" to match your domain to enable.
#
#<Location /server-status>
#   SetHandler server-status
#   Order deny,allow
#   Deny from all
#   Allow from .your-domain.com
#</Location>

#
# Allow remote server configuration reports, with the URL of
# http://servername/server-info (requires that mod_info.c be loaded).
# Change the ".your-domain.com" to match your domain to enable.
#
#<Location /server-info>
#   SetHandler server-info
#   Order deny,allow
#   Deny from all
#   Allow from .your-domain.com
#</Location>

#
# Proxy Server directives. Uncomment the following lines to
# enable the proxy server:
#
#<IfModule mod_proxy.c>
#ProxyRequests On
#
#<Proxy *>
#   Order deny,allow
#   Deny from all
#   Allow from .your-domain.com
#</Proxy>

#
# Enable/disable the handling of HTTP/1.1 "Via:" headers.
# ("Full" adds the server version; "Block" removes all outgoing Via: headers)
# Set to one of: Off | On | Full | Block
#
#ProxyVia On
```

```
#
# To enable the cache as well, edit and uncomment the following lines:
# (no cacheing without CacheRoot)
#
#CacheRoot "/etc/httpd/proxy"
#CacheSize 5
#CacheGcInterval 4
#CacheMaxExpire 24
#CacheLastModifiedFactor 0.1
#CacheDefaultExpire 1
#NoCache a-domain.com another-domain.edu joes.garage-sale.com

# </IfModule>
# End of proxy directives.

### Section 3: Virtual Hosts
#
# VirtualHost: If you want to maintain multiple domains/hostnames on your
# machine you can setup VirtualHost containers for them. Most
# configurations
# use only name-based virtual hosts so the server doesn't need to worry
# about
# IP addresses. This is indicated by the asterisks in the directives below.
#
# Please see the documentation at
# <URL:http://httpd.apache.org/docs-2.0/vhosts/>
# for further details before you try to setup virtual hosts.
#
# You may use the command line option '-S' to verify your virtual host
# configuration.

#
# Use name-based virtual hosting.
#NameVirtualHost 172.16.1.19

# Where do we put the lock and pif files?
#LockFile "/var/lock/httpd.lock"
#CoreDumpDirectory "/etc/httpd"
# Defaults for virtual hosts
# Logs
#
# Virtual hosts
#
# Virtual host Default Virtual Host
```



```
#<VirtualHost *>
#   ServerName _default_
#   ServerSignature email
#   DirectoryIndex index.php index.html index.htm index.shtml
#   <Directory "/var/www/html/">
#     AllowOverride none
#   </Directory>
#   ErrorDocument 400 "file:"
#   ErrorDocument 401 "file:"
#   ErrorDocument 403 "file:"
#   ErrorDocument 404 "file:"
#   ErrorDocument 405 "file:"
#   ErrorDocument 406 "file:"
#   ErrorDocument 407 "file:"
#   ErrorDocument 408 "file:"
#   ErrorDocument 409 "file:"
#   ErrorDocument 410 "file:"
#   ErrorDocument 411 "file:"
#   ErrorDocument 412 "file:"
#   ErrorDocument 413 "file:"
#   ErrorDocument 414 "file:"
#   ErrorDocument 415 "file:"
#   ErrorDocument 416 "file:"
#   ErrorDocument 417 "file:"
#   ErrorDocument 500 "file:"
#   ErrorDocument 501 "file:"
#   ErrorDocument 502 "file:"
#   ErrorDocument 503 "file:"
#   ErrorDocument 504 "file:"
#   ErrorDocument 505 "file:"

#   ServerSignature email
#   LogLevel debug
#   HostNameLookups off
#</VirtualHost>
#
# Virtual host Virtual Host 0
#<VirtualHost 172.16.1.19>
#   ServerName serviweb
#   ServerAlias liuKar
#   ServerSignature email
#   ErrorDocument 400 "file:"
#   ErrorDocument 401 "file:"
#   ErrorDocument 403 "file:"
#   ErrorDocument 404 "file:"
```



```
# ErrorDocument 405 "file:"
# ErrorDocument 406 "file:"
# ErrorDocument 407 "file:"
# ErrorDocument 408 "file:"
# ErrorDocument 409 "file:"
# ErrorDocument 410 "file:"
# ErrorDocument 411 "file:"
# ErrorDocument 412 "file:"
# ErrorDocument 413 "file:"
# ErrorDocument 414 "file:"
# ErrorDocument 415 "file:"
# ErrorDocument 416 "file:"
# ErrorDocument 417 "file:"
# ErrorDocument 500 "file:"
# ErrorDocument 501 "file:"
# ErrorDocument 502 "file:"
# ErrorDocument 503 "file:"
# ErrorDocument 504 "file:"
# ErrorDocument 505 "file:"

# ServerSignature email
# </VirtualHost>
#
# Each directory to which Apache has access can be configured with respect
# to which services and features are allowed and/or disabled in that
# directory (and its subdirectories).
#
# Note that from this point forward you must specifically allow
# particular features to be enabled - so if something's not working as
# you might expect, make sure that you have specifically enabled it
# below.
#
# <Directory "/">
#     Options FollowSymLinks

#     AllowOverride None

# </Directory>

# <Directory "/var/www/html">
#     Options Indexes Includes FollowSymLinks
#     AllowOverride None
#     Allow from from all
#     Order allow,deny
```



```
#</Directory>

#<Directory "/var/www/icons">
#     Options Indexes MultiViews

#     AllowOverride None
#     Allow from from all
#     Order allow,deny
#</Directory>

#<Directory "/var/www/cgi-bin">
#     Options ExecCGI
#     AllowOverride None
#     Allow from from all
#     Order allow,deny
#</Directory>
```

Una vez finalizada la configuración del archivo httpd.conf, podemos decir que nuestro sitio web tiene las características expuestas en la tabla 4.1.

**Tabla 4.1. Referencias específicas del servidor Web.**

<b>Descripción</b>	<b>Directiva</b>	<b>Utilizado por</b>
Núcleo implicado		Linux
Sistema Operativo		Red Hat 8.0
Software del Web		Apache 2.4.18-14
Web Obtenido		<a href="http://liukar.isi.unanleon.edu.ni">http://liukar.isi.unanleon.edu.ni</a>
Puerto utilizado		80
Interfaz de Red	Listen	172.16.4.244 (Interna) 192.107.104.133 (Externa)
Dirección del Administrador	ServerAdmin	rina@unanleon.edu.ni
Nombre de la máquina del Servidor	ServerName	isi.unanleon.edu.ni:80
Ubicación de los ficheros HTML	DocumentRoot	"/var/www/html"
Nombre del sub directorio para los html del usuario	UserDir	pub_html
Número de clientes conectados	MaxClients	150
Formato de página a recibir	DirectoryIndex	index.htm    index.html.var index.php    index.html



# configuración de servidor de correo

## V. INTRODUCCIÓN

Con el crecimiento poblacional, actualmente la información tiene mayor importancia en la vida personal y empresarial transformando así, el campo de los negocios. Si se cambian las comunicaciones en una empresa se está cambiando también, su organización. Siendo el **E-Mail** la base de cambio para muchas empresas tanto en las organizaciones humanas como en el factor que contribuye a su competitividad. No obstante cada vez se están haciendo más indispensable servidores de correo electrónico en las empresas, de esta forma cada una esta actualizada e identificada en el plano científico, por eso cada día recurren a lugares que brinden estos servicios, ya que es un servicio que cuesta menos en términos de tiempo y experiencia. El e-mail es un **medio nuevo** que pulveriza los condicionantes espaciotemporales, gracias a su asincronicidad, instantaneidad y ubicuidad de accesos permitiendo enviar, recibir y gestionar los mensajes ya sea de forma local o remotamente.

Como resultado de esta gran demanda decidimos aplicar la configuración de un servidor de correo, utilizando como sistema operativo a **Red Hat Linux 8.0** que ofrece varias alternativas para utilizar el correo electrónico, ya sea como usuario de un equipo de sobremesa o un administrador del sistema, y como agente de transferencia de mensaje (MTA) a **qmail** cuyo diseño es modular.

Como soporte de interfaz gráfica, utilizamos el paquete de correo web estándar Squirrelmail 1.5.0, ya que es el indicado para este tipo de servicios privados.



Generando automáticamente fechas y horas, todos los datos personales que el remitente haya incluido en su fichero de firma, distribuye copias, y realiza otras muchas funciones bajo el control del usuario.

Ante la amenaza del filtrado de correo no solicitado, es necesario tomar las medidas precisas, por eso decidimos recurrir al módulo "SpamAssassin 3.0.4" en su última versión, siendo un programa agradable lanzado libremente bajo la licencia 2.0 de apache diseñado para los servidores de correo de la ISP (Protocolo de Servicio de Internet), este programa permite identificar y rechazar inmediatamente el Spam entrante, ya que utiliza una gama diversa de pruebas para identificar el e-mail comercial.

En este trabajo se revisan el funcionamiento de este servicio, las herramientas que permiten gestionarlo y se describe la terminología y conceptos básicos utilizados en una configuración de un servidor de correo en linux con qmail.

## VI. MARCO TEÓRICO

El e-mail es la aplicación más utilizada en Internet y una de las más simples y útiles, un gran porcentaje de usuarios de Internet, tienen su primer contacto con la red a través del e-mail. Esta herramienta se ha convertido en parte fundamental para nuestro trabajo.

### 6.1 Correo electrónico.

**Correo Electrónico:** es un mensaje enviado electrónicamente a cualquier persona que disponga de una cuenta en Internet.

#### 6.1.1 Atributos más característicos del Correo Electrónico.

El correo electrónico lo prefieren ante cualquier otro medio de comunicación tradicional, debido a las propiedades que este brinda, entre las cuales podemos señalar:

##### a) Es un medio electrónico.

Su consecuencia es la práctica instantaneidad de comunicación entre emisor y receptor, incomparable con la lentitud física del correo postal.

##### b) Es un medio asíncrono.

La asincronicidad entre emisor y receptor es una de las características más eficaces del correo electrónico, si la comparamos con el teléfono, que exige la presencia física simultánea de ambos usuarios a uno y otro lado de la línea para que se establezca la comunicación.

##### c) Es un medio ubicuo.

Quizá sería mejor decir deslocalizado o virtual, porque puede accederse a un buzón electrónico tanto para emitir mensajes como para recibirlos (leerlos) desde cualquier parte del mundo donde se disponga de un punto de acceso a la red telefónica automática, de un módem, de un ordenador portátil y del software adecuado.

##### d) Es un medio digital.

Este atributo permite la universalidad de contenidos, y no sólo textos, como muchos creen, sino datos, gráficos, voz, programas, etc., y todo lo que en un futuro vaya deparando la evolución tecnológica.

### **e) Es un medio informático.**

Tal cualidad le provee de todas las funcionalidades de los ordenadores, le dota de lo que pudiéramos llamar una multifuncionalidad "ilimitada". Tanto el cuerpo del mensaje como las informaciones sobre el mensaje (direcciones, nombres, fechas, horas, firmas, destinatarios de originales, destinatarios de copias, listas,...) y el conjunto completo son estructuras de datos, susceptibles de tan amplia variedad de procesos como permita el software del sistema concreto de correo electrónico.

#### **6.1.2 Ventajas del Correo Electrónico.**

Al igual que los demás medios de información el correo electrónico posee ventajas que lo hacen singular del resto, entre las que podemos mencionar:

1. Rapidez.
2. Confiabilidad.
3. Reducción de costos, abarata considerablemente los gastos de teléfono, tanto en llamadas nacionales como internacionales.
4. Posibilidad de enviar información y que pueda ser modificada por el destinatario desde su propia computadora.
5. Flexibilidad de horario, tanto para el remitente como para el destinatario, sin importar el día y hora, el correo siempre estará ahí listo para ser leído.
6. Medio de comunicación interno y externo por el cual te puedes comunicar sin tener que moverte de tu oficina.
7. No solamente puedes enviar mensajes electrónicamente, sino que también puedes enviar documentos, fotografías, archivos de sonido y videos.

#### **6.1.3 Dirección del Correo Electrónico.**

**[maria@liukar.isi.unanleon.edu.ni](mailto:maria@liukar.isi.unanleon.edu.ni)**

Cada usuario de correo electrónico (e-mail) está identificado por una dirección única en Internet y que tiene las siguientes características:

La dirección de correo consta siempre de tres partes:



- a. La identificación del usuario (Maria): será el "username" que se le ha asignado en la máquina que se utiliza como servidor de correo, tiene la forma de una cuenta (un espacio en un servidor).
- b. La identificación del ordenador: liukar.isi.unanleon.edu.ni  
Identifica el dominio de Internet donde está localizado el servidor de correo, donde se alberga el buzón permanente y que está especificado en el URL (*Universal Resource Locator*) del sitio específico de INTERNET, y lo identifica unívocamente en el contexto de la red.  
  
Este puede ofrecernos información sobre diferentes aspectos del usuario, tales como su proveedor de conexión, zona geográfica o actividad principal del organismo al que pertenece.
- c. El carácter especial @, que se emplea para separar el identificador del usuario y del dominio. Este carácter nos sirve para identificar que la dirección es de correo electrónico.

#### 6.1.4 Composición de un Mensaje.

Todos los mensajes se pueden descomponer en dos partes:

##### a. Cabecera

En la cabecera es donde se incluye el nombre de la persona que envía el mensaje, el destinatario y el tema, y también se suelen incluir otros datos como la fecha y hora en que se escribió, y tras la cabecera se encuentra el texto del mensaje.

La cabecera queda formada por los campos:

**From:** (De), que indica el emisor del mensaje.

**To:** (Para), que señala el destinatario del mensaje y que puede ser a todo el mundo, a un usuario particular o mediante fórmulas como «todas las chicas» o «los aficionados al cine».

**Subject:** (Sobre), que pretende describir en una frase el contenido del mensaje. Tras estos datos, se escribe el texto principal.

##### b. Contenido.

El texto propiamente dicho.



### 6.1.5 Lectura del Correo Electrónico.

Un buzón es simplemente una serie de mensajes y se almacena habitualmente en un fichero del ordenador donde el agente de transporte ha determinado que debe depositar los mensajes para el destinatario de los mismos. Este destinatario deseará leerlo. Este ordenador necesita estar conectado permanentemente a Internet (al menos, es lo habitual), por lo que raramente será el ordenador personal del destinatario. Si el buzón está en una máquina UNIX, y el destinatario puede usar directamente esta máquina, tiene la opción de acceder directamente, vía UNIX a su buzón. Así lo hacen agentes de usuarios clásicos de UNIX como el elm o el pine. Lo habitual es que el destinatario tenga un acceso restringido al ordenador donde está depositado su buzón.

Para permitir el acceder a los buzones se puede recurrir a dos protocolos: POP3 e IMAP. Ambos protocolos requieren la identificación del usuario con un nombre de usuario y una contraseña, para proteger el buzón de miradas indiscretas.

Existen dos formas de gestionar nuestro correo electrónico: remota y local. Para trabajar con nuestro buzón tenemos dos posibilidades:

#### a. Correo Electrónico Remoto.

Nos conectamos con él (buzón), mediante algún programa de conexión remota (herramientas de terminal remota como telnet o emulación de terminal) y ejecutar el programa de gestión de correo residente en el servidor.

Es el apropiado para los usuarios que no tienen excesivo "respeto" hacia las máquinas multiusuario (los servidores de correo suelen estar configurados sobre plataformas UNIX, aunque, en la práctica, el contacto con el sistema operativo es muy leve) y que no disponen de un ordenador personal de uso exclusivo. Los principales ejemplos de programas para gestión de correo remoto son: mail, elm o Pine, todos ellos sobre plataforma UNIX.

Ventaja: tiene la privacidad de la información que envía o recibe.

Desventaja: tiene la menor facilidad de manejo.

#### b. Correo Electrónico Local.

Es utilizar un programa local (que corre en nuestro ordenador) de gestión de correo, que se conecta al servidor de correo para traer (con POP3 o IMAP) hasta nuestro ordenador los mensajes recibidos y enviar los salientes (con SMTP).

No hay que olvidar que seguimos dependiendo de un servidor de correo. Aunque de cara al usuario parezca que sólo se trabaja en un entorno formado únicamente por su equipo personal, realmente sigue haciendo falta otra máquina (servidor de correo) para que todo funcione correctamente. Es apropiado para usuarios que dispongan de un ordenador personal de uso exclusivo o que no sean excesivamente reservados con su correo. Los principales ejemplos de programas para gestión de correo local son: Eudora, Microsoft, Outlook, Netscape Messenger.

*Ventajas:*

- Tiene la facilidad de manejo.
- Tiene la facilidad para añadir ficheros en los mensajes. Nos evitamos el paso de transferir nosotros mismos el fichero al servidor de correo correspondiente.

*Desventaja:*

- La escasa protección de datos que ofrecen intrínsecamente los sistemas personales.

### **6.1.6 Agentes para el funcionamiento del servicio de E-Mail.**

Los sistemas de correo electrónico se componen de varias partes denominadas agentes y todos ellos desempeñan un papel específico en el proceso de transmitir y administrar mensajes. Cada agente se responsabiliza de una porción lógica del sistema. Aunque la mayoría de los usuarios sólo conocen el programa de correo electrónico como el sistema para enviar y recibir mensajes, cada uno de los agentes siguientes es también importante para garantizar que los mensajes lleguen al destino correcto. Actualmente existen cinco agentes:

<b>A.</b>	<b>MUA</b>	<i>Mail User Agent</i>	Agente de Usuario de Correo
<b>B.</b>	<b>MTA</b>	<i>Mail Transfer Agent</i>	Agente de Transferencia de Correo
<b>C.</b>	<b>MDA</b>	<i>Mail Delivery Agent</i>	Agente de Entrega de Correo
<b>D.</b>	<b>MSA</b>	<i>Mail Submission Agent</i>	Agente de Registro de Correo
<b>E.</b>	<b>MAA</b>	<i>Mail Access Agent</i>	Agente de Acceso al Correo

#### **A. MUA (Mail User Agent, Agente de Usuario de Correo)**

Un MUA (Agente de Usuario de Correo) es un programa que permite a un usuario, como mínimo, leer y escribir mensajes de correo electrónico.



Típicamente, esto se hace a través de una interfaz que puede ser gráfica (Ximian Evolution, Outlook, Webmail, etc) o en texto (Pine, Mutt, etc). A un MUA se le denomina a menudo **cliente de correo**.

Hay muchos programas MUA que ofrecen al usuario muchas más funciones, entre las que se incluyen la recuperación de mensajes mediante los protocolos POP e IMAP, la configuración de buzones de correo para almacenar los mensajes o ayuda para presentar los mensajes nuevos a un programa MTA que los enviará al destino final.

También debe tener funcionalidad MIME (*Multipurpose Internet Mail Extensions*, Extensiones de Correo de Internet Multipropósito).

## **B. MTA (Mail Transfer Agent, Agente de Transferencia de Correo)**

Un programa MTA (Agente de Transferencia de Correo) transfiere los mensajes de correo electrónico entre máquinas que usan el protocolo SMTP. Un mensaje puede pasar por varios MTA hasta llegar al destino final. La mayoría de los usuarios desconocen la existencia de estos agentes, incluso si cada mensaje se envía a través de un MTA como mínimo.

Los MTA escuchan en los puertos 25 y 587. Típicamente se contactan el uno al otro usando el puerto 25. Los agentes de registro (MSA) usan el puerto 587. A la transferencia de correo electrónico para un cliente se denomina reenvío (o envío).

## **C. MDA (Mail Delivery Agent, Agente de Entrega de Correo)**

Los agentes MTA (Agente de Transferencia de Correo) utilizan programas MDA (Agente de Entrega de Correo) para entregar el correo electrónico al buzón de un usuario concreto. En muchos casos, el agente MDA es realmente un LDA (*Local Delivery Agent*, Agente de Entrega Local), como bin/mail o Procmal.

Cualquier programa que gestione realmente un mensaje para entregarlo al punto donde lo leerá un agente MUA (Agente de Usuario de Correo) se puede considerar un agente MDA. Tenga en cuenta que los agentes MDA no transportan mensajes entre sistemas ni actúan como interfaz para el usuario final.

Muchos usuarios no utilizan directamente agentes MDA, porque sólo se necesitan agentes MTA y MUA para enviar y recibir correo. Sin embargo, algunos agentes MDA se pueden utilizar para ordenar los mensajes antes de que los lea el usuario, lo cual es de gran ayuda si recibe una gran cantidad de correo.



## D. MSA (*Mail Submission Agent*, Agente de Registro de Correo)

El MSA o Agente de Registro de Correo es un agente nuevo que divide la carga de trabajo del MTA en servicios con muchos usuarios y mejora el desempeño. La idea es que el agente de servicio se preocupe de las tareas relativas al direccionamiento, tomando cierta parte de la carga de trabajo del MTA primario. Éste simplemente puede confiar la validez de las direcciones cuando recibe un correo de agentes de registros conocidos. El MSA corrige direcciones, y arregla y reescribe encabezados. Procesa el correo de su propia cola y lo envía a un agente de transferencia local.

## E. MAA (*Mail Access Agent* Agente de Acceso al Correo)

El MAA es usado para recuperar del buzón de mensajes de un servidor de correo electrónico. Ejemplos de MAAs son el protocolo IMAP y POP.

### 6.1.7 Protocolos del Correo Electrónico.

El correo electrónico, al igual que otros servicios de red, utiliza diversos protocolos. Estos protocolos permiten que máquinas distintas, que se ejecutan a menudo en sistemas operativos diferentes y que tienen instalados programas de correo electrónico distintos, se comuniquen entre sí y transfieran los correos para que lleguen a los destinatarios adecuados.

Existen dos grupos de protocolos:

- Los que van a permitir a un usuario acceder a su buzón de mensajes en un servidor. Los dos protocolos más populares son:  
IMAP (Protocolo de Acceso a Mensajes de Internet)  
POP (Protocolo de Oficina de Correo).
- Los que van a permitir enviar mensajes a otros usuarios.  
Aquí tenemos el protocolo SMTP (*Simple Mail Transfer Protocol*, Protocolo Simple de Transferencia de Correo).

#### A. IMAP

**Internet Message Access Protocol.**

**Protocolo de Acceso a Mensajes de Internet.**

El protocolo IMAP es un método que utilizan las aplicaciones cliente de correo electrónico para tener acceso a los mensajes almacenados remotamente. Al utilizar el protocolo IMAP, normalmente denominado IMAP4 después de la versión del protocolo en cuestión, los mensajes de correo electrónico se conservan en el servidor de correo remoto, donde el usuario puede leerlos o



eliminarlos, además de cambiar el nombre o eliminar los buzones de correo para almacenar correo electrónico.

Además, el protocolo IMAP es totalmente compatible con importantes estándares de mensajes de Internet, como, MIME (*Multipurpose Internet Mail Extensions*, Extensiones de Correo de Internet Multipropósito), que permiten recibir ficheros adjuntos. Muchos clientes de correo electrónico que utilizan el protocolo IMAP también se pueden configurar para que se almacene temporalmente en caché una copia de los mensajes localmente, de modo que el usuario puede examinar los mensajes que ha leído anteriormente si no está conectado directamente al servidor IMAP.

El protocolo IMAP se puede configurar también actualmente para almacenar los mensajes localmente de modo que se puedan ver los mensajes mientras no se está conectado a la red.

## **Ventajas IMAP**

Algunas ventajas principales del protocolo IMAP:

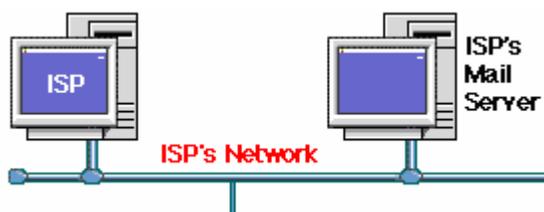
1. Puede manipular correos con distintos flags. Definibles por usuario.
2. Puede acceder y manipular múltiples buzones.
3. Puede almacenar correos tan bien como los recoge.
4. Permite actualizaciones concurrentes y acceso a buzones compartidos.
5. Diseñado para optimizar el acceso online, especialmente en accesos de baja velocidad.

## **B. POP**

### **Post Office Protocol**

### **Protocolo de Oficina de Correo**

Estos protocolos funcionan adecuadamente cuando los destinatarios están permanentemente conectados a INTERNET, unos años después de la publicación de los estándares se hizo más común la INTERNET para usuarios domésticos que desde sus casas se conectaban mediante un MODEM, esporádicamente a la INTERNET. Estos usuarios tienen un contrato con un ISP (*Internet Service Proveedor*) que está siempre conectado a la red y al llegar un mensaje de correo para un usuario de ese ISP, el mail-server del ISP debe guardar el mensaje hasta que el usuario se conecte y lo solicite a como se muestra en la figura 6.1.



**Figura 6.1. ISPs del mail Server.**

Este ambiente requirió la especificación de otro estándar para estos usuarios, de esta manera apareció en escena el protocolo de oficina postal POP.

El protocolo POP permite a los clientes de correo electrónico recuperar los mensajes de los servidores remotos y guardarlos en las máquinas locales. La mayoría de los clientes de correo que utilizan el protocolo POP se configuran automáticamente para eliminar el mensaje del servidor de correo después de transferirlo correctamente al sistema del cliente, aunque esto se puede cambiar.

El protocolo de oficina postal fue diseñado para trabajar conjuntamente con el protocolo TCP, inicialmente el proceso está escuchando el puerto 110, a la espera de una conexión, cuando esta se establece el servidor envía un saludo y luego comienza un diálogo en el que se intercambian comandos y respuestas, hasta que la conexión se libera.

### **Estados del POP3.**

Actualmente esta es la última versión (3) del protocolo POP. El POP3 va cambiando entre 3 distintos estados a lo largo de su vida, dependiendo de los resultados de algunos comandos especiales, los estados de POP3 son tres autorización, transacción y actualización los que detallaremos a continuación:

**Autorización:** en el que se entra cuando se establece la conexión TCP y sirve para que los usuarios se identifiquen ante el protocolo.

**Transacción:** cuando se hace una identificación positiva del usuario que quiere ingresar, aquí los mensajes pasan del servidor al cliente, una vez finalizado esto.

**Actualización:** donde elimina los mensajes que el usuario recibió, y así finaliza la conexión y se libera.

POP es un protocolo mucho más sencillo que IMAP, porque no se tienen que enviar tantos comandos entre el cliente y el servidor.

POP también es en cierta medida más conocido, aunque la mayoría de los clientes de correo electrónico pueden utilizar cualquiera de estos protocolos.



La mayoría de los usuarios que utilizan el protocolo POP sólo tienen un sistema y descargan los mensajes en sus máquinas para su almacenamiento. El protocolo POP también funciona adecuadamente si no se utiliza una conexión constante a Internet o a la red que contiene el servidor de correo.

Este protocolo permite una interfaz simple para la recepción de mensajes y se complementa perfectamente con el SMTP, en la forma en que este último se encarga del envío de correo y su tránsito por la INTERNET hasta el mail-server destino y el POP se encarga del transporte de los mensajes almacenados en el servidor a usuarios que esporádicamente se conectan a él. (no es necesario que los clientes estén conectados permanentemente en cambio los servidores si).

### C. SMTP vs. X.400

Un problema surgió cuando se intentaron conectar servidores de correo que utilizaban productos comerciales distintos, que aunque conceptualmente hacían lo mismo eran totalmente incompatibles. Este problema era debido a que hasta el momento no existía un **estándar** que reglamentara cómo debían implementar los productos este servicio. La necesidad de un estándar se hizo más patente cuando redes totalmente distintas comenzaron a conectarse mediante la INTERNET. Una compañía, posiblemente multinacional, que tuviera asiento en distintos países del mundo y quisiera intercambiar e-mail tenía que contratar un ISP (*INTERNET SERVICE PROVIDER*) y así tener acceso ilimitado a la INTERNET.

Como solución a este caos de variedades de mensajes de e-mail totalmente incompatible, surgieron dos soluciones, dos estándares: en 1982 bajo la forma de la RFC 821 y se denominó SMTP (*Simple Mail Transfer Protocol*), el Protocolo Simple de Transferencia de Correo, un par de años más tarde, llegó el estándar oficial de la CCITT para el manejo de mensajes en INTERNET y se llamó X.400 este estándar nunca llegó a imponerse en la INTERNET debido a su complejidad, lo poco flexible de las direcciones y a que llegó un poco tarde, el hecho es que el estándar de INTERNET para la transferencia de correo es el SMTP que se usa aún hoy ampliamente en toda la red.

### D. SMTP

**Simple Mail Transfer Protocol.**

**Protocolo Simple de Transferencia de Correo.**

El servidor SMTP de qmail no se ejecuta como un demonio de ejecución permanente. Un programa auxiliar, como por ejemplo inetd, xinetd, o



tcpserver se ejecuta como demonio. Cuando recibe una conexión TCP en el puerto 25, el puerto SMTP, ejecuta una copia de qmail-smtpd.

Conjunto de reglas que rigen el comportamiento de un servidor SMTP:

1. Acepta un mensaje entrante.
2. Comprueba las direcciones del mensaje.
3. Si son direcciones locales, almacena el mensaje para recuperarlo.
4. Si son direcciones remotas, envía el mensaje.
5. Si encuentra que el mensaje no se puede enviar (la cuenta ha excedido su cuota o el usuario ya no existe), devuelve un mensaje de error al remitente que explica el problema.

Mientras que los protocolos IMAP y POP permiten que un usuario reciba y lea el correo electrónico, el protocolo SMTP sirve para enviar correo electrónico.

Los mensajes salientes utilizan SMTP para pasar de la máquina del cliente al servidor, lugar desde el que se trasladan hasta el destino final. También dos servidores de correo que intentan transferir entre sí un mensaje utilizan SMTP para comunicarse, incluso si utilizan plataformas totalmente distintas.

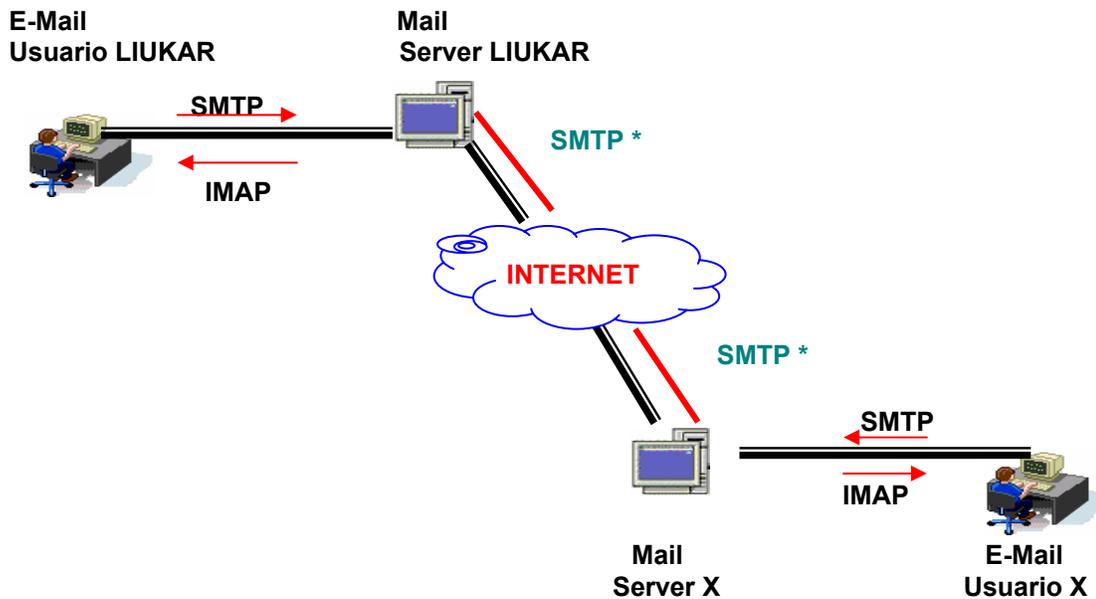
Al implementar el SMTP sobre los servicios del TCP se debe establecer una conexión entre un puerto X en el emisor y el puerto 25 del receptor. El protocolo ya tiene asignado este puerto para las conexiones en TCP. De esta manera el SMTP está escuchando el puerto 25 y cuando la conexión está establecida envía la respuesta 220.

SMTP usa el puerto 25 del servidor para comunicarse.

El protocolo SMTP también permite gestionar el reenvío de mensajes entre sistemas si el sistema receptor sabe el destino al que tiene que enviar el mensaje.

A diferencia de los protocolos IMAP y POP, el protocolo SMTP no requiere autenticación en su forma más básica. Esto ha provocado mucho correo basura o spam, ya que un usuario no local puede utilizar el sistema de otro para enviar o transmitir el correo a listas completas de destinatarios con los recursos y ancho de banda del sistema.

En la figura 6.2 se muestra como trabaja el protocolo IMAP en combinación con el SMTP, para la gestión del correo.



**Figura 6.2. SMTP trabajando con el IMAP.**

### 6.1.8 Re-transmisión (Relaying).

**RELAY:** permite la transferencia de correo hacia otros servidores.

Cuando llega un mensaje a un host se indica su camino inverso (cómo llegó hasta aquí) y su camino directo (cómo llegar a su destino). Si la transmisión tiene éxito el host receptor repetirá el procedimiento hasta llegar al host destino, el último del camino directo.

En caso de no tener éxito con la re-transmisión del mensaje, el host debe informar al emisor original del mensaje sobre las causas de la falla, y para eso utiliza la información contenida en el camino inverso, y construye un mensaje de **"mail inentregable"**.

El control de relay (reenvío) de correo es delegado en qmail a tcpserver por medio del fichero de configuración de reglas. En este fichero se especifican las IPs a las que se permite el reenvío de correos según la siguiente estructura:

```
127.0.0.1:allow,RELAYCLIENT=""  
192.168.2.:allow,RELAYCLIENT=""
```



Estas dos líneas permiten el reenvío de correo a la máquina 127.0.0.1 (la propia máquina) y a las máquinas de la red de clase C 192.168.2.1-255 (256 máquinas).

El fichero que contiene estas líneas suele llamarse `/etc/tcp.smtp` y para usarse debe obtenerse su versión `cdb` ejecutando:

```
tcprules /etc/tcp.smtp.cdb /etc/tcp.smtp.tmp < /etc/tcp.smtp
```

Tcprules es otra aplicación del paquete `ucspi-tcp`.

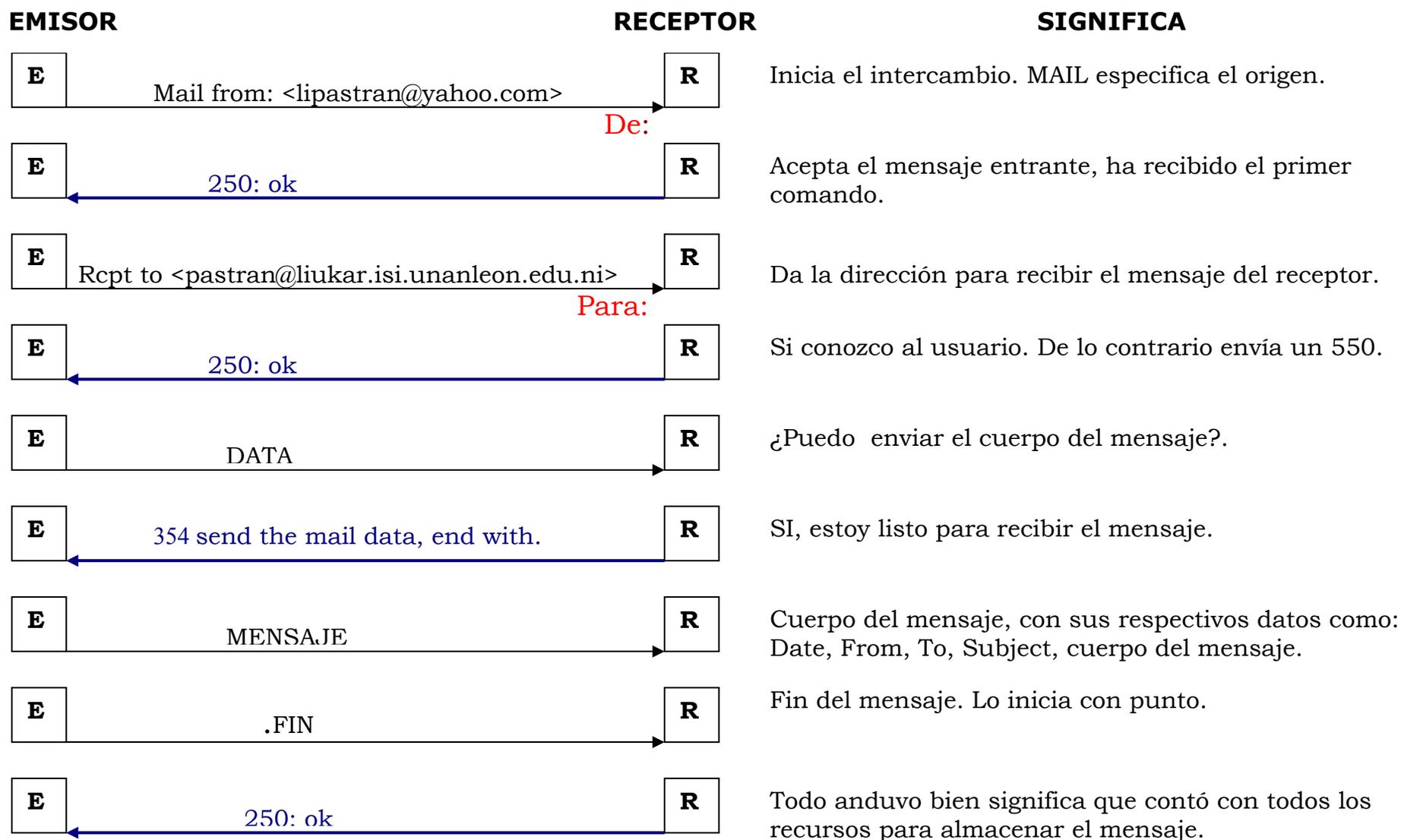
Cualquier máquina cuya IP no esté dentro de la configuración anteriormente mencionada no tendrá permitido el reenvío en nuestro servidor, obteniendo como resultado al intentar enviar correo, un mensaje de error en el servidor: `Relaying Denied`.

La mayor parte de los MTA de hoy día están configurados para desactivar completamente el relay, o para permitir que solo ciertos usuarios de confianza, utilicen el MTA como medio de reenvío. Hoy día, no se puede tener un relay abierto, la razón principal: el spam.

### **Ejemplo de Retransmisión.**

Este mensaje se envía con un emisor nulo (`MAIL FROM : <>`) para así evitar notificaciones, las razones para que un host no acepte un mensaje pueden ser varias y se desprenden de los comandos `RCPT` o `MAIL`.

En el envío y recepción de los correos remotos es necesario establecer la conexión entre los servidores involucrados, para que esto ocurra se necesita de la intervención del protocolo `SMTP`. En la figura 6.3 se da un claro ejemplo de este tipo de conexión.



**Figura 6.3. El procedimiento SMTP establece y libera una conexión satisfactoria.**



### 6.1.9 Como llega un mensaje desde el punto A al punto B.

Cuando un usuario de una máquina envía un mensaje a un usuario de otra máquina, pasan muchas cosas de las que quizás usted no era consciente. Supongamos que Alicia desea enviar un mensaje a Roberto.

**Alicia** [alicia@alfa.proveedor.com](mailto:alicia@alfa.proveedor.com)

**Roberto** [roberto@beta.proveedor.com](mailto:roberto@beta.proveedor.com)

1. Alicia redacta el mensaje con su Agente de Usuario de Correo, (MUA), por ejemplo mutt o pine. Especifica el destinatario en el campo To, el asunto del mensaje en el campo Subject, y luego escribe el texto del mensaje propiamente dicho. Se vería algo como lo siguiente:  
To: roberto@beta  
Subject: comida  
Te gustaría un pollo?
2. Cuando está contenta con el mensaje, le ordena al MUA que lo envíe.
3. El MTA puede añadir campos de cabecera adicionales como la fecha (Date) y el identificador de mensaje (Message-Id) y modificar los valores que Alicia ha introducido.

Ejemplo puede reemplazar

roberto@beta por "Roberto <roberto@beta.proveedor.com>"

Después, el MUA inyecta el mensaje en el sistema de correo. Hay dos maneras de hacerlo:

- a. Ejecutar un programa que proporcione el sistema de correo con la finalidad de inyectar mensajes.
- b. Abrir una conexión al protocolo simple de transferencia de correo (SMTP), bien en el sistema local o en el servidor de correo remoto. Para este ejemplo daremos por supuesto que el MUA utiliza un programa local de inyección para pasarle mensajes al agente de transporte de correo (MTA).

Los detalles de los procesos de inyección varían según el MTA, pero en los sistemas UNIX el estándar de hecho es el método del sendmail. Con este método, el agente de usuario de correo (MUA) puede poner el encabezado y el cuerpo en un fichero, separados por una línea en blanco y pasar el fichero al programa sendmail.



1. Si la inyección tiene éxito (porque el mensaje era sintácticamente correcto y se llamó correctamente a sendmail), el mensaje es ya responsabilidad del agente de transporte de correo (MTA). Los detalles varían en gran medida según el MTA, pero generalmente el MTA de la máquina alfa examina el encabezado para determinar a dónde enviar el mensaje; luego abre una conexión SMTP a la máquina beta, y luego le pasa el mensaje al MTA del sistema beta. El dialogo SMTP precisa que los mensajes se envíen en dos partes: el "sobre" o "envoltorio" (del inglés envelope), que especifica la dirección del destinatario (Roberto@beta.proveedor.com) y la dirección de retorno (alicia@alpha.proveedor.com), y el mensaje propiamente dicho, que está compuesto por la cabecera y el cuerpo.
2. Si el MTA de la máquina beta rechaza el mensaje, por ejemplo por no existir un usuario roberto en el sistema, el agente de transporte de correo (MTA) de la máquina alfa envía un mensaje de devolución (del inglés bounce) a la dirección de retorno, alicia@alfa, para notificarle el problema.
3. Si el agente de transporte de correo de beta acepta el mensaje, entonces examina la dirección del destinatario, y determina si es una dirección local de beta o está en un sistema remoto. En este caso es local, de manera que el MTA entrega el mensaje él mismo o lo pasa a un agente de entrega de correo (*en inglés Mail Delivery Agent, MDA*) como /bin/mail o procmail.
4. Si falla la entrega, quizás porque Roberto se ha excedido en su cuota de correo, el MTA de la máquina beta envía un mensaje de devolución a la dirección de retorno que figura en el "sobre", alicia@alfa.
5. Si la entrega tiene éxito, el mensaje aguardaría en el buzón de Roberto hasta que su agente de usuario de correo (MUA) lo lea y lo muestre.

## 6.2 qmail

Su nombre es "qmail", no "Qmail".

qmail creado por Dan J. Bernstein(DJB), un profesor de matemáticas.

qmail es un Agente de Transporte de Correo (MTA, *Mail Transport Agent en inglés*) que proporciona el servicio de correo electrónico para sistemas operativos Linux (UNIX). Se trata de un sustituto completo para el sistema popular sendmail que se suministra con los sistemas operativos UNIX. qmail utiliza el Simple Mail Transfer Protocol (SMTP, Protocolo Simple de Transferencia de Correo) para intercambiar mensajes con los MTA (Agentes de Transporte de Correo) de otros sistemas.



qmail está formado por un conjunto de programas que se integran en un paquete seguro y anima el uso de varias innovaciones en el correo (algo originado por Daniel J. Bernstein, otros no), incluyendo las cajas del formato del maildir para almacenar mensajes (los archivos del mbox también se soportan, y el estímulo a emigrar se da junto con una herramienta a las cajas del mbox convertido a las cajas del maildir) y los protocolos de QMTP (protocolo rápido de transferencia de correo).

El código fuente de qmail esta liberado y cualquier persona del mundo puede modificarlo para añadir funcionalidad al mismo, pero éstas deben ser aprobadas por el autor para ser permitidas.

Los competidores importantes de qmail son Exim y Postfix.

### **Quienes Usan qmail.**

Muchos de los servidores a los que nosotros accedemos con frecuencia están utilizando a qmail, estos son algunos de ellos:

- ❖ Hotmail (correo saliente).
- ❖ Yahoo!
- ❖ USA.net (correo saliente).
- ❖ Network Solutions.
- ❖ onelist.com (que actualmente está negociando la compra de egroups, otro servidor gratuito de listas de correo) yahoo en inglés.
- ❖ Algonet.se (proveedor sueco).
- ❖ gmx.de (proveedor alemán).
- ❖ Teleport (el mayor proveedor de Oregón).
- ❖ NetZero (proveedor gratuito).
- ❖ Red Hat (listas de correo).

#### **6.2.1 Razones para utilizar qmail.**

Estas son las razones suficientes que brinda qmail y que lo hacen ser seleccionado entre los demás MTA.

##### **a) Seguridad.**

qmail se diseño pensando en una seguridad alta. *Sendmail* arrastra una larga historia plagada de serios problemas de seguridad. Cuando se escribió *Sendmail*, la Red era un lugar mucho más amigable. Todo el mundo conocía a todo el mundo, y apenas había necesidad de diseñar y programar pensando en alta seguridad. Hoy en día Internet es un entorno mucho más hostil para los servidores de red. El autor de *Sendmail*, Eric Allman, ha hecho



un gran trabajo al ensamblar el programa, pero nada que se aleje de una redefinición del diseño podrá conseguir seguridad *real*.

### **b) Confiabilidad.**

Una vez que *qmail* ha aceptado un mensaje, garantiza que no se perderá. *qmail* soporta también un nuevo formato de bandeja de correo que funciona con seguridad *incluso en NFS* sin recurrir al bloqueo de ficheros.

El *qmail* también opcionalmente apoya *maildir*, un formato de caja de usuario nuevo, superconfiable. El *Maildirs*, a diferencia de archivos *mbox* y carpetas *mh*, no será corrompido si el sistema se estrella durante la entrega. Incluso mejor, no sólo un usuario puede leer seguramente su correo sobre el sistema de fichas de red, si no cualquier número de clientes de sistema de fichas de red puede entregarle el correo al mismo tiempo.

### **c) Eficiencia.**

Aunque las entregas remotas sean intrínsecamente limitadas por la lentitud de DNS y SMTP, *qmail* paraleliza el envío de correo, llevando a cabo de forma predeterminada hasta 20 entregas simultáneas de correo.

Sobre un Pentium bajo BSD/OS, *qmail* puede sostener fácilmente 200000 mensajes locales por día, esto es, mensajes separados inyectados y entregados a cajas en una verdadera prueba.

### **d) Simplicidad.**

*qmail* es más compacto y pequeño que cualquier otro MTA de características equivalentes.

## **6.2.2 Características de *qmail*.**

Algunas características principales de *qmail*:

1. Configuración.
2. Seguridad.
3. Construcción de mensajes.
4. Servicio SMTP.
5. Gestión de colas.
6. Devoluciones (*bounces*).
7. Rutados por dominio.
8. Entrega SMTP.
9. Reenvío de mensajes y listas de correo.
10. Entrega local de correo.
11. Servicio POP3.



### 6.2.3 Cómo trabaja qmail.

qmail trabaja bajo el siguiente esquema:

- a. Cualquiera puede ejecutarlo para cualquier propósito.
- b. Cualquiera puede estudiar el funcionamiento del programa, y mejorarlo según sus necesidades.
- c. Cualquiera puede distribuir copias del programa, siguiendo las siguientes normas: se puede distribuir el código original acompañado con parches, o los binarios en un .tar.gz con un MD5 idéntico al que él tiene en su web (o sea, sin modificaciones). También se pueden distribuir qmail modificados, siempre y cuándo él haya dado su aprobación, lo que quiere decir que él considera que el programa modificado "se comporta bien".
- d. Cualquiera puede mejorar el programa y mostrar sus mejoras al público.

Aún cuando muchas personas conocen este esquema, llegan a criticar este MTA haciendo comentarios de que no es GPL y no cumple con algunos estándares de "OpenSource".

Por otra parte cabe mencionar que qmail hablando técnicamente es uno de los MTA más seguros que existen, en cuanto a estabilidad y seguridad no hay quien le gane, esa es la razón por lo que muchos ISP incluyendo compañías grandes como yahoo, hotmail, por mencionar algunas, usan este "Free Software".

### 6.2.4 Comparación de qmail con otros MTA (Agentes de Transporte de Correo).

La mejor manera de saber que se esta usando algo bueno es compararlo con otros y esto lo vemos claramente en la tabla 6.1.

**Tabla 6.1. qmail con otros MTA.**

MTA	Desarrollo	Seguridad	Características	Rendimiento	Comp. Sendmail	Modular
qmail	normal	alta	altas	alto	Complementos	si
sendmail	alto	baja	altas	bajo	x	no
postfix	bajo	alta	normales	alto	si	si
exim	normal	Baja	altas	normal	si	no

## 6.2.5 Arquitectura de qmail.

qmail está compuesto de diversos subprogramas que realizan tareas específicas y que en conjunto constituyen el sistema de correo electrónico. La figura 6.4 proporciona una visión esquemática de estos componentes.

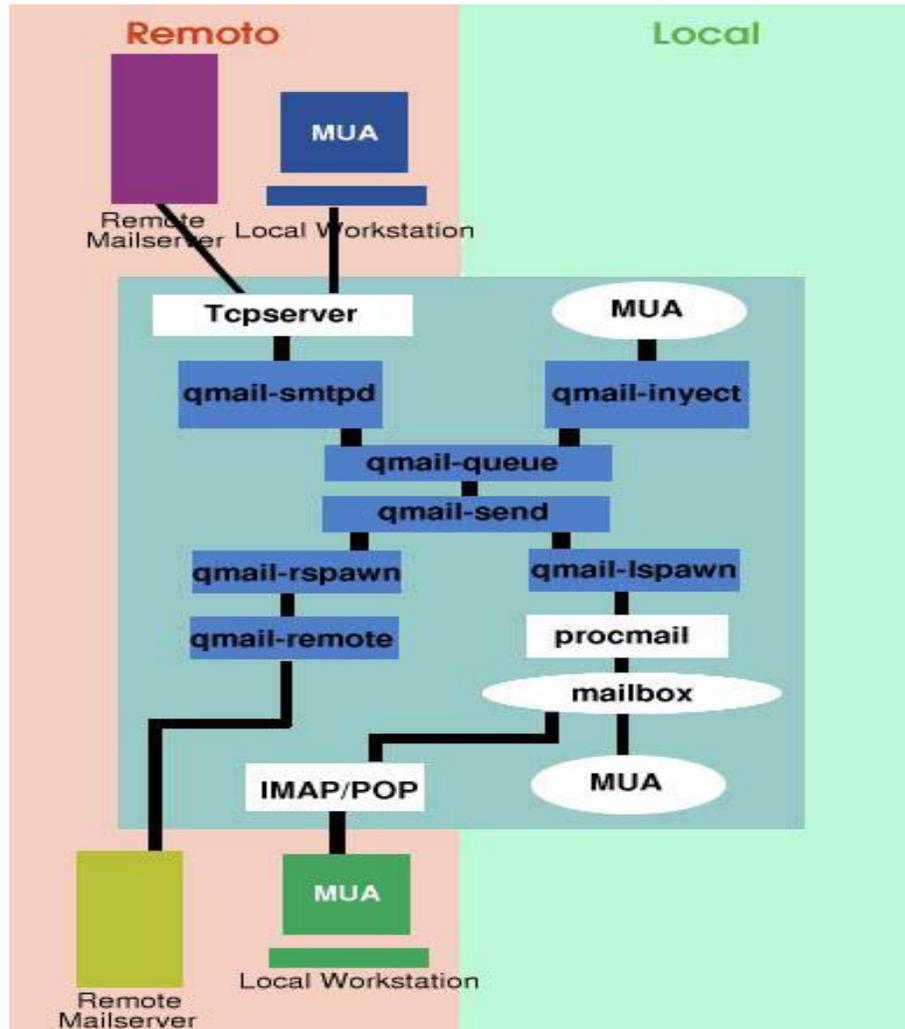


Figura 6.4 Arquitectura de los archivos de qmail.

### Descripción de los componentes utilizados en el esquema.

Para comprender un poco la arquitectura de qmail, detallamos un poco como están relacionados cada uno de sus componentes.

#### a) Origen local:

Un usuario Linux operando en el sistema (por ejemplo, con un terminal) remite un mensaje usando un MUA como **mail**. Esto normalmente originará una llamada a **qmail-inject** el cual llevará el mensaje al programa de



encolamiento **qmail-queue**. Acto seguido, **qmail-send** intentará remitirlo a su destino (si se puede) mediante los programas **qmail-lspawn** o **qmail-rspawn**.

**b) Origen remoto exterior:**

Un usuario de Internet ha enviado un mensaje hacia nuestro, este mensaje debe provenir de otro servidor de email mediante SMTP. **qmail-smtpd** descubre que el destinatario es de los nuestros, y acepta el mensaje, el cual pasa a la fase de encolamiento.

**c) Origen remoto de la LAN:**

Un cliente de nuestra red desea enviar un mensaje. Para esto ha configurado su MUA a fin de remitir hacia el servidor **qmail**. Esta remisión normalmente la efectúa usando SMTP, y por tanto nuevamente **qmail-smtpd** es el encargado de la recepción.

**d) Destino local:**

Los mensajes con destino local son guardados en el mailbox por **procmail** (o **qmail-local**) para ser recogidos por los MUAs de los usuarios que trabajan en el servidor. En el caso de que el usuario no esté en el servidor, sino, en una estación de trabajo, entonces su MUA deberá conectarse a un servidor IMAP o POP para obtener los mensajes del mailbox. En cualquier caso, **qmail-lspawn** es el encargado de controlar el agente de delivery local.

**e) Destino remoto:**

Los mensajes remotos se remiten con SMTP hacia otros MTA. **qmail-rspawn** es el encargado de los mensajes que se envían remotamente.

**NOTA:** El **tcpserver** es el encargado de despertar a **qmail-smtpd**.

### 6.2.6 Requisitos del Sistema para instalar qmail.

qmail se instala y ejecuta sobre la mayoría de sistemas UNIX y LINUX, aunque existen algunos requisitos:

- a) Aproximadamente 10 Mb de espacio libre en el área de compilación, y durante el proceso de compilación. Después de la compilación, puede liberar todo el espacio excepto 4 Mb, si elimina los ficheros objeto.



- b) Un sistema de desarrollo en C completo y en funcionamiento, con un compilador, los ficheros de cabecera del sistema, y las bibliotecas. Las instrucciones de compilación le mostrarán cómo saber si dispone de los componentes requeridos.
- c) Varios megabytes para los binarios, la documentación y los ficheros de configuración.
- d) Espacio en disco suficiente para la cola de correo. Los sistemas pequeños para un único usuario precisan solamente de un par de megabytes libres. Los servidores grandes pueden necesitar un par de gigas.
- e) Un sistema operativo compatible. La mayor parte de las variantes de UNIX son válidas.
- f) Se recomienda encarecidamente un servidor de nombres de dominio (DNS). Sin él, qmail sólo puede enviar a sistemas remotos que estén en su fichero de configuración smtpoutes.
- g) Conexiones a red adecuadas. qmail se diseñó para sistemas con buena conexión, así que es probable que no quiera usarlo para un servidor de listas de correo en una conexión telefónica de 28.8k. El paquete serialmail se diseñó para hacer que qmail fuese más compatible con sistemas de conexiones más parcas.

### 6.2.7 Requisitos para usar qmail.

Debemos tomar en cuenta ciertos requerimientos para poder usar qmail adecuadamente, estos pueden ser los siguientes:

- a. Descargar qmail, asegurándose que la descarga esta completa.
- b. Instalar qmail desde cero, o sea usando el archivo tar.gz.
- c. Disponer del compilador de lenguaje C. O usar los paquetes RPM (binarios). Pero este no es recomendable.
- d. Imprescindible instalar el programa tcpserver.
- e. Una conexión de red.
- f. Configurar el nameserver si se está instalando por primera vez un mailserver para las estaciones de un dominio.
- g. Tener un servidor web configurado adecuadamente.

### 6.2.8 Buzones usados por qmail.

qmail es un MTA que esta capacitado para trabajar con el formato de buzón que trae por defecto el mbox y también soporta el formato maildir.



### a. Formato de buzón Mbox.

El formato de buzón **mbox** es la forma de qmail para referirse al formato estándar de UNIX, mailbox, usado por muchos otros agentes de usuarios de correo (MUA), sin embargo no es fiable por definición, por que se almacenan múltiples mensajes en un único fichero, y puede caer en la corrupción de los mensajes en caso que el sistema de entrega falle.

Con mbox los mensajes van encabezados por una línea **From** y acaba por línea en blanco. Esta línea parece un campo de encabezado, pero no lo es: sólo es lo que el agente de entrega de correo añade para que los agentes de usuario de correo (programas clientes de correo) puedan averiguar donde comienza cada mensaje.

Solo un proceso puede abrir el archivo mbox en modo lectura/escritura. Durante la actualización del archivo mbox, todo el resto debe esperar. Por defecto el archivo es: /var/spool/mail/usuario

Por ejemplo:

```
./Mailbox
```

Esta línea hará que los mensajes se anexas a \$HOME/Mailbox.

### b. Formato de buzón Maildir.

El formato de buzón de **maildir** es creado por Dan Bernstein para obviar las limitaciones del formato mbox.

Cada mensaje en un buzón de tipo maildir se guarda en archivos separados dentro de uno de los subdirectorios, dependiendo de su estado. Un buzón maildir es un directorio que contiene tres subdirectorios:

**New** donde se almacena el correo sin leer.

**Cur** donde se almacenan los mensajes que ya se han leído.

**Tmp** donde se almacenan los mensajes que se están entregando.

### Ventajas de Maildir.

Algunas ventajas del formato maildir:

1. Cualquier número de máquinas podrán entregarle correo simultáneamente.
2. Es considerado a prueba de fallo para los mensajes entrantes.
3. Es rápido y fácil de usar por parte de los Agentes de Usuario de Correo (MUA).
4. Maildir funciona de maravilla a través de NFSs.



5. Una de las ventajas del formato maildir es que, aunque no utiliza el bloqueo para prevenir actualizaciones simultáneas desde diferentes agentes de entrega de correo, sin embargo es fiable. Esto quiere decir que los buzones de maildir pueden alojarse con seguridad en sistemas de ficheros montados mediante NFS (sistema de ficheros de red).

Por ejemplo:

./Maildir/.

## 6.3 Paquetes relacionados.

qmail al igual que otros MTA no trabaja por si solo, necesita una serie de paquetes que lo complementan, estos le proporcionan el soporte necesario para lograr ver su correcto funcionamiento y eficiencia.

### 6.3.1 UCSPI-TCP SERVER.

ucspi-tcp, es un acrónimo de UNIX *Client-Server Program Interface for TCP*. Interfaz de Programa de TCP Cliente-Servidor para UNÍX.

qmail necesita de un mecanismo que lance el demonio qmail-smtpd cada vez que llega un intento de conexión SMTP del exterior de un mailserver. Esto se puede hacer de diversas maneras.

El paquete ucspi-tcp consiste en un conjunto de herramientas que facilitan la creación de demonios tcp cliente-servidor. **tcpserver** es una alternativa más segura a **inetd**. Tiene integradas funcionalidades para control de acceso basado en reglas, y puede posponer conexiones de forma fácil cuando se alcanza la carga máxima configurada.

Proporciona además dos descriptores de ficheros, uno para leer y otro para escribir sobre el medio físico del que se trate. Es posible entonces asociarle un programa que leerá y escribirá sobre estos descriptores sin preocuparse del medio físico. Ello permite crear, a partir de un programa UNIX cualquiera, una aplicación cliente/servidor sobre el medio en cuestión. La ventaja ya que el programa encapsulado de esta manera deja de preocuparse por el medio que utiliza, que pasa a ser la tarea de la cápsula UCSPI.

Las utilidades UCSPI son de dos tipos: los clientes y los servidores.

- Un **cliente UCSPI** se conecta a un servidor en un puerto determinado y se comunica con el programa que se le especifique a través de los descriptores 6 y 7. El cliente no establece ninguna conexión suplementaria y muere con el programa.



- Un **servidor UCSPI** espera las conexiones de los clientes sobre un cierto puerto y comunica con el programa que se le especifique a través de los descriptores 0 y 1. Ello permite tener varias conexiones simultáneas.

tcpserver es preferible por las siguientes razones:

- tcpserver permite limitar el número de conexiones simultáneas a un servicio. Inetd tiene para la limitación del ritmo de conexiones un mecanismo que temporalmente desactiva servicios que están muy ocupados.
- tcpserver puede configurarse para denegar el acceso de determinadas máquinas o para reconocer las máquinas "locales" y para marcarlas como tales, de manera que qmail-smtpd pueda tratarlas de forma diferenciada.
- tcpserver es el único servidor que el autor de *qmail* soporta.

### 6.3.2 PHP4.

PHP "*Hypertext Preprocessor*" es un lenguaje de "código abierto" interpretado, de alto nivel, embebido en páginas HTML y ejecutado en el servidor. Con PHP nos referimos a un lenguaje de programación que está muy orientado al desarrollo de aplicaciones web. Cuando pedimos a nuestro servidor de web una página PHP, que no es más que un programa PHP que genera HTML, antes de enviar dicha página al cliente se la pasa al interprete de PHP. Este la interpreta y es el resultado de esta interpretación del programa PHP, contenido en la página PHP, lo que termina llegando al cliente.

Supongamos que el contenido de una página web que reside en el servidor, y cuyo nombre es "intro.php", tiene el siguiente contenido:

```
<? echo "<h1>Mensaje desde PHP</h1>";?>
```

Cuando un navegador le pida al servidor de web la página "intro.php", el servidor de web va a darse cuenta, por la extensión ".php", de que esta página ha de enviarse primero al interprete de PHP. Este recibe el contenido de la página y como resultado de esta ejecución (interpretación) genera una página HTML, que es la que envía al cliente a través de Apache. Como veremos en una página PHP se puede mezclar HTML y PHP, algo muy flexible pero que hay que manejar con cuidado ya que puede llevar a confusiones y, sobretodo, a que el equipo que diseñe las páginas web y el que programe la aplicación no puedan ser independientes.



Uno de los mayores quebraderos de cabeza de PHP en sus comienzos era que obligaba a tener Apache en código fuente y compilarlo. Por fin la instalación de PHP se puede hacer de forma sencilla y sin tener que modificar en casi nada a nuestro servidor de web. Tan sólo necesitamos tener una versión de Apache con soporte DSO para la carga dinámica de objetos.

Es necesario comprobar que el servidor web Apache no tenga un PHP sin soporte para PHP. La mejor forma de comprobar este dato es editar el fichero de configuración de Apache "httpd.conf" (normalmente en el directorio "/etc/httpd") y revisar que no se hagan referencias en él a PHP. Si las encontrara debería comentarlas y reiniciar el servidor web.

### 6.3.3 SQUIRRELMAIL.

En la figura 6.5 vemos la presentación del webmail de squirrelmail, claramente vemos que es una interfaz que solo nos permitirá introducir el nombre del usuario y su respectiva contraseña, y en la tabla 6.2 se brinda información a nivel general de las características de este webmail.

SquirrelMail  
webmail  
for  
nuts

SquirrelMail version 1.5.0  
By the SquirrelMail Development Team

**LIUKAR Login**

Name:

Password:

Login

**Figura 6.5. Interfaz del webmail de Squirrelmail.**

**Tabla 6.2. Características de Squirrelmail.**

Datos	Descripción
Version	1.5.0
Idioma	Inglés
Plataforma	Linux
Precio	GPL
Autor	The SquirrelMail Project Team
Nivel	Medio
Categoría Linux	E-Mail: Cliente
Arquitectura	i386
Distribución	Woody
web	<a href="http://www.squirrelmail.org/">http://www.squirrelmail.org/</a>
Sección	Correo
Novedad	Integración del plugin de DMN, algunas mejoras en la interfaz y varios errores corregidos.

**Webmail:** Servicio de administración del correo electrónico a través de una página web.

El squirrelmail es un programa webmail estándar escrito en PHP4. Incluye soporte para los protocolos IMAP y SMTP en PHP puro, todas las pantallas pueden generarse en HTML 4.0 puro, aunque es posible activar algunos módulos en JavaScript para mejorar la interfaz.

Squirrelmail permite revisar el correo que esta en un servidor con **imap** mediante cualquier navegador. Esto está muy bien para no tener que añadir más reglas a un firewall, para implantarlo en una oficina o simplemente porque no te gusta ni el mail, mutt, etc. para el correo y no quieres instalar ningún programa de gestión de correo.

Squirrelmail ofrece todas las posibilidades que se le pueden pedir a un cliente de correo electrónico, incluyendo MIME, agendas y manipulación de carpetas. Tiene muy pocos requisitos, y es muy fácil de configurar e instalar.

### **Ventajas de Squirrelmail.**

Squirrelmail es un webmail con muchas ventajas, como las que detallamos a continuación:

- a. Está escrito en PHP4 y es GPL, por lo que se puede ampliar, modificar fácilmente y es totalmente gratis.



- b. Tiene una arquitectura de plug-ins, con el que se pueden añadir otros plug-ins para incluir nuevas funciones. Es decir no hay que compilar nada.
- c. Es muchísimo más estable que bastantes clientes de correo (sobre todo si se compara con el Mozilla).
- d. Acceso a nuestro correo desde cualquier sitio, basta cualquier ordenador con conexión a Internet y un navegador.
- e. Acceso al correo de forma segura (a través de SSL).

### Requerimientos Básicos para instalar Squirrelmail.

Se deben considerar algunos detalles importantes antes de instalar squirrelmail, como los siguientes:

- PHP4. No hace falta tener instalada la librería IMAP de PHP.
- Servidor Web.
- Servidor IMAP.

### 6.3.4 DAEMONTOOLS.

El paquete daemontools es un sustituto para inetd o xinetd. La razón principal para utilizarlo aquí, es porque se recomienda usarlo con qmail y djbdns. daemontools es una colección de utilidades para el manejo de servicios UNIX.

El paquete Daemontools contiene una serie de utilidades que controlan y monitorizan los servicios. No es obligatorio pero es altamente recomendable, en especial para sistemas cargados. El paquete incluye:

**supervise** supervisa un servicio. Este arranca el servicio y reinicia el servicio si este muere. Arrancar un nuevo servicio es fácil: Todo *supervise* necesita un directorio con un script *run* que inicie el servicio.

**multilog** guarda los mensajes de error en uno o más logs. Él opcionalmente añade en cada línea la fecha y hora, y para cada registro, incluye o excluye las líneas que emparejan patrones especificados. Rota automáticamente registros para limitar la cantidad de espacio de disco usada. Si el disco está lleno, se detiene brevemente e intenta otra vez, sin perder ningún dato.

**svc** que dialoga con supervise y permite la parada, la pausa o el reinicio de un servicio.



***cyslog*** que mantiene un registro de un servicio, rotándolo automáticamente para mantenerlo por debajo del tamaño al que se ha configurado.

***accustamp*** que pone firmas de tiempo exactas a las entradas de registro.

***usually y errorsto*** que monitorizan los registros a la búsqueda de mensajes raros.

***setuser*** que ejecuta programas para el superusuario con el identificador de usuario (UID) y el identificador de grupo (GID) de un usuario normal.

### 6.3.5 DOT-FORWARD.

Se pronuncia punto forward, ha sido escrito por Dan Bernstein, es un paquete pequeño, fácil de instalar y de configurar. El paquete dot-forward es un complemento que proporciona compatibilidad con el fichero .forward de Sendmail. Dando a qmail la capacidad de usar estos ficheros.

Permiten a los usuarios controlar la entrega de los mensajes que reciben; qmail utiliza un mecanismo semejante: los ficheros .qmail. El paquete dot-forward da a qmail la capacidad de usar ficheros .forward de sendmail.

Cualquier agente de transporte de correo(MTA), que usa fichero .forward, tendra que usar dot-forward para evitar tener que convertir los ficheros .forward existentes a sus equivalentes .qmail (o simplemente, para hacer la transición a qmail menos visible para sus usuarios).

### 6.3.6 QUOTAS EN EL SISTEMA.

El sistema de correo electrónico debe almacenar los mensajes que miles de usuarios tienen en sus buzones. Para evitar que los sistemas de almacenamiento de los servidores lleguen a una situación de ocupación total del espacio disponible, lo cual causaría un importante perjuicio a todos los usuarios, existe un sistema que permite limitar el espacio ocupado por cada usuario.

Estos límites son los que se denominan CUOTAS DE DISCO (disk quota en inglés), y son establecidos en el sistema de archivos ext2fs, donde se le asigna a cada usuario un tamaño suficientemente razonable como para permitir un funcionamiento normal, limitándose así, la cantidad de espacio de almacenamiento en disco.

Las cuotas se aplican al tamaño total de todos los mensajes de un usuario, sin tener en cuenta cuantas carpetas de correo tenga el usuario. Las cuotas



de disco se implementan para asegurar la fiabilidad del servicio, ya que el espacio de disco disponible para el servicio de correo es limitado.

Todos los usuarios tienen impuesto un límite en la capacidad de disco duro que pueden ocupar en su directorio inicial (HOME).

La cuota le permite especificar los límites en dos aspectos de almacenamiento de discos:

1. El número de inodes que un usuario o un grupo de usuarios pueden poseer.
2. El número de los bloques de disco que pueden ser asignados a un usuario o un grupo de usuarios.

La idea detrás de la cuota es que los usuarios sean forzados a permanecer bajo su límite de consumo de disco, llevando su habilidad de consumir al espacio ilimitado de disco en un sistema. La cuota se maneja por un usuario, por la base de sistema de archivo. Si hay más de un sistema de archivo, entonces la cuota se debe poner para cada sistema de archivo separadamente. Varios instrumentos están disponibles para que usted pueda administrar y automatizar las políticas de la cuota en su sistema.

### **Aspectos a tener en cuenta respecto al comportamiento de las cuotas:**

Al trabajar con cuotas, antes debemos conocer como se comportaran las mismas en el sistema una vez activadas, para así saber que están funcionando.

1. Cuando la cuota está próxima a alcanzarse (normalmente a un 80%) recibirá un aviso de información. Esta situación es conocida también por su término en inglés: *overquota*.
2. Cuando reciba mensajes que sobrepasen la cuota, estos quedarán pendientes de envío al buzón del usuario mientras no exista espacio suficiente.

### **Recomendaciones para evitar sobrepasar la cuota:**

Para que cada usuario no rebase la cuota asignada y así no tenga problemas con la recepción de sus mensajes, es necesario considerar lo siguiente:

1. Vaciar periódicamente mensajes temporales, papelera y carpetas de mensajes enviados.
2. Vigilar el espacio disponible con cierta periodicidad.
3. Evitar enviar mensajes de gran tamaño a otros usuarios.



### 6.3.7 ANTISPAM.

Los antispam son muy útiles en los servidores de correo, ya que evitan que muchas personas inescrupulosas invadan nuestros buzones con correos basuras, los cuales están llenos de propagandas sin ningún fin personal, para ellos existen programas competentes para que nos auxilien en este problema, uno de ellos es el siguiente:

#### a. SPAMASSASSIN.

SpamaAssassin es un programa agradable lanzado libremente bajo la licencia 2.0 de Apache, diseñado para los servidores del correo de la ISP (Protocolo de Servicio de Internet) para identificar y rechazar inmediatamente el spam entrante antes de que consiga acercarse a su inbox (buzón). En la tabla 6.3 se puede ver algunos rasgos de este antispam.

**Tabla 6.3. Características generales de SpamAssassin.**

Descripción	Características
La versión 3.0.3 fue lanzada en	2005-04-28.
Licencia bajo	Perl.
Esto no es un paquete	GNU.
Interfaces de líneas de comando	No interactivo (Demonio).
Lenguaje de programación (origen)	Perl.
Asuntos	Filtros.
Sistemas Operativos	Todo el POSIX (Linux/BSD/UNIX-like OSes), Independiente del OS (escrita en una lengua interpretada).
Estado de Desarrollo	Production/Stable.
Es un módulo	Mail::SpamAssassin.

SpamAssassin es un filtro inteligente del e-mail que utiliza una gama diversa de pruebas para identificar el email comercial no solicitado (spam) realizándolo mediante el análisis del texto y el uso en tiempo real de algunas listas negras a través de Internet. A partir de su base de datos de reglas, recurre a una amplia evaluación heurística, la que se aplica a las cabeceras y al contenido del correo para clasificar el email.

Asignando ciertos puntos a frases de spam típicas tales como "strong buy" (fuerte compra), "you receive this mail because" (recibes este mensaje porque), "Viagra", "limited time offer" (oferta de tiempo limitado)... Si los puntos superan un determinado nivel el correo es declarado spam.



El Spamassassin utiliza la puerta 783 para scanear y para verificar email. SpamAssassin se escribió para entornos basados en UNIX que incluyan Perl, de preferencia 5.6.1 o recientes. Se requieren algunos módulos en el momento de su instalación. Se diseñó pensando en su flexibilidad e integración con varios Agentes de Transferencia de Correo MTA (Sendmail, Qmail, Exim, Postfix, Microsoft Exchange).

### **Ventajas SpamAssassin.**

El spamassassin tiene ciertas ventajas sobre otros antispam, como pueden ser:

- a. Arquitectura modular, tiene una que permite que otras tecnologías sean manejadas rápidamente contra el Spam y es diseñada para la integración fácil en cualquier sistema de correo electrónico.
- b. Acercamiento práctico de la multi-técnica de SpamAssassin.

La extensibilidad continúa dándole una ventaja sobre otros sistemas contra-Spam.

### **Características de SpamAssassin.**

Algunas características de SpamAssassin:

#### ***Ancho-espectro:***

SpamAssassin utiliza una variedad amplia de pruebas local y de la red para identificar firmas del Spam. Esto lo hace más duro para que los spammers identifiquen un aspecto sobre el cual puedan hacer sus mensajes a mano para trabajar.

#### ***Software libre:***

Se distribuye bajo los mismos términos y condiciones que otros paquetes de software populares de fuente abierta tales como el web server de Apache.

#### ***Fácil de extender:***

Las pruebas Contra-Spam y la configuración se almacenan en texto llano, haciéndolo fácil de configurar y agregar nuevas reglas.

#### ***Flexible:***

SpamAssassin encapsula su lógica en un API bien diseñado, abstracto, así que puede ser integrado dondequiera en la cabecera del email. Las clases de Mail::SpamAssassin se pueden utilizar en una variedad amplia de sistemas del email incluyendo procmail, sendmail, postfix, qmail, y mucho otros.

**Configuración Fácil:**

SpamAssassin requiere una configuración muy pequeña; usted no necesita ponerlo al día continuamente con los detalles de sus cuentas de correo, calidad de miembros de la lista que envía, etc. Una vez que esté clasificado, el sitio y las políticas de usuario específicas se pueden entonces aplicar contra el Spam. Las políticas se pueden aplicar en ambos servidores de correo y más adelante usar sus propias listas de usuario de correo.

En la instalación de SpamAssassin se necesita de un programa especial y de algunos módulos para su correcto funcionamiento. Los cuales son:

- b. Perl
- c. Digest-SHA1
- d. HTML-Parser
- e. ExtUtils-MakeMaer
- f. HTML::Tagset

**b. PERL.**

Practical Extraction and Report Languaje.  
Lenguaje práctico para informes y extracción.

Perl es software libre y fue creado por Larry Wall para simplificar tareas de administración de un sistema Unix, actualmente se ha convertido en una de las mejores herramientas para creación de scripts o de construcción de sitios Web. Perl fue el principal lenguaje de webs dinámicos antes de la aparición de lenguajes como PHP.

Perl es un lenguaje pensado para la manipulación de cadenas de caracteres, archivos y procesos. Esta manipulación se ve simplificada por el importante número de operadores a disposición del usuario. El lenguaje Perl se percibe habitualmente como un lenguaje intermedio entre los shell scripts y la programación en C.

Perl es un lenguaje de script (o de guiones), lo que quiere decir que no hace falta un fichero binario para poder ejecutar las instrucciones que hemos codificado usando este lenguaje, es decir, es interpretado, aunque el intérprete de Perl "compila" los programas antes de ejecutarlos. Perl tiene características de muchos lenguajes de programación (buscando lo mejor de cada uno), pero al que más se parece es a C.

Con Perl podemos programar casi de todo (hay librerías/módulos para casi cualquier cosa que queramos) para el desarrollo de casi cualquier tipo de aplicación, pero hay aplicaciones que requieren mucha rapidez en las cuales



es mejor utilizar otros lenguajes que no sean interpretados (la ejecución a través del intérprete introduce retardos). Perl es un lenguaje que permite programar de forma muy rápida y es excelente para el tratamiento de cadenas.

En la actualidad todas las distribuciones incluyen casi de serie un intérprete de Perl, es decir, el intérprete de Perl se instala casi por defecto (muy mínima debería ser la instalación para que no se incluyera o quitarlo a propósito).

Perl corre INCLUSO en muchos sistemas operativos y plataformas hardware poco populares o generalmente desconocidos.

Perl permite utilizar módulos para añadir mejoras o funcionalidades de otros autores, de forma que podemos reutilizar la mayor parte del código que otras personas han desarrollado de acuerdo a unos criterios comunes.

### **c. Digest-SHA1.**

Interfaz del Perl para el algoritmo SHA-1.

El módulo `Digest::SHA1` permite que usted utilice el algoritmo de resumen del mensaje del NIST SHA-1 dentro del programa de Perl. El algoritmo toma como entrada un mensaje de la longitud arbitraria y produce como salida "una huella digital" de 160-bit o "resumen del mensaje" de la entrada.

El módulo `Digest::SHA1` proporciona una interfaz procesal para el uso simple, tan bien como un objeto orientado a la interfaz que pueden manejar mensajes de la longitud arbitraria y que puede leer archivos directamente.

### **d. HTML-Parser.**

Clase del programa de análisis del HTML.

El módulo `HTML::Parser`, es un analizador gramatical, reconocerán el margen y lo separarán del texto simple (el contenido de datos de alias) en documentos de HTML. Como las clases diferentes de margen y texto son reconocidas, invocan los tratantes correspondientes del acontecimiento.

`HTML::Parser`, el programa de análisis no es un programa de análisis genérico del SGML. Hemos intentado hacerlo capaz de tratar el HTML que está realmente "fuera de allí", y analiza normalmente tan de cerca como sea posible a la manera que los browsers populares de la web lo hacen en vez terminantemente de seguir una de las muchas especificaciones del HTML de W3C. Donde hay desacuerdo, hay a menudo una opción que usted puede permitir conseguir con el comportamiento oficial.



El documento que se analizará se puede proveer en segmentos arbitrarios. Esto pone en marcha el análisis mientras que los documentos se reciben de la red posible.

#### **e. ExtUtils-MakeMaker.**

Cree un makefile del módulo.

Esta utilidad se diseña para escribir un makefile para un módulo de la extensión de un makefile.PL. Se basa en el modelo makefile.SH proporcionado por Andy Dougherty y el perl5-porters. Parte de la tarea de generar el makefile en varios subprogramas que puedan ser eliminados individualmente. Cada subprograma devuelve el texto que desea haber escrito al makefile.

MakeMaker es objeto orientado. Cada directorio debajo del directorio actual que contiene un makefile. Se trata el PL como un objeto separado. Esto permite escribir un número ilimitado de makefiles con una sola invocación de WriteMakefile().

#### **f. HTML::Tagset.**

Los datos útiles tabulan en el análisis del HTML.

Este módulo contiene varias consolas de datos útiles en varias clases de operaciones de análisis de HTML.

### **6.3.8 ANTIVIRUS.**

Los virus informáticos son programas diseñados para reproducirse a sí mismos y evitar su detección para interferir con el hardware, con su sistema operativo y los archivos de una computadora, ejecutando en él acciones no deseadas por sus usuarios habituales.

Como cualquier otro programa informático, un virus debe ser ejecutado para que funcione: es decir, el ordenador debe cargar el virus desde la memoria del ordenador y seguir sus instrucciones. El potencial de daño de un virus informático no depende de su complejidad sino del entorno donde actúa.

Pero existen los métodos para contrarrestar los virus como son los **antivirus**. Un antivirus es un programa de computadora cuyo propósito es combatir y erradicar los virus informáticos. Para que el antivirus sea productivo y efectivo hay que configurarlo cuidadosamente de tal forma que aprovechemos todas las cualidades que ellos poseen. Hay que saber cuáles son sus fortalezas y debilidades y tenerlas en cuenta a la hora de enfrentar a



los virus, ya que no sólo no será eficaz para el 100% de los casos, sino que nunca será una protección total ni definitiva.

En la actualidad existen muchos antivirus para combatir este problema, cada uno de ellos dirigido a diferentes entornos de acuerdo a la necesidad del usuario, algunos antivirus se realizan de manera automática en la revisión, uno de los antivirus más utilizado es el siguiente:

#### a. **CLAMAV.**

ClamAV es otra herramienta utilizada en la inspección de mensajes electrónicos que permite identificar si el contenido del correo es un virus. Debido a la misma proliferación de este tipo de ejecutables, ClamAV consulta una base de datos (operada por sus mismos creadores) en la que se compara el contenido del mensaje attachment para ser específico con una lista de ejecutables malignos conocidos.

Al igual que Spamassasin, ClamAV también es un pre-procesador de correos que inspecciona el contenido antes que el usuario descargue su correo a una PC.

También los parámetros de ClamAV son extensos, sin embargo, sus valores predefinidos ("default") son razonables. Antes de proceder con la ejecución de esta configuración, es necesario editar un sólo valor dentro del archivo clamav.conf, simplemente elimine o comente el primer renglón que inicia con Example.

#### **Configuración de clamav.**

La configuración de ClamAV se concentra en dos archivos :

**clamav.conf** este archivo contiene parámetros globales de ClamAV, como serían la generación de registros ("Logs") y límites de archivos a inspeccionar.

**freshclam.conf** incluye la configuración para consultar la Base de Datos ClamAV y así actualizar la información local referente a los virus más recientes.

Ambos ubicados bajo el directorio /usr/local/etc/.

#### **Ejecución de ClamAV.**

La ejecución adecuada de ClamAV consta de dos partes:

1. Es opcional, es el proceso de actualización para la Base de Datos local con Virus. En base a las configuraciones antes mencionadas, ClamAV puede



actualizar constantemente un depósito local con información de los Virus más recientes en circulación de correos electrónicos, puesto que esta configuración es la más recomendable, la actualización manual no será descrita en esta guía.

El proceso de actualización para ClamAV es llevado a cabo por el "Daemon" **clamd**, para que este proceso permanezca activo, simplemente ejecute el comando **clamd &**, de esta manera quedará ejecutándose de fondo y por consiguiente actualizando la información de ClamAV según haya sido configurada.

2. El segundo proceso de ejecución consta en aplicar ClamAV a todo mensaje electrónico entrante.

Recientemente se ha publicado una actualización que soluciona problemas de vulnerabilidades en este juego de herramientas antivirus de alta calidad para **UNIX**.

La **versión 0.88** de ClamAV, que soluciona un desbordamiento de búfer potencial en el módulo de gestión de ficheros **UPX**. Se han revisado los módulos UPX, FSG y Petite, y otras mejoras relacionadas con la seguridad de ClamAV.

#### **b. TNEF (Transport Neutral Encapsulation Format).**

Formato de Encapsulación Neutral de Transporte.

El Formato de Encapsulación Neutral de Transporte (TNEF) es un método propietario del accesorio de E-mail usado por Microsoft Outlook y Microsoft Exchange Server de Microsoft.

Es utilizado para codificar los mensajes ricos del formato de texto (rtf) enviados a través del Internet.

Un archivo unido con la codificación de TNEF es winmail, los más generalmente posibles son winmail.dat o win.dat.

Los archivos de algún TNEF contienen solamente la información usada por Outlook para generar una vista rica ajustada al formato del mensaje, documentos (OLE) encajados o características Outlook-specific tales como formas, botones de votación, y peticiones de la reunión. Otros archivos de TNEF pueden contener los archivos que se han unido a un mensaje de E-mail.

Los accesorios de TNEF pueden contener la información sensible de la seguridad tal como: nombre de la conexión del usuario y trayectorias del archivo, de qué controles de acceso podrían ser deducidos posiblemente.



Esto hace uso de los accesorios de TNEF por organizaciones de los E.E.U.U. conforme a los argumentos de la regla de HIPAA o de Sarbanes-Oxley, si no francamente sería ilegal.

No hay funciones Internet-standard de E-mail que requieran la codificación de TNEF, y cualquier archivo que se pueda enviar usando este método se puede también enviar usando el estándar del MIME apoyado por todos los servidores de correo y casi todos los clientes de correo. Sin embargo, hay algunas funciones de la infraestructura propietaria del E-mail de Microsoft (Microsoft Exchange Server) que requieren TNEF.

Cuando un mensaje que contiene la información de TNEF es recibido por un cliente de correo que no entienda TNEF, un accesorio se crea con el Winmail.dat. En algunos casos, el mail server buscará y pelará fuera del TNEF y del Winmail información del dat.

Hay otros servidores que se pueden configurar para buscar TNEF y el winmail.dat lo saca hacia fuera, pero el síntoma generalmente en ese caso es que los servidores quitan totalmente el accesorio del winmail.dat.

### **c. MAILDROP.**

Es un filtro de correo semejante a procmail, una de las características que posee es el soporte para buzones maildir.

Este programa se distribuye bajo términos de la licencia pública general del GNU.

Maildrop es un reemplazo para su agente local de la entrega del correo. Maildrop lee un mensaje del correo de la entrada entonces entrega el mensaje a su buzón. Maildrop sabe cómo entregar el correo a buzones de mbox-style mailboxes, y maildirs. "Maildir" es un formato de buzón utilizado por Courier y qmail.

Maildrop lee opcionalmente las instrucciones de un archivo, que describe cómo filtrar la correspondencia recibida. Estas instrucciones pueden dirigir maildrop a entregar el mensaje a un buzón alterno, o lo adelanta en otra parte.

Maildrop se escribe en C++, y es apreciablemente más grande que procmail. Sin embargo, utiliza los recursos mucho más eficientemente. Procmail desemejante, maildrop no leerá un mensaje de correo de 10 megabyte en la memoria, los mensajes grandes se salvan en un archivo temporario, y son filtrados del archivo temporal. Si la entrada uniforme al maildrop es un archivo, y no un pipe (tubo), un archivo temporal no sera necesario.

Maildrop antes de entregar un mensaje verifica la sintaxis de la instrucción antes de la entrega del filtro correo del archivo.



## VII. DISEÑO METODOLÓGICO

Luego de haber realizado el análisis requerido y de conocer cuidadosamente todos y cada uno de los paquetes que se ofrecen en la Internet necesarios para desarrollar un servidor de correo, es el momento de implementar nuestro propio servidor de correo **LIUKAR**, del cual detallaremos su configuración en el orden en que este fue instalado y lo que fue preciso realizar para su correcta instalación.

### Obtención del código fuente.

Una vez que verificamos si teníamos un sistema que cumple con los requisitos para instalar gmail, y cuales eran los requerimientos para usar el mismo. El primer paso que dimos fue obtener el código fuente para gmail y otros complementos. Para esto necesitamos gmail, claro, y los paquetes ucspi-tcp, Squirrelmail y PHP entre otros.

El gmail que se instala por si mismo a partir de un paquete es más fácil de configurar y funciona con mayor suavidad. Solo se instala gmail una vez por cada sistema, pero es probable que luego tenga más de una oportunidad para reconfigurarlo o para entender por que el mail no se entrega como uno lo quisiera.

Por esta razón decidimos instalar gmail desde cero usando el archivo **tar.gz** del código fuente, y no un .rpm de RedHat u otra utilidad de instalación. Antes de detallar la instalación hemos listado en la tabla 7.1 todos los paquetes y módulos a utilizar.

**Tabla 7.1. Paquetes a utilizar en el Servidor LIUKAR.**

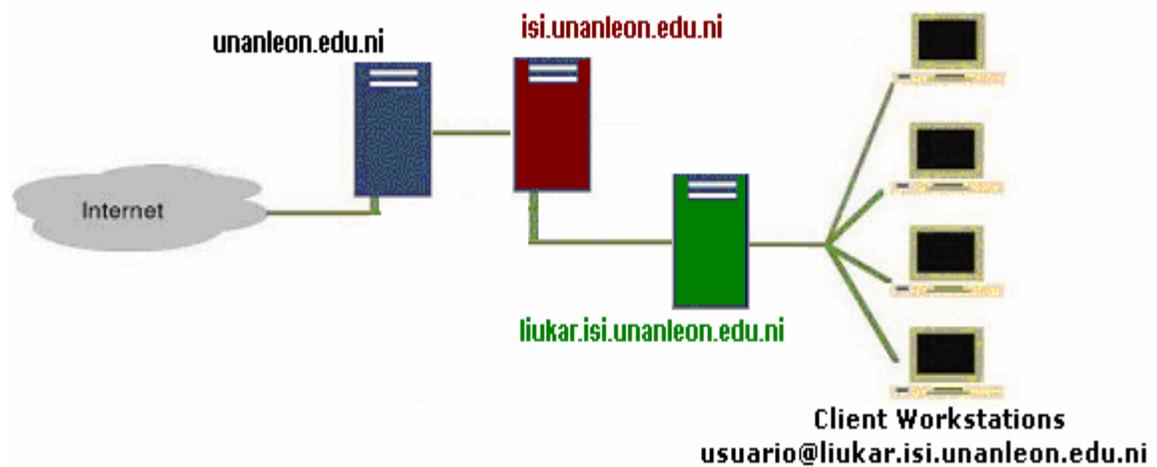
gmail-1.03.tar.gz	AntiSpam:Mail-SpamAssassin-3.0.4
ucspi-tcp-0.88.tar.gz	perl-5.8.7.tar.gz
php-5.0.4.tar.gz	Módulos para el antispam:
squirrelmail-1.5.0.tar.gz	Digest-SHA1-2.10.tar.gz
daemontools-0.76.tar	HTML-Parser-3.45.tar.gz
dot-forward-0.71.tar.gz	ExtUtils-MakeMaker-6.30.tar.gz
Quotas	HTML-Tageset-3.04.tar.gz
	Antivirus: Clamav-0.88
	tnef-1.3.4.tar.gz
	maildrop-2.0.1.tar.bz2



## 7.1 Mailserver de la organización.

Se trata de proporcionar el servicio de email a la organización, "isi.unanleon.edu.ni", la cual utilizará a "liukar" como servidor de correo. Las direcciones de los usuarios serán de la forma "usuario@liukar.isi.unanleon.edu.ni", y las estaciones obtendrán sus mensajes usando POP o IMAP.

Una visión de la estructura de esta organización es observada en la figura 7.1.



**Figura 7.1. Jerarquía del Servidor LIUKAR.**

### Configurar el nameserver.

Nuestro servidor no tiene salida directa hacia Internet ya que depende de otro servidor, haciéndose necesario que se registre nuestro servidor de correo "LIUKAR" en el servidor de "ISI".

Nos interesa la configuración del registro MX para el dominio (isi.unanleon.edu.ni) de nuestra organización. Asumiendo que el nameserver utiliza bind, el archivo de la zona isi.unanleon.edu.ni contiene algo como lo que se muestra en la figura 7.2 en el que se debe de registrar a nuestro servidor en este caso LIUKAR.



```
$TTL 86400
@      IN      SOA   @  root.localhost (
        4 ; serial
        28800 ; refresh
        7200 ; retry
        604800 ; expire
        86400 ; ttl
        )

@      IN      NS    localhost.

@      IN      MX    10    liukar

liukar IN      A      172.16.4.244
ns     IN      A      172.16.0.1
; otros hosts
@      IN      SOA   @  root.localhost (
        4 ; serial
        28800 ; refresh
        7200 ; retry
        604800 ; expire
        86400 ; ttl
        )

@      IN      NS    localhost.
```

**Figura 7.2 Fragmento del archivo para obtener permisos de entradas/salidas remotas de los correos.**

Con esto, todo el correo de Internet de la forma "usuario@liukar.isi.unanleon.edu.ni" se destina hacia el servidor de correo electrónico, liukar.isi.unanleon.edu.ni

## 7.2 Instalación de gmail.

### 7.2.1 Verificación del entorno de compilación.

```
# cc
```

```
cc: No input files specified
```

```
#
```

significa que tiene un compilador de C en la ruta de ficheros.



### 7.2.2 Crear directorio de trabajo para qmail.

```
# cd /usr/local
# tar -xzf qmail-1.03.tar.gz
# cd qmail-1.03
# mkdir /var/qmail
```

La sugerencia de la instalación es el directorio `/var/qmail` (hay pocos motivos para cambiarlo). Si se desea emplear otra ubicación, esta se debe registrar en el archivo `conf-qmail`. En lo que sigue, asumiremos la ruta arriba indicada.

### 7.2.3 Grupos y Usuarios para qmail.

qmail requiere de la creación de dos grupos y de la creación de diversos usuarios para su correcta ejecución. Estos son: *alias*, *qmaild*, *qmail*, *qmailp*, *qmailq*, *qmailr*, y *qmails*.

Grupo No. 1	<pre># groupadd nofiles # useradd -g nofiles -d /var/qmail/alias alias # useradd -g nofiles -d /var/qmail qmaild # useradd -g nofiles -d /var/qmail qmail # useradd -g nofiles -d /var/qmail qmailp</pre>
Grupo No. 2	<pre># groupadd qmail # useradd -g qmail -d /var/qmail qmailq # useradd -g qmail -d /var/qmail qmailr # useradd -g qmail -d /var/qmail qmails</pre>

### 7.2.4 Compilar y configurar el directorio de trabajo.

Aquí se generan los ejecutables de **qmail** y se prepara el directorio de trabajo de qmail.

```
# make setup check
```

### 7.2.5 Especificar el nombre del host.

Especificamos el nombre de nuestro host (incluyendo el dominio completo).



```
# ./config-fast liukar.isi.unanleon.edu.ni
```

### 7.2.6 Establecer los "alias" del sistema.

Existen tres alias del sistema que hay que crear en todas las instalaciones de qmail:

Alias	Finalidad
postmaster	Apunta al administrador de correo.
mailer-daemon	El destinatario de hecho para algunas devoluciones.
root	Redirige el correo de la cuenta del superusuario a la del administrador del sistema.

```
# (cd ~alias; touch .qmail-postmaster \
    .qmail-mailer-daemon .qmail-root)
# chmod 644 ~alias/.qmail*
```

### 7.2.7 Especificar el agente de procesamiento de correo.

El correo dirigido a los usuarios locales debe ser almacenado en algún archivo o directorio (el *mailbox*), a este directorio (mailbox) le cambiamos los permisos.

```
# cp /var/qmail/boot/proc /var/qmail/rc
# chmod 1777 /var/spool/mail
```

### 7.2.8 Eliminar Sendmail.

a. Nos aseguramos de que sendmail no está en ejecución.

```
# ps ax | grep sendmail
698 ?      S    0:00 sendmail: accepting connections
787 pts/0  S    0:00 grep sendmail
```

La primera línea indica que sendmail está en ejecución, por lo que debemos hacer que termine.

b. Hicimos que terminara sendmail.

```
# service sendmail stop
```



- c. Nos aseguramos de que sendmail no se vuelva a ejecutar.

```
# rpm -e sendmail
```

- d. Para forzar la desinstalación de sendmail y otro paquetes de él.

```
# rpm -e --nodeps sendmail  
# rpm -e --nodeps postfiz
```

- e. Volvemos a lanzar "ps" para asegurarnos que no esta Sendmail.

```
# ps ax | grep sendmail  
1170 pts/0 S 0:0 grep sendmail
```

### 7.2.9 Instalar el "reemplazo" de sendmail.

```
# ln -s /var/qmail/bin/sendmail /usr/lib/sendmail  
# ln -s /var/qmail/bin/sendmail /usr/sbin/sendmail
```

Diversos programas asumen la existencia de sendmail y lo invocan ciegamente. Por esto, **qmail** proporciona un "reemplazo" básico para sendmail, a fin de mantener operativas a las aplicaciones mencionadas.

### 7.2.10 Instalar los manuales.

qmail proporciona páginas de manual para diversas utilidades. Estas se instalan en /var/qmail/man. Sin embargo, el sistema "man" debe ser configurado para acceder a éste.

Añadimos el directorio de los manuales mediante la directiva MANPATH en el archivo /etc/man.config:

```
# vi /etc/man.config
```

```
...  
MANPATH /usr/share/man  
MANPATH /usr/man  
MANPATH /usr/X11R6/man  
MANPATH /usr/lib/perl5/man  
MANPATH /usr/kerberos/man  
MANPATH /usr/local/man
```



```
# Añadir para qmail
MANPATH /var/qmail/man
...
Luego probar algo como man qmail-send.
```

## 7.3 Instalación de ucspi-tcp (tcpserver).

### 7.3.1 Descomprimir.

```
# cd /usr/local
# tar xvzf /usr/local/ucspi-tcp-0.88.tar.gz
```

Todos los pasos de aquí en adelante serán desde adentro del directorio de ucspi-tcp-0.88.

### 7.3.2 Compilar e instalar tcpserver.

#### a. Compilar.

```
# cd ucspi-tcp-0.88
# make
# make setup check
```

Obtuvimos una serie de ejecutables en el mismo directorio.

#### b. Copiar ejecutables.

Copiamos los ejecutables **tcpserver** y **tcprules** a un directorio en el PATH. Sólo estos dos ejecutables serán necesarios.

```
# cp tcpserver tcprules /usr/sbin
# cp tcpserver tcpserver /usr/local/bin
```

## 7.4 Configurar el inicio automático de qmail.

Configuramos el sistema para que siempre se ejecute qmail al reiniciarse el computador.



### 7.4.1 Averiguar el UID y el GID del usuario "qmaild" y del grupo "nofiles".

El número 502 es el UID del usuario "qmaild", y el número asociado al grupo "nofiles" es 501. Es seguro que estos valores serán distintos en otro sistema.

```
# id qmaild
uid=502(qmaild) gid 501(nofiles)
groups =501(nofiles)
```

### 7.4.2 Añadir al final del archivo /etc/rc.d/rc.local.

```
# vi /etc/rc.d/rc.local
Dentro del archivo añadir el siguiente comando:
csh -cf '/var/qmail/rc &'
/usr/sbin/tcpserver -u 502 -g 500 0 smtp \
    /var/qmail/bin/qmail-smtpd &
```

## 7.5 Probando qmail.

qmail enviará mensajes de diagnóstico a syslog, por lo cual normalmente lo buscamos en el archivo /var/log/maillog que es donde syslog imprime los mensajes de email en RedHat (para más información, ver /etc/syslog.conf y el manual de syslog.conf).

### Es necesario:

Disponer de un computador auxiliar configurado para enviar y recibir correo.

### 7.5.1 Iniciando qmail.

- Reiniciamos el computador.
- Analizamos si los procesos de qmail están en ejecución realizando lo siguiente:

```
# ps axu | grep qmail
qmails 749  0.0 0.1  1360 228 ? S 13:20  0:00  qmail-send
qmaild 750  0.0 0.2  1376 364 ? S 13:20  0:00  usr/bi/tcpserver
qmaill 774  0.0 0.1  1328 348 ? S 13:20  0:00  splogger qmail
root   775  0.0 0.1  1328 220 ? S 13:20  0:00  qmail-lspawn |pre
qmailr 776  0.0 0.1  1324 212 ? S 13:20  0:00  qmail-rspawn
qmailq 777  0.0 0.5  1320 228 ? S 13:20  0:00  qmail-clean
root   1477 0.0 0.1  2632 668 ? S 13:20  0:00  grep qmail
```



Casi todos los valores numéricos serán distintos según la computadora con que se este trabajando (incluso hemos recortado un poco la salida por cuestiones de formato). Lo importante es el nombre de los procesos, y de los usuarios dueños de los mismos (a la extrema derecha e izquierda, respectivamente).

- c. En caso de no poder reiniciar el computador en este momento, es necesario inicializar qmail manualmente.

```
# csh -cf '/var/qmail/rc &'
# /usr/sbin/tcpserver -u 502 -g 500 0 smtp \
  /var/qmail/bin/qmail-smtpd &
```

### 7.5.2 Probar el delivery.

Observamos si los mensajes de qmail pueden ser distribuidos, es decir, si estos pueden ser enviados a los usuarios de nuestro computador o de otro, para ello realizamos las siguientes pruebas:

- a. "inyectamos" un mensaje con destino local **existente**.

```
# echo to: mariaexiste | /var/qmail/bin/qmail-inject
# mail
```

- b. "inyectamos" un mensaje con destino local **inexistente**.

```
# echo to: marianoexiste | /var/qmail/bin/qmail-inject
# mail
```

- c. "inyectamos" un mensaje a un computador **remoto**.

```
# echo to: mariaremoto@yahoo.com | /var/qmail/bin/qmail-inject
# mail
```

Pero esto requiere que qmail se conecte al puerto SMTP de "remoto" (donde también deberá haber un servidor de email) y que en él exista el usuario especificado.



### 7.5.3 Probar la recepción.

Observamos si **qmail** estaba aceptando mensajes, para esto tcpserver debe estar "escuchando" en el puerto SMTP (25). Existen dos formas de comprobarlo:

```
# netstat -a --inet | grep smtp
tcp 0 0 liukar:smtp *:* LISTEN
```

```
# netstat -an --inet | grep 25
tcp 0 0 liukar:1025 *:* LISTEN
```

## 7.6 Aceptar y remitir mensajes.

### 7.6.1 qmail debe aceptar correos para el dominio.

Cuando se instala qmail, éste asume que sólo debe aceptar mensajes destinados a su propio host; por eso configuramos qmail para que acepte las direcciones que son diferentes al de nuestro host. Para esto sólo añadimos nuestro dominio "isi.unanleon.edu.ni" (la parte de "host" de las direcciones electrónicas) a los archivos /var/qmail/control/rcpthosts y /var/qmail/control/locals:

```
# vi /var/qmail/control/rcpthosts
# vi /var/qmail/control/locals
```

En los dos archivos se pone lo mismo respectivamente:  
liukar.isi.unanleon.edu.ni  
isi.unanleon.edu.ni

El archivo rcpthosts permite que **qmail-smtp** acepte los mensajes con el host especificado.

El archivo locals permite que **qmail-send** efectúe el delivery a un usuario local (en su mailbox).

Con la configuración actual, los usuarios inscritos en liukar.isi.unanleon.edu.ni ya pueden recibir correo desde sus estaciones (vía POP y/o IMAP). Sin embargo, aún no pueden enviar correo.



## 7.6.2 Enviar correo desde las estaciones.

qmail deberá ser capaz de recibir correo desde las estaciones y redirigirlo hacia el exterior, esto es, servir de RELAY a las estaciones.

Esto normalmente está deshabilitado, pues en caso contrario cualquier persona de Internet podría usar nuestro servidor para enviar correo SPAM. Sólo nuestras estaciones deberán habilitarse.

Lamentablemente qmail (en la versión actual) no proporciona un mecanismo sencillo para habilitar el RELAY. Para esto utilizamos la variable de entorno RELAYCLIENT.

- Creamos un archivo llamado /etc/tcp.smtp con el siguiente contenido:

```
127.:allow,RELAYCLIENT=""
172.16.1.20:allow,RELAYCLIENT=""
172.16.1.21:allow,RELAYCLIENT=""
172.16.1.22:allow,RELAYCLIENT=""
172.16.1.23:allow,RELAYCLIENT=""
```

- Compilamos el archivo para crear el acceso rápido en /etc/tcp.smtp.cdb.

```
# /usr/sbin/tcpserver /etc/tcp.smtp.cdb \
/etc/tcp.smtp.tmp < /etc/tcp.smtp
```

- Modificamos la invocación a tcpserver en /etc/rc.d/rc.local del siguiente modo:

```
csh -cf '/var/qmail/rc &'
/usr/sbin/tcpserver -x /etc/tcp.smtp.cdb -u 502 \
-g 501 0 smtp /var/qmail/bin/qmail-smtpd &
```

## 7.7 Instalación del webmail (squirrelmail).

### 7.7.1 Instalar PHP4.

Debido a que Squirrelmail es un programa webmail escrito en PHP4, se hace necesario tener este procesador para que trabaje correctamente con nuestro webmail, donde Squirrelmail utiliza la extensión .php para los archivos PHP4.



```
# cd /usr/local
# tar xvfz php-5.0.4.tar.gz
# cd php-4.0b
# ./configure --enable-track-vars --enable-force-cgi-redirect --with-gettext
# make
```

### 7.7.2 Instalar Squirrelmail.

Se tiene que modificar el archivo xinetd.d

```
# cd /etc/xinetd.d
# vi imap
Cambiar:
    Donde dice disable = yes
    Poner disable = no
```

Luego de haber realizado los cambios se hace lo siguiente:

```
# mkdir /var/www/html/mail
# cd /var/www/html/mail
# tar xvfz squirrelmail-1.5.0.tar.gz
# cd squirrelmail-1.5.0

Dar el permiso al servidor web.
# chown -R apache:apache data

Crear el directorio donde los archivos insertados serán guardados
antes que se envíen.
# cd /var/
# mkdir buzón
# chgrp -R buzón
# chmod 730 buzón
```

### Configuración de Squirrelmail.

Después de instalar Squirrelmail se tiene que configurar.

```
# cd /var/www/html/mail/squirrelmail-1.5.0
# ./configure
```

Aparecerá un menú en el cual se tiene que seleccionar lo siguiente:



- 1. Preferencia de la organización.
  - 1. Nombre de la organización: LIUKAR
- 2. Ajustes del servidor.
  - 1. Dominio: liukar.isi.unanleon.edu.ni
- Modificar el archivo de /etc/http/conf/httpd.conf.
  - # vi etc/http/conf/httpd.conf
  - Cambiar donde dice:
    - DocumentRoot "/var/www/html"
  - Poner:
    - DocumentRoot "/var/www/html/mail/squirrelmail-1.5.0"
  - Esto se realiza con el fin de que la página de correo será la principal.

## 7.8 Instalación de daemontools.

```
# cd /usr/local
# qunzip daemontools-0.76.tar
# tar -xpf daemontools-0.76.tar
# cd admin/daemontools-0.76
# package/intall
```

Se comprueba que está instalado buscando los siguientes archivos:

```
"/bin/sh/command/svscanboot"
"/syscan/service"
```

## 7.9 Instalación de dot-forward.

```
*** Descomprimirlo
# cd /usr/local/dot-forward-0.71
# make setup check
```

## 7.10 Formato maildir.

1. **Crear la carpeta automáticamente para cada usuario.**

```
# /var/qmail/bin/maildirmake /etc/skel/Maildir
```



## 2. Cambiar el archivo `/etc/login.defs`

Descomentar la línea `QMAIL_DIR Maildir/`  
Por tanto el archivo quedará de la siguiente forma:

```
QMAIL_DIR Maildir/  
#MAIL_DIR /var/spool/mail  
#MAIL_FILE .mail
```

## 3. Crea el archivo "imap" en el directorio `/etc/xinetd.d`

```
# vi /etc/xinetd.d/imap.rpmsave  
# cp /etc/xinetd.d/imap.rpmsave /etc/xinetd.d/imap
```

## 4. Contenido de `/etc/xinetd.d/imap`

Cambiar donde dice `disable = Yes` poner `NO`, quedando el archivo de la siguiente forma:

```
# default: off  
# description: The IMAP service allows remote users  
# to access their mail using \  
# an IMAP client such as Mutt, Pine, fetchmail,  
# or Netscape \ Communicator.  
service imap  
{  
    disable          = no  
    socket_type      = stream  
    wait             = no  
    user             = root  
    server           = /usr/sbin/imapd  
    log_on_success   += HOST DURATION  
    log_on_failure   += HOST  
}
```

## 5. Permita que los usuarios locales inyecten correo por medio de SMTP.

```
# echo '127.:allow,RELAYCLIENT=""' >> /etc/tcp.smtp  
# /usr/local/sbin/qmail cdb
```



## 7.11 Asignación de cuotas a los usuarios.

Es necesario establecer cuotas al Sistema de Correo. Donde nuestro servidor le proporciona un espacio limitado de 5 MB a cada usuario registrado en el sistema.

### Procedimientos.

1. Debe iniciarse el sistema en nivel de corrida 1 (mono usuario), ya que se requiere no existan procesos activos utilizando el contenido de la partición a la cual se le aplicará la cuota de disco.

```
# vi etc/inittab
dentro de aquí cambiar
id: 5 :intdefault cambiar por id: 1 :intdefault
```

2. Obviamente, durante la instalación Red Hat Linux, debió asignarse una partición dedicada al límite de cuotas, estos son los directorios /var y /home.
3. Con la finalidad de añadir el soporte para cuotas en las particiones anteriormente mencionadas, se debe añadir en el fichero /etc/fstab los parámetros *usrquota* y *grpquota* (defaults,usrquota,grpquota) a las líneas que definen la configuración de las particiones /var y /home:

```
LABEL= /var    /var    ext2 defaults,usrquota,grpquota 1 2
LABEL= /home  /home  ext2 defaults,usrquota,grpquota 1 2
```

4. Debe remontar las particiones para que surtan efecto los cambios:

```
mount -o remount /var
mount -o remount /home
```

5. Se deben crear los ficheros *aquota.user*, *aquota.group*, *quota.user* y *quota.group*, los cuales se utilizarán en adelante para almacenar la información y estado de las cuotas en cada partición.

```
cd /var
touch aquota.user aquota.group quota.user quota.group

cd /home
touch aquota.user aquota.group quota.user quota.group
```

6. Ejecutar:

```
quotacheck -avug
```



La primera vez que se ejecuta el mandato anterior es normal que marque advertencias refiriéndose a posibles ficheros truncados que en realidad no eran otra cosa sino ficheros de texto simples vacíos a los cuales se les acaba de convertir en formato binario. Si se ejecuta de nuevo quotacheck - avug, no deberá mostrar advertencia alguna.

7. Para activar las cuotas de disco recién configuradas, solo bastará ejecutar:

```
quotaon /var
quotaon /home
```

8. Ir al nivel de corrida 3 (init 3) a fin de aplicar cuota de disco a algunos usuarios. (Estará en modo Texto, multiusuario).

```
# vi etc/inittab
dentro de aquí cambiar
id: 1 :intdefault  cambiar por  id: 3 :intdefault
```

9. Asignar cuotas de disco a cualquier usuario o grupo, solo hará falta utilizar edquota citando el nombre del usuario al cual se le quiere aplicar:

```
edquota maria
```

Lo anterior deberá devolver algo como lo siguiente a través de vi u otro editor de texto simple:

```
Disk quotas for user maria (uid 500):
Filesystem  blocks soft hard inodes soft hard
/dev/hda6   0      0  0      0      0  0
/dev/hda5   24     0  0      10     0  0
```

10. Suponiendo que se quiere asignar una cuota de disco de 5 MB con una tolerancia de hasta 6 MB por hasta 6 días:

```
Disk quotas for user maria (uid 500):
Filesystem  blocks soft  hard  inodes soft  hard
/dev/hda6   0     5120  6144    0     0    0
/dev/hda5   24     5120  6144   10     0    0
```

11. Si se quiere que todo aplique para los usuarios existentes, a partir de UID 510, por ejemplo, suponiendo que tiene al usuario "maria" como molde (note por favor el acento grave en el mandato justo antes de awk, no es una comilla ni apostrofe):

```
edquota -p maria `awk -F: '$3 > 510 {print $1}' /etc/passwd`
```



Asumiendo que estés usando csh, y que asignes el UID de tus usuarios a partir de 511 en adelante.

12. Cambiar el archivo y ponerlo como estaba al inicio. En id: **5** :init

```
# vi etc/inittab
dentro de aquí cambiar
id: 3 :intdefault cambiar por id: 5 :intdefault
```

## 7.12 Instalación de Spamassassin (Antispam).

SpamAssassin necesita de otros programas o módulos antes de ser instalado para que no de ningún tipo de problemas. Veamos algunos de estos.

### 7.12.1 Instalando Perl-5.7.8.

```
Descomprimirlo desde el explorador
# cd /usr/src/qmailsource/
# cd perl-5.8.7
# ./Configure
# make
# make install
```

### 7.12.2 Instalar los Módulos siguientes.

a. Cada uno de los módulos a instalar fueron descomprimidos en: /usr/src/qmailsource/ desde el explorador de Linux.

- Digest-SHA1-2.10.tar.gz
- HTML-Parser-3.45.tar.gz
- ExtUtils-MakeMaker-6.30.tar.gz
- HTML-Tageset-3.04.tar.gz

b. Ejecutar cada módulo:

```
# cd /usr/src/qmailsource
# cd Digest-SHA1-2.10
Entrar en el directorio del módulo y escribir:
# perl Makefile.PL
# make
# make test
# make install
```



### 7.12.3 Instalación de SpamAssassin.

```
# cd /usr/src/qmailsource
# tar xvjf Mail-SpamAssassin-3.0.4.tar.bz2
# cd Mail-SpamAssassin-2.63
# perl Makefile.PL PREFIX=/usr SYSCONFDIR=/etc
# make
# make install
Agregamos al grupo y al usuario bajo los cuales el
Spamassassin vaya a girar.

# groupadd spamd
# useradd -g spamd -s /bin/false spamd
```

### 7.12.4 Instalar el script de SpamAssassin.

```
# cd /etc/rc.d
# vi rc.spamassassin
```

Se añade el siguiente contenido script:

```
#!/bin/sh
# Spamd init script for Slackware 9.0
# August, 2th 2003
# Martin Ostlund, nomicon
# Altered by: Claudio Borges
# Sat Apr 17 09:08:29 BRT 2004

PATH=/sbin:/bin:/usr/sbin:/usr/bin:/usr/local/bin
DAEMON=/usr/bin/spamd
NAME=spamd
SNAME=rc.spamassassin
DESC="SpamAssassin Mail Filter Daemon"
PIDFILE="/var/run/$NAME.pid"
PNAME="spamd"
DOPTIONS="-x -u spamd -d --pidfile=$PIDFILE"

KILL="/bin/kill"
KILLALL="/bin/killall"
# Defaults - don't touch, edit /etc/mail/spamassassin/local.cf
ENABLED=0
OPTIONS=""
```



```
set -e
case "$1" in
start)
    echo -n "Starting $DESC: "
    $PNAME $OPTIONS $DOPTIONS

    echo "$NAME."
    ;;
stop)
    echo -n "Stopping $DESC: "
    $KILL -9 `cat $PIDFILE`
    /bin/rm $PIDFILE
    echo "$NAME."
    ;;
restart|force-reload)
    echo -n "Restarting $DESC: "
    $0 stop
    $0 start

    echo "$NAME."
    ;;
*)
    ME=/etc/rc.d/$SNAME
    echo "Usage: $ME {start|stop|restart|force-reload}" >&2
    exit 1
    ;;
esac
exit 0
```

### 7.12.5 Inicializar SpamAssassin.

```
# chmod +x /etc/rc.d/rc.spamassassin
# echo "/etc/rc.d/rc.spamassassin start" >> /etc/rc.d/rc.local
```

### 7.12.6 Crear el fichero .cf de SpamAssassin.

```
# /etc/mail/spamassassin
# vi local.cf

required_hits 8.5
rewrite_subject 1
subject_tag [SPAM]
report_header 1
```



### 7.12.7 Comprobar la instalación de SpamAssassin.

```
#spamassassin -t < sample-nospam.txt > nospam.out
#spamassassin -t < sample-spam.txt > spam.out

Ver proceso de SpamAssassin
#ps auxw | grep -i spam
```

### 7.13 Instalación de Clamav (Antivirus).

El clamav al igual que el spamassassin, tambien necesita de otros paquetes para poder realizar su función adecuadamente.

#### 7.13.1 Instalar TNEF.

```
# cd /usr/local/src
# tar xzf tnef-1.3.4.tar.gz
# cd tnef-1.3.4
# ./configure
# make
# make install
cd ..
```

#### 7.13.2 Instalar MAILDROP.

```
# cd /usr/local/src
# bunzip2 maildrop-2.0.1.tar.bz2
# tar xf maildrop-2.0.1.tar
# chown -R root.root maildrop-2.0.1.tar
# cd maildrop-2.0.1
# ./configure
# make
# make install-strip
# make install-man
```

#### 7.13.3 Instalar CLAMAV.

- a. Descomprimirlo desde el explorador en /usr/local/src.
- b. Crear los grupos.

```
# groupadd clamav
# useradd -g clamav -s /bin/false -c "Clam AntiVirus" clamav
```

**c. Instalar Clamav.**

```
# cd /usr/local/src
# cd clamav-0.88
# ./configure
# make
# make check
# make install
```

**d. Copiar el ejecutable.**

```
Verificar que el permiso sea 755
# cd /init/RedHat
# cp clamd /etc/rc.d/init.d/
#/etc/rc.d/init.d/clamd start
```

**e. Modificar los archivos.**

```
# vi /usr/local/etc/clamd.conf
Dentro del archivo comentar la línea que dice "example" y poner lo
siguiente:
    LogSyslog
    FixStaleSocket
    User clamav
```

```
# vi /usr/local/etc/freshclam.conf
Dentro del archivo comentar la línea que dice "example" y poner lo
siguiente:
LogSyslog
DatabaseOwner clamav
DatabaseMirror db.au.clamav.net (where "au" matches your country
code)
NotifyClamd
```

**f. Activar el archivo.**

```
a) Añadir:
      # vi /etc/rc.d/rc.local
Añadir:
      /usr/local/bin/freshclam -d
b) Lanzar el archivo.
      #/usr/local/bin/freshclam -d
```

**g. Verificar la instalación.**

```
# ps axf
Buscar el siguiente archivo:
18144 ? S 0:00 /usr/local/sbin/clamd
Reinicializar el antivirus.
#/etc/rc.d/init.d/clamd start
Iniciando clamav.
```

OK

## VIII. CONCLUSIÓN

El desarrollo de nuestro proyecto monográfico esta regido por cada uno de los objetivos que nos formulamos desde el principio, los cuales fueron llevados a la práctica y alcanzados exitosamente a pesar de los tropiezos encontrados.

Como resultado final de este trabajo se dio origen al nuevo servidor llamado **LIUKAR** (LIU= Liuva y KAR= Karen), el que consigue asomarse a la gran red de Internet, este lo creamos luego de haber acoplado adecuadamente la configuración de un servidor Web y la configuración de un servidor de correo, ambos servidores se encuentran en la misma máquina. Siendo el propósito principal de LIUKAR brindar el servicio de correo electrónico.

En la implementación de este servicio todos los paquetes utilizados fueron seleccionados cuidadosamente tanto por su compatibilidad entre ellos como por ser las mejores opciones que se ofrecen en el ámbito de la configuración de servidor web y de correo electrónico.

Siendo uno de ellos y el principal el MTA qmail con el que garantizamos mejor eficiencia en el transporte de correo de cada usuario y con la ayuda del nuevo formato de buzón de correo, maildir, nos cercioramos que los mensajes no se corromperán en caso de que el sistema falle durante el envío.

En LIUKAR, se encuentran solamente las cuentas de correo de aquellos usuarios que el administrador del mismo autorice, asignando a cada usuario un espacio físico estándar de 5 MB en el servidor.

Los usuarios de LIUKAR podrán acceder a sus cuentas desde cualquier máquina que tenga acceso a Internet a través de nuestro portal Web: <http://liukar.isi.unanleon.edu.ni>, en el cual se muestra la interfaz de acceso a nuestro servidor de correo donde se solicita al usuario nombre de cuenta y contraseña, de esta manera podrá leer, borrar, reenviar, adjuntar archivos, entre otras cosas. Al igual que enviar y recibir correos tanto local como remotamente.

## IX. RECOMENDACIONES

Con la experiencia que adquirimos en LIUKAR nos permitimos sugerir algunas recomendaciones que estamos seguras serán de gran utilidad a aquellos que deseen involucrarse en este entorno.

- 1.** Tener conocimientos básicos para trabajar con Linux, principalmente con sus comandos.
- 2.** Se recomienda hacer una copia de seguridad de los actuales contenidos de estos programas, como son las páginas web y las bases de datos que estén en funcionamiento hasta el momento de la instalación.
- 3.** Si es posible, realizar las instalaciones en un sistema “de prueba”. Esto le dará la oportunidad de cometer errores sin perder correo importante o interrumpir el servicio de correo a sus usuarios.
- 4.** Si no tiene un sistema de repuesto, y su sistema ya gestiona el correo usando sendmail, qmail u otro MTA (agente de transporte de correo), puede probar la mayor parte de los componentes de qmail sin interferir con el servicio existente.
- 5.** Cuando migre su sistema de otro MTA a qmail (incluso aunque tenga alguna otra experiencia con qmail a sus espaldas) es una buena idea plantearse una planificación.
- 6.** Asegurarse de eliminar todos los MTA existentes en el sistema.
- 7.** Instalar qmail desde cero usando el archivo tar.gz del código fuente, y no un .rpm de RedHat u otra utilidad de instalación.
- 8.** Sugerimos a los estudiantes interesados en la configuración de un servidor de correo y que deseen utilizar como MTA a qmail, indagar más sobre el funcionamiento del mismo por tener una arquitectura distinta a los otros MTA y por ser muy complejo y difícil de configurar.
- 9.** Sugerimos a los estudiantes interesados en la parte de redes que pueden ocupar este documento para ampliarlo en el ámbito de seguridad de servidores web.



## X. GLOSARIO

### A

#### **ACK**

Acknowledgement (acuse de recibo) Un tipo de mensaje que se envía para indicar que un bloque de datos ha llegado a su destino sin errores. Un acuse de recibo puede también ser negativo (*no acknowledgement -- NOACK*), es decir, indicar que un bloque de datos no ha llegado a su destino.

#### **ALMACENAR**

Incluir los datos en una memoria, externa o interna a la computadora, adecuada para conservarlos. Sinónimos de este término son escribir, guardar, grabar y salvar.

#### **ANSI**

Siglas de American National Standard Institute (instituto nacional americano de estándares). Se trata de una organización americana que se encarga de la formulación de normas en diversos sectores técnicos. En Windows es el juego de códigos empleado para definir los caracteres que se introducen en los documentos.

#### **APACHE**

Software para servidores web.

#### **APLICACIÓN**

Es el problema o conjunto de problemas para los que se diseña una solución mediante computadora. Ejemplos de aplicaciones son los procesadores de texto (procesamiento o tratamiento de la palabra), las bases de datos (organización y procesamiento de datos) y las hojas de cálculo (organización y procesamiento de números). En Windows se emplea este término indistintamente con el de programa.

#### **ARCHIVO**

Es un conjunto de datos relacionados de manera lógica, como puede ser el conjunto de los nombres, direcciones y teléfonos de los empleados de una empresa determinada.

#### **ARRANCAR**

Poner en marcha una computadora o un programa.

**AT&T**

American Telephone and Telegraph Corporation. (Corporación Americana de Telefonía y Telegrafía).

**B****BINARIO**

Es un sistema de numeración en el que los dígitos se representan utilizando únicamente dos cifras, 0 y 1. Como adjetivo indica dos opciones alternativas.

**BIOS**

Siglas de Basic Input/Output System (sistema básico de entrada/salida). Es un programa cargado en ROM por el fabricante que gestiona la configuración básica del sistema. Entre otras cosas, se emplea para controlar los procesos de entrada y salida entre una computadora y sus periféricos.

**BROWSER**

Un programa que sirve para mirar o interactuar con la información contenida en la red, se conoce con el nombre de Navegadores. El primer browser con interfaz gráfica que se conoció en Internet Mosaic.

Ejemplo de browser:

Explorer, Netscape Communicator, Netscape Navigator.

**C****CABECERA**

Encabezamiento de un impreso o documento. También se aplica a la información preliminar incluida al comienzo de un bloque de datos relativa al bloque siguiente. En comunicaciones es un bloque de caracteres que indica las características del mensaje.

**CG**

Computer Graphics. Gráficos de Computador.

**CGI**

Common Gateway Interface. Interfaz de Acceso Común. Programas usados para hacer llamadas a rutinas o controlar otros programas o bases de datos desde una página Web. También pueden generar directamente HTML.

**CÓDIGO**

Es un conjunto de símbolos y reglas que sirven para representar datos de forma que puedan ser reconocidos por una computadora.

**CÓDIGO HAMMING**

Es un sistema de detección y corrección automática de errores en información electrónica. De lo que se trata, explicado básicamente, es de asociar una serie de bits de validación o paridad a los bits de datos, de tal forma que una alteración en cualquiera de esos bits de datos pueda ser detectada y corregida adecuadamente.

**COMANDO**

Término que define una instrucción, mandato u orden dado a la computadora mediante el cual el usuario le informa de las operaciones o tareas que quiere realizar con su ayuda.

**CONFIGURACIÓN**

Es un conjunto de opciones que se seleccionan antes de empezar a trabajar con un dispositivo y que sirven para especificar precisamente su modo de funcionamiento, adaptándolo a las especiales condiciones de su hardware.

**CONFIGURAR**

Desde el punto de vista de software, se refiere a establecer, desde un programa especial, las características de un dispositivo periférico; desde el punto de vista de hardware, consiste en personalizar físicamente dichas características.

**CPU**

Siglas de Central Processing Unit (unidad central de proceso). También llamado procesador, es el núcleo y componente principal de una computadora y permite controlar y procesar todas las operaciones realizadas. Parte de la computadora que contiene el procesador central. También se aplica este término al mismo procesador.

**CRACKER**

Individuo con amplios conocimientos informáticos que desprotege/piratea programas o produce daños en sistemas o redes.

**CRC**

Son las siglas de Cyclic Redundancy Control (control de redundancia cíclica). Es un método de comprobar si una transmisión de datos se ha producido o no correctamente.

**D****DATO**

Es un término genérico empleado para designar números, letras u otros caracteres existentes en una computadora o en su memoria y sobre los cuales actúan los programas.

**DATOS**

Cualquier información que pueda ser usada para cálculo, comparación u otro procesamiento o que requiera ser recordada para un uso futuro. Algunas veces, se usa para referirse a registros u otra información involucrada en un programa a diferencia del programa mismo.

**DIRECTORIO RAÍZ**

Es el directorio de nivel superior de un disco que se crea en el momento de dar formato al disco. A partir de este directorio se pueden crear otros directorios y subdirectorios, así como ficheros. También llamado directorio principal.

**DISCO**

Placa recubierta de material magnético que permite almacenar información. Se le llama así por su forma.

**DISCO DURO**

También llamado disco rígido o disco fijo. Es el soporte de almacenamiento de información más utilizado en las computadoras, por su gran capacidad. Normalmente suele ser interno a la computadora. Es un dispositivo muy delicado formado por una serie de discos apilados uno encima del otro y acomodados en un compartimiento estanco en los que se graban los datos.

**DISCO ÓPTICO**

Es un tipo de soporte de información que no emplea técnicas de grabación magnética. En él la lectura y escritura de datos se realiza mediante rayos de luz láser. Tienen una capacidad de almacenamiento muy superior a la de los discos flexibles, pero son bastante más caros y mucho menos fiables.

**DNS**

Sistema de Denominación de Dominio. Es un sistema utilizado en internet para convertir los nombres de dominios a sus nodos de red publicados abiertamente en direcciones.

**DOS**

Siglas de Disk Operating System (sistema operativo de disco). Es uno de los tipos de sistema operativo más utilizados en computadoras. Se emplea generalmente para el control de las unidades de disco.

**DIRECTIVA**

Establece reglas estrictas e impone obligaciones para garantizar la seguridad.

**F****FICHERO**

Un fichero es la unidad mínima de almacenamiento de información. Los archivos son un tipo de ficheros, es decir, son ficheros que pueden albergar otros ficheros. En general, archivo y fichero se consideran sinónimos, a excepción del entorno Windows, donde a los ficheros se les denomina archivos, es decir, todo documento en Windows se almacena en un archivo. Sin embargo, en este entorno se denomina Fichero a una utilidad incluida que es una sencilla base de datos de dos campos.

**FICHERO .htaccess?**

Es un fichero de texto que puesto en cada directorio de nuestro espacio web, permitirá personalizar y modificar ciertas acciones que va a realizar el servidor de web (http).

**FTP**

File Transfer Protocol. Protocolo de Transferencia de Archivos. Uno de los potocolos de tranferencia de ficheros mas usado en Internet.

**G****GET**

Petición normal.

**H****HACKER**

Experto en informática capaz de entrar en sistemas cuyo acceso es restringido. No necesariamente con malas intenciones.

**HARDWARE**

Es el término que indica todas las partes físicas, eléctricas y mecánicas de una computadora. Significa literalmente "partes duras" y se emplea en contraposición al término software, que significa "partes blandas", es decir, los programas de una computadora. A los componentes que es posible ver y tocar se les llama en jerga computacional "hardware", palabra inglesa cuyo significado es máquina o "cosa dura".

**HANDLER**

Representación interna en Apache de una acción a realizar cuando un documento es solicitado.

**HERRAMIENTA**

Término aplicado a un programa que desarrolla servicios específicos.

**HIPERTEXTO**

Término que se aplica a los enlaces (links) que existen en una página web y que son interpretados por los navegadores como accesos a otras páginas web. Sistema de navegación por la Web mediante vínculos implementados en páginas visualizadas.

**HIPERVÍNCULO**

También se conoce como enlace. Es una referencia a otra página que se encuentra en la red Internet. Se suele distinguir por tener un color distinto al resto del texto, estar subrayado o representado por una imagen. Al hacer click sobre él, el navegador carga la página correspondiente.

**HOST**

Anfitrión. Es cualquier ordenador que tiene un número IP y que puede tanto enviar como recibir información por una red. Computador conectado a Internet. Computador en general.

**HTM**

Extensión bajo MsDos de los ficheros de tipo HTML.

**HTML**

Lenguaje de Marcas de Hipertexto. Es el lenguaje de programación utilizado para crear las páginas de Internet. En él se definen las imágenes, los textos y los hipervínculos que estarán incluidos en la página. En HTML se permite la utilización de dos formatos de imagen: el GIF y el JPG. En algunas variaciones de este lenguaje se han incluido otros formatos gráficos como el PNG. También permite la introducción de otros contenidos como video, sonido o incluso programas que se ejecutan en nuestro ordenador al abrir la página.

**HTTP**

El protocolo usado para la transferencia de documentos en las páginas del WWW (HyperText Transfer Protocol), que permite la interacción entre los servidores y el navegador.

**I****IMPLEMENTACIÓN**

Es una forma de llevar a la práctica un determinado concepto de diseño bajo unas ciertas circunstancias.

**INFORMÁTICA**

Contracción de INFORmación autoMÁTICA. Es un campo de conocimientos que abarca todos los aspectos relacionados con computadoras y con el tratamiento automático de la información.

**INTERNET**

Conjunto de redes y ruteadores que utilizan el protocolo TCP/IP y que funciona como una sola gran red. Es la red de redes. Nacida como experimento del ministerio de defensa americano. Es la red electrónica de computadoras mas grande del mundo.

**INTRANET**

Se llaman así a las redes tipo Internet pero que son de uso interno, por ejemplo, la red corporativa de una empresa que utilizará protocolo TCP/IP y servicios similares como WWW. IP Internet Protocol. Protocolo de Internet. Bajo este se agrupan los protocolos de internet. También se refiere a las direcciones de red Internet.

**IP**

Protocolo Internet. Es un protocolo de bajo nivel para redes que describe la manera cómo el usuario puede comunicarse con los miembros Internet. Es la misma IP de TCP/IP. El IP es el identificador de cada host dentro de su red de redes.

**ISO**

International Standard Organization. Organización Internacional de Estándares.

**ISP**

Internet Service Provider. Proveedor de Servicios Internet. Ofrecen a sus clientes una conexión a la red Internet, correo electrónico, chat, news y un teléfono de consultas. Los gratuitos tienen algunas limitaciones y los de pago ofrecen más servicios con mayor calidad.

**J****JAVA**

Lenguaje de programación orientado a objeto parecido al C++. Usado en WWW para la telecarga y telejecucion de programas en el computador cliente. Desarrollado por Sun microsystems, con el propósito de mejorar las capacidades de las páginas de Web. Los programas en JAVA son llamados Applets.

**JAVASCRIPT**

Formalmente llamado LiveScript, este lenguaje fue desarrollado por Netscape. Concebido después del JAVA; su principal diferencia radica en que el programa se halla embebido en un archivo HTML, en lugar de ser un ejecutable que se carga cuando Ud. carga una página de Web.

**L**

**LAN**

Local Area Network. Red de Area Local. Una red de area local es un sistema de comunicación de alta velocidad de transmisión. Estos sistemas están diseñados para permitir la comunicación y transmisión de datos entre estaciones de trabajo inteligentes, comúnmente conocidas como Computadoras Personales. Todas las PCs, conectadas a una red local, pueden enviar y recibir información. Como su mismo nombre lo indica, una red local es un sistema que cubre distancias cortas. Una red local se limita a una planta o un edificio.

**LINK**

Enlace. Unión. Hiperenlace. Se llama así a las partes de una página WEB que nos llevan a otra parte de la misma o nos enlaza con otro servidor.

**LIUKAR**

Es la unión de las tres primeras letras de los nombres de las autoras de este trabajo monográfico.

LIU= Liuva  
KAR= Karen

**LOGIN**

Entrada de identificación, conexión. Igual que logon.

**M****MAIL**

El correo electrónico es el servicio más básico, antiguo, y más utilizado dentro de Internet. La mensajería electrónica es el medio más eficaz y más rápido de comunicación, permite intercambiar además de mensajes, programas, audio, video e imágenes.

**MEMORIA**

Se designa de este modo a un área de almacenamiento de una computadora que contiene datos e instrucciones.

**MEMORIA CACHÉ**

Es un tipo de memoria especial de alta velocidad que se utiliza como memoria intermedia o buffer entre la memoria central y la CPU. Su objetivo principal es reducir los tiempos de acceso.

**MICROPROCESADOR**

Es un componente electrónico de una computadora, también llamado circuito



integrado o chip, que contiene las partes fundamentales de una computadora y los circuitos necesarios para que la computadora lleve a cabo sus cálculos.

### **MIME**

Multipurpose Internet Mail Extension (**Extensiones Multipropósito del Correo Internet**). Sistema que permite integrar dentro de un mensaje de correo electrónico ficheros binarios (imágenes, sonido, programas ejecutables, etc.).

Los tipos Mime son para dar formato a mensajes no-ASCII, de forma que pudieran ser enviados por Internet e interpretados correctamente por los programas de correo locales. Los tipos MIME especifican los tipos de datos, como por ejemplo texto, imagen, audio, etc., que los archivos contienen. MIME adjunta a cada fichero un archivo de cabecera donde se indica el tipo y el subtipo del contenido de los datos del mismo.

### **MÓDULO**

Término referido a componentes de un programa o sistema que se pueden identificar por separado y a los que es posible dirigirse también separadamente. Es equivalente al término segmento.

### **MULTIMEDIA**

Tratamiento informático avanzado de las tecnologías más recientes de sonido e imagen que engloba e integra funciones como la animación gráfica, la manipulación y digitalización de imágenes y sonido.

### **MULTIPROCESO**

Sistema en el que se utilizan varios procesadores funcionando simultáneamente y compartiendo tanto las memorias centrales como las auxiliares y los periféricos.

### **MPS**

Una directiva etiquetada con el estado "MPM" pertenece a un módulo de Multiprocesamiento.

## **N**

### **NSF**

**National Science Foundation -- NSF (Fundación Nacional para la Ciencia)** Organismo gubernamental norteamericano cuyo objetivo es promover el avance de la ciencia. La NSF financia a investigadores científicos, proyectos científicos e infraestructuras que mejoran la calidad de la investigación científica. La red NSFNET, financiada también por la NSF, fue en el pasado una parte esencial de las comunicaciones académicas y de investigación.

**O****OSI**

Open Systems Interconnection. Interconexión de Sistemas Abiertos. Modelo de referencia de interconexión de sistemas abiertos propuesto por la ISO.

Divide las tareas de la red en siete niveles.

**P****PACKET**

(Paquete) Cantidad mínima de datos que se transmite en una red o entre dispositivos. Tiene una estructura y longitud distinta según el protocolo al que pertenezca. También llamado TRAMA.

**PAP**

Password Authentication Protocol. Protocolo de Autenticación de Password. Protocolo que permite al sistema verificar la identidad del otro punto de la conexión mediante password.

**PARTICIÓN DE DISCO**

Es una zona de un disco duro formada por áreas de memoria consecutivas; y separada de forma lógica de otras áreas que también lo forman.

**PETICION**

Una petición de un cliente a un servidor ha de incluir el método que se aplica al recurso, el identificador del recurso y la versión del protocolo que usa para realizar la petición.

**PUERTO**

(Port) Punto de conexión de un ordenador que permite enlazarlo a otros dispositivos tales como el módem. También se denomina así a la cifra que, en un URL, aparece detrás del signo << .>>, después del nombre del dominio.

**PUT**

Actualiza información sobre un objeto del servidor. Es similar a POST, pero en este caso, la información enviada al servidor debe ser almacenada en la URL que acompaña al comando. Así se puede actualizar el contenido de un documento.

**POST**

Envío de formulario.

**POSIX**

Estándar universal que define cómo debe ser un sistema operativo de "tipo



UNIX" y que especifica una serie de normas para operación de las aplicaciones que se ejecutan en éste sistema operativo.

### **PROCESADOR**

La parte central de un sistema computador que proporciona y controla las funciones aritméticas, lógicas y de transferencias requeridas para comparar, mover, calcular y, de cualquier manera, manipular y procesar datos.

### **PROTOCOLO**

Los protocolos son reglas y procedimientos para la comunicación. Se aplican las reglas del protocolo al entorno informático, cuando dos equipos están conectados en red, las reglas y procedimientos técnicos que dictan su comunicación e interacción se denominan protocolos. El que hace posible Internet es el TCP/IP (Transmission Control Protocol / Internet Protocol).

## **R**

### **RAM**

Random Access Memory. Memoria de Acceso Aleatorio. Varios son los tipos de memoria que se usa en las computadoras. La más conocida son las RAM. Se les llama así porque es posible dirigirse directamente a la célula donde se encuentra almacenada la información. Su principal característica es que la información se almacena en ellas provisoriamente, pudiendo ser grabadas una y otra vez, al igual que un cassette de sonido. La memoria RAM se puede comparar a un escritorio, donde se colocan los papeles con que se va a trabajar. Mientras más grande el escritorio más papeles soporta simultáneamente para ser procesados.

### **RAID**

Son las siglas en inglés de Redundant Array of Inexpensive Disks, Arreglo Redundante de discos baratos, aunque a veces ni tan redundante, ni tan barato.

Existen varios niveles RAID 1, 2, 3, 4 y 5 dependiendo de la necesidad. Esta tecnología permite que en vez de haber un solo disco de gran capacidad en equipos servidores y exponernos a que pueda fallar, tenemos varios discos que actúan como espejo de forma que si uno falla automáticamente se van volcando los datos en otro y así mantenemos la disponibilidad del servicio hasta que la unidad dañada sea sustituida.

### **REDES**

Entornos complejos que incluyen múltiples medios, protocolos e interconexiones a otras redes externas.



## RPM

Los paquetes rpm son archivos que llevan incluidos dentro de ellos todos los ficheros que componen un determinado programa. Internamente están comprimidos, pero nosotros sólo debemos pensar en ellos en términos de Instalación, Actualización, Borrado y Consultas. Dentro del rpm van los ficheros del programa a instalar, su descripción, a que directorios van a ir instalados, scripts de auto-configuración en algunos casos, etc.

## S

### SISTEMAS OPERATIVOS

Un sistema Operativo (SO) es en sí mismo un programa de computadora. Sin embargo, es un programa muy especial, quizá el más complejo e importante en una computadora. El SO despierta a la computadora y hace que reconozca a la CPU, la memoria, el teclado, el sistema de vídeo y las unidades de disco. Además, proporciona la facilidad para que los usuarios se comuniquen con la computadora y sirve de plataforma a partir de la cual se corran programas de aplicación.

El sistema operativo es el gestor y organizador de todas las actividades que realiza la computadora. Marca las pautas según las cuales se intercambia información entre la memoria central y la externa, y determina las operaciones elementales que puede realizar el procesador. El sistema operativo, debe ser cargado en la memoria central antes que ninguna otra información.

### SCRIPT:

Grupo de caracteres que ha sido programado para automatizar en Internet una tarea que se realiza de forma muy habitual, como por ejemplo una conexión a Internet.

### SHELL

Es un procedimiento mediante el cual se puede acceder temporalmente al sistema operativo desde el interior de un programa. En Windows es una ventana de aplicación especial que permite lanzar otras aplicaciones.

### SOFTWARE

Esta palabra inglesa que significa "cosa suave", tiene dos significados: (a) uno amplio, de "procedimientos lógicos, para la cooperación armónica de un grupo de personas y máquinas, persiguiendo un objetivo común"; (b) el otro restringido, de "programas de computadora", o conjunto de instrucciones, que se pone en la memoria de una computadora para dirigir



sus

operaciones.

Es un conjunto de instrucciones que cargadas en el hardware de una computadora hacen que esta pueda funcionar y realizar tareas. Puede traducirse en castellano como "partes blandas" y es el término contrario a HARDWARE, "partes duras".

### **SOFTWARE LIBRE**

(Freeware) Aplicaciones informáticas que pueden ser libremente copiadas, distribuidas, estudiadas y modificadas por el usuario, según los parámetros establecidos por su creador.

### **SCRIPTS**

Conjunto de instrucciones que se ejecutan como una macro.

### **SSI**

Es un mecanismo que permite incluir en los documentos accedidos información dinámica del servidor como: la fecha actual, el contenido de un fichero, el tamaño de un fichero, etc. Además provee una poderosa interfaz para la ejecución de programas CGI y de *shell scripts*.

### **SSL**

Secure Socket Layer -- SSL (Capa de Conexión Segura) Protocolo creado por Netscape con el fin de posibilitar la transmisión cifrada y segura de información a través de la red. Se encuentra en la capa de transporte para TCP.

## **T**

### **TCP/IP**

Protocolo de Control de Transmisión / Protocolo Internet (TCP/IP), es hoy el estándar de factor para las comunicaciones de Internetwork y sirve como el protocolo de transporte para Internet, permitiendo que millones de computadoras se comuniquen a nivel mundial.

### **TELNET**

Protocolo y aplicaciones que permiten conexión como terminal remota a una computadora anfitriona, en una localización remota.

## **U**

### **URL**

Universal Resource Locator. Localizador Homogéneo de Recursos. Nombre genérico de la dirección en Internet, Indica al usuario dónde localizar un archivo HTML determinado, en la Web.

**UTILIDAD**

Programa que desempeña una tarea específica de uso general. Forma parte del software del sistema.

**W****WEB**

Site. Sitio en el World Wide Web. Conjunto de páginas Web que forman una unidad de presentación, como una revista o libro. Un sitio está formado por una colección de páginas Web WWW (World Wide Web) Servidor de información, desarrollado en el CERN (Laboratorio Europeo de Física de Partículas), buscando construir un sistema distribuido hipermedia e hipertexto. También llamado WEB y W3. Existen gran cantidad de clientes WWW para diferentes plataformas.



# XI. anexos



## ANEXO 1

### DIRECTIVAS DE CONFIGURACIÓN EN EL FICHERO HTTP.CONF

Para comprender mejor el archivo de configuración de HHTTP.CONF, es necesario conocer cada una de las directivas que este utiliza y la manera como estas funcionan, por eso les detallamos cada una de las directivas a continuación.

#### Manual oficial de referencia de Red Hat Linux Apache

##### 1. ServerTokens

**Sintaxis:** ServerTokens Major|Minor|Min[imal]|Prod[uctOnly]|OS|Full

Defecto: ServerTokens Full

Contexto : server config

Esta directiva delimita qué información muestra el servidor sobre los componentes que está ejecutando.

Un valor posible y aceptable sería:

ServerTokens OS

que mostraría el sistema operativo sobre el que se ejecuta el servidor apache. Si no le importa la información que puede presentar al exterior puede omitir o comentar esta línea.

Si quiere ajustar más la información los valores posibles son:

ServerTokens Minimal|ProductOnly|OS|Full

Como ejemplo podemos ver:

ServerTokens Prod[uctOnly] Server envía (por ejemplo): Server: Apache

ServerTokens Min[imal] Server envía (por ejemplo): Server: Apache/1.3.0

ServerTokens OS Server envía (por ejemplo): Server: Apache/1.3.0 (Unix)

ServerTokens Full (o sin especificar) Server envía (por ejemplo): Server:

Apache/1.3.0 (Unix) PHP/3.0 MyMod/1.2

**Full:** muestra todo.

**OS:** muestra la versión del sistema operativo en la que esta corriendo Apache.

**Min:** muestra solo la versión de apache.



**Prod:** muestra el nombre del servidor solamente.

## 2. ServerRoot

**Sintaxis:** ServerRoot <directorio>

El valor por defecto es: ServerRoot "/etc/httpd"

El comando ServerRoot es el directorio principal donde se encuentran todos los ficheros del servidor. Tanto el servidor seguro como el no seguro utilizan un comando ServerRoot del /etc/httpd.

Permite indicar el directorio raíz del servidor. A partir de aquí se buscarán el resto de los ficheros de configuración, el directorio de *logs*, los módulos, etc. Todo fichero especificado de forma relativa (su *path* no comienza con "/" ) se buscará a partir de este directorio.

Salvo que se indiquen rutas absolutas como se indicaba con anterioridad. Es importante no añadir una barra al final del nombre del último directorio.

## 3. PidFile

**Sintaxis:** PidFile <fichero>

Especifica el fichero donde se almacena el identificador del proceso servidor inicial. El comando PidFile nombra el fichero en el que el servidor graba su ID de proceso (pid). El Red Hat Linux Advanced Server está configurado para grabar su pid en /var/run/httpd.pid.

Si el nombre de fichero no es absoluto entonces se asume que esta concierne al ServerRoot.

## 4. Timeout

**Sintaxis:** TimeOut <segundos>

Por defecto: Timeout 300

El comando Timeout define, en segundos, el tiempo que el servidor esperará para recibir y enviar peticiones durante la comunicación. Específicamente, el comando Timeout define cuánto esperará el servidor para recibir peticiones GET, cuánto esperará para recibir paquetes TCP en una petición POST o PUT y cuánto esperará entre una ACK y otro respondiendo a paquetes TCP. El comando Timeout está ajustado a 300 segundos, que es el tiempo apropiado para la mayoría de las situaciones.

Timeout indica el número de segundos antes de que se cancele una conexión por falta de respuesta.



## 5. KeepAlive

**Sintaxis:** KeepAlive On/Off  
Por defecto está habilitada.

El comando KeepAlive determina si el servidor permitirá varias peticiones de conexión a la vez (es decir, conexiones continuas). KeepAlive puede usarse para impedir que un cliente consuma muchos recursos del servidor. El comando KeepAlive se establece en off por defecto, lo que significa que no se permiten varias conexiones a la vez. Puede definirse en **on** para que sí sean posibles.

En la mayoría de los casos, deseará que Keepalive esté desactivado ya que Apache sólo puede tener un número limitado de procesos hijos y estos procesos dependerán de otros y esperarán a que se establezca una conexión futura desde el mismo cliente.

Si activa Keepalives, también deberá establecer KeepAliveTimeout en un valor bajo y controlar el registro de errores (error\_log) de los servidores para estar advertido si no dispone de suficientes procesos hijos para gestionar las peticiones.

Estas conexiones permiten a través de una misma conexión TCP realizar varios pedidos HTTP. De esta forma se mejora el tiempo de respuesta del servidor. Sólo es posible si el *browser* soporta esta característica (Netscape y Microsoft Internet Explorer la soportan).

## 6. MaxKeepAliveRequests

Por defecto es 100.

Esta directiva establece el número máximo de peticiones permitidas por cada conexión que se produzca a la vez. El Grupo Apache recomienda un valor alto, lo que mejoraría el rendimiento. El valor predeterminado del comando MaxKeepAliveRequests es de 100 que debería bastar en la mayoría de los casos. Si se indica 0 el número será ilimitado.

## 7. KeepAliveTimeout

**Sintaxis:** KeepAliveTimeOut <cantidad>  
Por defecto es 15 segundos.

La directiva KeepAliveTimeout establece el número de segundos que el servidor esperará a la siguiente petición del mismo cliente a través de la misma conexión (conexión persistente), tras haber dado servicio a una petición, antes de cerrar la conexión. Una vez recibida la petición, aplica la directiva Timeout en su lugar.



## 8. IfModule

Por defecto no está compilado.

<IfModule> y </IfModule> envuelven a directivas que son condicionales. Las directivas contenidas dentro de IfModule son procesadas si se cumple una de las dos condiciones. Las directivas son procesadas si el módulo contenido en la etiqueta <IfModule> está cargado en el servidor Apache. O, si una "!" (exclamación) aparece antes del nombre; las directivas son procesadas sólo si el módulo en la etiqueta <IfModule> *no* está compilado.

El fichero `mod_mime_magic.c` está incluido en IfModule. El módulo `mod_mime_magic` puede compararse al comando UNIX `file`, que examina los primeros bytes de un fichero, y usa "números mágicos" y otros trucos para decidir el tipo MIME del fichero.

Si el módulo `mod_mime_magic` está compilado en Apache, estas etiquetas IfModule le dicen al módulo `mod_mime_magic` module donde está el fichero de los trucos: `share/magic` en este caso.

El módulo `mod_mime_magic` module no está compilado por defecto. Si se quiere usar, tendrá que saber cómo añadir módulos al servidor.

## 9. StartServers

**Sintaxis:** `StartServers <cantidad>`

Por defecto tiene 8.

`StartServers` establece cuántos procesos hijos serán creados al arrancar. Ya que el servidor Web crea y elimina dinámicamente servidores según el tráfico, no se necesitará cambiar este parámetro. El servidor por defecto está configurado para arrancar ocho procesos al arrancar.

## 10. MinSpareServers y MaxSpareServers

Por defecto el valor de `MinSpareServers` es 5.

Por defecto el valor de `MaxSpareServers` es 20.

`MinSpareServers`. Especifica la cantidad mínima.

`MaxSpareServers`. Especifica la cantidad máxima de procesos servidores.

El servidor Web Apache se adapta dinámicamente a la carga percibida manteniendo un número apropiado de servidores libres basado en el tráfico. El servidor comprueba el número de servidores que esperan peticiones y elimina algunos si el número es más alto que `MaxSpareServers` o crea algunos si el número de servidores es menor que `MinSpareServers`.

El valor predeterminado es suficiente en la mayoría de los casos.

El número de `MinSpareServers` no debería de ser elevado ya que creará una gran carga incluso cuando el tráfico fuese bajo.



## 11. MaxClients

**Sintaxis:** MaxClients <cantidad>

Ejemplo: MaxClients 200.

Por defecto es 150.

Permite indicar la cantidad máxima de clientes conectados simultáneamente al servidor.

Debe mantener el comando MaxClients a un valor alto porque no se permitirán nuevas conexiones una vez que se alcance el número máximo de clientes simultáneamente conectados.

El valor por defecto es **150**, pero no puede superar el valor de **256** sin que se haya recompilado Apache. La principal razón de tener el parámetro MaxClients es evitar que un servidor errático vuelva inestable al sistema operativo.

## 12. MaxRequestsPerChild

**Sintaxis:** MaxRequestsPerChild cantidad

Ejemplo: MaxRequestsPerChild 200

Por defecto es 100.

El comando MaxRequestsPerChild establece el número máximo de peticiones que cada proceso hijo procesa antes de morir. La principal razón para tener el comando MaxRequestsPerChild es evitar que procesos de larga vida pierdan memoria.

Los valores son cero o cien:

Si se especifica 0 el número será ilimitado, haciendo que el proceso hijo sirva solicitudes siempre. pero permite que los procesos Apache consuman poco a poco gran cantidad de memoria cuando un script mod\_perl defectuoso, o incluso un módulo defectuoso de Apache, filtra memoria.

Poner límites a este número permite liberar la memoria asociada al proceso, en caso de que esta se acumule accidentalmente, además de disminuir el número de procesos activos, cuando la carga del servidor disminuye. Este valor no incluye la cantidad de los pedidos en una conexión persistente, los cuales se cuentan como uno solo.

## 13. MaxSpareThreads

El valor por defecto es 10.

La directiva MaxSpareThreads determina el número máximo de hilos parados; deje el valor por defecto tal y como está. En el modo threaded, que



es el modo por defecto, Apache mata los procesos hijos para controlar la cuenta de hilos mínimos y máximos.

#### 14. **MixSpareThreads**

Por defecto es 5.

La directiva `MinSpareThreads` determina el número mínimo de hilos parados (ociosos). Estos hilos sobrantes se utilizan para servir solicitudes y los kilos de sobra se crean para mantener el mínimo tamaño del grupo de hilos parados.

#### 15. **ThreadsPerChild**

**Sintaxis:** `ThreadsPerChild` number.

Esta directiva determina la cantidad de hilos que se crean por cada proceso hijo que el servidor permitirá utilizar. Este valor representa el número máximo de conexiones que el servidor puede manejar a la vez.

#### 16. **NumServers**

El directorio de `NumServers` determina el número de los procesos hijos vivos en el mismo tiempo. Este número debe ser bastante grande para manejar los pedidos del sitio, debe ser de tipo entero. Para aumentar este valor más allá del valor de 8, es necesario cambiar el valor del `ServerLimit` directivo y parar y recomenzar el servidor.

#### 17. **Listen**

**Sintaxis:** `Listen` <[dirección IP:]puerto>

Ejemplos: `Listen 8080`  
`Listen 10.11.12.13:80`  
`Listen 10.11.12.14:800`

Las interfaces de red se indican mediante la dirección IP asociada. Por defecto, Apache escucha a través de todas las interfaces de red, pero solamente por el puerto que se especifica con la directiva `Port`.

Establece los puertos en los que Red Hat Linux Advanced Server acepta las peticiones entrantes. Red Hat Linux Advanced Server está configurado para escuchar en el puerto 80 para comunicaciones no seguras y (en máquinas virtuales que define el servidor seguro) en el puerto 443 para comunicaciones seguras.

Para puertos por debajo de 1024, el comando `httpd` deberá ser ejecutado como `root`.

Para el puerto 1024 y superiores, el comando `httpd` puede ser ejecutado como si fuera un usuario cualquiera.



El comando Listen también se puede usar para especificar direcciones IP específicas en las cuales aceptará conexiones el servidor.

## 18. Include

**Sintaxis:** Include <fichero>

**Ejemplo:** Include "conf/especial.conf"

Este directorio permite la inclusión de otros archivos de la configuración dentro de los archivos de la configuración del servidor.

(fnmatch())

Los caracteres del comodín del Rind style (fnmatch()) se pueden utilizar para incluir varios archivos inmediatamente, en orden alfabético. Además, si incluyen los puntos a un directorio, más bien que a un archivo, Apache leerá todos los archivos en ese directorio y cualquier subdirectorio. Pero incluir directorios enteros no se recomienda, porque es fácil dejar accidentalmente ficheros temporales en un directorio que pueda hacer el httpd fallar.

La trayectoria del archivo especificada puede ser un camino absoluto, o puede estar concerniente al directorio de ServerRoot. Ejemplos:

Incluya /usr/local/apache2/conf/SSL.conf

Incluya /usr/local/apache2/conf/vhosts/\*.conf

O, abastecimiento de las trayectorias concerniente a su directorio de ServerRoot:

Incluya conf/SSL.conf

Incluya conf/vhosts/\*.conf

## 19. LoadModule

**Sintaxis:** LoadModule <nombre> <binario>

Ejemplos:

LoadModule dir\_module modules/mod\_dir.so

LoadModule cgi\_module modules/mod\_cgi.so

LoadModule alias\_module modules/mod\_alias.so

Se usa para cargar módulos Dynamic Shared Object(DSO). Nótese que el orden de los módulos es importante.

Permite cargar un módulo del Apache. Los módulos incorporan nuevas funcionalidades al servidor. Se indican a través de un nombre genérico y el fichero donde se encuentra definido el binario correspondiente. Los módulos que trae la distribución del Apache para Red Hat se agrupan en el directorio /usr/lib/apache/ enlazado simbólicamente por /etc/httpd/modules.

## 20. User

**Sintaxis:** User <usuario> | <UID>

Por defecto es el 80.



La directiva User establece el userid usado por el servidor para responder a peticiones. El valor de User determina el acceso al servidor. Cualquier fichero

al que no pueda acceder este usuario será también inaccesible al visitante de la web. El comando predeterminado para User es *apache*.

User debería sólo tener privilegios de tal manera que sólo se pueda acceder a ficheros que se supone que todo el mundo puede ver. El comando User también es dueño del cualquier proceso CGI que arranque el servidor. Al comando User no se le debería permitir ejecutar ningún código que no esté pensado para responder peticiones HTTP.

**Nota** Por razones de seguridad, Apache rechazará la ejecución como User root. Usar root para User creará grandes problemas de seguridad para el servidor de Web.

El proceso httpd padre se ejecuta como root durante operaciones normales, pero pasa al usuario apache inmediatamente. El servidor debe arrancar como root porque necesita un puerto por debajo de 1024 (El puerto predeterminado para comunicaciones seguras es 443; el puerto por defecto para comunicaciones no seguras es el puerto 80). Los puertos por debajo de 1024 están reservados para el sistema, así que sólo se pueden usar si se es root. Una vez que el servidor se ha conectado al puerto, pasa el proceso a User antes de aceptar peticiones.

## 21. Group

**Sintaxis:** Group <grupo> | <GID>

El comando Group es similar a User. Group establece el grupo en que el servidor responde a las peticiones. El valor predeterminado del comando Group es *apache*.

## 22. ServerAdmin

**Sintaxis:** ServerAdmin <dirección de correo>

Ejemplo: ServerAdmin pepito@mail.disaic.cu

ServerAdmin debería ser la dirección de correo del administrador del Red Hat Linux Advanced Server. Esta dirección de correo aparecerá en los mensajes de error generados por el servidor para páginas web, de tal manera que los usuarios pueden comunicar errores enviando correo al administrador.

El comando ServerAdmin ya se encuentra por defecto en la dirección *root@localhost*.

Una forma buena y típica de configurar ServerAdmin es situarlo en la dirección *webmaster@your\_domain.com*.



Después cree un alias del webmaster para la persona responsable del servidor en `/etc/aliases`.

Finalmente, ejecute `/usr/bin/newaliases` para añadir el nuevo alias.

### 23. ServerName

**Sintaxis:** `ServerName <nombre>`

Ejemplo: `ServerName www.deltha.disaic.cu`

El comando `ServerName` puede usarse para establecer el nombre de la máquina del servidor diferente al nombre real de máquina como por ejemplo, usar `www.your_domain.com` aunque el nombre real del servidor sea `foo.your_domain.com`. Nótese que `ServerName` debe ser un nombre "Domain Name Service" (DNS) válido que se tenga derecho a usar (no basta con inventar uno).

Si se especifica `ServerName`, hay que asegurarse de incluir la pareja nombre-dirección IP en el fichero `/etc/hosts`.

### 24. UseCanonicalName

**Sintaxis:** `UseCanonicalName On|Off|dns`

**Ejemplo:** `UseCanonicalName dns`

`UseCanonicalName` ya aparece en `on`.

Indica que se debe hacer cuando el Apache necesita construir un URL que se referencia a si mismo, por ejemplo cuando se accede a un directorio sin el carácter `/` al final.

Si se pone esta directiva en:

**On** entonces se emplearán los valores de las directivas `ServerName` y `Port` para construir el nombre canónico del servidor. Permite que los URLs contengan sus propias referencias. Cuando el servidor se refiere a si mismo en respuesta a peticiones de clientes.

**Off** hará que se tome el nombre y el puerto provistos por el cliente.

**Dns** hará una resolución inversa de la dirección IP de la interfaz de red a la cual se conectó el cliente del nombre de dominio.

### 25. DocumentRoot

**Sintaxis:** `DocumentRoot <directorio>`

**Ejemplo:** `DocumentRoot "/usr/local/web"`

`DocumentRoot` es el directorio que contiene la mayoría de los ficheros HTML que se entregarán en respuesta a peticiones.



Indica el directorio a partir del cual se localizarán todos los documentos provistos por el servidor. El directorio para servidores seguros y no seguros por defecto es ***/var/www/html***.

Ejemplo:

Se realiza el pedido ***http://www.deltha.disaic.cu/manual/index.html*** el documento devuelto será ***/var/www/html/manual/index.html***.

## 26. Directory

**Sintaxis:** <Directory directorio> ... </Directory>

Esta directiva es muy importante pues se emplea para encerrar un conjunto de directivas a aplicar a uno o varios directorios (y sus subdirectorios) en el servidor. Los directorios se pueden indicar de la forma acostumbrada o utilizando expresiones regulares si se coloca como primer argumento de la directiva el caracter "~". Por ejemplo para referirse a todos los directorios que comiencen con la palabra "prueba" y a continuación de dos a cuatro caracteres "a", "b" o "c" se puede emplear la declaración:

```
<Directory ~ /var/www/html/prueba[abc]{2,4}>
```

```
...
```

```
</Directory>
```

Las directivas aplicables a un directorio son las siguientes:

- Options
- AllowOverride
- Allow y Deny

Las etiquetas <Directory /path/to/directory> y </Directory> se usan para agrupar directivas de configuración que sólo se aplican a ese directorio y sus subdirectorios.

Por defecto, se aplican parámetros muy restrictivos al directorio raíz, utilizando Options. Con esta configuración, cualquier directorio del sistema que necesite valores más permisivos ha de ser configurado explícitamente.

## 27. Location

**Sintaxis:** <Location URL> ... </Location>

**Ejemplo:**

```
<Location /server-status>  
    SetHandler server-status  
    Order deny,allow  
    Deny from all  
    Allow from localhost  
</Location>
```



Se emplea para aplicar un conjunto de directivas teniendo como base el URL accedido. Se evalúa después de las directivas `<Directory>` y `<Files>`, y de leer los ficheros de control de acceso (directiva `AccessControlFiles`). Puede ser útil cuando se combina con la directiva `SetHandler` que permite indicar un manipulador para los documentos solicitados.

### **Ejemplos de manipuladores son:**

**`server-status`:** este devuelve una página descriptiva del estado del servidor (se puede ampliar la información mostrada habilitando la directiva `ExtendedStatus`).

**`server-info`:** este devuelve una página con información acerca de la configuración actual del Apache.

Las etiquetas `<Location>` y `</Location>` permiten controlar el acceso específico a cada URL.

La utilización de las etiquetas `Location`, permite al comando `DocumentRoot` (referido a `/`) tener parámetros menos rígidos para que el servidor sirva las peticiones HTTP.

- a) El primer uso de `Location` es configurar `Options` y proporcionar guías extra de configuración para `DocumentRoot`. Estas directivas de configuración, que se encuentran dentro de las etiquetas `<Location "/">` y `</Location>`, son necesarias para permitir el acceso a documentos en `DocumentRoot`.
- b) El siguiente uso de `Location` es en las etiquetas `IfModule mod_perl.c`. Estas directivas de configuración funcionan si el DSO `mod_perl.so` está cargado.

La etiqueta `Location` nombra el directorio `/var/www/perl` (un Alias para `/perl`) como el directorio desde el cual se sirven scripts de Perl. Si se pide un documento con un URL que contenga `/perl` en el recorrido, el servidor buscará en `/var/www/perl/` el script de Perl apropiado.

Los comentarios de otras opciones de `<Location>` están en `httpd.conf`. Si se quiere activar su funcionalidad, se necesitará anular el comentario de la sección apropiada de las directivas.



## 28. Options

**Sintaxis:** Options [+|-]<opción> [+|-]<opción> ...

Permite indicar que características del servidor están disponibles cuando se sirve el directorio correspondiente al cliente. Estas características son:

**FollowSymLinks:** indica que se pueden seguir los enlaces simbólicos presentes en el directorio.

1. **ExecCGI:** indica que se pueden ejecutar los *scripts* de tipo CGI.
2. **Includes:** indica que se permite el mecanismo SSI (*Server Side Include*).
3. **Indexes:** indica que si el cliente solicita un directorio donde no exista ninguno de los ficheros especificados en *DirectoryIndex* se le devuelva una lista con formato del contenido del directorio.
4. **SymLinksIfOwnerMatch:** indica que se sigan los enlaces simbólicos solo si el fichero o directorio enlazado posee el mismo dueño que el enlace.
5. **IncludeNoExec:** es similar a *Includes* con la diferencia de que no se permiten los comandos SSI *include* y *exec* para programas CGI.
6. **MultiViews:** indica que se "negocie" entre el servidor y el cliente el documento a mostrar cuando existen varias posibilidades. Por ejemplo, si se solicita el documento */images/map* y existe el directorio *images* con la opción habilitada pero no existe */images/map*, entonces el servidor busca todos los documentos con nombre *map.\** y "negocia" con el cliente cual de estos, de existir alguno, retorna al cliente.
7. **All:** incluye todas las opciones con excepción de *MultiViews*. Este es el valor por defecto.

Cuando varias directivas *Option* son aplicables a un directorio se toma la más específica, o sea no se mezclan salvo que estas se indiquen precedidas por un + o un -, en cuyo caso se añade la opción o se elimina respectivamente. Por ejemplo en la declaración siguiente:

```
<Directory /var/www/html/documents>
    Options Indexes FollowSymLinks
</Directory>
<Directory /var/www/html/documents/linux>
    Options Includes ExecCGI
</Directory>
```



Las opciones aplicadas al directorio `/var/www/html/documents/linux` son solamente `Includes` y `ExecCGI`. En cambio si se pone:

```
<Directory /var/www/html/documents>
  Options Indexes FollowSymLinks
</Directory>
<Directory /var/www/html/documents/linux>
  Options +Includes +ExecCGI -FollowSymLinks
</Directory>
```

Las opciones serían `Indexes`, `Includes` y `ExecCGI`

Por ejemplo, en los parámetros restrictivos especificados para el directorio raíz, el comando `Options` sólo permite `FollowSymLinks`. No hay características permitidas, salvo que el servidor pueda seguir enlaces simbólicos en el directorio raíz.

## 29. AllowOverride

**Sintaxis:** `AllowOverride <valores>`

`AllowOverride` establece qué directivas `Options` puede obviar un fichero `.htaccess`. Por defecto, tanto el directorio raíz como `DocumentRoot` están configurados para no permitir la prevaencia de `.htaccess`.

Indica que directivas presentes en un fichero especificado en `AccessFileName` sobrescriben el valor de las encontradas en el fichero de configuración principal (`httpd.conf`). Los valores que toma `AllowOverride` pueden ser:

1. **All y None:** indican que se sobrescriban todas las directivas o que no se sobrescriba ninguna.
2. **Options:** permite que se sobrescriba la directiva `Options`.
3. **Indexes:** permite que se sobrescriban todas las directivas relacionadas con la indexación de los directorios (Ej. `AddDescription`, `AddIcon`, `AddIconByEncoding`, `AddIconByType`, `DefaultIcon`, `DirectoryIndex`, `FancyIndexing`, `HeaderName`, `IndexIgnore`, `IndexOptions`, `ReadmeName`, etc.)
4. **Limit:** permite la sobrescritura de directivas que controlan el acceso por máquina (`Order`, `Allow` y `Deny`).



- 5. AuthConfig:** permite el uso de directivas que controlan el acceso por autenticación (Ej. AuthDBMGroupFile, AuthDBMUserFile, AuthGroupFile, AuthName, AuthType, AuthUserFile, Require, etc.)
- 6. FileInfo:** permite la sobrescritura de las directivas que controlan el tipo de documentos (Ej. AddEncoding, AddLanguage, AddType, DefaultType, ErrorDocument, LanguagePriority, etc.)

### 30. Order

**Sintaxis:** Allow from <host> <host> ...

**Ejemplos:**

```
Allow from .disaic.cu 200.198.100.  
Deny from 192.168.0.0/16 200.198.100.34  
Allow from all
```

Order simplemente controla el orden en que allow y deny se evalúan. Sus posibles valores son allow, deny o deny, allow. Siempre se evalúan ambas directivas o sea, no se hace corto circuito cuando se encuentra una expresión que se corresponda con el *host* cliente. El servidor está configurado para evaluar Allow antes que deny para el directorio DocumentRoot.

### 31. Allow

Allow especifica qué *hosts* puede acceder a un directorio dado. El *hosts* puede ser all, un nombre de dominio, una dirección IP, una dirección IP parcial, un par de máscaras de red, etc. El directorio DocumentRoot está configurado para permitir peticiones de all (cualquiera).

### 32. Deny

Deny funciona como allow, pero especifica a quién se niega el acceso. DocumentRoot no está configurado para rechazar peticiones de nadie.

### 33. LocationMatch

LocationMatch proporciona control de acceso según petición basándose en una expresión regular. Limitando el alcance de los directorios incluidos de URL.

haría que cualquier petición que comenzara y terminara por /, es decir el directorio principal del servidor se le desactivara la generación de índices y generara una página de error.

Por ejemplo:

```
<LocationMatch "/(extra|special)/data">
```

podría mostrar el URLs que contiene la subsecuencia extra/datos/ o especial/datos.



### 34. ErrorDocument

**Sintaxis:** ErrorDocument <código> <documento>

**Ejemplos:**

```
ErrorDocument 403 /errors/missing.html
```

```
ErrorDocument 500 "El servidor se fundió
```

```
# el caracter " indica el comienzo de un texto.
```

Por defecto, en caso de error, el servidor muestra un mensaje de error (generalmente críptico) para el cliente. En vez de usar esta opción ya predeterminada, puede usarse ErrorDocument para devolver un mensaje de error personalizado o redireccionar al cliente a un URL local o remoto. ErrorDocument simplemente asocia un código de respuesta HTTP con un mensaje o un URL que se devolverá al cliente.

Se emplea para redefinir el documento que devuelve el Apache ante cada código de error. Por defecto para cada tipo de error se devuelve una página *Web* integrada al programa servidor, pero mediante el empleo de esta directiva se puede lograr devolver un texto, un documento local o en otro servidor a través de su URL, o ejecutar un programa que genere dinámicamente la respuesta. Algunos códigos de error son:

**500:** Reservado para errores internos del servidor. Puede producirse debido a una mala configuración del servicio o a errores en la ejecución de un programa CGI.

**403:** Se genera cuando el cliente no tiene acceso al documento solicitado.

**404:** Se genera cuando no existe el documento solicitado.

### 35. UserDir

**Sintaxis:** UserDir <directorio>

UserDir es el nombre del subdirectorio dentro del directorio de cada usuario dónde estarán los ficheros HTML que serán servidos por el servidor de Web.

Accedidos mediante el servicio de la forma *http://host/~login*.

Por defecto, el subdirectorio es ***public\_html***.

Ejemplo:

El servidor podría recibir la siguiente petición:

```
http://your_domain/~username/foo.html
```

El servidor buscaría el fichero:

```
/home/username/public_html/foo.html
```

```
Ejm:/home/Maria/public_html/foo.html
```



Donde `/home/username` es el directorio del usuario (nótese que la ruta predeterminada a los directorios de los usuarios puede variar entre sistemas).

Hay que asegurarse que los permisos de los directorios de usuario sean correctos. El valor de los permisos deben ser de 0711. Los bits de lectura (r) y ejecución (x) deben estar activados en el directorio del usuario `public_html` (0755 valdrá). El valor de los permisos con que se servirán los ficheros desde `public_html` debe ser 0644 por lo menos.

### 36. Limit

**Sintaxis:** `<Limit métodos> ... </Limit>`

Permite el uso de los directorios que controlan el acceso del anfitrión (Allow, Deny and Order) (permitir, Niegue y ordenar).

El propósito de esta directiva es restringir el acceso de acuerdo a los métodos del protocolo HTTP que invoque el cliente. Estos pueden ser: GET, POST, PUT, DELETE, CONNECT, OPTIONS, TRACE, PATCH, PROPFIND, PROPPATCH, MKCOL, COPY, MOVE, LOCK, y UNLOCK. Similar a *Limit* existe la directiva *LimitExcept* que funciona de forma opuesta.

### 37. DirectoryIndex

**Sintaxis:** `DirectoryIndex <ficheros>`

`DirectoryIndex` es la página por defecto que entrega el servidor cuando hay una petición de índice de un directorio especificado con una barra (/) al final del nombre del directorio.

#### **Ejemplo:**

Cuando un usuario pide la página <http://liukar.isi.unanleon.edu.ni>, recibe la página `DirectoryIndex` si existe, o un listado generado por el servidor.

El valor por defecto para `DirectoryIndex` es **`index.html`, `index.htm`, `index.shtml` e `index.cgi`**. El servidor intentará encontrar cualquiera de estos cuatro, y entregará el primero que encuentre. Si no encuentra ninguno y si `Options Indexes` se encuentra en el directorio, el servidor generará un listado, en formato HTML, de los subdirectorios y ficheros del directorio.

### 38. AccessFileName

**Sintaxis:** `DirectoryIndex <ficheros>`

`AccessFileName` denomina el fichero que el servidor utilizará para controlar el acceso en cada directorio determinado.



Estos ficheros pueden pertenecer al directorio correspondiente o a los directorios que forman su *path*. Siempre se utilizará el primero de los especificados que aparezca en cada uno de los directorios de la jerarquía. Por defecto, el servidor utilizará **.htaccess**, si existe, para controlar el acceso en cada directorio.

De esta forma para acceder al documento `/var/www/html/maria/index.html` se leerán en orden los siguientes ficheros: `/.htaccess`, `/var/.htaccess`, `/var/www/.htaccess`, `/var/www/html/.htaccess` y `/var/www/html/manual/.htaccess` y de los que existan se tomarán las restricciones.

### 39. Files

**Sintaxis:** `<Files fichero> ... </Files>`

El comando Files controla el acceso a cualquier fichero que empiece con `.ht`. Estas directivas niegan acceso a todo tipo de fichero `.htaccess` (u otros ficheros que empiecen por `.ht`) por razones de seguridad.

Permite especificar directivas a aplicar de acuerdo a los ficheros accedidos. Es similar a `<Directory>` y `<Location>`, aunque a diferencia de estas se puede utilizar dentro de los ficheros especificados en la directiva `AccessControlFiles`. También se permite la utilización de expresiones regulares o de la directiva equivalente `<FilesMatch>`.

La directiva `<Files>` limita el alcance de las directivas incluidas por el nombre del archivo. Las directivas dadas dentro de esta sección serán aplicadas a cualquier objeto con un `basename` (el último componente del nombre del archivo) correspondencia del nombre del archivo especificado.

`<Files>` las secciones son tratadas en el orden en que ellos aparecen en el archivo de configuración, después de que las secciones `<Directory>` y los archivos `.htaccess` son leídos, pero ante `<Location>` secciones. Note que `<Files>` puede ser anidado dentro de secciones `<de Directorio>` para restringir la parte del sistema de ficheros al que ellos se aplican.

### 40. TypesConfig

**Sintaxis:** `TypesConfig <fichero>`

`TypesConfig` denomina el fichero que establece la lista predeterminada de mapeado de tipos MIME (*Multipurpose Internet Mail Extensions*) (extensiones de ficheros a tipos de contenido).



El fichero por defecto para Red Hat de la directiva `TypesConfig` es `/etc/mime.types`. En vez de modificar el `/etc/mime.types`, se recomienda añadir mapeados de tipos MIMEs con `AddType`.

#### 41. **DefaultType**

**Sintaxis:** `DefaultType <tipo MIME>`

`DefaultType` establece el contenido por defecto que el servidor utilizará para documentos cuyos tipos MIME no puedan ser determinados. El servidor predispone el texto para cualquier fichero con un tipo de contenido indeterminado. Por defecto, para Red Hat es **text/plain**, o sea texto plano.

#### 42. **MimeMagicFile**

Permite la determinación de TIPO MIME basada en contenido de archivo que usan el archivo especificado mágico.

La directiva `MimeMagicFile` puede ser usada para permitir este módulo, el archivo de falta es distribuido en `conf/magic`. Los caminos no arraigados son con relación al `ServerRoot`. Los anfitriones virtuales usarán el mismo archivo que el servidor principal a menos que un ajuste más específico sea usado, en cuyo caso el ajuste más específico anula el archivo del servidor principal. Ejemplo: `MimeMagicFile conf/magic`.

#### 43. **HostnameLookups**

**Sintaxis:** `HostnameLookups On|Off|Double`

`HostnameLookups` puede aparecer en `on` o en `off`. Indica si se resuelve o no la dirección IP del cliente a nombre de dominio,

**On:** Si el servidor permite la directiva `HostnameLookups` (poniéndolo en `on`), el servidor resolverá automáticamente la dirección IP de cada conexión que pida un documento del servidor. Resolver la dirección IP implica que el servidor hará una o más conexiones al DNS para averiguar qué nombre de máquina se corresponde con una dirección IP.

**Off:** Generalmente, debería dejarse `HostnameLookups` en `off` porque las peticiones de DNS añaden carga al servidor y pueden ralentizarlo. Si el servidor tiene carga, los efectos de `HostnameLookups` serán considerables.

**Double** indica que cuando se haga la consulta inversa se haga también la resolución del nombre devuelto debiendo coincidir una de las direcciones retornadas con la dirección original.



HostnameLookups influye también en Internet en general. Cada conexión individual provoca una sobrecarga en el servidor. Por ello, para beneficio del servidor y de Internet en general, debería dejarse HostnameLookups en off, para evitar tener que resolver el nombre de cada cliente siempre que haga un pedido.

Para ver los nombres de host de los ficheros .log, puede ejecutar una de las muchas herramientas de análisis de ficheros .log que realizan consultas de DNS de una forma más eficaz y ejecuciones en grupo para cambiar los ficheros .log.

#### 44. ErrorLog

**Sintaxis:** ErrorLog <fichero> | <tubería> | syslog[:facilidad]

ErrorLog nombra el fichero donde se guardan los errores del servidor. Como viene indicado, el fichero de error del servidor es /var/log/httpd/error\_log. El log de errores es un buen sitio para ver si el servidor genera errores y no se desconoce como ha sucedido.

Si el argumento de la directiva comienza con un caracter “|” expresará que las trazas serán filtradas mediante una tubería a través del comando siguiente. Por último se puede expresar que se use el servicio **syslog** indicando una facilidad propia del servicio y que por defecto es **local7**.

#### 45. LogLevel

**Sintaxis:** LogLevel <prioridad>

Indica a partir de que prioridad se guardan las trazas (establece cómo serán de abundantes los logs de error). Las prioridades son las mismas definidas en el servicio *syslog*: debug, info, notice, warn, error, crit, alert y emerg. Siempre una prioridad incluye a las superiores (de menor a mayor). Por defecto para Red Hat, se indica la prioridad warn (nivel medio).

#### 46. LogFormat

**Sintaxis:** LogFormat <formato> [nombre]

LogFormat pone el formato para los mensajes en el log de acceso. El comando LogFormat real que se use dependerá de los parámetros definidos en la directiva CustomLog.

Permite definir formatos para las trazas (los errores) utilizando algunas macros. A cada formato se le puede asignar un nombre para asociarlo luego a un fichero de trazas determinado. De no colocarse un nombre se referirá al fichero que especifica la directiva **TransferLog**.



### **Las Macros son:**

- 1) **%h**: el nombre o dirección IP del *host* cliente.
- 2) **%s**: el estado del pedido original del cliente. Para referirse al estado del último pedido se emplea %>s.
- 3) **%t**: la fecha del pedido.
- 4) **%b**: la cantidad de *bytes* del pedido excluyendo el encabezamiento HTTP.
- 5) **%m**: el método HTTP de la solicitud (GET, POST, PUT, HEAD, etc.).
- 6) **%r**: la primera línea del pedido.
- 7) **%l**: el *login* del usuario cliente si existe un servicio que permita determinarlo.
- 8) **%u**: el *login* del usuario una vez autenticado para realizar el pedido.

### **47. CustomLog**

**Sintaxis:** CustomLog <fichero> <formato> | <nombre>

Permite asociar un formato a un fichero de trazas, definiéndolo en la directiva o utilizando uno definido mediante LogFormat.

CustomLog identifica el log y el formato de log. La configuración por defecto de CustomLog del Red Hat Linux Advanced Server, define el log en el que se guardan los accesos al servidor /var/log/httpd/access\_log. Habrá que saber la localización de este fichero si se quieren generar estadísticas de rendimiento del servidor.

CustomLog pone el formato combinado para el fichero.

### **48. ServerSignature**

**Sintaxis:** ServerSignature On|Off|Email

El comando ServerSignature añade una línea que contiene la versión del servidor Apache y el ServerName de la máquina a los documentos generados por el servidor.

Ejem: mensajes de error devueltos a clientes.

**on** : ya aparece por defecto en ServerSignature. Se coloca una línea con el nombre del servidor y el puerto por el que atendió el pedido.

**Off** : para no añadir nada.



**EMail** añadirá una etiqueta HTML mailto: ServerAdmin a la línea de firma. incluye además la dirección especificada en la directiva ServerAdmin.

#### 49. Alias

**Sintaxis:** Alias <URL> <fichero>|<directorio>

El comando Alias permite que hayan directorios fuera del directorio raíz DocumentRoot (recursos fuera) a los que puede acceder el servidor. Cualquier URL que termine en un alias será automáticamente traducido por el recorrido del alias. Por defecto, ya existe un alias configurado. El servidor puede acceder al directorio icons pero el directorio no está en DocumentRoot. icons, un alias, está en **/var/www/icons/**, y no en /var/www/html/icons/.

#### 50. DavLockDB

La posición del DAV cierra la base de datos.

Use la directiva DavLockDB para especificar el camino lleno a la base de datos de cerradura, excluyendo una extensión. Si el camino no es absoluto, será tomado con relación a ServerRoot. La realización de mod\_dav\_fs usa una base de datos SDBM para rastrear cerraduras de usuario.

Ejemplo: DavLockDB var/DavLock

El directorio que contiene el archivo de base de datos de cerradura debe ser escribible por el Usuario y Grupo bajo el cual apache corre. Para razones de seguridad, usted debería crear un directorio para este fin más bien que cambiar los permisos sobre un directorio existente. En el susodicho ejemplo, el apache creará archivos en el directorio var/ bajo el ServerRoot con el nombre del archivo bajo DavLock y nombre de extensión elegido por el servidor.

#### 51. IndexOptions

**Sintaxis:** IndexOptions [+|-]<opción> [+|-]<opción> ...

El comando IndexOptions controla la apariencia de la lista de los índices para los directorios accedidos. generados por el servidor, al añadir iconos y texto descriptivo, etc.

Si IndexOptions aparece como en, el servidor podrá generar el listado de un directorio al recibir una petición HTTP como la que sigue:

`http://your_domain/this_directory/`



Primero el servidor busca en el directorio un fichero de los de la lista de DirectoryIndex (p.ej., index.html). Si el servidor no encuentra ninguno de los ficheros, genera un listado del directorio en HTML. Se puede modificar la apariencia del listado utilizando ciertas directivas en `httpd.conf`, entre las que se encuentra `IndexOptions`.

**FancyIndexing**: es la configuración predeterminada activa, esta genera una salida elegante con iconos, descripciones y posibilidades de ordenación. Existen otras múltiples opciones.

Si se activa `FancyIndexing`, al hacer click en la cabecera de las columnas del listado, el listado se ordena según esa columna. Otro click en la misma cabecera cambiará el orden de ascendente a descendente y viceversa. `FancyIndexing` también muestra distintos iconos para distintos ficheros, según la extensión. Si se usa la directiva `AddDescription` y se activa `FancyIndexing`, se añade una pequeña descripción para el fichero en el listado generado.

`IndexOptions` tiene otros parámetros que pueden activarse para controlar la apariencia de los listados. Los parámetros incluyen `IconHeight` e `IconWidth`, para hacer que el servidor incluya etiquetas `HEIGHT` y `WIDTH` para los iconos; el comando `IconsAreLinks`, hace que los iconos formen parte del enlace HTML junto con el nombre del fichero, y otros.

## 52. **AddIconByEncoding**

Esta directiva denomina qué iconos se mostrarán con los ficheros según su codificación MIME, en los listados de directorio. Por ejemplo, por defecto, el servidor muestra el icono `compressed.gif` junto a ficheros con codificación MIME `x-compress` y `x-gzip` en los listados de directorio.

## 53. **AddIconByType**

Esta directiva denomina qué iconos se mostrarán con los ficheros según su codificación MIME, en los listados del directorio. Por ejemplo, por defecto, el servidor muestra el icono `text.gif` junto a ficheros con tipo MIME "text" en los listados del directorio.

## 54. **AddIcon**

`AddIcon` dice al servidor qué icono mostrar en los listados del directorio para ciertos tipos de ficheros según la extensión. Por ejemplo, el servidor muestra el icono `binary.gif` para ficheros con extensiones `.bin` o `.exe`.



### **55. DefaultIcon**

El comando DefaultIcon nombra el icono de los listados del directorio que recibirán los ficheros sin un icono específico. El fichero de imagen predeterminado para esos ficheros unknown.gif es DefaultIcon.

### **56. ReadmeName**

La directiva ReadmeName determina el fichero (si existe dentro del directorio) que se adjuntará a los listados de los directorios. El servidor intentará primero incluirlo como documento HTML y luego como texto. El valor predeterminado de ReadmeName es README.

### **57. HeaderName**

La directiva HeaderName dicta el fichero (si existe dentro del directorio) que se antepondrá al comienzo de los listados de los directorios. Al igual que con ReadmeName, el servidor intentará incluirlo como documento HTML si es posible, o como texto.

### **58. IndexIgnore**

El comando IndexIgnore lista las extensiones, los nombres de los ficheros parciales, las expresiones regulares o los nombres completos. El servidor no incluirá los ficheros que encajen en estos patrones en los listados de directorios.

### **59. AddEncoding**

El comando AddEncoding dice qué extensiones especifican un tipo particular de codificación. AddEncoding se puede usar para decirle a los navegadores (no a todos) que descompriman ciertos ficheros mientras los descargan.

### **60. AddLanguage**

La directiva AddLanguage asocia extensiones a contenidos específicos de idiomas. Esta directiva es útil para la negociación de contenidos, cuando el servidor devuelve uno de entre varios documentos según las preferencias de idiomas del cliente.

### **61. LanguagePriority**

La directiva LanguagePriority permite dar la prioridad a ciertos ficheros en distintos idiomas, que entrarán en vigor si el cliente no especifica la preferencia de idioma.



## 62. ForceLanguagePriority

La acción a tomar si un solo documento no es encontrado aceptable.

La directiva ForceLanguagePriority usa el LanguagePriority dado para satisfacer negociación donde el servidor no podía devolver por otra parte un documento de correspondencia solo.

Los ForceLanguagePriority Prefer usa LanguagePriority para servir un resultado válido, más bien que devolver un resultado de HTTP 300 (MÚLTIPLES OPCIONES) cuando hay varias opciones igualmente válidas.

## 63. AddDefaultCharset

Juego de caracteres por defecto a ser añadido para una respuesta sin un juego de caracteres explícito.

Esta directiva especifica el nombre del juego de caracteres que será añadido a cualquier respuesta que no tiene ningún parámetro sobre el contenido teclear los jefes HTTP. Este anulará cualquier juego de caracteres especificado en el cuerpo del documento vía una etiqueta de META. Un ajuste de AddDefaultCharset incapacita esta funcionalidad. El AddDefaultCharset Sobre permite la falta interna del apache charset de iso-8859-1 como requerido por la directiva. Usted puede especificar también un suplente charset para ser usado. Por ejemplo: AddDefaultCharset utf-8

## 64. AddCharset

Traza un mapa de las extensiones del nombre del archivo dadas al contenido especificado charset.

La directiva AddCharset traza un mapa de las extensiones del nombre del archivo dadas al contenido especificado charset. charset es la PANTOMIMA charset el parámetro de nombres del archivo que contienen la extensión. Esta correlación es añadida a alguno ya vigente, anulando cualquier correlación que ya existe para la misma extensión.

Ejemplo:

```
AddLanguage ja .ja
AddCharset EUC-JP .euc
AddCharset ISO-2022-JP .jis
AddCharset SHIFT_JIS .sjis
```

Entonces el documento xxxx.ja.jis será tratado como siendo un documento japonés cuyo charset es iSO-2022-JP (como va al documento xxxx.jis.ja). La directiva AddCharset es útil para ambos para informar al cliente sobre la codificación de carácter del documento de modo que el documento pueda ser interpretado y mostrado apropiadamente, y para contener negociación, donde el servidor vuelve uno de varios documentos basados en la preferencia charset del cliente.



El argumento de extensión es insensible a mayúsculas y minúsculas, y puede ser especificado con o sin un punto principal.

### **65. AddType**

Use la directiva `AddType` para definir parejas de tipos MIME y sus extensiones. Por ejemplo, si usa el PHP4, el servidor está usando `AddType` para que se reconozcan ficheros con extensiones PHP (`.php4`, `.php3`, `.phtml`, `.php`) como tipos MIME PHP.

La siguiente línea `AddType` permite al servidor reconocer las extensiones `.shtml` (para la inclusión en el servidor):

```
AddType text/html .shtml
```

Se necesitará la línea de arriba dentro de las etiquetas de máquina virtual para cuando se permita la inclusión desde el servidor.

### **66. AddHandler**

La directiva `AddHandler` mapea y amplía gestores específicos. Por ejemplo, el gestor `cgi-script` puede usarse para hacer que la extensión `.cgi` automáticamente sea manejada como un script CGI. Esto funciona, incluso para ficheros fuera de `ScriptAlias`, si se siguen las instrucciones dadas.

Hay una línea `AddHandler CGI` en `httpd.conf` como la siguiente:

```
AddHandler cgi-script .cgi
```

### **67. AddOutputFilter**

Las extensiones del nombre del archivo de mapas a los filtros que tratarán respuestas del servidor.

La directiva `AddOutputFilter` traza un mapa de la extensión del nombre del archivo a los filtros que tratarán respuestas del servidor antes de que ellos sean enviados al cliente. Este está además de cualquier filtro definido en otra parte, incluso `SetOutputFilter` y directiva `AddOutputFilterByType`. Esta correlación es combinada sobre alguno ya vigente, anulando cualquier correlación que ya existe para la misma extensión.

Por ejemplo, la configuración siguiente tratará todos los archivos `.shtml` para el lado del servidor, incluye y comprimirá entonces la salida que usa `mod_deflate`.

```
AddOutputFilter INCLUDES;DEFLATE shtml.
```

Si más de un filtro es especificado, ellos deben ser separados por punto y coma en la orden en la cual ellos deberían tratar el contenido. Tanto el filtro como los argumentos de extensión son insensibles a mayúsculas y minúsculas, y la extensión puede ser especificada con o sin un punto principal.



## **68. BrowserMatch**

La directiva BrowserMatch permite al servidor definir variables de entorno y/o tomar acciones según sea el campo de cabecera User-Agent, que identifica al cliente. Por defecto, el servidor usa BrowserMatch para denegar la conexión a navegadores con problemas conocidos y para desactivar "keepalives" y vaciados de cabecera de HTTP para navegadores con problemas de esas características.

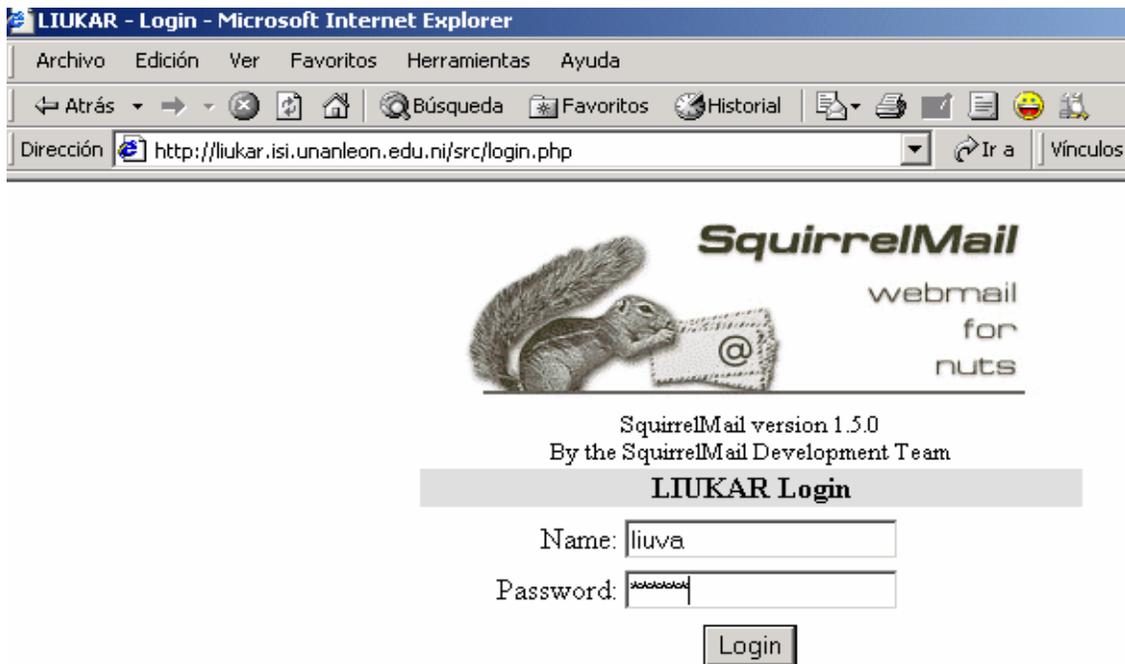


## ANEXO 2

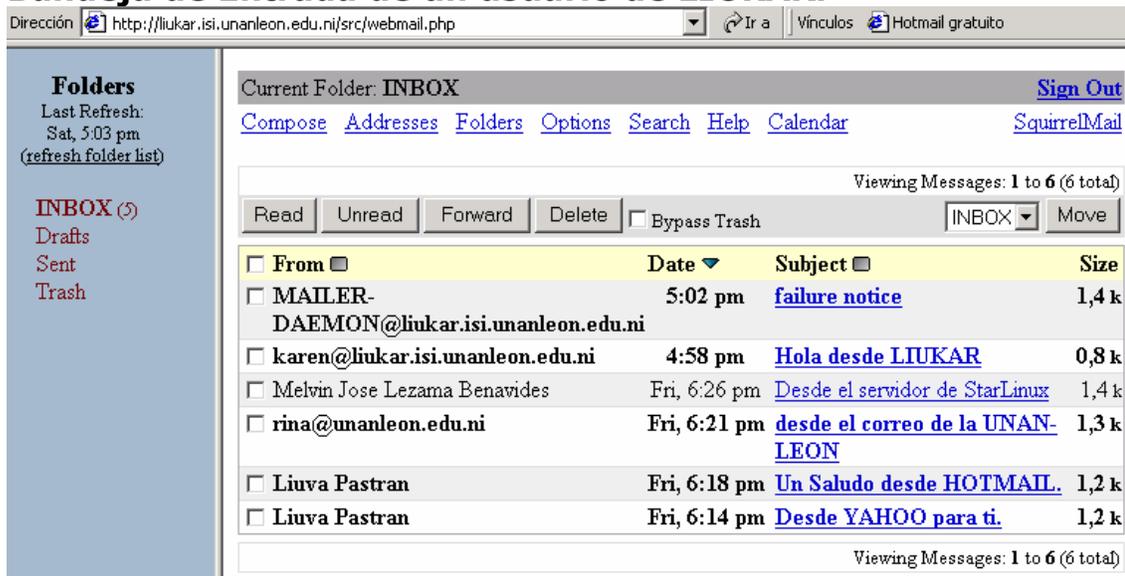
### INTERFAZ DEL SERVIDOR DE CORREO LIUKAR

Estas son algunas imágenes que muestran a LIUKAR en funcionamiento, enviando y recibiendo mensajes tanto de él mismo como desde otros servidores.

#### 1. Página de presentación del correo.



#### 2. Bandeja de Entrada de un usuario de LIUKAR.





### 3. Contenido de un mensaje de un usuario no existente.

Dirección <http://liukar.isi.unanleon.edu.ni/src/webmail.php> Ir a Vinculos Hotmail gratuito

[Previous | [Next](#)] [Delete & Prev | [Delete & Next](#)] [[Message List](#)]

[Reply](#) | [Reply All](#) | [Forward](#) | [Forward as Attachment](#) | [Delete \(Bypass Trash\)](#) Move to: INBOX Move

**Subject:** failure notice  
**From:** MAILER-DAEMON@liukar.isi.unanleon.edu.ni  
**Date:** Sat, February 18, 2006 5:02 pm  
**To:** liuva@liukar.isi.unanleon.edu.ni  
**Priority:** Normal  
**Options:** [View Full Header](#) | [View Printable Version](#)

Hi. This is the qmail-send program at liukar.isi.unanleon.edu.ni. I'm afraid I wasn't able to deliver your message to the following addresses This is a permanent error; I've given up. Sorry it didn't work out.

<[noexiste@seguroregresa.com](mailto:noexiste@seguroregresa.com)>:  
 Sorry, I couldn't find any host named seguroregresa.com. (#5.1.2)

--- Below this line is a copy of the message.

Return-Path: <[liuva@liukar.isi.unanleon.edu.ni](mailto:liuva@liukar.isi.unanleon.edu.ni)>  
 Received: (qmail 22274 invoked from network); 18 Feb 2006 23:02:07 -0000  
 Received: from unknown (HELO 172.16.4.244) (127.0.0.1)  
 Received: from unknown (HELO 172.16.4.244) (127.0.0.1)  
   by localhost with SMTP; 18 Feb 2006 23:02:07 -0000  
 Received: from 172.16.0.1 (proxying for unknown)  
   (SquirrelMail authenticated user liuva);  
   by 172.16.4.244 with HTTP;  
   Sat, 18 Feb 2006 17:02:07 -0600 (CST)  
 Message-ID: <[57748.172.16.0.1.1140303727.squirrel@172.16.0.1](mailto:57748.172.16.0.1.1140303727.squirrel@172.16.0.1)>  
 Date: Sat, 18 Feb 2006 17:02:07 -0600 (CST)  
 Subject: El destino no existe  
 From: [liuva@liukar.isi.unanleon.edu.ni](mailto:liuva@liukar.isi.unanleon.edu.ni)  
 To: [noexiste@seguroregresa.com](mailto:noexiste@seguroregresa.com)  
 User-Agent: SquirrelMail/1.5.0  
 MIME-Version: 1.0  
 Content-Type: text/plain; charset=iso-8859-1  
 Content-Transfer-Encoding: 8bit  
 X-Priority: 3  
 Importance: Normal

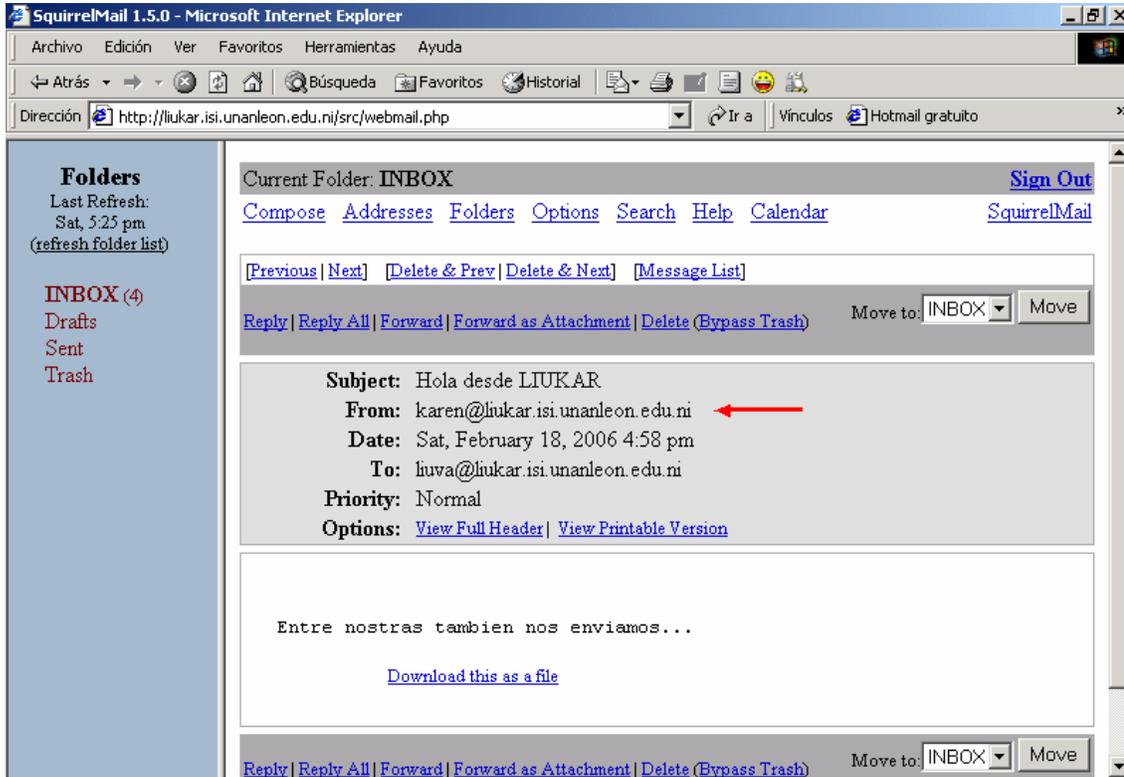
Va a informar al usuario que no existe dicho usuario.

[Download this as a file](#)

[Reply](#) | [Reply All](#) | [Forward](#) | [Forward as Attachment](#) | [Delete \(Bypass Trash\)](#) Move to: INBOX Move



### 4. Contenido de un mensaje de un usuario de LIUKAR.



### 5. Contenido de un mensaje de un usuario de StarLinux.





### 6. Contenido de un mensaje de un usuario de UNAN-LEON.

Dirección <http://liukar.isi.unanleon.edu.ni/src/webmail.php> Ir a Vinculos Hotmail gratuito

Current Folder: INBOX [Sign Out](#)  
[Compose](#) [Addresses](#) [Folders](#) [Options](#) [Search](#) [Help](#) [Calendar](#) [SquirrelMail](#)

[Previous](#) | [Next](#) | [Delete & Prev](#) | [Delete & Next](#) | [Message List](#)

[Reply](#) | [Reply All](#) | [Forward](#) | [Forward as Attachment](#) | [Delete \(Bypass Trash\)](#) Move to: INBOX

**Subject:** desde el correo de la UNAN-LEON  
**From:** rina@unanleon.edu.ni ←  
**Date:** Fri, February 17, 2006 6:21 pm  
**To:** luva@liukar.isi.unanleon.edu.ni  
**Priority:** Normal  
**Options:** [View Full Header](#) | [View Printable Version](#)

HOLA

### 7. Contenido de un mensaje de un usuario de HOTMAIL.

Dirección <http://liukar.isi.unanleon.edu.ni/src/webmail.php> Ir a Vinculos Hotmail gratuito

Current Folder: INBOX [Sign Out](#)  
[Compose](#) [Addresses](#) [Folders](#) [Options](#) [Search](#) [Help](#) [Calendar](#) [SquirrelMail](#)

[Previous](#) | [Next](#) | [Delete & Prev](#) | [Delete & Next](#) | [Message List](#)

[Reply](#) | [Reply All](#) | [Forward](#) | [Forward as Attachment](#) | [Delete \(Bypass Trash\)](#) Move to: INBOX

**Subject:** Un Saludo desde HOTMAIL.  
**From:** "Luva Pastran" <lipastran@hotmail.com> ←  
**Date:** Fri, February 17, 2006 6:18 pm  
**To:** luva@liukar.isi.unanleon.edu.ni  
**Priority:** Normal  
**Options:** [View Full Header](#) | [View Printable Version](#)

Probando la recepcion para el servidor de LIUKAR.

Las mejores tiendas, los precios mas bajos, entregas en todo el mundo,  
 YupiMSN Compras: <http://latam.msn.com/compras/>

[Download this as a file](#)



### 8. Contenido de un mensaje de un usuario de YAHOO.

Dirección <http://liukar.isi.unanleon.edu.ni/src/webmail.php> Ir a Vínculos Hotmail gratuito

Current Folder: INBOX [Sign Out](#)  
[Compose](#) [Addresses](#) [Folders](#) [Options](#) [Search](#) [Help](#) [Calendar](#) [SquirrelMail](#)

[\[Previous\]](#) [Next](#) | [\[Delete & Prev\]](#) [Delete & Next](#) | [\[Message List\]](#)

[Reply](#) | [Reply All](#) | [Forward](#) | [Forward as Attachment](#) | [Delete \(Bypass Trash\)](#) Move to: INBOX Move

**Subject:** Desde YAHOO para ti  
**From:** "Liuva Pastran" <lipastran@yahoo.com> ←  
**Date:** Fri, February 17, 2006 6:14 pm  
**To:** liuva@liukar.isi.unanleon.edu.ni  
**Priority:** Normal  
**Options:** [View Full Header](#) | [View Printable Version](#)

Hola

---

Correo Yahoo!  
Espacio para todos tus mensajes, antivirus y antispam ¡gratis!  
Regístrate ya - <http://correo.espanol.yahoo.com/>

### 9. BANDEJA DE ENVIADOS.

SquirrelMail 1.5.0 - Microsoft Internet Explorer

Archivo Edición Ver Favoritos Herramientas Ayuda

← Atrás → Búsqueda Favoritos Historial

Dirección <http://liukar.isi.unanleon.edu.ni/src/webmail.php> Ir a Vínculos

Current Folder: Sent [Sign Out](#)  
[Compose](#) [Addresses](#) [Folders](#) [Options](#) [Search](#) [Help](#) [Calendar](#) [SquirrelMail](#)

Viewing Messages: 1 to 6 (6 total)

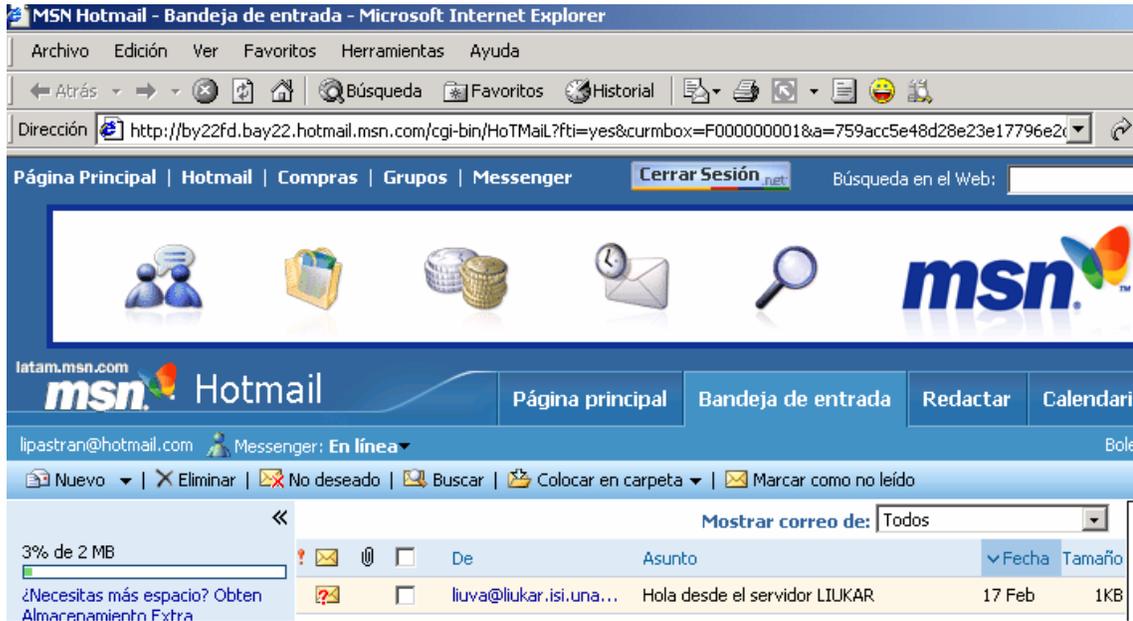
Read Unread Forward Delete  Bypass Trash INBOX Move

<input type="checkbox"/> To <input type="checkbox"/>	Date ▼	Subject <input type="checkbox"/>	Size
<input type="checkbox"/> rina@unanleon.edu.ni	10:35 am	<a href="#">Hola Profesora si funciona LIUKAR</a>	0,6 k
<input type="checkbox"/> karen@liukar.isi.unanleon.edu.ni	10:31 am	<a href="#">Entre nosotras nos enviamos</a>	0,6 k
<input type="checkbox"/> noexiste@seguroregresa.com	Sat, 5:02 pm	<a href="#">El destino no existe</a>	0,6 k
<input type="checkbox"/> mjlb@starlinux.net	Fri, 6:30 pm	<a href="#">Hola desde el servidor LIUKAR</a>	0,6 k
<input type="checkbox"/> lipastran@hotmail.com	Fri, 5:54 pm	<a href="#">Hola desde el servidor LIUKAR</a>	0,6 k
<input type="checkbox"/> lipastran@yahoo.com	Fri, 5:49 pm	<a href="#">mensaje desde LIUKAR</a>	0,5 k

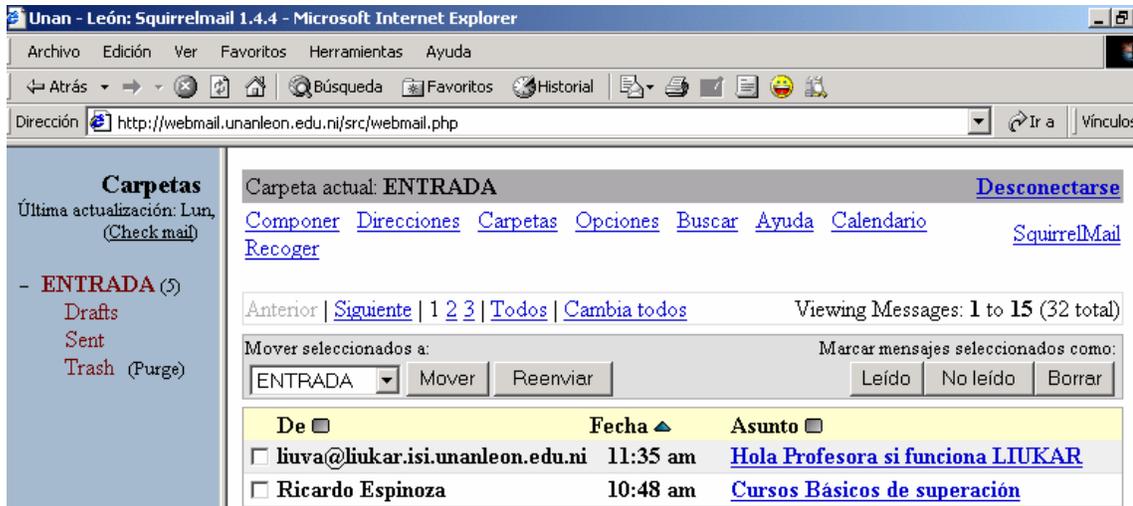
Viewing Messages: 1 to 6 (6 total)



### 10. Bandeja de entrada de **HOTMAIL**.



### 11. Bandeja de entrada de **UNAN-LEÓN**.





### 12. Bandeja de entrada de YAHOO.

**YAHOO! CORREO EN ESPAÑOL** Hola lipastran [Salir, Mi cuenta] [Inicio](#) - [Ayuda](#)

**Arma tu estrategia de juego! NO FALLES!!**  
Crea tu equipo en los torneos de fútbol virtual  
**COPA TOYOTA LIBERTADORES** YAHOO! Deportes [Haz click](#)

Correo | Contactos | Agenda | Bloc de notas | [Novedades](#) - [Actualizaciones](#) - [Opciones](#)

Revisar correo | Redactar |  | Buscar en mensajes | Buscar en la Web

**Carpetas** [Añadir - Modificar]

- Bandeja de entrada (3)
- Borrador
- Enviado
- Lotes de correo [Vaciar]
- Papelera [Vaciar]
- Mis carpetas [Ocultar]

**Bandeja de entrada** 23% de 1.0GB

Ver: Todos Mensajes 1-34 de 34Primero | Anterior | Siguiente | Último

Eliminar Spam Marcar Mover...

<input type="checkbox"/>	Remitente	Asunto:	Fecha	Tamaño
<input type="checkbox"/>	enero@liukar.isi.unanleon.edu.ni	1 sabado 7	sáb 07 ene 2006	8k
<input type="checkbox"/>	karen@liukar.isi.unanleon.edu.ni	archivo adjunto	vie 06 ene 2006	3k
<input type="checkbox"/>	karen@liukar.isi.unanleon.edu.ni	Re: viernes 6 ener 06	vie 06 ene 2006	2k

### 13. Contenido de un mensaje con archivo adjunto de yahoo.

[Modificar](#)

**Bandeja de entrada** (2)

- Borrador
- Enviado
- Lotes de correo [Vaciar]
- Papelera [Vaciar]
- Mis carpetas [Mostrar]

Eliminar Responder Reenviar Es spam Mover

Este mensaje no está marcado con una bandera. [ [Marcar mensaje](#) - [Presentación imprimible](#) ]  
[Marcar como no leído](#) ]

**Fecha:** Fri, 6 Jan 2006 12:00:15 -0600 (CST)

**Asunto:** archivo adjunto

**De:** karen@liukar.isi.unanleon.edu.ni [Añadir a Libreta de contactos](#)

**Para:** mateo@liukar.isi.unanleon.edu.ni

**CC:** karen@liukar.isi.unanleon.edu.ni, lipastran@yahoo.com

Prueba de que se envian archivos adjuntos desde el servidor LIUKAR

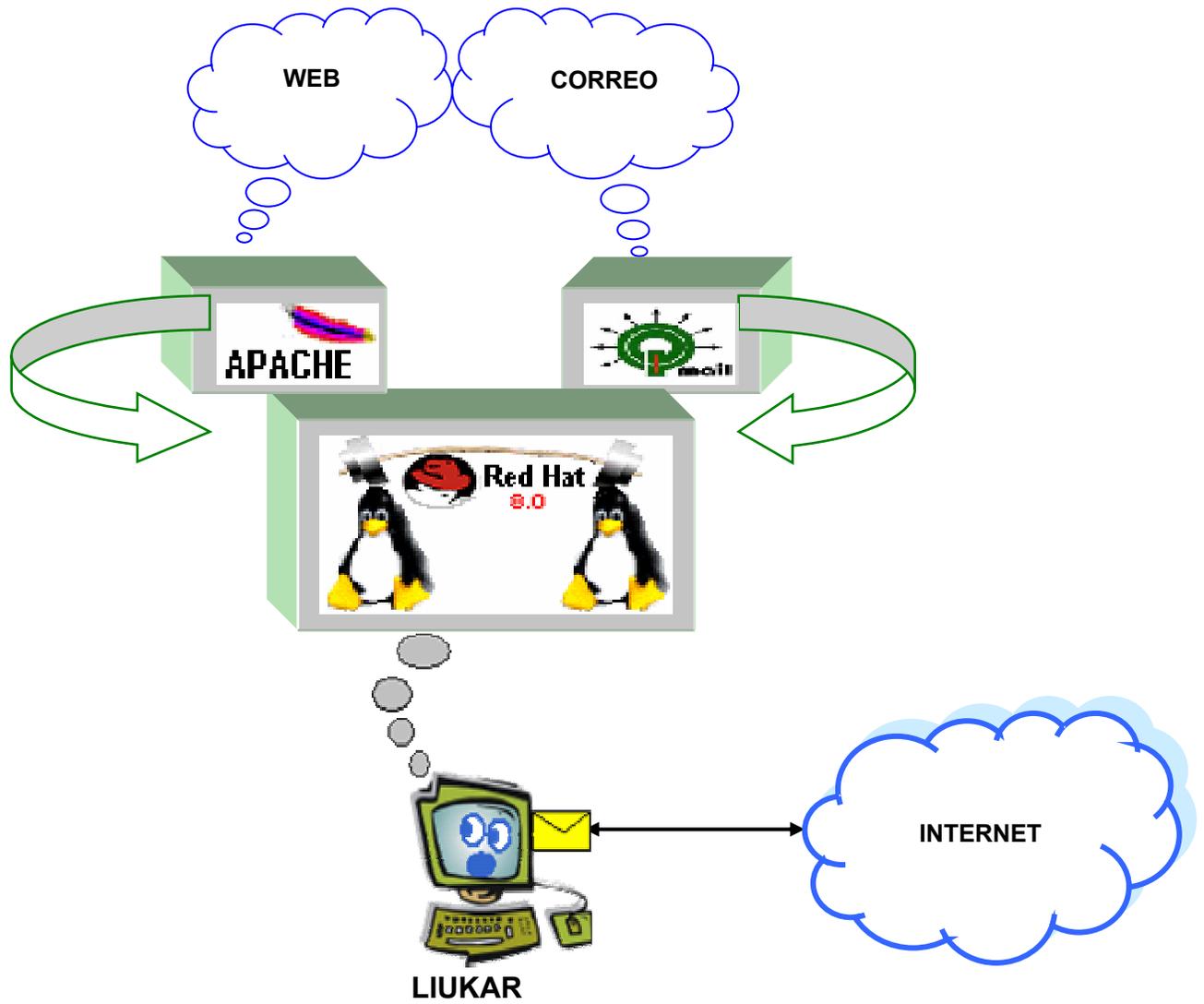
**Archivos adjuntos** Análisis antivirus ofrecido por: **AntiVirus**

**Archivos:**

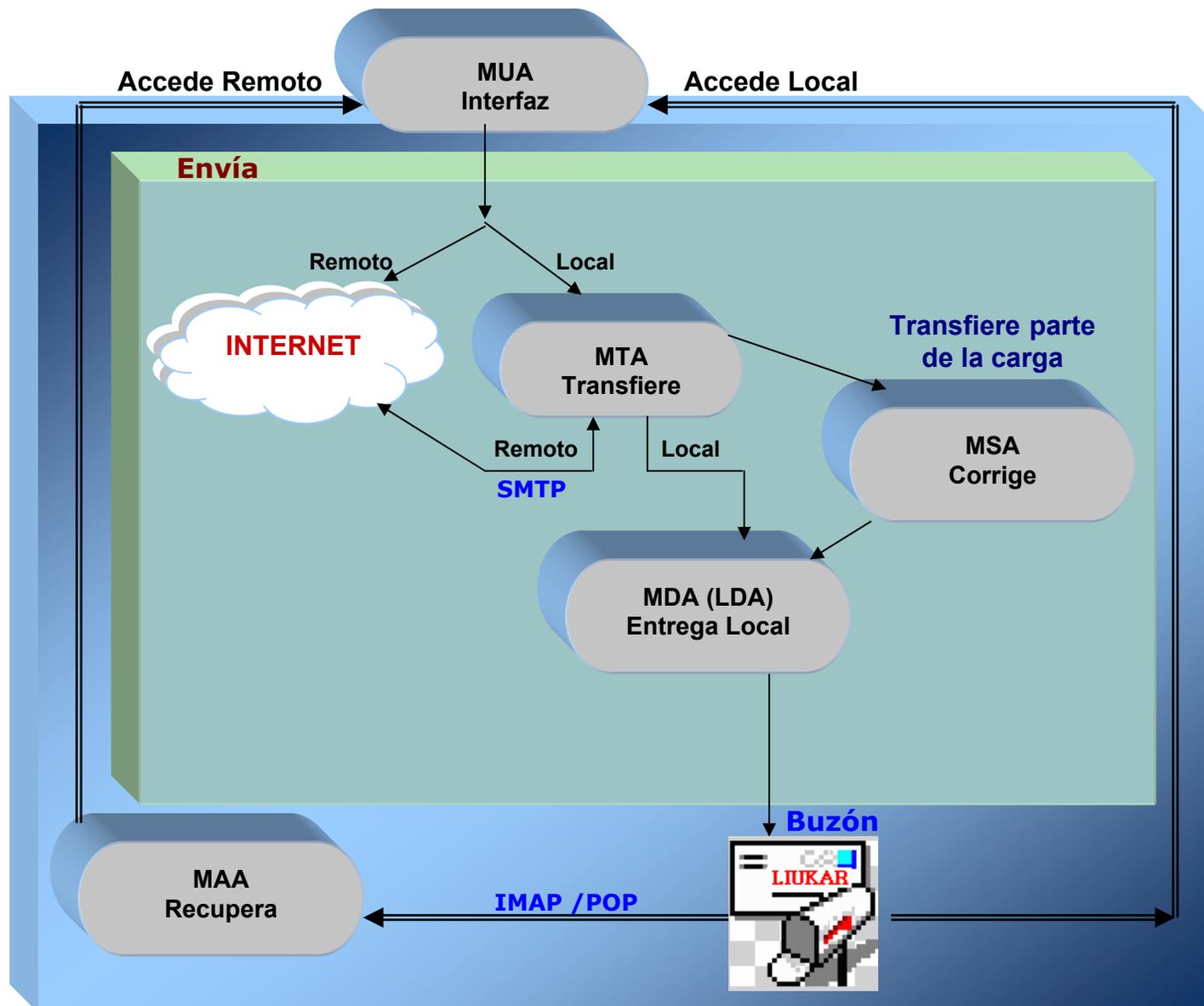
- Undefectoenlamujer** (1002) [Guardar en la computadora](#) - [Guardar en Yahoo! Mi maletín](#)



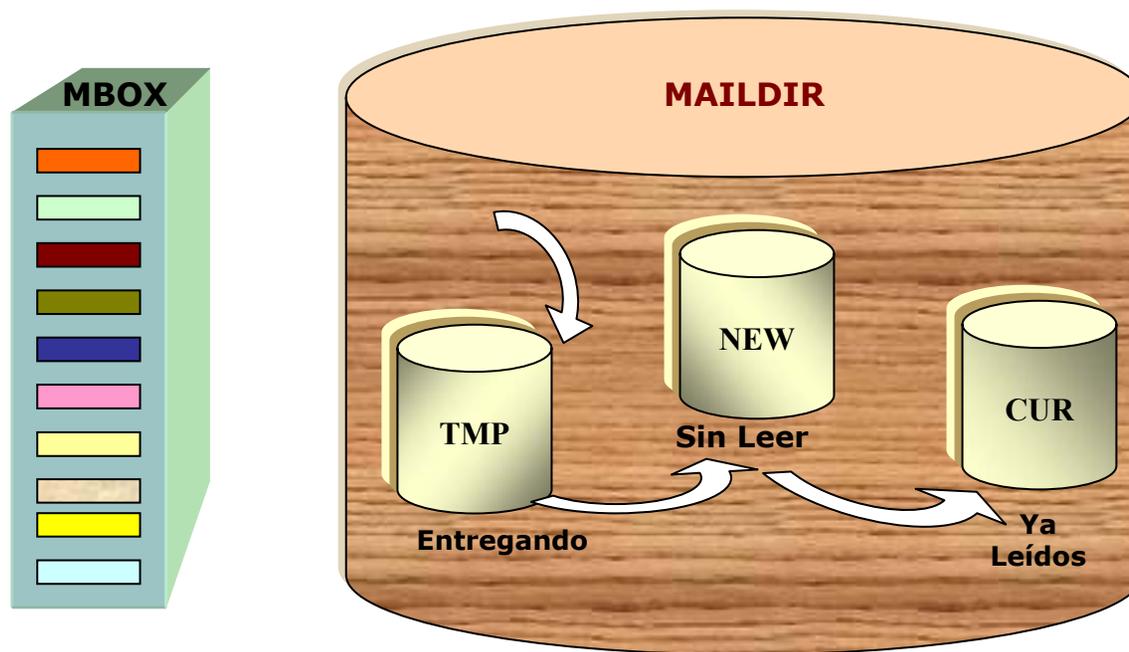
### 14. Descripción gráfica del Servidor Web y de Correo "LIUKAR"



15. Gráfica que muestra como trabajan Agentes y Protocolos del Correo.



## 16. Selección del Mecanismo de Almacenamiento de Mensajes.





## XI. BIBLIOGRAFÍA



### **Capítulo 18. Configuración del Servidor HTTP de Apache.**

[www.europe.redhat.com](http://www.europe.redhat.com)  
[www.httpd.apache.org/docs-2.0](http://www.httpd.apache.org/docs-2.0)



### **Manual Oficial de Referencia de Red Hat Linux.**

<http://www.tu-chemnitz.de/docs/lindocs/RH73/RH-DOCS/rhl-rg-es-7.3/apache-config.html>



### **Análisis de los Requerimientos Tecnológicos para la Implementación de Servidores Web Seguros.**

[www.monografias.com/trabajo12/rete.shtml](http://www.monografias.com/trabajo12/rete.shtml)



### **Definición Arquitectura Cliente – Servidor.**

<http://www.monografias.com/trabajos24/arquitectura-cliente-servidor/arquitectura-cliente-servidor.shtml>



### **Documentación de qmail.**

<http://www.es.qmail.org/documentacion>



### **Páginas de Descargas de Paquetes Utilizados.**

qmail: <http://cr.yip.to/software/qmail-1.03.tar.gz>  
ucspi-tcp: <http://cr.yip.to/ucspi-tcp/ucspi-tcp-0.88.tar.gz>  
Daemontools: <http://cr.yip.to/daemontools/daemontools-0.76.tar.gz>  
php-50.4.tar.gz <http://>  
squirrelmail-1.5.0.tar.gz <http://www.squirrelmail.org/>

**Páginas de Descargas de Paquetes Utilizados.**

qmail: <http://cr.yip.to/software/qmail-1.03.tar.gz>  
ucspi-tcp: <http://cr.yip.to/ucspi-tcp/ucspi-tcp-0.88.tar.gz>  
Daemontools: <http://cr.yip.to/daemontools/daemontools-0.76.tar.gz>  
php-50.4.tar.gz <http://>  
squirrelmail-1.5.0.tar.gz <http://www.squirrelmail.org/>  
daemontools-0.76.tar <http://cr.yip.to/daemontools/daemontools-0.76.tar.gz>  
dot-forward <ftp://koobera.math.uic.edu/www/software/dot-forward-0.71.tar.gz>

## Quotas

<http://www.linuxparatodos.net/geeklog/staticpages/index.php?page=04-disk-quota>  
SpamAssassin.(AntiSpam) <http://search.cpan.org>  
Clamav (Antivirus) <http://www.clamav.net>  
<http://www.osmosislatina.com/spamvirus/configuracionclamav.htm>  
<http://mingo.ath.cx/virus/>