Software Libre y Educación:

servicios de red, gestores de contenidos y seguridad

Servidor Web y Correo electrónico

José Ángel Bernal, Fernando Gordillo, Hugo Santander y Paco Villegas

31 de marzo de 2005

Índice general

1. Servidor Web Apache **5**
- 1.1. Servidor Web (apache). . . . 5
- 1.2. Instalación 7
  - 1.2.1. Red Hat/Fedora 7
  - 1.2.2. Guadalinex (Debian) 8
  - 1.2.3. ¡A navegar! 10
- 1.3. Configuración 12
  - 1.3.1. Document root 12
  - 1.3.2. Ficheros de configuración 13
- 1.4. /etc/http/httpd.conf 15
  - 1.4.1. Debian 19
- 1.5. Autenticación 21
- 1.6. Host Virtuales 24
- 1.7. Servidores Seguros 26
  - 1.7.1. Autenticación del cliente mediante certificados 32
- 1.8. Reescribir las URL 35
- 1.9. Loganalizadores 36
  - 1.9.1. webalizer 36
  - 1.9.2. awstats 38

2. Correo electrónico **43**
- 2.1. Introducción 43
  - 2.1.1. ¿Cuántos invitados tenemos para cenar? 45
  - 2.1.2. ¿Cómo se encamina el correo? 47
  - 2.1.3. Eso no es todo, aún hay más 48
- 2.2. Agentes de Transporte 51
  - 2.2.1. Postfix 51
  - 2.2.2. Sendmail 61
- 2.3. Agente de entrega: Fetchmail 68
  - 2.3.1. Configuración 69
- 2.4. Mozilla Mail y Ximian Evolution 72
  - 2.4.1. Mozilla Mail 72
  - 2.4.2. Agente de Usuario: Ximian Evolution 75
- 2.5. Luchemos contra el SPAM: amavisd-new y spamassassin 78
  - 2.5.1. Instalación de SpamAssassin 78
  - 2.5.2. Instalación de Amavisd-new 80
  - 2.5.3. Modificaciones en Postfix 82
- 2.6. Gestores de listas de correo: Mailman 82
  - 2.6.1. ¿Qué es una lista de correo? 82
  - 2.6.2. Mailman 83
- 2.7. Correo Web: SquirrelMail 89
  - 2.7.1. Instalación 89

2.7.2. Configuración 91

Capítulo 1

Servidor Web Apache

> Apache es un producto fantástico. Hace todo lo que se quiere que haga, y nada de lo que no se quiere. Es rápido, fiable y barato. ¿Qué más se podría pedir de una unidad de software?
>
> Apache puede ser todo esto porque es *open source*. (*Servidor Apache*, Rick Bowen & ken Coar)

1.1. Servidor Web (apache).

El servidor Apache es un servidor web HTTP de software libre que funciona en diversos sistemas operativos (Unix/Linux, Windows, MacOS, etc.). El objetivo del proyecto es proporcionar un servidor HTTP seguro, eficiente, extensible y que cumpla con los estándares[1].

Apache es el servidor web más usado en Internet desde Abril de 1996. Las últimas estadísticas, de febrero de 2005 proporcionadas por Netcraft Web Server Survey, muestran que más del 68 % de los sitios web de Internet utilizan Apache. Apache es un proyecto de la Apache Software Foundation. (www.apache.org).

Veamos algunas pinceladas de su historia. Poco después del nacimiento de la Web en el CERN, un grupo de personas del Centro Nacional de Actividades de Supercomputación (*National Center for Supercomputing Activities*, NCSA), de la Universidad de Illinois, creó un servidor web (HTTPd NCSA) que fue el más usado en la web hasta mediados de 1994.

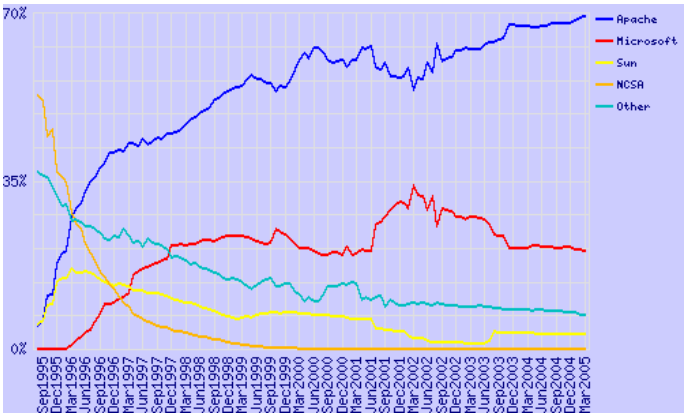


Figura 1.1: Estadísticas uso Apache

Su principal desarrollador (Rob McCool) abandonó poco después el NCSA y el proyecto. Sin embargo, bastantes personas siguieron trabajando con el servidor HTTPd NCSA y así fueron surgiendo diversos parches para el código fuente.

Fue entonces cuando un grupo de desarrolladores (ocho personas en principio) comenzaron a trabajar sobre HTTPd y los parches que habían ido mejorándolo: surgía el proyecto Apache (`http://www.apache.org/`).

[1] Y parece que hasta ahora lo ha conseguido.

[2] La última versión estable es la 2.0.53. Para saber las nuevas funcionalidades de Apache 2.0 `http://httpd.apache.org/docs-2.0/es/new_features_2_0.html`. Todavía siguen utilizándose las versiones 1.3 (1.3.33 es la última) sobre todo por compatibilidad con módulos de terceros, aunque se recomiendan las versiones 2.0 porque incorporan muchas nuevas funcionalidades.

La primera versión oficial, Apache 0.6.2, se lanzó en abril de 1995 (el nombre proviene de "A PAtCHy" release, ya que en principio era una versión parcheada del HTTPd 1.3 NCSA). El 1 de diciembre de 1995 se hizo pública la versión 1.0[2]

En 1998 se llegó a un acuerdo con IBM que permitió conseguir que Apache funcionara también bajo Windows, convirtiéndose en una excelente alternativa a IIS (*Microsoft Internet Information Server*).

Apache es el servidor Web (protocolo HTTP) más utilizado en el mundo actualmente. Se encuentra muy por encima de sus competidores, ya sean gratuitos o comerciales[3]. Por supuesto, es el más utilizado en sistemas Linux.

En su forma más simple, un servidor web transmite páginas en formato HTML a los navegadores cliente (Firefox, Netscape, Internet Explorer, Opera, Lynx...) utilizando el protocolo HTTP. Pero un servidor web hoy día puede hacer mucho más, ya sea por sus propios medios o mediante su integración con otros programas o módulos. Prácticamente todos los programas y aplicaciones informáticas tienden a que su forma de presentación para el cliente sea en formato web.

Existen varias formas en las que Apache puede proveernos contenidos:

- **Páginas estáticas** Es el modo básico y más primitivo, pero que en un gran número de casos es lo único que se necesita: transferir ficheros HTML, imágenes... Puede que con un servidor Linux de bajas prestaciones (incluso un 486) consigamos estupendos resultados, si es sólo esto lo que necesitamos.

- **Contenido dinámico** La información cambia constantemente, y un medio para mantener nuestras páginas actualizadas, es generarlas dinámicamente desde una base de datos, ficheros u otras fuentes de datos.

Apache posee muchas facilidades para generar este tipo de contenido.

1. **Soporte del protocolo HTTP 1.1**. Además mantiene la compatibilidad con el HTPP 1.0.

2. **Scripts CGI** y **FastCGI**. CGI viene de *common gateway interface*. Los scripts CGI son programas externos que se llaman desde el propio servidor cuando una página lo requiere. El CGI recibe información del servidor web y genera como salida una página web dinámica para el cliente. El script puede realizarse en cualquier lenguaje de programación siempre que siga las reglas del interfaz CGI. El problema es que es un proceso lento, al tenerse que lanzar un proceso externo al servidor web por cada petición. Perl es uno de los lenguajes más utilizados para ello, aunque también se utilizan scripts de una shell Unix/Linux.

3. **Host virtuales**. Permite atender varios sitios Web en dominios distintos, desde la misma máquina.

4. **Autenticación HTTP**. Permite restringir recursos a determinados usuarios o grupos (distintos de los del sistema).

5. **Intérpretes incluidos en Apache.** Tienen la ventaja sobre los cgi de que están incluidos en el propio Apache y no hay que lanzar un nuevo proceso por cada petición. Los módulos más utilizados son PHP y `mod_perl`.

6. Soporte de **SSI**[4] (*Server Side Includes*) y de **SSL**[5] (*Secure Sockets Layer*)

[3] En Marzo de 2005 casi el 70 % de los servidores Web usan Apache, para saber exactamente los datos en la actualidad se puede consultar
`http://news.netcraft.com/archives/web_server_survey.html`

[4] Directivas que permiten añadir funcionalidad añadida al HTML interpretándolas antes de mandar la página al navegador.

[5] Proporciona cifrado de datos para asegurar la privacidad y fiabilidad de la comunicación Web. Se utiliza criptografía asimétrica y certificados digitales para intercambiar una clave de sesión simétrica.

7. **Servlets y JSP en Java**. Es una opción que se utiliza en los servidores de aplicaciones, por ejemplo Tomcat, JBoss, Oracle IAS, WebSphere de IBM o BEA Weblogic. Su gran ventaja sería la portabilidad y escalabilidad. Desarrollamos en Java y podemos ejecutarlo en cualquier máquina virtual compatible. Un modelo muy utilizado en la actualidad es el de las arquitecturas en capas. Una arquitectura en tres capas utilizaría un cliente web para la capa de presentación, un servidor de aplicaciones que proporciona la lógica de negocio, es decir, el cómo se ejecutan los procesos dentro de la organización, y un servidor de bases de datos.

   La figura que vemos a continuación nos presenta un servidor web tradicional (en la parte derecha) y un modelo de arquitectura en tres capas (a la izquierda).

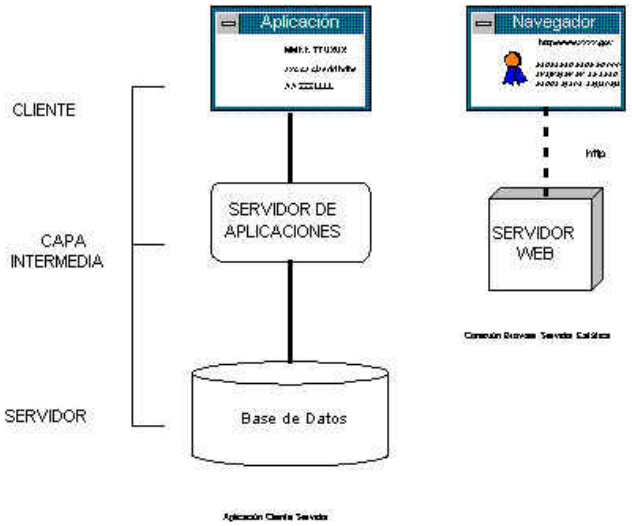


Figura 1.2: Arquitectura en capas

1.2. Instalación

Montar un servidor Web en nuestro Linux es muy sencillo. El servidor web que viene en la distribución es el Apache, funciona como servidor independiente (no lo activa `xinetd`) y escucha por defecto en el puerto 80. Nos aseguramos de que tengamos el paquete apache instalado en nuestra máquina.

1.2.1. Red Hat/Fedora

Instalemos los paquetes[6]

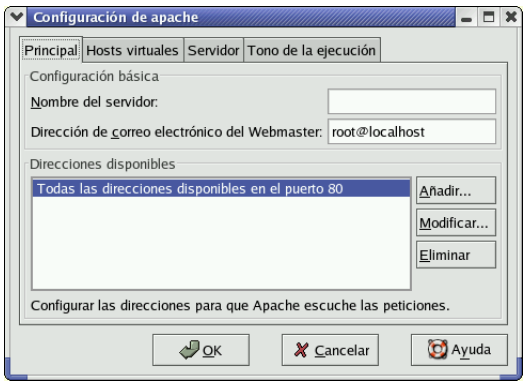
```
## apt-get install httpd httpd-manual httpd-devel system-config-httpd
Leyendo listas de paquetes... Done
Construyendo árbol de dependencias... Done
Se instalarán los siguientes paquetes extras:
  apr apr-devel apr-util apr-util-devel httpd-suexec pcre-devel
Se instalarán los paquetes NUEVOS siguientes:
  apr apr-devel apr-util apr-util-devel httpd httpd-devel httpd-
manual httpd-suexec pcre-devel
  system-config-httpd
0 upgraded, 10 newly installed, 0 removed and 166 not upgraded.
```

[6]La lista de dependencias no tiene por qué coincidir con la de vuestros sistemas.

```
Need to get 4163kB of archives.
After unpacking 18,3MB of additional disk space will be used.
¿Quiere continuar? [S/n]
```

Estos serían el paquete básico de apache, el manual y el paquete de desarrollo (sólo el primero es imprescindible).

Hay un cuarto paquete que nos puede facilitar enormemente la configuración de Apache. Se trata de `system-config-httpd`, es una herramienta gráfica para la configuración de Apache.



Si se desea usar, se puede consultar el *Manual de personalización de Red Hat Linux* [12]. Una nota a tener en cuenta y que aparece en en él:

> **"Aviso**
>
> No modifique "a mano" el fichero de configuración de Apache `/etc/httpd/conf/httpd.conf` si desea utilizar esta herramienta. Dicha herramienta crea este fichero después de que haya grabado los cambios y haya salido del programa. Si desea añadir módulos u opciones que no se encuentren en la herramienta no podrá usarla."

En el momento del arranque, si existe el enlace `/etc/rc.d/rc3.d/S85httpd,` se arrancará de forma automática. Si no existe, podemos crearlo con cualquiera de las herramientas: `serviceconf`, `ntsysv` o `chkconfig`.

Si queremos activarlo de forma manual alguna vez, podemos ejecutar

```
#/etc/rc.d/init.d/httpd start
```

o

```
#service httpd start
```

y si lo queremos parar

```
#/etc/rc.d/init.d/httpd stop
```

1.2.2. Guadalinex (Debian)

En este caso, lo único que cambia es el nombre de los paquetes[7]:

[7]Es mejor asegurarnos primero de que nuestra lista de paquetes está actualizada, ejecutando `#apt-get update`

```
# apt-get install apache2-mpm-prefork apache2-doc apache2-dev
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Nota, seleccionando apache2-threaded-dev en lugar de apache2-dev
Se instalarán los siguientes paquetes extras:
  apache2-common apache2-threaded-dev autoconf autotools-
dev libapr0 libapr0-dev libdb4.2-dev libexpat1-dev
  libldap2-dev libpcre3-dev libssl-dev libtool ssl-cert zlib1g-dev
Paquetes sugeridos:
  autoconf2.13 autobook autoconf-archive gnu-standards db4.2-
doc libtool-doc g77 fortran77-compiler gcj
Paquetes recomendados
  automaken libltdl3-dev
Se instalarán los siguientes paquetes NUEVOS:
  apache2-common apache2-doc apache2-mpm-prefork apache2-threaded-
dev autoconf autotools-dev libapr0 libapr0-dev
  libdb4.2-dev libexpat1-dev libldap2-dev libpcre3-dev libssl-
dev libtool ssl-cert zlib1g-dev
0 actualizados, 16 se instalarán, 0 para eliminar y 119 no actualiza-
dos.
Necesito descargar 10,3MB de archivos.
Se utilizarán 30,5MB de espacio de disco adicional después de desempa-
quetar.
¿Desea continuar? [S/n] n
```

De nuevo, el único paquete fundamental es el primero[8]. El segundo (`apache2-doc`) es la documentación que acompaña al programa, y el tercero el paquete de desarrollo.

En general, el demonio se iniciará de forma automática y se crearán las entradas adecuadas en los script de arranque. Si esto no es así, ejecutaremos:

```
# /etc/init.d/apache2 start
```

para ponerlo en marcha, y

```
# update-rc.d apache2 defaults
```

para añadir las entradas adecuadas para que el servidor web se inicie en los niveles de ejecución estándar de Debian.

¿Qué es eso de `prefork`?

Apache 2 basa su arquitectura en módulos multiproceso (*Multi-Processing Modules*). Si desde Debian ejecutamos

```
# apt-get install apache2
```

se instala el paquete `apache2-mpm-worker`. En la distribución Sarge de Debian tenemos los paquetes

```
apache2-mpm-worker
apache2-mpm-threadpool
apache2-mpm-prefork
apache2-mpm-perchild
```

Con ellos podemos cambiar la forma en que el servidor Web inicia los procesos y las solicitudes (basándose en hijos o hilos). El significado de cada uno se estos paquetes es[9]:

[8] Debian suele instalar la versión 1.3.x de Apache por defecto con algunos programas. Puede convivir con Apache 2, porque se instalan en directorios diferentes, aunque debemos tener cuidado de arrancar solamente uno de ellos, o si no, el primero que arranque se apoderará del puerto 80 y será el "vencedor".

[9] Si se desea saber más sobre las diferencias existentes entre ellos se pueden consultar:

worker un híbrido multihilos y multiprocesos de servidor Web.

threadpool cada proceso hijo puede tener varios hilos.

prefork servidor sin hilos. Es el más fácil de enlazar con php

perchild cada proceso hijo puede tener varios hilos, además, permite que los procesos demonio puedan ser asignados a usuarios dierentes.

Hemos optado por el módulo *prefork* (servidor sin hilos, donde para cada solicitud al servidor Web es necesario que se inicie un proceso hijo que la atienda) por coherencia con el que se instala para Fedora: con este módulo Apache 2 se comporta de igual forma que la versión 1.3 de Apache, aceptando las mismas directivas.

Podemos saber qué MPM estamos usando con[10]

```
# apache2 -V
Server version: Apache/2.0.51
Server built:   Sep 18 2004 17:21:03
Server's Module Magic Number: 20020903:9
Architecture:   32-bit
Server compiled with....
 -D APACHE_MPM_DIR="server/mpm/prefork"
 -D APR_HAS_SENDFILE
 -D APR_HAS_MMAP
 -D APR_HAVE_IPV6 (IPv4-mapped addresses enabled)
 -D APR_USE_SYSVSEM_SERIALIZE
 -D APR_USE_PTHREAD_SERIALIZE
 -D SINGLE_LISTEN_UNSERIALIZED_ACCEPT
 -D APR_HAS_OTHER_CHILD
 -D AP_HAVE_RELIABLE_PIPED_LOGS
 -D HTTPD_ROOT=""
 -D SUEXEC_BIN="/usr/lib/apache2/suexec2"
 -D DEFAULT_PIDLOG="/var/run/httpd.pid"
 -D DEFAULT_SCOREBOARD="logs/apache_runtime_status"
 -D DEFAULT_LOCKFILE="/var/run/accept.lock"
 -D DEFAULT_ERRORLOG="logs/error_log"
 -D AP_TYPES_CONFIG_FILE="/etc/apache2/mime.types"
 -D SERVER_CONFIG_FILE="/etc/apache2/apache2.conf"
```

1.2.3. ¡A navegar!

Una vez instalado el servidor, disponemos de un script en ambas distribuciones que nos permite controlar su estado, se trata de

`apache2ctl` en Debian

`apachectl` en Fedora

- Las páginas 45-46 de *Servidor Apache 2* [10]
- `http://httpd.apache.org/docs-2.0/es/mpm.html`

[10] Para Fedora

```
httpd -V
```

Como podemos observar la salida nos informa de bastantes más aspectos de la configuración del servidor Web

En el resto del tema será el que usemos tanto para Fedora como para Debian[11]. Admite los argumentos en línea de comandos:

`start` para arrancar el servidor. Si ya está en marcha, nos avisará de ello.

`graceful` con este parámetro le indicamos al servidor que relea los ficheros de configuración sin cerrar las conexionas activas. Las conexiones nuevas se iniciarán con la nueva configuración.

`restart` reinicia el servidor (en su caso con la nueva configuración), pero a diferencia del anterior, cierra las conexionas activas.

`stop` cierra el servidor y, por tanto, las conexiones activas.

➥ **Para practicar** Aunque los anteriores son los más usuales, también podemos usar: `fullstatus`, `status` y `configtest`. Comprobar qué funcionalidad tienen. ■

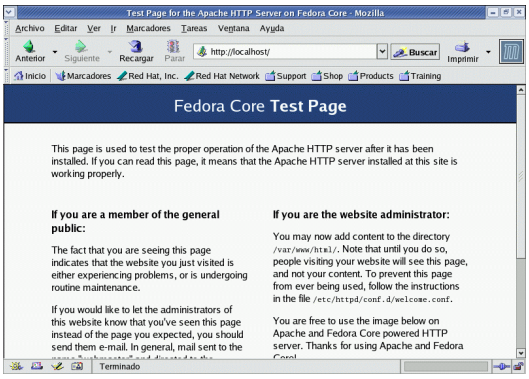
Pongamos en marcha nuestro servidor Web con

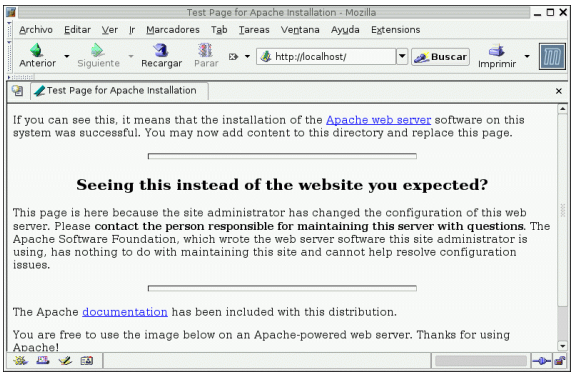
```
#apache2ctl start
```

y, para comprobar que funciona, podemos apuntar nuestro navegador preferido a la dirección

`http://172.26.0.2`[12]

Si conseguimos una pantalla de bienvenida del servidor apache (*It worked!*), ya hemos contactado con nuestro servidor.





Si hemos instalado el paquete del manual podremos acceder a la extensa documentación[13](en castellano) que acompaña al programa desde esta misma página:

[11] Dependiendo de la distribución que tengamos en ese momento como referencia, usaremos la versión adecuada. Tendréis que adecuar el comando a la distribución con que trabajéis en vuestro ordenador.

[12] Si es ésta nuestra dirección, o con `http://localhost` y nos serviría incluso si no tenemos tarjeta de red ni configuración de red.

[13] A nuestra disposición también en `http://httpd.apache.org/docs-2.0/es/`

1.3. Configuración

1.3.1. Document root

El siguiente paso es poner nuestras propias páginas en el servidor, en vez de las de bienvenida de apache. Sin más que ponerlas en el directorio

Fedora: `/var/www/html`[14]

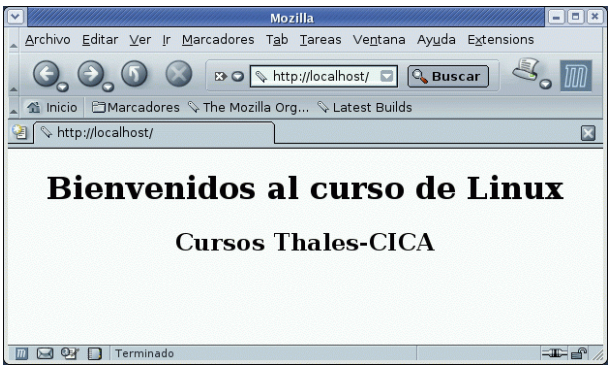
Debian: `/var/www/apache2-default/`

y empezando con la página `index.html`, podremos ver nuestras propias páginas.

➥ **Para practicar:** Comprobar que si ponemos el fichero `index.html`

```
$cat index.html
<center>
  <h1>Bienvenidos al curso de Linux</h1>
  <h2>Cursos Thales-CICA</a>
</center>
```

en la ruta adecuada se obtiene:



■

[14] En versiones antiguas estaban en `/home/httpd/html`

1.3.2. Ficheros de configuración

Fedora: `/etc/httpd`

Los ficheros de configuración se encuentran en el directorio `/etc/httpd`. Si no es porque realmente lo necesitemos, con la configuración que viene por defecto podemos trabajar de forma satisfactoria.

- El fichero de configuración se llama `httpd.conf`[15] y se encuentra en `/etc/httpd/conf`
- `/etc/httpd/conf.d/` en este directorio se guardan los archivos de configuración para módulos individuales como `ssl.conf`, `perl.conf, php.conf`, etc[16]. Se incluyen en el fichero de configuración a través de la directiva

  ```
  Include conf.d/*.conf
  ```

  que aparece en `/etc/httpd/conf/httpd.con` y sus nombres han de terminar en `.conf`

Debian: `/etc/apache2/apache2.conf`

En Debian el archivo de configuración principal es `/etc/apache2/apache2.conf` y aunque en ese mismo directorio existe `httpd.conf`, está vacío[17].

En el subdirectorio `/etc/apache2` se encuentran además los ficheros:

`magic` lo normal es que no tengamos que modificar nunca este fichero. En él se almacenan los datos "mágicos" del módulo `mod_mime_magic` (para determinar el tipo MIME del fichero mirando unos pocos bytes del contenido)

`ports.conf` directivas de configuración para los puertos y direcciones IP a la escucha.

y los directorios:

`conf.d/` los archivos de este directorio se incluyen en el fichero de configuración a través de la directiva

```
Include /etc/apache2/conf.d/[^.#]*
```

`mods-avalaible/` archivos `.load` (contienen las directivas de Apache para cargar un módulo) y `.conf` (necesarios para configurar ese módulo)

`mods-enabled/` si deseamos activar un módulo hay que crear un enlace simbólico en este directorio para los archivos `.load` (y `.conf` si existen) asociados con el módulo del directorio `mods-available`. Se incluyen en el fichero de configuración por medio de la directivas:

```
Include /etc/apache2/mods-enabled/*.load
Include /etc/apache2/mods-enabled/*.conf
```

Hay dos herramientas que permiten gestionar estos enlaces de forma sencilla

`a2enmod modulo` para crear los enlaces simbólicos asociados al módulo.

`a2dismod modulo` para eliminar los enlaces simbólicos para ese módulo.

[15] Antes, Apache utilizaba además `srm.conf` y `access.conf`, pero en las versiones actuales, el contenido de ambos se ha incluido en `httpd.conf`

[16] También está el fichero de configuración (`welcome.conf`) de la página de bienvenida para el caso de que no exista un fichero `index.html` en el raíz de apache.

[17] Se incluye en el fichero de configuración con:

```
Include /etc/apache2/httpd.conf
```

↬ Veamos un ejemplo sobre su uso. Supongamos que no deseamos que nuestros usuarios puedan alojar su Web en sus `$HOME` de usuario[18] y que sean acesibles mediante `http://www.misitio.com/~usuario`. Para eso necesitamos ver que no esté activo el módulo `mod_userdir` de Apache. Listemos el contenido de:

```
# ls /etc/apache2/mods-enabled/
cgi.load userdir.conf userdir.load
```

Nos encontramos con que está activo, para eliminar los enlaces simbólicos ejecutamos

```
root@eco:~# a2dismod userdir
Module userdir disabled; run /etc/init.d/apache2 force-
reload to fully disable.
root@eco:~# /etc/init.d/apache2 force-reload
Forcing reload of web server: Apache2.
root@eco:~# ls /etc/apache2/mods-enabled/
cgi.load
```

Y, tras reiniciar el servidor, comprobamos que todo está bien (se ha eliminado el enlace). Si ahora deseamos volver a la situación original sólo hemos de escribir

```
# a2enmod userdir
Module userdir installed; run /etc/init.d/apache2 force-
reload to enable.
root@eco:~# /etc/init.d/apache2 force-reload
Forcing reload of web server: Apache2.
root@eco:~# ls /etc/apache2/mods-enabled/
cgi.load  userdir.conf  userdir.load
```

Y todo queda como al principio.

`sites-available/` como `mods-available`, excepto que contiene ficheros de configuración para hosts virtuales diferentes que podrían usarse con apache (el *hostname* no tiene que corresponder exactamente con el nombre de archivo). En la instalación se crea un fichero de configuración para el host virtual por defecto de nombre `default`.

`sites-enabled/` contiene enlaces simbólicos a los lugares `sites-available` que deseamos activar. Se incluyen en `apache2.conf` por la directiva

```
Include /etc/apache2/sites-enabled/[^.#]*
```

De igual manera que antes, disponemos de dos herramientas que nos facilitan este trabajo:

`a2ensite site` para crear los enlaces simbólicos asociados al sitio.

`a2dissite site` para eliminar los enlaces simbólicos para ese sitio.

↬ Por ejemplo, si tenemos un host virtual (trataremos esto después) de nombre *matematicas*, creamos su fichero de configuración de nombre *mate* (notar que los nombres no tienen por qué coincidir) en el directorio `/etc/apache2/sites-available`. Para activarlo ejecutaremos (reiniciando Apache para que los cambios sean efectivos)

`a2ensite mate` para crear los enlaces simbólicos asociados a "matematicas" en `/etc/apache2/sites-enabled`

`a2dissite mate` para eliminar los enlaces simbólicos para el *host virtual* "matematicas".

[18] Más adelante ampliaremos sobre este tema

1.4. /etc/http/httpd.conf

Como ya se ha comentado en la introducción, nos centraremos en documentar este fichero. Si optáis por trabajar con Debian sólo hay que adecuar lo aquí explicado a los ficheros de configuración antes comentados. El que se haga así se justifica desde la perspectiva de que es el sistema más estándar y documentado de trabajar con el servidor Web Apache.

Además, en las notas a pie de página explicitaremos las diferencias respecto a la configuración por defecto con Debian.

El archivo `httpd.conf`[19] está bien comentado y es bastante autoexplicativo. La configuración predeterminada funciona para los ordenadores de la mayoría de los usuarios, así que probablemente no se necesitará cambiar ninguna de las directivas en el fichero `httpd.conf`. Sin embargo, es bueno conocer las opciones de configuración más importantes.

Consta de tres secciones:

- Configuración global.
- Configuración del servidor principal.
- Configuración de los Servidores Virtuales.

Antes de modificar el fichero `httpd.conf` es conveniente hacer una copia del fichero original, dándole por ejemplo, el nombre `httpd.conf.ori, httpd.conf.20050319` u otro que nos sea significativo[20]. Si cometemos un error mientras estamos modificando el fichero de configuración, no debemos preocuparnos, porque siempre dispondremos de una copia de seguridad.

Si cometemos un error y nuestro servidor web no funciona correctamente, el primer sitio donde acudir es a lo último que acabamos de modificar en `httpd.conf`. Después podemos consultar el fichero de archivo de errores[21] (`/var/log/httpd/error_log`), las últimas entradas deberían servirnos de ayuda para saber lo que ha pasado.

La configuración de Apache se basa en una serie de directivas que tienen posibilidad de ser usadas dentro de un contexto, es decir, un ámbito en el que pueden ser aplicadas. Hay cuatro posibilidades que no son excluyentes:

- configuración global del servidor,
- secciones para configurar los host virtuales,
- secciones de configuración de directorios
- archivos `.htaccess`

  ↬ Por ejemplo, la directiva `ErrorLog` que nos permite establecer la ubicación del fichero para el registro de error, sólo se puede usar en la configuración global del servidor o al configurar los host virtuales.

A continuación se dan breves descripciones de las directivas incluidas en el fichero `httpd.conf`, ordenadas según se encuentran en él. Para una referencia más amplia de algunas de estas directivas véase la documentación instalada (`http://localhost/manual/es/mod/quickreference.html`).

ServerRoot El comando `ServerRoot` se va a referir al directorio principal donde se encuentran todos los ficheros de configuración y trabajo del servidor. Su valor es `/etc/httpd`[22].

[19] Recordar que como material adicional en Moodle disponéis del fichero de configuración de Fedora.
[20] El objetivo es poder volver al estado original en caso de que algo vaya mal.
[21] `Debian: /var/log/apache2/error.log`
[22] `Debian: /etc/apache2`

User La directiva `User` establece el *userid* que utiliza el servidor para ejecutarse y responder a las peticiones. El valor de `User` determina el acceso que tendrá el servidor web a los ficheros y directorios en los que se encuentran las páginas. Cualquier fichero al que no pueda acceder este usuario, será inaccesible para el servidor web y como consecuencia, también inaccesible al visitante de la web. El comando predeterminado para `User` es `apache`[23].

El usuario `User` también es dueño de cualquier proceso CGI que arranque el servidor y no se le debería permitir ejecutar ningún código que no esté pensado para responder peticiones HTTP.

El proceso httpd padre se inicia 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 por defecto es el 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.

Group El comando `Group` es similar a `User`. `Group` establece el grupo en el que el servidor responde a las peticiones. El valor predeterminado del comando Group también es `apache`[24], en este caso como grupo, y no como usuario.

DocumentRoot `DocumentRoot` es el directorio que contiene la mayoría de los archivos HTML que se entregarán en respuesta a peticiones. El directorio predeterminado `DocumentRoot` es `/var/www/html`[25]. Por ejemplo, el servidor puede recibir una petición para el siguiente documento:

`http://localhost/prueba.html`

El servidor buscará el fichero en el siguiente directorio por defecto:

`/var/www/html/prueba.html`

Directory: Las etiquetas `<Directory /path/a/directorio>` y `</Directory>` se usan para agrupar directivas de configuración que sólo se aplican a ese directorio y sus subdirectorios. Cualquier directiva aplicable a un directorio puede usarse en las etiquetas `<Directory>`. Las etiquetas `<File>` pueden aplicarse de la misma forma a un fichero específico.

Por defecto, se aplican parámetros muy restrictivos al directorio raíz (`/`).

```
<Directory />
   Options FollowSymLinks
   AllowOverride None
</Directory>
```

Con la directiva `Options` establecemos qué características están disponibles para el directorio en el que se establece, su sintaxis es:

```
Options [+|-]opcion [+|-]opcion ...
```

y las opciones pueden ser

`All` Todas las opciones excepto para `MultiViews`. Es el entorno por defecto.

`ExecCGI` Se permite ejecutar scripts CGI usando `mod_cgi`.

`FollowSymlinks` El servidor seguirá los enlaces simbólicos en este directorio. Aunque el servidor siga los enlaces simbólicos, no cambia el nombre de path usado para comparar las secciones `<Directory>`. Esta opción se ignora si se establece dentro de una sección `<Location>`

[23] `Debian: www-data`
[24] `Debian: www-data`
[25] `Debian: /var/www, aunque la orden RedirectMatch ^/$ /apache2-default/ hace que vaya a /var/www/apache2-default`

`Includes` Se permiten inclusiones por parte del servidor proporcionadas por `mod_include`.

`IncludesNOEXEC` Se permiten inclusiones por parte del servidor, pero están desactivados `#exec cmd` y `#exec cgi`. Está activo para scripts CGI `#include` virtual desde directorios `ScriptAlias`.

`Indexes` Si hay una petición de una URL de un directorio y en él no hay `DirectoryIndex` (ej: `index.html`), `mod_autoindex` devolverá un listado formateado del directorio.

`MultiViews` Los contenidos negociados "MultiViews" se permiten usando `mod_negotiation`.

`SymLinksIfOwnerMatch` El servidor sólo seguirá los enlaces simbólicos para aquellos archivos o directorios que posean la misma identidad de usuario que el enlace. Esta opción se ignora si se establece dentro de una sección `<Location>`

Normalmente, si se pueden aplicar varias `Options` a un directorio sólo se usa la más específica, ignorándose las demás; las opciones no se mezclan. En cualquier caso, si todas las opciones de la directiva `Options` van precedidas por el símbolo + o -, se mezclarán. Cualquier opción precedida por + se añadirá a las opciones en vigor, y cualquiera precedida por -, se eliminará.

Tal cual está, es equivalente a

```
Options FollowSymLinks -ExecCGI -Includes -Indexes -Multiviews
```

e implicaría que está permitido atravesar los enlaces simbólicos en todo el sistema.

Con la opción *`AllowOverride`* puesta al valor *`None`* establecemos que el servidor no leerá el archivo especificado en `FileName` (por defecto, `.htaccess`). Esta directiva permite especificar qué partes del servidor pueden ser establecidas en los archivos `.htaccess`, los valores que puede tomar (además del comentado) son:

`AuthConfig` permite el uso de directivas de autorización (por ejemplo: `AuthName`, `AuthType`, `Require`, ...)

`FileInfo` permite el uso de directivas que establecen el tipo de documento (por ejemplo: `DefaultType`, `ErrorDocument`, `...`)

`Indexes` permite usar directivas para controlar la forma en que se realizan los índices de los directorios (por ejemplo: `AddIcon`, `IndexOptions`, etc)

`Limit` permite el uso de directivas para establecer el control de acceso (`Allow`, `Deny` y `Order`)

`Options` permite usar directivas que controlan opciones específicas del directorio (`Options` y `XBitHack`)

Con esta configuración, cualquier directorio del sistema que necesite valores más permisivos ha de ser configurado explícitamente.

El directorio cgi-bin está configurado para permitir la ejecución de scripts CGI, con la opción `ExecCGI`. Si se necesita ejecutar un script CGI en cualquier otro directorio, habrá que configurar ExecCGI para ese directorio. Por ejemplo, si cgi-bin es `/var/www/cgi-bin`, pero se quieren ejecutar scripts CGI desde `/home/usuario/cgi-bin`, se añadirá una directiva `ExecCGI` y un par de directivas `Directory` como las siguientes, en el fichero `httpd.conf`:

```
<Directory /home/usuario/cgi-bin>
   Options +ExecCGI
</Directory>
```

Para permitir la ejecución de scripts CGI en `/home/usuario/cgi-bin`, habrá que llevar a cabo pasos extra aparte de configurar `ExecCGI`. El valor de los permisos para scripts CGI y el recorrido entero a los scripts, debe ser de 0755 (accesible para el usuario que ejecuta apache). Además, el dueño del script y del directorio deben ser el mismo.

UserDir `UserDir` es el nombre del subdirectorio, dentro del directorio de cada usuario, donde estarán los archivos HTML que podrán ser servidos. Por defecto, el subdirectorio es `public_html`. Por ejemplo, el servidor podría recibir la siguiente petición:

`http://localhost/~usuario/prueba.html`

El servidor buscaría el fichero:

`/home/usuario/public_html/prueba.html`

En el ejemplo, `/home/usuario` es el directorio del usuario.

Hay que asegurarse que

- Los permisos sean los adecuados
  - De los directorios de usuario sean correctos (por ejemplo 711).
  - Los bits de lectura (r) y ejecución (x) deben estar activados en el directorio `public_html` (0755 valdrá).
  - El valor de los permisos con que se servirán los ficheros desde `public_html` debe ser 0644 por lo menos.
- Para **Fedora**[26] permitir el acceso usando el módulo `mod_userdir`. Con él conseguimos que Apache permita o deniegue esta forma de acceso. Así, si deseamos activar esta posibilidad, hemos de cambiar la sección como sigue:

```
<IfModule mod_userdir.c>
  #UserDir disable
  UserDir public_html
</IfModule>
```

y releer después la configuración del servicio[27].

ErrorLog `ErrorLog` nombra el fichero donde se guardan los errores del servidor. Por defecto, el fichero de error del servidor es `/var/log/httpd/error_log`[28]. El `log` de errores es un buen sitio para ver si el servidor genera errores y no se sabe muy bien qué pasó.

↬ Una línea de ejemplo puede ser

```
[Sun Mar 21 05:39:18 2004] [error] [client 66.90.73.73] Fi-
le does not exist:
/var/www/html/sumthin
```

CustomLog con esta directiva establecemos la ubicación y formato del archivo de registro de los accesos. Por defecto[29]

```
CustomLog log/ac-
cess.log "%h%l%u%t \"%r\"%>s%b \"%{Referer}i\" \"%{User-Agent}i\""
```

es decir[30]:

- registramos el host remoto (`%h`), la identidad del cliente (`%l`), si se necesita autenticación para la URL solicitada, el nombre de usuario (`%u`) y el tiempo de solicitud (`%t`).

[26] En Guadalinex se activa por defecto (si no fuese así: `#a2enmod userdir` y reiniciar el servidor). Para ver su fichero de configuración consultar:

`/etc/apache2/mods-available/userdir.conf`

[27] `# service httpd reload`

[28] `Debian: /var/log/apache2/error.log`

[29] Equivale a `combined`. Se define en el propio fichero de configuración de apache.

[30] No se analizan todos los posibles valores, sólo los que aparecen por defecto.

- se entrecomilla la primera línea de la solicitud (`%r`), almacenamos el estado devuelto por el servidor en respuesta a la solicitud (`%>s`) y los bytes enviados (`%b`)
- además de la cabecera enviada por el cliente al solicitar la página web, almacenamos la URL de la página solicitada (`%{Referer}i`) y el navegador web usado (`%{User-Agent}i`).

↬ Una línea de ejemplo:

```
80.83.190.1 - paco [21/Mar/2005:15:47:59 +0100] "GET /isoqlog/images/pk.gif
HTTP/1.0" 200 246 "http://www.midominio.com/isoqlog/" "Mozi-
lla/5.0 (X11; U; Linux
i686; es-ES; rv:1.4.1) Gecko/20031114"
```

1.4.1. Debian

Como ya hemos comentado, la forma de organizar la configuración del servidor Apache en Guadalinex es más modular que la Fedora (véase 1.3.2 en la página 13). Cuestiones específicas que merece la pena destacar o recordar respecto a lo ya estudiado:

- Recordar que el fichero principal de configuración del servidor es `/etc/apache2/apache2.conf` y que con varias directivas `Include` se incluyen en él los ficheros de configuración de
  - Los módulos activos del directorio `/etc/apache2/mods-enabled`
  - El fichero `/etc/apache2/httpd.conf` que en principio está vacio[31].
  - Los ficheros de configuración de los hosts virtuales activos: `/etc/apache2/sites-enabled`
- Como ya hemos comentado, en el fichero `/etc/apache2/sites-avalaible/default` definimos la configuración del host virtual por defecto. De él comentar sólo dos cuestiones:
  1. Con los asteriscos de las directivas

     ```
     NameVirtualHost *
     <VirtualHost *>
         ....
     </Virtualhost>
     ```

     indicamos que se aplica a cualquier dirección IP y puerto en los que escuche Apache. Es decir, se aplica a todos nuestros interfaces de red. Para saber más véase `http://httpd.apache.org/docs-2.0/es/mod/core.html#virtualhost`.
  2. Con la directiva

     ```
     RedirectMatch ^/$ /apache2-default/
     ```

     redirigimos las solicitudes de la página principal (especificada mediante una expresión regular[32]) a la nueva ubicación. Es decir, al escribir `http://www.midominio.org` se nos redirigirá a `http://www.midominio.org/apache2-default/`. Si bien lo mantendremos así, sería buena opción comentarla.

➥ Para practicar: Web de usuarios

1. Montar el servidor web Apache y comprobar que los usuarios del sistema pueden acceder a sus páginas web personales[33]. Supongamos que en nuestra máquina hay un usuario de nombre THALES

[31] Después veremos una práctica en la que lo modificamos.

[32] Con `^/$` indicamos la cadena que comienza (`^`) y termina (`$`) con `/`, es decir, la petición es `"/"`, como por ejemplo `http://localhost/`.

[33] Con Guadalinex sólo hay que realizar los apartados: a), b) y e)

a) Para el usuario THALES crear el directorio `$HOME/public_html`

```
$mkdir public_html
```

b) Poner en él un fichero html simple de nombre `index.html`, por ejemplo:

```
<html>
<body>
<h1>Esta es la web de thales</h1>
</body>
</html>
```

c) Modificar los permisos del `$HOME` de THALES, así como del directorio `public_html` para que Apache pueda acceder a él:

```
$chmod o+x $HOME
$chmod o+rx $HOME/public_html
```

d) Permitir que Apache acceda a directorios de usuario mediante `http://servidor_web/~usuario`. Para ello cambiemos la sección del fichero de configuración del servidor como sigue

```
<IfModule mod_userdir.c>
# UserDir disable
UserDir public_html
</IfModule>
```

y releer después la configuración del servicio

```
#apachectl restart
```

e) Comprobar que funciona apuntando con nuestro navegador a la página web

```
http://127.0.0.1/~thales
```

2. Si bien la solución anterior es la estándar, "afea" bastante eso de tener que escribir la virgulilla[34] para acceder a las Web de usuario. Veamos la forma de apañar el entuerto. Partimos de que se ha realizado la práctica anterior y

a) Tenemos que instalar `mod_perl`, de manera que podremos ejecutar código basado en este lenguaje interactuando con el servidor Apache. Para instalarlo:

1) Fedora

```
# apt-get install mod_perl
Leyendo listas de paquetes... Done
Construyendo árbol de dependencias... Done
Se instalarán los paquetes NUEVOS siguientes:
  mod_perl
0 upgraded, 1 newly installed, 0 removed and 170 not upgraded.
Need to get 1486kB of archives.
After unpacking 3890kB of additional disk space will be used.
```

2) Guadalinex

```
# apt-get install libapache2-mod-perl2
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Se instalarán los siguientes paquetes extras:
  libdevel-symdump-perl
Se instalarán los siguientes paquetes NUEVOS:
  libapache2-mod-perl2 libdevel-symdump-perl
0 actualizados, 2 se instalarán, 0 para eliminar y 0 no actualiza-
dos.
Necesito descargar 623kB de archivos.
Se utilizarán 2781kB de espacio de disco adicional después de desem-
paquetar.
¿Desea continuar? [S/n]
```

[34] Incluso es difícil de localizar a veces.

Y, puede que activar el módulo
`#a2enmod perl`

b) Añadamos en el fichero de configuración de apache (podemos optar por ponerlo en ambos sistemas en `httpd.conf`) el script de perl

```
<Perl>
        opendir H, '/home/';
        my @dir = readdir(H);
        closedir H;
        foreach my $u (@dir) {
                next if $u =~ m/^\./;
                if (-e "/home/$u/public_html") {
                        push @Alias, ["/$u", "/home/$u/public_html
                            /"];
                }
        }
</Perl>
```

En ambos sistemas, habrá que reiniciar el servicio:

```
#apache2ctl restart
```

Básicamente, con él lo que hacemos es leer los directorios de usuario que contiene un directorio `public_html` y crear Alias asociados a ellos de la forma

```
Alias /thales /home/thales/public_html
```

Es decir, cuando escribamos `http://localhost/thales`, se redirigirá al directorio para el que hemos creado el Alias (`/home/thales/public_html`)

¿Queda mejor así, verdad? **-:)** ■

1.5. Autenticación

Podemos conseguir que para acceder a determinados recursos un cliente tenga que autenticarse ante el servidor. Por ejemplo, si deseamos mantener información sensible en nuestro sitio web (el módulo encargado es `mod_auth`[35]). La información contenida en esa zona sólo deberá ser vista por el usuario o grupo que establezcamos.

El proceso de autenticación es simple. El cliente envía su nombre y contraseña[36]. A continuación Apache comprueba su archivo[37] de nombres y contraseñas cifradas para ver si el cliente tiene derecho a acceder.

Podemos establecer de dos formas diferentes la autenticación:

- Globalmente: agregando una sección `<Directory /path/a/directorio>` y `</Directory>` en nuestro archivo `httpd.conf`[38] por cada directorio que deseemos proteger.

  ↬ Por ejemplo con

  ```
  <Directory /public/>
  ```

[35] No hay que hacer nada para cargarlo.
[36] En estos casos, además debemos configurar para que utilice SSL, porque un usuario y una contraseña en protocolos no cifrados, durarán sin ser conocidos, menos que un cubito de hielo en agosto.
[37] Puede ser un archivo de texto o una base de datos
[38] En Debian `/etc/apache2/sites-available/default`

```
  AuthType Basic
  AuthName "Pagina de thales"
  AuthUserFile /var/www/passwd/.htpasswd
  AuthGroupFile /dev/null
  require user thales
</Directory>
```

conseguimos el mismo resultado que con la práctica que se realiza un poco más adelante en esta página.

- Usando los ficheros especiales `.htaccess`[39]. Las directivas que se pongan dentro de los ficheros `.htaccess` se aplicarán sólo al directorio que lo contiene, así como a todos sus subdirectorios. Los archivos `.htaccess` se leen cada vez que hay una petición de páginas y, por tanto, no hay que reiniciar el servidor Web para que se activen los cambios que realicemos en ellos.

Las directivas de autenticación del módulo `mod_auth` (módulo de autenticación de Apache) son:

AuthUserfile asigna el nombre del archivo de texto que contendrá los nombres de usuario y contraseñas usadas en la autenticación HTTP básica

AuthGroupFile asigna el nombre del archivo de texto que contendrá la lista de grupos de usuarios usadas en la autenticación HTTP básica.

↬ Una línea de este archivo (en él se crea un grupo de nombre curso con tres usuarios) puede ser

```
curso: thales mileto pitagoras
```

AuthAuthoritative toma los valores `On` y `Off` (por defecto está en `On`). Permite que si usamos en un directorio varios métodos de autenticación diferentes y falla el primero, se pase al segundo.

➥ **Para practicar** Crear un directorio con acceso restringido al usuario THALES

1. Creemos el directorio[40]:

   ```
   # mkdir /var/www/html/public
   ```

   y pongamos en él una página web simple (la de antes nos puede servir) de nombre `index.html`.

2. Creemos el directorio en donde almacenar las claves de acceso, por ejemplo:

   ```
   # mkdir /var/www/passwd
   ```

   Hay varias formas de trabajar con archivos de contraseñas. Si son "pocos" usuarios[41]:

[39] Se puede especificar cualquier otro nombre en la directiva `AccessFileName`, pero éste es el valor por defecto.

[40]
```
Debian: # mkdir /var/www/apache2-default/public
```

[41]

- Este "pocos" hay que entenderlo con cierta flexibilidad. Hasta 100 usuarios más o menos va "de muerte". Si el número de usuarios a autenticar es mucho mayor, mejor usar el módulo `mod_auth_mysql`.
- Si no queremos que se resienta la seguridad, es muy importante tener en cuenta que el fichero `.htpasswd` esté situado fuera de DocumentRoot
- En Debian también podemos usar

  ```
  # htpasswd2 -c /var/www/passwd/.htpasswd thales
  ```

```
# htpasswd -c /var/www/passwd/.htpasswd thales
```

Así creamos (`-c`) el archivo con el primer usuario y se nos pedirá la contraseña (hacer notar que no tiene por qué ser un usuario del sistema). Después, para añadir otros usuarios, el parámetro `-c` no hay que ponerlo[42].

3. Creemos en `/var/www/html/public`[43] el fichero `.htaccess`

```
# cat /var/www/html/public/.htaccess
AuthType Basic
AuthName "Pagina restringida de thales"
AuthUserFile /var/www/passwd/.htpasswd
AuthGroupFile /dev/null
require user thales
```

Comentemos un poco el fichero: con la directiva `AuthType` con el valor `Basic` indicamos que la contraseña se negociará en texto plano. En el cuadro de verificación de contraseña, veremos el texto "Página restringida de thales". Por último indicamos a Apache el archivo en donde buscar la contraseña, que el grupo no importa y que el nombre de usuario requerido es THALES.

4. Modifiquemos el fichero[44] `/etc/httpd/conf/httpd.conf`

Si desde nuestro navegador web intentamos cargar la página:

`http://127.0.0.1/public/index.html`

podremos cargarla sin problema. Esto se debe a que en el fichero

`/etc/httpp/conf/http.conf`

hay una sección[45] como la que sigue (pero con menos comentarios y en inglés):

```
<Directory "/var/www/html">
# Puede ser "None", "All", o cualquier combinación de "Indexes",
# "Includes", "FollowSymLinks", "ExecCGI", o "MultiViews".
#
# Notar que "MultiViews" debe ser *explícitamente* llamado --- "Options
    All"
# no se lo proporciona.
# Si se descomenta, podríamos ver el contenido de los subdirectorios
    para
# los que no haya fichero html de inicio y además podríamos seguir
# enlaces simbólicos (con el problema de seguridad que representa)
  # Options Indexes FollowSymLinks
#
# Controla qué opciones pueden omitir los archivos .htaccess de los
    directorios
# Puede ser "All", o cualquier combinación de "Options", "FileInfo",
# "AuthConfig" y "Limit"
# En vuestro fichero estará la línea que sigue y eso implica
```

[42] Por defecto en Fedora el fichero creado tiene de dueño al root y de permisos 611. De esa forma, el servicio httpd puede leerlo sin problemas.

Como el servicio httpd se ejecuta como usuario apache, si usamos una versión de Apache que lo cree con permisos 600, el demonio no podrá leer su contenido. En este caso hay que cambiarlo de dueño (o relajarle los permisos)

```
# chown apache /var/www/passwd/.htpasswd
```

de esta forma el servidor Web podrá leer la contraseña introducida.

[43]

```
Debian: /var/www/apache2-default/public
```

[44] En Guadalinex el fichero a modificar es:
`/etc/apache2/mods-available/userdir.conf`

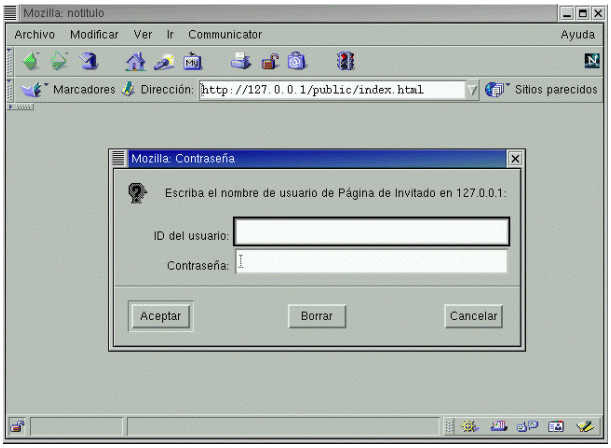
[45] En Guadalinex la sección se corresponde con `/var/www`

```
# que los ficheros .htacces no pueden modificar nada.
  # AllowOverride None
# Al poner esta otra permitimos que controlen la autenticación
  AllowOverride AuthConfig
#
# Controla quién puede obtener la respuesta de este servidor.
# Tal cual está primero se procesan primero las directivas allow y
# después las deny, además, la seguna línea permite el acceso a todo el
    mundo
  Order allow,deny
  Allow from all
</Directory>
```

O sea que, cambiamos el valor `None` de la directiva `AllowOverride` a `AuthConfig` para que podamos controlar la autentificación desde los archivos `.htaccess`, hacemos que apache relea el fichero de configuración

```
#apachectl restart
```

y ya debería ir todo bien.



1.6. Host Virtuales

Mediante los *host* virtuales, Apache nos brinda la posibilidad de alojar varios dominios en una sola máquina.

↬ Supongamos que queremos albergar dos nombres de servidor web en la misma máquina `servidor1.midominio.com` y `servidor2.midominio.com`[46] respondiendo los mismos a una sola IP y con una sola instancia de Apache configurada y que responde las llamadas por el puerto 80. Usando host virtuales, podemos conseguir que en el caso de que sea invocado `servidor1.midominio.com` vaya a leer los archivos en el directorio que hayamos configurado como `DocumentRoot`, y consecuentemente los mande al navegador de quien lo haya pedido, y en el caso que sea convocado como `servidor2.midominio.com`, vaya a leer los archivos a otro directorio. Es decir, habrá un directorio (`DocumentRoot`) para cada uno de los servidores virtuales que definamos.

Apache soporta dos tipos de host virtuales:

[46] Puede ser también de otro dominio totalmente distinto, como por ejemplo, servidor2.otrodominio.org

Host virtuales basados en nombres permiten alojar varios nombres de host (o dominios) en una misma máquina

Host virtuales basados en IP una máquina responde a diferentes direcciones IPs. Ya que las direcciones "públicas" no las regalan, no es lo más habitual para un centro y no lo vamos a estudiar.

En general, el caso más interesante es el primero (que una sola máquina responda a varios nombres) y es el que vamos a analizar.

➥**Para practicar:** vamos a configurar Apache para que responda de forma diferente cuando se invoque como `www.midominio.com` y `tux.midominio.com`. [47] Para eso supondremos que trabajamos sobre la IP 172.26.0.2.


El host virtual heredará los parametros del host principal que no se cambien para él.

1. El primer paso consiste en configurar los servicios DNS para que apunten a la misma dirección IP[48].
2.

Fedora: Modiquemos el fichero de configuración de apache como sigue[49]:

```
NameVirtualHost 172.26.0.2
<VirtualHost www.midominio.com>
        ServerAdmin webmaster@midominio.com
        DocumentRoot /var/www/htmlwww
        ServerName www.midominio.com
        ErrorLog /var/log/httpd/www-error.log
        CustomLog /var/log/httpd/www-access.log common
</VirtualHost>
<VirtualHost tux.midominio.com>
        ServerAdmin webmaster1@midominio.com
        DocumentRoot /var/www/htmltux
        ServerName tux.midominio.com
        ScriptAlias /cgi-bin/ /var/www/tux/cgi-bin
        Alias /images /var/www/tux/images
        ErrorLog /var/log/httpd/tux-error.log
        CustomLog /var/log/httpd/tux-access.log common
</VirtualHost>
```

Debian:

Podemos optar por crear dos ficheros de contenido

```
$cat /etc/apache2/sites-avalaible/www
NameVirtualHost 172.26.0.2
<VirtualHost www.midominio.com>
        ServerAdmin webmaster@midominio.com
        DocumentRoot /var/www/htmlwww
        ServerName www.midominio.com
        ErrorLog /var/log/httpd/www-error.log
```

[47] Podemos trasladar de forma inmediata el ejemplo al caso de que sean dominios diferentes.

[48] Si trabajamos sólo en local lo podemos hacer desde el fichero `/etc/hosts`

[49] Los directorios que aparecen en el ejemplo se tienen que crear previamente y hay que adecuarlos a nuestra configuración personal.

Si usamos

```
NameVirtualHost *
<VirtualHost *>
....
```

Apache escuchará en todos los interfaces de red que tengamos en la máquina.

```
        CustomLog /var/log/httpd/www-access.log common
</VirtualHost>
$cat /etc/apache2/sites-avalaible/tux
<VirtualHost tux.midominio.com>
        ServerAdmin webmaster1@midominio.com
        DocumentRoot /var/www/htmltux
        ServerName tux.midominio.com
        ScriptAlias /cgi-bin/ /var/www/tux/cgi-bin
        Alias /images /var/www/tux/images
        ErrorLog /var/log/httpd/tux-error.log
        CustomLog /var/log/httpd/tux-access.log common
</VirtualHost>
```

Y activarlos con

```
#a2ensites www
#a2ensites tux
```

3. Reiniciar el servidor para que lea los cambios realizados en el fichero de configuración.

Con la directiva `NameVirtualHost 172.26.0.2` le decimos a Apache que las peticiones para la dirección IP sean subdivididas por nombre. Opcionalmente puede añadirse también un puerto.

La sección `<VirtualHost>` estará identificada por la dirección IP del sitio que queremos que sirva Apache. La dirección IP tiene que ser la misma que la dirección definida en `NameVirtualHost`. El efecto de esto es que cuando Apache recibe una petición dirigida al nombre del `host`, comprueba los bloques `<VirtualHost>` que tienen la misma dirección IP que la declarada con la directiva `NameVirtualHost`, para encontrar uno que incluya el nombre de servidor (indicado en `ServerName`) que se ha solicitado. En caso que no usemos `NameVirtualHost`, Apache buscará un bloque `<VirtualHost>` con la dirección IP correcta y usa el `ServerName` en la respuesta.

En cada uno de los bloques modificamos la dirección de correo del administrador y el `DocumentRoot` para el *host* virtual especificado en `ServerName`. Además, optamos por variar el directorio que contendrá los scripts CGI (`ScriptAlias /cgi-bin/ /var/www/tux/cgi-bin`) y directorios de imágenes (`Alias /images /var/www/tux/images`). Optamos también por dividir los ficheros para almacenar la salida de errores y log, ya que esto nos puede ayudar a la hora de comprobar el comportamiento por separado de los distintos servidores virtuales.

Con la configuración anterior, cualquier acceso con un navegador web a la dirección IP o un nombre distinto de los definidos en el bloque `<VirtualHost>` dará como resultado que Apache nos dé acceso al primero de los servidores virtuales definidos.

Hay que destacar que las posibilidades que ofrece Apache para el tratamiento son amplias, permitiendo la combinación de las opciones basadas en nombre e IP en una misma instancia de Apache. ■

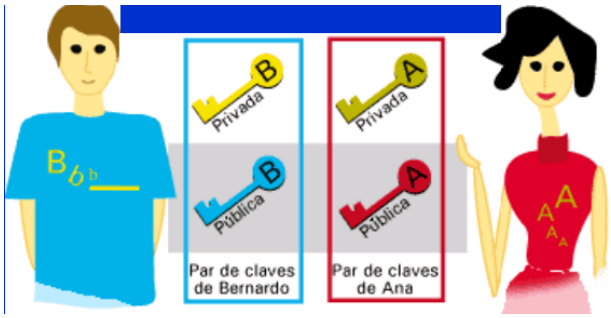
1.7. Servidores Seguros

Muy someramente, *https* se basa en los dos tipos de criptografía que conocimos en la entrega anterior: criptografía simétrica y criptografía asimétrica. ¿Porqué utilizar las dos? Porque aprovecha las ventajas de cada una y evita sus inconvenientes.

La criptografía asimétrica[50] es muy buena para la autenticación (ya que cada usuario protege su clave secreta), pero es muy lenta para el cifrado. Sin embargo, la criptografía simétrica es muy rápida en el cifrado y mala para la gestión de claves.

En el ejemplo, nuestros amigos Bernardo y Ana tienen su pareja de claves asimétricas (pública y privada), debiendo custodiar su clave privada para que nadie pueda conocerla.

[50] En la criptografía asimétrica, existen dos claves: una privada y otra pública. La clave privada debe permanecer bajo el exclusivo control del propietario y la pública puede (y debe) ser conocida por todos.



Así, el esquema utilizado por SSL, basado en el intercambio de claves Diffie-Hellman, quedaría como sigue: se utiliza la criptografía de clave pública o asimétrica, para realizar el intercambio seguro de una clave simétrica. No importa que el cifrado asimétrico sea lento, porque sólo se intercambia una clave, que es muy poca información. Esta clave simétrica, que es rápida para el cifrado, es la que se utilizará para cifrar los datos transmitidos en la sesión.

Un certificado digital contiene la clave pública, a la que se le añaden una serie de datos identificativos[51] y todo ello firmado por alguien en quien el resto de usuarios confían, denominado Autoridad Certificadora (CA). Esta tercera parte de confianza es la que permite que personas que no se conocen entre sí, puedan confiar en los certificados que se presentan el uno al otro. Por ejemplo, uno muy conocido puede ser la Fábrica Nacional de Moneda y Timbre[52].

La Autoridad Certificadora (AC[53]) lo que hace es firmar los certificados que emite. Un certificado contiene el nombre de la AC, el nombre del usuario, la clave pública del usuario y cualquier otra información, como puede ser un indicador de tiempo de validez. El certificado se firma con la clave privada de la AC. Todos los usuarios poseen la clave pública de la AC. Esto permite que cualquiera pueda comprobar la validez del certificado y si éste ha sido modificado.

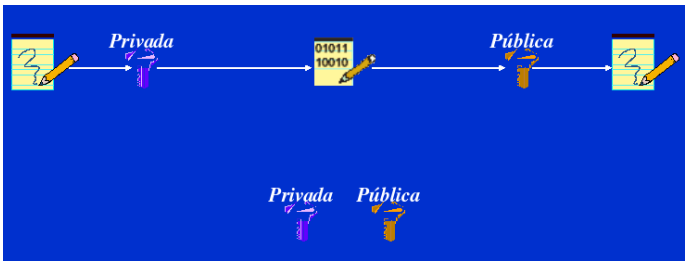
El proceso de firma electrónica se basa en lo siguiente:

1. Se cifra el mensaje[54] con la clave privada del remitente (la persona que firma).

2. Para comprobar la firma basta con descifrar con la clave pública del remitente, que es conocida. Si coincide, podemos asegurar dos cosas: que el mensaje no ha sido modificado y que

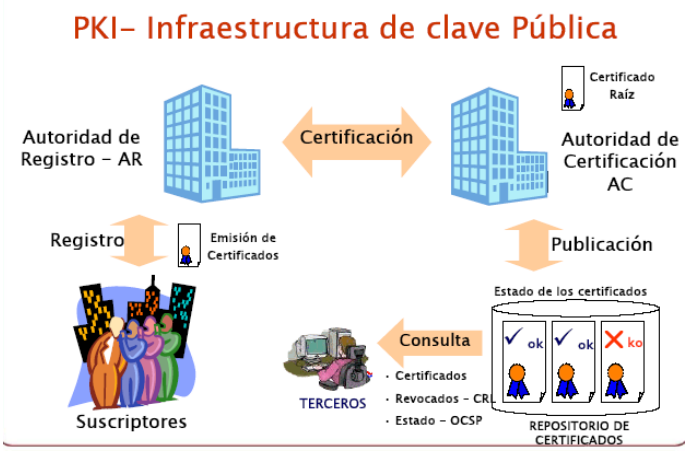
[51] Para un servidor pueden ser su nombre DNS (www.midominio.com). Para una persona física pueden ser su nombre, apellidos y DNI.
[52] Si confiamos en los billetes que hace, ¿no vamos a confiar en sus certificados...?
[53] O más común, CA, de Certification Authority
[54] Normalmente no es el mensaje lo que se cifra, sino una huella (o hash) del mismo.

la única persona que lo ha podido firmar ha sido el remitente, ya que es el único que posee la clave privada que se corresponde con la clave pública con que se ha descifrado.



Podemos solicitar un certificado a una Autoridad Certificadora, que nos puede llevar un dinero por ello, pero mediante openssl podemos erigirnos en nuestra propia Autoridad Certificadora[55], transmitir la información cifrada y sin gastarnos un solo euro.

Una infraestructura de clave pública (PKI[56]) es compleja. A continuación presentamos un gráfico con todos los elementos que pueden componerla.



No necesitaremos todos los elementos para montar nuestra PKI casera, aunque OpenSSL nos ofrece toda las funciones necesarias. Vayamos al grano.

Si queremos que nuestro servidor funcione con el protocolo SSL (https) para el envío de información cifrada, tenemos que instalar los paquetes `mod_ssl` y `openssl`.

Instalémoslos:

Fedora

```
# apt-get install mod_ssl openssl
Leyendo listas de paquetes... Done
Construyendo árbol de dependencias... Done
openssl is already the newest version.
Se instalarán los siguientes paquetes extras:
  distcache
Se instalarán los paquetes NUEVOS siguientes:
  distcache mod_ssl
0 upgraded, 2 newly installed, 0 removed and 170 not upgraded.
Need to get 205kB of archives.
After unpacking 397kB of additional disk space will be used.
¿Quiere continuar? [S/n]
```

[55] Claro está, que solamente nuestros amigos y familiares confiarán en nuestra Autoridad Certificadora.
[56] De *Public Key Infraestructure*

El fichero de configuración para el servidor en modo SSL se encuentra en `/etc/httpd/conf.d/ssl.conf`. Para un modo de trabajo normal, no es necesario modificarlo. Solamente personalizamos, por ejemplo, dónde se encuentran las páginas a mostrar (**DocumentRoot** y **Directory**).

Bajo el directorio `/etc/httpd/conf/` nos encontramos el directorio `ssl.key` con el fichero `server.key`. Este fichero contiene las claves pública y privada para nuestro servidor. En el directorio `ssl.crt` aparece el fichero `server.crt` que contiene el certificado (clave pública más los datos identificativos del servidor, firmados por la CA). Si queremos construir nuestro propio certificado seguiremos los siguientes pasos. Nos situamos en el directorio `/etc/httpd/conf`. Modificamos el fichero `server.key`, por ejemplo con

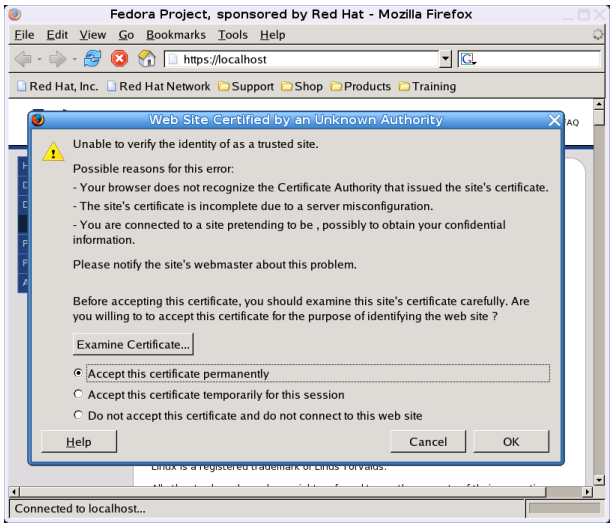
```
#touch ssl.key/server.key
```

y a continuación

```
# make testcert
umask 77 ; \
/usr/bin/openssl genrsa -des3 1024 > /etc/httpd/conf/ssl.key/server.key
Generating RSA private key, 1024 bit long modulus
...........++++++
...........++++++
e is 65537 (0x10001)
Enter pass phrase:
Verifying - Enter pass phrase:
umask 77 ; \
/usr/bin/openssl req -new -key /etc/httpd/conf/ssl.key/server.key -x509 -
    days 365 -out /etc/httpd/conf/ssl.crt/server.crt
Enter pass phrase for /etc/httpd/conf/ssl.key/server.key:
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [GB]:
State or Province Name (full name) [Berkshire]:
Locality Name (eg, city) [Newbury]:
Organization Name (eg, company) [My Company Ltd]:
Organizational Unit Name (eg, section) []:
Common Name (eg, your name or your server's hostname) []:
Email Address []:
```

Donde podemos poner los datos que se ajusten a nuestro servidor.

Si reiniciamos apache, tendremos disponible la conexión por el puerto 443 mediante SSL. Podemos comprobarlo apuntando nuestro navegador a `https://localhost`.



Podemos conseguir que haya dos zonas en nuestro servidor: una en la que usaremos el servidor seguro (`/var/www/htmls`) y otra de acceso "no seguro" (`/var/www/html`). Para ello, sólo hay que modificar el fichero `/etc/httpd/conf.d/ssl.conf`, adecuando las líneas que siguen a nuestro objetivo

```
...
<VirtualHost _default_:443>
# General setup for the virtual host
DocumentRoot "/var/www/htmls"
ServerName www.midominio.org:443
ServerAdmin root@localhost
....
```

Debian

La configuración la vamos a realizar sobre el servidor web de ejemplo `www.midominio.com`. En primer lugar activemos el módulo `mod_ssl` con[57]

```
# a2enmod ssl
```

Situémonos en `/etc/apache2/ssl`, que estará vacío. Generamos las claves asimétricas y el certificado de la CA, y que en nuestro caso también van a ser del servidor SSL.

```
root@guadalinex:/etc/apache2/ssl# apache2-ssl-certificate
creating selfsigned certificate
replace it with one signed by a certification authority (CA)
enter your ServerName at the Common Name prompt
If you want your certificate to expire after x days call this programm
with -days x
Generating a 1024 bit RSA private key
.....................................................................++++++
.....................................++++++
writing new private key to '/etc/apache2/ssl/apache.pem'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
```

[57] Para desactivarlo usaremos:
`# a2dismod ssl`

```
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [GB]:ES
State or Province Name (full name) [Some-State]:Andalucia
Locality Name (eg, city) []:MiLocalidad
Organization Name (eg, company; recommended) []:Thales-CICA
Organizational Unit Name (eg, section) []:MiIES
server name (eg. ssl.domain.tld; required!!!) []:www.midominio.com
Email Address []:webmaster@midominio.com
```

Con ello hemos creado el fichero `apache.pem` que contiene las claves (RSA PRIVATE KEY) y el certificado (CERTIFICATE) pertenecientes al servidor `www.midominio.com`, firmado por nosotros mismos[58].

```
root@guadalinex:/etc/apache2/ssl# more apache.pem

  -----BEGIN RSA PRIVATE KEY-----
  MIICXAIBAAKBgQCpAwn3fz9mqI+7UmD+kWsuUiw948U8wA43RHE/b0BErWBWwBNV
  ........
  ........
  rLvUPW5CBkOmlEe29xupk8wc39X1fcKgYrAtrPeb9vk=
  -----END RSA PRIVATE KEY-----
  -----BEGIN CERTIFICATE-----
  MIICuzCCAiQCCQDTe7ZZdsCq7TANBgkqhkiG9w0BAQQFADCBoTELMAkGA1UEBhMC
  ..........
  ..........
  +2jF+jYy+wyGHO3MSF+CwRRve4DEA8aAemNZEL4/aA==
  -----END CERTIFICATE-----
```

Pasemos a configurar el servidor SSL. Como base tomaremos el fichero de ejemplo proporcionado por Debian, que se encuentra en `/usr/share/doc/apache2/examples/ssl.conf.gz`. Lo descomprimimos y copiamos en `/etc/apache2/sites-available`. A continuación listamos las líneas de ese fichero que hemos descomentado/modificado para dejarlo listo para funcionar:

```
root@guadalinex:/etc/apache2/sites-available# more ssl.conf
#
...
<VirtualHost www.midominio.com:443>
...
DocumentRoot "/var/www/htmls"
ServerName www.midominio.com:443
ServerAdmin you@midominio.com
ErrorLog /var/log/apache2/error443_log
TransferLog /var/log/apache2/access443_log
...
SSLCertificateFile /etc/apache2/ssl/apache.pem
...
SSLCertificateKeyFile /etc/apache2/ssl/apache.pem
...
SSLCertificateChainFile /etc/apache2/ssl/apache.pem
...
SSLCACertificatePath /etc/apache2/ssl
SSLCACertificateFile /etc/apache2/ssl/apache.pem
```

[58] Para "aligerar" la salida hemos puestos puntos supensivos en lo que serían las claves generadas.

```
...
SSLVerifyClient none
...
```

Para que esté listo y en funcionamiento, creamos el directorio `/var/www/htmls`, donde vamos a situar las páginas del servidor seguro. Al menos, situad un fichero `index.html` para que podáis probar.

Añadimos a los sitios disponibles con:

```
#a2ensite ssl.conf
```

Y reiniciamos apache para que obtenga la nueva configuración[59].

```
#/etc/init.d/apache2 restart
```

Sin más dilación, apuntamos nuestro navegador a `https://www.midominio.com`[60]

y ¡FUNCIONA!

¡Funcionó! ¡El Servidor Apache Seguro ha sido instalado con éxito!

El ajustar los valores a vuestro servidor Web se deja como ejercicio -:)

1.7.1. Autenticación del cliente mediante certificados

Lo visto hasta ahora, que no es poco, nos sirve para que las comunicaciones entre el servidor web y los navegadores se realice cifrada. Pero aún hay un paso más. El servidor se ha identificado mediante un certificado, pero podemos hacer que el cliente también lo haga.

Lo que tenemos que hacer es generar certificados para los clientes y poner la directiva SSLVerifyClient con el valor "require" (en vez del valor "none" que trae por defecto).

Adecuamos el entorno para que nuestra Autoridad Certificadora funcione correctamente (vamos a utilizar los parámetros que vienen por defecto en el fichero openssl.cnf)

Creamos los directorios y ficheros necesarios, basándonos en la estructura Debian.

```
#cd /etc/apache2/ssl
#mkdir demoCA
#mkdir demoCA/private
#mkdir demoCA/newcerts
#touch demoCA/index.txt
```

Copiar las claves de la CA al directorio correspondiente con el nombre cakey.pem

```
#cp  apache.pem demoCA/private/cakey.pem
```

Copiar el certificado de la CA al lugar necesario

```
#cp apache.pem demoCA/cacert.pem
```

[59] O con `#apache2ctl restart`

[60] Este nombre debe existir en el DNS o en nuestro fichero `/etc/hosts`.

Editar el fichero demoCA/serial y guardarlo en modo texto, tal que su contenido sea una línea con el valor 01, como se muestra a continuación.

```
#more demoCA/serial
01
```

Generaramos un Certificado para uso personal.

Generar clave privada

```
#openssl genrsa -des3 -out usuario.key 1024
Generating RSA private key, 1024 bit long modulus
............++++++
.....................................................................................++++++
e is 65537 (0x10001)
Enter pass phrase for usuario.key:
Verifying - Enter pass phrase for usuario.key:
```

Generar la petición de certificado en el formato estándar PKCS#10 para que la firme la Autoridad Certificadora. La extensión suele ser .csr, de Certificate Signing Request[61] y contiene la clave pública y los datos identificativos para que los firme la CA.

En el caso de una persona, los campos identificativos serán el Common Name (CN) con el nombre y DNI de la persona (esto es una convención utilizada, por ejemplo, por la Fábrica Nacional de Moneda y Timbre) y la dirección de correo electrónico, que servirá para enviar correos cifrados y firmados. En este ejemplo, la persona es Juan Perez Gomez con DNI. 29.999.999 y dirección de correo electrónico juan.perez@midominio.com.

El valor del campo Organization Name (Nombre de la Organización), debe coincidir con el de la Autoridad Certificadora. En nuestro ejemplo, Thales-CICA.

```
#openssl req -new -key usuario.key -out usuario.csr
Enter pass phrase for usuario.key:
You are about to be asked to enter information that will be incorpora-
ted
into your certificate request.
What you are about to enter is what is called a Distinguished Na-
me or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:ES
State or Province Name (full name) [Some-State]:Andalucia
Locality Name (eg, city) []:MiLocalidad
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Thales-CICA
Organizational Unit Name (eg, section) []:MiIES
Common Name (eg, YOUR name) []:Juan Perez Gomez DNI: 29.999.999
Email Address []:juan.perez@midominio.com
Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
```

La Autoridad Certificadora firma (avala) el certificado.

[61] Petición de Firma de Certificado

```
$openssl ca -in usuario.csr -out usuario.crt
Using configuration from /usr/lib/ssl/openssl.cnf
Enter pass phrase for ./demoCA/private/cakey.pem:
DEBUG[load_index]: unique_subject = "yes"
Check that the request matches the signature
Signature ok
Certificate Details:
Serial Number: 2 (0x2)
Validity
Not Before: Feb 26 20:48:49 2005 GMT
Not After : Feb 26 20:48:49 2006 GMT
Subject:
countryName = ES
stateOrProvinceName = Andalucia
organizationName = Thales-CICA
organizationalUnitName = MiIES
commonName = Juan Perez Gomez DNI: 29.999.999
emailAddress = juan.perez@midominio.com
X509v3 extensions:
X509v3 Basic Constraints:
CA:FALSE
Netscape Comment:
OpenSSL Generated Certificate
X509v3 Subject Key Identifier:
C1:30:57:F7:CB:DC:D2:F4:CB:AC:C8:EF:02:39:9C:0C:A4:A2:87:34
X509v3 Authority Key Identifier:
keyid:6C:A3:10:13:42:EB:77:CD:6D:28:A4:F5:E5:D4:6E:5C:DC:EC:1A:86
DirName:/C=ES/ST=Andalucia/L=MiLocalidad/O=Thales-
CICA/OU=MiIES/CN=CA CEP/emailAddress=webmaster@midominio.com
serial:85:2D:C3:B5:5E:95:44:5E
Certificate is to be certified un-
til Feb 26 20:48:49 2006 GMT (365 days)
Sign the certificate? [y/n]:y
1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
```

El siguiente paso es exportar el certificado y las claves a formato PKCS#12. El fichero PKCS#12 contiene las claves asimétricas (privada y pública) y el certificado (con la clave pública y los datos del usuario, ya firmados por la autoridad certificadora).

```
$openssl pkcs12 -export -in usuario.crt -inkey usuario.key -
out usuario.p12
Enter pass phrase for usuario.key:
Enter Export Password:
Verifying - Enter Export Password:
```

El fichero PKCS#12 irá protegido por una contraseña. Podemos grabarlo en un soporte e importarlo desde un navegador,cliente de correo..., para ser utilizado. Debemos guardarlo bien, porque contiene nuestra clave privada.

Una vez importado en el navegador, nos conectamos a nuestro servidor que nos requerirá identificarnos con un certificado de cliente válido. Y si no, no nos permitirá entrar.

1.8. Reescribir las URL

Apache tiene mucha más funcionalidad que la hasta ahora comentada: SSI, proxy, ... Además permite reescribir las URL[62]. De esta forma, podemos redirigir una llamada al servidor a otro archivo distinto o a una URL diferente. Para hacer esto, es necesario usar las reglas de reescritura de URLs. Su estudio se escapa de lo que pretendemos en esta entrega, así que sólo vamos a analizar una de las múltiples posibilidades que presentan, usando un problema de ejemplo.

➥ **Problema:** en nuestro centro de enseñanza tenemos un dominio contratado de nombre `micentro.org`. Deseamos que los departamentos de matemáticas y lengua accedan a su Web de centro con una URL de la forma `www.matematicas.micentro.org` y `www.lengua.micentro.org`. Además, para facilitar la gestión de ambos sitios, deseamos que los `$HOME` de usuario coincidan con los directorios usados por Apache para servir las páginas Web[63]

Para resolver el problema debemos:

1. Añadir los registros correpondientes en nuestro DNS usando un registro CNAME (alias) para cada departamento didáctico.

2. Crear los `$HOME` de usuario adecuados para esos departamentos didácticos. Por ejemplo:

```
#mkdir /var/www/matematicas
#chown matematicas.matematicas /var/www/matematicas
#chmod 711 /var/www/matematicas
```

3. Añadir los usuarios al sistema y modificar el fichero `/etc/passwd` para que el `$HOME` de usuario de matemáticas sea el directorio antes creado.

```
#adduser matematicas
$cat /etc/passwd
  ...
matematicas:x:507:511::/var/www/matematicas:/bin/bash
```

4. Añadiremos las reglas de reescritura que siguen a nuestro fichero de configuración de Apache:

```
# por defecto está en Off, con esta directiva se
# activan las reglas de reescritura
RewriteEngine On
# Comprobamos si la variable de entorno HTTP-HOST es del tipo deseado
RewriteCond%{HTTP_HOST} ^www\.[^.]+\.micentro\.org
RewriteRule ^(.+)%{HTTP_POST}$1
RewriteRule ^www\.([^.]+)\.micentro\.com(.*) /var/www/$1/$2
```

Lo que faltaba: ¡expresiones regulares!. Sólo vamos a comentar un poco qué se hace para que no parezca un texto de Mortadelo y Filemón (`¡Arg#'¡@·`).

Con la primera regla comprobamos que la variable de `HTTP_HOST` es del tipo deseado

(`www.departamento.micentro.org`), después, encadenamos dos reglas (`C`) cuya misión es detectar las variables que se van a sustituir. En expresiones regulares la primera cadena de referencia (`$1`) se corresponde con la cadena encontrada en el primer paréntesis[64] `(^[.]+)` y `$2` con el segundo paréntesis[65] `(.*)`[66]

[62] El módulo encargado de esta labor es `mod_rewrite`.
En Guadalinex hemos de activarlo con

```
#a2enmod rewrite
```

y reiniciar el servidor.

[63] Notar que este problema se puede resolver fácilmente con `host virtuales`.

[64] En el ejemplo se corresponde con "matematicas"

[65] Si nuestra URL es de la forma `http://www.matematicas.micentro.org/algebra`, se corresponde con "algebra"

[66] El carácter punto (`.`) sustituye a cualquier carácter excepto el fin de línea. El asterisco (`*`) significa cero o más repeticiones de la expresión regular de que se trate y el símbolo de más (`+`), significa repeticiones de una o más veces de la expresión regular. Luego `.*` significa cualquier cadena de caracteres hasta el fin de línea.

Cuando desde un navegador Web escribamos `http://www.matematicas.micentro.org` se obtendría la página deseada.

No os asustéis, esto es solamente un ejemplo de la potencia que podéis llegar a tener, pero para la gran mayoría de los casos no tendréis que utilizarla. Tenéis un coche de 230 caballos, pero por ciudad debéis ir a 50 km/h. ■

1.9. Loganalizadores[67]

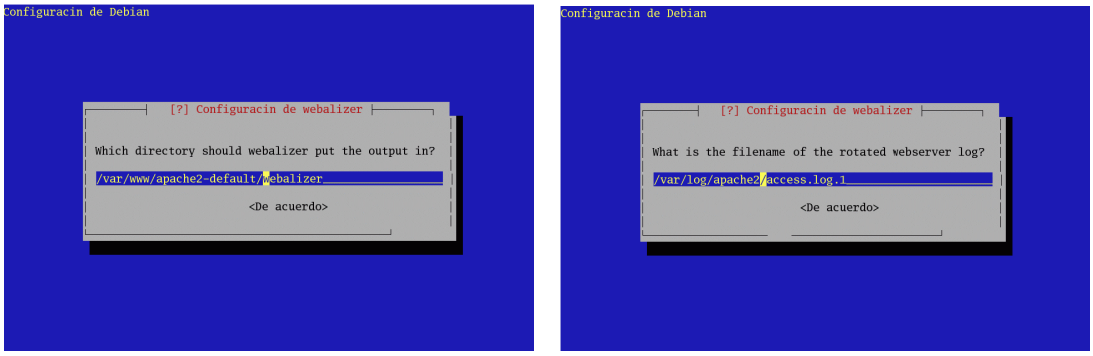
Todo servidor web que se precie tiene que disponer de estadísticas de uso para que el resto del mundo sepa la ingente cantidad de páginas web servidas. Vamos a analizar dos analizadores de accesos de Apache

1.9.1. webalizer

Se trata de un clásico (`http://www.mrunix.net/webalizer/`) . Si optamos por instalarlo desde internet[68] usaríamos:

```
# apt-get install webalizer
```

para ambas versiones de GNUlinux. En la instalación para Debian hemos de responder a un par de cuestiones que no presentan mayor dificultad (salvo adecuar el directorio DocumentRoot de Debian y la ruta del fichero de *log* de Apache que es `/var/log/apache2/access.log.1`). Ambas cuestiones se pueden modificar después.

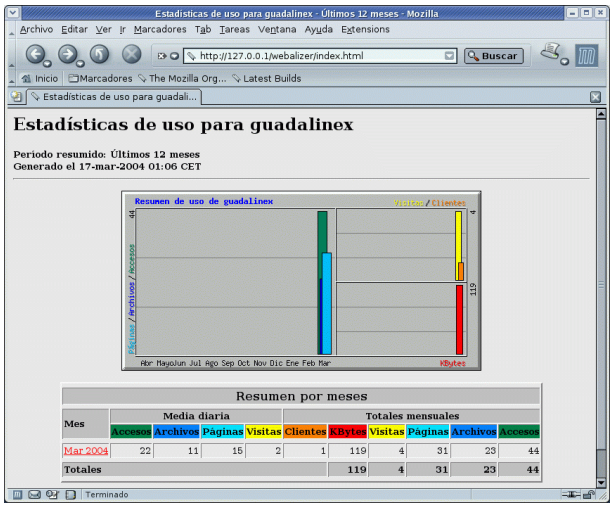


Con

```
#webalizer
```

generaremos las estadísticas que obtiene el programa. En ambos sistemas añade una entrada a `/etc/cron.daily` y, por tanto, nuestras estadísticas de uso se actualizarán a diario.

[67] O analizadores de accesos
[68] En Fedora lo tenemos en los CDs: `webalizer-2.01_20-25.i386.rpm`



La configuración por defecto para acceder a ellas desde Fedora es `http://localhost/usage`.

El fichero de configuración de Webalizer es `/etc/webalizer.conf` y no presenta dificultad. En general, no es necesario cambiarlo. Si deseamos modificar los parámetros configurados en la instalación de Webalizer para Debian, lo haremos con las directivas:

```
LogFile       /var/log/apache2/acces_log
OutputDir     /var/www/html/usage
ReporTitle    Usage Statistic for
```

adecuándolas a nuestro sistema.

➥ **Para practicar:** Webalizer en castellano con Fedora

Vamos a "construirnos" la versión castellanizada de Webalizer. Para conseguirlo:

1. Nos bajamos el fichero fuente de la aplicación:

   a) Descomentamos la línea

   ```
   rpm-
   src http://ayo.freshrpms.net fedora/linux/3/i386 core updates freshrpms
   ```

   del fichero `/etc/apt/source.list`

   b) Nos bajamos el fichero fuente y lo desempaquetamos

   ```
   # apt-get update; apt-get source webalizer
   # rpm -ivh webalizer-2.01_10-25.src.rpm
   ```

2. Modifiquemos el fichero `/usr/src/redhat/SPEC/webalizer.spec`, en la sección `%configure` añadamos `--with-language=spanish`. Quedaría:

   ```
   %configure --enable-dns --with-dblib=/lib --with-language=spanish
   ```

3. Creemos el nuevo paquete:

   ```
   # rpmbuild -bb /usr/src/redhat/SPEC/webalizer.spec
   ```

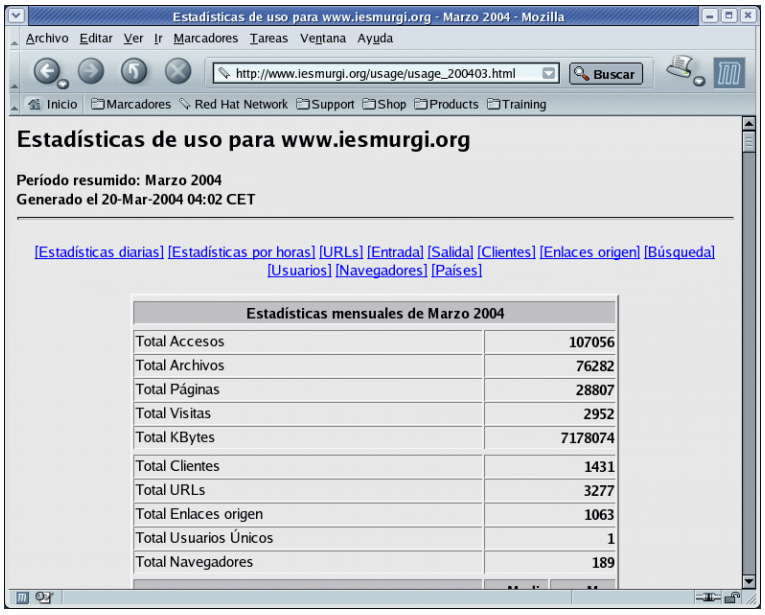
4. Lo instalamos[69]

[69] Si ya habíamos instalado el de Fedora habrá que eliminarlo

```
#rpm -e webalizer
```

```
#rpm -Uvh /usr/src/redhat/RPMS/i386/webalizer-2.01_10-25.i386.rpm
```

Como muestra:



1.9.2. awstats

Se trata de un loganalizador para Apache `http://www.awstats.org/` que se puede configurar para que nos muestre estadísticas para nuestro agente de transporte de correo[70] (MTA, como postfix, sendmail, etc.)[71]. Es más completo que el anterior.

Para Fedora.

Nos bajamos el paquete
`http://prdownloads.sourceforge.net/awstats/awstats-6.4-1.noarch.rpm`
y lo instalamos con

```
# rpm -ivh awstats-6.4-1.noarch.rpm
Preparing...                ########################################### [100%]
   1:awstats                ########################################### [100%]

     ----- AWStats 6.4 - Laurent Destailleur -----
     AWStats files have been installed in /usr/local/awstats

     If first install, follow instructions in documentation
     (/usr/local/awstats/docs/index.html) to setup AWStats in 3 steps:
     Step 1 : Install and Setup with awstats_configure.pl
     Step 2 : Build/Update Statistics with awstats.pl
     Step 3 : Read Statistics
```

[70] Más sobre esto un poco más adelante.
[71] Véase `http://cvs.sourceforge.net/viewcvs.py/awstats/awstats/docs/awstats_faq.html?rev=1.52`

Configuración. Nos situamos en el lugar adecuado

```
cd /usr/local/awstats
```

y ejecutemos el script de Perl[72]

```
# tools/awstats_configure.pl

----- AWStats configure 1.0 (build 1.4) (c) Laurent Destailleur -----
This tool will help you to configure AWStats to analyze statistics for
one web server. If you need to analyze load balanced servers log files, to
analyze downloaded log files without web server, to analyze mail or ftp log
files, or need to manage rotated logs, you will have to complete the config
file manually according to your needs.
Read the AWStats documentation (docs/index.html).

-----> Running OS detected: linux

-----> Check for web server install
configure did not find your Apache web server path.

Enter full config file path of you web server.
Example: /etc/httpd/apache.conf
Example: d:\Program files\apache group\apache\conf\httpd.conf
Config file path (CTRL+C to cancel):
> /etc/httpd/conf/httpd.conf

-----> Check and complete web server config file '/etc/httpd/conf/httpd.conf'
  AWStats directives already present.

-----> Update model config file '/etc/awstats/awstats.model.conf'
  File awstats.model.conf updated.

-----> Need to create a new config file ?
Do you want me to build a new AWStats config/profile
file (required if first install) [y/N] ? y

-----> Define config file name to create
What is the name of your web site or profile analysis ?
Example: www.mysite.com
Example: demo
Your web site, virtual server or profile name:
> midominio.com

Directory path to store config file(s) (Enter for default):
> /etc/awstats/awstats.midominio.com.conf
-----> Create config file '/etc/awstats/aws-
tats.midominio.com.conf/awstats.picasa.org.conf'
 Config file /etc/awstats/aws-
tats.midominio.com.conf/awstats.picasa.org.conf created.
-----> Restart Web server with '/sbin/service httpd restart'
Parando httpd:                                             [  OK  ]
Iniciando httpd:                                           [  OK  ]
-----> Add update process inside a scheduler
Sorry, configure.pl does not support automatic add to cron yet.
You can do it manually by adding the following command to your cron:
```

[72]Las líneas de tamaño mayor son las que tenemos que adecuar a nuestro sistema

```
/usr/local/awstats/wwwroot/cgi-bin/awstats -update -config=midominio.com
Or if you have several config files and prefer having only one command:
/usr/local/awstats/tools/awstats_updateall.pl now
Press ENTER to continue...

A SIMPLE config file has been created: /etc/awstats/awstats.midominio.com.conf
You should have a look inside to check and change manually main parameters.
You can then manually update your statistics for 'midominio.com' with command:
> perl awstats.pl -update -config=midominio.com
You can also read your statistics for 'midominio.com' with URL:
> http://localhost/awstats/awstats.pl?config=midominio.com

Press ENTER to finish...
```

Los únicos valores que hemos introducido se corresponden con: la ruta del fichero de configuración del servidor apache y el nombre del dominio.

Se creará el fichero de configuración para este dominio en `/etc/awstats/awstats.midominio.com.conf`. Tenemos que ajustarle el nombre del fichero de log de apache

```
LogFile="/var/log/httpd/acces_log"
```

y crear el directorio (de dueño apache) en donde se almacenarán los datos de los distintos dominios

```
# mkdir /var/lib/awstats
# chown apache /var/lib/awstats/
```

En la parte final de la salida del script aparece información relevante sobre la forma de activar el programa:

- Para que se actualicen las estadísticas usando `cron`, añadiremos (si deseamos que se actualicen cada hora[73]):

  ```
  # cat /etc/cron.hourly/awstats
  /usr/local/awstats/wwwroot/cgi-bin/awstats.pl -update -
  config=midominio.com
  ```

- Si tenemos distintos ficheros de configuración y preferimos tener sólo un comando:

  ```
  /usr/local/awstats/tools/awstats_updateall.pl now
  ```

- Para actualizar de forma manual las estadísticas de `midominio.com` escribiremos:

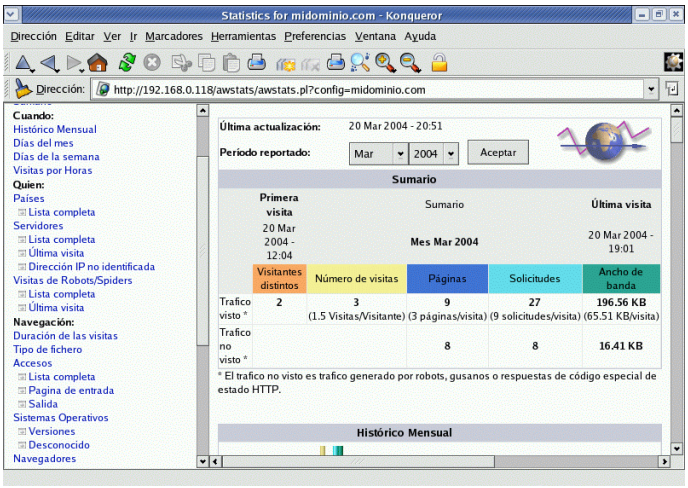
  ```
  perl /usr/local/awstats/wwwroot/cgi-bin/awstats.pl -update -
  config=midominio.com
  ```

- Podemos acceder a la Web de estadísticas de `midominio.com` usando la URL:

  ```
  http://localhost/awstats/awstats.pl?config=midominio.com
  ```

Para acceder a las estadísticas escribiremos la URL antes comentada:

[73] Los permisos han de ser adecuados.



Para Debian:

Para poder trabajar con él hay que adecuar nuestro Apache para que permita trabajar con scripts cgi:

```
#a2enmod cgi; apache2ctl graceful
```

La instalamos con

```
# apt-get update awstats
```

Crea automáticamente una entrada en `/etc/cron.d` que tendremos que ajustar a nuestro sistema:

```
# cat /etc/cron.d/awstats
0,10,20,30,40,50 * * * * www-data [ -x /usr/lib/cgi-bin/awstats.pl
-a -f /etc/awstats/awstats.conf -a -r /var/log/apache/access.log ]
&& /usr/lib/cgi-bin/awstats.pl -config=awstats -update >/dev/null
```

El cambio en el fichero anterior consiste en sustituir el nombre en donde se almacenan los log de apache (`/var/log/apache2/access.log`).

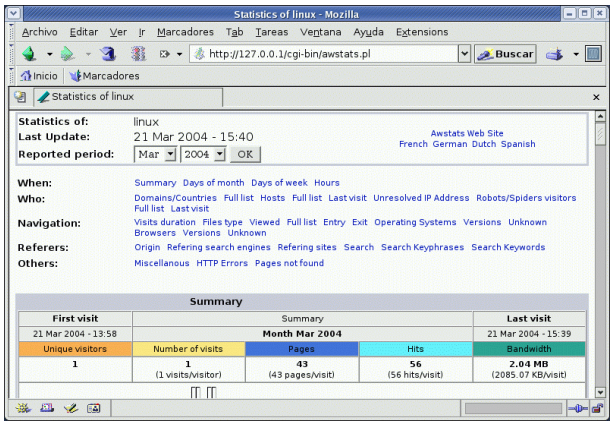
Nos crea un fichero de configuración (`/etc/awstats/awstats.conf`) y disponemos otro de ejemplo[74] `/usr/share/doc/awstats/examples/awstats.model.conf.gz`.

En `awstats.conf` hemos de modificar al menos las directivas:

```
LogFile="/var/log/apache2/acces.log"
SiteDomain="midominio.com"
```

Una vez que todo está como debe, podemos acceder a las estadísticas generadas por el programa escribiendo `http://www.midominio.com/cgi-bin/awstats.pl`

[74] En ese mismo directorio están algunas de las herramientas antes comentadas para Fedora. Por ejemplo: `awstats-update` que usaremos si deseamos actualizar de forma manual las estadísticas.


Statistics of linux - Mozilla
Archivo Editar Ver Ir Marcadores Tab Tareas Ventana Ayuda Extensions
http://127.0.0.1/cgi-bin/awstats.pl
Buscar
Inicio Marcadores
Statistics of linux
Statistics of: linux
Last Update: 21 Mar 2004 - 15:40
Reported period: Mar 2004 OK
Awstats Web Site
French German Dutch Spanish
When: Summary Days of month Days of week Hours
Who: Domains/Countries Full list Hosts Full list Last visit Unresolved IP Address Robots/Spiders visitors Full list Last visit
Navigation: Visits duration Files type Viewed Full list Entry Exit Operating Systems Versions Unknown Browsers Versions Unknown
Referers: Origin Refering search engines Refering sites Search Search Keyphrases Search Keywords
Others: Miscellanous HTTP Errors Pages not found
Summary
First visit
Summary
Last visit
21 Mar 2004 - 13:58
Month Mar 2004
21 Mar 2004 - 15:39
Unique visitors
Number of visits
Pages
Hits
Bandwidth
1
1 (1 visits/visitor)
43 (43 pages/visit)
56 (56 hits/visit)
2.04 MB (2085.07 KB/visit)

Capítulo 2

Correo electrónico

> Asegurarse de que el correo electrónico de los usuarios se envía y recibe correctamente, es uno de los trabajos más importantes de un administrador de sistemas, y que se hace extremadamente visible en caso de que las cosas vayan mal. *Administering E-mail.* AELEEN FRISCH

2.1. Introducción

Reconocido como la aplicación más utilizada de Internet, junto con la todopoderosa Web, el correo electrónico es utilizado para prácticamente cualquier tipo de comunicación. Cada vez más, las empresas y organizaciones dependen de su buen funcionamiento para las relaciones entre sus empleados y con el exterior. Paraos a pensar cuántos mensajes mandáis y recibís a lo largo del día en el trabajo, en casa, desde un cibercafé... Y si se combinan con los SMS a móviles, realmente nos damos cuenta de que estamos un nuevo tipo de sociedad.

Una de los factores que han llevado al éxito al correo electrónico es su simplicidad de uso. Cualquier persona[1] con unas breves nociones de acceso al sistema operativo y de uso del programa de correo electrónico, rápidamente es capaz de enviar y recibir correos con una facilidad pasmosa. Sin embargo, no son tan conocidos los mecanismos que hacen que los mensajes lleguen a través de Internet al destinatario que se encuentra a cientos o miles de kilómetros de distancia. ¿Qué son SMTP, POP o IMAP, los agentes de transporte y los agentes de usuario? Enseguida lo sabremos.

Seguramente hoy habrás recibido varios, o puede que muchos, mensajes de correo electrónico. Para aceder a ellos y verlos, utilizas algún cliente de correo como Outlook, Eudora, Mozilla o Ximian Evolution. También puedes acceder por medio de un navegador de Internet, lo que se conoce como webmail.

Sin embargo, sea del tipo que sea el cliente, como mínimo hará las siguientes cosas:

* Presentar una lista de los mensajes que han llegado a tu dirección de correo electrónico, mostrando una cabecera que dice quién envió el mensaje, el motivo del mensaje y la fecha y hora.

* Permite seleccionar las cabeceras de los mensajes y leer el cuerpo del mensaje seleccionado.

* Permite componer nuevos mensajes y enviarlos. Indicamos al menos la dirección de correo del destinatario, el tema del mensaje y el contenido o cuerpo del mensaje.

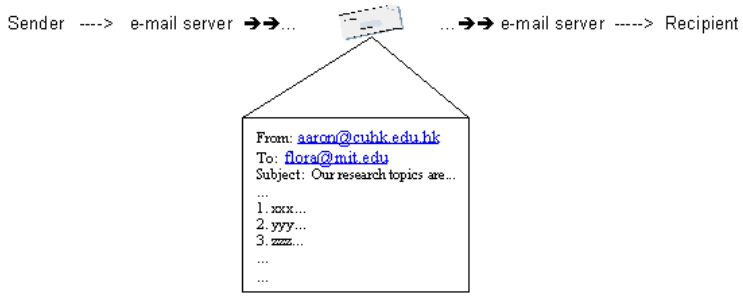
* Permite adjuntar ficheros a los mensajes y recuperar los ficheros adjuntos a los mensajes recibidos.

El formato de un mensaje de correo electrónico de Internet es simple: varios atributos obligatorios y otros opcionales, que forman una *cabecera*, separados por una línea en blanco del *cuerpo* del mensaje, que constituyen los datos objeto de la comunicación. Este formato viene definido en el RFC 2822[2]. La cabecera, al igual que en un sobre de correo postal, tiene toda la información

[1] Y si es joven, aún más.

[2] Este RFC ha sustituido al famoso RFC 822, que podemos encontrar en muchas referencias. RFC-2822 `http://www.faqs.org/rfcs/rfc2822.html`

que necesita el correo para llegar a su destino, o para su devolución al remitente en caso de que no haya sido posible su entrega.



Si en un sistema Linux enviamos un mensaje[3] y vemos el buzón donde se guardan los correos de un determinado usuario[4], podemos observar algo como lo siguiente:

```
From josber@midominio.com Sun Mar 21 22:44:17 2004
Return-Path: <josber@midominio.com>
Received: from greco.midominio.es (greco.midominio.com [195.123.25.23])
by mileto.cica.es (8.12.8/8.12.5) with ESMTP id i2LLiHJY006254
for <jabernal@mileto.cica.es>; Sun, 21 Mar 2004 22:44:17 +0100
Received: (from nobody@localhost)
by greco.midominio.com (8.8.5/8.8.5) id XAA11277
for <jabernal@mileto.cica.es>; Sun, 21 Mar 2004 23:03:40 +0100
From: josber@midominio.com
Message-Id: <200403212203.XAA11277@greco.midominio.com>
X-Authentication-Warning: greco.midominio.com: nobody set sender
          to <josber@midominio.com> using -f
Received: from alberti.midominio.com by greco.midominio.com via
          smap (V2.1/2.1+anti-relay+anti-spam)
id xma011274; Sun, 21 Mar 04 23:03:31 +0100
To: jabernal@mileto.cica.es
Subject: prueba
Date: Sun, 21 Mar 2004 21:44:26 GMT
X-Mailer: Endymion MailMan Standard Edition v3.0.35
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="MailMan_Boundary"

This is a multi-part message in MIME format.
--MailMan_Boundary
Content-Type: text/plain
Hola,
Te adjunto el fichero que solicitaste.
Un saludo.
--MailMan_Boundary
Content-Type: text/plain; name="hola.txt"
Content-Transfer-Encoding: base64
Content-Disposition: attachment; filename="hola.txt"
aG9sYQo=
--MailMan_Boundary--
```

[3]Cuando tengamos correctamente configurado nuestro sistema podremos practicar.

[4]Por ejemplo, en el directorio `/var/spool/mail`, y con el nombre de fichero de nuestro usuario. Atención: el superusuario podrá ver el correo de todos los usuarios, a menos que se envíe cifrado.

Si vamos desmenuzando el fichero, podemos ver que:

- La primera línea nos indica el inicio de este primer mensaje. Si hubiera varios mensajes en el buzón[5] del usuario, cada uno iría precedido por una línea de este tipo, con el remitente, la fecha y hora de recepción.
- El resto corresponde al mensaje, que se divide en dos partes, una cabecera y un cuerpo. La cabecera se compone de líneas con pares `campo:valor`. Tras una línea en blanco aparece el cuerpo del mensaje.
  - En la cabecera, el primer campo que aparece en nuestro ejemplo es el campo `Return-Path` que indica a dónde debe devolverse el mensaje en caso de que no se pueda entregar satisfactoriamente. En caso de que no exista este campo, se utilizaría el valor del campo `From`.
  - Cada campo `Received`, nos indica el periplo que ha realizado el mensaje hasta llegar a su destino. Por cada estafeta de correos[6] por las que pasa, se le añaden nuevas líneas que indican de dónde se ha recibido y a quién debe entregarla. En este caso concreto, vemos que el viaje del mensaje ha sido desde la máquina `alberti.midominio.com` hasta `greco.midominio.com` en un primer paso. De ahí llega a `mileto.cica.es` que es la máquina donde se encuentra su destino.
  - El campo `From` indica el remitente del correo. En este caso josber@midominio.com.
  - El campo `To` nos indica a quién va dirigido el mensaje. En este caso a jabernal@mileto.cica.es.
  - El campo `Subject` nos sirve para indicar brevemente el tema del mensaje.
  - El campo `MIME-Version` nos indica que el cuerpo se encuentra expresado en el formato MIME que se describe en las RFC siguientes: RFC2045, RFC2046 y RFC2049. Este formato nos permite enviar imágenes, ficheros binarios, de música... codificados de manera que pueden ser representados por caracteres ASCII que son los que son transportados por el correo electrónico. Normalmente en codificados en Base64
  - El campo `Content-Type` nos dice que el cuerpo se divide en varias partes separadas por la palabra `MailMan_Boundary`
- Tras la línea en blanco, viene el cuerpo del mensaje, que es el verdadero contenido. Éste consta de dos partes separadas por la palabra `MailMan_Boundary`:
  - La primera parte es un texto plano, que es lo que escribimos normalmente en el cuerpo del mensaje.
  - En la segunda parte, se nos indica que hay un fichero adjunto que se corresponde con un fichero de texto plano cuyo nombre será `hola.txt`. El fichero se ha mandado como adjunto[7] y está codificado para su transmisión. El contenido es `aG9sYQo=`, que corresponde a la codificación en base64 del contenido del fichero[8].

2.1.1. ¿Cuántos invitados tenemos para cenar?

El emisor de un mensaje de correo electrónico utiliza un programa para crear y enviar el correo. Este programa se denomina Agente de Usuario de Correo[9].

Una vez creado, el mensaje se traslada hacia el destinatario sobre un medio de transporte, que puede ser Internet o una red privada, utilizando uno o varios Agentes de Transporte de Correo[10].

[5] *Mailbox*: Nombre que recibe el fichero o estructura que guarda los mensajes de cada usuario.
[6] Que son los servidores con los agentes de transporte.
[7] *Attachment*, en inglés.
[8] El contenido era simplemente "hola".
[9] *Mail User Agent* (MUA)
[10] *Mail Transfer Agent* (MTA)

El mensaje al fin es entregado al destinatario usando un Agente de Entrega de Correo[11] y almacenado en su buzón[12] hasta que éste lo lee utilizando nuevamente un Agente de Usuario.

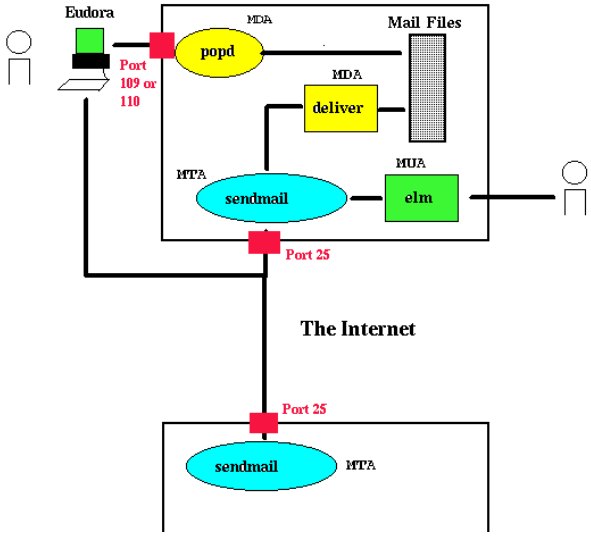
Así, los componentes del sistema de correo son:

***Mail User Agent* (MUA)** Un programa usado para crear y recibir mensajes de correo electrónico.

***Mail Transfer Agent* (MTA)** El medio por el cual los mensajes de correo electrónico se transfieren de máquina en máquina hasta llegar a su destino.

***Mail Delivery Agent* (MDA)** El programa que se encarga de depositar el mensaje de correo en el buzón del destinatario, una vez entregado por un MTA al servidor de correo.

En la siguiente figura podemos observar cómo interaccionan algunos Agentes de Usuario (MUA) en verde (Eudora y elm), Agentes de Entrega (MDA) en amarillo (popd y deliver) y Agentes de Transporte (MTA) en azul (sendmail).



Para cada una de las categorías de programas, tenemos varias alternativas, dependiendo de nuestras preferencias y de cuál sea nuestro sistema operativo. Incluso hay programas, que realizan más de una de las funciones. Como ejemplos de programas dentro de cada una de ellas, podemos citar:

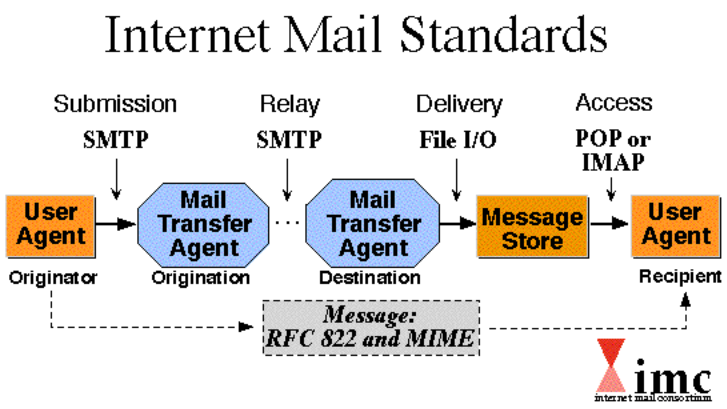
MUA Outlook, Eudora, Netscape Mail, Mozilla, Sylpheed, ximian-evolution, mail, elm, mutt, pine, mh/nmh.

MTA Sendmail, Postfix, smail, qmail, exim.

MDA mail, cyrus, deliver, mail.local, procmail, fetchmail.

El mecanismo funciona de la siguiente manera. El Agente de Usuario confecciona el mensaje y lo envía mediante el protocolo SMTP a un Agente de Transporte. Este Agente de Transporte lo envía a través de la red a otros Agentes de Transporte, hasta que al final llega al Agente de Transporte que corresponde al destinatario del mensaje. El Agente de Entrega[13] lo deposita en el buzón de correo electrónico del destinatario. Allí el mensaje espera hasta que el destinatario acceda con un Agente de Usuario mediante los protocolos POP o IMAP.

[11] *Mail Delivery Agent* (MDA)
[12] *Mailbox*, en inglés.
[13] Muchas veces este Agente de Entrega se encuentra integrado con el Agente de Transporte.



Podemos obervar cómo el contenido del mensaje llega desde el remitente al destinatario siguiendo los formatos RFC2822 y MIME.

2.1.2. ¿Cómo se encamina el correo?

Cuando un Agente de Transporte recibe el encargo de transportar un mensaje de correo electrónico, supongamos que a la dirección linux@cica.es, lo primero que hace es comprobar si es él mismo el encargado de manejar el correo para *el* dominio *cica.es*. Si lo es, no tiene que encargarle el trabajo a otro agente de transporte, sino que él mismo será capaz de entregarlo. En caso de que no lo sea, le pregunta al sistema DNS qué máquina o máquinas son las encargadas de ello. Los registros MX[14] son los que ofrecen esta información.

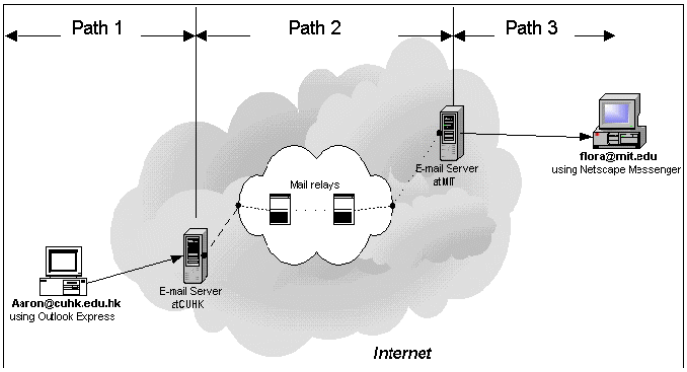
Por ejemplo, una consulta al DNS sobre los registros MX del dominio `cica.es` nos daría:

```
cica.es. 961 IN MX 15 mailgw2.cica.es.
cica.es. 961 IN MX 20 mail.rediris.es.
cica.es. 961 IN MX 10 mailgw.cica.es.
```

Veamos qué implicaciones tienen estos datos en el correo. Cada una de las líneas indica que el registro MX designa a una máquina que recibe correo para el dominio cica.es. De todas ellas, la preferente será la que tenga la prioridad más baja (el valor 10 será el preferido antes que el 15, y éste antes que el 20), y si no está disponible se irá al siguiente con menor prioridad. En este caso, si no hay ningún problema en la red o la máquina, `mailgw.cica.es` será la máquina que recibirá los correos para la dirección linux@cica.es.

En el caso de que no exista registro MX y el destino sea un host[15] (por ejemplo `mileto.cica.es`) también se le puede enviar correo electrónico a esa máquina concreta, como por ejemplo a la dirección linux@mileto.cica.es.

En la siguiente figura observamos que el correo puede pasar por varios Agentes de Transporte (MTA) hasta llegar a su destino.

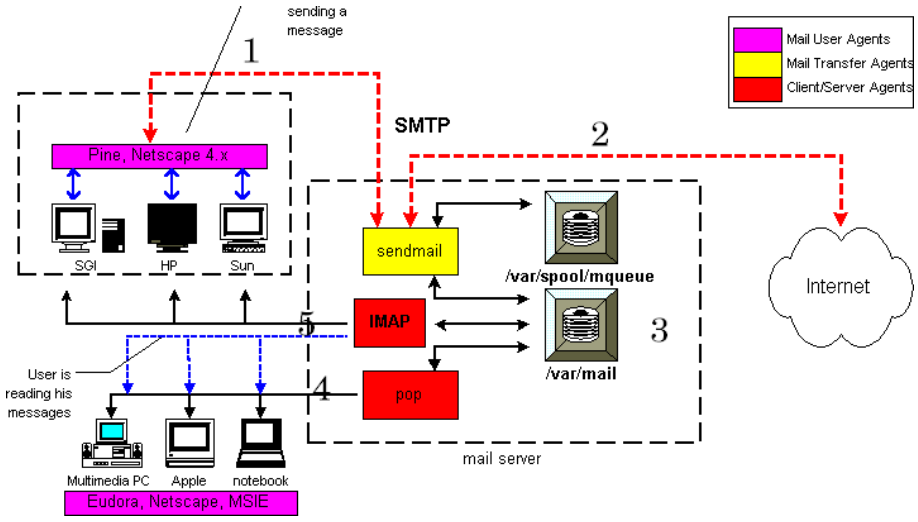


[14] *Mail eXchanger*

[15] Existiendo un registro tipo A en el DNS para él.

2.1.3. Eso no es todo, aún hay más

Descendamos a un plano más práctico y veamos qué programas y protocolos nos encontramos.



En la figura anterior, podemos ver los diferentes pasos en el proceso de envío y recepción del correo electrónico.

- En el paso 1, vemos que mediante un Agente de Usuario (pine, Netscape, Outlook...) componemos un mensaje y lo enviamos mediante el protocolo SMTP a un servidor de correo.
- En el paso 2, el Agente de Transporte debe mirar a qué otro Agente de Transporte debe enviarlo en caso de que él no sea el receptor.
- En el caso de que el correo para la dirección del destinatario lo gestione ese servidor de correo, en el paso 3 se guarda en el buzón correspondiente.
- Cuando el destinatario quiere leer su correo, lo hace bien mediante el protocolo POP (paso 4) o el protocolo IMAP (paso 5).

Protocolo SMTP

SMTP (*Simple Mail Transfer Protocol*[16]) es un protocolo cliente-servidor basado en TCP. Su funcionamiento es muy simple. Una vez que se establece la conexión, el cliente envía comandos al servidor con la cabecera y el cuerpo del mensaje.

Este protocolo se basa en el envío de comandos de cuatro caracteres y códigos de respuesta de tres dígitos, más una serie de comentarios que lo hacen más legible. Acualmente se utiliza una versión conocida como SMTP Extendido o ESMTP.

A continuación mostramos una conversación entre un cliente y un servidor SMTP. Con una fuente un poco mayor mostramos los comandos que vamos tecleando

```
[root@linux entrega04-3]# telnet localhost 25
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
220 linux.midominio.org ESMTP Send-
mail 8.12.10/8.12.10; Wed, 24 Mar 2004 21:30:4 8 +0100
EHLO linux.midominio.com
```

[16] Protocolo Simple de Transferencia de Correo

```
250-linux.midominio.com Hello localhost.localdomain [127.0.0.1], plea-
sed to meet you
250-ENHANCEDSTATUSCODES
250-PIPELINING
250-8BITMIME
250-SIZE
250-DSN
250-ETRN
250-AUTH DIGEST-MD5 CRAM-MD5 LOGIN PLAIN
250-DELIVERBY
250 HELP
MAIL From:<jabernal@linux.midominio.com>
250 2.1.0 <jabernal@linux.midominio.com>... Sender ok
RCPT To:<linux@mileto.cica.es>
250 2.1.5 <linux@mileto.cica.es>... Recipient ok
DATA
354 Enter mail, end with "." on a line by itself
saludos
.
250 2.0.0 i2OKUlgx002363 Message accepted for delivery
QUIT
221 2.0.0 linux.midominio.com closing connection
Connection closed by foreign host.
```

Vamos a actuar como lo haría un Agente de Usuario. Nos conectamos al puerto 25 de nuestra máquina local que es donde escucha el demonio SMTP por convención. La línea precedida por el código de respuesta 220 nos dice que el servidor es `linux.midominio.com`, que habla el protocolo ESMTP y que es Sendmail.

- Somos corteses y le mandamos el comando Hola[17]. Nos contesta con las capacidades que tiene el servidor.
- Con el comando `MAIL` le indicamos que vamos a enviar un mensaje e indicamos el remitente con el valor `From:`.
- El destinatario lo indicamos con el comando `RCPT` y con el valor del campo `To:`.
- Tras el comando `DATA` iniciamos el cuerpo del mensaje que acabaremos con una línea que empieza por punto y finalmente nos salimos con `QUIT`.

Como véis, SMTP no es complejo, pero tampoco es como para que estemos hablando SMTP con todo bicho viviente que nos encontremos. El cliente de correo lo hablará por nosotros y solamente tendremos que preocuparnos de rellenar los campos correspondientes a la información.

El protocolo SMTP, como habéis visto, es fácil de engañar. Hasta hace poco tiempo no se ha tomado en serio la seguridad y la consecuencia son los virus, spam, hoax y demás jungla. Afortunadamente, poco a poco va incorporando medidas de seguridad como autenticación, cifrado, certificados digitales, etc. en el correo.

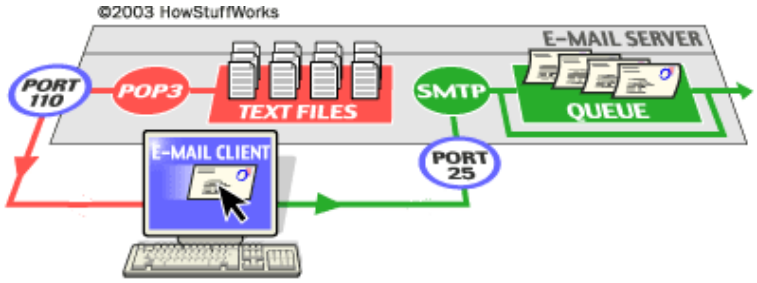
Protocolo POP

El protocolo POP (*Post Office Protocol*[18]) se diseñó para permitir una gestión del correo sin tener que estar conectados continuamente con el servidor. La idea es conectarse con el servidor, descargarse al ordenador local los correos electrónicos y poder trabajar con ellos sin necesidad de

[17] No, no es un error, es así EHLO
[18] Protocolo de Oficina de Correos

estar conectados con el servidor continuamente, ni siquiera conectados a la red[19]. Lo normal es que el correo al descargarlo, se borre del servidor, aunque hay opciones para conservarlo allí.

La siguiente figura es muy descriptiva de cómo el cliente de correo (MUA) envía al servidor (MTA) el correo al puerto 25 mediante el protocolo SMTP y lo recibe conectándose al puerto 110 mediante el protocolo POP.



Veamos un ejemplo de una sesión POP a "pelo[20]". Como con el protocolo SMTP, nuestros comandos los pondremos en una fuente mayor.

```
[root@linux entrega04-3]# telnet localhost 110
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
+OK POP3 localhost.localdomain v2003.83rh server ready
user juanjo
+OK User name accepted, password please
pass secreto
+OK Mailbox open, 6 messages
list
+OK Mailbox scan listing follows
1 615
2 687
3 842
4 913
5 1027
6 1109
.
retr 1
+OK 615 octets
Return-Path: <jabernal@linux.midominio.com>
Received: from linux.midominio.com (localhost.localdomain [127.0.0.1])
by linux.midominio.com (8.12.10/8.12.10) with ESMTP id i2NN73UL003368
for <jabernal@linux.midominio.com>; Wed, 24 Mar 2004 00:07:04 +0100
Received: (from jabernal@localhost)
by linux.midominio.com (8.12.10/8.12.10/Submit) id i2NN72dK003366
for jabernal; Wed, 24 Mar 2004 00:07:02 +0100
Date: Wed, 24 Mar 2004 00:07:02 +0100
From: jabernal@linux.midominio.com
Message-Id: <200403232307.i2NN72dK003366@linux.midominio.com>
To: jabernal@linux.midominio.com
Subject: sd
Status: RO
sd
.
```

[19] ¡Recuerdas aquellos tiempos en los que no había ADSL!.
[20] No le contéis a nadie que estáis hablando SMTP y POP. Puede que os tomen por locos.

```
quit
+OK Sayonara
Connection closed by foreign host.
```

¿Qué hemos hecho?

- Pues nos hemos conectado al puerto 110 de nuestro servidor, en donde está escuchando nuestro servidor POP.
- Nos identificamos poniendo nuestro nombre de usuario y nuestra password[21].
- El comando `list` nos muestra los mensajes que están en nuestro buzón en el servidor, con su número de orden y su tamaño. Si quisiéramos recuperar alguno, con el comando ***retr*** y el número del mensaje, podremos descargárnoslo.

Protocolo IMAP

El protocolo IMAP (*Internet Messaging Access Protocol*[22]) es más potente que POP en la mayoría de los casos. En el modo desconectado (*offline*) sus capacidades son similares, pero es en el modo conectado (*online*) donde IMAP lo supera con creces. IMAP permite la manipulación de buzones en el servidor remoto como si fueran locales.

En conexiones de poco ancho de banda, permite capturar la estructura del mensaje sin descargarlo[23] y seleccionar qué parte del mensaje nos interesa descargarnos.

Posee adicionalmente la capacidad de manipular un mensaje en el buzón remoto, permitiendo marcar los mensajes como leídos, borrados, contestados. La tendencia es a utilizar servidores con este protocolo en vez de POP. Pero claro está, esto depende de que nuestro proveedor del servicio de correo o administrador del sistema nos ofrezca esta posibilidad.

Si utilizamos un sistema de webmail[24] y deseamos poder crear carpetas, éste es el protocolo necesario.

El servidor Cyrus es un potente sistema que soporta IMAP, POP y sus equivalentes seguros, IMAPS y POPS. Es complejo, pero quien quiera probarlo, puede encontrar un excelente tutorial en `http://www.linuxsilo.net/articles/postfix.html`

2.2. Agentes de Transporte

2.2.1. Postfix

¿Qué es Postfix[25]? Es un servidor de correo (MTA) que inició su existencia en 1998 como una alternativa de *Wietse Venema* al ampliamente usado Sendmail. Inicialmente se distribuyó bajo el nombre de *IBM Secure Mailer*, pasando posteriormente a la denominación de Postfix. En su diseño han primado factores como la seguridad, la eficiencia y la facilidad de configuración y administración, junto con la compatibilidad con Sendmail y con otros sistemas de correo. El exterior está "sendmailizado" pero su interior es totalmente diferente.

Siendo el correo electrónico hoy día una herramienta de trabajo vital en multitud de entornos de trabajo, plantearse sustituir el sistema de correo actual por otro nuevo es una decisión muy delicada. Se debe garantizar que la migración se va a producir sin inconvenientes para los usuarios y con el mínimo tiempo de parada del servicio. Algunas de las virtudes de Postfix que pueden decidir su uso son:

[21] Como ya sabéis, esta información no cifrada viajando por la red es un peligro. Afortunadamente, el servidor POP ofrece un servicio cifrado POP3S en el puerto 995.
[22] Protocolo de Acceso a Mensajería de Internet.
[23] Cosa que no puede hacer POP.
[24] En esta entrega veremos squirrelmail.
[25] Más información en `http://www.postfix.org/`

- Diseño modular, no es un único programa monolítico.
- La seguridad y el rendimiento han sido condicionantes desde el comienzo de su diseño.
- Soporte para las tecnologías más usadas hoy día: LDAP, Bases de datos (MySQL), autentificación mediante SASL, LMTP, etc.
- Soporte para dominios virtuales.
- Facilidad de configuración.
- Compatibilidad hacia/desde fuera con Sendmail.
- Abundante documentación, y de calidad.
- Fácil integración con antivirus.
- Tiene múltiples formas de obtener información de "lo que está pasando" para resolver problemas o simplemente, para aprender.
- Se pueden lanzar varias instancias de Postfix en la misma máquina con distintas configuraciones, usando cada una distintas direcciones IP, distintos puertos, etc.
- Filtrado de cabeceras y cuerpos de mensajes por expresiones regulares.
- Utilidades para la gestión del correo, entre otras para la gestión de las colas de mensajes.

Por último, pero no menos importante, hay que decir que el código fuente de Postfix (por supuesto de dominio público) es un ejemplo de diseño, claridad y documentación, lo cual facilita su mantenimiento (por su autor o, en el futuro, por otros) así como la incorporación de nuevas capacidades, corrección de errores, etc.

En el website de Postfix, `http://www.postfix.org/`, pueden encontrarse enlaces a documentación que profundiza en sus características y en otro aspectos.

Arquitectura de Postfix

Al contrario que Sendmail, que se basaba en una estructura monolítica[26], el diseño de Postfix se basa en la división en distintos procesos del tratamiento que se realiza del correo a través del MTA. Estos procesos se comunican entre sí a través de sockets, siendo la información transmitida la mínima posible. El conjunto de todos estos procesos constituye Postfix.

Una gran contribución a la estabilidad y velocidad del servidor Postfix es la forma inteligente en que su creador implementó las colas de correo. Postfix utiliza varias colas diferentes, cada una manejada de forma diferente:

- *Maildrop queue*. El correo que es entregado localmente en el sistema es aceptado por la cola *Maildrop*. El correo se chequea para formatearlo apropiadamente antes de ser entregado a la cola *Incoming*.
- *Incoming queue*. Esta cola recibe correo de otros *hosts*, clientes o de la cola *Maildrop*. Mientras sigue llegando correo y Postfix no puede manejarlo, en esta cola se quedan los correos.
- *Active queue*. Es la cola utilizada para entregar los mensajes. La *Active queue* tiene un tamaño limitado, y los mensajes solamente serán aceptados si hay espacio en ella. Esto quiere decir que las colas *Incoming* y *Deferred* tienen que esperar hasta que la cola *Active* pueda aceptar más mensajes.

[26] Un único programa que lo hace todo.

Instalación de Postfix

Guadalinex

```
#apt-get install postfix
```

El resultado de la ejecución se muestra a continuación:

```
root@guadalinex:/home/mowgli# apt-get install postfix
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Paquetes sugeridos:
procmail postfix-mysql postfix-pgsql postfix-ldap postfix-pcre
Paquetes recomendados
resolvconf
Se instalarán los siguientes paquetes NUEVOS:
postfix
0 actualizados, 1 se instalarán, 0 para eliminar y 575 no actualizados.
Necesito descargar 801kB de archivos.
Se utilizarán 1970kB de espacio de disco adicional después de desempaquetar.
```

En este momento se inicia de manera automática la configuración de postfix:

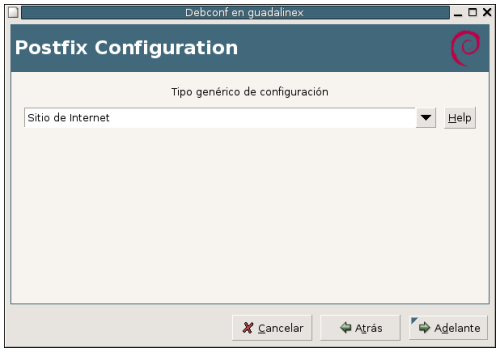


Figura 2.1: Instalacion Postfix - Tipo genérico de configuración

Como tipo genérico de configuración dejamos la opción por defecto, **Sitio de Internet**, y pulsamos [**Adelante**], el asistente de configuración nos muestra la siguiente pantalla[27]:

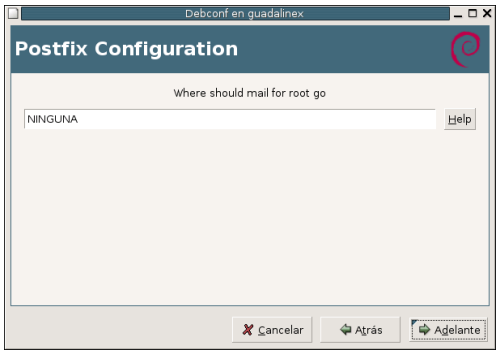


Figura 2.2: Instalacion Postfix - Redirección correo root

[27]Para ver los distintos tipos de instalación y una descripción de los mismos podemos visualizar la pantalla de ayuda a través del botón [**Help**]

Si queremos que el correo dirigido a `root` o a cualquier usuario con `uid 0` se redirija a algún alias, escribiremos uno en la pantalla anterior. Este alias se añade al fichero `/etc/aliases`. En el caso de no querer que se redireccione a ningún alias, se deja la opción por defecto y el correo se redireccionará a `/var/mail/nobody`. En nuestro caso dejaremos la opción por defecto y pulsaremos [**Adelante**], mostrándose la siguiente pantalla:

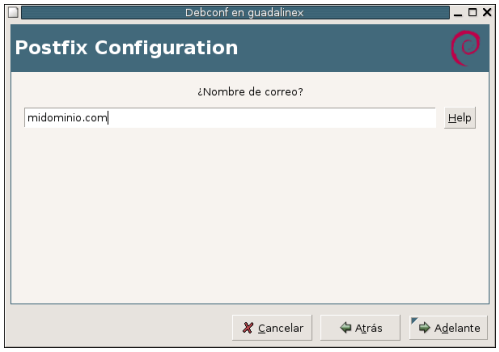


Figura 2.3: Instalación Postfix - Dominio de correo

El "nombre de correo" es la porción del nombre de máquina de la dirección que será mostrada en las noticias y correos salientes (después del nombre de usuario y el signo @). Este nombre será usado por otros programas además de Postfix; deberá ser el único nombre de dominio completo (FDQN) desde el que parecerá originarse el correo. Para nuestro caso utilizaremos `midominio.com`, y pulsaremos [**Adelante**], se muestra la siguiente pantalla del asistente de configuración:

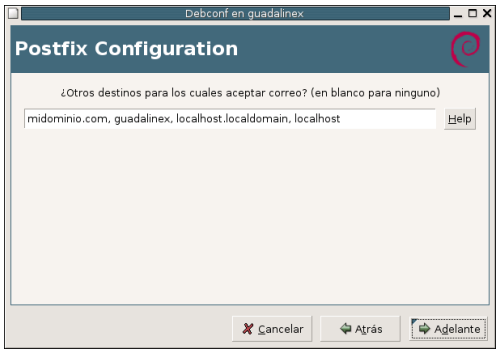


Figura 2.4: Instalación Postfix - Destinos para los que se acepta correo

Deberemos insertar una lista separada por comas, de dominios de los que esta máquina deberá considerarse destino final, en nuestro caso dejaremos los que muestra por defecto, que no son más que el dominio que hemos configurado y el correo local, pulsamos [**Adelante**] y se muestra la siguiente pantalla del asistente:

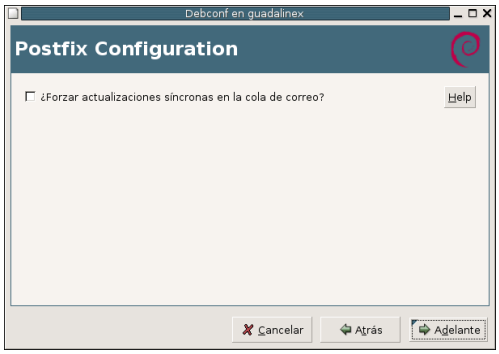


Figura 2.5: Instalación Postfix - Actualización síncrona cola correo

Si se fuerzan las actualizaciones síncronas, el correo será procesado más lentamente. Si no se fuerzan, existe la posibilidad remota de perder algunos correos si el sistema se colapsa en un momento inoportuno y no está usando un sistema de ficheros transaccional (como `ext3`). En nuestro caso el sistema de ficheros es `ext3`, luego dejamos la configuración por defecto `off`.

Pulsamos [**Adelante**] en el resto de opciones, aceptando los valores por defecto, hasta que concluye el asistente de configuración. Continúa la instalación de los paquetes.

Como podemos observar, Postfix queda configurado e iniciado una vez que concluye la instalación[28]. Comprobaremos que la instalación se ha realizado correctamente enviando un correo a través de nuestro servidor (en negrita y a mayor tamaño mostramos lo que vamos tecleando:

```
root@guadalinex:/home/mowgli# telnet localhost 25
Trying ::1...
Connected to localhost.
Escape character is '^]'.
220 guadalinex ESMTP Postfix (Debian/GNU)
EHLO localhost
250-guadalinex
250-PIPELINING
250-SIZE 10240000
250-VRFY
250-ETRN
250 8BITMIME
MAIL FROM: <mowgli@midominio.com>
250 Ok
RCPT TO: <curso@midominio.com>
250 Ok
DATA
354 End data with <CR><LF>.<CR><LF>
SUBJECT: Prueba de correo
FROM: Mowgli <mowgli@midominio.com>
TO: Curso <curso@midominio.com>
Prueba de correo
.
250 Ok: queued as EAE8E3B7E7
QUIT
221 Bye
Connection closed by foreign host.
```

Si todo ha ido correctamente, podemos conectarnos al buzón del usuario curso, y comprobar que ha recibido el correo.

[28] En el caso de que ya tuviésemos instalado Postfix o quisieramos volver a realizar la configuración básica, podemos ejecutar el script de postinstalación de este paquete con el siguiente comando: `dpkg-reconfigure postfix`

```
root@guadalinex:/home/mowgli# more /var/mail/curso
From mowgli@midominio.com Sun Jan 23 12:22:07 2005
Return-Path: <mowgli@midominio.com>
X-Original-To: curso@midominio.com
Delivered-To: curso@midominio.com
Received: from localhost (localhost [IPv6:::1])
by guadalinex (Postfix) with ESMTP id EAE8E3B7E7
for <curso@midominio.com>; Sun, 23 Jan 2005 12:20:53 +0100 (CET)
SUBJECT: Prueba de correo
From: Mowgli <mowgli@midominio.com>
To: Curso <curso@midominio.com>
Message-Id: <20050123112053.EAE8E3B7E7@guadalinex>
Date: Sun, 23 Jan 2005 12:20:53 +0100 (CET)
Prueba de correo
```

Personalizar la configuración

La configuración de Postfix se realiza mediante dos ficheros principales (situados en el directorio `/etc/postfix`) y varias tablas opcionales que puede crear el administrador de correo. Los ficheros de configuración son:

`master.cf` Aquí se configuran los procesos que pueden arrancarse y algunos parámetros como el número de cada uno que puede haber simultáneamente, etc. Normalmente sólo hay que tocarlo si queremos usar un sistema alternativo de entrega de correo local (si usamos Cyrus, Courier, por ejemplo), si queremos integrar un antivirus y cosas así.

`main.cf` Todos los parámetros relacionados con la función que debe realizar Postfix los definimos aquí. Será el fichero sobre el que se realicen las modificaciones más habituales.

En el fichero `main.cf` podemos asignar valores a dos cosas:

- Parámetros. Les asignamos valores como nombres de hosts, direcciones IP, número de bytes en el caso de algunos límites, etc.
- Clases de restricciones. Les asignamos una serie de "restricciones", que definen fundamentalmente de/para quién vamos a aceptar correo.

A las clases de restricciones (que en realidad también son parámetros) se le asignan una serie de valores que definen, fundamentalmente, de/para quién se aceptará correo. En Postfix hay un gran número de parámetros, pero la mayoría define situaciones fuera de lo común o aspectos relacionados con la administración avanzada (afinar el rendimiento, límites, códigos SMTP a devolver a los clientes ante determinadas circunstancias, etc). En realidad lo normal es que haya que tocar poco más de media docena de ellos, aparte de las tablas necesarias para alias, dominios virtuales, etc.

En cuanto a las tablas de configuración pueden estar en una gran variedad de formatos, en dos variantes: De acceso por clave (Berkeley DB, MySQL, etc.) o De acceso secuencial (expresiones regulares).

Tal como se realizó la configuración por defecto se generaron los ficheros de configuración. En el caso de `main.cf` se obtiene el siguiente fichero[29]:

```
# See /usr/share/postfix/main.cf.dist for a commented, more complete version

smtpd_banner = $myhostname ESMTP $mail_name (Debian/GNU)
biff = no
```

[29] Además, añadir que tal cual aparece en ese fichero, en `/usr/share/postfix/main.cf.dist` hay una versión completa y comentada del mismo.

```
# appending .domain is the MUA's job.
append_dot_mydomain = no

# Uncomment the next line to generate "delayed mail" warnings
#delay_warning_time = 4h

myhostname = guada04
alias_maps = hash:/etc/aliases
alias_database = hash:/etc/aliases
mydestination = midominio.com, guadalinex, localhost.localdomain, localhost
relayhost =
mynetworks = 127.0.0.0/8
mailbox_command =
mailbox_size_limit = 0
recipient_delimiter = +
myorigin = /etc/mailname
```

Listado 2.1: /conf/main.cf

Algunos de los parámetros más interesantes en la configuración de Postfix en el fichero `main.cf`:

`queue_directory` Especifica la localización de la cola de Postfix.

`daemon_directory` Especifica la localización de todos los demonios de Postfix

`myhostname` Especifica el nombre de host en internet para este sistema de correo. Por defecto se utiliza el FQDM obtenido con `gethostname()`. Este parámetro se utiliza posteriormente dentro de otros parámetros.

`mydomain` Especifica el nombre de dominio local. Por defecto se utiliza el valor de `$myhostname` menos el primer componente. Este parámetro se utiliza posteriormente dentro de otros parámetros.

`myorigin` Especifica el dominio que aparece en los correos locales como origen. Por defecto se utiliza `$myhostname`, aunque en el caso de utilizar un dominio con múltiples máquinas debe usarse `$mydomain`.

`mydestination` Especifica la lista de dominios para los que esta máquina se considera destino final.

`mynetworks_style` Especifica la lista de clientes SMTP en los que se confia y a los que se les permite el envío de correo a través de Postfix. Mediante el uso de los valores `subnet`, `class` y `host` se permite el acceso a los clientes pertenecientes a la subred, clase o únicamente al host local respectivamente.

`mynetworks` Especifica el direccionamiento para el cual se permite el reenvío de correo a través de Postfix. En caso de definir este parámetro se ignora lo definido en `$mynetworks_style`.

`relay_domains` Especifica los dominios para los que está permitido el reenvío de correo.

`relayhost` Especifica el host hacia el que se envía el correo y que hace de reenviador en caso de que no esté directamente conectado a internet.

`alias_maps` Especifica la lista de base de datos de alias para el reenviador de correo local.

`alias_database` Especifica la base de datos de alias que se genera cada vez que se ejecuta `newaliases`.

`home_mailbox` Especifica la ruta, relativa al directorio `$HOME`, del fichero con el buzón de correo. Por defecto es `/var/spool/mail/user` o `/var/mail/user`. Si se especifica `Maildir/` se utilizará el formato utilizado por qmail.

`mail_spool_directory` Especifica la ruta donde se almacenan los buzones con formato Unix mailbox.

`header_checks` Especifica una tabla de patrones con la que se compara las cabeceras de los mensajes.

El otro fichero que utiliza Postfix en su configuración es `master.cf`. Define el comportamiento del programa master. Dicho programa forma parte de Postfix y se ejecuta de forma continua en el servidor de correo. Recibe indicaciones de unos procesos e inicia otros. Es el "director de orquesta" de Postfix. Mediante `master.cf` se define la forma correcta en que se debe llamar a cada uno de los procesos. Cada entrada en el fichero es un conjunto de ocho campos separados por blancos o tabuladores:

- *Service*. Nombre del servicio que se está configurando.
- *Type*. Tipo de comunicación de transporte utilizado por el servicio.
- *Private*. Restricciones de seguridad a procesos externos.
- *Unprivileged*. Ejecución en modo no privilegiado.
- *Chroot*. Indica si el servicio se ejecuta en un directorio de acceso restringido.
- *Wakeup*. Segundos que deben transcurrir para que el proceso master despierte el servicio.
- *Maxprocess*. Número máximo de procesos que puede usar el servicio.
- *Command*. Nombre del programa a ejecutar y parámetros a pasar.

Este fichero no es necesario modificarlo con la instalación por defecto:

```
#
# Postfix master process configuration file.  Each logical line
# describes how a Postfix daemon program should be run.
#
# A logical line starts with non-whitespace, non-comment text.
# Empty lines and whitespace-only lines are ignored, as are comment
# lines whose first non-whitespace character is a '#'.
# A line that starts with whitespace continues a logical line.
#
# The fields that make up each line are described below. A "-" field
# value requests that a default value be used for that field.
#
# Service: any name that is valid for the specified transport type
# (the next field).  With INET transports, a service is specified as
# host:port.  The host part (and colon) may be omitted.  Either host
# or port may be given in symbolic form or in numeric form.  Examples
# for the SMTP server:  localhost:smtp receives mail via the loopback
# interface only; 10025 receives mail on port 10025.
#
# Transport type: "inet" for Internet sockets, "unix" for UNIX-domain
# sockets, "fifo" for named pipes.
#
# Private: whether or not access is restricted to the mail system.
# Default is private service.  Internet (inet) sockets can't be private.
#
# Unprivileged: whether the service runs with root privileges or as
# the owner of the Postfix system (the owner name is controlled by the
# mail_owner configuration variable in the main.cf file).  Only the
# pipe, virtual and local delivery daemons require privileges.
```

```
#
# Chroot: whether or not the service runs chrooted to the mail queue
# directory (pathname is controlled by the queue_directory configuration
# variable in the main.cf file). Presently, all Postfix daemons can run
# chrooted, except for the pipe, virtual and local delivery daemons.
# The proxymap server can run chrooted, but doing so defeats most of
# the purpose of having that service in the first place.
# The files in the examples/chroot-setup subdirectory describe how
# to set up a Postfix chroot environment for your type of machine.
#
# Wakeup time: automatically wake up the named service after the
# specified number of seconds. A ? at the end of the wakeup time
# field requests that wake up events be sent only to services that
# are actually being used. Specify 0 for no wakeup. Presently, only
# the pickup, queue manager and flush daemons need a wakeup timer.
#
# Max procs: the maximum number of processes that may execute this
# service simultaneously. Default is to use a globally configurable
# limit (the default_process_limit configuration parameter in main.cf).
# Specify 0 for no process count limit.
#
# Command + args: the command to be executed. The command name is
# relative to the Postfix program directory (pathname is controlled by
# the daemon_directory configuration variable). Adding one or more
# -v options turns on verbose logging for that service; adding a -D
# option enables symbolic debugging (see the debugger_command variable
# in the main.cf configuration file). See individual command man pages
# for specific command-line options, if any.
#
# General main.cf options can be overridden for specific services.
# To override one or more main.cf options, specify them as arguments
# below, preceding each option by "-o". There must be no whitespace
# in the option itself (separate multiple values for an option by
# commas).
#
# In order to use the "uucp" message tranport below, set up entries
# in the transport table.
#
# In order to use the "cyrus" message transport below, configure it
# in main.cf as the mailbox_transport.
#
# SPECIFY ONLY PROGRAMS THAT ARE WRITTEN TO RUN AS POSTFIX DAEMONS.
# ALL DAEMONS SPECIFIED HERE MUST SPEAK A POSTFIX-INTERNAL PROTOCOL.
#
# DO NOT SHARE THE POSTFIX QUEUE BETWEEN MULTIPLE POSTFIX INSTANCES.
#
# ==========================================================================
# service type  private unpriv  chroot  wakeup  maxproc command + args
#               (yes)   (yes)   (yes)   (never) (100)
# ==========================================================================
127.0.0.1:smtp inet n       -       -       -       -       smtpd
::1:smtp       inet n       -       -       -       -       smtpd
#submission inet n       -       -       -       -       smtpd
#       -o smtpd_etrn_restrictions=reject
#628        inet  n       -       -       -       -       qmqpd
pickup     fifo  n       -       -       60      1       pickup
cleanup    unix  n       -       -       -       0       cleanup
qmgr       fifo  n       -       -       300     1       qmgr
```

```
#qmgr     fifo   n       -       -       300     1       oqmgr
rewrite   unix   -       -       -       -       -       trivial-rewrite
bounce    unix   -       -       -       -       0       bounce
defer     unix   -       -       -       -       0       bounce
trace     unix   -       -       -       -       0       bounce
verify    unix   -       -       -       -       1       verify
flush     unix   n       -       -       1000?   0       flush
proxymap  unix   -       -       n       -       -       proxymap
smtp      unix   -       -       -       -       -       smtp
relay     unix   -       -       -       -       -       smtp
#         -o smtp_helo_timeout=5 -o smtp_connect_timeout=5
showq     unix   n       -       -       -       -       showq
error     unix   -       -       -       -       -       error
local     unix   -       -       n       n       -       -       local
virtual   unix   -       -       n       n       -       -       virtual
lmtp      unix   -       -       -       n       -       -       lmtp
anvil     unix   -       -       -       n       -       1       anvil
#
# Interfaces to non-Postfix software. Be sure to examine the manual
# pages of the non-Postfix software to find out what options it wants.
#
# maildrop. See the Postfix MAILDROP_README file for details.
#
maildrop  unix   -       n       n       -       -       pipe
  flags=DRhu user=vmail argv=/usr/local/bin/maildrop -d ${recipient}
uucp      unix   -       n       n       -       -       pipe
  flags=Fqhu user=uucp argv=uux -r -n -z -a$sender - $nexthop!rmail (
    $recipient)
ifmail    unix   -       n       n       -       -       pipe
  flags=F user=ftn argv=/usr/lib/ifmail/ifmail -r $nexthop ($recipient)
bsmtp     unix   -       n       n       -       -       pipe
  flags=Fq. user=bsmtp argv=/usr/lib/bsmtp/bsmtp -d -t$nexthop -f$sender
    $recipient
scalemail-backend unix - - n - n - - 2 - pipe
  flags=R user=scalemail argv=/usr/lib/scalemail/bin/scalemail-store ${
    nexthop} ${user} ${extension}

# only used by postfix-tls
#tlsmgr    fifo   -       -       n       300     1       tlsmgr
#smtps     inet   n       -       n       -       -       smtpd -o
    smtpd_tls_wrappermode=yes -o smtpd_sasl_auth_enable=yes
#587       inet   n       -       n       -       -       smtpd -o
    smtpd_enforce_tls=yes -o smtpd_sasl_auth_enable=yes
```

Listado 2.2: /conf/master.cf

Arranque de Postfix

Una vez instalado, se procede a arrancar el demonio, si no lo está ya. Para ello se utiliza el script creado en `/etc/init.d`. Con este script se comprueba también la correcta definición en el fichero de configuración `main.cf`

```
#/etc/init.d/postfix check
```

Una vez que se ha comprobado que no hay errores, es necesario indicar a Postfix que cargue de nuevo la configuración:

```
#/etc/init.d/postfix reload
```

Para obtener más información referente a posibles errores o el estado de ejecución de Postfix se recurrirá a los ficheros de log:

```
/var/log/mail.err
/var/log/mail.info
/var/log/mail.log
/var/log/mail.warn
```

Dependiendo del nivel de criticidad del aviso se almacenará en uno u otro fichero.

2.2.2. Sendmail

> El Rey Gordio de Frigia hizo un nudo tan fuerte que nadie pudo deshacerlo. El Nudo Gordiano permaneció así, o al menos eso dice la historia, hasta que llegó Alejandro Magno y utilizó una forma diferente de deshacer nudos. Sería interesante si el nudo que es sendmail pudiera ser desatado con un rápido golpe de una nueva visión, pero !ay¡, no es posible. En vez de ello, se debe coger un enfoque más mundano, así que en este libro lo desataremos de la manera difícil, hebra a hebra. *Sendmail* Bryan Costales y Eric Allman.

Estas palabras describen a la perfección lo que ha sido Sendmail durante mucho tiempo, una de las bestias negras de los administradores de sistemas Unix. Una relación amor/odio se entablaba con esta gran obra de ingeniería. Por una parte, su potencia y capacidades eran insustituibles y por otra, su complejidad de configuración y sus errores de seguridad le hacían temible. Por ello, Postfix está ganando poco a poco cuota de poder en el dominio de los agentes de transporte de correo en el "mundo libre".

Por fortuna, esa situación ha cambiado un poco. La inclusión del preprocesador de macros *m4* para la configuración y sus reescrituras para mejorar el diseño y la seguridad, han mejorado un poco la situación.

Sendmail fue escrito por Eric Allman en la Universidad de California en Berkeley para el Unix de BSD. Ha sido portado a todas las plataformas existentes y todas las distribuciones de Linux la incorporan. Vamos a hablar sobre él por motivos históricos y porque aún se utiliza mucho, aunque postfix va imponiéndose.

Instalación de Sendmail

Para instalar los paquetes que necesitamos para el correo utilizaremos com de costumbre, `apt-get`.

Si vamos a utilizar sendmail, es importante utilizar una versión reciente, ya que son muchas las mejoras de seguridad que incorporan. Por ejemplo, el que vamos a instalar de prueba, será sendmail-8.12.10. Con la utilidad apt-get, obtendremos una versión actualizada.

Los paquetes que instalaremos serán[30] *sendmail*, el agente de transporte, *sendmail-cf*, las utilidades para la configuración y *dovecot*, es el paquete en que se encuentran los servidores de POP e IMAP.

```
[root@linux entrega04-3]# apt-get install sendmail sendmail-cf dovecot
Leyendo listas de paquetes... Done
Construyendo Árbol de dependencias... Done
Se instalarán los paquetes NUEVOS siguientes:
imap sendmail sendmail-cf
0 upgraded, 3 newly installed, 0 removed and 51 not upgraded.
Need to get 2600kB of archives.
After unpacking 5926kB of additional disk space will be used.
```

[30] En Debian:
`#apt-get install sendmail dovecot`

```
Get:1 http://ayo.freshrpms.net fedora/linux/1/i386/core dove-
cot 1:2002d-3 [1287kB]
Get:2 http://ayo.freshrpms.net fedora/linux/1/i386/core send-
mail 8.12.10-1.1.1 [1018kB]
Get:3 http://ayo.freshrpms.net fedora/linux/1/i386/core sendmail-
cf 8.12.10-1.1.1 [294kB]
Fetched 2600kB in 1m41s (25,5kB/s)
Committing changes...
Preparing... ########################################### [100%]
1:sendmail-cf ########################################### [ 33%]
2:dovecot ############################################### [ 67%]
3:sendmail ############################################## [100%]
Done.
```

Una vez instalados, mediante la utilidad[31] `setup`, podremos configurarlos para que arranquen automáticamente. Los servicios que activaremos serán[32]:

sendmail Demonio del Agente de Transporte. Utiliza el puerto 25.

imap Servidor para acceder a los buzones de usuario utilizando el protocolo IMAP. Utiliza el puerto 143.

imaps Igual que imap pero con un protocolo cifrado. Utiliza el puerto 993.

ipop3 Servidor del protocolo POP[33]. Utiliza el puerto 110.

ipop3s Servidor POP seguro. Utiliza el puerto 995.

Configuración de Sendmail

Pasemos a configurar sendmail. Miremos el fichero `sendmail.cf`, que se encuentra en el directorio `/etc/mail`[34].

```
# strip group: syntax (not inside angle brackets!) and trailing semico-
lon
R$* $: $1 <@> mark addresses
R$* < $* > $* <@> $: $1 < $2 > $3 unmark <addr>
R@ $* <@> $: @ $1 unmark @host:...
R$* [ IPv6 : $+ ] <@> $: $1 [ IPv6 : $2 ] unmark IPv6 addr
R$* :: $* <@> $: $1 :: $2 unmark node::addr
R:include: $* <@> $: :include: $1 unmark :include:...
R$* : $* [ $* ] $: $1 : $2 [ $3 ] <@> remark if leading colon
R$* : $* <@> $: $2 strip colon if marked
R$* <@> $: $1 unmark
R$* ; $1 strip trailing semi
R$* < $+ :; > $* $@ $2 :; <@> catch <list:;>
R$* < $* ; > $1 < $2 > bogus bracketed semi
```

Como véis, quien sea capaz de entender esto, no debe ser una persona normal. Es una de las razones de la mala fama (y en parte merecida) de sendmail.

[31] Si bien se ejecuta de forma automática

```
# update-rc.d sendmail defaults
```

para Debian.

[32] Al menos sendmail y un protocolo de acceso a los buzones.

[33] El protocolo POP en su versión 2 (ipop2) no se utiliza normalmente.

[34] O en `/etc`

Pero gracias a la utilización del preprocesador de macros `m4`, la tarea se nos ha vuelto más fácil. Bueno, aún así nos llevará un poco comprenderla totalmente. Nuestro fichero de configuración será `/etc/mail/sendmail.mc`[35] y a partir de él obtendremos el fichero `sendmail.cf`, que es el que leerá sendmail.

En el directorio `/usr/share/sendmail-cf/cf` existen ejemplos de ficheros `.mc` para múltiples sistemas. Escogeremos el correspondiente a nuestro sistema.

Tenemos un punto a nuestro favor. La configuración por defecto nos servirá casi sin modificaciones en un tanto por ciento muy elevado de casos. Cuando lo tengamos a nuestro gusto, simplemente tecleamos `#make` en el directorio `/etc/mail` y se generará automáticamente el fichero `sendmail.cf`[36]. Comentaremos las líneas más interesantes.

```
[root@linux mail]# more sendmail.mc
divert(-1)dnl
dnl #
dnl # This is the sendmail macro config file for m4. If you make chan-
ges to
dnl # /etc/mail/sendmail.mc, you will need to regenerate the
dnl # /etc/mail/sendmail.cf file by confirming that the sendmail-
cf package is
dnl # installed and then performing a
dnl #
dnl # make -C /etc/mail
dnl #
```

Las líneas `divert` y `dnl` son comentarios.

```
include(‘/usr/share/sendmail-cf/m4/cf.m4’)dnl
```

Carga el fichero `cf.m4` que necesita.

```
VERSIONID(‘setup for Red Hat Linux’)dnl
OSTYPE(‘linux’)dnl
```

Decimos la versión y el sistema operativo. Le servirá para adoptar opciones personalizadas. En este caso, cargará el fichero `/usr/share/sendmail-cf/ostype/linux.m4`.

```
dnl #
dnl # Uncomment and edit the following line if your out-
going mail needs to
dnl # be sent out through an external mail server:
dnl #
dnl define(‘SMART_HOST’,‘smtp.miproveedor.com’)
```

Opción que está comentada. El `Smart_host` es un agente de transporte al que le paso la pelota y que él se encargue de los siguientes pasos. Muy útil si estamos en una red privada y solamente ese "host inteligente" puede salir hacia el exterior. Para utilizarla, tendríamos que poner cuál es ese host en nuestro caso y descomentarla quitando el `dnl`.

[35] En un sistema Fedora o RedHat

[36] También podemos ejecutar a mano:
`#m4 ${CFDIR}/m4/cf.m4 fichero.mc >fichero.cf`

- `m4` es el procesador de macros
- necesita del fichero `cf.m4`
- actúa sobre el fichero `.mc`
- genera el fichero `.cf`

```
dnl #
define(‘confDEF_USER_ID’,"8:12")dnl
```

Usuario y grupo que ejecutarán el proceso sendmail (normalmente usuario mail y grupo mail).

```
dnl define(‘confAUTO_REBUILD’)dnl
define(‘confTO_CONNECT’, ‘1m’)dnl
define(‘confTRY_NULL_MX_LIST’,true)dnl
define(‘confDONT_PROBE_INTERFACES’,true)dnl
define(‘PROCMAIL_MAILER_PATH’,‘/usr/bin/procmail’)dnl
define(‘ALIAS_FILE’, ‘/etc/aliases’)dnl
```

Cuál es el fichero de alias. Crea una dirección de correo virtual y la asocia a otra dirección. Por ejemplo la línea `webmaster: admin` dice que todos los correos que vayan a la dirección webmaster@dominio-configurado.com, siendo `dominio-configurado.com` el que está recogiendo nuestro sendmail, vayan a la dirección admin@dominio-configurado.com

```
dnl define(‘STATUS_FILE’, ‘/etc/mail/statistics’)dnl
define(‘UUCP_MAILER_MAX’, ‘2000000’)dnl
define(‘confUSERDB_SPEC’, ‘/etc/mail/userdb.db’)dnl
define(‘confPRIVACY_FLAGS’, ‘authwar-
nings,novrfy,noexpn,restrictqrun’)dnl
define(‘confAUTH_OPTIONS’, ‘A’)dnl
dnl #
dnl # The following allows relaying if the user authenticates, and di-
sallows
dnl # plaintext authentication (PLAIN/LOGIN) on non-TLS links
dnl #
dnl define(‘confAUTH_OPTIONS’, ‘A p’)dnl
dnl #
dnl # PLAIN is the preferred plaintext authentication met-
hod and used by
dnl # Mozilla Mail and Evolution, though Outlook Express and ot-
her MUAs do
dnl # use LOGIN. Other mechanisms should be used if the connec-
tion is not
dnl # guaranteed secure.
dnl #
dnl TRUST_AUTH_MECH(‘EXTERNAL DIGEST-MD5 CRAM-MD5 LOGIN PLAIN’)dnl
dnl define(‘confAUTH_MECHANISMS’, ‘EXTERNAL GSSAPI DIGEST-MD5 CRAM-
MD5 LOGIN PLAIN’)dnl
dnl # Rudimentary information on creating certificates for send-
mail TLS:
dnl # make -C /usr/share/ssl/certs usage
dnl #
dnl define(‘confCACERT_PATH’,‘/usr/share/ssl/certs’)
dnl define(‘confCACERT’,‘/usr/share/ssl/certs/ca-bundle.crt’)
dnl define(‘confSERVER_CERT’,‘/usr/share/ssl/certs/sendmail.pem’)
dnl define(‘confSERVER_KEY’,‘/usr/share/ssl/certs/sendmail.pem’)
dnl #
dnl # This allows sendmail to use a keyfile that is shared with OpenL-
DAP’s
dnl # slapd, which requires the file to be readble by group ldap
dnl #
dnl define(‘confDONT_BLAME_SENDMAIL’,‘groupreadablekeyfile’)dnl
```

```
dnl #
dnl define(‘confTO_QUEUEWARN’, ‘4h’)dnl
```

Los mensajes que recoge sendmail los pone en una cola (un directorio en donde los va guardando) y los envía cuando puede. Por ejemplo, si nuestra conexión a Internet no es permanente o el agente de transporte destino no está operativo. Este parámetro designa el tiempo (4 horas) que al cumplirse, nos envía un mensaje indicando que no lo ha podido entregar.

```
dnl define(‘confTO_QUEUERETURN’, ‘5d’)dnl
```

Si en 5 días no ha conseguido entregarlo al destinatario, nos lo devuelve.

```
dnl define(‘confQUEUE_LA’, ‘12’)dnl
dnl define(‘confREFUSE_LA’, ‘18’)dnl
define(‘confTO_IDENT’, ‘0’)dnl
dnl FEATURE(delay_checks)dnl
FEATURE(‘no_default_msa’,‘dnl’)dnl
FEATURE(‘smrsh’,‘/usr/sbin/smrsh’)dnl
FEATURE(‘mailertable’,‘hash -o /etc/mail/mailertable.db’)dnl
FEATURE(‘virtusertable’,‘hash -o /etc/mail/virtusertable.db’)dnl
FEATURE(redirect)dnl
FEATURE(always_add_domain)dnl
```

Siempre añade el dominio para completar las direcciones de correo electrónico. Por ejemplo, estando en el dominio midominio.com, un correo enviado al usuario linux, se completará como linux@midominio.com.

```
FEATURE(use_cw_file)dnl
FEATURE(use_ct_file)dnl
dnl #
dnl # The -
t option will retry delivery if e.g. the user runs over his quota.
dnl #
FEATURE(local_procmail,‘’,‘procmail -t -Y -a $h -d $u’)dnl
FEATURE(‘access_db’,‘hash -T<TMPF> -o /etc/mail/access.db’)dnl
FEATURE(‘blacklist_recipients’)dnl
EXPOSED_USER(‘root’)dnl
dnl #
dnl # The following causes sendmail to only listen on the IPv4 loop-
back address
dnl # 127.0.0.1 and not on any other network devices. Remove the loop-
back
dnl # address restriction to accept email from the internet or intra-
net.
dnl #
DAEMON_OPTIONS(‘Port=smtp,Addr=127.0.0.1, Name=MTA’)dnl
```

Muy importante. Debemos poner nuestra dirección de red local en vez de 127.0.0.1 si queremos que los clientes puedan comunicar con el servidor.

```
dnl #
dnl # The following causes sendmail to additionally lis-
ten to port 587 for
dnl # mail from MUAs that authenticate. Roa-
ming users who can’t reach their
```

```
dnl # preferred sendmail daemon due to port 25 being blocked or redi-
rected find
dnl # this useful.
dnl DAEMON_OPTIONS(‘Port=submission, Name=MSA, M=Ea’)dnl
dnl #
dnl # The following causes sendmail to additionally lis-
ten to port 465, but
dnl # starting immediately in TLS mode upon connec-
ting. Port 25 or 587 followed
dnl # by STARTTLS is preferred, but roaming clients using Outlook Ex-
press can’t
dnl # do STARTTLS on ports other than 25. Mozi-
lla Mail can ONLY use STARTTLS
dnl # and doesn’t support the deprecated smtps; Evolu-
tion <1.1.1 uses smtps
dnl # when SSL is enabled--
STARTTLS support is available in version 1.1.1.
dnl #
dnl # For this to work your OpenSSL certificates must be configured.
dnl #
dnl DAEMON_OPTIONS(‘Port=smtps, Name=TLSMTA, M=s’)dnl
dnl #
dnl # The following causes sendmail to additionally lis-
ten on the IPv6 loopback
dnl # device. Remove the loopback address restriction lis-
ten to the network.
dnl #
dnl # NOTE: binding both IPv4 and IPv6 daemon to the same port requires
dnl # a kernel patch
dnl #
dnl DAEMON_OPTIONS(‘port=smtp,Addr=::1, Name=MTA-v6, Family=inet6’)dnl
dnl #
dnl # We strongly recommend not accepting unresolvable do-
mains if you want to
dnl # protect yourself from spam. However, the laptop and users on com-
puters
dnl # that do not have 24x7 DNS do need this.
dnl #
FEATURE(‘accept_unresolvable_domains’)dnl
dnl #
dnl FEATURE(‘relay_based_on_MX’)dnl
dnl #
dnl # Also accept email sent to "localhost.localdomain" as local email.
dnl #
LOCAL_DOMAIN(‘localhost.localdomain’)dnl
dnl #
dnl # The following example makes mail from this host and any additio-
nal
dnl # specified domains appear to be sent from mydomain.com
dnl #
dnl MASQUERADE_AS(‘mydomain.com’)dnl
```

Si nuestra máquina se llama `mail.midominio.com`, el correo que salga de ella, si no hacemos algo en contrario, será con direcciones del tipo: usuario@mail.midominio.com. Si queremos que salgan

con direcciones del dominio, es decir, usuario@midominio.com, utilizamos el enmascaramiento. Es normal dentro de una organización utilizar el dominio para el correo, y no direcciones de máquinas particulares.

```
dnl #
dnl # masquerade not just the headers, but the envelope as well
dnl #
dnl FEATURE(masquerade_envelope)dnl
dnl #
dnl # masquerade not just @mydomainalias.com, but @*.mydomaina-
lias.com as well
dnl #
dnl FEATURE(masquerade_entire_domain)dnl
dnl #
dnl MASQUERADE_DOMAIN(localhost)dnl
dnl MASQUERADE_DOMAIN(localhost.localdomain)dnl
dnl MASQUERADE_DOMAIN(mydomainalias.com)dnl
dnl MASQUERADE_DOMAIN(mydomain.lan)dnl
MAILER(smtp)dnl
MAILER(procmail)dnl
```

Lo conveniente es no tocar el fichero `sendmail.cf`, sino todos los cambios realizarlos sobre el fichero `sendmail.mc` y de éste generar el fichero `sendmail.cf`.

Control del Spam con Sendmail

El spam o correo no solicitado hoy día es un grave problema. Cuando surgió Internet, lo fundamental era que los protocolos y los sistemas funcionasen. Además, las comunidades de usuarios seguían unos códigos de conducta ética denominados "etiquetas de red[37]" y cualquiera podía utilizar el servidor SMTP para enviar sus correos sin mayores restricciones.

Hoy día esto no es así. Personas sin escrúpulos utilizan la red para sus fines no demasiado éticos, sin importarles mucho el resto de usuarios. Alguien puede utilizar nuestro agente de transporte y mandar miles de correos a través de él. Además, si hemos dejado nuestra máquina desprotegida y alguien la ha utilizado para enviar correo basura, nos pueden meter en una lista negra y no permitírsenos el envío de correo. Sendmail ha tenido que adaptarse a este nuevo entorno muy diferente de aquél en el que nació.

Por ello, la cofiguración por defecto, cada vez viene más cerrada. Para que se puedan enviar correos desde las máquinas clientes de nuestra red local, hay que permitíselo expresamente. Para aquellas IP locales o dominios a los que optemos por permitir que envíen correos a través de nuestro sendmail, añadiremos una entrada en el fichero `/etc/mail/access` del tipo:

```
dirección_IP RELAY
```

por ejemplo, para permitir utilizar el envío SMTP a las máquinas de la red local, añadiríamos la línea:

```
172.26.0.* RELAY
```

después de salvar el fichero, para que los cambios tengan efecto ejecutaremos en el directorio `/etc/mail` el comando

```
# make
```

para generar, a partir del fichero de texto `/etc/mail/access`, el fichero de base de datos mucho más eficiente `/etc/mail/access.db`.

[37] Net etiquette

¿Qué pasa cuando los clientes que deben conectar al servidor acceden desde el exterior con direcciones diferentes y no controlables? Existen dos soluciones posibles para controlar el uso de nuestro sendmail. Una solución se llama "POP before SMTP" y consiste en realizar una conexión POP con usuario y contraseña antes de poder conectar por SMTP. La otra se denomina SMTP_AUTH y consiste en mandar un usuario y contraseña para conectarnos al servidor SMTP. Presentamos las líneas que hay que utilizar en el fichero `sendmail.mc` de Fedora para esta segunda opción:

```
define(‘confAUTH_OPTIONS’, ‘A’)dnl
TRUST_AUTH_MECH(‘EXTERNAL DIGEST-MD5 CRAM-MD5 LOGIN PLAIN’)dnl
define(‘confAUTH_MECHANISMS’, ‘EXTERNAL GSSAPI DIGEST-MD5 CRAM-
MD5 LOGIN PLAIN’)
```

2.3. Agente de entrega: Fetchmail

Fetchmail es una utilidad que permite recuperar y reenviar correo[38]. Se encarga de buscar el correo en los servidores remotos y lo reenvía al sistema de reparto de la máquina local, desde donde podemos recuperarlo con cualquiera de los MUA normales. Podemos entenderlo como un programa que nos va a permitir bajarnos el correo de los distintos servidores y redistribuirlo entre los distintos usuarios del sistema. En sistemas con conexión permanente a internet no es muy útil.

Fetchmail se puede ejecutar en modo demonio para que sondee uno o más sistemas en un intervalo determinado.

El programa puede recoger correo de servidores que soporten cualquiera de los siguientes protocolos: POP2, POP3, IMAP2bis, IMAP4 e IMAPPrev1; también puede usar la extensión ESMTP ETRN y ODMR.

Como fetchmail se pensó principalmente para su uso sobre enlaces no permanentes TCP/IP (por ejemplo conexiones SLIP o PPP por módem), puede ser útil como un agente de transferencia de mensajes para aquellos sitios que no permiten (por motivos de seguridad) transaciones SMTP con `sendmail`.

Normalmente, fetchmail reparte cada mensaje recuperado vía SMTP hacia el puerto 25 sobre la máquina en que se está ejecutando (`localhost`), como si pasara sobre un enlace TCP/IP normal. El correo será después repartido localmente por el MDA (*Mail Delivery Agent*) de su sistema (postfix, sendmail, smail, mmdf, exim, qmail). Todos los mecanismos de reparto y agentes de transporte local funcionarán automáticamente.

Si no hay ningún puerto 25 a la escucha (y en la configuración de fetchmail indicamos un MDA local) se usará el MDA para el reparto local.

Para instalarlo ejecutaremos[39]:

```
#apt-get install fetchmail
```

En Fedora se instala por defecto, mientras que en Guadalinex tendremos que configurar varias opciones en el proceso de instalación.

[38] Agente de entrega o MDA

[39] Si en Debian tenemos problemas con la instalación, lo podemos solucionar con:

1. Añadiremos un grupo de nombre `nogroup`:

```
$cat /etc/group
...
nogroup:x:65534
```

2. Añadiremos el usuario fetchmail

```
#useradd -g 65534 -d /var/run/fetchmail fetchmail
```

2.3.1. Configuración

Para configurar fetchmail se usa el fichero `$HOME/.fetchmailrc`. Todos los parámetros disponibles para este fichero se pueden pasar a fetchmail desde la línea de comandos. Un fichero `.fetchmailrc` puede tener:

- Opciones globales de la conexión. Los parámetros más comunes para esta sección son:

  `set daemon segundos` se ejecuta como demonio e intenta bajar el correo cada "segundos" segundos.

  `set postmaster usuario` todos los correos con problemas de entrega se mandan a "usuario"

  `set syslog /directorio/fichero` fichero para registrar los logs de fetchmail (`/var/log/maillog`)

- Opciones de servidor en el que se busca el correo. Se escriben detrás de `poll` o `skip`.

  - Con `poll` le decimos a fetchmail que baje el correo del servidor especificado cuando se ejecuta sin argumentos, es el habitual.
  - Si anteponemos `skip`, no se bajará el correo de ese servidor salvo que se lo pasemos a fetchmail como argumento en la línea de comandos.

  Las más usuales son

  `interval n` sólo se chequea este servidor cada `n` ciclos. Útil para configurar los servidores de los que rara vez recibimos correo.

  `port puerto` para asignar un número de puerto distinto del habitual

  `proto PROTOCOLO` para especificar el protocolo[40]: POP2, POP3, IMAP, APOP, KPOP

- Opciones de usuario necesarias para autenticarse ante un servidor de correo en concreto[41].

`fetchall` recoger todos los mensajes del servidor o servidores, incluso los ya vistos. Si no se especifica, se bajarán sólo los mensajes nuevos.

`fetchlimit n` número máximo de mensajes para bajar en una conexión

`flush` elimina los mensajes ya vistos, antes de iniciar la descarga de los mensajes nuevos.

`keep` para no borrar los mensajes del servidor

`limit numerobytes` limitamos el tamaño de los correos bajados

`pass8bits` permite caracteres de 8 bits

`password` contraseña a usar para ese usuario (equivale a `pass`)

`ssl` conecta con el servidor usando una conexión SSL siempre que el servidor la soporte.

`to usuario` nombre de usuario local al que enviar el correo

`user usuario` nombre de usuario en el servidor de correo

Vamos a configurarlo como root[42], el motivo de hacerlo así (no es obligatorio) es que sea éste el encargado de bajarse el correo de los distintos servidores para después distribuirlo a los distintos usuarios.

Para eso crearemos en el directorio del `/root` un fichero de nombre `.fetchmailrc` con las líneas:

[40] Cuidado que tienen que ir en mayúsculas.

[41] A `fetchall`, `flush`, `keep`, `pass8bits` se le puede anteponer "no" para hacer lo contrario de lo explicado. Por ejemplo `no keep` borra los mensajes.

Para conocer todas las opciones disponibles: `http://www.catb.org/~esr/fetchmail/fetchmail-man.html`

[42] Lo aquí expuesto es igualmente válido si en vez del root es cualquier usuario del sistema.

```
# valores por defecto
defaults
# recoger todos los mensajes del servidor/es
fetchall
#borrarlos después de bajarlos. Si en vez de flush
#escribimos keep los mensajes no se borran del servidor.
flush
#permite caracteres de 8 bits
pass8bits
#una entrada poll por cada servidor de correo
poll servidor_de_correo_1
   #protocolo usado por el servidor. En general será POP3
   proto pop3
   #nombre de usuario en el servidor de correo
   user "usuario1"
   #contraseña en el servidor de correo
   pass "password1"
   #usuario local al que dirigir el correo
   to usuario_local_1
poll servidor_de_correo_2
   proto pop3
   user "usuario2"
   pass "password2"
   to usuario_local_2
#todos tenemos amigos de esos que piensan que el correo es
#para mandar fotos, vídeos, etc. Si deseamos limitar el tamaño
#del correo bajado a un máximo de 2MB escribiremos
limit 2000000
#si deseamos que trabaje como demonio añadiremos esta línea
#para controlar nosotros la ejecución del programa comentar esta
    línea
set daemon 300
```

Veamos un fichero `.fetchmailrc` de ejemplo para dos servidores de correo y reenvío a distintos usuarios del sistema (thales y mileto):

```
$ cat .fetchmailrc
defaults
fetchall
flush
#keep
pass8bits
poll tux.midominio.com
  proto pop3
  user "pvillegas"
  pass "contraseña_1"
  to thales
poll mileto.cica.es
  proto pop3
  user "ed00linux"
  pass "contraseña_2"
  to mileto

set daemon 300
```

Donde lo único que se ha modificado son las contraseñas de acceso a ambos servidores de correo. Este fichero debe tener unos permisos de lectura y escritura sólo del root y nadie más, así tendremos que usar

```
# chmod 0600 /root/.fetchmailrc
```

Para recibir el correo sólo tenemos que ejecutar como el usuario que tiene el fichero `.fetcmahilrc` el comando[43]:

```
$fetchmail
```

Los correos así bajados se almacenan en `/var/spool/mail` en espera de que los leamos con nuestro MUA preferido.

Si hemos optado por dejar `fetchmail` a la escucha y deseamos matarlo hay que usar

```
fetchmail --quit
```

Para Fedora: En general, interesa que `fetchmail` se inicie en el arranque, para eso sólo tenemos que poner en el subdirectorio `/etc/init.d` el fichero

```
$ cat /etc/init.d/fetchmaild
#!/bin/sh
#
# description: Automatiza el arranque de fetchmail con
# el arranque del sistema.
#
# chkconfig: 345 11 92
# config:/root/.fetchmailrc
# pidfile: /var/run/gpm.pid
# Fuente de funciones
. /etc/init.d/functions
# Obtenemos configuración
. /etc/sysconfig/network
# Verificamos que haya conexión a red.
if [ ${NETWORKING} = "no" ]
  then
   exit 0
fi
# Comprobamos si está presente fetchmail
[ -x /usr/bin/fetchmail ]  exit 0
start() {
  echo -n "Starting mail retrieval: fetchmail "
  /usr/bin/fetchmail -f /root/.fetchmailrc
  echo "."
}
stop() {
  echo -n "Stopping mail retrieval: fetchmail "
  /usr/bin/fetchmail -q
  echo "."
}
case "$1" in
 start)
 start;;
 stop)
 stop;;
 status)
 status fetchmail
```

[43] Si hemos escrito la línea

```
set daemon 300
```

se quedará cargado como demonio y nos bajará el correo cada 5 minutos. Si no la hemos añadido, tendremos que ejecutar este comando cada vez que deseemos bajarnos el correo.

```
 RETVAL=$?;;
*)
echo "Usage: /etc/init.d/fetchmail {start|stop|status}" >&2
exit 1;;
esac
exit 0
```

y añadirlo al nivel de arranque deseado.

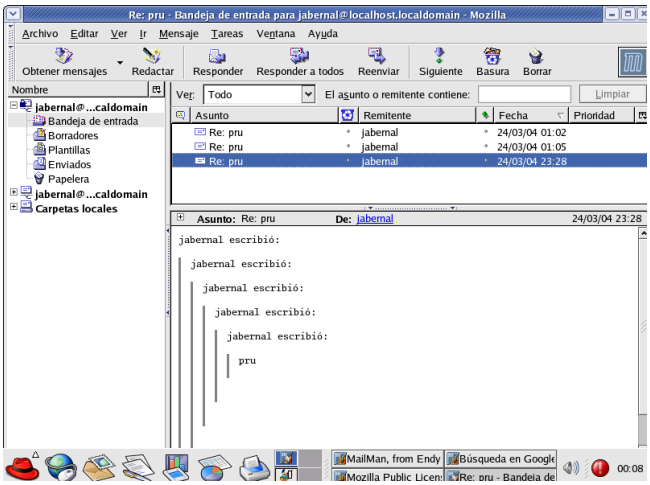
2.4. Agentes de usuario: Mozilla Mail y Ximian Evolution

Como agentes de usuario, elegiremos Mozilla Mail y Ximian Evolution. Aunque cada uno se sentirá más cómodo con su cliente de correo favorito. Los presentados aquí son a efectos de mostrar la configuración en ambos.

2.4.1. Mozilla Mail

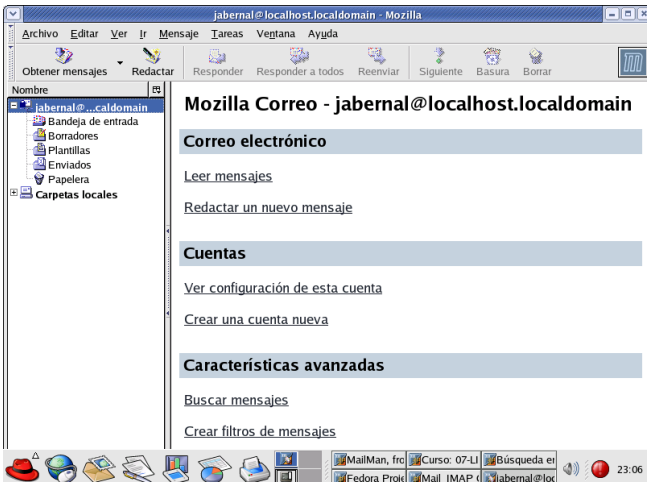
El de Mozilla tiene prácticamente todo lo que podemos necesitar y podemos utilizarlo tanto en sistemas Linux como en sistemas Windows.

Tras lanzar Mozilla y seleccionar **Ventana→Correo y Noticias**, nos aparece la ventana del cliente de correo de Mozilla.



Mozilla Mail, que así se llama el cliente de correo de Mozilla, permite gestionar varias cuentas de correo simultáneamente. Vemos que en la parte izquierda de la ventana aparecen las distintas cuentas de correo con sus carpetas correspondientes. En la parte derecha disponemos del área de mensajes, que nos muestra la lista de mensajes y el cuerpo de los que seleccionemos.

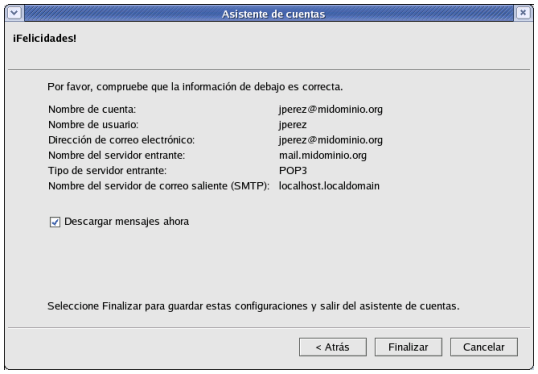
Para crear una cuenta nueva, nos situamos en alguna de las cuentas de correo existentes y nos presentará las opciones principales del cliente de correo.



También desde **Editar→Configuración de cuentas de correo y noticias**, podemos añadir acceso a una nueva cuenta de correo electrónico. La primera elección es de si se trata de una cuenta de correo electrónico o de noticias (*News*[44])

Pasamos a detallar nuestra identidad para esa cuenta de correo, indicando nuestro nombre y dirección de correo electrónico de la que se trata.

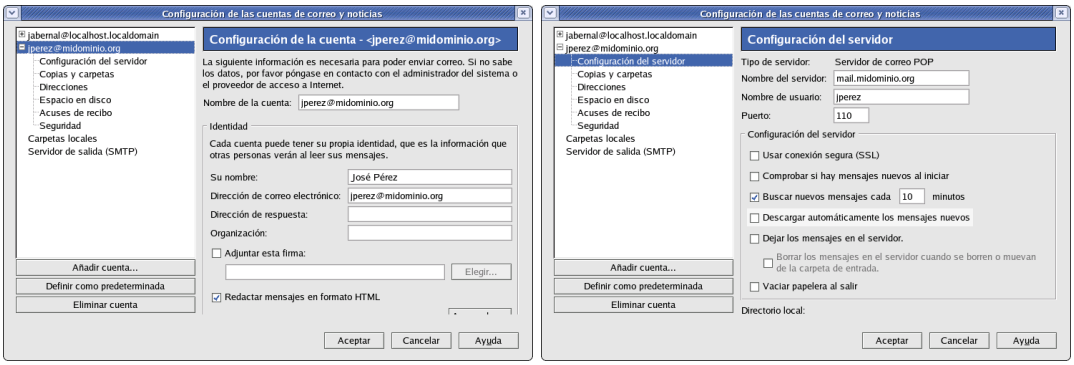
Seleccionamos a continuación si el acceso para descargarnos los correos estará disponible por POP o por IMAP, y a qué servidor nos conectaremos.



El proceso nos muestra los datos introducidos antes de aceptar la configuración.

Posteriormente, podremos acceder a los datos de la cuenta para comprobarlos o modificarlos en caso necesario, mediante la **Configuración de las cuentas de correo y noticias**.

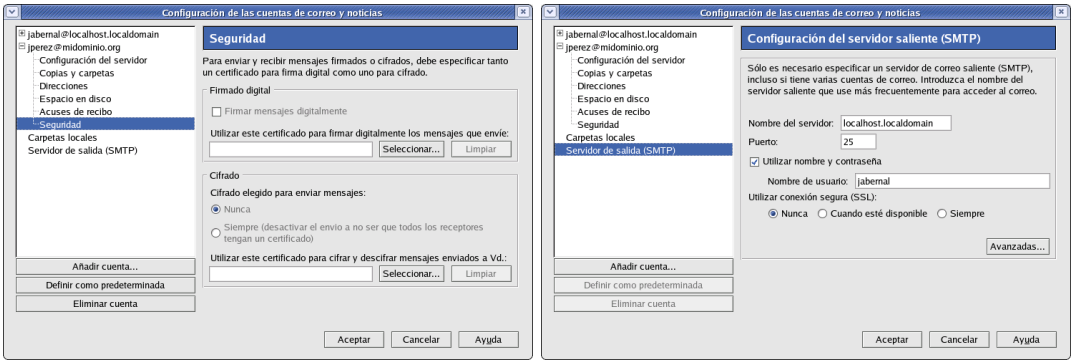
[44] Hoy día la mayor parte de grupos de news son accesibles mediante navegador web, no siendo necesarias cuentas de news específicas.



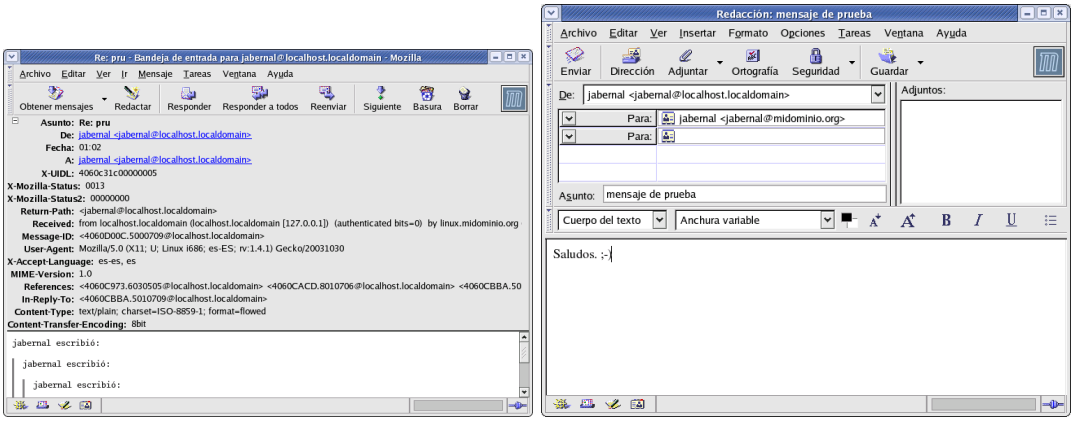
En la opción de Configuración del servidor podremos cambiar las opciones del servidor POP (o IMAP) si hubiera sido el caso. Podemos modificar el servidor, el nombre de usuario que se conecta, el puerto (que por defecto es 110), y especificar una conexión segura mediante POPS. Para POP, la opción de dejar los mensajes en el servidor nos puede permitir acceder desde distintos lugares (desde el trabajo, desde casa...) y acceder a los mensajes. Si no los dejamos en el servidor y los descargamos a un cliente, por ejemplo desde el trabajo, ya no podremos verlos desde otro lugar. Para no cargar demasiado el servidor, podemos dejar solamente aquellos de los últimos 10 días, o borrarlos al dejar la **Bandeja de Entrada** (o *Inbox*).

En las opciones de seguridad, podemos optar por firmar y cifrar digitalmente los mensajes de correo, utilizando certificados digitales x509.v3.

Pasemos a las opciones del servidor SMTP, que nos servirá para enviar el correo. Especificamos el nombre del servidor y el puerto en el que escucha, nomalmente el 25. Si hemos configurado SMTP_AUTH, podemos indicarle el nombre del usuario y la contraseña, para que nos permita utilizarlo. Además, tenemos la opción de utilizar el cifrado SSL.



En la siguiente ventana, se muestra un mensaje mostrando todos los campos de la cabecera y parte del cuerpo del mensaje.

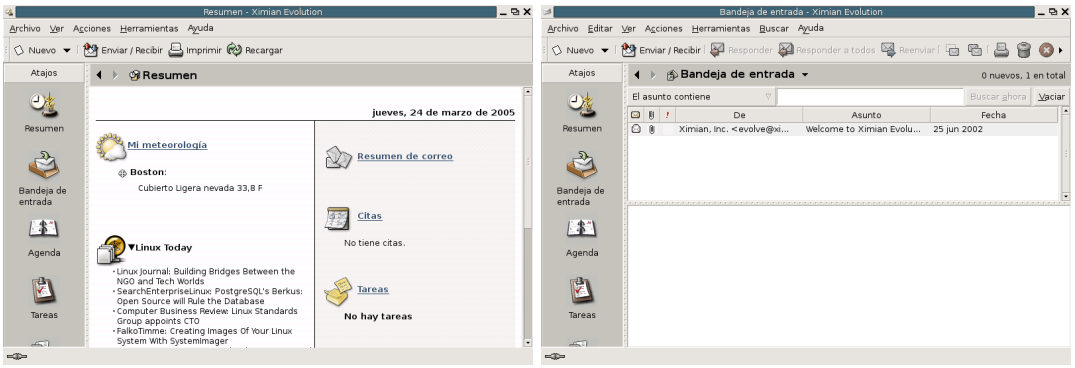


La opción de **Redactar** nos permite componer un nuevo mensaje, que enviaremos mediante la conexión al servidor de correo y el protocolo SMTP.

2.4.2. Agente de Usuario: Ximian Evolution

Otro agente de usuario es Ximian Evolution. La elección de éste está motivada porque es el que instala por defecto Guadalinex, es bastante completo y tiene todo lo que necesitamos.

Tras lanzarlo, una vez configurada la cuenta inicial, nos aparecerá la siguiente ventana:



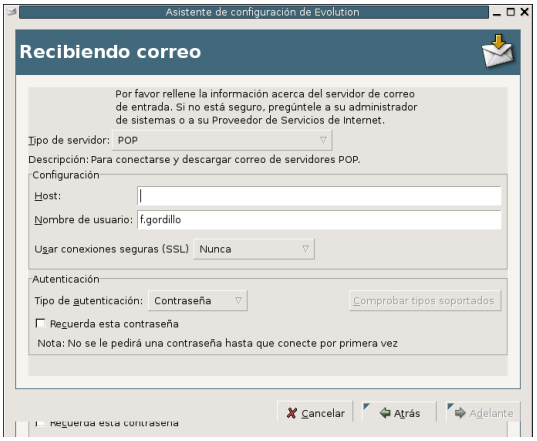
Como podemos observar, Ximian Evolution, además de la opción de cliente de correo, nos ofrece la posibilidad de gestionar una agenda, ... En nuestro caso lo que nos interesa son sus capacidades como cliente de correo, para ello en la parte derecha seleccionaremos **Resumen de Correo**. Esta opción nos mostrará el área correspondiente al cliente de correo:

Ximian Evolution permite gestionar varias cuentas de correo simultáneamente. Si seleccionamos **Ver→Barra de Carpetas**, vemos que en la parte izquierda de la ventana aparecen las distintas cuentas de correo con sus carpetas correspondientes. En la parte derecha disponemos del área de mensajes, que nos muestra la lista de mensajes y el cuerpo de los que seleccionemos.

La configuración inicial es similar a la necesaria para crear una cuenta nueva. Para crear la nueva cuenta en las opciones de menú seleccionamos **Herramientas→ Configuración**, se muestra la ventana de configuración:

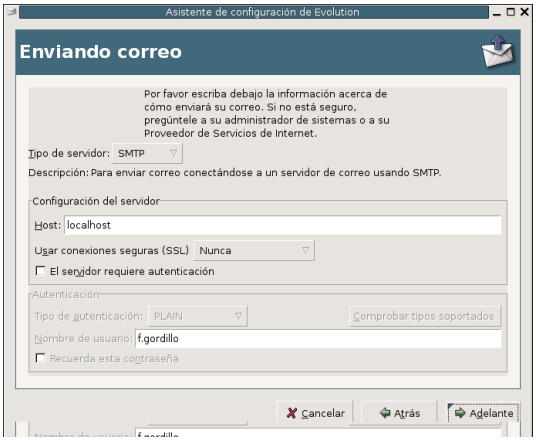
Seleccionamos la opción **Cuentas de correo**, a la derecha se muestran las cuentas de correo ya configuradas. Para añadir una nueva cuenta pulsamos **Añadir**: Se inicia el asistente de configuración para cuentas de correo, que nos aparece al iniciarlo por primera vez y que nos guiará de una forma muy intuitiva para configurar nuestra cuenta de correo:

En la siguiente pantalla podemos configurar nuestra identidad, para ello proporcionaremos nuestro nombre y nuestra dirección de correo:



A continuación proporcionaremos la información relativa al servidor de correo de entrada. Indicamos si el acceso al servidor está disponible por IMAP o POP (IMAP), dónde se encuentra nuestro servidor de correo (localhost), cuál es nuestro usuario en el servidor (mileto) y qué tipo de contraseña emplearemos. Además, si hemos configurado un canal seguro para las comunicaciones, indicaremos que se emplee siempre SSL:

Configuramos el servidor de correo saliente, debemos indicar la ubicación (localhost), el nombre de usuario y el tipo de autenticación, dado que nuestro servidor requiere autenticación y además indicaremos que se usen conexiones seguras siempre:

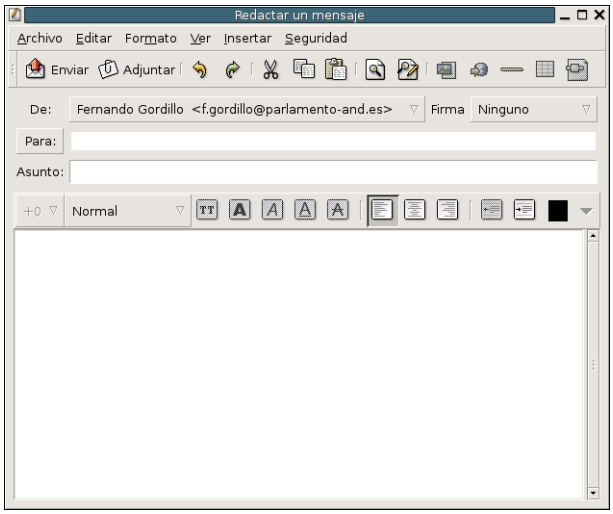


Finalmente, se nos pregunta acerca del nombre que emplearemos para identificar nuestra cuenta en el espacio de trabajo:

Para concluir, lo único que nos queda es suscribirnos a las carpetas que nos interesen de nuestro buzón de correo. Recordemos que cuando usamos IMAP, las carpetas y mensajes están directamente en el servidor. Imaginemos que de nuestra estructura de carpetas sólo nos interesa gestionar el contenido de alguna en concreto en el ordenador donde estemos trabajando, por ejemplo: si estamos en el ordenador del trabajo, quizás nos interese suscribirnos a la carpeta **Trabajo** donde almacenemos los correos de trabajo y en el ordenador de casa suscribirnos a la carpeta **Personal**. En nuestro caso aún no hemos creado carpeta por lo que nos suscribiremos a la carpeta por defecto INBOX. Seleccionamos el menú **Herramientas→Suscribirse a carpetas**...:

Seleccionamos el servidor, y la carpeta a la que deseamos suscribirnos y pulsamos [**Suscribir**]. A continuación pulsamos [**Actualizar**] para que los cambios se apliquen en el área de trabajo. Cuando nos suscribimos a una carpeta, sólo vemos el contenido de esa carpeta, no el de las carpetas hijas. Aunque en nuestra área de trabajo se muestre la estructura jerárquica, hasta dicha carpeta. Por ejemplo, si tenemos una carpeta Curso, que contiene dos carpetas Tema1 y Tema2, nos podremos suscribir a Curso, o a Tema1 o a Tema2. En cualquier caso sólo veremos el contenido de la carpeta a la que nos suscribamos, es decir, no por suscribirnos a la carpeta Curso tendremos acceso al contenido de Tema1 y Tema2.

Por último, comprobaremos que la configuración es correcta. Para ello nos enviamos a nosotros mismos un correo. En el menú seleccionamos **Nuevo**, se abre una ventana para que redactemos el correo:



Redactamos el correo destinado a nuestra cuenta y pulsamos **Enviar**. Antes de enviar, se nos solicita la contraseña del nuestro usuario en el servidor de correo, la escribimos y seleccionamos la opción de **Recuerda esta contraseña** si nos interesa.

Desde el área de trabajo pulsamos **Enviar/Recibir** para descargar el correo, que automáticamente aparecerá en nuestra bandeja **INBOX**.

2.5. Luchemos contra el SPAM: amavisd-new y spamassassin

Amavisd-new Amavisd-new[45] es una interfaz entre el MTA y uno o más filtros de contenidos, como puede ser un antivirus o un módulo antispam[46], como `SpamAssassin`. Está escrito en Perl y se comunica con el MTA vía (E)SMTP, LMTP, o mediante el uso de otros programas. No existen problemas de sincronización en su diseño que pudieran causar pérdidas de correos.

[45] Las secciones 2.5 y parte de la 2.6 están basadas en el tutorial que tenéis a vuestra disposición en `http://www.linuxsilo.net/articles/postfix.html`. Se han adecuado los contenidos a los objetivos del curso.
[46] O correo publicitario no deseado

Cuando está habilitado el uso de `SpamAssassin` (SA), se llama a SA una sola vez por mensaje (independientemente del número de destinatarios). En esta entrega veremos el funcionamiento con spamassassin y en una entrega posterior veremos cómo integrarlo con el antivirus clamav.

SpamAssassin SpamAssassin es un filtro de correo que trata de identificar el spam 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, utiliza un amplio abanico de pruebas heurísticas en las cabeceras y el cuerpo de los correos para identificar el spam, también conocido como correo electrónico comercial no solicitado. Una vez identificado, el correo puede ser opcionalmente marcado como spam y más tarde filtrado usando el cliente de correo del usuario. SpamAssassin normalmente identifica acertadamente entre un 95 y un 99 % del spam, dependiendo del tipo de correo que se reciba.

2.5.1. Instalación de SpamAssassin

Para instalarlo, los paquetes que necesitamos son[47] `spamassassin` y `spamc`:

```
# apt-get install spamassassin spamc
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Se instalarán los siguientes paquetes extras:
 libdigest-sha1-perl
Paquetes sugeridos:
 libnet-smtp-perl libmail-spf-query-perl razor libnet-ident-perl libio-
socket-ssl-perl  libdbi-perl dcc-client pyzor
Paquetes recomendados
 libnet-dns-perl
Se instalarán los siguientes paquetes NUEVOS:
 libdigest-sha1-perl spamassassin spamc
0 actualizados, 3 se instalarán, 0 para eliminar y 566 no actualiza-
dos.
Necesito descargar 847kB de archivos.
Se utilizarán 2683kB de espacio de disco adicional después de desempa-
quetar.
¿Desea continuar? [S/n]
```

Debido a que la configuración con la cual se ejecuta SpamAssassin al ser llamado desde Amavisd-new es la que se establece en el fichero de configuración de este último, no tendremos que tocar nada. En Debian[48] se trata del fichero `/etc/default/spamassassin`. Podemos observar que el demonio de SpamAssassin no se ejecuta por defecto[49], que es el comportamiento que nos conviene:

```
ENABLED=0
OPTIONS="-c -m 10 -a -H"
```

La ventaja principal de ejecutarlo como demonio sería su eficiencia, pues las comunicaciones se establecerían a través del puerto 783 en lugar de tener que arrancar un ejecutable cada vez que se tuviera que analizar un correo. En cambio, se correrían ciertos riesgos de seguridad, pues el paquete Debian nos deja una configuración por defecto que hace que se ejecute como root (en la

[47] Sólo el primero en Fedora

[48] **Fedora:** `/etc/sysconfig/spamassassin` y en este caso sí se ejecuta por defecto como demonio.

[49] Si deseamos que se ejecute como demonio optaremos por:

```
ENABLED=1
```

documentación se explica cómo cambiarlo para que se ejecute como un usuario no privilegiado). Entonces, una posible vulnerabilidad a causa de un error en el código podría darnos permisos de root. En cambio, debido a que se usará SpamAssassin a través de Amavisd-new, éste será llamado a través del módulo de Perl `Mail::SpamAssassin`, manteniendo Perl el motor de reglas siempre cargado en memoria y consiguiendo la misma eficiencia que con el demonio. De hecho, éste es el comportamiento por defecto de los paquetes Debian de estas dos aplicaciones.

Si se desean utilizar los filtros bayesianos del SpamAssassin, y es muy recomendable hacerlo si se quiere tener un alto porcentaje de acierto, será preciso entrenarlo. Según el manual, varios miles de mensajes deben ser proporcionados a SpamAssassin, tanto de spam como de *ham* (que es el correo bueno, el que no es spam). Para ello se usa la herramienta `sa-learn` . Con

sa-learn –spam <directorio>

lo instruimos para que recoja información de correos que sabemos con certeza que son spam, y con

sa-learn –ham <directorio>

lo instruimos para que recoja información de correos que sabemos con certeza que no son spam. Asimismo, `sa-learn` tiene una opción que permite pasarle un fichero que contenga una lista de directorios, uno en cada línea, en los cuales buscará el tipo de correo que le especifiquemos. Este parámetro, `--folders=file`, es muy útil si queremos recoger una lista de buzones de usuarios que sabemos con seguridad que sólo guardan spam o ham y utilizarlos para continuamente mejorar nuestros filtros desde un job del cron, pues esta herramienta mantiene una lista de los correos que ya ha analizado y se los salta cada vez, haciendo este proceso bastante eficiente. Es importante tener en cuenta que la base de datos bayesiana se encuentra en `/var/lib/amavis/.spamassassin`, pues SpamAssassin es llamado a través de Amavisd-new (módulo `Mail::SpamAssassin` de Perl). Por lo tanto, cuando queramos usar la herramienta `sa-learn` deberemos hacerlo siempre con el usuario amavis.

2.5.2. Instalación de Amavisd-new

Para instalarlo en Debian, tan sólo es necesario instalar un paquete[50], como root:

```
root@guadalinex:/home/mowgli# apt-get install amavisd-new
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Se instalarán los siguientes paquetes extras:
libarchive-tar-perl libarchive-zip-perl libcompress-zlib-
perl libconvert-tnef-perl libconvert-uulib-perl libio-multiplex-
perl libio-string-perl libio-stringy-perl libio-zlib-perl libmailtools-
perl libmime-perl libnet-server-perl libtimedate-perl libunix-syslog-
perl Paquetes sugeridos: zoo nomarch apt-listchanges libmail-audit-
perl libio-socket-ssl-perl
Se instalarán los siguientes paquetes NUEVOS:
amavisd-new libarchive-tar-perl libarchive-zip-perl libcompress-zlib-
perl libconvert-tnef-perl libconvert-uulib-perl libio-multiplex-
perl libio-string-perl libio-stringy-perl libio-zlib-perl libmailtools-
perl libmime-perl libnet-server-perl libtimedate-perl libunix-syslog-
```

[50] El paquete para Fedora lo podemos encontrar en `http://dag.wieers.com/packages/amavisd-new/`. Si lo bajamos y deseamos instalarlo con el comando rpm presenta múltiples problemas de dependencias. Así que en este caso, lo más sencillo es usar los repositorios de la distribución anterior e instalarlo después. Para eso, añadiremos la línea

```
rpm http://apt.sw.be fedora/3/en/i386 dag
```

al fichero `/etc/apt/source.list` de Fedora y después ejecutamos:

```
#apt-get update
#apt-get install amavisd-new
```

```
perl 0 actualizados, 15 se instalarán, 0 para eliminar y 562 no actua-
lizados. Necesito descargar 1385kB de archivos.
Se utilizarán 4592kB de espacio de disco adicional después de desempa-
quetar.
¿Desea continuar? [S/n]
```

El fichero de configuración de Amavisd-new es bastante largo, pero tan sólo es necesario realizar unos pocos cambios a la configuración que viene por defecto. A continuación se presentan las líneas del fichero `/etc/amavis/amavisd.conf` que necesitan ser modificadas, en el formato definitivo (es decir, con las modificaciones ya realizadas):

```
$mydomain = 'midominio.com';
$myhostname = 'mail.midominio.com';
# @bypass_spam_checks_acl = qw( . );
$final_spam_destiny = D_PASS;
$warnbannedsender = 1;
$warnbadhsender = 1;
# $virus_quarantine_to = 'virus-quarantine';
$virus_quarantine_to ="virus-quarantine\@$mydomain";
# $sa_spam_subject_tag = '***SPAM*** ';
$banned_filename_re = new_RE(
# qr'^UNDECIPHERABLE$',
  qr'\.[^.]*\.(exe|vbs|pif|scr|bat|cmd|com|dll)$'i,
  qr'[{}]',
# qr'.\.(exe|vbs|pif|scr|bat|cmd|com)$'i,   qr'.\.(ade|adp|bas|bat|chm|cmd|com|cpl|crt|exe|hlp|l
   js|jse|lnk|mdb|mde|msc|msi|msp|mst|pcd|pif|reg|scr|sct|shs|shb|vb|
   vbe|vbs|wsc|wsf|wsh)$'ix,
  qr'.\.(mim|b64|bhx|hqx|xxe|uu|uue)$'i,
# qr'^\.(zip|lha|tnef|cab)$'i,
  qr'^\.exe$'i,
  qr'^application/x-msdownload$'i,
  qr'^application/x-msdos-program$'i,
  qr'^message/partial$'i,
  qr'^message/external-body$'i,
);
```

Estas modificaciones realizadas al fichero dejan una configuración específica para que amavisd-new se comporte de la manera que nosotros queremos, pero lo más habitual es que deba personalizarse para cada caso. Acto seguido se resumen los porqués de los cambios realizados:

- `bypass_spam_checks_acl`: comentamos esta línea para que Amavisd-new use SpamAssassin (por defecto viene deshabilitado su uso).
- `final_spam_destiny`: dejamos pasar los correos identificados como spam, aunque siguen siendo marcados como tales mediante cabeceras en el correo. De este modo, los destinatarios seguirán recibiendo toda su correspondencia pero podrán filtrarla fácilmente usando el cliente de correo y las cabeceras que Amavisd-new habrá añadido al mensaje.
- `warnbannedsender`: activamos el envío de un mensaje de aviso al remitente de un mensaje que contuviera algún fichero adjunto con una de las extensiones prohibidas que más abajo se detallan.
- `warnbadhsender`: igual que el anterior para ficheros con cabeceras mal formadas.
- `virus_quarantine_to`: activamos la cuarentena de los correos con virus. De este modo, cualquier correo que contenga un virus detectado, será redirigido a la cuenta especificada. Así, podremos revisarlos y decidir qué hacer con ellos.

- `sa_spam_subject_tag`: al comentar esta sentencia se desactiva la modificación del asunto del mensaje, pues con las cabeceras que se han añadido es suficiente para que nuestro cliente de correo filtre adecuadamente.

- `banned_filename_re`: rechazamos correos que contengan ficheros adjuntos con alguna de las extensiones mencionadas en esta variable (únicamente se permiten ficheros comprimidos), principalmente ejecutables y scripts.

Tras esto ya podemos reiniciar el servicio mediante el comando

```
#/etc/init.d/amavisd-new restart
```

y observar su carga en el log `/var/log/mail.log`, donde se informa de todos los módulos cargados al iniciar.

Con la instalación de amavisd vista en este apartado será inmediato instalar el antivirus clamav en una entrega posterior.

2.5.3. Modificaciones en Postfix

Las modificaciones a realizar en Postfix son muy sencillas.

- Fichero `/etc/postfix/master.cf`: Editamos el fichero y le añadimos las siguientes líneas:

```
127.0.0.1:10025 inetn   -   n   -   -   smtpd
-o content_filter=
-o local_recipient_maps=
-o relay_recipient_maps=
-o smtpd_restriction_classes=
-o smtpd_client_restrictions=
-o smtpd_helo_restrictions=
-o smtpd_sender_restrictions=
-o smtpd_recipient_restrictions=permit_mynetworks,reject
-o mynetworks=127.0.0.0/8
-o strict_rfc821_envelopes=yes
-o smtpd_error_sleep_time=0
-o smtpd_soft_error_limit=1001
-o smtpd_hard_error_limit=1000
smtp-amavis unix - - n - 2 lmtp
-o lmtp_data_done_timeout=1200
-o lmtp_send_xforward_command=yes
```

- Fichero `/etc/postfix/main.cf`: En este fichero debemos añadir la siguiente línea:

```
content_filter = smtp-amavis:[127.0.0.1]:10024
```

Con estas modificaciones, lo que conseguimos es añadir un filtro de contenido a Postfix, el cual redirigirá el tráfico al puerto 10024 de la interfaz loopback. Una vez amavisd-new haya finalizado su trabajo, devolverá el mensaje a Postfix a través del puerto 10025, donde hemos habilitado un smtpd.

Ahora tan sólo queda reiniciar el servidor Postfix:

```
# /etc/init.d/postfix restart
Stopping mail transport agent: Postfix.
Starting mail transport agent: Postfix.
```

Para comprobar que todo está funcionando correctamente, podemos enviar un correo a un usuario de nuestro servidor y comprobaremos que se está filtrando el contenido a través de amavisd-new, observando el fichero de log `/var/log/mail.log`.

Una medida de seguridad que podemos tomar es utilizar otros puertos diferentes. Tendremos que ir afinando la configuración de amavisd en lo que respecta a las políticas de filtrado, una primera forma de comenzar es desechando los virus y permitiendo, pero con una modificación de las cabeceras, el spam y los correos no permitidos ("banned").

2.6. Gestores de listas de correo: Mailman

2.6.1. ¿Qué es una lista de correo?

Una lista de correo es simplemente una dirección de correo que contiene un grupo de direcciones a las cuales se envía la misma información. En el caso de las listas de correo electrónico, se usa una lista de direcciones de correo electrónico de gente interesada en escuchar o discutir sobre un tema determinado.

Entre los tipos más comunes de listas de correo electrónico están las listas de anuncios y las listas de discusión.

Las listas de anuncios sirven para que una o más personas puedan informar a un grupo más numeroso de personas.

Una lista de discusión permite a un grupo de personas, debatir temas entre ellos mismos, pudiendo cada uno enviar mensajes a la lista y hacer que se distribuyan a todos los integrantes del grupo. Esta discusión también se puede moderar, de manera que sólo los mensajes a los cuales el administrador les haya dado el visto bueno serán distribuidos a la lista. También es posible hacer que sólo a ciertas personas se les permita enviar mensajes a la lista. Dando lugar a la división entre las listas abiertas, en las que cualquiera puede enviar, y las listas cerradas, en las que sólo sus integrantes pueden enviar información.

Algunos términos comunes son:

- Un "envío" denota un mensaje que se envía a una lista de correo.
- A las personas que son parte de una lista de correo electrónico normalmente se las llama "suscriptores" de la lista.
- "Los administradores de las listas" son personas encargadas de mantener esas listas. Las listas pueden tener uno o más administradores.
- Una lista puede tener también personas encargadas de leer los mensajes enviados a la lista y decidir si éstos deberían ser distribuidos a todos los suscriptores. A estas personas se les llama moderadores.
- A menudo varias listas de correo electrónico utilizan el mismo software. A la persona que mantiene el software gracias al cual funcionan las listas se le llama el "administrador del sitio." A menudo el administrador del sitio también administra listas individuales.

2.6.2. Mailman

GNU Mailman es un programa que permite administrar listas de correo electrónico, con soporte para un rango amplio de tipos de listas de correo, tales como listas de discusión general y listas de sólo anuncios. Mailman tiene características para los suscriptores, tales como: facilidad en la suscripción y desuscripción, opciones de privacidad, y capacidad de detener temporalmente la recepción de los envíos a la lista.

Mailman también tiene muchas características para facilitar la tarea a los administradores de listas y a los administradores de sitio.

Instalación y Configuración

Para la instalación de mailman vamos a seguir los pasos en Guadalinex 2004 y posteriormente comentaremos las diferencias con Fedora.

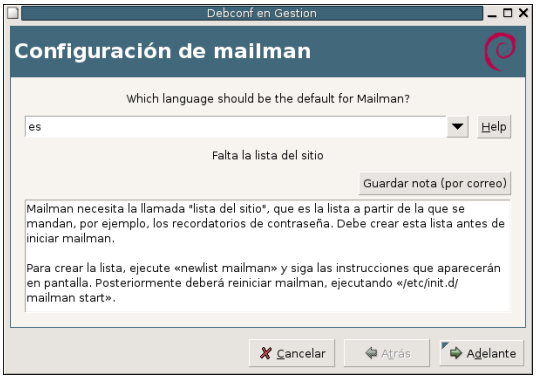
Antes de realizar la instalación de mailman debemos tener en cuenta las dependencias de dicho paquete y el gestor de correo que se va a utilizar. En este caso, y siguiendo el orden del curso, se va a utiliar como MTA el software postfix, supuestamente ya instalado.

En el momento de escribir esta documentación, la versión que utiliza guadalinex para mailman es la 2.1.5-1, si realizamos la instalación:

```
# apt-get install mailman
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Se instalarán los siguientes paquetes extras:
  pwgen
Paquetes sugeridos:
  python2.2-korean-codecs python2.3-korean-codecs python-japanese-
codecs
Se instalarán los siguientes paquetes NUEVOS:
  mailman pwgen
0 actualizados, 2 se instalarán, 0 para eliminar y 119 no actualizados.
Necesito descargar 6618kB de archivos.
Se utilizarán 32,4MB de espacio de disco adicional después de desempa-
quetar.
¿Desea continuar? [S/n] s
```

Mailman dependende además de Apache, nosotros partiremos de la premisa de que Apache 2 ya está instalado.

Durante la instalación nos aparecerán las pantallas de postconfiguración para el idioma, que dará igual lo que seleccionemos porque se instalará en inglés, y posteriormente nos indica que debemos crear la lista mailman antes de arrancar.



Tras la instalación del paquete, es conveniente leer la documentación disponible en `/usr/share/doc/mailman`, principalmente en `/usr/share/doc/mailman/README.Debian.gz` y en `/usr/share/doc/mailman/README.POSTFIX.gz`. En ella se detallan las modificaciones necesarias en Apache para el correcto funcionamiento de la interfaz web, así como la forma de integrar Postfix y Mailman.

A continuación debemos modificar ligeramente el fichero `/etc/mailman/mm_cfg.py` a fin de que Mailman sepa que se está trabajando con Postfix como *Mail Transport Agent*, para eso descomentamos la siguiente línea:

```
MTA = 'Postfix'
```

La configuración por defecto de Postfix nos deja la directiva `alias_maps` apuntando a `/etc/aliases`. Ya que no nos interesa estar modificando este fichero y ejecutando el comando newaliases de Postfix cada vez que creemos o borremos una lista, utilizaremos el fichero de alias propio de Mailman, que es automáticamente actualizado por los comandos `newlist` y `rmlist`.

El primer paso será generarlo:

```
# cd /var/lib/mailman
# bin/genaliases
```

A continuación añadiremos ese fichero de alias a la directiva `alias_maps` del `/etc/postfix/main.cf`, además de otras directivas necesarias:

```
alias_maps=hash:/etc/aliases,hash:/var/lib/mailman/data/aliases
mailman_destination_recipient_limit = 1
unknown_local_recipient_reject_code = 550
owner_request_special = no
recipient_delimiter = +
```

Algunas de estás opciones estarán ya definidas previamente.

Y solicitaremos a Postfix que recargue la configuración:

```
#/etc/init.d/postfix reload
```

El tercer paso de la instalación de Mailman nos avisa de que es necesario crear una *sitelist* llamada mailman y que hasta que no la creemos, el demonio del Mailman no arrancará. Ahora es el momento de crearla y, para ello, ejecutamos el siguiente comando:

```
# newlist mailman
Enter the email of the person running the list: mileto@midominio.com
Initial mailman password:
Hit enter to notify mailman owner...
```

Nótese que Mailman no nos muestra la lista de alias que nos requiere que añadamos a nuestro fichero de alias. Esto es debido a la configuración realizada más arriba, eliminándose de esta manera este tedioso paso. Podemos, por lo tanto, pulsar enter y pasar a iniciar el demonio de Mailman mediante el comando

```
#/etc/init.d/mailman start
```

Una vez iniciado el servicio, recibiremos el correo que nos notifica la creación de la lista en la dirección de correo que hayamos especificado (mileto@midominio.com en este ejemplo).

Las listas que creemos en el futuro tampoco nos solicitarán que añadamos manualmente la lista de alias. Cuando se añada o quite una lista, el fichero `/var/lib/mailman/data/aliases.db` será automáticamente actualizado, pero no se ejecutará automáticamente un

```
#/etc/init.d/postfix reload
```

Esto es debido a que es necesario ser root para ejecutar este comando y los scripts suid-root no son seguros. El único efecto de esto es que le llevará aproximadamente un minuto a Postfix darse cuenta de los cambios y actualizar sus tablas, si bien esto se puede considerar una inconveniencia menor.

Por último nos queda confirmar que se puede acceder vía web. Este paso dependerá del apache instalado y la configuración que tenga. Los scripts de mailman se encuentran en `/usr/lib/cgi-bin/mailman/`, también se necesita acceso a los archivos de la lista en `/var/lib/mailman/archives/public/` y a las imagenes `/usr/share/images/mailman/`. Según el apache que se esté utilizando, habilitaremos la ejecución de cgi's mediante `a2enmod cgi` (en la versión apache2). Es posible definir alias para que el acceso sea más sencillo, aquí la gestión la haremos utilizando el path por defecto.

Gestión de las listas

Normalmente será el administrador del site el que utilice la línea de comandos para gestionar la creación, propiedades y borrado de las listas. Para esto, utilizará los comandos del directorio `/usr/lib/mailman/bin/` entre los cuales están:

newlist para generar una nueva lista

add_members con el que se añaden miembros a una lista

mmsitepass para modificar passwords, por ejemplo -c para el creador de listas

list_* serie de comandos para listar los elementos listas, miembros,...

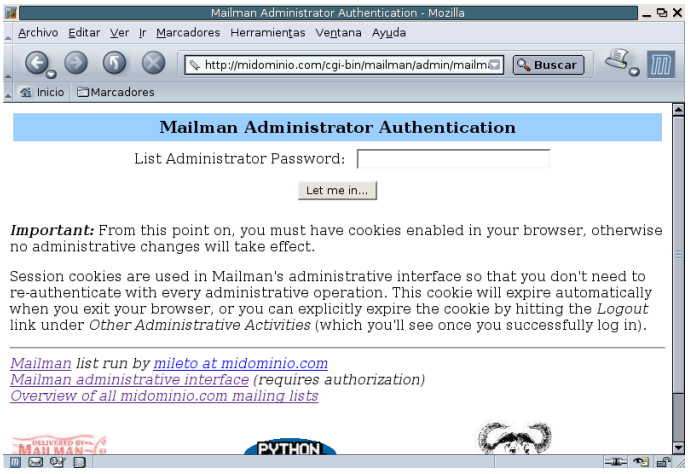
remove_members borrar miembros de listas

rmlist borrar listas

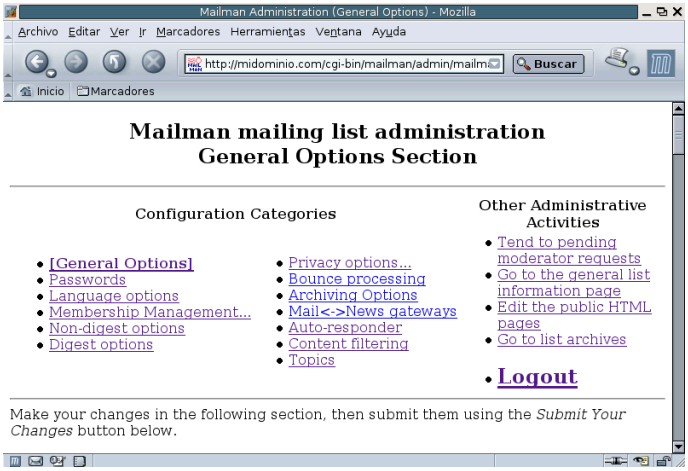
Los ficheros de log se encuentran en `/var/log/mailman/` y los de configuración en `/etc/mailman` y `/var/lib/mailman`. Además del administrador del site que gestionará los ficheros y comandos anteriores, existen otros roles con privilegios como el administrador de una lista que controla todo lo referente a dicha lista, el moderador de una lista que aprueba los mensajes para su envío y el creador de listas que puede crear listas aunque no gestionarlas.

Todos los roles al igual que los usuarios de las listas podrán utilizar el correo electrónico para algunas operaciones así como utilizar el interfaz web. En nuestro caso, vamos a ver con mayor detalle el interfaz web y dado que nuestro dominio lo hemos puesto como midominio.com vamos a hacer que en `/etc/hosts` ese dominio apunte al localhost o bien modificamos el fichero `/etc/mailman/mm_cfg.py` para indicar la URL del Host. De esta forma accedemos a

`http://localhost/cgi-bin/mailman/admin/nombre_lista`



Con la contraseña de administración, accedemos al menú donde podemos configurar todos los parámetros de la lista.



Para crear nuevas listas se aconseja no habilitar el uso del creador de listas por motivos claros de seguridad, es aconsejable crearla en la línea de comandos y posteriormente configurarla vía web.

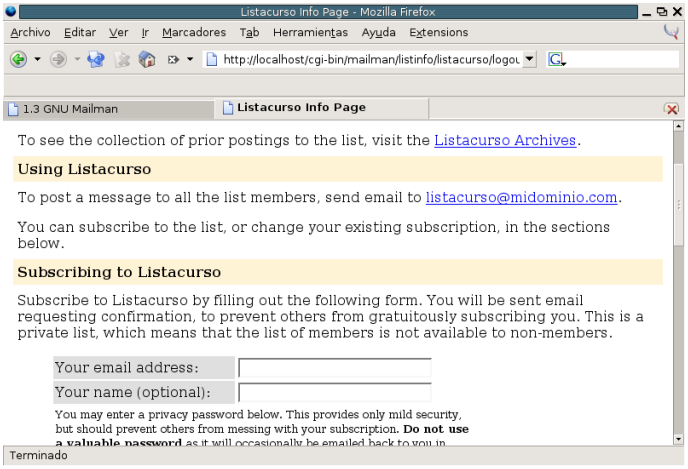
Ejemplo de ello:

```
#newlist listacurso
```

Desde

`http://midominio.com/cgi-bin/mailman/admin/listacurso` configuamos la lista y desde

`http://midominio.com/cgi-bin/mailman/listinfo/listacurso` permite que el usuario se de alta y de baja en la lista



Con más detalle tenemos:

Página de información de la lista (listinfo) Usualmente se encuentra en `http://SERVIDORWEB/mailman/listinfo/NOMBRELISTA`

La página listinfo es el punto de inicio del interfaz del suscriptor. Como se podría asumir por el nombre dado, esta página contiene información acerca de la lista NOMBRELISTA. A partir de esta página se puede llegar al resto de páginas del suscriptor, así que realmente sólo se necesita conocer esta dirección.

Página de opciones del suscriptor Usualmente se encuentra en
`http://SERVIDOR/mailman/options/NOMBRELISTA/CORREO`

También se puede acceder a esta página yendo a la página listinfo y escribiendo su dirección de correo en el cuadro de texto junto al botón etiquetado "**Opciones de Edición y Desuscripción**" (éste está cerca del final de la página).

La página de opciones de suscriptor le permite entrar/salir y cambiar la configuración de sus opciones, así como también desuscribirse u obtener una copia de su contraseña por correo electrónico.

Para acceder a su página de opciones de suscriptor: Si aún no ha entrado, encontrará un cuadro de texto cerca de la parte superior de la página para introducir la contraseña. Escriba su contraseña en el cuadro de texto mencionado y haga clic en el botón "**Cambiar**".

Una vez dentro, podrá mirar y cambiar toda la configuración personal de su lista.

Archivos de la Lista Usualmente los encontrará en
`http://SERVIDORWEB/pipermail/NOMBRELISTA`
si la lista se archiva públicamente, y
`http://SERVIDORWEB/mailman/private/NOMBRELISTA`
si la lista se archiva en forma privada.

Las páginas de los archivos de la lista disponen de una copia de los mensajes enviados a la lista, normalmente agrupados por mes. En cada grupo mensual, los envíos se indexan por autor, fecha, hilo y asunto.

Toda lista de correo tiene un conjunto de direcciones de correo electrónico a las cuales se pueden enviar los mensajes, incluyendo, una dirección para enviar los mensajes a la lista, una dirección destinada a recibir mensajes devueltos y direcciones para procesar órdenes de correo.

Para una lista de correo ficticia llamada *listacurso@midominio.com*, nos encontraremos estas direcciones:

listacurso@midominio.com ésta es la dirección de correo para enviar mensajes a la lista.

listacurso-join@midominio.com enviando un mensaje a esta dirección, un nuevo miembro puede solicitar suscripción a la lista, pero si se hace, Mailman ignora tanto la cabecera de Asunto: como el cuerpo de tal mensaje.

listacurso-leave@midominio.com enviando un mensaje a esta dirección un miembro puede solicitar que se le dé de baja de la lista. Igual que con la dirección -join, se ignora la cabecera Asunto: y el cuerpo del mensaje.

listacurso-owner@midominio.com Esta dirección permite contactar con el propietario o moderador de la lista. Esta es la dirección que deberá utilizar cuando necesite contactar a la persona o personas encargadas de la lista.

listacurso-request@midominio.com Esta dirección está asociada a un robot de correo que procesa órdenes de correo electrónico que se pueden usar para definir las distintas opciones de los suscriptores, así como también para procesar otras órdenes.

listacurso-bounces@midominio.com Esta dirección se usa para el procesamiento automático de los mensajes devueltos de Mailman.

listacurso-confirm@midominio.com Esta dirección se usa para procesar mensajes de confirmación de las solicitudes de suscripción y desuscripción.

También hay una dirección -admin que permite contactar a los administradores de las listas. Esta dirección sólo existe por compatibilidad con las versiones más antiguas de Mailman.

Para cambiar las opciones, se usa la dirección `NOMBRELISTA-request`.

Para un mayor conocimiento de mailman, se recomienda la lectura de la guía de usuario y de administrador, así como posibles cambios, en `http://www.gnu.org/software/mailman`

Mailman en Fedora Core 3

Para Fedora Core 3 podemos realizar la instalación mediante rpm o bien utilizar el apt-get, con este último obtendríamos lo siguiente:

```
#apt-get install mailman
Leyendo listas de paquetes... Done
Construyendo árbol de dependencias... Done
Se instalarán los paquetes NUEVOS siguientes:
mailman
0 upgraded, 1 newly installed, 0 removed and 156 not upgraded.
Need to get 7979kB of archives.
After unpacking 27,8MB of additional disk space will be used.
Get:1 http://ayo.freshrpms.net fedora/linux/3/i386/updates mailman
3:2.1.5-30.fc
3 [7979kB]
Fetched 7979kB in 10s (754kB/s)
Committing changes...
Preparing... ##########################################[100%]
1:mailman ##########################################[100%]
Done.
```

Una vez instalado, es conveniente leer la documentación de `/usr/share/doc/mailman-version/README.POSTFIX` para ver que los pasos de la instalación son similares a los indicados en el apartado anterior. Destacar algunos cambios que, aunque poco significativos, se encuentran en esta distribución:

`/etc/mailman/sitelist.cf` Este fichero contiene parámetros de configuración por defecto de las listas de correo

`/etc/httpd/conf.d/mailman.conf` En este fichero se encuentra la configuración de apache para acceder vía web.

`/usr/lib/mailman/` Directorio en el que se encuentran los ficheros de configuración, binarios y plantillas en varios idiomas.

`/usr/share/doc/mailman-2.1.5` Directorio de documentación donde podremos leer sobre las diferentes instalaciones e integraciones.

2.7. Correo Web: SquirrelMail

> ¿Qué es exactamente SquirrelMail? Se trata de un interface, o cliente, de correo escrito en PHP4. Se ha diseñado para permitir acceso al correo a través de su servidor desde cualquier parte del mundo empleando la "web".

2.7.1. Instalación

Squirrelmail es un potente sistema de correo web. ¿Cuántas veces cuando estamos fuera de casa o del trabajo sentimos la necesidad de acceder a nuestro correo con todas sus funciones? Leer correo, enviarlo, acceder a la libreta de direcciones... Sin más programas que un navegador y acceso a Internet, podremos acceder desde cualquier sitio[51] a nuestro correo, en el caso de que tengamos nuestro servidor accesible desde Internet.

SquirrelMail es un programa de webmail escrito en PHP4 que proporciona toda la funcionalidad que esperamos de un cliente de correo, incluyendo ficheros adjuntos, libreta de direcciones y uso

[51] Desde casa, el trabajo, un cibercafé, el ordenador de un amigo...

de carpetas. Soporta los protocolos IMAP y SMTP y todas las páginas se generan en HTML 4.0 (sin Javascript) para obtener la máxima compatibilidad con los navegadores.

Las conexiones a las carpetas de usuario se realizan mediante el protocolo IMAP, que permite la creación y manipulación de carpetas en el servidor. Así que partimos de que Apache, PHP4 e IMAP ya funcionan correctamente.

Stopsign Hay que activar un servidor `imap`, aunque hay varios disponibles (Cyrus, uw-imapd...), nos decantaremos por dovecot, que funciona sin problemas tanto en Fedora como Guadalinex. Lo podemos instalar en ambos, si no lo está ya, con

```
#apt-get install dovecot
```

En Fedora

Para instalarlo, utilizamos `apt-get`. Él se encargará de instalar los paquetes necesarios.

```
[root@linux root]# apt-get install squirrelmail
Leyendo listas de paquetes... Done
Construyendo árbol de dependencias... Done
Se instalarán los paquetes NUEVOS siguientes:
squirrelmail
0 upgraded, 1 newly installed, 0 removed and 51 not upgraded.
Need to get 2585kB of archives.
After unpacking 8194kB of additional disk space will be used.
Get:1 http://ayo.freshrpms.net fedora/linux/1/i386/core squirrelmail 1.4.0-1 [2585kB]
Fetched 2585kB in 1m39s (25,9kB/s)
Committing changes...
Preparing... ########################################### [100%]
1:squirrelmail ########################################### [100%]
Done.
```

En Debian

Partimos de que se han instalado los paquetes de apache2 y php4. Después ejecutaremos:

```
root@guadalinex:~/curso-linux/3# apt-get install squirrelmail
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Se instalarán los siguientes paquetes extras:
  libapache2-mod-php4 libzzip-0-12 php4-common squirrelmail-locales
Paquetes sugeridos:
  php4-pear phpdoc imap-server
Se instalarán los siguientes paquetes NUEVOS:
  libapache2-mod-php4 libzzip-0-12 php4-
common squirrelmail squirrelmail-locales
0 actualizados, 5 se instalarán, 0 para eliminar y 608 no actualizados.
Se necesita descargar 1805kB/6377kB de archivos.
Se utilizarán 22,3MB de espacio de disco adicional después de desempa-
quetar.
¿Desea continuar? [S/n]
Des:1 http://ftp.fi.debian.org sarge/main php4-common 4:4.3.10-
9 [164kB]
Des:2 http://ftp.fi.debian.org sarge/main libapache2-mod-php4 4:4.3.10-
9 [1642kB]
Descargados 1805kB en 37s (48,1kB/s)
Seleccionando el paquete libzzip-0-12 previamente no seleccionado.
```

```
(Leyendo la base de datos ...
112469 ficheros y directorios instalados actualmente.)
Desempaquetando libzzip-0-12 (de .../libzzip-0-12_0.12.83-
4_i386.deb) ...
Seleccionando el paquete php4-common previamente no seleccionado.
Desempaquetando php4-common (de .../php4-common_4%3a4.3.10-
9_i386.deb) ...
Seleccionando el paquete libapache2-mod-
php4 previamente no seleccionado.
Desempaquetando libapache2-mod-php4 (de .../libapache2-mod-
php4_4%3a4.3.10-9_i386.deb) ...
Seleccionando el paquete squirrelmail-
locales previamente no seleccionado.
Desempaquetando squirrelmail-locales (de .../squirrelmail-
locales_1.4.4-20050122-1_all.deb) ...Seleccionando el paquete squirrel-
mail previamente no seleccionado.
Desempaquetando squirrelmail (de .../squirrelmail_2%3a1.4.4-
3_all.deb) ...
Configurando libzzip-0-12 (0.12.83-4) ...
Configurando php4-common (4.3.10-9) ...
Configurando libapache2-mod-php4 (4.3.10-9) ...
Forcing reload of web server: Apache2.
Configurando squirrelmail-locales (1.4.4-20050122-1) ...
Configurando squirrelmail (1.4.4-3) ...
Installing default squirrelmail config.
Run /usr/sbin/squirrelmail-
configure as root to configure/upgrade config.
Para configurarlo, podemos ver la sección siguiente. También exis-
te una utilidad de configuración:
#/usr/sbin/squirrelmail-configure
```

para conseguir el mismo entorno que en el gráfico 2.6 en la página 93.

2.7.2. Configuración

Empecemos la configuración. Para ello, veamos los ficheros implicados[52]:

Como ya sabemos, apache leerá los ficheros[53] del directorio `/etc/httpd/conf.d`. En este caso `squirrelmail.conf`

```
[root@linux images]# more /etc/httpd/conf.d/squirrelmail.conf
#
# SquirrelMail is a webmail package written in PHP.
#
Alias /webmail /usr/share/squirrelmail
```

Lo que obtenemos de él es que apuntando nuestro navegador a `http://localhost/webmail`[54] accedemos a la entrada del correo web, como vemos en la figura. En realidad, los ficheros necesarios están ubicados en `/usr/share/squirrelmail`.

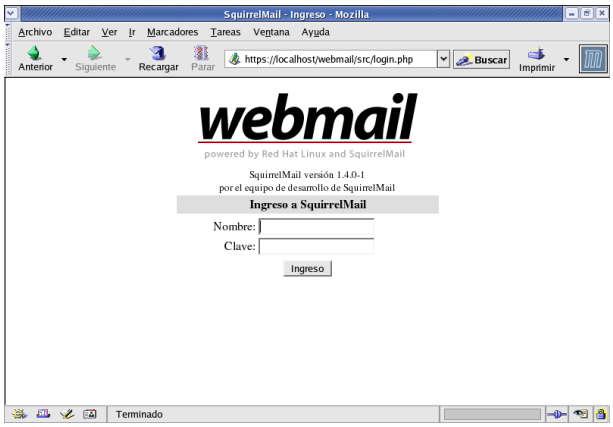
[52] Lo haremos sobre Fedora, trasladarlo a Debian es similar.

[53] En Debian, tendremos que añadir la línea

```
Alias /webmail /usr/share/squirrelmail
```

al fichero de configuración de Apache: `/etc/apache2/apache2.conf`,
O hacer el siguiente enlace: #ln -s /usr/share/squirrelmail /var/www/webmail

[54] O `http://www.midominio.com` y, lo mejor por SSL: `https://www.midominio.com`



La configuración de la herramienta se encuentra en el fichero `/etc/squirrelmail/config.php`. Veamos sus características principales:

```
<?php
/**
* SquirrelMail Configuration File
* Created using the configure script, conf.pl
*/
global $version;
$config_version = '1.4.0';
$config_use_color = 1;
$org_name = "SquirrelMail";
$org_logo = SM_PATH . 'images/sm_logo.png';
$org_logo_width = '308';
$org_logo_height = '111';
$org_title = "SquirrelMail $version";
$signout_page = ' ';
$frame_top = '_top';
$provider_uri = 'http://www.squirrelmail.org/';
$provider_name = 'SquirrelMail';
$motd = "";
//$squirrelmail_default_language = 'en_US';
$squirrelmail_default_language = 'es_ES';
```

Cambiamos a `es_ES` para que nos aparezca en castellano.

```
$domain = 'localhost';
$imapServerAddress = 'localhost';
$imapPort = 143;
$useSendmail = true;
$smtpServerAddress = 'localhost';
$smtpPort = 25;
$sendmail_path = '/usr/sbin/sendmail';
$pop_before_smtp = false;
$imap_server_type = 'other';
```

Configuramos el dominio de correo y los servidores IMAP y SMTP y puertos a los que se conecta. Con la configuración por defecto, funcionaría directamente sobre la misma máquina en la que está el servidor web, aunque podría ser otro servidor distinto.

```
$invert_time = false;
```

```
$optional_delimiter = '/';
$default_folder_prefix = 'mail/';
$trash_folder = 'Trash';
$sent_folder = 'Sent';
$draft_folder = 'Drafts';
$default_move_to_trash = true;
$default_move_to_sent = true;
```

Contiene opciones de las carpetas por defecto. Existen más opciones, pero normalmente no las tocaremos.

También podremos configurar nuestro sistema SquirrelMail mediante menús con el comando[55]

`/usr/share/squirrelmail/config/conf.pl`

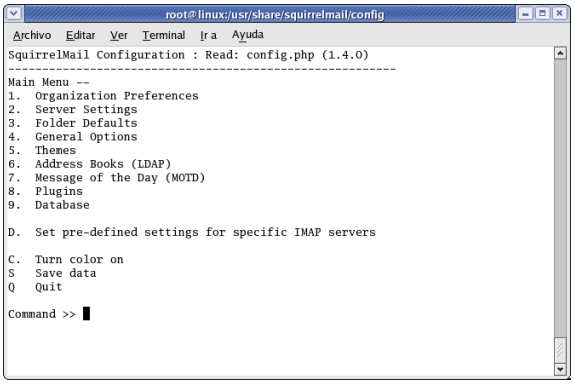


Figura 2.6: Configuración Squirrelmail

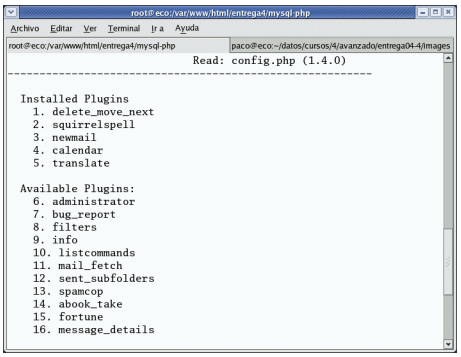
Por ejemplo, si pulsamos 1, podremos establecer el nombre de nuestro sitio, el idioma por defecto, el logo, ...:

```
Organization Preferences
1. Organization Name : Mileto
2. Organization Logo : ../images/logo.jpg
3. Org. Logo Width/Height : (500/100)
4. Organization Title : SquirrelMail $version
5. Signout Page :
6. Default Language : es_ES
```

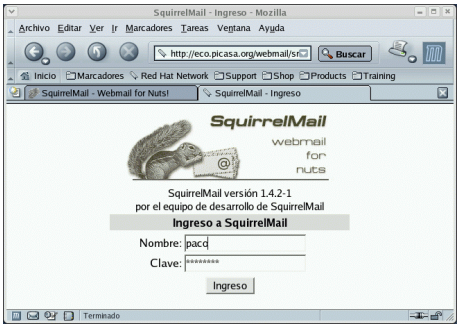
Si en el menú principal pulsamos sobre 8, accederemos a la posibilidad de añadir funcionalidades añadidas a la aplicación, por ejemplo, calendarios, filtros, etc

[55] En Debian

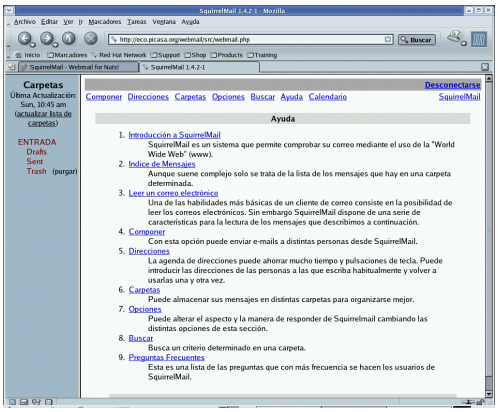
```
#/usr/sbin/squirrelmail-configure
```



Una vez que nuestro sistema está configurado a nuestro gusto, no tenemos más que conectarnos[56] e introduciendo el usuario y contraseña del correo, accedemos a éste.

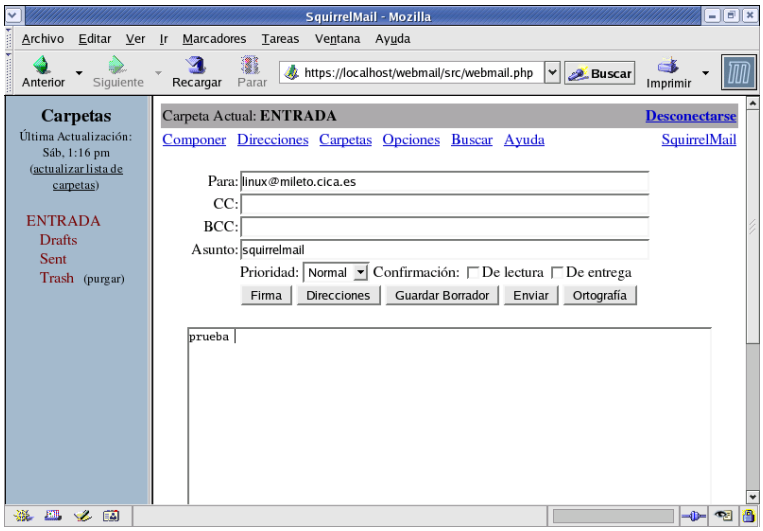


Comprobamos que las carpetas se sitúan a la izquierda y en la derecha tenemos en la parte superior el menú principal con las opciones: **Componer**, **Direcciones**, **Carpetas**, **Opciones**, **Buscar** y **Ayuda**. Si pulsamos sobre está última accederemos a la completa ayuda (en castellano) que nos permite conocer todos los aspectos relacionados con el uso del programa.

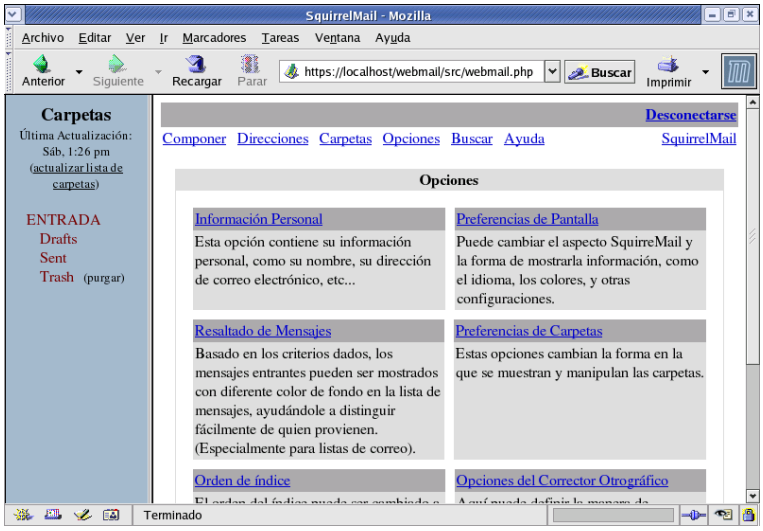


A partir de aquí, el cliente es muy intuitivo. Podemos enviar correo nuevo con la opción **Componer**:

56 `http://localhost/webmail`



Las opciones de personalización del cliente son variadas para ponerlo a nuestro gusto, y con la ventaja de que accedemos desde el lugar que queramos.



Bibliografía

[1] *Sendmail*, 2nd Edition, Bryan Costales & Eric Allman 2nd Edition January 1997 ISBN: 1-56592-222-0

[2] *TCP/IP Network Administration*, 2nd Edition, Craig Hunt 2nd Edition December 1997 ISBN: 1-56592-322-7

[3] *Guía de Administración de Redes con Linux* Olaf Kirch & Terry Dawson, Editado por O'Reilly (printed version) (c) 2000 O'Reilly & Associates Proyecto LuCAS por la traducción al español (c) 2002 HispaLiNUX

[4] `Bibliographyhttp://www.networkcomputing.com/unixworld/tutorial/008/008.txt.html`

[5] `http://www.catb.org/~esr/fetchmail/fetchmail-man.html`

[6] *How to set up SMTP AUTH* `http://www.jonfullmer.com/smtpauth/`

[7] *Proyecto de traducción de la documentación de Apache al español.*`http://quark.fe.up.pt/ApachES/`

[8] *The Apache Software Foundation*`http://www.apache.org/`

[9] *Servidor Apache.* Rich Bowen & Ken Coar. Prentice Hall

[10] *Servidor Apache 2.* Mohammed J. Kabir. Anaya Multimedia.

[11] Capítulos 9 y 10 del *Manual de referencia de Red Hat Linux* `http://europe.redhat.com/documentation/rhl9/rhl-rg-es-9/ch-httpd.php3`

[12] Capítulos 19 y 20 del *Manual de personalización de Red Hat Linux* `http://europe.redhat.com/documentation/rhl9/rhl-cg-es-9/`