Tema 7. Servicios de correo electrónico

Tema 7. Servicio de correo electrónico.

El servicio de correo electrónico (*email*) permite a los usuarios el envío de mensajes electrónicos a través de redes TCP/IP. Es uno de los servicios más usados y casi imprescindibles para los usuarios de Internet de forma que en algunas ocasiones ha llegado a sustituir al correo postal.

El servicio es fácil de usar, pero la administración y configuración del servicio puede ser complicada ya que intervienen varios componentes.

1. Características.

El servicio de correo electrónico permite el intercambio de mensajes entre un emisor (remitente) y uno o varios receptores (destinatarios).

Los destinatarios pueden leer los mensajes y responderlos cuando quieran sin que tengan que estar conectados o en línea. Es decir, realizan una comunicación asíncrona.

Los mensajes básicamente son de texto, los cuales pueden contener otros tipos de información a través de los archivos adjuntos: imágenes, presentaciones, hojas de cálculo, archivos de vídeo, audio, etc.

Los mensajes tienen como característica que la información que la compone está bien estructurada: nombre de emisor, destinatarios, fecha de envío, copia de mensajes, etc.

El servicio de correo electrónico tiene como gran ventaja que el intercambio de mensajes se realiza de forma rápida y económica. Como contrapartidas no se garantiza que los mensajes lleguen a su destino, ni que el remitente sea quien dice ser.

También existen otros tipos de problemas, como son el *spam* (mensajes no deseados o de remitentes desconocidos o falsos), la confidencialidad, suplantaciones de identidad, envío de software malicioso, etc.

Los primeros sistemas de correo aunque muy simples eran difíciles de utilizar. En la actualidad con los nuevos protocolos, el servicio es más completo y más fácil de usar por parte de los usuarios. Actualmente los protocolos de e-mail añaden nuevas características como son:

- Envío de un mismo mensaje a un grupo de usuarios mediante la gestión de listas de correo.
- Notificación al emisor si el receptor ha recibido o leído el mensaje.
- Se usan interfaces gráficas de usuario para facilitar el envío y recepción de mensajes.
- Se puede reenviar a otros usuarios los mensajes que se reciben.
- Con los buzones de correo se puede consultar el correo desde cualquier ubicación y equipo.

Existen varios estándares para la gestión de los servicios de gestión de correo, como son:

- X.400. Desarrollado por la antigua CCITT (ahora ITU) y posteriormente aprobado por la OSI. Es un sistema robusto y más completo que el usado en las redes TCP/IP, pero usa los estándares X.500 para las direcciones de correo que son más complicados. Actualmente casi en desuso.
- MOTIS de la OSI. Es una evolución del X.400 que sigue manteniendo la complejidad de éste.
- MHS (*Message Handing Service*). Protocolo usado en las redes Novell NetWare que cayó en desuso debido al avance de las redes TCP/IP impulsadas por Internet.
- SMTP. Protocolo basados en la arquitectura TCP/IP.

Estudiaremos los protocolos usados en la arquitectura TCP/IP, ya que los demás han dejado de usarse bien por su complejidad o por que se usan sólo en el ámbito de ciertas redes locales.

2. Componentes.

El sistema de gestión del correo electrónico se basa en los siguientes componentes:

- Mensajes de correo. El texto que se envía, al cual pueden acompañar documentos adjuntos.
- Direcciones de correo. Los usuarios disponen de direcciones de correo que están asociadas a una cuenta de correo. La dirección identifica los mensajes que envía un usuario, y la cuenta asociada permite a través de sus credenciales acceder a los mensajes enviados y recibidos.

- Servidores de correo. Los servidores de correo permiten el reenvío de mensajes de correo y están constituidos por una serie de componentes:
  - Buzón de correo (**mailbox**). Espacio donde se almacenan los mensajes de correo y que estará asociado a una cuenta de correo. Al buzón se accede mediante los credenciales de la cuenta.
  - Alias de correo. Cuenta que no tiene un buzón asociado, de forma que el correo dirigido a dicha cuenta se reenvía a un conjunto de cuentas que sí tienen un buzón asociado.
  - Lista de distribución. Cuenta de correo virtual que engloba a varios destinatarios.
  - Agentes de transferencia de correo (**MTA**, **Mail Transport Agent**). Cuando se envía un mensaje de correo, el mensaje se encamina desde el servidor de correo del remitente hasta llegar al servidor de correo electrónico del destinatario. Estos servidores de correo electrónico se denominan (**MTA**, **Mail Transport Agent** - Agente de Transporte de Correo). En Internet, los MTA se comunican entre sí usando el protocolo **SMTP**, y por lo tanto se los llama también **servidores SMTP** (o a veces servidores de correo saliente). Por lo tanto los MTA pueden actuar como cliente y como servidor del protocolo SMTP.
  - Agente de Entrega de Correo (**MDA**, **Mail Delivery Agent**). Cuando el MTA del destinatario entrega el correo electrónico, lo hace al servidor del correo entrante (llamado **MDA**, **Mail Delivery Agent** -Agente de Entrega de Correo), el cual recibe y deposita los mensajes en los buzones de los destinatarios. Además puede realizar funciones adicionales de distribución, filtrado y clasificación de mensajes.
  - Servidores POP/IMAP. Existen dos protocolos principales para que un cliente pueda recuperar los mensajes electrónicos depositados en los buzones de un MDA: los protocolos POP e IMAP. Mediante estos protocolos se permiten a los clientes de correo acceder a los buzones para obtener, consultar, borrar o modificar los mensajes almacenados.
  - Componentes adicionales. Permiten añadir nuevas funcionalidades integrándose con el MTA y/o el MDA. Por ejemplo: filtros *antispam*, filtros antivirus, listas de distribución, etc.
- Clientes de correo o agentes de usuario de correo, **MUA** (*Mail User Agent*). Permiten a los usuarios redactar, enviar y consultar los mensajes almacenados en los buzones. Actúan como clientes SMTP para enviar mensajes y como clientes POP/IMAP para acceder a los buzones. Cuando el **MUA** es un programa instalado en el sistema del usuario, se llama cliente de correo electrónico (Mozilla Thunderbird, Microsoft Outlook, Eudora Mail, Lotus Notes, etc.). Si se usa una aplicación web para interactuar con el servidor de correo, se llama webmail o correo web.
- Protocolos. La comunicación entre los componentes anteriores se basa en los protocolos siguientes:
  - Protocolos de transferencia de correo: SMTP y SMTP extendido (ESMTP). Realizan la comunicación entre los clientes (MUAs) y los MTAs; y también entre los MTAs.
  - Protocolos de entrega de correo: POP e IMAP. Realizan la comunicación entre los clientes (MUAs) y los servidores POP/IMAP.

3. Funcionamiento.

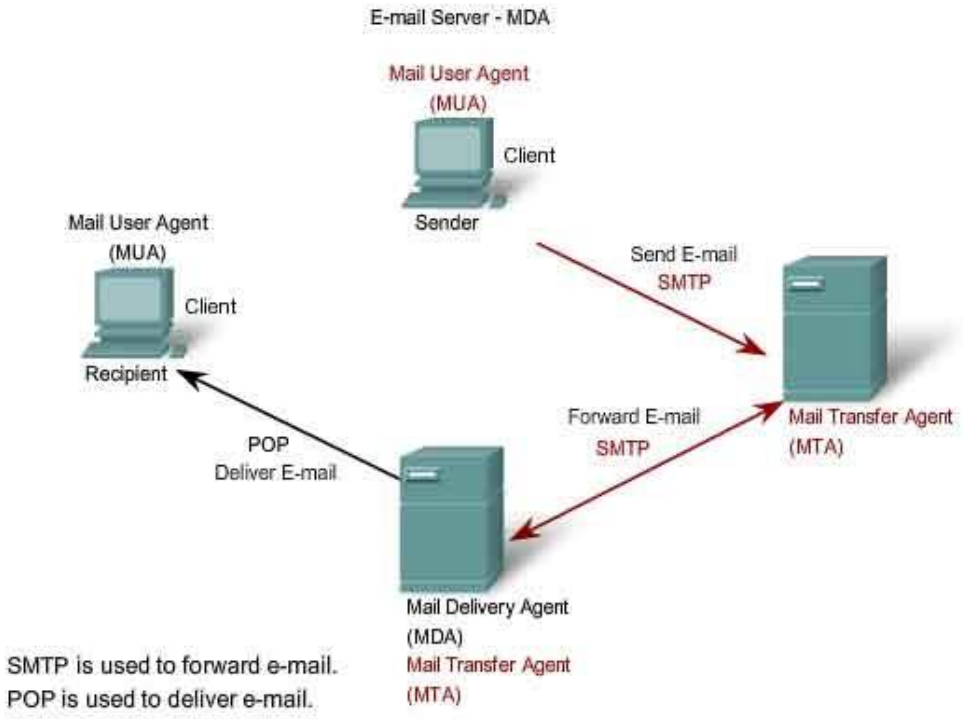
En la descripción anterior, se ha hecho referencia a tres tipos de agentes (MTA, MDA y MUA), los cuales realizan una serie de funciones. Dichas funciones se implementan mediante programas, los cuales algunos de ellos pueden realizar varias funciones.

Podemos hacer un resumen relacionando los elementos anteriores y su forma de operar. Partiremos de que los usuarios disponen de cuentas de correo, y que cada cuenta tiene asociado un buzón de correo en un servidor de correo.

- Un usuario para escribir su mensaje de correo utiliza un cliente de correo (MUA), el cual enviará el mensaje a su servidor de correo (MTA). Para ello usa el protocolo SMTP.
- Si el destinatario tiene cuenta en dicho servidor de correo, el mensaje se almacenará en el buzón del destinatario, tarea que realiza el MDA. Si no tiene cuenta, el MTA del servidor de correo se encarga de enviar el mensaje a otros MTA, hasta llegar al servidor donde el destinatario tiene cuenta y su posterior almacenamiento en su buzón. Para el envío entre MTAs se usa el protocolo SMTP.
- El destinatario usa su cliente MUA para acceder a los servidores POP o IMAP para obtener y consultar los mensajes de sus buzones. Para ello se usa el protocolo POP y/o IMAP.

3.1. Proceso detallado del servicio de correo.

Ya hemos visto en líneas generales el proceso para el envío y recepción de mensajes. Ahora lo detallaremos en función de las diferentes posibilidades que pueden presentarse.



1. **El usuario escribe un mensaje de correo usando un cliente de correo (MUA).**
2. El cliente de correo (MUA) envía el mensaje usando el protocolo SMTP a un servidor de correo. Dependiendo del servidor SMTP pueden darse diferentes posibilidades:
   a) Enviarlo a un servidor de correo local (MTA local). Situación que se produce si en la misma máquina donde está el cliente de correo hay un MTA. (Poco probable…)
   b) Enviarlo a un servidor de correo remoto (MTA). Este servidor remoto puede ser:
      b1) El servidor de correo del usuario destinatario. No es lo habitual ya que un servidor de correo debería estar configurado para no admitir mensajes de cualquier origen.
      b2) El servidor de correo del usuario que envía el mensaje. Es lo habitual ya que los MTAs deben estar configurados para recibir o reenviar mensajes sólo de usuarios, equipos, redes o dominios autorizados.
      b3) Un servidor de correo tipo *relay* que debe estar configurado para admitir mensajes del usuario, dominio o red desde donde se envía, para luego reenviarlos a otros servidores.
3. El servidor de correo recibe el mensaje usando el protocolo SMTP y
   a) Si el mensaje está dirigido a un usuario cuyo buzón está en el servidor, el correo se almacena en el buzón de usuario. Lo puede hacer el MDA. El MDA no tiene que estar en el mismo equipo del servidor de correo, puede ubicarse en otro equipo del mismo dominio.
   b) Si el correo está dirigido a un usuario que no tiene cuenta en el servidor, tenemos dos posibilidades a la hora del reenvío del mensaje:
   - Reenviar el mensaje directamente al servidor de correo del destinatario del mensaje, o
   - Reenviar el mensaje a otro servidor de tipo *relay* que vuelve a realizar el mismo proceso de reenvío. Lógicamente debe estar autorizado para admitir mensajes del usuario, equipo, dominio o red desde donde se envía.

Como vemos, es posible que un mensaje pase por diferentes MTAs hasta llegar al MTA destino. Los servidores de correo cuando reenvían un mensaje, le añaden información para que se pueda conocer por qué servidores ha transitado.

Cuando un servidor de correo debe reenviar un mensaje a otro MTA, debe conocer cuál es el servidor de correo del dominio al que envía el mensaje. Para ello debe consultar a un servidor DNS que le resuelva la dirección IP del servidor de correo de dicho dominio (registros MX).

4. Cuando un usuario quiere consultar su buzón de correo, puede

   a) Iniciar sesión en el servidor de correo entrante (MDA), (no es habitual que el cliente tenga cuenta de usuario en la misma máquina que la del servidor de correo) para acceder directamente a la carpeta del buzón, o usar un cliente de correo local para acceder al buzón.

   b) Usar un cliente de correo que se comunique con el servidor de correo entrante (MDA), para descargar el correo usando un protocolo de entrega como POP o IMAP.

Lo habitual en este proceso de envío y recepción de mensajes es seguir el camino indicado por los pasos 1, 2, b2, 3b y 4b, como indica la figura anterior.

3.2. Direcciones y cuentas de correo.

Una dirección de correo identifica los mensajes que envía un usuario y se compone de dos cadenas de caracteresseparadas por el carácter "@". Cada dirección de correo tiene asociada una cuenta de usuario con credeniales válidos en un servidor de correo entrante para acceder al buzón donde se depositan los mensajes.

Lo habitual es que la primera parte de la dirección de correo se corresponde con el nombre de la cuenta, aunque esto no es obligatorio. La segunda parte de la dirección indica el dominio al que pertenece. Recordemos que los servidores DNS registran cuales son los servidores de correo del dominio mediante registros tipo MX.

Una cuenta de correo ***redirigida*** es una cuenta que no tiene asociado un buzón, y por lo tanto no puede almacenar mensajes. Los mensajes enviados a este tipo de cuenta se redirigen a otra cuenta de otro dominio que sí tenga buzón. Un ***alias*** también es una cuenta sin buzón, pero que apunta a una cuenta normal del mismo dominio de correo.

Las cuentas redirigidas y los alias permiten crear varias identidades con varias direcciones de correo, pero asociadas a un mismo buzón. Suele ser útiles en procesos de registro o suscripción; como cuando deseamos darnos de baja en un servicio en Internet: damos de baja el alias sin tener que hacerlo con la cuenta principal.

4. Composición de los mensajes.

Los mensajes de correo son mensajes de texto compuestos de dos partes: **cabecera** y **cuerpo**.

El *cuerpo* contiene el mensaje en sí. Inicialmente el cuerpo sólo podía contener el texto en formato ASCII pero ahora, gracias a las extensiones *MIME*, se pueden enviar contenidos en otros formatos.

La *cabecera* está formada por una serie de líneas de texto con el formato *nombre:valor*. Permite almacenar información que se usa en la transferencia y clasificación del mensaje. Las líneas de cabecera son incluidas y consultadas por los componentes que intervienen en la transferencia del mensaje. Los tipos de cabecera más usados son:

- From: indica quien es el autor del mensaje. Se expresa mediante la dirección de correo asociada al buzón del remitente.
- To: son las direcciones de los destinatarios separados por comas.
- Cc: dirección de otros destinatarios que también reciben el mensaje (destinatarios secundarios).
- Bcc: es la dirección de otros destinatarios que recibirán el mensaje pero que no sabrán a quienes más se les envía el mensaje. En los clientes de correo aparece como un campo llamado CCO.
- Sender: indica quién envía el mensaje. No tiene que coincidir con el valor de *From*.
- Received: campo añadido por cada MTA por los que pasa el mensaje. Incluye una identificación del MTA, fecha y hora de la recepción, etc.
- Return-Path: generado por el MTA final indicando cual es el emisor real del mensaje.

- Subject: Resumen corto del mensaje que se presenta al destinatario.
- Date: Fecha y hora de envío del mensaje.
- Reply-To: dirección a la que responder. La añade el MUA emisor y puede ser diferente a la que aparece en el *From*. Útil cuando el remite tiene diferentes cuentas y quiere recibir las respuestas en una misma dirección de correo.
- Message-Id: Número único de identificación del mensaje.
- In-Reply-To: Indicador del mensaje al que se corresponde la respuesta.
- Delivered-To: dirección real a la que es entregado el mensaje (sin alias).

Aparte de estas cabeceras, el estándar RFC 822 permite que los MUAs y MTAs puedan definir sus propias cabeceras con el fin de añadir información adicional a los mensajes. Como convenio se utiliza el prefijo X- para indicar que la cabecera no es estándar.

4.1. MIME. (Extensiones Multipropósito de Correo Internet).

Las llamadas extensiones *MIME* están recogidas en los estándares RFC 2045/46 y permiten definir cabeceras pero ubicadas en el cuerpo del mensaje. Su función es la de permitir la inclusión de caracteres especiales, el uso de alfabetos no latinos y sobre todo la posibilidad de incluir archivos con contenidos no textuales (audio, video, imágenes, etc.); los llamados archivos *adjuntos*.

MIME describe el tipo de contenido del mensaje y el tipo de código usado empleando cabeceras en el cuerpo del mensaje. Esto tiene múltiples ventajas, como la capacidad de enviar múltiples adjuntos en un solo mensaje, el uso de texto enriquecido (diseños, fuentes, colores, etc.) y de caracteres no ASCII.

El hecho de que las extensiones *MIME* se añadan al cuerpo del mensaje es para que manteniendo la estructura del mensaje, se puedan añadir más informaciones. Así todos los MTAs que existen en Internet y que son complejos de reprogramar siguen siendo válidos (las cabeceras no cambian), aunque haya que modificar los MUAs para manejar estas extensiones que se incluyen el cuerpo.

MIME usa directivas especiales en los encabezados para describir el formato utilizado en el cuerpo de un mensaje, de modo que el cliente de correo electrónico pueda interpretarlo correctamente: Algunas son:

- MIME-Version: identifica la versión *MIME* usada.
- Content-Description: describe el contenido del cuerpo del mensaje.
- Content-Id: es un identificador único del contenido.
- Content-Transfer-Encoding: indica el sistema de codificación del texto del mensaje: ASCII, base64, etc.
- Content-Type: describe la naturaleza del mensaje. Puede ser *Text, Image, Audio, Video, Application, Message* o *Multipart.* A su vez, estos tipos se dividen en subtipos. Por ejemplo para el tipo *text* se tienen los subtipos *text/plain, text/rtf* y *text/richtext*.

El estándar MIME permite mensajes compuestos, es decir mensajes que incluyen múltiples adjuntos. Para hacerlo, MIME usa un estándar denominado boundary empleado como separador consistente en una cadena arbitraria. Es esencial que el valor de este separador no se encuentre dentro del contenido del mensaje. Por ejemplo:

```
Content-Type: multipart/mixed;
boundary="------------020005090303070203010601"
```

Con cada separador se delimita una porción de contenido que comienza con los encabezados Content-Type y Content-Transfer-Encoding. Existen varios tipos de separadores:

- multipart/mixed define una serie de elementos múltiples.
- multipart/alternative define alternativas para la misma información, como por ejemplo un mensaje en formato de texto o HTML. Si el cliente de correo electrónico puede mostrar mensajes con una disposición y está configurado para hacerlo, mostrará la versión HTML; de lo contrario, mostrará la versión de texto.
- multipart/parallel define datos presentes al mismo tiempo (como sonido e imagen).
- multipart/signed define una firma digital para los datos del mensaje.
- multipart/related define los datos relacionados

5. Servidores de correo.

Los MTA se encargan de la recepción y reenvío de mensajes entre equipos usando el protocolo SMTP. Un MTA actúa como cliente SMTP cuando envía mensajes a otros MTA, y como servidor cuando recibe mensajes de otros MTAs o de MUAs. Por defecto escuchan por el puerto 25/TCP.

Las dos funciones principales de un MTA son:

- Almacenar los mensajes dirigidos a los destinatarios que tienen cuenta en el dominio gestionado por el servidor. El MTA puede almacenar el mensaje directamente en el buzón o bien usa un MDA para dicha tarea. Existen diferentes formatos de almacenamiento de los mensajes en los buzones, aunque los más usados son, como veremos, *mbox* y *maildir*.
- Reenviar los mensajes (*relay*) dirigidos a los destinatarios que NO tienen cuenta en el dominio gestionado por el servidor.

Una de las funciones que deben estar más controladas en los MTAs es la retransmisión de mensajes. Si no es así, un MTA podría usarse por usuarios desconocidos (no controlados) para enviar correos maliciosos, con publicidad, etc., a otros usuarios. Para controlar el reenvío de los MTAs, hay que configurarlos de forma que haya que autenticarse ante ellos previamente, y activar filtros que impidan el reenvío de mensajes provenientes de ciertas direcciones IP o dominios sospechosos. Los MTAs que realizan estas funciones de reenvío de forma controlada se denominan *smart host*.

Los llamados servidores *open relay* son MTAs que permiten el reenvío de mensajes desde cualquier lugar sin controlar de dónde proceden o quién los envía. Se emplean frecuentemente para generar *spam*. No obstante existen listas en Internet (listas negras), que incluyen direcciones IP o dominios que contienen MTAs que actúan como *open relay* y que son consultadas por otros MTAs para rechazar o no los mensajes provenientes de dichos MTAs.

Respecto a desarrollos software, existen muchos MTAs tanto para sistemas libres como propietarios: *sendmail, Postfix, Microsoft Exchange, Qmail, Exim, IMail, hMailSever, etc.*

La aplicación *Microsoft Exchange* para Windows es realmente un software colaborativo, que incluye servicios de correo electrónico, y que se integra mediante *Active Directory* en los dominios de Windows. No es un componente de Windows Server, sino una aplicación que requiere licencias para su utilización.

No obstante, Windows Server dispone de componentes para instalar un MTA, pero no de componentes para POP/IMAP, por lo que la gestión de correo electrónico es realmente muy básica.

En Internet, la mayor parte de los MTAs instalados corren en sistemas GNU/Linux y en concreto los más usados son *sendmail*, *Postfix*, *Qmail* y *Exim*. El MTA *sendmail* es el más tradicional, pero es difícil de configurar (el archivo de configuración *sendmail.cf* es bastante extenso y complicado) y tiene algunos problemas de seguridad. Normalmente se configura con otras herramientas, como *webmin*.

Postfix es un sistema diseñado por IBM bastante más fácil de configurar y más optimizado que *sendmail*. No está tan extendido como *sendmail*, pero parecer ser su digno sustituto. *QMail* también es más fácil de configurar y seguro que *sendmail* pero su implantación es menor. *Exim* es el MTA que actualmente se instala en las distribuciones de *Debian*.

6. Agentes de entrega de correo.

Los agentes de entrega de correo (MDA) son programas invocados por los MTAs para realizar la tarea de depositar los mensajes en los buzones de los usuarios. Además realizan funciones de clasificación, filtrado y distribución de los mensajes. Muchos vienen integrados en el mismo MTA o ser un módulo o software adicional.

Algunos de MDA son *procmail* (muy usado junto al MTA *sendmail*), y *maildrop* que funciona con el MTA *Courier Mail Server*.

7. Servidores POP/IMAP.

Los servidores POP/IMAP permiten a los clientes acceder a sus buzones desde equipos remotos. Los servidores POP escuchan por el puerto 110/TCP, y los servidores IMAP por el 143/TCP.

Pueden integrarse con el MTA, pero normalmente es un software o módulo adicional. Algunos de ellos son: *Dovecot, Microsoft Exchange Server, Cyrus-imapd, Curier-imap* e *IMail*.

8. Agentes de usuario de correo (clientes de correo).

Los MUAs o clientes de correo permiten a los usuarios redactar los mensajes, adjuntarle archivos y enviarlos a MTAs. Otras funciones que la mayoría implementa son la de obtener el correo de los buzones de los servidores y almacenarlos en el equipo local del usuario (buzones locales), leer el correo, clasificar el correo en carpetas, establecer filtros, gestionar los contactos, cifrado de los mensajes, integración con antivirus, etc.

Actúan como clientes SMTP cuando envían mensajes a los MTAs, y como clientes POP/IMAP cuando acceden a los buzones de los servidores POP/IMAP.

Los clientes podemos clasificarlos en modo texto, gráficos y clientes web (los llamados *webmails*).

Los clientes en modo texto se ejecutan mediante línea de comandos y su interfaz está en modo texto. En los sistemas *Linux* es habitual su uso cuando los usuarios pueden acceder directamente al equipo que contiene los buzones sin necesidad de un servidor POP/IMAP. Ejemplos: *mailx*, *mutt*, *pine*, etc.

Los clientes gráficos muestran una interfaz gráfica al usuario lo que permite un uso más fácil e intuitivo. Podemos encontrar muchos de ellos como *Mozilla Thunderbird*, *Evolution*, *Eudora*, *Microsoft Outlook*, *Windows Live Mail*, *KMail*, *OpenOffice*, etc.

Los *webmails* son aplicaciones web que ejecutan en un servidor web un cliente de correo tipo IMAP. Proporcionan mucha flexibilidad ya que podemos gestionar el correo desde cualquier ordenador conectado a Internet, en lugar de tener que usar la maquina donde tenemos instalado el cliente tradicional. Algunos de ellos son *SquirrelMail*, *Roundcube*, *Horde*, *OWA (Outlook Web Access)*, *openwebmail*, etc.

La mayoría de los clientes de correo, independientemente del sistema operativo utilizado se configuran de forma bastante similar. Los parámetros necesarios básicamente son:

- Nombre de la cuenta. Sirve para identificarse en el cliente de correo.
- Dirección de correo electrónico. Se debe especificar con el nombre completo de la forma *usuario@dominio*.
- Tipo de cuenta. Indica el protocolo para leer los mensajes almacenados en los buzones del servidor. Puede ser POP3, IMAP, Exchange, etc.
- Servidor de correo entrante. Nombre del servidor que almacena mediante buzones los mensajes que podrán ser leídos mediante POP3 o IMAP.
- Servidor de correo saliente (SMTP). Nombre del servidor de correo que actúa como MTA enviando los mensajes mediante el protocolo SMTP.

9. Protocolos de correo.

La comunicación entre los componentes del servicio de correo se basa en los protocolos SMTP, POP e IMAP. A continuación comentaremos sus principales características.

9.1. Protocolos de transferencia de correo SMTP y ESMTP.

SMTP (*Simple Mail Transfer Protocol*) define las reglas que utilizan los MTAs para intercambiar mensajes, y las reglas de los MUAs para enviar los mensajes a los MTAs.

Sus especificaciones están publicadas en RFC 821, usando TCP como protocolo de transporte. Su funcionamiento se basa en el intercambio de mensajes del tipo *comando* y *respuesta* en formato texto entre el cliente y el servidor sin mecanismos de autentificación entre ellos.

La comunicación se realiza de la siguiente manera:

- Se establece la conexión TCP con el servidor (el cliente inicia la conexión y el servidor contesta).
- El cliente saluda con el comando HELO al servidor, el cual responde al saludo.
- Transmisión de mensajes (se envían comandos desde el cliente y se reciben respuestas).
- Despedida y cierre de la conexión.

Los comandos SMTP son palabras que usan los clientes para realizar peticiones y enviar información a los servidores. Algunos de ellos admiten parámetros. Cada comando termina con las secuencia de caracteres de control <CR> (retorno de carro) y <LF> (salto de línea).

Los servidores SMTP responden a los clientes informando de su estado con números de 3 cifras (código de estado) seguidos de un mensaje de descripción. El primer dígito indica el tipo de respuesta y los otros dos concretan dicha respuesta.

Por ejemplo, supongamos que tenemos el servicio SMTP de W2012 Server instalado con su configuración por defecto. Vemos como inicialmente el dominio de correo que gestiona coincide con el nombre del equipo; mantendremos dicho nombre. Para enviar un mensaje de correo a maroto@winp, nos conectamos mediante *telnet* al servidor de correo del dominio por el puerto 25 para empezar la conexión.

```
C:\Documents and Settings\Administrador>telnet localhost 25
220 winp Microsoft ESMTP MAIL Service, Version: 8.5.9600.16384 ready at  Wed, 19
Dec 2018 19:41:36 +0100
helo localhost
250 winp Hello [127.0.0.1]
mail from:pepe@winp
250 2.1.0 pepe@winp....Sender OK
rcpt to:maroto@winp
250 2.1.5 maroto@winp
data
354 Start mail input; end with <CRLF>.<CRLF>
subject:  Novedades
Hola, como hoy estreno nueva peluca, he pensado que si tal, que si cual…
.
250 2.6.0 <WINPFJh8QRr4MfOvSaK00000001@winp> Queued mail for delivery
quit
221 2.0.0 winp Service closing transmission channel
Se ha perdido la conexión con el host.
C:\Documents and Settings\Administrador>
```

Podemos ver el mensaje guardado en la carpeta C:\inetpub\mailroot\Drop, carpeta donde se almacenan temporalmente los mensajes que deberán ser entregados. Como tanto la cuenta del remitente como la del destinario pertenecen al dominio local (winp), el mensaje no se intentará entregar a otro servidor.

Como hemos podido comprobar SMTP no usa autentificación. Esta fue una de las razones del origen del protocolo **ESMTP (Extended SMTP)**. ESMTP define un conjunto de extensiones a SMTP normalizadas en el RFC 1651 que solucionan problemas e incorporan nuevas funcionalidades. La contribución más importante es la posibilidad de utilizar diferentes tipos de autentificación (a través del comando AUTH) entre clientes y servidores.

Los clientes que quieran una conexión mediante ESMTP deben usar el comando de saludo EHLO *(Extended helo)* en lugar del habitual HELO. En este caso el servidor responderá con una lista de comandos ESMTP indicando de esta forma que extensiones SMTP soporta. Si el servidor no soporta el protocolo ESMTP, rechazará el saludo (código de respuesta 500), y el cliente deberá usar el protocolo SMTP.

Como ejemplo podemos configurar el servidor de Windows 2012 para que acepte autentificación. Si desde las propiedades de *[SMTP Virtual Server #1]*, activamos la ficha *Acceso* y pulsamos el botón *Autentificación*, podremos activar la autentificación *básica* y la *Integrada de Windows*.

Si reiniciamos el servidor y realizamos una conexión, pero saludando con el comando EHLO, podremos ver como el servidor contesta con las extensiones SMTP que tiene disponibles. Entre ellas vemos AUTH NTLM LOGIN, las que permite las autentificaciones *Windows integrada* y *básica* (cifrado en *Base64*).

```
C:\Documents and Settings\Administrador>telnet localhost 25
220 winp Microsoft ESMTP MAIL Service, Version: 7.5.7601.17514 ready at  Sat,
19 Dec 2015 13:15:27 +0100
ehlo localhost
250-winp.aula.izv Hello [127.0.0.1]
250-AUTH NTLM LOGIN
250-AUTH=LOGIN
250-TURN
250-SIZE 2097152
250-ETRN
250-PIPELINING
250-DSN
```

```
250-ENHANCEDSTATUSCODES
250-8bitmime
250-BINARYMIME
250-CHUNKING
250-VRFY
250 OK
```

9.2. Protocolos de entrega de correo.

Los protocolos POP e IMAP son los protocolos utilizados para acceder a los buzones de correo almacenados en servidores remotos.

9.2.1. Protocolo POP.

POP (*Post Office Protocol*) es un protocolo de transporte TCP basado en el intercambio de mensajes (comandos y respuestas) en formato texto entre cliente y servidor que escucha por el puerto 110. La versión actual para POP es la 3 (POP3) y sus especificaciones se recogen en el RFC 1225.

POP es un protocolo simple que permite acceder a los buzones de correo remoto. Básicamente su funcionamiento consiste en conectarse al servidor, descargar los mensajes de correo (eliminándolos del servidor o dejando una copia de ellos), y cerrar la conexión.

Antiguamente POP3 era muy usado ya que permite conectarse al servidor sólo durante el tiempo necesario para la descarga de los mensajes, y luego leerlos tranquilamente en la máquina local. Actualmente con las conexiones de tipo "tarifa plana" el protocolo POP tiene menos interés.

POP3 necesita autentificación como protección contra los accesos ilegales al buzón de correo de los usuarios. Inicialmente utilizaba un mecanismo sin cifrado donde la transmisión de credenciales se realiza en texto plano. Aunque esta posibilidad todavía está disponible, en la actualidad se cuenta con diversos métodos de autenticación que ofrecen cifrado.

9.2.2. Protocolo IMAP.

IMAP (*Internet Message Access Protocol*) es un protocolo de transporte TCP basado en el intercambio de mensajes (comandos y respuestas) en formato texto entre cliente y servidor que escucha por el puerto 143. La versión que se usa actualmente es la 4 (IMAP4) y sus especificaciones están expuestas en el RFC 1064.

IMAP es un protocolo más avanzado que POP ya que permite:

- Operar en modo desconectado (*offline*) estableciéndose la conexión para descargar correos. En este modo opera de forma análoga a como lo hace el protocolo POP.
- Operar en modo conectado (*online*), donde el cliente se mantiene conectado al servidor permitiéndose la manipulación de los buzones como si fueran locales. En este modo se pueden realizar consultas, búsquedas, borrar mensajes, moverlos, etc.
- Permite acceder a parte de los mensajes; por ejemplo leer un mensaje, pero no tener que descargar los archivos adjuntos que tuviera.
- Gestionar buzones: crear buzones, borrar buzones, así como acceso simultáneo al mismo buzón por parte de varios usuarios.
- Usa mecanismos de autentificación de forma que el usuario tiene que autenticarse para acceder a su buzón. Además se permite que los credenciales de autentificación se transmitan cifrados.

10. Seguridad y privacidad.

Los protocolos en que se basa el servicio de correo son antiguos, y cuando se definieron sus especificaciones, al igual que sucedió con los protocolos FTP o HTTP, no se pensó en la seguridad.

Los problemas principales de seguridad y privacidad que nos encontramos son:

- Originalmente SMTP no contemplaba mecanismos de autentificación, por lo que en principio cualquiera podía utilizar los MTAs para enviar mensajes. Este agujero de seguridad es uno de los usados para generar *spam*.
- El intercambio de información se realiza en texto plano sin cifrar, lo que permite mediante un análisis de tráfico de la red (*sniffing*) obtener los mensajes y manipularlos.

- No se garantiza que los clientes y los servidores son quienes dicen ser. Esto permite que se pueda suplantar la identidad de los mismos, enviando mensajes con remitentes falsos (*spam*) o suplantar a un MTA de forma que recibe los mensajes de los usuarios con cuentas en él (*spoofing*).
- Los mensajes pueden contener código malicioso (virus, troyanos, etc.) en sus adjuntos.
- Por defecto no se cifran ni se firman los mensajes, por lo que en sus buzones de almacenamiento o en sus viajes por diferentes redes, lo mensajes pueden ser leídos, modificados o borrados por terceras personas no autorizadas.

Las soluciones a estos problemas de seguridad vienen dadas por usar herramientas añadidas a los programas cliente o paquetes auxiliares que trabajan conjuntamente con los protocolos de correo.

10.1. SASL. (*Simple Authentication and Security Layer*)

SASL. (*Simple Authentication and Security Layer*) es un mecanismo de autentificación para aquellos protocolos que están orientados a conexión como son SMTP, POP, IMAP, LDAP o XMPP.

SASL permite distintos tipo de autentificación (*anonymous*, *cram-md5, plain, ntlm …*). Si nuestro servidor de correo va a enviar correo a dominios externos al nuestro, tendremos que configurar la autenticación SASL en el servidor.

10.2. SPA (*Secure Password Authentication*)

SPA (*Secure Password Authentication*). Es un método de autentificación implementado sólo por los servidores y clientes de correo de Microsoft para los protocolos SMTP, POP e IMAP. Este método se desarrolló basándose en Integrated Windows Authentication (NTLM), el mecanismo de autentificación de Windows NT Lan Manager.

10.3. Autentificación SMTP.

Para aumentar la seguridad y controlar quienes pueden usar un servidor de correo para enviar mensajes se han implementado mecanismos de autentificación para SMTP. Básicamente actualmente existen dos:

- **SMTP Authentication** (Autentificación SMTP). Es una parte ESMTP (Extended Simple Mail Transfer Protoco SMTP) que permite autenticar a los usuarios que usan clientes MUAs antes de enviar un correo, y también autentificar a los MTAs que envían mensajes a otros MTAs.

  Se implementa mediante el comando `AUTH` permitiéndose distintos tipo de autentificación (*anonymous, cram-md5, login, plain, ntlm …*).

  Si se usa autentificación *plain* o *login* los nombres de usuario y contraseña se cifran usando el débil sistema de codificación *base64*, por eso se suelen integrar con protocolos SSL/TLS para aumentar la seguridad mediante el cifrado de los credenciales.
- Autentificando al cliente y a los servidores SMTP mediante el uso de certificados digitales. Este mecanismo puede usarse cuando se usa el protocolo SMTP Seguro (SMTPS).

10.4. Protocolos seguros.

Al igual que ocurre con otros protocolos, como FTP o HTTP, una solución para aumentar la seguridad es usar protocolos seguros. De esta forma SMTP, POP e IMAP se pueden encapsular en SSL/TLS para garantizar la confidencialidad y la integridad de la información transmitida mediante cifrado, así como la autenticidad usando certificados. En el servicio de correo se puede usar los siguientes protocolos seguros:

- SMTPS. Encapsula SMTP en SSL/TLS. Los servidores usan el puerto 465/TCP.
- POPS. Encapsula POP3 en SSL/TLS. Los servidores usan el puerto 995/TCP.
- IMAPS. Encapsula IMAP4 en SSL/TLS. Los servidores usan el puerto 993/TCP.
- STARTTLS. Es un mecanismo que permite negociar conexiones SSL/TLS sobre protocolos de texto plano. En particular, existen extensiones STARTTLS para SMTP, POP e IMAP con la ventaja que no se necesita otro puerto para establecer la conexión cifrada.

  Para SMTP el sistema más utilizado es STARTTLS, implementándose como una de las extensiones de ESMTP (Extended SMTP) mediante el comando `STARTTLS`. De esta forma se negocia una conexión SSL/TLS con los servidores, pero usando el puerto estándar 25/TCP.

OJO. *Actualmente, muchos sistemas ofrecen a los clientes (MUA) el envío de mensajes sólo por el puerto 587 obligando a utilizar STARTTLS y autenticación. Es decir bloquean el puerto 25 para las conexiones desde los PCs de usuario ya que muchas de éstas se convierten en fuente de spam por haber sido infectadas. El puerto 25 se mantiene sólo para envío de correo entre MTAs.*

Resumiendo, una práctica muy habitual es configurar los MTAs con SMTP AUTH para garantizar la autentificación, y con STARTTLS para garantizar la confidencialidad y la integridad de la información mediante cifrado usando los puertos 25/TCP o 587/TCP.

Podemos configurar el servidor SMTP de Windows 2012 para usar cifrado SSL/TLS mediante el mecanismo STARTTLS. Para ello desde las propiedades de *[SMTP Virtual Server #1]*, activamos la ficha *Acceso* y marcamos la casilla *Requerir cifrado TLS* siempre que exista un certificado para el servidor.

Si reiniciamos el servidor y realizamos una conexión, pero saludando con el comando EHLO, podremos ver como el servidor contesta presentando la extensión STARTTLS.

```
C:\Documents and Settings\Administrador>telnet localhost 25
220 winp Microsoft ESMTP MAIL Service, Version: 7.5.7601.17514 ready at  Sat,
19 Dec 2015 13:35:28 +0100
ehlo localhost
250-winp.aula.izv Hello [127.0.0.1]
250-AUTH NTLM LOGIN
250-AUTH=LOGIN
250-TURN
250-SIZE 2097152
250-ETRN
250-PIPELINING
250-DSN
250-ENHANCEDSTATUSCODES
250-8bitmime
250-BINARYMIME
250-CHUNKING
250-VRFY
250-TLS
250-STARTTLS
250 OK
```

10.5. Firma y cifrado de mensajes.

Aunque usemos protocolos seguros, si un mensaje viaja a través de servidores que no usan protocolos seguros no se garantiza que los mensajes no sean leídos o manipulados por terceros en su viaje. Además, un administrador puede acceder a los buzones de los usuarios y leer o modificar los mensajes.

Por otro lado para asegurarnos que el remitente del mensaje es quien dice ser, el mensaje debería ir autentificado mediante una firma digital.

La solución a estos problemas es firmar y cifrar los contenidos de los mensajes usando algoritmos criptográficos. De esta forma, aunque no se use un protocolo seguro, por lo menos el texto del mensaje se envía cifrado. Los principales sistemas de firma y cifrado de mensajes son:

- S/MINE (Secure MIME). Es un estándar para criptografía de clave pública para cifrar mensajes para el firmado de correo electrónico encapsulado en MIME. S/MIME se basa en la encriptación asimétrica usando para ello a un par de claves (clave privada/clave pública). Se requiere de una autoridad certificadora que asegure la identidad del firmante.
- Certificados X.509. Son los certificados definidos según el estándar UIT-T. De este tipo son los certificados que usa el protocolo SSL/TLS, base de los protocolos HTTPS, FTPS y otros. El uso de estos certificados se basa en el uso de un par de claves pública/privada. Con la clave privada se puede firmar digitalmente documentos, en particular mensajes de correo.
- PGP (*Pretty Good Privacy*). Es un software criptográfico que permite firmar, comprimir y enviar el mensaje cifrado junto con la clave de cifrado también cifrada.

  La clave se cifra con una clave pública suministrada por el receptor, mientras que el descifrado de la clave se realiza con la clave privada del receptor. No se usa autoridades de certificación (AC) centrales. El sistema se basa en la confianza entre usuarios, de forma que los usuarios confían en los certificados de los otros usuarios que previamente han intercambiado.

Basado en PGP se ha desarrollado el estándar *OpenPGP*. Una implementación de este estándar es *GnuPG*, la cual usa una interfaz en modo texto. Muchos clientes de correo integran su uso mediante *plugins*.

En la URL https://www.ionos.es/digitalguide/correo-electronico/seguridad-correo-electronico/correo-seguro-protege-tu-informacion-con-el-cifrado-ssl/ podemos encontrar un manual para cifrar y firmar mensajes de correo mediante PGP para los webmail (Gmail, Yahoo, etc.) más populares, usando la extensión Mailvelope (https://www.mailvelope.com/en/) para los navegadores Firefox y Chrome.

10.6. Filtrado de correo

Los servidores de correo permiten definir reglas para filtrar y restringir el envío de mensajes, o marcar mensajes que no cumplan ciertas características. Estas funcionalidades suelen integrarse en los MTAs y/o pueden ampliarse con componentes o módulos adicionales.

10.6.1. Protección frente a virus.

El correo electrónico genera un enorme tráfico de información entre distintas zonas geográficas, facilitando la propagación de virus, sobre todo *gusanos* y *troyanos*. Por ello es imprescindible instalar herramientas antivirus tanto en los servidores de correo como en los equipos cliente. La mayoría de los fabricantes de antivirus disponen de productos específicos para integrarlos en los servidores de correo.

10.6.2. Filtros y métodos *antispam*.

El *spam* lo constituyen los mensajes no solicitados por los usuarios que son enviados de forma masiva con contenidos publicitarios. Muchos se identifican por usar direcciones de remitentes falsas o inexistentes que detectan los DNS mediante consultas inversas. Otras veces es difícil de detectar, porque se usan direcciones de correo que han sido usurpadas a sus legítimos propietarios.

A parte de la molestia para los usuarios que reciben *spam*, éste puede hacer desbordar los buzones de los usuarios y dificulta la consulta de los correos normales o legítimos. Actualmente el *spam* supone el 80% de los mensajes enviados, provocando costes a las empresas ya que deben invertir recursos para filtrarlos y eliminarlos.

Los MTAs suelen combinar sus propios métodos con filtros *antispam*. Los principales métodos *antispam* son:

- Métodos preventivos.
  - Usar el campo CCO. De esta forma se ocultan las direcciones de correo de los contactos. Atención a los mensajes que piden un *forward*. Es conveniente usar *alias* cuando enviamos mensajes a direcciones que no son de total confianza.
  - Enmascaramiento de las direcciones de correo (con logos, *capchas*…) para evitar que sean capturadas por *spambots* (programas que rastrean la Web en busca de direcciones de correo).
- Métodos correctivos. (Implantados en MUAs, MTAs y filtros *antispam*).
  - Filtros basados en el contenido de los mensajes (en las cabeceras y en los contenidos de los mensajes). Filtros basados en métodos OCR para detectar imágenes. Filtros heurísticos basados en palabras clave y patrones de comportamiento. Filtros estadísticos.
  - Métodos no basados en los contenidos de los mensajes. Uso de listas negras de direcciones IP o dominios que se consultan en tiempo real para marcar o no el mensaje como *spam*. Esquemas de autorización de envíos para verificar si el correo es enviado por un MTA autorizado.

11. Servidor de correo en Windows Server 2012.

A partir de Windows Server 2012 y usando Microsoft IIS, ya no se puede configurar un servicio de correo completo. Aunque se mantiene la posibilidad de proporcionar servicios de envío de correo electrónico (SMTP), ya no se puede configurar su recuperación mediante POP o IMAP. Si se necesita un auténtico servicio de correo en Windows 2008 o posterior, deberemos usar Microsoft Exchange o usar una aplicación de terceros como puede ser *hMailServer*.

No obstante, la posibilidad de sólo enviar correo es útil cuando por ejemplo desde una aplicación web se generan mensajes de correo, y se usa el mecanismo de reenvío para entregarlos a sus destinatarios. Por supuesto que esto sólo sería válido si nuestro servidor SMTP tuviera una IP pública y un nombre de dominio DNS público que otros servidores SMTP pudieran reconocer.

11.1. Microsoft IIS SMTP.

Suponemos que los servicios de IIS ya están instalados en el equipo que hará de servidor del servicio SMTP; de no ser así, se realizaría la instalación tal como se indica en el tema 4 (Servicios web).

Para instalar el servicio SMTP partimos de *Herramientas administrativas -> Administrador del servidor* y desde el *Panel* activamos el enlace de *Agregar roles y características.* Pulsamos repetidamente *Siguiente* hasta que el asistente presenta en la ventana lateral izquierda *Características.* Seleccionamos la característica *Servidor SMTP*. Confirmamos pulsado *Agregar característica*. Pulsamos seguidamente el botón *Siguiente* y finalizamos con el botón *Instalar*.

En Windows Server 2012, la versión que se dispone de IIS es la 8. Esta versión no contempla la posibilidad de administrar el componente SMTP de IIS. No obstante se puede gestionar mediante la consola de IIS 6.0 la cual se ha instalado de forma automática al instalar la característica *Servidor SMTP*.

Es importante antes de comenzar con el proceso de instalación recordar que nuestros mensajes no podrán salir del dominio local, ya que no disponemos de un dominio DNS público en el cual nuestro servidor SMTP sea identificado. Para las pruebas, supondremos que nuestro equipo Windows Server 2012 se denomina winp y será el servidor de correo del dominio local aula.izv. Para ello añadiremos al servidor DNS del dominio un registro de tipo *MX* con el nombre del servidor de correo winp.aula.izv.

Para configurar el servidor SMTP, deberemos ejecutar desde las *Herramientas administrativas* el *Administrador de Internet Information Services (IIS) 6.0*. Si nos situamos sobre el icono que representa al servidor *[SMTP Virtual Server #1]*, y lo expandimos, veremos los dominios que gestionará nuestro servidor.

El servicio SMTP de Windows permite gestionar varios servidores de correo virtuales. Cada uno de ellos se asocia a un dominio distinto y gestiona todos los usuarios asociados a dichos dominios.

Por defecto el servidor SMTP predeterminado gestionará un dominio de nombre igual al del equipo. Esto implicaría que las cuentas de correo serían del tipo usuario@winp. Vamos a cambiar el nombre del dominio por *aula.izv* de forma que las cuentas serán ahora del tipo usuario@aula.izv. Para ello nos situamos en el icono que representa al dominio, y seleccionamos la opción "*cambiar nombre*" de su menú contextual.

Para iniciar los servicios SMTP hay que pulsar con el botón derecho del ratón sobre el icono *[SMTP Virtual Server #1]* y pulsar la opción "*Iniciar*". Desde el mismo menú contextual podemos suspender o detener el servicio.

La configuración del servicio SMTP se realiza seleccionando su icono y pulsando la opción "*Propiedades*" de su menú contextual, apareciendo diferentes opciones agrupadas en fichas. Las principales opciones se agrupan en las siguientes fichas:

- General. Permite establecer: la dirección IP y puerto del servidor por la que atenderá el servicio, el tiempo máximo que una conexión puede estar inactiva, el número máximo de conexiones simultáneas y habilitar un registro de *logs* del servicio junto a su formato.
- Acceso.
  - Mediante el botón "*Autentificación*" se configura el acceso al servidor. Por defecto está activado el acceso *anónimo* por lo que no se pide autentificación cuando un cliente u otro servidor SMTP se conecta al servidor para enviar correo. Evidentemente esta situación no es adecuada ya que otros SMTP podrían usar nuestro servidor para reenviar mensajes. En la autentificación *básica* (AUTH LOGIN), el nombre de usuario y la contraseña se envían cifrados usando el algoritmo *base64*. En la autentificación *Windows integrada* (AUTH NTLM), se usan los mecanismos de seguridad de acceso que integra Windows Server.
  - El apartado "*Comunicación segura*" se usa un certificado de servidor y una clave pública para que los clientes puedan utilizar TLS para enviar los mensajes cifrados al servidor. (STARTTLS).
  - El botón *Conexión* permite establecer las IP o el nombre de dominio de los equipos que podrán usar este servidor para enviar correo.

  - Retransmisión. De forma predeterminada, cualquiera que tenga acceso al servidor mediante sus credenciales de autenticación podrá usarlo para hacer *relay*, es decir, para retransmitir mensajes usando el servidor SMTP como un *smart host*. Lo normal es limitar esta posibilidad indicando las direcciones IP o nombres de dominio de los equipos que tendrán o no permiso para retransmitir correo a través del servidor. De forma predeterminada la lista está vacía, lo cual es lógico por que podríamos propagar *spam* si nuestro servidor estuviera en Internet.

- Mensajes. Permite establecer los límites y requisitos de la transmisión (tamaño de los mensajes, número máximo de mensajes por conexión, número máximo de destinatarios por mensaje, etc.)
- Entrega. Aquí se establecen las opciones de entrega y enrutamiento. Entre ellas se incluyen la configuración para el correo retrasado, como son los intervalos de reintento en la entrega de mensajes y el límite del número de saltos a servidores durante la entrega. Otras opciones son:
  - *Seguridad de salida…* Establece el método de autentificación y cifrado que usará nuestro servidor frente a otro servidor que actúe como un *smart host* para reenviar mensajes.
  - *Conexiones salientes…* Establece límites al número máximo de conexiones, y el nº del puerto del servidor de correo tipo *smart host* al que se conectará nuestro servidor.
  - *Opciones avanzadas…* Podemos establecer un dominio de enmascaramiento que se muestre en la cabecera *From* del mensaje, en lugar del nombre del dominio original del remitente. También hay que indicar el nombre de dominio de nuestro servidor SMTP (*), y en su caso, el nombre de dominio del equipo que hará de *smart host*.
- Enrutamiento LDAP permite especificar a un servidor LDAP que almacene información acerca de los clientes de correo y sus buzones. El servidor SMTP utiliza el protocolo LDAP para comunicarse con los servicios de directorio.
- Seguridad permite agregar cuentas de usuario y grupos de Microsoft Windows a la lista de operadores de servidores SMTP.

(*) Si pretendemos que nuestro servidor SMTP envíe mensajes a cuentas de correo pertenecientes a dominios públicos, tendríamos que usar un nombre de dominio público con el cual se presentaría a los otros servidores de correo. Este nombre deberá ser válido y además su registro inverso (PTR) debe coincidir con la IP pública del servidor. Si no es así los servidores receptores no aceptarían los mensajes.

11.2. hMailServer.

hMailServer (http://www.hmailserver.com) es un servidor de correo gratuito para plataformas Windows. Soporta SMTP, POP e IMAP y es fácil de integrar con otros componentes como *webmails*, filtros *antispam*, antivirus, etc. Para el almacenamiento de las cuentas de correo puede usar diferentes gestores de bases de datos como MySQL, PostgreSQL o Microsoft SQL Server.

Durante el proceso de instalación seleccionaremos el software del *servidor* y las herramientas de *administración* del servidor. Elegiremos como base de datos para las cuentas, la base de datos interna *Microsoft SQL Compact*. Posteriormente nos pedirá una contraseña para poder administrar el servidor. Terminada la instalación ya podremos administrar el servidor local, aunque podemos hacerlo siempre desde *C:\Program Files(x86)\hMailServer\Bin\hMailAdmin.exe*. Para iniciar o parar el servicio se ejecutan los *scripts* correspondientes ubicados en *Menú Inicio ->Programas -> hMailServer ->Service*.

Podemos comprobar cómo el servidor está escuchando por los puertos 25 y 587 (SMTP) 110 (POP) y 143 (IMAP) mediante el comando `netstat -a -p TCP -n`.

En la ventana de bienvenida del administrador podremos añadir los dominios de correo a administrar activando el botón *AddDomain*. Usaremos el dominio `aula.izv`. Creado el dominio podemos añadir cuentas de correo para dicho dominio mediante *Accounts*. Crearemos por ejemplo las cuentas *cuenta1* y *cuenta2*.

Por defecto, cuando un usuario comete tres errores de conexión con el servidor, lo bloquea. Para las pruebas, vamos a deshabilitar esta opción. Iremos a *Settings -> Advanced -> Auto-ban* y desmarcaremos la casilla *Enabled*.

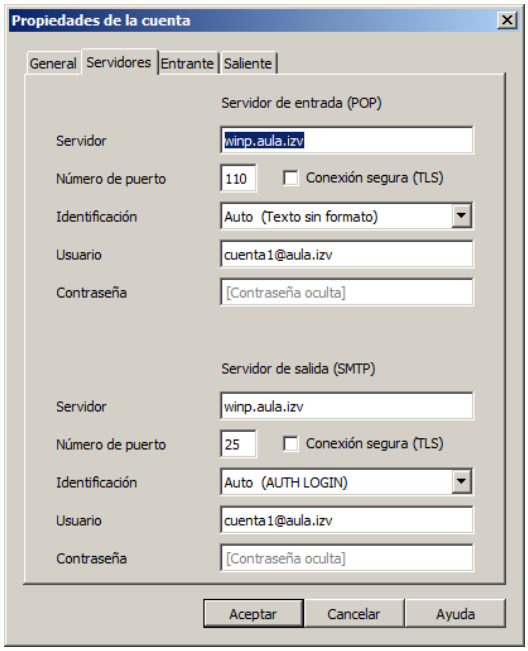
Después de todos los cambios deberemos reiniciar el servidor.

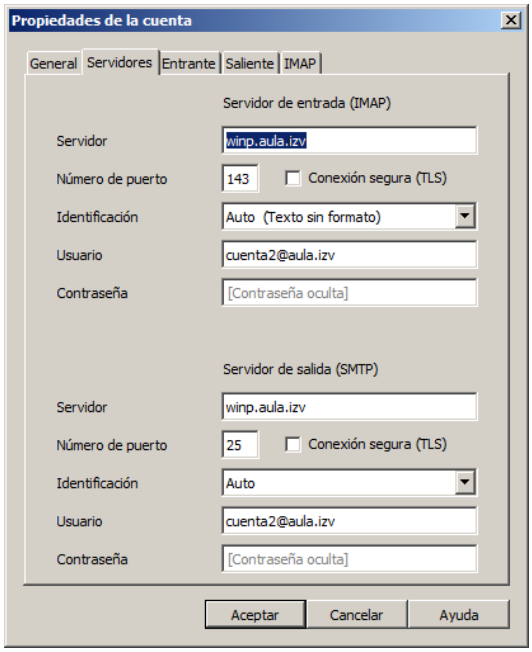
Ya es posible enviar y recibir mensajes de forma local para las cuentas del dominio `aula.izv` creadas anteriormente. Pero antes debemos añadir una regla de entrada en el *Firewall* de Windows 2012 Server para que `hMailServer` pueda abrir los puertos necesarios para las conexiones SMTP, POP e IMAP.

Para las pruebas podemos usar cualquier cliente de correo. Instalaremos en Windows Opera Mail. En el proceso de instalación crearemos inicialmente una cuenta de correo tipo POP para el usuario *cuenta1* con dirección cuenta1@aula.izv y nombre de usuario cuenta1@aula.izv.

Como servidor de correo entrante (POP) y de salida (SMTP) indicaremos el FQDN de nuestra máquina Windows Server, en nuestro caso winp.aula.izv. No marcamos la opción de usar conexiones seguras (TLS) ya que nuestro servidor de correo no está todavía configurado para ello.

Desde la opción *Cuentas de correo...* del botón *Opera Mail* de la barra de menús de la aplicación, se presenta *Administración de cuentas*. Desde aquí crearemos de forma análoga otra cuenta para el usuario *cuenta2*, pero esta vez de tipo IMAP. Las figuras muestran la configuración de las cuentas obtenidas desde *Administración de cuentas*.

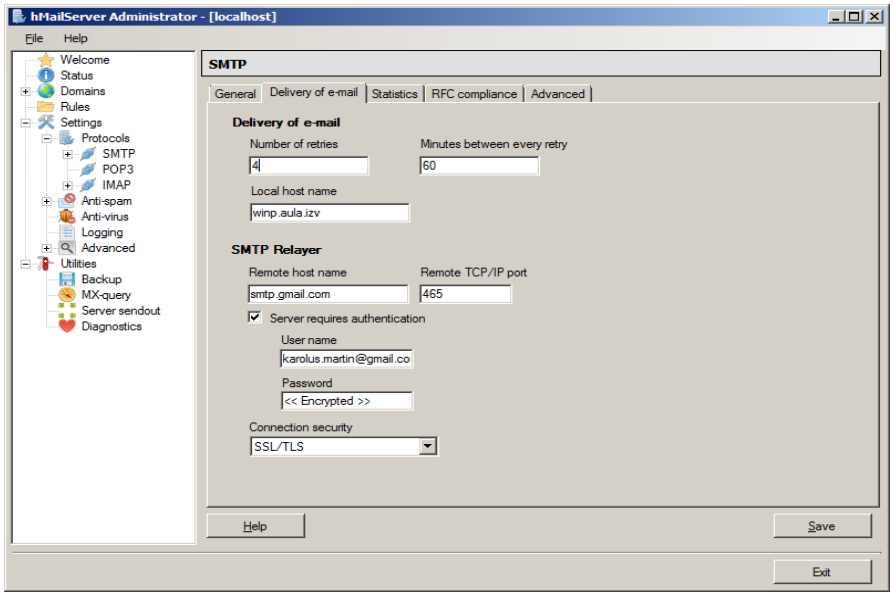




Ahora se puede probar el servicio creando un mensaje desde una cuenta y enviarlo a la otra.

La posibilidad de usar un servidor de correo externo que haga de *relay* para enviar mensajes dirigidos a cuentas de correo no locales depende de las posibilidades de dicho servidor de correo. Por ejemplo, con Gmail podemos hacerlo, incluso si la IP de nuestro servidor de correo local usa IP dinámicas. Como es lógico es necesario tener una cuenta en Gmail.

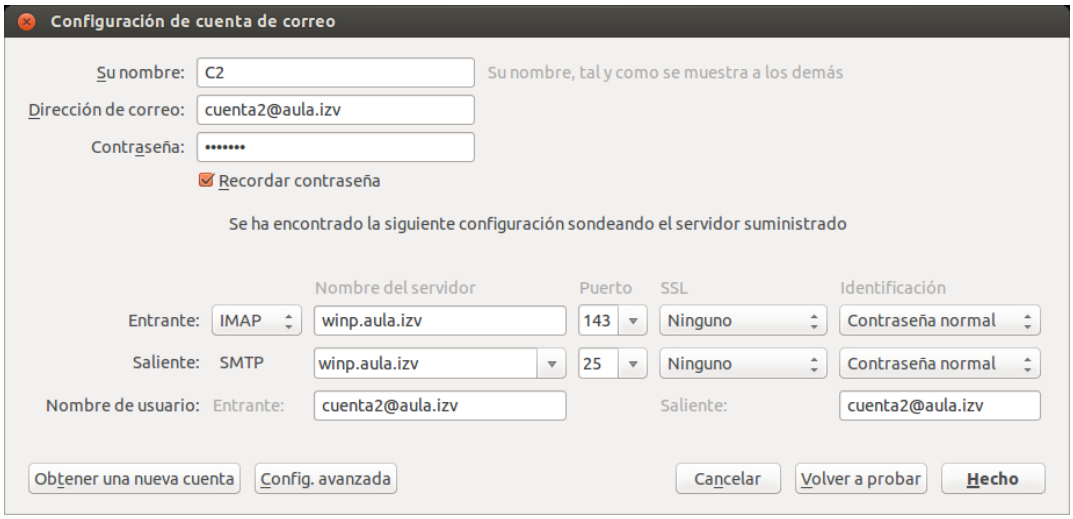
Para ello debemos activar *Settings -> Protocols ->SMTP* y seleccionar la ficha *Delivery of e-mail* e indicar los datos de conexión tal como aparecen en la figura siguiente:



Para probar el reenvío, deberíamos componer un mensaje usando como remitente una cuenta de aula.izv, como por ejemplo *cuenta1@aula.izv* y como destinatario una cuenta de un dominio externo, como por ejemplo *usuario@hotmail.com*.

Para probar servicio de correo desde un cliente en Linux, podemos usar el cliente *Thunderbird* que ya viene instalado en varias distribuciones de Ubuntu. Las cuentas se crean desde la opción *Herramientas -> Configuración de cuenta -> Operaciones sobre la cuenta -> Añadir cuenta de correo*.

Crearemos una cuenta IMAP para *cuenta2@aula.izv*. Usaremos la configuración de la imagen:



Existe un botón para comprobar la configuración correcta de una cuenta.

Sólo basta enviar un mensaje con la cuenta *cuenta2* y enviarlo a la *cuenta1* y comprobar que el mensaje se puede enviar. Para probar la recepción, desde Windows componemos un mensaje con la *cuenta1* y lo enviamos a la *cuenta2* y lo recuperamos en Linux posteriormente con IMAP.

En Linux podemos probar con otro cliente como es *Claws Mail*. A la hora de configurar una cuenta hay que tener cuidado cuando nos pide el *Usuario* ya que hay que indicar lo mismo que la dirección de correo; así para la cuenta cuenta1@aula.izv el usuario es el mismo valor. Además en la configuración de la cuenta, en la pestaña *Enviar* hay que indicar explícitamente autentificación SMTP AUTH (no es el valor por defecto) e indicar el usuario (el mismo que la dirección de correo) y su contraseña. Por defecto en el "home" del usuario se crea una carpeta de nombre *Mail* conteniendo varias subcarpetas para organizar los mensajes enviados, recibido, borradores, papelera, etc.

La configuración de hMailServer por defecto no usa protocolos seguros. Podemos habilitarlos mediante los protocolos STMPS (486), POPS (995) e IMAPS (993), o bien usar la extensión STARTTLS manteniéndose los puertos estándar.

Lo primero es disponer de un certificado de servidor y una clave privada para éste. Si usamos OpenSSL para ello, cuando creemos la clave privada deberemos crearla sin contraseña de protección. Para ello se ejecuta el comando habitual (ver tema 4), pero sin el modificador "-des3". Suponiendo que disponemos de ambos, desde *Settings -> Advanced -> SSL Certificates* pulsamos el botón *Add* e indicamos un nombre para el certificado, así como los archivos del certificado y la clave privada.

Para habilitar STMPS, POPS e IMAPS activamos *Settings -> Advanced ->TCP/IP ports* y añadimos para cada protocolo:

SMTP	IP del servidor	puerto:465	Seguridad: SSL/TLS Nombre del certificado
POP	IP del servidor	puerto:995	Seguridad: SSL/TLS Nombre del certificado
IMAP	IP del servidor	puerto:993	Seguridad: SSL/TLS Nombre del certificado

Finalmente habrá que reiniciar el servidor. Ahora habrá que modificar todos los clientes para indicar para cada protocolo los nuevos puertos, y marcar de alguna forma que las conexiones deben ser seguras.

Si vamos a usar la extensión STARTTLS, entonces no necesitamos añadir los nuevos protocolos y sus puertos. Basta con modificar los ya existentes, manteniendo los puertos estándar pero indicando como *Conecction Security* la opción *STARTTLS (Required)* y el nombre del certificado creado anteriormente. Por supuesto los clientes deben ajustarse a esta nueva configuración.

Ahora las conexiones presentarán un mensaje advirtiendo que el servidor presenta certificados autofirmados, los cuales deberemos aceptar explícitamente para poder enviar y recoger mensajes.

12. Servidor de correo en Linux.

Los pasos necesarios para instalar un sistema de gestión de correo son: configurar el DNS del dominio indicando cual va a ser el equipo servidor de correo en dicho dominio, instalar y configurar el servidor de correo (MTA), e instalar y configurar el servidor POP/IMAP.

Supondremos que nuestro equipo Linux se denomina ubu y será el servidor de correo del dominio aula.izv. Para ello añadiremos o modificamos en el servidor DNS del dominio, un registro de tipo *MX* con el nombre del servidor de correo ubu.aula.izv.

Como MTA usaremos *Postfix* ya que es relativamente fácil de configurar y administrar en comparación con *sendmail*, y se puede ampliar su funcionalidad a través de módulos. El servidor POP/IMAP que emplearemos será *Dovecot*, ya que se integra muy bien con *Postfix*.

12.1. Instalación y configuración del MTA Postfix.

Para instalar el paquete *Postfix* ejecutaremos en un terminal:

```
$sudo apt-get install postfix
```

Durante el proceso de instalación del paquete, una ventana nos pedirá qué tipo de servidor de correo se necesita para ajustar la configuración inicial. Seleccionamos la opción "*Internet*", y como nombre del sistema de correo aula.izv, es decir, el nombre de dominio que el MTA deberá gestionar.

Las otras opciones de configuración inicial que podemos establecer son:

- Sin configuración. No realiza ninguna modificación en la configuración inicial de Postfix.
- Internet con *smarthost*. El servidor no enviará los mensajes directamente a los destinatarios, sino a través de otro servidor de correo. Se suele usar si no disponemos de una IP fija.
- Sistema satélite. Configura el servidor sólo como *smarthost.*
- Solo correo local. Se trabaja sin red y sólo entrega correo a los usuarios locales del propio equipo.

Establecido el tipo de configuración, continúa la instalación y se inicia el servicio.

El servicio postfix se manipula usando los siguientes comandos dependiendo de cuál sea la versión del sistema de inicio y gestión de servicios de la distribución empleada.

```
$ sudo /etc/init.d/postfix {start | stop | restart | reload | status}
$ sudo service postfix {start | stop | restart | reload | status}
$ sudo systemctl {start | stop | restart | reload | status} postfix
```

Terminada la instalación se habrán creado los archivos de configuración, el usuario *postifx* que se incluye en el grupo del mismo nombre, y los certificados digitales autofirmados para poder realizar conexiones seguras usando SMTPS o mediante STARTTLS.

Una vez instalado deberíamos comprobar que el servidor está iniciado y el servicio SMTP está a la escucha por el puerto 25/TCP con

```
$ps –ef | grep postfix
$netstat -ltn
```

Los archivos de configuración principales de *Postfix* son:

- /etc/postfix/main.cf. Este es el archivo de configuración principal. En él se definen una serie de parámetros que establecen una configuración mínima del servicio. Los parámetros usan el formato *parámetro=valor* y los comentarios a nivel de línea empiezan con "#". Se puede usar *$parámetro* para asignar el valor de un parámetro a otro. Los parámetros que no se especifican en el archivo de configuración usarán un valor por defecto.

  Para ver todos los parámetros de configuración que actualmente usa el servicio se usa el comando:

  ```
  $sudo postconf
  ```

- /etc/postfix/master.cf. Archivo donde se establece la configuración de los servicios y cómo se conectan los clientes al servidor.

- /etc/aliases. Este archivo permite definir alias de cuentas de correo. Es casi obligado establecer un alias para la cuenta *root*, de forma que los mensajes que reciba el usuario *root*, se envíen a un usuario normal. Para ello se añade al archivo la línea:

  root otro_usuario
- /var/log/mail.log. Archivo principal de *logs* donde se registra todo lo que suceda relacionado con el envío de correo.
- /var/log/mail.info. Archivo donde se registran las acciones del servidor.
- /var/log/mail.err. Archivo donde se registran los errores.
- /var/log/mail.warn. Archivo donde se registran los avisos.

En *Postfix* los usuarios pueden ser locales (usuarios que tienen cuenta en el equipo) o virtuales (usuarios cuyas credenciales se almacenan en archivos, bases de datos, servicios de directorios, etc.).

12.1.1. Tipos de buzones de correo (mailbox).

Existen diferentes formas de organizar los buzones de correo. Básicamente se usan dos formatos diferentes: ***mbox*** y ***maildir***.

En el formato *mbox*, los mensajes enviados a un usuario se van añadiendo sucesivamente en un único archivo. Esto implica que se necesita algún tipo de bloqueo para evitar que éste pueda corromperse cuando dos o más procesos accedan simultáneamente al archivo buzón.

El formato *maildir* no bloquea los ficheros para mantener la integridad del mensaje, porque cada mensaje se almacena en un archivo distinto con un nombre único. Cada usuario debe disponer de un directorio donde almacenar los archivos con los mensajes, normalmente llamado Maildir para organizar los mensajes. Para ello se crean tres subdirectorios llamados: tmp (mensajes pendientes de entregar), new (mensajes nuevos), y cur (mensajes ya leídos)

En los sistemas UNIX tradicionalmente se ha usado el tipo *mbox*, de manera que para cada usuario que recibe mensajes, se crea un archivo en donde se van acumulando todos los mensajes recibidos. Este archivo, llamado *buzón de entrada* suele ser /var/mail/usuario o bien /var/spool/mail/usuario. En particular, Postfix tiene establecido por defecto cual es el directorio donde almacena este buzón de entrada a través del parámetro **mail_spool_directory=/var/mail**. Este sistema es compatible con el estándar *sendmail*, pero no es seguro ya que todos los mensajes de todos los usuarios se almacenan en el mismo directorio.

Cuando una aplicación cliente lee los mensajes, éstos se mueven desde su *buzón de entrada* a su buzón particular llamado *buzón de correo*. En Postfix, por defecto, cuando un usuario lee su correo de forma local, (sin usar un servicio POP/IMAP), el archivo buzón de correo se crea en su $HOME. Así cada usuario tiene su archivo de mensajes leídos (su buzón) en su carpeta personal. El nombre de este archivo buzón puede variar y dependerá de cómo esté configurado el cliente de correo. Este comportamiento se debe a que en Postfix la directiva **home_mailbox** por defecto no tiene asignado ningún valor.

Pero si a **home_mailbox** le asignamos un nombre de archivo, entonces en el $HOME de cada usuario se creará dicho archivo y se usará como *buzón de entrada*, es decir, en dicho archivo se guardarán todos los mensajes que reciba dicho usuario (ya no se guardarán en /var/mail/usuario). Luego cuando una aplicación cliente lea los mensajes, los moverá a una carpeta que dependerá de cómo esté configurado dicho cliente.

Si quisiéramos buzones de tipo *maildir*, también llamados *compatibles con qmail*, tendríamos que indicar a Postfix el directorio donde cada usuario tendrá sus mensajes. Para ello también se usa el comando **home_mailbox**, pero para indicar que es un directorio y no un archivo en formato *mbox*, se añade una / al final del nombre del directorio de la forma **home_mailbox=nombre_directorio_maildir/.**

Por lo tanto, si mediante **home_mailbox** establecemos los buzones como tipo *maildir*, el *buzón de entrada* estará en la carpeta **$HOME/nombre_directorio_maildir/new**, y el *buzón de correo*, donde se guardan los mensajes leídos estará en **$HOME/nombre_directorio_maildir/cur**.

Tipo de buzón	Buzón de entrada	Buzón de salida
Por defecto (mbox) home_mailbox=	/var/mail/usuario	Depende de la aplicación que lee
(mbox) home_mailbox=archivo	$HOME/archivo	Depende de la aplicación que lee
(maildir) home_mailbox=carpeta/	$HOME/carpeta/new	$HOME/carpeta/cur

12.1.2. Configuración por defecto de *Postfix*.

La configuración por defecto establecida durante la instalación de *Postfix* es la siguiente:

- Gestiona buzones de cuentas de usuarios locales con el nombre de dominio elegido durante la instalación. En nuestro caso, como elegimos como dominio aula.izv, sólo se gestionarán buzones de cuentas del tipo usuario@aula.izv, donde usuario deber ser un usuario local del equipo donde corre Postfix.

  Por ejemplo, si *jose* y *maria* son los dos únicos usuarios con cuenta en el servidor, solo se admitirán mensajes locales dirigidos a jose@aula.izv o a maria@aula.izv, pero no a rafael@aula.izv.

- El servidor no pide autentificación. Cualquiera puede enviar mensajes destinados a cualquier usuario de cualquier dominio, pero sólo si han sido enviados desde el propio equipo. (En principio, sólo el equipo local es "confiable"…).

  Esto es así porque los equipos "confiables" son los que pertenecen a 127.0.0.0/8. Si se quiere ampliar el envío desde otros equipos de otras redes, habría que añadir las IP de dichos equipos.

  Lo normal, sería que el servidor pida autentificación. Es decir, considerar equipos confiables no a los que pertenezcan a una red determinada, sino aquellos clientes que se puedan autenticar adecuadamente (AUTH SMTP).

- Permite, pero no obliga, conexiones seguras mediante STARTTLS usando un certificado digital autofirmado. Las conexiones seguras y las no seguras usan el mismo puerto 25/TCP.

- Por defecto NO está configurado para conexiones seguras (SMTPS) mediante SSL/TLS.

- Usa buzones tipo *mbox*. El archivo que hace de buzón de entrada se denomina /var/mail/usuario, y cuando el usuario lea sus mensajes, éstos se guardan en un archivo en el $HOME del usuario.

En Linux la aplicación GNU *Mailutils* proporciona un conjunto de librerías y utilidades para gestionar el correo electrónico. En particular proporciona un cliente de correo para enviar y leer mensajes. Por defecto este cliente opera con el correo local con buzones de tipo *mbox*. Usa como buzón de entrada la ruta /var/mail/usuario y deposita los mensajes leídos en /home/usuario/mbox.

Vamos a crear, si no existen, un par de usuarios locales: *jose* y *maria*.

- Iniciamos sesión como usuario *jose* y con el comando *mail* de Mailutils enviamos un mensaje a *maria*.

```
jose@ubu:~$mail maria@aula.izv
Cc:
Subject: Saludos
Hola Maria:
¿Cómo te va?
Que si tal, que si cual...
Hasta pronto
.
jose@ubu:~$
```

El mensaje se termina con una línea donde se escribe un sólo punto; y para enviarlo se usa Ctrl+D.

- Si queremos comprobar los mensajes que están en la cola de envíos del servidor SMTP usamos:

```
$mailq
```

- Si la cola está vacía, significa que el mensaje ya ha sido enviado.

Por defecto Postfix usa como buzón de entrada para *maría* el archivo /var/mail/maria. Podemos iniciar sesión como *maria* y comprobar que el mensaje se ha añadido en /var/mail/maria. También podemos consultar los archivos de *logs* para ver cómo ha quedado registrado el mensaje. Todos los mensajes lo podemos leer editando directamente el archivo, pero esto no es muy operativo. Lo usual es usar una aplicación que acceda a dicho buzón de entrada y lo muestre

- Para leer el mensaje de forma local, teclearemos el comando *mail* de Mailutils, y como por defecto busca los mensajes en /var/mail/maria lo encontrará:

```
$maria@ubu:~$mail
```

Se nos mostrará todos los correos nuevos con un número identificador de mensaje:

```
Mail version 8.1.2 01/15/2001.  Type ? for help.
"/var/mail/maria": 3 messages 3 new
>N  1 jose@aula.izv  Thu Jan 26 12:46   14/489   prueba
 N  2 jose@aula.izv  Thu Jan 26 12:46   14/498   Saludos
```

Tecleando el número del mensaje leeremos el e-mail directamente:

```
& 1
Message 1:
From jose@aula.izv  Thu Jan 26 12:46:46 2012
X-Original-To: maria
To: maria@aula.izv
Subject: prueba
Date: Thu, 26 Jan 2012 12:46:46 +0100 (CET)
From: jose@aula.izv  (jose)

 Mi primera prueba
```

- Para borrar todos los mensajes se usa:

  & d*

- Para borrar un mensaje particular numerado como "X":

  & dX

- Para obtener más ayuda sobre los comandos:

  & d?

Por defecto los mensajes que se van leyendo se van moviendo de /var/mail/maria a /home/maria/mbox.

Ahora usaremos el cliente *Claws Mail* para leer los mensajes de correo, pero antes componemos un mensaje de *jose* para *maría* y lo enviamos. El mensaje quedará depositado en /var/mail/maria.

Para configurar en una cuenta en *Claws Mail* para *maría*, debemos iniciar sesión como maría, lanzar la aplicación *Claws Mail* y suministrar al asistente los siguientes datos:

- Debemos indicar el tipo de buzón como "*fichero local* (*mbox*)", y la ruta del buzón (/var/mail/maria). Hay que tener en cuenta que no tenemos instalado un servicio POP/IMAP y que el acceso al buzón se hará de forma local.
- En la configuración de envío, no hay que configurar ningún método de autentificación (por defecto no está establecido en el servidor) ni de cifrado (está configurado como voluntario, no obligatorio). Por último decidimos el nombre de la carpeta local donde se clasificarán los mensajes de correo. Por defecto será la carpeta local "Mail".

Cuando leamos los mensajes de *maría* con *Claws Mail* se guardarán en /home/maria/Mail/inbox.

Si en Postfix establecemos en el archivo de configuración /etc/postfix/main.cf el parámetro:

```
home_mailbox=buzon_entrada
```

y reiniciamos el servicio, habremos cambiado el lugar del buzón de entrada, y los mensajes para *maría* se guardarán en el archivo /home/maria/buzon_entrada. Ahora con el comando *mail* de Mailutils no podremos leer los mensajes de *maria* ya que por defecto los busca en /var/mail/maria, Tendríamos que reconfigurar el comando *mail* de Mailutils para que los buscara en el nuevo buzón. Igual pasaría con *Claws Mail*, necesitaríamos modificar la cuenta de *maría* indicando que la ruta del *fichero local* (*mbox*)", es /home/maria/buzon_entrada.

12.1.3. Configurando *Postfix*.

La configuración de *Postfix* se realiza en el archivo /etc/postfix/main.cf mediante parámetros. Para ver los parámetros definidos explícitamente que no tienen valor por defecto se usa la orden:

```
$ sudo postconf -n
```

Y para ver todos los parámetros con sus valores por defecto:

```
$ sudo postconf -d
```

A un parámetro se le puede asignar el valor de otro, para ello se usa el símbolo $ por delante del nombre del parámetro para hacer referencia a su valor.

Algunos de los parámetros básicos y de mayor uso con sus valores por defecto en *main.cf* son:

- myhostname. Establece el nombre del equipo servidor MTA. Se fija durante la instalación.
  myhostname = ubup
- smtpd_banner. Establece el mensaje que el MTA mostrará a los clientes cuando se conecten.
  smtpd_banner = $myhostname ESMTP $mailname

- mydomain. Indica el nombre de dominio del equipo servidor MTA.
  mydomain = localdomain
- myorigin. Establece el nombre del dominio para el correo saliente, es decir, el que aparece en los mensajes enviados desde el equipo. Además, este nombre se agregará por defecto a toda dirección de destinatario cuyo dominio no se especifique.
  myorigin = /etc/mailname
- mydestination. Dominios que se aceptan como correo entrante local. Se pueden especificar varios dominios. Es decir, los mensajes dirigidos a estos dominios serán para reparto local (*local delivery*). Los que no se dirigen a estos dominios se enviarán a esos otros dominios (*forward*).
  mydestination = aula.izv, ubu, localhost.localdomain, localhost
- mynetworks. IPs desde las que el servidor puede aceptar correos para enviarlos a sus destinatarios. Por así decirlo son los equipos o redes "confiables". Atención: si se permitieran de todas las redes y no se configurara autentificación SMTP el servidor se consideraría *open relay*.
  mynetworks = 127.0.0.0/8 [::ffff:127.0.0.0]/104 [::1]/128
- relay_domains. De forma complementaria al parámetro anterior, el servidor de correo también aceptará mensajes si se dirigen a ciertos destinos. Por defecto a los especificados en mydestination
  relay_domains = $ mydestination
- relayhost. Indica el servidor al que se reenvía el correo para que sea este quien lo envíe a los destinatarios (*smart host*). Si no se especifica, el correo se envía directamente a los destinatarios.
  relayhost =
- mailbox_size_limit. Tamaño máximo en bytes de los buzones de correo. El valor 0 indica sin límite.
  mailbox_size_limit = 0
- message_size_limit. Tamaño máximo en bytes de los mensajes. El valor 0 indica sin límite.
  message_size_limit = 10240000
- home_mailbox. Indica el nombre del archivo (*mbox*) o carpeta (*maildir*) del buzón del usuario.
  home_mailbox =
- inet_interfaces. Interfaces de red en las que *Postfix* escuchará las peticiones de los clientes.
  inet_interfaces = all
- smtpd_use_tls. Indica la disponibilidad de usar la extensión STARTTLS. El valor yes permite al cliente el cifrado de la conexión, pero no le obliga, A partir de la versión 2.3 es aconsejable usar smtpd_tls_security_level con los valores no, may (disponible, pero no obligatorio) y encrypt.
  smtpd_use_tls = yes
- luser_relay. Indica la dirección que recogerá todos los mensajes dirigidos a usuarios del dominio desconocidos. Si no se especifica ninguna dirección, dichos mensajes serán rechazados.
  luser_relay =

12.1.4. Ejemplo configuración *Postfix*.

Supongamos que editamos el archivo /etc/postfix/main.cf con estos parámetros

```
# Indica formato buzón tipo maildir en la carpeta $HOME/Maildir
home_mailbox = Maildir/

# Se aceptarán mensajes provenientes desde del equipo local y de la red 192.168.210.0/24 para su reenvío
mynetworks = 127.0.0.0/8 192.168.210.0/24

# Dominio que aparece en los mensajes enviados
myorigin = aula.izv

# Dominios que se aceptan como correo entrante local y que no se reenviarán
mydestination = aula.izv, ubu, localhost.domain, localhost

# Tamaño máximo de los buzones de correo: 50Mb
mailbox_size_limit = 52437800

# Tamaño máximo de los mensajes de correo: 1Mb
message_size_limit = 1048576
```

Con esta configuración, se acepta enviar mensajes desde el equipo local y desde los equipos de la red 192.168.210.0/24 (red donde se encuentra el servidor de correo), y dirigidos a cualquier dirección de correo. Los mensajes dirigidos a cuentas locales, serán del tipo usuario@aula.izv, donde usuario debe ser un usuario con cuenta en el servidor.

Ojo, recordar que un servidor de correo puede permitir un mensaje y colocarlo en la cola para su entrega, pero luego ser incapaz de entregarlo porque por ejemplo el servidor no tiene una IP pública fija o porque no existe un registro MX en un servidor DNS en el que aparezca registrado el servidor.

Además, los buzones usarán el formato *maildir* y se ubicarán en el directorio $HOME/Maildir. Recordar que debemos reiniciar el servicio para que los cambios tengan efecto.

```
$sudo systemctl restart postfix
```

Si hacemos una prueba de correo usando el comando *mail*, antes debemos cambiar su configuración por defecto ya que ahora no tenemos buzones tipo *mbox*. Se puede lanzar la aplicación *mail* seguida de parámetros que indiquen que los buzones son de tipo *Maildir* y su ubicación, o bien crear un archivo de configuración (/etc/mailutils.conf) donde mediante directivas indicar dicha configuración.

En nuestro caso, si queremos indicar que *mail* debe usar buzones de tipo *maildir* localizados en el $HOME de cada usuario, añadiríamos (o crearíamos si no existe) al archivo /etc/mailutils.conf las siguientes directivas:

```
mailbox{
      mailbox-pattern "maildir:///home/${user}/Maildir";
      mailbox-type maildir;
}
```

Ahora podemos realizar una prueba de correo enviando un mensaje de *jose* a *maria* usando el comando *mail* de Mailutils. Podemos comprobar como los buzones se ubican en /home/maria/Maildir.

Como ya vimos anteriormente, *Claws Mail* permite el acceso a buzones locales, pero sólo si son de tipo *mbox*. No obstante se tiene la opción de establecer cuentas sólo para enviar. Cómo por ahora sólo queremos enviar un mensaje, crearemos una nueva cuenta para *jose* en *Claws Mail* de este tipo. Para ello:

- Iniciamos sesión como *jose* en el equipo servidor y ejecutamos *Claws Mail*. En el menú principal seleccionamos *Configuración -> Editar Cuenta -> Nuevo.*
- En la pestaña *Básicas* tecleamos los datos de identificación de la cuentas, y en lista *Protocolo* seleccionamos *Ninguno (sólo SMTP)*. Finalmente indicamos el nombre de dominio del servidor.
- En la pestaña *Enviar* por defecto no se incluye autentificación. No la indicamos ya que por defecto *Postfix* no la necesita.
- Ahora tendremos configuradas 2 cuentas para *jose*. Podemos indicar esta nueva cuenta como la predeterminada en la ventana *Editar Cuenta.*
- Ya podemos componer un mensaje, y enviarlo a *maria*.
- Iniciamos sesión como *maria* y antes de nada exploramos su buzón de entrada (ahora es el directorio /home/maria/Maildir/new) para comprobar si le ha llegado el mensaje. Ojo, los buzones de usuario no se crean por parte de *postfix* hasta que éstos no reciban un mensaje.

12.1.5. Ejemplo configuración *Postfix* usando STARTTLS (SSL/TLS).

Como comentamos, *postfix* ya está configurado por defecto para permitir, pero no obligar, conexiones SMTP cifradas por el puerto 25 usando STARTTLS con un certificado digital autofirmado. En la sección *#TLSparameters* de /etc/postfix/main.cf podemos ver los parámetros que permiten esta configuración:

```
smtpd_tls_cert_file=/etc/ssl/certs/ssl-cert-snakeoil.pem
smtpd_tls_key_file=/etc/ssl/private/ssl-cert-snakeoil.key
smtpd_use_tls=yes
# Para versiones posteriores a la 2.3 se aconseja usar smtpd_tls_security_level = may
```

Realizaremos una prueba. Para ello:

- Iniciamos sesión como *jose* y modificamos su cuenta en *Claws Mail* usando STARTTLS como método de seguridad para enviar correo. (Mantenemos el resto de la configuración).
- Enviamos un mensaje a *maria*. Al enviar el mensaje, el servidor nos presentará su certificado digital. Como es autofirmado y no lo puede verificar, pide que lo confirmemos.
- Iniciamos sesión como *maria* y en /home/maria/Maildir/new podremos ver el mensaje almacenado.

13. Instalación y configuración del servidor POP/IMAP *Dovecot*.

Hasta ahora hemos accedido a los buzones de correo de los usuarios iniciando sesión en el mismo equipo servidor de forma local, pero esto no es lo habitual. Lo normal es que los usuarios accedan a sus mensajes remotamente desde sus puestos de trabajo usando un servidor POP/IMAP para la entrega.

Con la aplicación *Dovecot* podemos instalar un servidor POP, un servidor IMAP o ambos mediante paquetes independientes. En nuestro caso instalaremos ambos tipos de servidores de entrega de correo.

Para instalar ambos paquetes ejecutaremos en un terminal:

```
$sudo apt-get install dovecot-imapd dovecot-pop3d
```

Durante la instalación se crearán los archivos de configuración en los directorios /etc/dovecot/ y /usr/share/dovecot.

El servicio *dovecot* se manipula usando los siguientes comandos dependiendo de cuál sea la versión del sistema de inicio y gestión de servicios de la distribución empleada.

```
$ sudo /etc/init.d/dovecot {start | stop | restart | reload | status}
$ sudo service dovecot {start | stop | restart | reload | status}
$ sudo systemctl {start | stop | restart | reload | status} dovecot
```

El archivo principal de configuración es dovecot.conf, el cual contiene la directiva !include conf.d/*.conf para incluir el resto de la configuración que se distribuye en diferentes archivos agrupando directivas según una misma funcionalidad. Estos archivos de configuración son leídos en orden alfabético.

Para mostrar los parámetros definidos explícitamente en los ficheros de configuración se usa:

```
# dovecot -n
```

Y para mostrar todos los parámetros, incluidos los que conservan su valor por defecto.

```
# dovecot -a
```

Deberíamos comprobar que el servidor está a la escucha en los puertos TCP/110 (POP), TCP/143 (IMAP), mediante el comando netstat -ltn

13.1.1. Configuración de los tipos de buzones

En el archivo **/etc/dovecot/conf.d/10-mail.conf**, mediante el parámetro mail_location se establece el formato y ubicación de los buzones de los usuarios. Por defecto se establecen buzones de tipo *mbox* con el buzón de entrada en la carpeta /var/mail/usuario. Se propone que cuando el cliente lea los mensajes los organice dentro de una carpeta en su $HOME denominada mail, pero esto dependerá de la aplicación cliente que puede organizar los mensajes que lee en otra ubicación.

Esta configuración por defecto queda establecida con el parámetro mail_location

```
mail_location = mbox:~/mail:INBOX=/var/mail/%u
```

La manera de fijar el buzón dependerá del tipo de éste. Por lo tanto:

a) Si el buzón tiene el formato *mbox*, usaríamos:

```
mail_location = mbox:~/carpeta:INBOX=/var/mail/%u
```

Donde estaríamos indicando:

- El tipo de buzón: mbox
- La ubicación del buzón de correo de cada usuario: ~/carpeta. En este caso estará en el *home* de cada usuario y se llamará "carpeta". En las conexiones POP, los mensajes leídos se organizan dentro de esta "carpeta" (normalmente en una subcarpeta INBOX). En conexiones IMAP, los mensajes permanecen siempre en el servidor y no se descargan.
- El buzón de entrada: /var/mail/%u. En este caso se ubica en /var/mail/usuario.

IMPORTANTE. Algunos clientes de correo obtienen el nombre de la "carpeta" de forma automática del servidor y la crean pero no la utilizan. Otros esperan que la carpeta se denomine mail o Mail y las crean con estos nombres. En otros clientes el nombre es configurable, como es el caso de *ClawsMail*, que propone como nombre Mail, pero se puede cambiar.

b) Si usamos buzones tipo *maildir* usaríamos:

mail_location = maildir:~/carpeta

Donde se indica:
- El tipo de buzón: maildir
- La ubicación del buzón de correo de cada usuario: ~/carpeta. En este caso estará en el *home* de cada usuario y se llamará "carpeta".

Normalmente el nombre de la carpeta buzón se denominara Maildir, pero, si la carpeta la denomináramos con otro nombre, necesitaríamos explícitamente indicarlo con el parámetro mail_location.

Tipo buzón	**Buzón de entrada**	**Buzón de correo**
Por defecto *mbox*	Por defecto busca en /var/mail/usuario Modificable mediante mail_location= mbox:~/carpeta:INBOX=**/var/mail/%u**	Se establece mediante mail_location= mbox:~/**carpeta**:INBOX=/var/mail/%u (su uso depende del cliente)
maildir	Por defecto busca en $HOME/**Maildir**/new Modificable mediante mail_location = maildir:~/**carpeta**	Por defecto $HOME/**Maildir**/cur Modificable mediante mail_location = maildir:~/**carpeta**

13.1.2. Configuración por defecto de Dovecot.

Entre los diferentes ficheros de dovecot y la propia configuración predeterminada encontramos estos valores:

```
# protocolos habilitados
protocols = "imap pop3"
# Tipos de buzón y localización
mail_location = mbox:~/mail:INBOX=/var/mail/%u
# Mecanismos de autenticación. Soporta: plain, login, cram-md5, digest-md5, ntlm, …
auth_mechanisms = plain
# Aunque esté comentado, el valor por defecto es "yes". Es decir NO se permite autentificación con texto
# plano salvo que SSL/TLS esté disponible; (por defecto SSL/TLS no está disponible).
#disable_plaintext_auth = yes
# Para establecer conexiones seguras. Valores "yes" (disponible pero no obligatorio)  o "required" (obligatorio)
ssl = no
# Certificado del servidor y clave primaria.
#ssl_cert = </etc/dovecot/dovecot.pem
#ssl_key = </etc/dovecot/private/dovecot.pem
```

La configuración por defecto indica:

- Buzones tipo *mbox* con el buzón de entrada en /var/mail/usuario. Se creará la carpeta /home/usuario/mail para organizar los mensajes leídos, pero el cliente de correo puede utilizar otra.
- El sistema de autentificación envía la contraseña en texto plano.
- No se permite autentificación en texto plano salvo que la conexión sea cifrada.
- La conexión no se realiza cifrada.

Con esta configuración por defecto NO se puede acceder a los buzones ya que una contraseña en texto plano debería enviarse cifrada, pero el cifrado está deshabilitado. (En realidad se puede acceder a los buzones si la conexión se realiza sólo desde el propio equipo servidor, ya que se considera "fiable". Desde cualquier otro equipo no se podría).

13.1.3. Ejemplo de configuración de Dovecot. Acceso a buzones.

Tenemos dos soluciones para poder acceder a los buzones: o permitimos la autentificación con texto plano sin cifrar la conexión, o ciframos la conexión para que se permita.

Cambiaremos la configuración para usar buzones tipo *maildir* y para permitir autentificación con texto plano sin cifrar la conexión. Para ello:

a) Vamos a editar el archivo **/etc/dovecot/conf.d/10-mail.conf** de manera que quede de la forma:

```
mail_location = maildir: ~/Maildir          # formato y localización de los buzones de los usuarios
```

b) Editamos el archivo **/etc/dovecot/conf.d/10-auth.conf**

```
disable_plaintext_auth = no      # se permite autentificación mediante texto plano aunque no se use cifrado
```

c) Después de salvar los cambios debemos reiniciar el servicio.

Con estas modificaciones, el servidor estará habilitado para:

- Usar los protocolos POP/IMAP.
- Gestionar **buzones con formato *maildir* localizados en $HOME/Maildir**
- Se permite la **autentificación de tipo texto plano** aunque la conexión no esté cifrada.

Configuración "automática" de clientes de correo

Algunos clientes de correo están programados para intentar por varios mecanismos distintos, configurar de forma automática los parámetros referentes a los servidores de correo, puertos usados, cifrados, etc. sobre todo si se tratan de cuentas pertenecientes a los principales gestores de correo: Gmail, Hotmail, Exchange, etc.

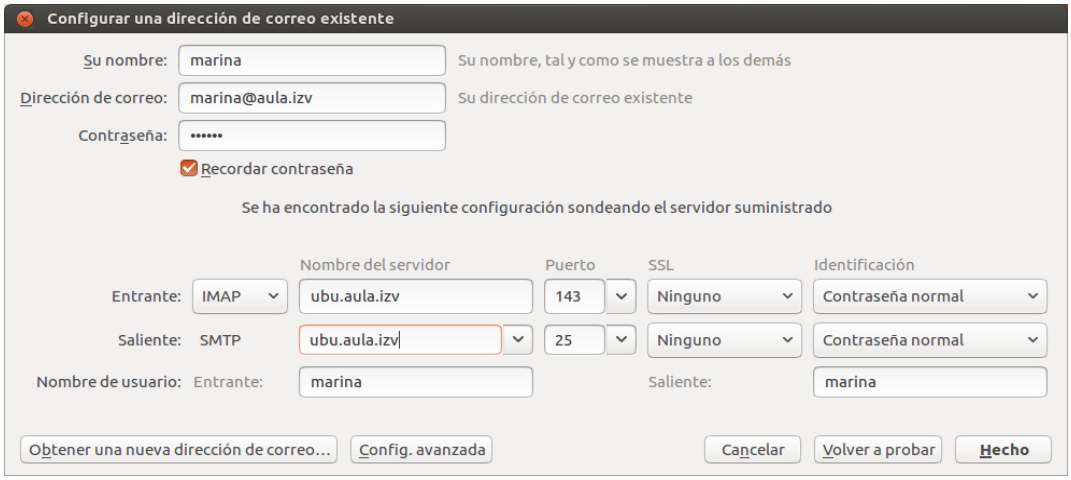
En instalaciones más modestas, y si tenemos capacidad de gestionar el servidor DNS del dominio, podemos añadir registros en los archivos de zona que informen a los clientes de correo qué equipos del dominio gestionan el correo del dominio. Para ello deberíamos indicar con un registro MX cuál es el servidor de correo del dominio, y usar alias de los dominios que esperan encontrar los clientes de correo, imap.dominio, pop.dominio y smtp.dominio para indicar cuales son los servidores IMAP, POP y SMTP.

Por ejemplo, para nuestro caso en donde usamos el equipo ubup.aula.izv como servidor de correo donde corren los servicios SMTP, POP e IMAP, añadiríamos en el archivo de zona para el dominio aula.izv los siguientes registros:

```
ubup.aula.izv.   IN      A       192.168.1.100
aula.izv.        IN      MX  10  ubup.aula.izv.
smtp.aula.izv.   IN      CNAME   ubup.aula.izv.
imap.aula.izv.   IN      CNAME   ubup.aula.izv.
pop.aula.izv.    IN      CNAME   ubup.aula.izv.
```

El cliente de correo *Thunderbird* es uno de esos clientes que intentan realizar la configuración automática de las cuentas de correo. Para probar la última configuración del servicio de correo, vamos a iniciar sesión con el usuario *maria* y crear una cuenta de correo con el cliente *Thunderbird*.

Ahora podemos elegir buzones tipo POP o IMAP. Optaremos por indicar como servidor de correo tipo IMAP. Configurada la cuenta como indica la figura, comprobamos que podemos recuperar el correo del buzón de *maria*.



Si el cliente de correo no lograra finalmente la configuración automática, siempre podremos realizar de forma manual.

13.1.4. Ejemplo configuración *Dovecot* usando conexión segura (SSL/TLS).

Ahora usaremos una configuración que nos permita acceder a los buzones de forma segura mediante cifrado de la conexión. Para ello necesitamos un certificado de servidor y una clave privada para éste. Dependiendo de la versión de *dovecot*, puede que los archivos con los certificados estén ya creados.

En caso de necesitar crearlos o querer crear otros nuevos, el paquete dovecot incluye el *script* /usr/share/dovecot/mkcert.sh que permite crear fácilmente un certificado autofirmado y una clave privada.

El certificado se crea usando los parámetros incluidos en /usr/share/dovecot/dovecot-openssl.cnf. Lo más importante de este archivo de configuración es ajustar el parámetro CN con el nombre del dominio del servidor *dovecot.*

Para ejecutar el script simplemente haremos:

```
alumno@ubu:/usr/share/dovecot$ sudo ./mkcert.sh
```

Por defecto los dos archivos (certificado y clave privada) creados por el *script* se denominan /etc/dovecot/dovecot.pem y /etc/dovecot/private/dovecot.pem y serán los que deberemos indicar en el archivo **/etc/dovecot/conf.d/10-ssl.conf.**

- Editamos el archivo **/etc/dovecot/conf.d/10-ssl.conf**:

  ```
  # SSL/TLS está habilitado por defecto, no siendo obligatorio su uso por parte del cliente. Vamos a "obligarlo"
  ssl = required
  # Habilitamos el certificado del servidor y clave primaria quitando los comentarios
  ssl_cert = </etc/dovecot/dovecot.pem
  ssl_key = </etc/dovecot/private/dovecot.pem
  ```

- Editamos el archivo **/etc/dovecot/conf.d/10-auth.conf** y cambiamos el parámetro disable_plaintext_auth a valor "yes". De esta forma podremos usar autentificación mediante texto plano ya que la conexión estará cifrada.

  ```
  disable_plaintext_auth = yes
  ```

- Finalmente reiniciamos el servicio *dovecot.*

Cuando el parámetro ssl toma los valores yes o required, *dovecot* habilita los puertos 995 y 993 para usar conexiones seguras (POPS e IMAPS) y también mantiene los puertos estándar por si el cliente quiere realizar la conexión cifrada mediante el mecanismo STARTTLS. Podemos comprobarlo viendo los puertos que están a la escucha en el servidor.

Para probar la configuración, vamos a iniciar sesión con el usuario *maria* y añadimos al cliente *Claws Mail* una cuenta para que pueda acceder a su buzón. Optaremos por indicar como servidor de correo tipo POP3 y en la configuración para SSL/TLS podemos elegir:

- Opción *Usar STARTTLS para iniciar la sesión cifrada*. En cuyo caso se mantendría el puerto estándar 110 o bien:
- Opción *Usar SSL/TLS* para emplear el puerto de conexión 995.

Configurada la cuenta, la establecemos como *cuenta primaria*. Al leer el correo se presentará el típico mensaje de que el certificado del servidor es autofirmado, el cual deberemos aceptar.

En la URL http://wiki2.dovecot.org/QuickConfiguration podemos encontrar una guía para las múltiples configuraciones que admite *dovecot.*

14. Clientes de correo web (*webmail*).

Un *webmail* es un es un cliente de correo electrónico que provee una interfaz web para acceder al correo electrónico. Los clientes de webmail permiten recibir y enviar mensajes de correo electrónico utilizando los protocolos POP3/IMAP y SMTP desde servidores de correo tanto locales como remotos.

La ventaja de un cliente *webmail* es que permite a los usuarios gestionar su correo desde cualquier lugar ya que se trata de una aplicación web escrita en algún lenguaje de servidor (PHP, Perl, etc.) que solo necesita de un navegador o explorador web para su ejecución.

La mayoría de los *webmail* son de software libre. Algunos de ellos son Squirrelmail, RoundCube, Zimbra, Horde, Openwebmail y BlogMail.

Como ejemplo usaremos Squirrelmail (http://www.squirrelmail.org), un *webmail* multiplataforma escrito en PHP y que soporta los protocolos SMTP e IMAP. Por lo tanto para su funcionamiento se necesitará un servidor web con soporte PHP (tenemos *Apache*) y un servidor IMAP (tenemos Dovecot POP/IMAP).

El paquete correspondiente (*squirrelmail*) instala los módulos necesarios de PHP y de Apache para su funcionamiento. La configuración personalizada de la aplicación se realiza mediante la orden:

```
$sudo squirrelmail-configure
```

En el menú que aparece seleccionaremos en este orden:

- Opción D. Permite seleccionar con cual servidor de correo IMAP vamos a trabajar. Escribiremos *"dovecot"*.
- Opción 2 (*server settings*), seleccionamos la opción 1 (*domain*) y escribimos el dominio de nuestro sitio web: aula.izv. En este mismo menú podemos repasar si el nombre de la máquina y los puertos son los correctos. A continuación pulsamos la opción *R* para regresar al menú principal.
- En la opción 3 del menú principal (*Folder defaults*) vamos a especificar el nombre que queremos que se nos muestre para las carpetas *Papelera*, *Enviados* y *Borradores*. Podemos dejar los nombres por defecto que aparecen, o sustituirlas por otros nombres.
- Mediante la opción 10 (*Languages*) llegamos al menú *Language preferences*. Con la opción *Default Language* podemos escoger como idioma predeterminado español con *es_ES*.
- Finalmente con *S* salvamos los datos de configuración y con *Q* salimos del configurador.

Como el código PHP de Squirrelmail se localiza en /usr/share/squirrelmail, crearemos un enlace (que llamaremos webmail) desde el directorio base del servidor web a dicha carpeta. Para ello haremos:

```
$ cd /var/www/html        (o a la carpeta raíz del servidor web por defecto de Apache …)
$ sudo ln -s /usr/share/squirrelmail webmail
```

Reiniciamos Apache y escribimos en la barra de direcciones del navegador http://servidor-web/webmail donde servidor-web será el nombre de dominio o la dirección IP del servidor web donde corra el servidor *Apache*.

Si todo va bien aparecerá la página de *login* del webmail donde los usuarios se identificarán antes de acceder a la interfaz que permite gestionar sus mensajes.

Cabe la posibilidad de ejecutar una aplicación web que realiza un test sobre la configuración realizada de Squirrelmail mediante la URL http://servidor-web/webmail/src/configtest.php.

Para que realmente el idioma de la aplicación sea el español, debemos generar el archivo *locale* para la página de códigos ISO-8859-1. Un archivo *locale* indica el juego de caracteres a usar para un país o región. Por defecto Ubuntu usa UTF-8 (ver variable de entorno LANG en /etc/default/locale), por lo que deberemos añadir el *locale es_ES* que anteriormente indicamos para Squirrelmail.

```
$ sudo locale-gen es_ES
```

Desde la interfaz web de Squirrelmail se permite su configuración de forma personalizada e incrementar sus prestaciones mediante múltiples *plugins* descargables. (Una lista de ellos agrupados por categorías la podemos encontrar en http://squirrelmail.org/plugins.php).