

Unidad 5



Servicio de correo electrónico y mensajería instantánea

Servicios de
red e internet



Índice



5.1. Protocolo de transferencia de mensajes

- 5.1.1. Funcionamiento del servicio
- 5.1.2. Protocolo SMTP

5.2. Protocolos y servicios de descarga de correo

- 5.2.1. Protocolo POP
- 5.2.2. Protocolo IMAP

5.3. Clientes de correo electrónico

5.4. Cuentas de correo, alias y buzones de usuario

- 5.4.1. Instalación del servicio de envío de mensajes
- 5.4.2. Instalación del servicio de almacenamiento en buzones

5.5. Correo seguro, firma digital y cifrado de mensajes

5.6. Servicio de mensajería instantánea

- 5.6.1. Características del servicio de mensajería instantánea
- 5.6.2. Protocolos
- 5.6.3. Instalación del servicio



Introducción

El correo electrónico es un medio de comunicación escrito que es muy útil como herramienta de envío y recepción de mensajes a cualquier usuario de internet en todo el mundo.

Según la RAE email o correo electrónico puede tener tres significados:

- > Sistema de transmisión de mensajes o archivos de un terminal a otro a través de redes informáticas.
- > Dirección para la recepción de mensajes enviados mediante este sistema.
- > Mensaje enviado mediante este sistema.

Los correos electrónicos tienen modelos de acceso distintos:

- > **Correo POP:** que es gestionado a través de aplicaciones o gestores de correo y se encargan de almacenar los mensajes en el PC por lo que no requiere una constante conexión a internet.
- > **Correo Web o webmail:** este se gestiona a través de un navegador, almacena los mensajes fuera del PC y necesita una conexión constante para su uso.

Al finalizar esta unidad

- + Seremos capaces de describir los diferentes protocolos que intervienen en el envío y recogida del correo electrónico.
- + Sabremos como crear cuentas de usuario y verificar el acceso a ellas.
- + Podremos aplicar métodos para impedir usos indebidos del servidor de correo electrónico.
- + Conoceremos como utilizar la firma digital y el correo cifrado.
- + Sabremos como establecer el servidor de correo como un servicio seguro.
- + Podremos instalar el servicio de mensajería instantánea.
- + Conoceremos como utilizar clientes gráficos y de texto de mensajería instantánea.
- + Seremos capaces de determinar el tipo de listas de distribución y los modos de acceso permitidos.
- + Podremos crear cuentas de usuario y verificar el acceso a los servicios de mensajería instantánea, noticias y listas de distribución.

5.1.

Protocolo de transferencia de mensajes

Hablamos de servicios de correo electrónico para lo que comúnmente conocemos como *email*, que consiste en ciertas herramientas que al unísono permiten el envío y recepción de mensajes a través de la red eficientemente. Con dicho servicio, los usuarios pueden gestionar sus propios correos, para poder manejar ellos su información de forma más organizada y flexible.

5.1.1. Funcionamiento del servicio

En el correo postal que se ha usado durante muchos años, y que sigue en uso, la manera de enviar mensajes necesita de varios factores o agentes que intervengan en el proceso y lo lleven a cabo.

Para el correo electrónico, también necesitamos de agentes que nos garanticen que el servicio va a funcionar de manera correcta, y estos son:

- > **MUA (Mail User Agent):** es el agente que se encarga de redactar, enviar, recibir y organizar correos electrónicos. SE trata al final del *software* cliente que es solicitante del servicio y que se configura de modo que se conecte con el servidor para cumplir con las especificaciones necesarias de nuestro servicio. Los MUA más conocidos son Outlook, Gmail y Thunderbird.
- > **MT (Mail Transfer Agent):** es el agente que se encarga de gestionar los correos electrónicos que el MUA envía y si es necesario, redireccionarlos hacia su correcto destino. Hay veces que es el propio *daemon* del servicio de correo el que está a la escucha de peticiones. Por lo general, las peticiones las recibe el MTA en el puerto 25 con protocolo SMTP. Los más conocidos son *Postfix* y *Microsoft Exchange Server*.
- > **MDA (Mail Delivery Agent):** el elemento que se encarga de la organización de los buzones de almacenamiento de los usuarios y que tiene el papel de estar siempre pendiente de las peticiones de consulta a los buzones mediante los MUA. Por lo general, el MDA se centra en las solicitudes de los puertos 110 si se usa el protocolo POP y 143 si se usa el protocolo IMAP. Uno de los MDA más conocidos es *Dovecot*.

Podemos ver en la siguiente ilustración como funcionaría un servidor de correo electrónico y las relaciones entre elementos:

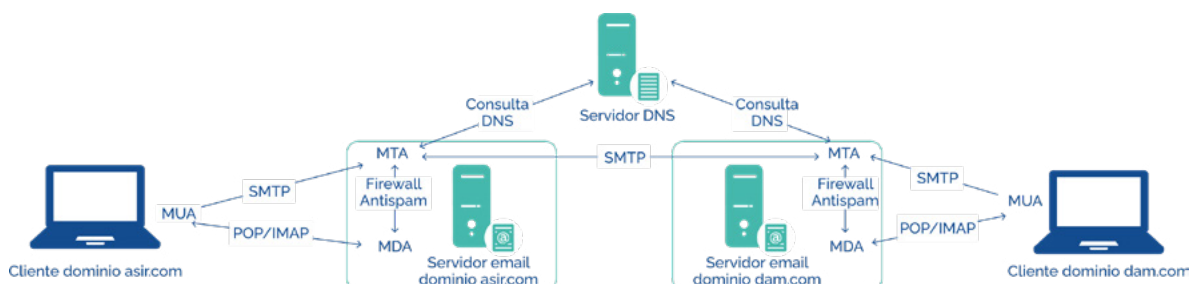


Imagen 1. Funcionamiento del servicio de email

Aquí podemos ver que los clientes se encuentran configurados de modo que trabajen con cada servidor de correo que le corresponde. Entonces, cuando los MUA envían correos, los envía cada uno a su MTA usando el protocolo SMTP. El usuario siempre debe de tener cuenta en el servidor de correo para poder realizar esto, ya demás tener la cuenta correctamente configurada.

En cada uno de los servidores de correo, se administra un nombre de dominio, para que los nombres de cada uno de los usuarios sean del siguiente formato:

`usuario@nombre_dominio`

Cuando enviamos un mensaje a un destinatario alojado en el mismo dominio que el nuestro, el MTA enviará dicho mensaje al MDA para que este último los almacene cuando su sistema de ficheros y poder acceder a él por parte del destinatario cada vez que quiera. Si, por el contrario, el destinatario pertenece a otro dominio, el MTA debe de consultar a los servidores DNS correspondientes para saber cuál es el servidor de correo que administra dicho dominio. Una vez que lo sabe, envía el mensaje al MTA que le corresponde.

Cada vez que los clientes de correo desean consultar su correo usando el MUJA, se hacen peticiones al MDA de su dominio para que use el protocolo POP o IMAP y nos devuelva la información solicitada.

Debe de haber un elemento instalado entre el MTA y el MDA que provea de seguridad al sistema, este suele ser un *firewall*.

5.1.2. Protocolo SMTP

El protocolo SMTP o *Simple Mail Transfer Protocol* es una agrupación de normas que están creadas con la intención de que el envío de los mensajes de correo electrónico sea lo más eficiente posible. El protocolo es totalmente independiente al medio de transmisión, por lo que solo necesitamos que exista una conexión TCP bidireccional con un servidor SMTP para poder enviar mensajes. La norma que lo rige es la RFC 5321.

La principal característica del protocolo SMTP es que se pueden reenviar correos de distintas redes y sistemas de modo que el mismo mensaje pasará por distintos servidores hasta que encuentre su destino. El puerto de escucha suele ser el 25 TCP en el servidor.

El procedimiento mediante el que envía los mensajes es algo similar al de los servicios que hemos visto en unidades anteriores en general, estableciéndose una conversación entre cliente y servidor con distintos comandos y respuestas que dan las ordenes que se deben de seguir. El servidor siempre enviará un código de respuesta a cada comando que el cliente envíe.

La siguiente ilustración identifica como funciona la comunicación entre MUA y MTA, en este caso el servidor MTA:

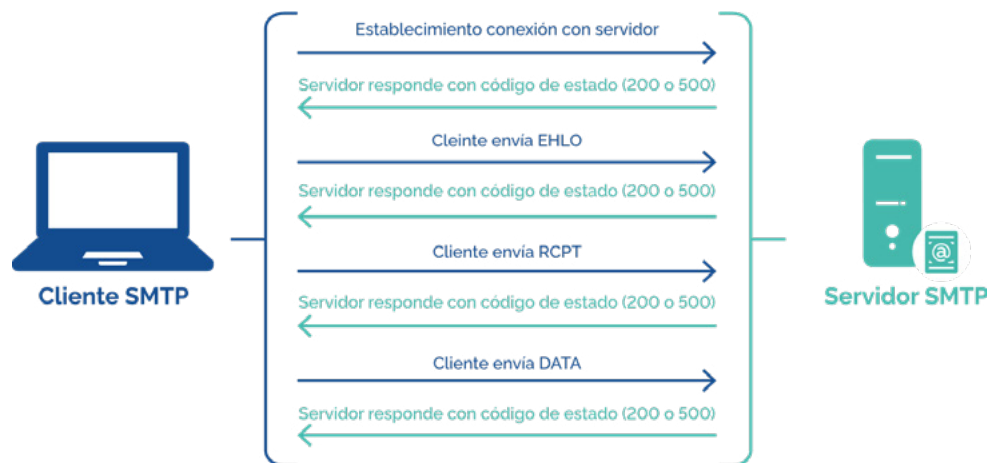


Imagen 2. Funcionamiento SMTP



5.2.

Protocolos y servicios de descarga de correo

Nos referimos a la descarga de correo como la consulta de un cliente a sus mensajes alojados en un servidor. Esto quiere decir que hasta que el MUA no realice la solicitud, el servidor no proporcionará ninguna información.

Tenemos dos protocolos encargados de la descarga de información que se usarán dependiendo del tipo de mensaje que tengamos almacenado en los buzones: POP, *Post Office Protocol* o IMAP, *Internet Message Access Protocol*.

5.2.1. Protocolo POP

El protocolo POP se encuentra actualmente en la versión 3, que es la más actual, lo que hace que también se conozca como *POP3*. Este protocolo se rige por la norma RFC 1939 y se diseña para su uso en equipos con recursos limitados. El protocolo es sencillo, ligero y con pocas funcionalidades que puedan usar clientes y servidores sin gran cantidad de recursos. El puerto que usa POP3 por defecto para la escucha de los mensajes es 110 TCP.

Para que los clientes puedan realizar operaciones usando POP3, el servidor también debe de estar configurado bajo POP3.

Las principales características de este protocolo son las siguientes:

- > Se diseña el protocolo para clientes con poca velocidad de conexión.
- > Existe un modo de operación *offline* que funciona del siguiente modo: se descargan los mensajes del servidor de POP3 de manera local y se borran de este servidor de manera posterior.
- > El contenido del mensaje por lo general irá en texto plano.

Cada vez que un cliente quiera trabajar con servidores POP3, su modo de trabajo básico se resumirá en tres fases:

1. **Establecimiento de la conexión TCP.**
2. **Fase de autorización.** En dicha fase, el servidor informará al cliente de que está disponible para el intercambio de instrucciones. En dicho mensaje se incluye un código de estado que indica que, si ha sido satisfactorio, se implantará uno de los dos siguientes mecanismos encargados de la identificación del usuario:
 - a. **Mediante los comandos USER & PASS.** Dichos comandos se envían al servidor en texto plano, identificando el usuario y la contraseña de quien se quiere *loguear*.
 - b. **Mediante el comando APOP o AUTH.** Es más seguro que el anterior ya que se implementan identificaciones con criptografía y cadenas de identidad.
3. **Fase de transacción.** Si ya hemos establecido una autorización en la anterior fase, el cliente comienza a ejecutar comandos sobre el servidor, como pueden ser borrar u organizar mensajes además de mostrar las estadísticas del buzón.

En fase de autorización y en fase de transacción se puede ejecutar la orden **QUIT**, que finaliza la conexión con el servidor POP3.



Imagen 3. Funcionamiento básico del protocolo POP3



5.2.2. Protocolo IMAP

El protocolo *IMAP*, o en su versión más reciente, *IMAP4* es otro de los protocolos de correo electrónico y se encuentra regulado por la RFC 2060. Se diferencia en POP3 en que IMAP4 permite operaciones más complejas, lo que se traduce en que es el protocolo que se implanta en los sistemas que cuentan con mayores recursos. El puerto por el que escucha es por defecto el 143 TCP. Además de todo esto, sus principales características son:

- > Permite que haya una mejor organización de los mensajes ya que cada buzón lo asigna a distintas carpetas.
- > Se pueden crear, borrar o modificar carpetas de buzones.
- > Permite funcionamiento *online*, es decir, se pueden alojar copias de los mensajes en los servidores y posteriormente consultarlo desde otros equipos.

La conexión IMAP se basa en establecer conexión con el servidor, que este último reconozca al cliente y luego empezar a ejecutar una secuencia de comandos que se encuentran separados por líneas. Cada uno de los distintos comandos tiene un prefijo identificador al que llamamos *etiqueta*. A cada uno de los comandos el cliente devuelve un mensaje de respuesta que precede un asterisco “*”.





5.3.

Cientes de correo electrónico

Como hemos hablado anteriormente, el MUA se trata del cliente de correo que se encarga de dos principales funciones:

- > Solicita al servidor de correo que se envíe el mensaje.
- > Solicita al servidor de correo que se transfieran los buzones de usuario con los mensajes.

Otras funciones que tiene el MUA es organizar mensajes, clasificarlos, borrarlos o modificarlos, agenda de contacto en algunos muy específicos, etc.

Para configurar un servicio de correo en un cliente, mínimo necesitamos los datos siguientes:

- > **Usuario:** nombre que se da de alta en el servicio de correo, siempre va seguido de *@nombre_dominio*.
- > **Clave:** la contraseña que usa el usuario para acceder al servicio.
- > **Nombre del servidor SMTP:** el FQDN completo del servidor SMTP, para Outlook, por ejemplo, *smtp.outlook.com*.
- > **Nombre del servidor POP o IMAP:** el FQDN completo del servidor de POP o de IMAP. Siguiendo el ejemplo anterior: *pop.outlook.com* o *imap.outlook.com*.

Tenemos dos tipos de clientes de correo principalmente:

- > **Aplicaciones instaladas en el dispositivo.** Son los clientes que se instalan en el sistema operativo de un equipo.
- > **Webmail.** Es el caso de cuando accedemos al cliente de correo mediante un navegador web. Normalmente, consultamos a la URL del servicio en el navegador. Es importante que se especifique en el servicio el permiso a acceder vía web, ya que suele ser una configuración aparte.

Además de lo que acabamos de nombrar, tenemos utilidades en las líneas de comandos de los sistemas que permiten mandar correos electrónicos.

Cuando instalamos y configuramos y un cliente de correo, es importante que sigamos correctamente las instrucciones que nos dicta el servidor.

En las siguientes imágenes vemos por ejemplo como *Thunderbird* sobre Ubuntu nos da la oportunidad de iniciar sesión con un usuario ya existente o de crear una nueva cuenta de correo.

Imagen 4. Iniciar sesión con una cuenta existente

Imagen 5. Crear nueva cuenta de correo



5.4.

Cuentas de correo, alias y buzones de usuario

Vamos ahora a explicar cómo funciona *Postfix* que es un MTA que funciona sobre Linux y se encarga de gestionar cuentas, alias y buzones con una administración fácil, segura y rápida.

Para la administración de los buzones, instalaremos *Dovecot*, que es un gestor de almacenamiento de buzones que funciona con POP e IMAP.

Hay que indicar que todo se va a realizar sobre Ubuntu Server 21.10.

5.4.1. Instalación del servicio de envío de mensajes

Vamos a ver los pasos que debemos de seguir para instalar el servicio de *Postfix*:

1. Entramos en la terminal como superusuario.
2. Actualizamos los repositorios y los paquetes de instalación.
3. Instalamos *Postfix*:

```
apt install postfix
```

Mientras se instala el servicio se nos preguntará elegir alguna de las opciones de configuración del servicio.

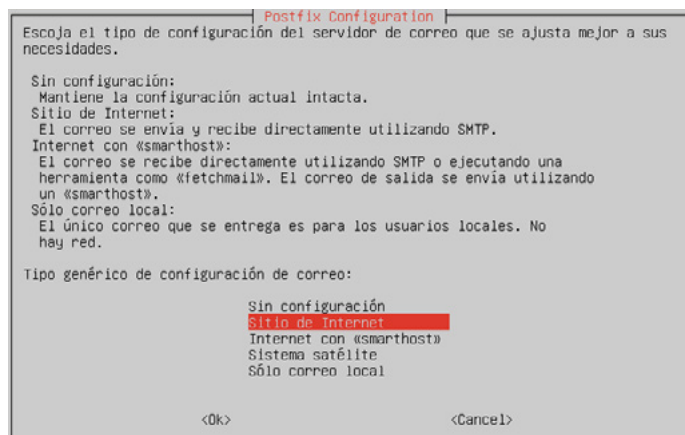


Imagen 6. Configuración de Postfix 1

Los distintos valores de configuración que se ofrecen son los siguientes:

- » **Sin configuración.** Esta opción indica que el fichero de configuración se va a quedar en blanco.
- » **Sitio de internet.** Si se configura esta opción es porque el servidor se encuentra conectado a internet de manera constante y entrega correos a los demás SMTP de manera directa.
- » **Internet con smarthost.** No se envían directamente los correos a los SMTP destino, se pasa por otro SMTP al que llamamos *smarthost*.

- » **Sistema satélite (smarthost).** El equipo únicamente tiene la función de reenviar los correos a distintos MTA:
- » **Solo correo local.** Solo se pueden gestionar correos de forma local en la red.

Realmente, estas opciones lo que realizan es rellenar de un modo u otro el fichero `/etc/postfix/main.cf` para dotar de dicha configuración al servidor. Este archivo sigue la estructura típica, *directiva=valor*.

Por eso, si elegimos 'Sin configuración', dicho archivo estará con una configuración vacía.

Si queremos modificar esta configuración, bien porque nos equivoquemos o porque el sistema cambie, podríamos o bien editar nosotros a manos el fichero o de otro modo, que es lanzando el siguiente comando:

dpkg-reconfigure postfix

Vemos que nos pregunta por seleccionar otro modo de configuración y si elegimos *Sitio de internet*, por ejemplo, podemos ver que nos preguntan diversas opciones, entre las que destacan las siguientes:

- > Dominios que deben considerar nuestra máquina como el destino principal:

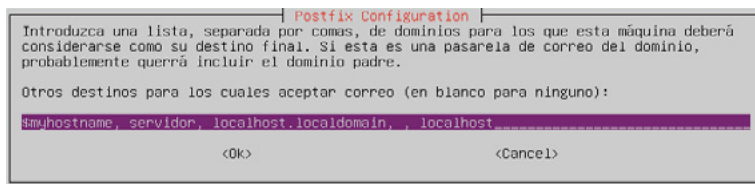


Imagen 7. Configuración de Postfix 2

- > Bloques de redes a las que se pueden reenviar correos;

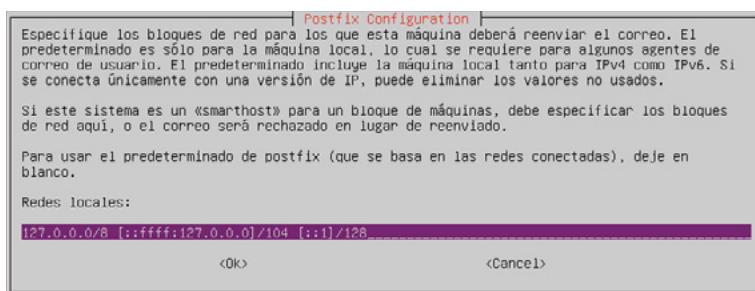


Imagen 8. Configuración de Postfix 3

- > Limite en bytes del tamaño de los buzones:

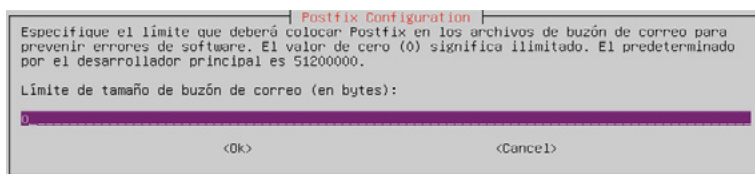


Imagen 9. Configuración de Postfix 4



- > Protocolos IP que permiten escuchar en el proceso de recepción de correos:

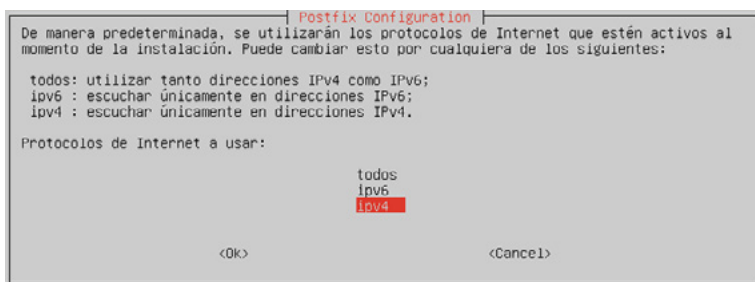


Imagen 10. Configuración de Postfix 5

- > Una vez configurado (las que no hemos incluido es porque no son importantes), hará el propio sistema todos los cambios pertinentes en el fichero de configuración.

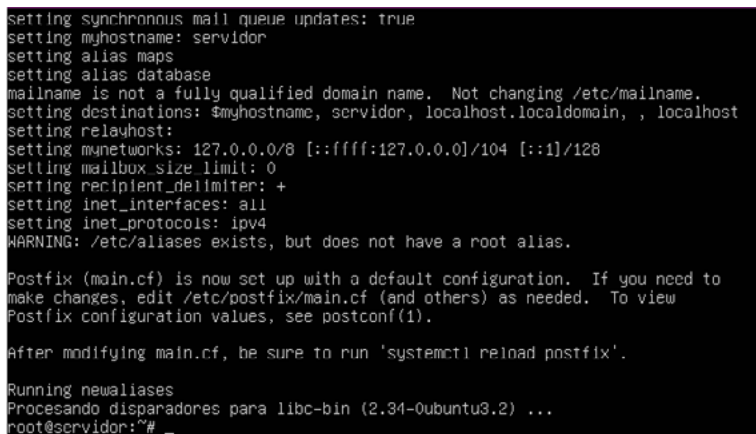


Imagen 11. Configuración de Postfix 6

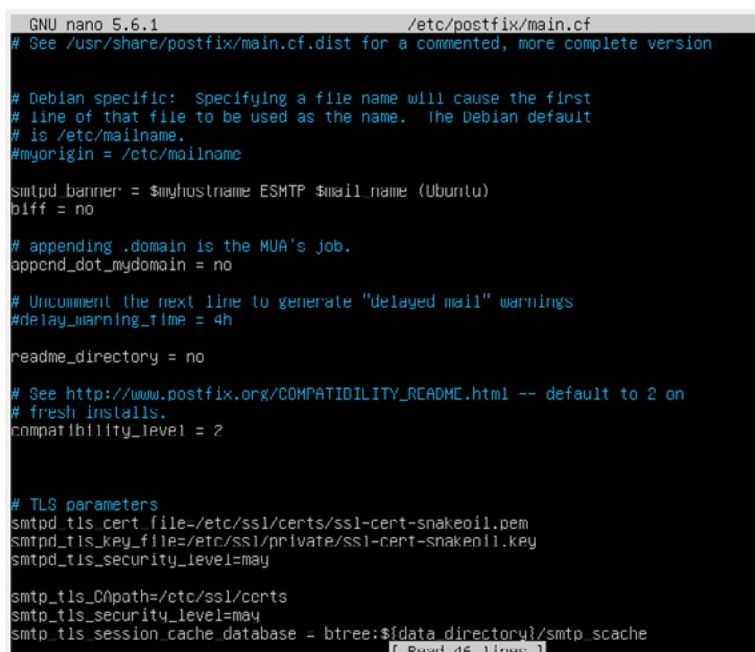


Imagen 12. Fichero main.cf



Además, si comprobamos el estado podemos ver que se encuentra activo:

```
root@servidor:~# systemctl status postfix
• postfix.service - Postfix Mail Transport Agent
  Loaded: loaded (/lib/systemd/system/postfix.service; enabled; vendor preset: enabled)
  Active: active (exited) since Thu 2022-07-14 11:36:12 UTC; 22s ago
  Process: 5222 ExecStart=/bin/true (code=exited, status=0/SUCCESS)
  Main PID: 5222 (code=exited, status=0/SUCCESS)
  CPU: 813us

jul 14 11:36:12 servidor systemd[1]: Starting Postfix Mail Transport Agent...
jul 14 11:36:12 servidor systemd[1]: Finished Postfix Mail Transport Agent.
root@servidor:~#
```

Imagen 13. Estado del servicio: activo

5.4.2. Instalación del servicio de almacenamiento en buzones

Vamos a ver ahora lo que sería una instalación completa de un servicio de almacenamiento de buzones. Hay que tener en cuenta, que los usuarios con los que se trabaja en un servicio de correo, en el caso de Ubuntu, son los usuarios locales del sistema.

Tenemos dos métodos generales de almacenamiento de buzones:

- > **Mbox:** el método por defecto de *Dovecot*. El fichero generado es por buzón de usuario, lo que hace que sea un servicio fácilmente administrable, pero más inconsistente. Suele usarse en sistemas como pocos usuarios y poco tráfico de mensajería.
- > **Maildir:** el fichero se genera por cada correo de usuario. El acceso a buzones y mensajes es bastante más rápido. Más eficiente en seguridad y suele ser usado cuando ya hay un volumen considerable de mensajes y usuarios, eso sí, su administración es más compleja que la anterior.

Vamos a ver ahora los pasos que debemos de seguir a modo de ejemplo para la instalación y configuración de un servidor de correo siguiendo los siguientes pasos:

1. Instalación de *Postfix* para gestionar los correos internos.
2. Instalación de *Dovecot* para consultar los buzones.
3. Modificación de *Postfix* para enviar correo al exterior.

En nuestro ejemplo, vamos a instalar *Dovecot* y *Postfix* sobre Ubuntu Server para crear un servicio de correo electrónico de nuestra empresa *Universae* que además podrá mandar correos al exterior usando los servicios de Gmail.



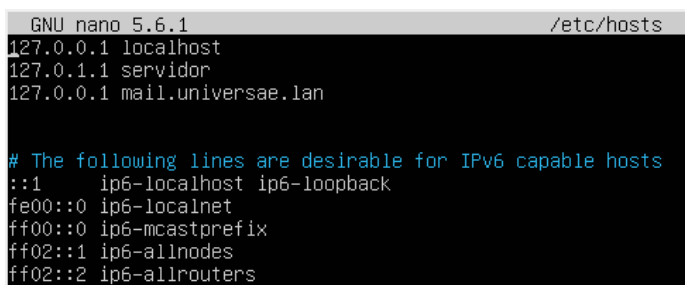
Pasos previos

Lo primero que debemos de hacer, es configurar el DNS de la empresa de modo que reconozca el nombre de dominio de nuestro servidor de correo.

En nuestro caso, vamos a añadir al fichero de nombres la dirección de dicho servidor como *localhost*, pero lo ideal sería modificar el DNS.

Editamos el fichero */etc/hosts* del siguiente modo:

```
127.0.0.1      servidor.dominio
```



```
GNU nano 5.6.1 /etc/hosts
127.0.0.1 localhost
127.0.1.1 servidor
127.0.0.1 mail.universae.lan

# The following lines are desirable for IPv6 capable hosts
::1      ip6-localhost ip6-loopback
fe00::0  ip6-localnet
ff00::0  ip6-mcastprefix
ff02::1  ip6-allnodes
ff02::2  ip6-allrouters
```

Imagen 14. Fichero */etc/hosts*

Después de esta acción, reiniciamos el equipo para que los cambios surtan efecto y lo comprobamos, por ejemplo, con un [ping](#).

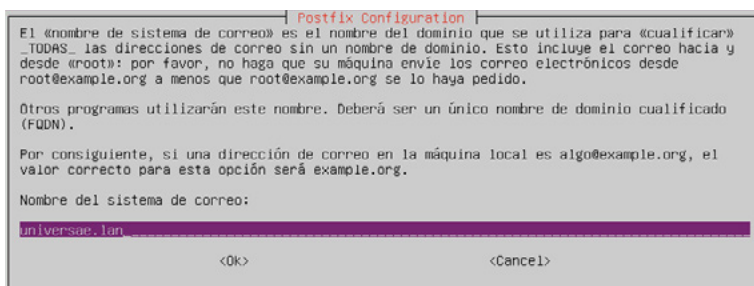
Instalación de Postfix y creación de usuarios

Seguimos los siguientes pasos:

1. Lo primero es instalar *postfix* en nuestro servidor *postfix*:

```
apt install postfix
```

La instalación nos irá solicitando ciertos parámetros que podemos dejar por defecto, haciendo solo caso a la primera pantalla, que nos dice que configuración queremos para *postfix* y la siguiente pantalla, que nos indica cual será el nombre del sistema de correo, en el que tendremos que especificar el dominio que usaremos para nuestro correo electrónico:



```
Postfix Configuration
El «nombre de sistema de correo» es el nombre del dominio que se utiliza para «cualificar»
todas las direcciones de correo sin un nombre de dominio. Esto incluye el correo hacia y
desde «root»: por favor, no haga que su máquina envíe los correo electrónicos desde
root@example.org a menos que root@example.org se lo haya pedido.

Otros programas utilizarán este nombre. Deberá ser un único nombre de dominio cualificado
(FQDN).

Por consiguiente, si una dirección de correo en la máquina local es algo@example.org, el
valor correcto para esta opción será example.org.

Nombre del sistema de correo:
universae.lan
<Ok> <Cancel>
```

Imagen 15. Inserción del nombre del sistema de correo



- Ahora debemos de instalar las utilidades para enviar correos desde terminal:

`apt install mailutils`

```
root@servidor:~# apt install mailutils
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  gsasl-common guile-3.0-libs libgc1 libgsasl7 libidn11 libltd17 libmailutils8 libmysqlclient21
  libntlm0 libpq5 mailutils-common mysql-common
Paquetes sugeridos:
  mailutils-mh mailutils-doc
Se instalarán los siguientes paquetes NUEVOS:
  gsasl-common guile-3.0-libs libgc1 libgsasl7 libidn11 libltd17 libmailutils8 libmysqlclient21
  libntlm0 libpq5 mailutils mailutils-common mysql-common
0 actualizados, 13 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
Se necesita descargar 10,1 MB de archivos.
Se utilizarán 66,9 MB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n]
```

Imagen 16. Instalación de las utilidades del correo

- El siguiente paso es comprobar la versión de *postfix* y ver que sea válida, para esto debe estar por encima de la versión 3.4.10.

`postconf mail_version`

```
profesor@universae:/var/mail$ postconf mail_version
mail_version = 3.5.6
profesor@universae:/var/mail$ _
```

Imagen 17. Versión de Postfix

- Vamos ya con la configuración de *postfix*:
 - Modificamos en el fichero `/etc/postfix/main.cf` las siguientes líneas:

```
inet_interfaces = loopback-only
mydestination = localhost.$mydomain, local-
host, $myhostname
```

Esta parte del fichero queda del siguiente modo:

```
smtpd_relay_restrictions = permit_mynetworks permit_sasl_authenticated defer_unauth_destination
myhostname = universae.ian
alias_maps = hash:/etc/aliases
alias_database = hash:/etc/aliases
myorigin = /etc/mailname
mydestination = localhost.$mydomain, localhost, $myhostname
relayhost =
mynetworks = 127.0.0.0/0 [::ffff:127.0.0.0]/104 [::1]/128
mailbox_size_limit = 0
recipient_delimiter = +
inet_interfaces = loopback-only
inet_protocols = all
```

Imagen 18. Configuración en *main.cf*

- Añadimos a los usuarios que van a usar el correo con el comando `adduser` y respondiendo a las preguntas que el sistema nos propone:



```

root@servidor:~# adduser alumno1
Adding user `alumno1' ...
Adding new group `alumno1' (1002) ...
Adding new user `alumno1' (1002) with group `alumno1' ...
Creating home directory `/home/alumno1' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for alumno1
Enter the new value, or press ENTER for the default
    Full Name []:
    Room Number []:
    Work Phone []:
    Home Phone []:
    Other []:
Is the information correct? [Y/n] y
root@servidor:~# adduser alumno2
Adding user `alumno2' ...
Adding new group `alumno2' (1003) ...
Adding new user `alumno2' (1003) with group `alumno2' ...
Creating home directory `/home/alumno2' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for alumno2
Enter the new value, or press ENTER for the default
    Full Name []:
    Room Number []:
    Work Phone []:
    Home Phone []:
    Other []:
Is the information correct? [Y/n] y
root@servidor:~#

```

Imagen 19. Adición de usuarios locales para los correos

6. Por último, vamos a comprobar el servicio de mensajería de correo con *mailutils*:
 - a. Lanzamos el siguiente comando:


```
echo "Mensaje correo" | mail -s "Asunto del correo" usuario@dominio
```
 - b. Nos vamos a la carpeta */var/mail*.
 - c. Debe de haber un archivo con el nombre del destinatario de correo y solo él o *root* puede abrirlo.
 - d. El contenido nos dirá que el mensaje ha sido recibido de manera correcta.

```

profesor@universae:/var/mail$ echo "Esto es una prueba" | mail -s "Prueba" alumno1@universae.ian
profesor@universae:/var/mail$ su alumno1
Password:
alumno1@universae:/var/mail$ ls
alumno1
alumno1@universae:/var/mail$ cat alumno1
From profesor@universae.ian Fri Jul 15 09:18:42 2022
Return-Path: <profesor@universae.ian>
X-Original-To: alumno1@universae.ian
Delivered-To: alumno1@universae.ian
Received: by universae.ian (Postfix, from userid 1000)
        id D2042378D; Fri, 15 Jul 2022 08:18:42 +0000 (UTC)
Subject: Asunto del correo
To: <alumno1@universae.ian>
X-Mailer: mail (GNU Mailutils 3.11.1)
Message-Id: <20220715081842.D2042378D@universae.ian>
Date: Fri, 15 Jul 2022 08:18:42 +0000 (UTC)
From: Profesor <profesor@universae.ian>

Este es el cuerpo del correo

From profesor@universae.ian Fri Jul 15 10:23:27 2022
Return-Path: <profesor@universae.ian>
X-Original-To: alumno1@universae.ian
Delivered-To: alumno1@universae.ian
Received: by universae.ian (Postfix, from userid 1000)
        id 296CF37C1; Fri, 15 Jul 2022 10:23:27 +0000 (UTC)
Subject: Prueba
To: <alumno1@universae.ian>
X-Mailer: mail (GNU Mailutils 3.11.1)
Message-Id: <20220715102327.296CF37C1@universae.ian>
Date: Fri, 15 Jul 2022 10:23:27 +0000 (UTC)
From: Profesor <profesor@universae.ian>

Esto es una prueba

```

Imagen 20. Prueba de envío de correo y sus comprobaciones



Si el correo no se ha podido reenviar bien, nos saldría el error correspondiente y además se reenviaría de nuevo el correo al usuario que lo ha mandado, por lo que el fichero que aparecería sería con el nombre del emisor del mensaje.

Instalación de Dovecot

Para trabajar con *Dovecot* y los buzones de correo, seguimos los siguientes pasos:

1. Lo primero es instalar *dovecot* y los complementos de los distintos protocolos que puede usar, con el comando:

```
apt install dovecot-core dovecot-imapd dovecot-pop3d
```

```
root@servidor:~# apt install dovecot-core dovecot-imapd dovecot-pop3d
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  libexttextcat-2.0-0 libexttextcat-data
Paquetes sugeridos:
  dovecot-gssapi dovecot-ldap dovecot-imtpd dovecot-lucene dovecot-managesieved dovecot-mysql
  dovecot-pgsql dovecot-sieve dovecot-solr dovecot-sqlite dovecot-submissiond ntp
Se instalarán los siguientes paquetes NUEVOS:
  dovecot-core dovecot-imapd dovecot-pop3d libexttextcat-2.0-0 libexttextcat-data
0 actualizados, 5 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
Se necesita descargar 3.697 kB de archivos.
Se utilizarán 11,9 MB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n]
```

Imagen 21. Instalación de *dovecot* y sus complementos

2. Comprobamos que el servicio funciona consultado su estado:

```
root@servidor:~# systemctl status dovecot
• dovecot.service - Dovecot IMAP/POP3 email server
   Loaded: loaded (/lib/systemd/system/dovecot.service; enabled; vendor preset: enabled)
   Active: active (running) since Thu 2022-07-14 12:32:43 UTC; 2min 38s ago
     Docs: man:dovecot(1)
           http://wiki2.dovecot.org/
   Main PID: 9186 (dovecot)
    Tasks: 4 (limit: 2289)
   Memory: 3.0M
      CPU: 10ms
   CGroup: /system.slice/dovecot.service
           └─9186 /usr/sbin/dovecot -F
             └─9187 dovecot/anvil
               └─9188 dovecot/log
                 └─9192 dovecot/config

Jul 14 12:32:43 servidor systemd[1]: dovecot.service: Deactivated successfully.
Jul 14 12:32:43 servidor systemd[1]: Stopped Dovecot IMAP/POP3 email server.
Jul 14 12:32:43 servidor systemd[1]: Starting Dovecot IMAP/POP3 email server...
Jul 14 12:32:43 servidor dovecot[9186]: master: Dovecot v2.3.13 (09f716dc2) starting up for imap, pop3d
Jul 14 12:32:43 servidor systemd[1]: Started Dovecot IMAP/POP3 email server.
lines 1-20/20 (END)
```

Imagen 22. Comprobación del estado del servicio

3. En el fichero */etc/dovecot/conf.d/10-auth.conf* configuramos las siguientes directivas:

```
disable_plaintext_auth = no
auth_mechanisms = plain login
```

4. Reiniciamos el servicio de *dovecot*.
5. Ahora, en un cliente, como puede ser *Thunderbird*, configuramos alguna cuenta de correo, teniendo en cuenta que podemos usar cualquier protocolo, *POP* o *IMAP*. Las demás opciones, serán servidor y demás, que deberemos de rellenar con los datos que ya conocemos.

```
# See also ssl=required setting.
disable_plaintext_auth = no
```

Imagen 23. Configuración de *dovecot* 1

```
# NOTE: See also disable_plaintext_auth setting.
auth_mechanisms = plain login
```

Imagen 24. Configuración de *dovecot* 2



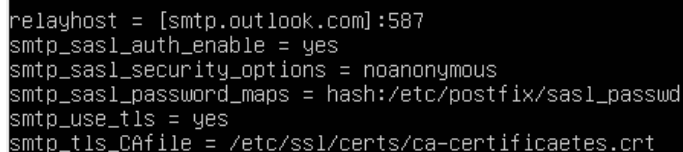
Modificación de *Postfix* para poder enviar correos al exterior

Si queremos poder enviar correos al exterior con el servicio que hemos instalado hasta ahora, debemos de tener en cuenta que estamos trabajando con un dominio interno, es decir, sin acceso al exterior, lo que hace que debamos de realizar una configuración especial.

Para configurar esto con cuentas de *Gmail*, como hemos dicho antes, debemos de seguir los siguientes pasos:

1. Lo primero es editar el fichero `/etc/postfix/main.cf`, en concreto modificando las siguientes directivas:

```
relayhost = [smtp.gmail.com]:587
smtp_sasl_auth_enable = yes
smtp_sasl_security_options = noanonymous
smtp_sasl_password_maps = hash:/etc/postfix/
sasl_passwd
smtp_use_tls = yes
smtp_tls_CAfile = /etc/ssl/certs/ca-certificates.crt
```




```
relayhost = [smtp.outlook.com]:587
smtp_sasl_auth_enable = yes
smtp_sasl_security_options = noanonymous
smtp_sasl_password_maps = hash:/etc/postfix/sasl_passwd
smtp_use_tls = yes
smtp_tls_CAfile = /etc/ssl/certs/ca-certificates.crt
```

Imagen 25. Editando el fichero *main.cf*

2. Ahora, lo que debemos de hacer es crear el archivo `/etc/postfix/sasl_passwd` y añadir la siguiente línea:

```
[smtp.gmail.com]:587    usuario@gmail.
                        com:password
```

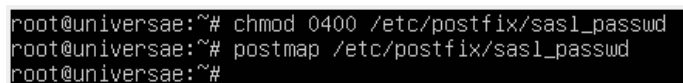


```
GNU nano 5.6.1 /etc/postfix/sasl_passwd *
[smtp.gmail.com]:587 profesor@gmail.com:#Hola123
```

Imagen 26. Fichero *sasl_passwd*

3. Debemos de darle permisos a este archivo y hacerle un `postmap` para que se cree el fichero de *lookup*:

```
chmod 0400 /etc/postfix/sasl_passwd
postmap /etc/postfix/sasl_passwd
```



```
root@universae:~# chmod 0400 /etc/postfix/sasl_passwd
root@universae:~# postmap /etc/postfix/sasl_passwd
root@universae:~#
```

Imagen 27. Comandos finales de modificación

4. Reiniciamos el servicio.

Cuando hemos realizado todas estas acciones, esto ya debe de funcionar, pero va a ser fácil de comprobar porque habrá que hacer ciertas modificaciones en nuestras cuentas de *Gmail* para que permita que se reenvíen los correos. Estas configuraciones serán específicas dependiendo el proveedor de correo que hayamos seleccionado, ya que quedamos supeditados a las configuraciones de sus servidores.

5.5.

Correo seguro, firma digital y cifrado de mensajes

En las empresas, uno de los principales aspectos para tener en cuenta cuando se usa el servicio de correo electrónico es que el envío y la recepción de mensajes. Los servidores de correo se encuentran preparados para conectar mediante TLS, pero tenemos dos elementos principales que nos ayudan a que el nivel de seguridad, privacidad y autenticidad sea mayor:

- > **Firma digital.** Este elemento nos ayuda a identificar al usuario emisor de un mensaje. Cuando mandamos correos electrónicos. Esto se crea mediante el uso de los certificados digitales. Un certificado digital se basa en un fichero que se instala en el equipo que se usa de modo que el usuario queda autenticado como el usuario exclusivo de dicho certificado. El certificado otorga un valioso plus de veracidad a los correos electrónicos.
- > **Cifrado.** Este método se basa en algoritmos de encriptación, que suele ser RSA, para ocultar la información para que en caso de que se alcance la información, no se pueda leer o al menos resulte más complicado.

El siguiente diagrama nos puede dar una descripción de cómo funciona el correo seguro:

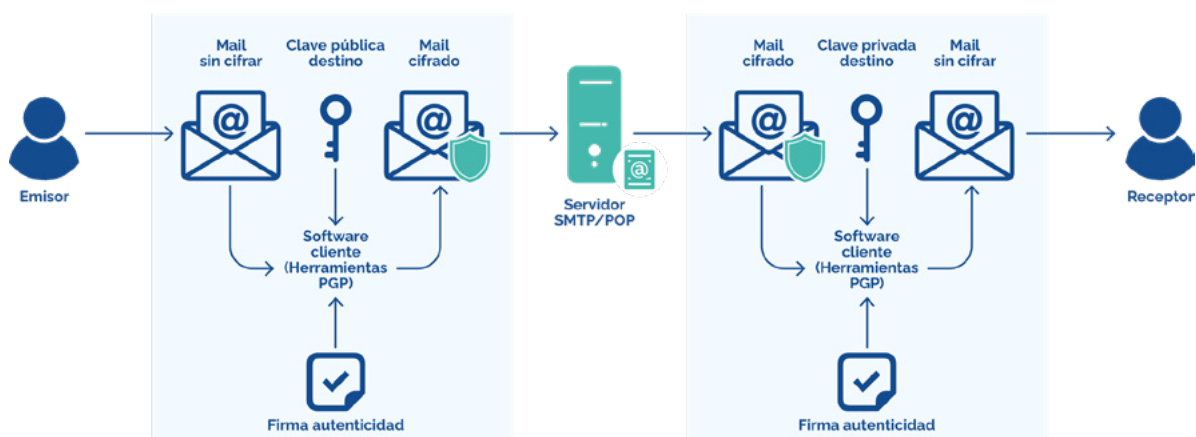


Imagen 28. Funcionamiento del servicio de encriptación de correo

Para poder desarrollar esto, debemos de saber que para *Thunderbird* existen extensiones que nos ayudan a desarrollar los aspectos de seguridad antes nombrados. Por ejemplo, para encriptación de correos tenemos *Kleopatra* y para firmar de forma segura los correos electrónicos podemos usar *Enigmail*. Ambos son *software* perteneciente a la *suite OpenPGP*.



5.6.

Servicio de mensajería instantánea

El servicio de mensajería instantánea se basa en poder enviar mensajes en tiempo real, independientemente del formato de mensaje, siendo principalmente el objetivo el envío de estos a través de la red, aunque también puede ser de manera local. Esto suele ser instantáneo, o al menos casi, siendo necesario que tanto el receptor como el emisor del mensaje se encuentren conectados a aplicaciones cliente.

5.6.1. Características del servicio de mensajería instantánea

Podemos basar el servicio de mensajería instantánea en dos aspectos:

- > **Agenda de contactos:** la agenda de contactos es donde se almacenan todos los usuarios con lo que se pueden intercambiar mensajes. En casi todas las aplicaciones de mensajería se puede ver el estado en el que se encuentra el usuario, si está conectado o no, si está en estado desconocido, en espera, etc. Además, se suelen permitir las opciones sobre la agenda, como modificación, adición y borrado de contactos.
- > **Conversación:** es el entorno en el que se muestran los mensajes que intercambian ambos usuarios. Estas pueden ser privadas o comunitarias, conocidas como grupos donde varios usuarios al mismo tiempo intercambian información.

Además, dentro de los servicios de mensajería existen algunos clientes que permiten complementario al envío de mensajes, mandar imágenes o ficheros. Hemos visto por ejemplo que *Whatsapp* permite también realizar videollamadas individuales o grupales.

5.6.2. Protocolos

El protocolo que más usan los servicios de mensajería instantánea es *XMPP*, pero por lo general, cada aplicación de mensajería instantánea suele implementar el suyo propio con el mismo nombre que la aplicación. Las aplicaciones de IM (acrónimo de mensajería instantánea), son:

- > **Skype:** esta aplicación permite, además de enviar mensajes de texto, poder enviar también audio y vídeo. El protocolo usado tiene el mismo nombre y se encuentra en propiedad de Microsoft, basándose en el protocolo SIP implementado para VoIP. En temas de seguridad, se usa el protocolo AES de 256 bits en su versión gratuita y para la versión de pago se usa un algoritmo RSA de 2048 bits. Su versión más moderna y sustituta ha pasado a llamarse *Microsoft Teams*.
- > **WhatsApp:** usa una versión del protocolo *XMPP*, pero adaptada a sus necesidades. En temas de seguridad, usa un cifrado de extremo a extremo.
- > **Hangouts:** es el sistema de IM desarrollado por Google que funciona con un novedoso protocolo para las comunicaciones. También incorpora las videollamadas entre sus funciones.



Casi todos los protocolos de código abierto que se implementan no son otra cosa que protocolos muy similares a XMPP, pero con ligeras modificaciones. XMPP, *Extensible Messaging and Presence Protocol* se encuentra regido por la RFC 6120, por lo que estos protocolos también.

Los pasos que lleva a cabo el cliente de mensajería instantánea con un protocolo XMPP funcionando (o alguno similar), son los siguientes:

1. Determina la IP y el puerto del servidor al que desea conectarse.
2. Abre una sesión TCP.
3. Abre una sesión para crear un flujo de datos XML sobre TCP.
4. Se negocia la comunicación TLS para que los datos que se envíen estén cifrados.
5. Se autentica mediante SASL.
6. Reserva recursos para intercambiar cadenas XML.
7. Intercambia pequeñas cadenas XML con otros usuarios.
8. Cierra el flujo XML.
9. Cierra la conexión.

Cada vez que se inicie el flujo de datos XML los datos se encuentran formateados usando las etiquetas `<stream></stream>`. Este flujo sirve como contenedor para cadenas XML. En principio, tenemos tres cadenas distintas de XML:

- > **Cadenas de mensaje:** especifican la información que se encuentra formateada mediante las cadenas `<message></message>`. Aquí se encuentra el contenido principal que se quiere enviar.
- > **Cadenas de presencia:** corresponden a los mensajes que se envían en modo *broadcast* indicando únicamente el estado y la información de los distintos usuarios, como información a la red. Se encuentran formateadas en `<presence></presence>`.
- > **Cadenas IQ:** son las cadenas de tipo informativo usadas cuando se requiere alguna acción suplementaria sobre algún dato ya enviado o en proceso de envío. Se encuentran incluidas entre `<iq></iq>`.

El servicio que se va a ver a continuación y uno de los principales en incorporación de XMPP como protocolo de mensajería instantánea es *ejabberd*. Las principales características de este servicio son:

- > Se encuentra descentralizado.
- > Es seguro.
- > Es flexible.
- > Es abierto.
- > Es multiplataforma.



5.6.3. Instalación del servicio

Para instalar el servicio de *ejabberd* en Ubuntu, con los repositorios actualizados debemos de instalar el paquete con el mismo nombre.

Cuando se haya instalado, *ejabberd* cuenta con diversos ficheros dentro del directorio `/etc/ejabberd/`, pero debemos de destacar dos:

- > **ejabberd.yml**: es el fichero de configuración para el servicio.
- > **ejabberdctl.cfg**: es el fichero de configuración para el comando **ejabberdctl**, usado para ejecutar órdenes sobre el servidor de IM. Los comandos más usados son los siguientes:
 - » **ejabberdctl register usuario servidor contraseña**: se usa para añadir un usuario nueva al servicio de mensajería.
 - » **ejabberdctl unregister usuario**: se da de baja un usuario del servicio de IM.
 - » **ejabberdctl registered-users servidor**: nos saca por pantalla una lista de los usuarios que se encuentran dados de alta en el servicio de mensajería.
 - » **ejabberdctl status**: nos muestra el estado actual del servicio.
 - » **ejabberdctl start/stop/restart**: inicia, para o reinicia el servicio de IM.

```
root@universae:~# ls /etc/ejabberd/  
ejabberdctl.cfg ejabberd.pem ejabberd.yml inetrc modules.d  
root@universae:~#
```

Imagen 25. Ficheros de configuración de *ejabberd*





Para configurar el servicio muy básicamente debemos de seguir los siguientes pasos:

1. Lo primero que se debe de hacer es en el fichero `/etc/ejabberd/ejabberd.yml` definir el servidor que va a servir como servidor de mensajería instantánea, añadiendo lo siguiente:

Imagen 28. Configuración de ejabberd 3

```
hosts:
- localhost
- "servidor.dominio.dominio"
```

```
hosts:
- localhost
- "servidor.universae.ian"
```

Imagen 26. Configuración de ejabberd 1

2. Se reinicia el servicio y se comprueba que funciona correctamente.

```
root@universae:~# ejabberdctl restart
root@universae:~# ejabberdctl status
The node ejabberd@localhost is started with status: started
ejabberd 21.12-1 is running in that node
```

Imagen 27. Configuración de ejabberd 2

3. Añadimos los usuarios que queremos que usen el servicio.

```
root@universae:~# ejabberdctl register alumno1 servidor.universae.ian password
User alumno1@servidor.universae.ian successfully registered
root@universae:~# ejabberdctl register alumno2 servidor.universae.ian password
User alumno2@servidor.universae.ian successfully registered
root@universae:~# _
```

4. Comprobamos que los usuarios se encuentran registrados correctamente.

```
root@universae:~# ejabberdctl registered-users servidor.universae.ian
alumno1
alumno2
root@universae:~#
```

Imagen 29. Configuración de ejabberd 4

5. Por último, podríamos usar alguna aplicación cliente para comprobar que todo ha funcionado de manera correcta y sin problemas.



 www.universae.com

