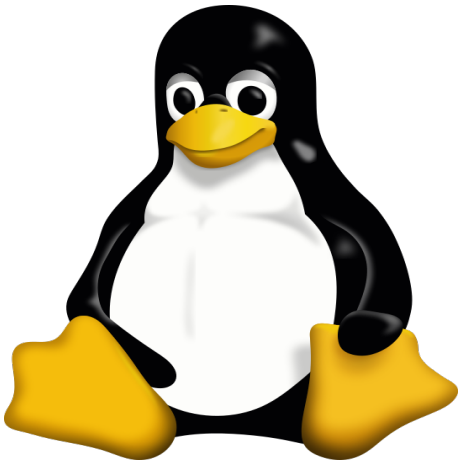


**Bootcamp Especialidad GNU/Linux (2023)**

# Lab 10 - Servidor de Correo

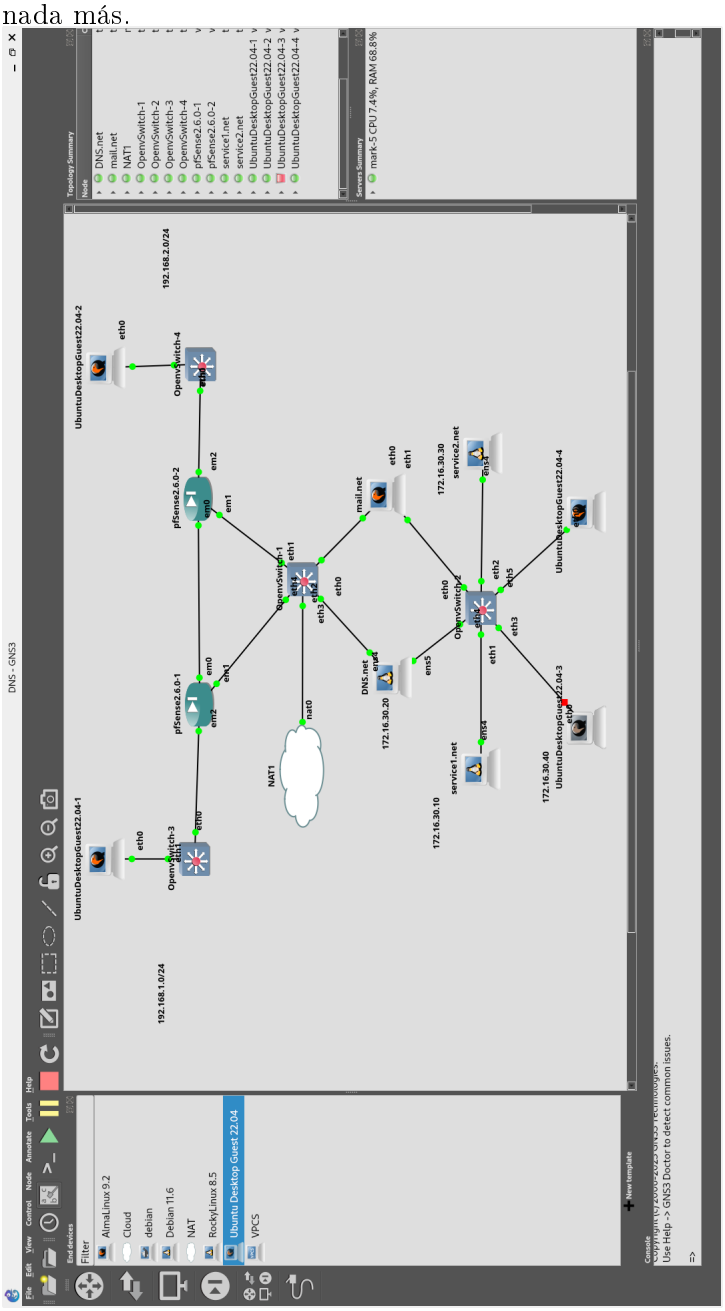
Pedro Antonio Mayorgas Parejo

4 de septiembre de 2023

# Índice

# 1. Topología de red

Se ha seguido una topología de red que nos ha permitido conectar por capa 2 los dispositivos y que estos puedan acceder a servicios de DNS compartidos sin tener que modificar nada más.



## 2. Preparación de DNS

En primer lugar, tenemos que preparar el servicio DNS, para poder utilizar el registro MX utilizado para los servicios de correo asociados. Así como podemos indicar la preferencia de los servidores de correo, porque podemos indicar más de uno. El valor más bajo de preferencia, es el que mayor prioridad queremos que tenga, por convención se usa el 10, 20, 30...

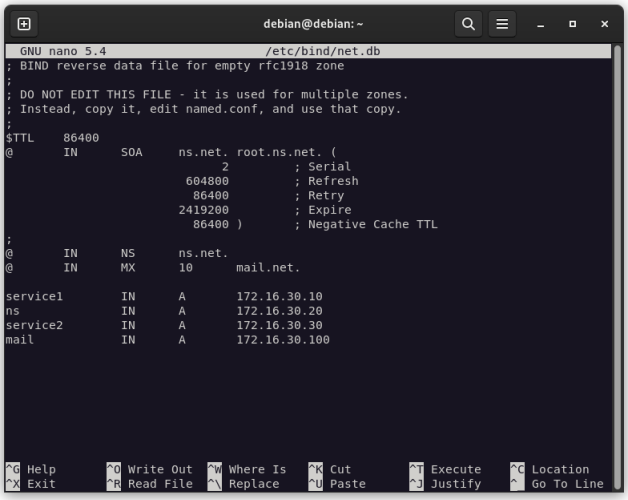



Figura 1: Registros de DNS, de la zona.

Una vez que hayamos terminado de preparar los registros de la zona MX, también tenemos que indicar el registro A, para que la resolución de DNS, tenga una dirección IP a la que acudir. La siguiente captura son las pruebas, donde usamos el comando dig, para obtener un servidor de correo con el registro MX de la zona.

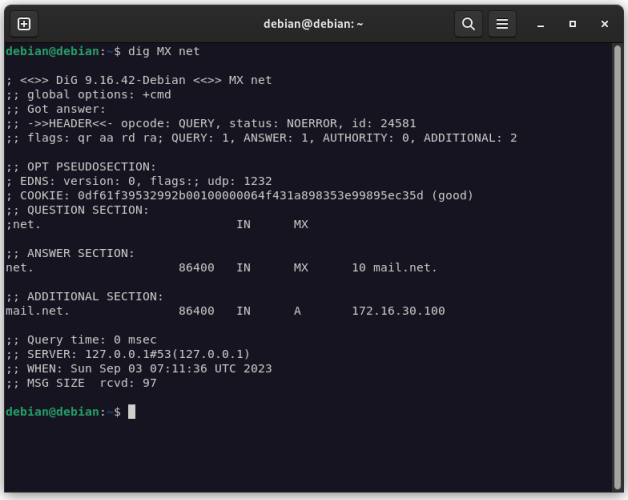



Figura 2: Pruebas del servicio DNS.

Todos los comandos que han sido utilizados son:

```
1 sudo systemctl restart bind9.service
2 sudo systemctl status bind9.service
```

Una vez que se han reiniciado los servicios, debemos también comprobar que se hayan cargado las zonas de DNS correctamente, para ello con el último comando que se ha indicado anteriormente, se debe comproabr que aparezca un **all zones loaded**.

## 3. Instalación de Postfix MTA

Postfix es un MTA Mail Transfer Agent, que te permite el envío de correos electrónicos por el protocolo SMTP (Simple Mail Transfer Protocol). Es un MTA muy robusto que permite una amplia configuración.

Para la instalación de Postfix, hemos utilizado una distribución basada en Red Hat, para poder seguir un manual profesional de configuración del MTA. La mayoría de los cambios se han concrentrado en el fichero **main.cf**, dicho fichero se localiza en **/etc/postfix/-main.cf**. En la documentación de Red Hat se indica que hay que concentrarse en la configuración inicial de los siguientes parámetros del fichero de configuración, para que pueda configurarse de manera básica el MTA.

Comando para instalar Postfix en las distribuciones Red Hat:

```
1 sudo dnf update
2 sudo dnf install postfix
```

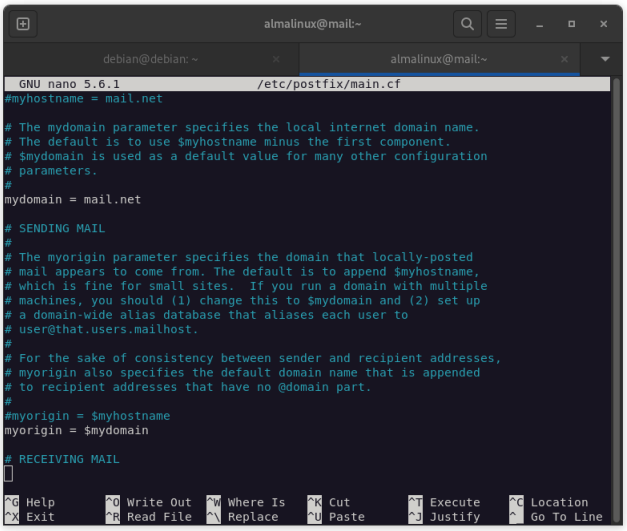



Figura 3: Parámetros de main.cf - Configuración del dominio y origen.

En la captura anterior, se tiene que configurar el dominio local del servicio de correo. Puede ser mx.dominio.tld, o dominio.tld indistintamente. Luego el siguiente parámetro que hemos configurado, es el de origin, que es el parámetro utilizado para indicar el origen de mensajes enviados desde este MTA o sus usuarios.

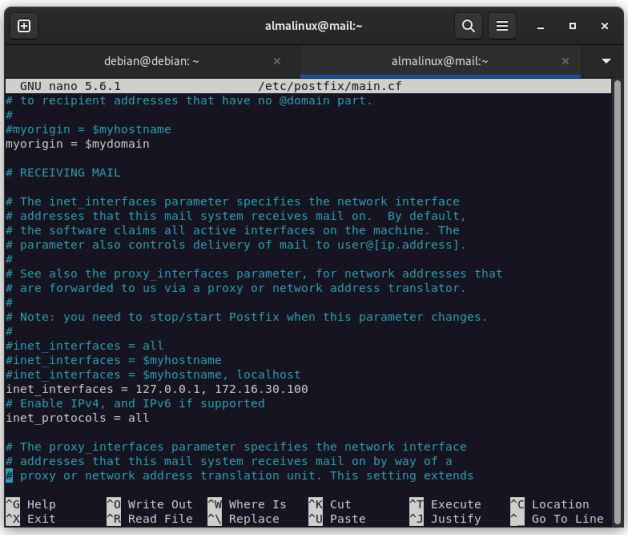



Figura 4: Parámetros de main.cf - Configuración de las interfaces de recepción de email.

Ahora tenemos que configurar las interfaces por las que Postfix recibe los correos, en el manual de administración se puede indicar un parámetro especial **all**, pero esto implica

que se ponga el correo en escucha en todas las interfaces de red indistintamente,lo cual no suele ser bueno. En su lugar, he configurado dos interfaces, en la cual una es la dirección IP externa y otra que debería de estár de manera obligatoria que es la dirección de loopback o localhost.

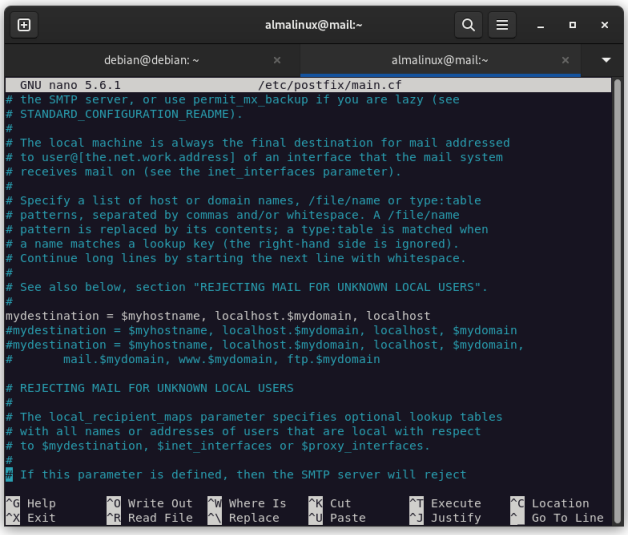



Figura 5: Parámetros de main.cf - Configuración de destinatarios en el MTA.

Luego tenemos que configurar el parámetro de **mydestination**, que consiste en cómo acepta de manera interna el MTA que es un destinatario que se envía de manera local, en vez de hacer reenvío. La configuración indicada en el manual de postfix es el caso 1, donde es el caso por defecto.

Una vez terminada la configuración anterior, debemos ejecutar el comando siguiente para comprobar si la configuración del main.cf de Postfix contiene algún error.

```
sudo postfix check
```

Luego de configurar Postfix, debemos activar y habilitar el servicio con el siguiente comando:

```
sudo systemctl enable --now postfix.service
# Verify
sudo systemctl status postfix
```

Luego tenemos que probar el servicio de correo, para ello instalar la siguiente utilidad que nos permite usar el MTA para enviar correos a los usuarios locales, claro que tenemos que crear previamente al usuario para poder enviarle el mail, así como para poder autenticarnos y leer su correo final en /var/log/maillog o en

```
sudo dnf install sendmail
nano mail.txt
sendmail pedro@mail.net < mail.txt
```

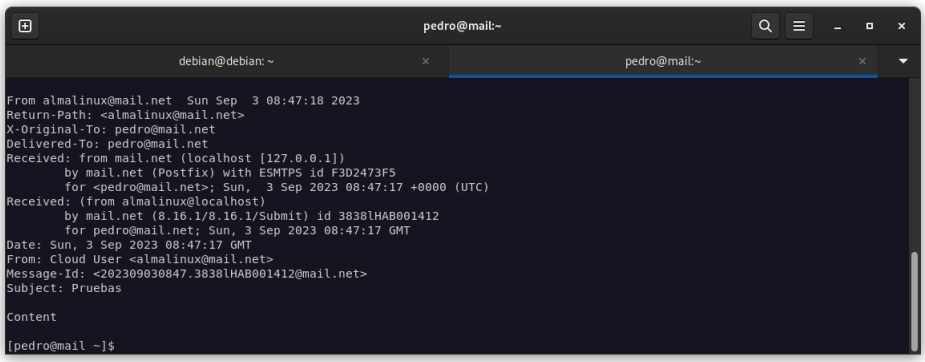



Figura 6: Recepción del correo electrónico.

Para poder ver los correos, vamos a **/var/log/mail/$USER**.

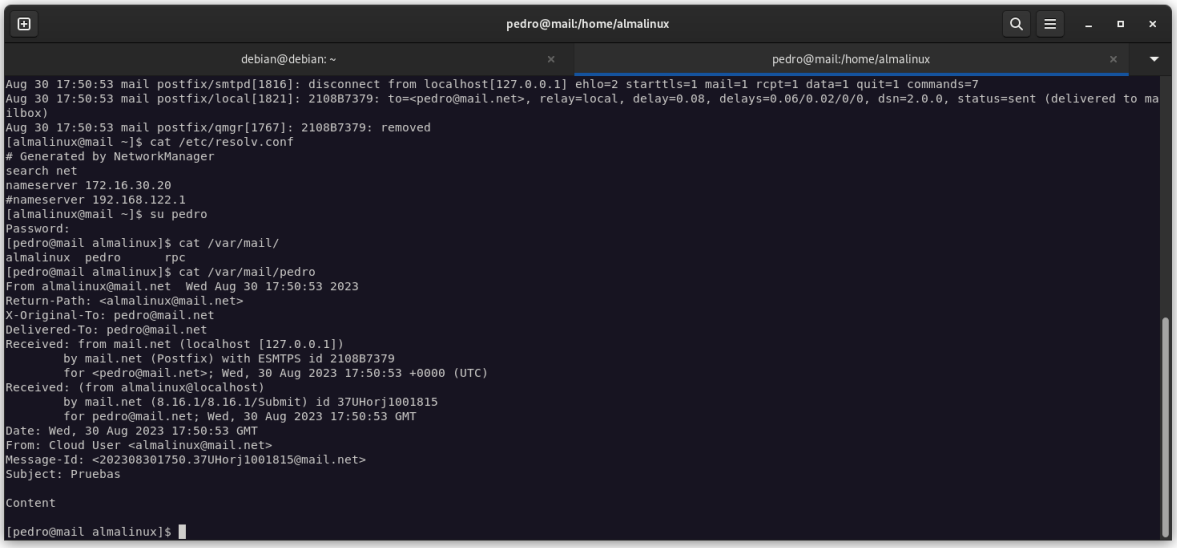



Figura 7: Correos en /var/log/mail/.

Por último tenemos que configurar el **home_mailbox**, para que podamos tenerlo en el $HOME del usuario, sobre todo a la hora de que Dovecot pueda leer los mensajes para utilizarlos con los protocolos clientes de IMAP o POP3.

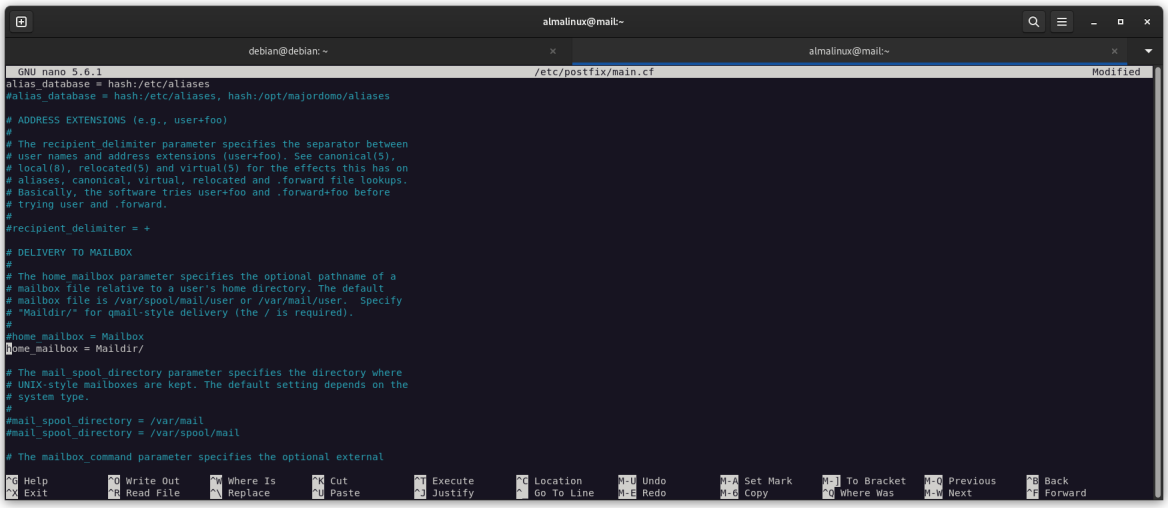



Figura 8: Configuración del home de los correos.

Ahora inicializamos el servicio y comprobamos si hay errores.

```
sudo postfix check
# Habilitamos el arranque del servicio
sudo systemctl enable postfix.service
sudo systemctl start postfix.service
# Comprobamos si hay errores
sudo journalctl -xeu dovecot.service
```

## 3.1. TLS en Postfix

Ahora tenemos que configurar TLS en el Postfix para hacer el envío y recepción seguros con TLS. Primero tenemos que generar una clave privada que nos permita el cifrado asimétrico clásico de RSA que se usa en TLS, luego tenemos que generar a partir de esa clave privada el certificado público de tipo x509 que el servidor provee a los clientes.

```
# PRIVATE KEY GEN
sudo openssl genrsa -out /etc/pki/mail.key.pem 2048
# PUBLIC AND CERT x509 GEN
sudo openssl req -key /etc/pki/mail.key.pem -new -x509 -days 365 -out
    /etc/pki/mail.cert.pem
```

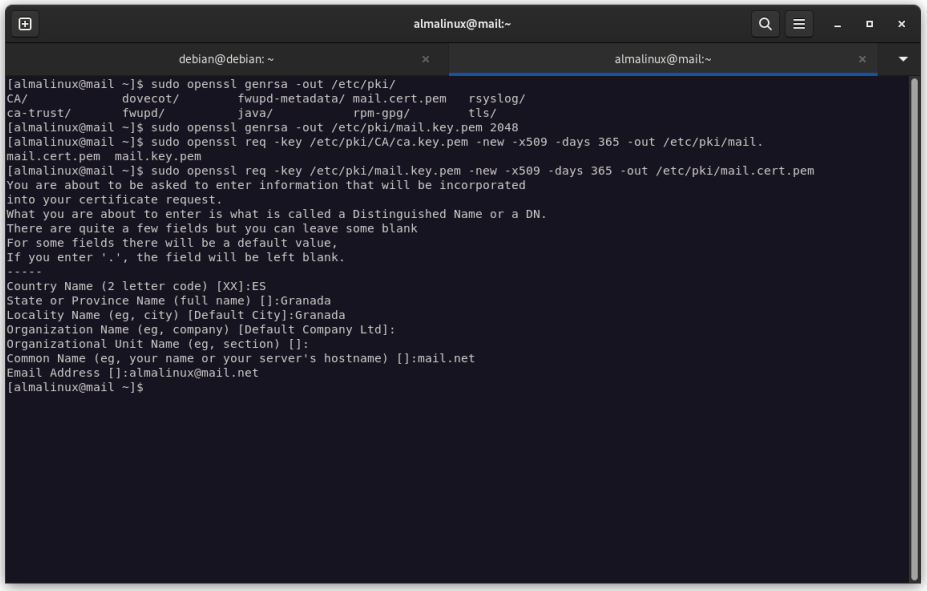



Figura 9: Generación del certificado SSL para mail.net

Ahora tenemos que configurar el servidor Postfix, para que pueda utilizar correctamente el certificaodo, por defecto en las últimas actualizaciones al parecer usa un certificado flojo, para poder ofrecer conexiones un poco más seguras que en texto plano. Para configurar dicho certificado debemos ir a **/etc/postfix/main.cf**. Tenemos que modificar los siguientes parámetros.

**Nota:** Los parámetros los tenemos que indicar dos veces, debido a que uno es para la recepción y otro es para el envío. Entonces si uno de ellos se omite, ocurrirá que se envíen los datos en texto plano.

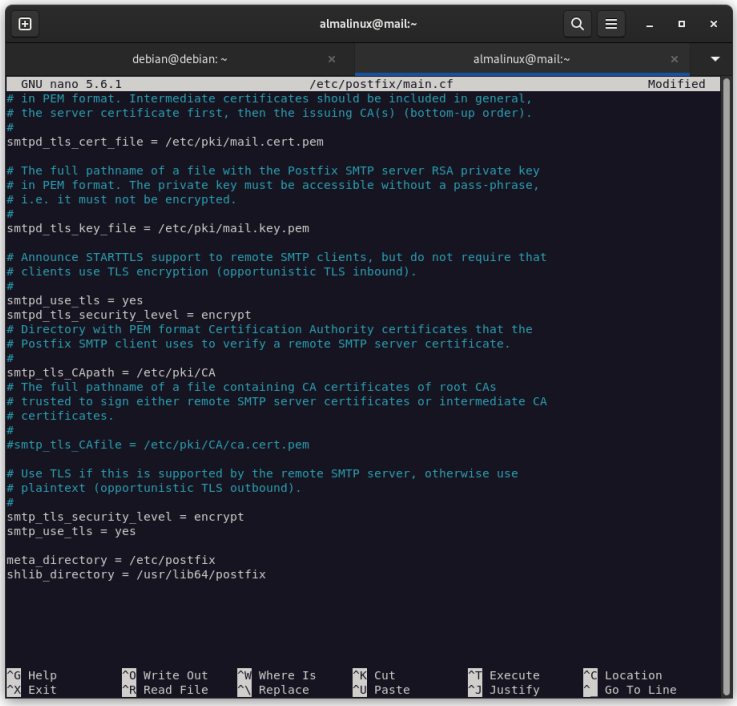



Figura 10: Configuración utilizada para STARTTLS

Ahora inicializamos el servicio y comprobamos si hay errores.

```
# Comprobador de si hay fallos en los ficheros de configuracion
sudo postfix check
sudo systemctl restart postfix.service
# Comprobamos si hay errores adicionales
sudo journalctl -xeu postfix.service
```

Finalmente para probar el STARTTLS, podemos instalar netcat y interaccionar con el servidor para que nos pase los parámetros. Dentro de ellos, debe poner STARTTLS.

```
nc mail.net 25
ehlo test
```

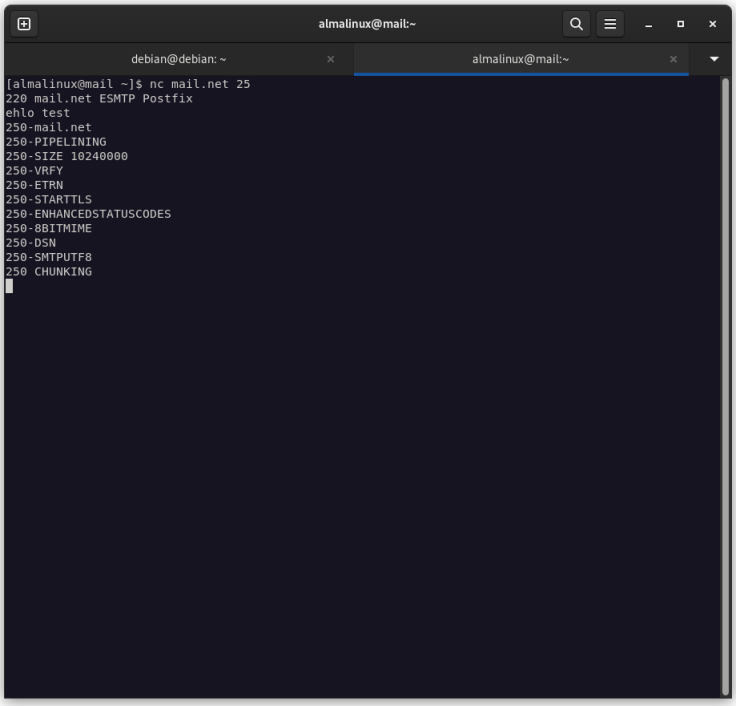



Figura 11: Pruebas de STARTTLS

# 4. Instalación y configuración de Dovecot

Ahora tenemos que realizar la instalación de Dovecot con el siguiente comando:

```
1 sudo dnf install dovecot
```

Los ficheros de configuración de Dovecot, se encuentran en **/etc/dovecot/dovecot.conf**, lo primero que debemos configurar son los protocolos que va a soportar. Así como las direcciones IP que debe escuchar.

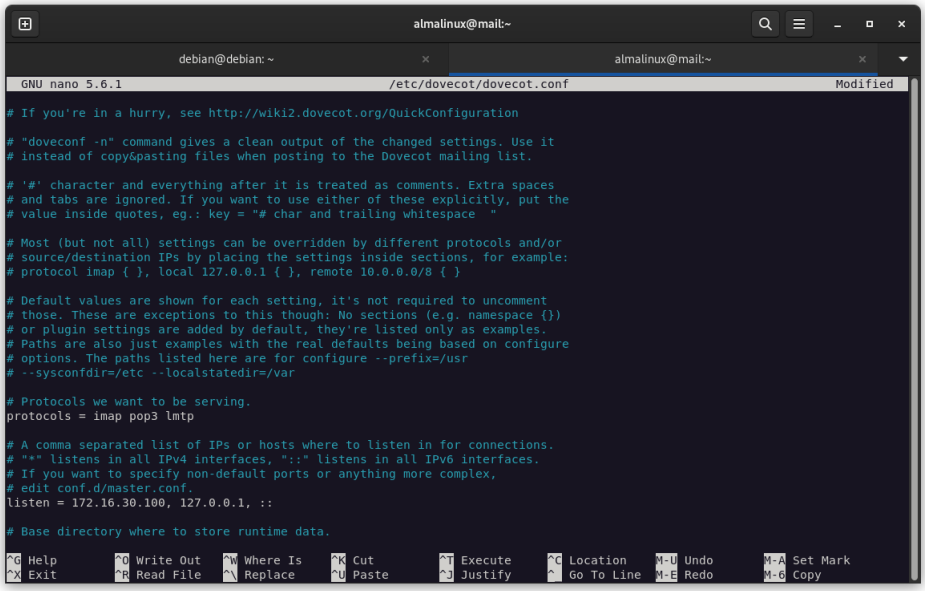



Figura 12: Configuración de protocolos de Dovecot

Luego debemos indicar el mecanismo de autenticación de login adicional al de plain. Todo eso está localizado en **/etc/dovecot/10-auth.conf**.

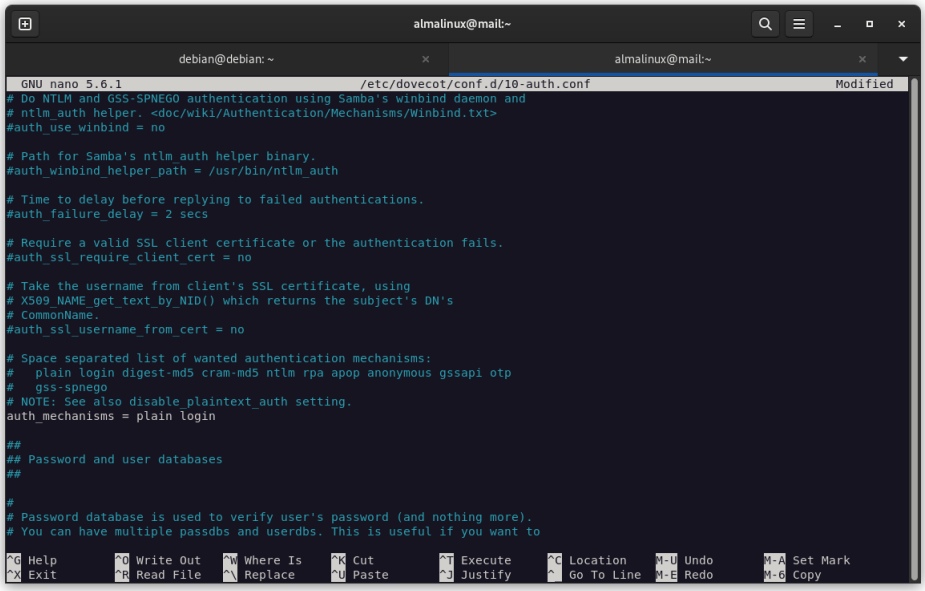



Figura 13: Configuración de autenticación de Dovecot

Luego debemos configurar el directorio de la bandeja de entrada y salida del usuario. Puede estar ubicado en el **$HOME** o en **/var/mail/$USER**.

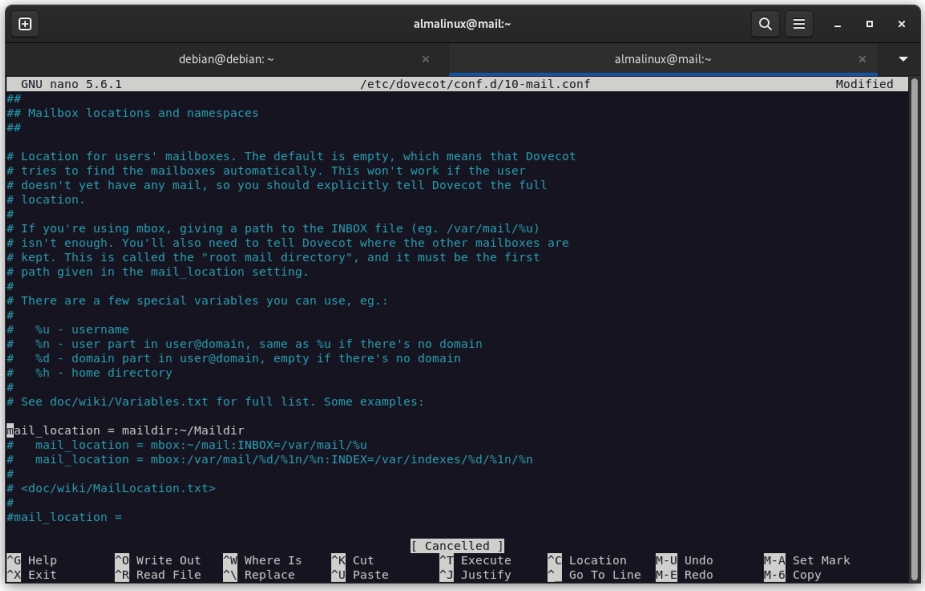



Figura 14: Directorio del mail

Ahora queremos configurar que Postfix, pueda usar Dovecot para la recepción de los

mensajes y que este los cataloge en los subdirectorios correspondientes. Para ello debemos contactarlos con el Socket correspondiente. El protocolo que usan es lmtp entre ellos, se ha configurado anteriormente

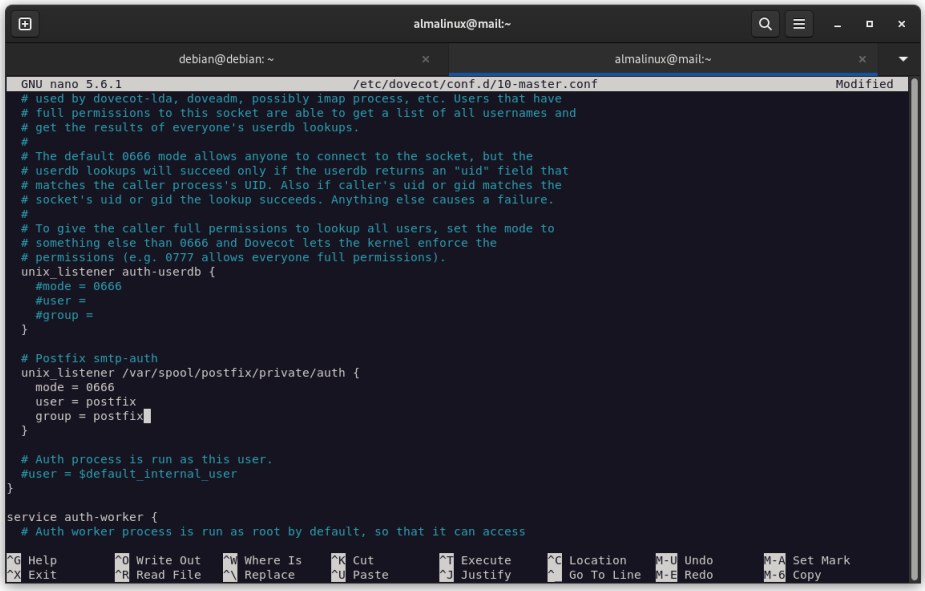



Figura 15: Configuración del Socket

Por último configuramos los certificados compartidos con Postfix, para el uso de conexiones seguras en la ruta **/etc/dovecot/10-ssl.conf**

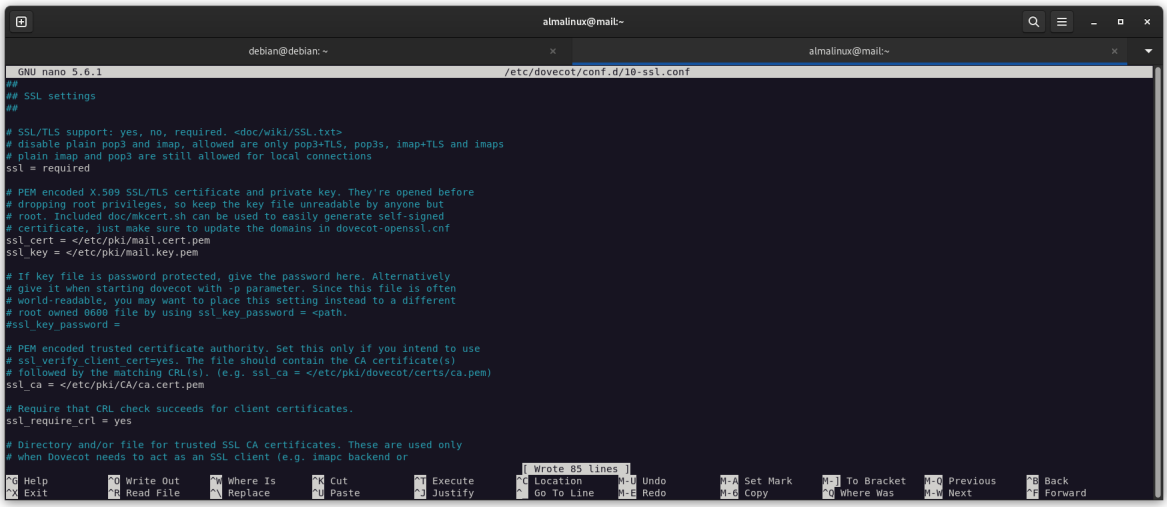



Figura 16: Configuración SSL

Ahora inicializamos el servicio y comprobamos si hay errores.

```
1 sudo systemctl enable dovecot.service
2 sudo systemctl start dovecot.service
3 # Comprobamos si hay errores
4 sudo journalctl -xeu dovecot.service
```

Ahora nos vamos a un cliente de correo (Thunderbird) y usamos IMAP siguiendo los siguientes pasos.

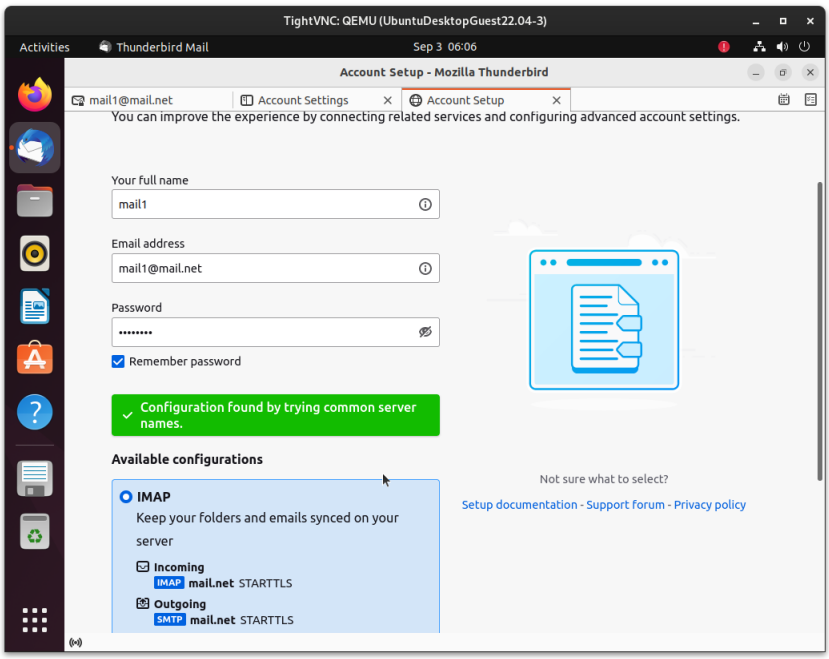



Figura 17: Introduciendo las credenciales

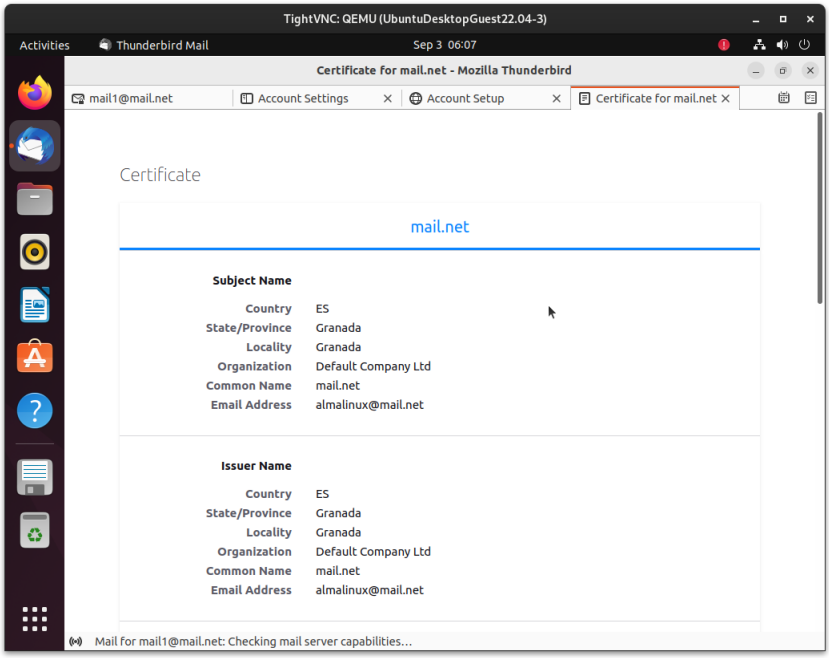



Figura 18: Verificando el certificado.

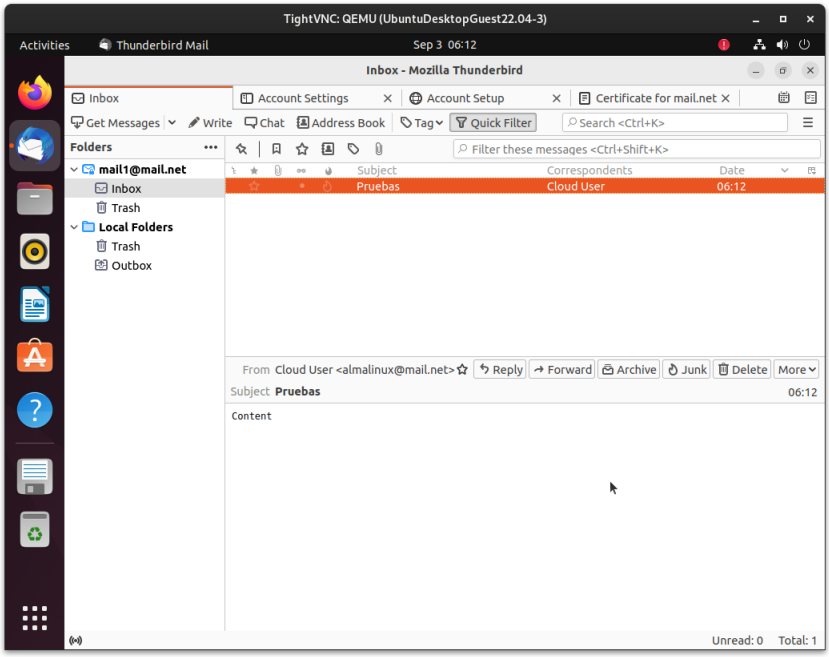



Figura 19: Recepción de mensajes.

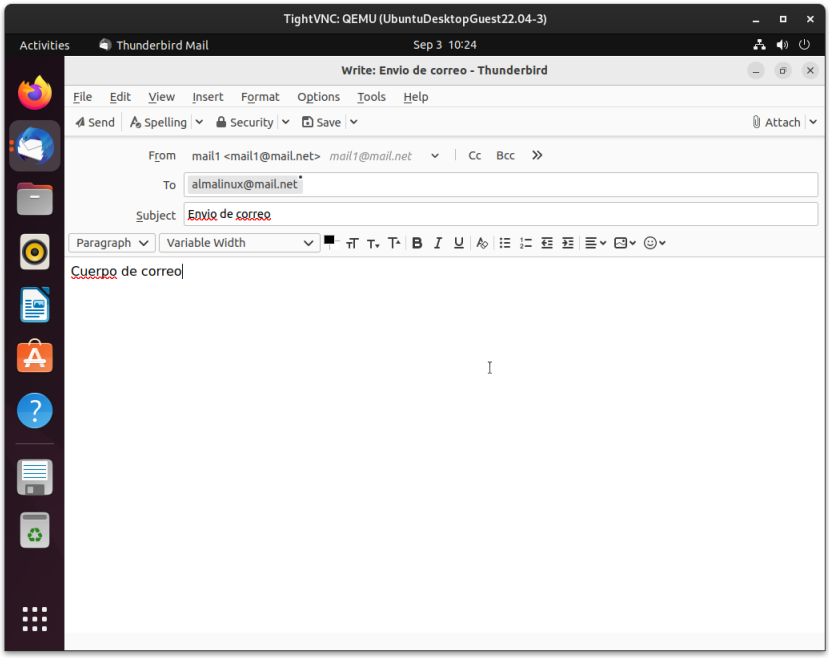



Figura 20: Envío de mensajes.

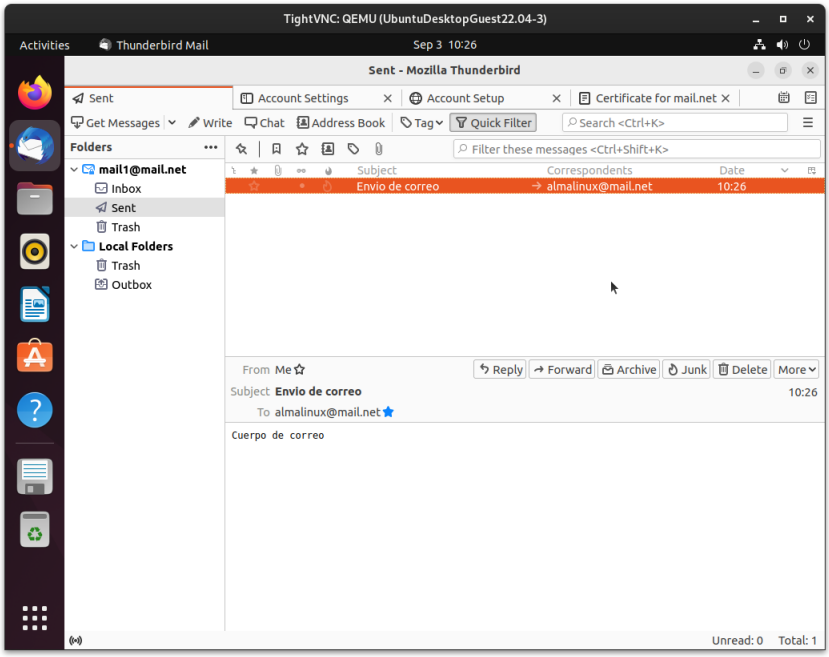



Figura 21: Envío de mensajes - parte 2.

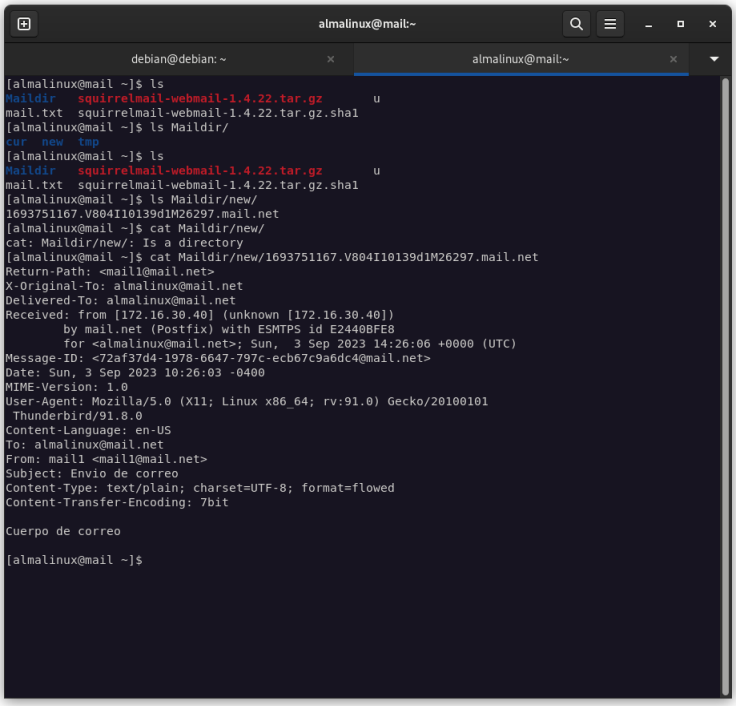



Figura 22: Envío de mensajes - parte 3.

# 5. Instalación de un webmail - squirrelmail

Para la instalación de Squirrelmail hemos utilizado de base Nginx con php-fpm, los paquetes que he utilizado apra prepara el entorno son:

```
1 sudo dnf install nginx php gettext php-mbstring php-xml
```

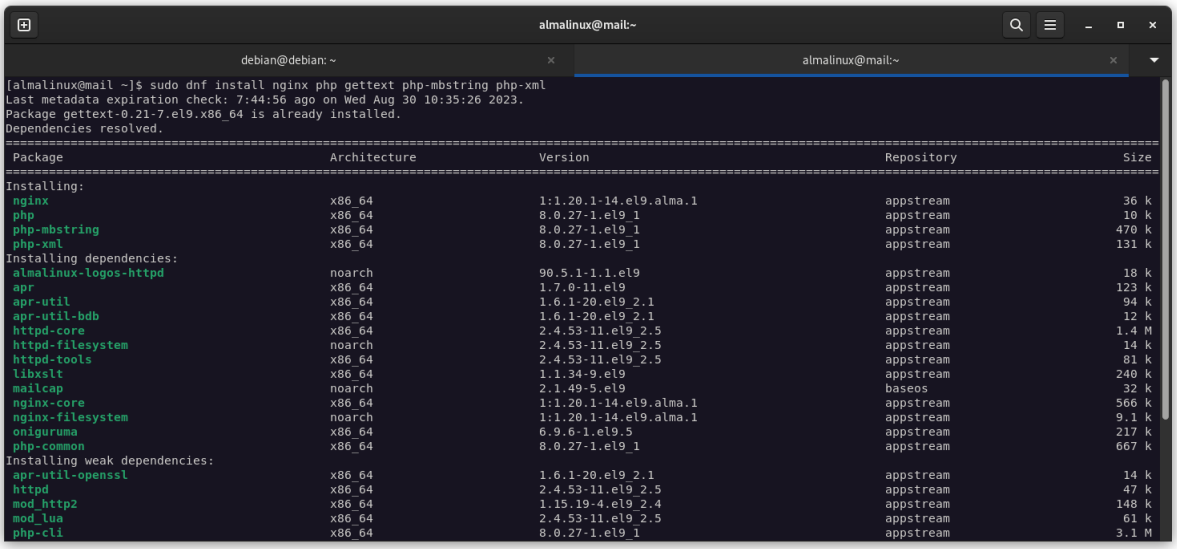



Figura 23: Instalación del servicio web.

Ahora configuramos la ruta de la carpeta junto al nombre de dominio que va a tener el servicio web. Dicho nombre de dominio es el mismo que va a tener el servidor de correo.

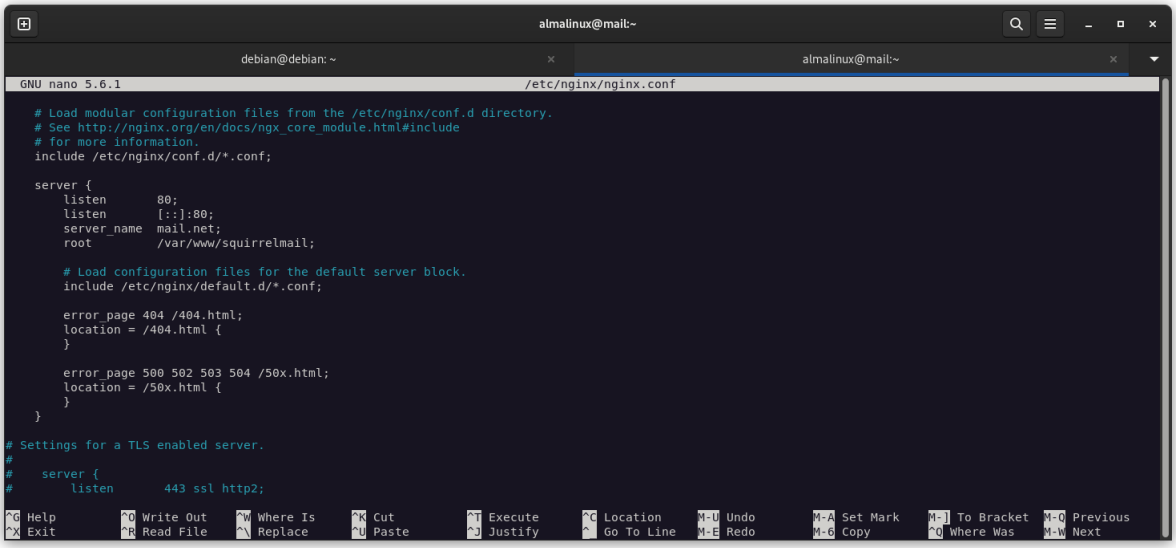



Figura 24: Configuración de NGINX.

Ahora para configurar squirrelmail, aparte de descargarnos los ficheros desde su web, tenemos que instalar perl, porque tiene un script que te ayuda a crear el fichero de configuración que necesita el servicio.

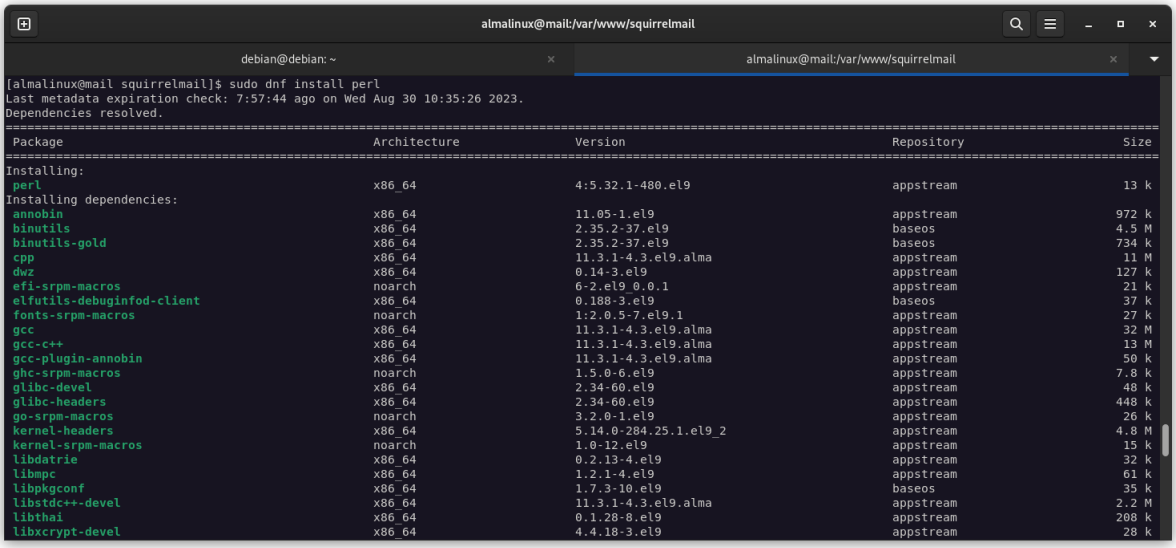



Figura 25: Instalación de perl.

```
1 sudo dnf install perl
2 # Luego de instalar, en /var/www/squirrelmail
3 sudo perl config/conf.pl
```

Ahora lanzamos el menú de configuración.

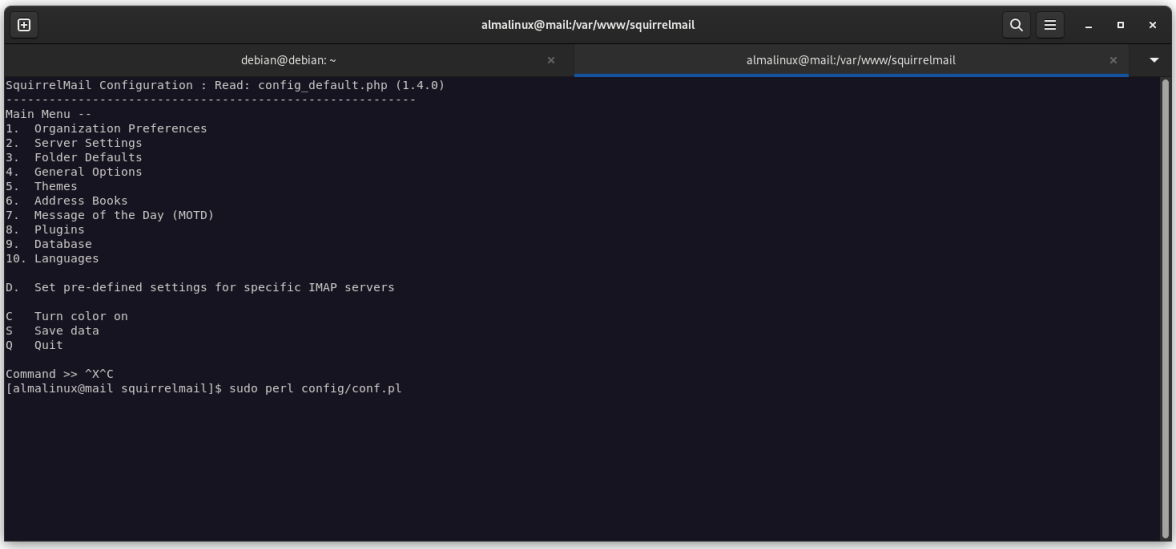



Figura 26: Configurador de squirrelmail.

Nos vamos a la opción 2 para configurar el servidor web. Cambiamos el nombre de dominio y el resto lo dejamos en localhost.

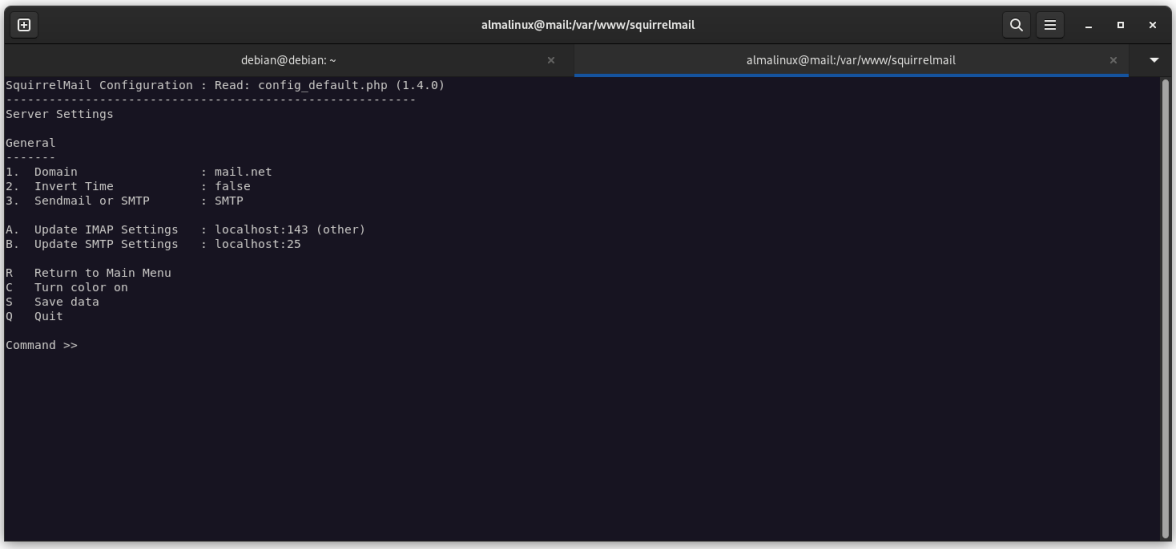



Figura 27: Configurador de squirrelmail - Opción 2.

Finalmente ejecutamos los siguientes comandos al terminar de ejecutar el script configurador, esto es necesario para generar el correcto funcionamiento del servicio web que

necesita unos directorios especiales.

```
# Necesita que las carpetas las tenga el mismo usuario que el servicio web
sudo chown -R nginx:nginx /var/www/squirrelmail/
sudo mkdir -p /var/local/squirrelmail/data
sudo chown -R nginx:nginx /var/local/squirrelmail/
# Esto es neceario para permitir la conectividad con IMAP
sudo setsebool httpd_can_network_connect=1
```

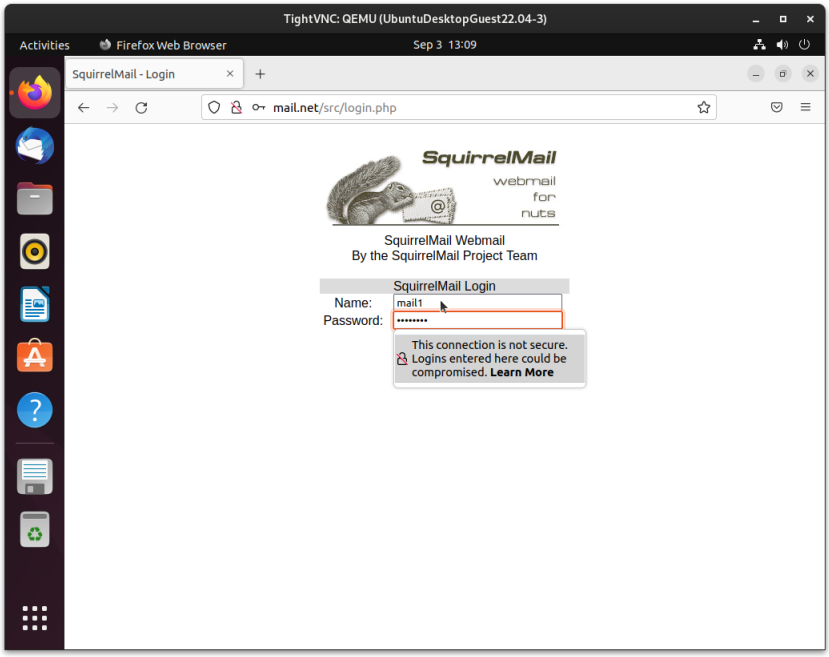



Figura 28: Servicio web activado.

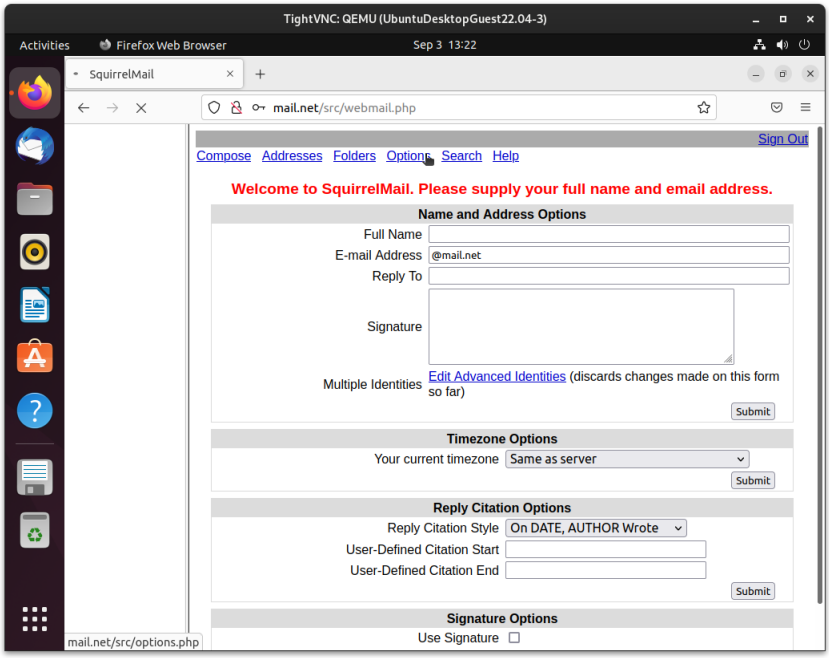



Figura 29: Servicio web activado - Parte 2.

Notas: Debido a la versión de PHP, algunas funciones no funcionan como deben debido a lo obsoleto del software, por lo que se ha modificado el siguiente archivo la parte del if (donde se ha quitado un !), para que pueda funcionar y verse el menú. Pero sigue teniendo un estado erróneo. En concreto no te deja autenticarte como usuario, quedándose en un bucle donde no puedes salir del menú de autenticación.

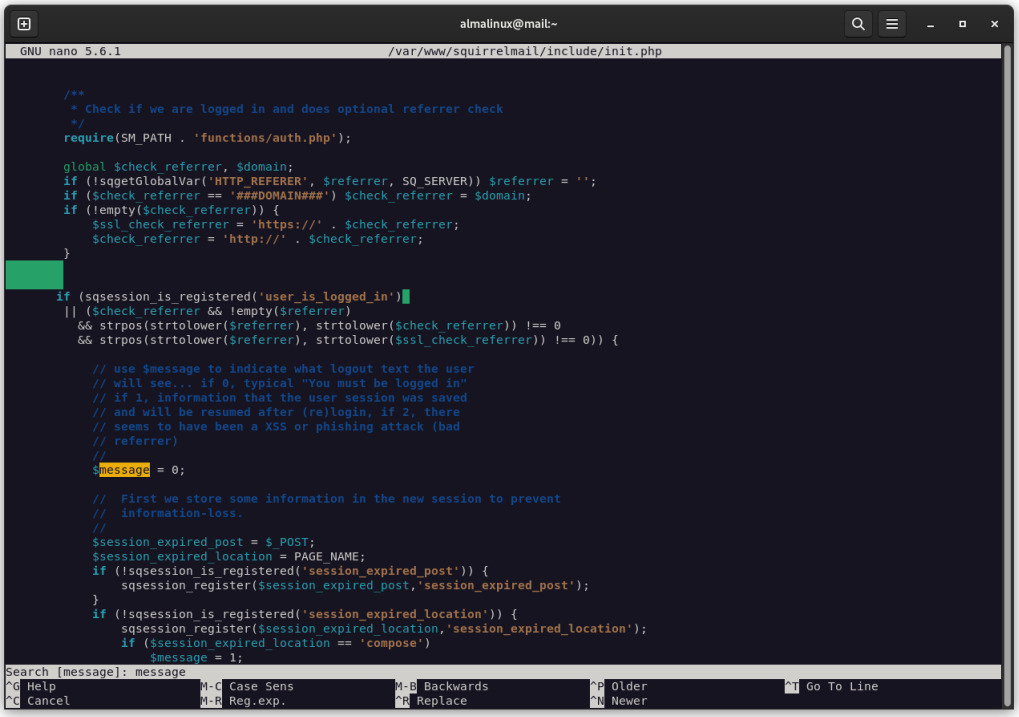



Figura 30: Parte que da error.