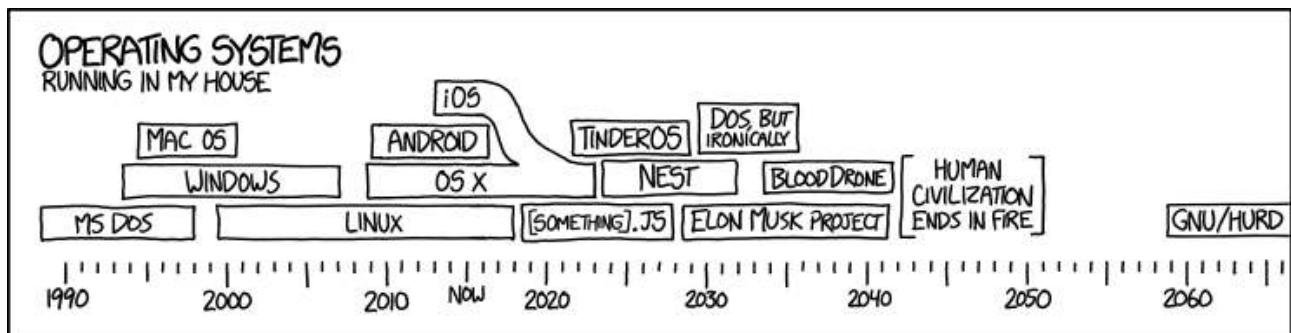


Grado en Ingeniería Informática

# Administración de Sistemas Operativos y Redes de los Computadores

## Práctica 3 *Servicios avanzados*



**Autor**  
Ivan Parkhomchyk Patapchyk

**Curso Académico**  
2025-2026

# Índice

<b>1. Servidor FTP</b>	<b>4</b>
1.1.  Debian GNU/Linux 13 (Trixie) con vsftpd . . . . .	4
1.2.  FreeBSD 14.3 con proftpd . . . . .	7
1.3.  Microsoft Windows Server 2025 con IIS-FTP . . . . .	10
1.3.1. Creación de usuarios . . . . .	10
1.3.2. Estructura de directorios y permisos . . . . .	10
1.3.3. Configuración del servicio . . . . .	11
<b>2. Mensajería instantánea</b>	<b>13</b>
2.1.  Debian GNU/Linux 13 (Trixie) con ejabberd . . . . .	13
2.2.  FreeBSD 14.3 con ejabberd . . . . .	20
2.3.  Microsoft Windows Server 2025 con Openfire . . . . .	26
2.3.1. Descarga e instalación . . . . .	27
2.3.2. Configuración . . . . .	27
2.3.3. Creación de usuarios . . . . .	27
2.3.4. Extracción de <i>logs</i> y configuración . . . . .	28
<b>3. Servidor de trabajo en grupo (<i>groupware</i>)</b>	<b>29</b>
3.1.  Debian GNU/Linux 13 (Trixie) con citadel . . . . .	30
3.1.1. Motivo de la elección de esta alternativa . . . . .	32
3.2.  FreeBSD 14.3 con SOGo . . . . .	34
3.2.1. Prestaciones y características del servicio . . . . .	41
3.3.  Microsoft Windows Server 2025 con SmarterMail . . . . .	44
3.3.1. Características, justificación y registros del servicio propuestos . . . . .	48
<b>4. Proxy Caché (con Squid)</b>	<b>50</b>
4.1.  Debian GNU/Linux 13 (Trixie) . . . . .	50
4.2.  FreeBSD 14.3 . . . . .	52
4.3.  Microsoft Windows Server 2025 . . . . .	55
4.3.1. Instalación y configuración . . . . .	55
4.3.2. Comprobación y ACLs . . . . .	56
<b>5. Rutado, cortafuegos y VPN</b>	<b>57</b>
5.1. Enrutado ( <i>iptables</i> ) en  Debian GNU/Linux 13 (Trixie) . . . . .	58
5.2. Firewall ( <i>pf</i> ) en  FreeBSD 14.3 . . . . .	60
5.3. VPN (PPTP) en  Microsoft Windows Server 2025 . . . . .	62
5.3.1. Instalación de característica opcional y configuración del servicio . . . . .	62
5.3.2. Configuración de los puertos de puertos . . . . .	62
5.3.3. Configuración de direcciones . . . . .	62
5.3.4. Permisos de usuario + reglas de <i>firewall</i> . . . . .	62
5.3.5. Prueba empírica de éxito entre <i>host</i> y <i>guest</i> . . . . .	63
5.3.6. Comandos útiles . . . . .	64
<b>6. Monitorización de servicios</b>	<b>65</b>
6.1.  Debian GNU/Linux 13 (Trixie) con Nagios . . . . .	65
6.2.  FreeBSD 14.3 con Nagios . . . . .	68
6.3.  Microsoft Windows Server 2025 con Windows Admin Central (WAC) . . . . .	72

6.4.  Microsoft Windows Server 2025 con PRTG . . . . .	74
<b>7. RAID 5</b>	<b>76</b>
7.1.  Debian GNU/Linux 13 (Trixie) con <code>mdadm</code> . . . . .	78
7.2.  FreeBSD 14.3 con <code>zpool</code> . . . . .	81
7.3.  Microsoft Windows Server 2025 con <code>diskmgmt.msc</code> . . . . .	84
7.3.1. Comandos útiles (Inicialización y extracción) . . . . .	84
7.3.2. Creación del arreglo . . . . .	84
7.3.3. Simulación de fallo . . . . .	84
7.3.4. Recuperación del fallo . . . . .	85
<b>8. Copias de seguridad</b>	<b>88</b>
8.1.  Debian GNU/Linux 13 (Trixie) con <code>tar</code> y <code>rsync</code> . . . . .	89
8.2.  FreeBSD 14.3 con <code>tar</code> y <code>rsync</code> . . . . .	92
8.3.  Microsoft Windows Server 2025 con <code>tar</code> y <code>rsync</code> . . . . .	95
8.4.  Microsoft Windows Server 2025 con Cobian Backup 11 . . . . .	97
8.4.1. Instalación del servicio . . . . .	97
8.4.2. <i>Backup</i> absoluto . . . . .	97
8.4.3. <i>Backup</i> diferencial . . . . .	98
8.4.4. <i>Backup</i> de sincronización . . . . .	98
<b>9. Correo electrónico</b>	<b>100</b>
9.1.  Debian GNU/Linux 13 (Trixie) . . . . .	100
9.2.  FreeBSD 14.3 . . . . .	104
9.3.  Microsoft Windows Server 2025 <b>SIN HACER</b> . . . . .	114
<b>10. Servidor de terminales de usuario (PXE)</b>	<b>115</b>
10.1.  Debian GNU/Linux 13 (Trixie) . . . . .	116
10.2.  FreeBSD 14.3 . . . . .	121
10.3.  Microsoft Windows Server 2025 <b>SIN HACER</b> . . . . .	125
<b>11. Administración de instalaciones remotas + WoL</b>	<b>126</b>
11.1.  Debian GNU/Linux 13 (Trixie) . . . . .	127
11.2.  FreeBSD 14.3 <b>SIN HACER</b> . . . . .	133
11.3.  Microsoft Windows Server 2025 <b>SIN HACER</b> . . . . .	133
<b>A. Definiciones de siglas y abreviaturas</b>	<b>135</b>
<b>B. Comparación de licencias</b>	<b>135</b>
<b>C. Estructura de los ficheros de entrega</b>	<b>136</b>
<b>D. Esquema de particionado optado</b>	<b>138</b>
<b>E. <i>Scripts</i> de utilidad</b>	<b>139</b>
E.1. Preinicialización con <code>nmcli</code> . . . . .	139
E.2. Tratamiento de paquetes mágicos <code>naponlan</code> . . . . .	140
E.3. Problema al establecer una red <i>Host-Only</i> . . . . .	144
E.4. <code>Makefile</code> empleado para compilar T <sub>E</sub> X . . . . .	144

# Ítems de la práctica

---

# 1. Servidor FTP

«Servidor FTP (*vsftpd*, *proftpd*, *Serv-U*). Será necesario enjaular a un usuario en su directorio de trabajo. Para la demostración se realizará una sesión desde un cliente *ftp* con dos usuarios, uno enjaulado y otro sin enjaular.»

---

El **FTP** (*File Transfer Protocol* o Protocolo de Transferencia de Archivos) es un protocolo de red estándar ubicado en la capa de aplicación del modelo TCP/IP. Su función principal es permitir la transferencia de archivos entre sistemas conectados a una red TCP de manera eficiente.

El servicio se basa en una arquitectura **cliente-servidor**:

- **El Servidor:** Escucha peticiones (típicamente en el puerto 21 para control y 20 para datos), gestiona la autenticación de usuarios y el acceso al sistema de archivos.
- **El Cliente:** Inicia la conexión para subir (*upload*) o descargar (*download*) ficheros.

En cuanto a sus orígenes, el FTP es uno de los protocolos más antiguos de Internet, anterior incluso a la web (HTTP) y al protocolo TCP/IP moderno. Sus orígenes se remontan a **1971**, definido originalmente en el **RFC 114** por Abhay Bhushan del MIT. La especificación actual, que utilizamos hoy en día, está definida en el **RFC 959** de **1985**. A pesar de su antigüedad, sigue siendo un estándar industrial debido a su robustez y soporte universal.

Sobre uso, aunque históricamente se ha hecho servir para la distribución de SW y alojamiento web, en el contexto de la administración de sistemas su uso es crítico para la gestión remota de archivos. Es comúnmente utilizado para la transferencia entre un servidor web y su administración, así como para usos más caseros, como la transferencia de datos entre videoconsolas y el ordenador del usuario doméstico por la intranet.

En el desarrollo de esta práctica, la implementación del servicio FTP tiene un doble objetivo técnico:

1. **Gestión de usuarios y seguridad:** Implementar mecanismos de autenticación y control de acceso.
2. **Aislamiento (Enjaulado/Chroot):** Tal como se especifica en los requisitos de evaluación, se configurará el servidor para demostrar la capacidad de «enjaular» a los usuarios. Esto consiste en restringir la visión del sistema de archivos de un usuario a su propio directorio de trabajo, impidiendo que pueda navegar por la raíz del sistema y comprometer la seguridad del servidor, mientras que otros usuarios privilegiados mantendrán acceso global.

## 1.1. ☑ Debian GNU/Linux 13 (Trixie) con *vsftpd*

```
#!/bin/sh
su -

#-----
# Variables de entorno
#-----
P3AS0RC_SERVICIO=ftp
```

```

P3ASORC_SISTEMA=linux

# NO TOCAR
P3ASORC_MEMORIA=/home/ivan/SUPERMEMORIA/$P3ASORC_SERVICIO
P3ASORC_CONFIG=$P3ASORC_MEMORIA/ficheros_configuracion
P3ASORC_LOG=$P3ASORC_MEMORIA/$P3ASORC_SISTEMA.log
P3ASORC_HISTORIAL=$P3ASORC_MEMORIA/history$P3ASORC_SISTEMA.txt

# ---
P3ASORC_FTP_HOME=/var/practica2/home
P3ASORC_FTP_USER1=usuario_jaula
P3ASORC_FTP_USER2=usuario_libre

P3ASORC_FTP_CONFIG_1=/etc/vsftpd.conf
P3ASORC_FTP_CONFIG_2=/etc/vsftpd.chroot_list
P3ASORC_FTP_LOG=/var/log/vsftpd.log

#-----
# Servicio
#-----
# PASO 1: Usuarios
# Borrar logs anteriores
rm -rf $P3ASORC_FTP_LOG

# Borrar si existen de antes
userdel $P3ASORC_FTP_USER1
userdel $P3ASORC_FTP_USER2
rm -rf $P3ASORC_FTP_HOME

# Crearlos
mkdir -p $P3ASORC_FTP_HOME
useradd -m -d $P3ASORC_FTP_HOME/$P3ASORC_FTP_USER1 -s /bin/bash
$P3ASORC_FTP_USER1
echo -e '1\n1' | passwd $P3ASORC_FTP_USER1
useradd -m -d $P3ASORC_FTP_HOME/$P3ASORC_FTP_USER2 -s /bin/bash
$P3ASORC_FTP_USER2
echo -e '1\n1' | passwd $P3ASORC_FTP_USER2

# PASO 2: Instalar paquetes
apt remove -y vsftpd
apt install -y vsftpd

# PASO 3: Configurar
# Enjaula a todos
cat << EOF > /etc/vsftpd.conf
xferlog_enable=YES
xferlog_file=$P3ASORC_FTP_LOG
log_ftp_protocol=YES
listen=YES
listen_ipv6=NO
anonymous_enable=NO
local_enable=YES
write_enable=YES
dirmessage_enable=YES
use_localtime=YES
xferlog_enable=YES
connect_from_port_20=YES
chroot_local_user=YES

```

```

chroot_list_enable=YES
chroot_list_file=$P3ASORC_FTP_CONFIG_2
allow_writeable_chroot=YES
secure_chroot_dir=/var/run/vsftpd/empty
pam_service_name=vsftpd
rsa_cert_file=/etc/ssl/certs/ssl-cert-snakeoil.pem
rsa_private_key_file=/etc/ssl/private/ssl-cert-snakeoil.key
ssl_enable=NO
EOF

# Menos al libre
echo "$P3ASORC_FTP_USER2" > $P3ASORC_FTP_CONFIG_2

# PASO 4: Aplica cambios
systemctl restart vsftpd

#-----
# Valida servicio
#-----
systemctl status vsftpd
netstat -tunlp | grep -E 'ftp'

#-----
# Extraer logs, configs e historial
#-----
rm -rf $P3ASORC_MEMORIA
mkdir -p $P3ASORC_MEMORIA
mkdir -p $P3ASORC_CONFIG

cp $P3ASORC_FTP_CONFIG_1 $P3ASORC_CONFIG
cp $P3ASORC_FTP_CONFIG_2 $P3ASORC_CONFIG
cp $P3ASORC_FTP_LOG $P3ASORC_LOG
journalctl -u vsftpd --no-pager >> $P3ASORC_LOG

history > $P3ASORC_HISTORIAL
chmod 777 -R $P3ASORC_MEMORIA

tree $P3ASORC_MEMORIA

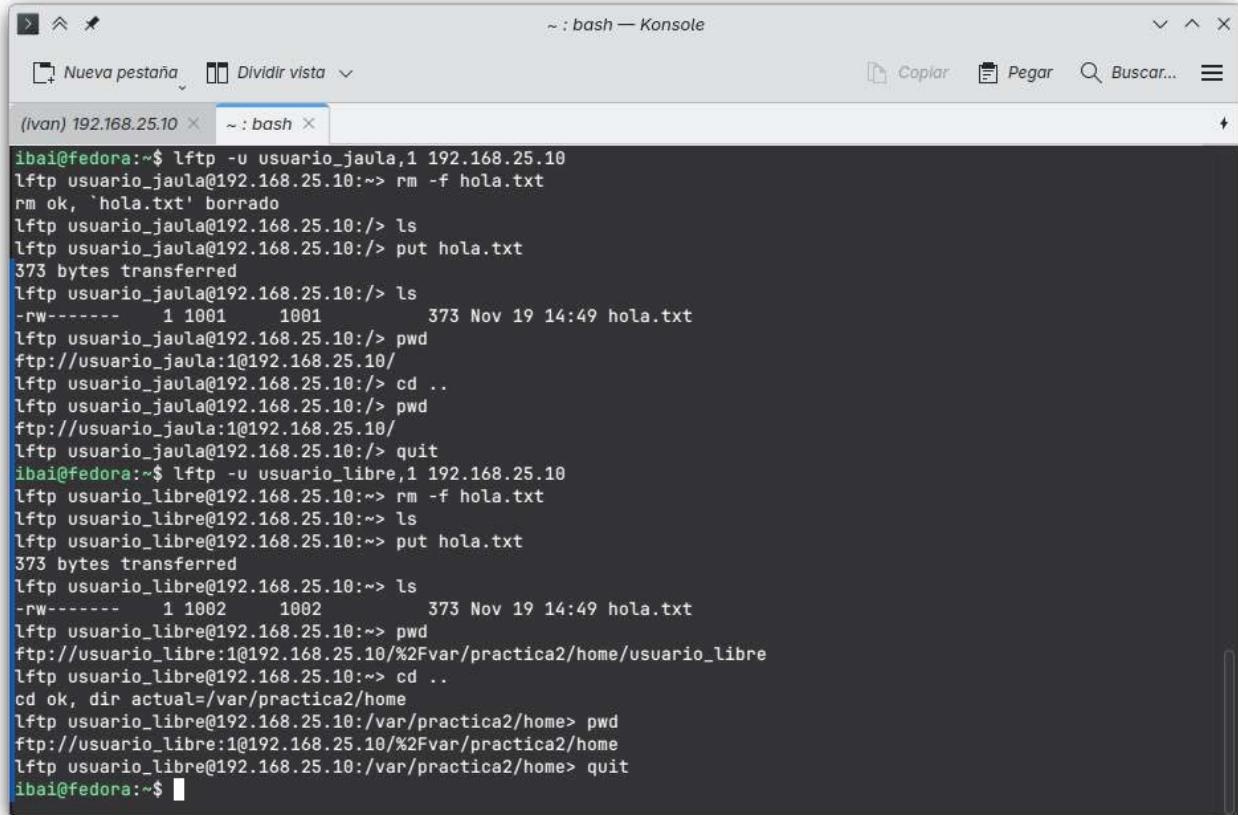
#-----
# Comprobación desde host
#-----
sudo dnf install -y lftp

cat << EOF
rm -f hola.txt
ls
put hola.txt
ls
pwd
cd ..
pwd
quit
EOF

lftp -u usuario_jaula,1 192.168.25.10
lftp -u usuario_libre,1 192.168.25.10

```

Fragmento de código 1: Configuración y comprobación de este servicio en  Debian GNU/Linux 13 (Trixie).



The screenshot shows a terminal window titled "bash — Konsole". It contains a transcript of an lftp session. The session starts with a user named "ibai" connecting to a host at 192.168.25.10. The user performs several operations: removes a file named "hola.txt", lists files, and uploads a new "hola.txt" file. This is followed by another lftp session from a user named "usuario\_jaula" who also performs similar operations. Finally, an lftp session from a user named "usuario\_libre" is shown, which is identical to the previous ones. The terminal interface includes standard Linux navigation keys like arrow keys and function keys.

```
ibai@fedora:~$ lftp -u usuario_jaula,1 192.168.25.10
lftp usuario_jaula@192.168.25.10:~> rm -f hola.txt
rm ok, `hola.txt' borrado
lftp usuario_jaula@192.168.25.10:~> ls
lftp usuario_jaula@192.168.25.10:~> put hola.txt
373 bytes transferred
lftp usuario_jaula@192.168.25.10:~> ls
-rw----- 1 1001 1001 373 Nov 19 14:49 hola.txt
lftp usuario_jaula@192.168.25.10:~> pwd
ftp://usuario_jaula:1@192.168.25.10/
lftp usuario_jaula@192.168.25.10:~> cd ..
lftp usuario_jaula@192.168.25.10:~> pwd
ftp://usuario_jaula:1@192.168.25.10/
lftp usuario_jaula@192.168.25.10:~> quit
ibai@fedora:~$ lftp -u usuario_libre,1 192.168.25.10
lftp usuario_libre@192.168.25.10:~> rm -f hola.txt
lftp usuario_libre@192.168.25.10:~> ls
lftp usuario_libre@192.168.25.10:~> put hola.txt
373 bytes transferred
lftp usuario_libre@192.168.25.10:~> ls
-rw----- 1 1002 1002 373 Nov 19 14:49 hola.txt
lftp usuario_libre@192.168.25.10:~> pwd
ftp://usuario_libre:1@192.168.25.10/%2Fvar/practica2/home/usuario_libre
lftp usuario_libre@192.168.25.10:~> cd ..
cd ok, dir actual=/var/practica2/home
lftp usuario_libre@192.168.25.10:/var/practica2/home> pwd
ftp://usuario_libre:1@192.168.25.10/%2Fvar/practica2/home
lftp usuario_libre@192.168.25.10:/var/practica2/home> quit
ibai@fedora:~$
```

Figura 1: Demostración desde *host* donde se muestra una sesión desde un cliente FTP con dos usuarios, uno enjaulado y otro sin enjaular al servidor alojado en  Debian GNU/Linux 13 (Trixie).

## 1.2. ● FreeBSD 14.3 con proftpd

```
#!/bin/sh
su -
#-----
# Variables de entorno
#-----
P3ASORC_SERVICIO=ftp
P3ASORC_SISTEMA=unix

# NO TOCAR
P3ASORC_MEMORIA=/home/ivan/SUPERMEMORIA/$P3ASORC_SERVICIO
P3ASORC_CONFIG=$P3ASORC_MEMORIA/ficheros_configuracion
P3ASORC_LOG=$P3ASORC_MEMORIA/$P3ASORC_SISTEMA.log
P3ASORC_HISTORIAL=$P3ASORC_MEMORIA/history$P3ASORC_SISTEMA.txt
P3ASORC_BSD_RC=/etc/rc.conf
```

```

# ---
P3ASORC_FTP_HOME=/var/practica2/home
P3ASORC_FTP_USER1=usuario_jaula
P3ASORC_FTP_USER2=usuario_libre

P3ASORC_FTP_CONFIG_1=/usr/local/etc/proftpd.conf
P3ASORC_FTP_LOG=/var/log/proftpd.log
P3ASORC_FTP_LOG1=/var/log/proftpd.xtr.log
P3ASORC_FTP_LOG2=/var/log/proftpd.cmd.log

#-----
# Servicio
#-----
# PASO 1: Usuarios
# Borrar logs anteriores
rm -rf $P3ASORC_FTP_LOG
rm -rf $P3ASORC_FTP_LOG1
rm -rf $P3ASORC_FTP_LOG2

# Borrar si existen de antes
pw user del -n $P3ASORC_FTP_USER1 -r
pw user del -n $P3ASORC_FTP_USER2 -r
rm -rf $P3ASORC_FTP_HOME

# Crearlos
mkdir -p $P3ASORC_FTP_HOME
echo "1" | pw user add -n $P3ASORC_FTP_USER1 \
-d $P3ASORC_FTP_HOME/$P3ASORC_FTP_USER1 \
-s /usr/local/bin/bash \
-m -h 0

echo "1" | pw user add -n $P3ASORC_FTP_USER2 \
-d $P3ASORC_FTP_HOME/$P3ASORC_FTP_USER2 \
-s /usr/local/bin/bash \
-m -h 0

# PASO 2: Instalar paquetes
pkg remove -y proftpd
pkg install -y proftpd

# PASO 3: Configurar
# Enjaula a todos menos a uno
cat << EOF > $P3ASORC_FTP_CONFIG_1
RequireValidShell off
UseReverseDNS off
LogFormat detallado "%t [IP: %h] [User: %u] \"%%r\" (Status: %s) - Bytes: %b"
SystemLog $P3ASORC_FTP_LOG
LogLevel 9
TransferLog $P3ASORC_FTP_LOG1
ExtendedLog $P3ASORC_FTP_LOG2 ALL detallado
ServerName "Museo del Prado FTP"
ServerType standalone
DefaultServer on
ScoreboardFile /var/run/proftpd/proftpd.scoreboard
Port 21
UseIPv6 on
Umask 022
MaxInstances 30

```

```

CommandBufferSize      512
User                  nobody
Group                nogroup
DefaultRoot ~ !$P3ASORC_FTP_USER2
AllowOverwrite        on
<Limit SITE_CHMOD>
  DenyAll
</Limit>
EOF

# PASO 4: Aplica cambios
sysrc proftpd_enable="YES"
service proftpd restart

#-----
# Valida servicio
#-----
service proftpd status
sockstat | grep -E 'ftp'

#-----
# Extraer logs, configs e historial
#-----
rm -rf $P3ASORC_MEMORIA
mkdir -p $P3ASORC_MEMORIA
mkdir -p $P3ASORC_CONFIG

cp $P3ASORC_FTP_CONFIG_1 $P3ASORC_CONFIG
cp $P3ASORC_FTP_LOG $P3ASORC_LOG
cat $P3ASORC_FTP_LOG1 >> $P3ASORC_LOG
cat $P3ASORC_FTP_LOG2 >> $P3ASORC_LOG
cp $P3ASORC_BSD_RC $P3ASORC_CONFIG

history > $P3ASORC_HISTORIAL
chmod 777 -R $P3ASORC_MEMORIA

tree $P3ASORC_MEMORIA

#-----
# Comprobación desde host
#-----
sudo dnf install -y lftp

cat << EOF
rm -f hola.txt
ls
put hola.txt
ls
pwd
cd ..
pwd
quit
EOF

lftp -u usuario_jaula,1 192.168.25.11
lftp -u usuario_libre,1 192.168.25.11

```

Fragmento de código 2: Configuración y comprobación de este servicio en ● FreeBSD 14.3.

```

(livan) 192.168.25.11 ~ : bash ~
ibai@fedora:~$ lftp -u usuario_jaula,1 192.168.25.11
lftp usuario_jaula@192.168.25.11:~> rm -f hola.txt
lftp usuario_jaula@192.168.25.11:/> ls
lftp usuario_jaula@192.168.25.11:/> put hola.txt
373 bytes transferred
lftp usuario_jaula@192.168.25.11:/> ls
-rw-r--r-- 1 usuario_jaula usuario_jaula 373 Nov  4 21:44 hola.txt
lftp usuario_jaula@192.168.25.11:/> pwd
ftp://usuario_jaula:1@192.168.25.11/
lftp usuario_jaula@192.168.25.11:/> cd ..
lftp usuario_jaula@192.168.25.11:/> pwd
ftp://usuario_jaula:1@192.168.25.11/
lftp usuario_jaula@192.168.25.11:/> quit
ibai@fedora:~$ lftp -u usuario_libre,1 192.168.25.11
lftp usuario_libre@192.168.25.11:~> rm -f hola.txt
lftp usuario_libre@192.168.25.11:~> ls
lftp usuario_libre@192.168.25.11:~> put hola.txt
373 bytes transferred
lftp usuario_libre@192.168.25.11:~> ls
-rw-r--r-- 1 usuario_libre usuario_libre 373 Nov  4 21:44 hola.txt
lftp usuario_libre@192.168.25.11:~> pwd
ftp://usuario_libre:1@192.168.25.11/%2Fvar/practica2/home/usuario_libre
lftp usuario_libre@192.168.25.11:~> cd ..
cd ok, dir actual=/var/practica2/home
lftp usuario_libre@192.168.25.11:/var/practica2/home> pwd
ftp://usuario_libre:1@192.168.25.11/%2Fvar/practica2/home
lftp usuario_libre@192.168.25.11:/var/practica2/home> quit
ibai@fedora:~$ 

```

Figura 2: Demostración desde *host* donde se muestra una sesión desde un cliente FTP con dos usuarios, uno enjaulado y otro sin enjaular al servidor alojado en ● FreeBSD 14.3.

## 1.3. Microsoft Windows Server 2025 con IIS-FTP

### 1.3.1. Creación de usuarios

- Acceder a Administrador del Servidor > Herramientas > Administración de equipos.
- Dirigirse a Usuarios y grupos locales > Usuarios.
- Desde aquí crear el usuario *usuario\_jaula* con la contraseña *PatataCaliente123*.
- Volver a crear otro usuario *usuario\_libre* con la contraseña *PatataCaliente123*.
- Crear el grupo *UsuariosFTP* y asignar a los recién creados.

### 1.3.2. Estructura de directorios y permisos

Emplearemos la unidad D: para albergar este servicio.

- En el Explorador de Archivos, ir a la unidad D:.
- Crear todo directorio padre de D:\FTP\_Raiz\LocalUser\usuario\_jaula.
- Clic derecho en D:\FTP\_Raiz > Propiedades > Seguridad.
- Editar > Agregar > *usuario\_libre* y asignar todo permiso posible.
- Navegar a D:\FTP\_Raiz\LocalUser\usuario\_jaula y click derecho sobre la carpeta, luego navegar hacia Propiedades > Seguridad.

- Editar > Agregar > usuario\_jaula y asignar todo permiso posible.

### 1.3.3. Configuración del servicio

1. Abrir *Administrador de Internet Information Services (IIS)*.

2. Clic derecho en *Sitios* > **Agregar sitio FTP...**

- **Nombre del sitio:** MiServidorFTP.
- **Ruta de acceso física:** D:\FTP\_Raiz.

3. **Enlaces:**

- **Dirección IP:** Todas las no asignadas.
- **SSL:** Sin SSL.

4. **Autenticación y Autorización:**

- **Autenticación:** Básica.
- **Autorización:** No seleccionada (Se dejará vacío para usar reglas explícitas).
- Finalizar.

5. Seleccionar el sitio MiServidorFTP. Doble clic en **Reglas de autorización de FTP**.

6. Eliminar cualquier regla genérica existente.

7. Clic derecho > **Agregar regla de permiso...**

- **Usuarios especificados:** usuario\_jaula → **Permisos:** Leer, Escribir.

8. Clic derecho > **Agregar regla de permiso...**

- **Usuarios especificados:** usuario\_libre → **Permisos:** Leer, Escribir.

9. Seleccionar MiServidorFTP y doble clic en **Aislamiento de usuario de FTP**.

10. Seleccionar la opción: *Directorio de nombres de usuario (deshabilitar directorios virtuales globales)*. Luego en **Aplicar**.

11. En el panel izquierdo de IIS, desplegar el sitio MiServidorFTP.

12. Se observará la carpeta LocalUser. Desplegarla (si no aparece contenido, clic derecho > Actualizar).

13. **Creación del enlace a la raíz:**

- Clic derecho sobre la carpeta amarilla LocalUser > **Agregar directorio virtual...**
- **Alias:** usuario\_libre (Debe coincidir con el nombre de usuario).
- **Ruta de acceso física:** D:\FTP\_Raiz.
- Clic en **Aceptar**.

14. Desde el registro FTP pulsar sobre seleccionar campos W3C y marcar todo.

```

#!/bin/sh
#-----
# Comprobación desde host
#-----
sudo dnf install -y lftp

cat << EOF
rm -f hola.txt
ls
put hola.txt
ls
pwd
cd ..
pwd
quit
EOF

lftp -u usuario_jaula,PatataCaliente123 192.168.25.12
lftp -u usuario_libre,PatataCaliente123 -e "cd LocalUser/usuario_libre"
192.168.25.12

```

Fragmento de código 3: Configuración y comprobación de este servicio en Microsoft Windows Server 2025.

The screenshot shows a terminal window titled 'bash — Konsole'. The session starts with the user 'ibai' at a Fedora host. The user runs an LFTP command to connect to a server at 192.168.25.12. The session then splits into two parallel LFTP sessions:

- User 1 (Enjaulado):** Connected as 'usuario\_jaula' (password 'PatataCaliente123'). The user removes a file ('hola.txt'), lists files ('ls'), and uploads a file ('put hola.txt'). The upload is successful, showing '373 bytes transferred'.
- User 2 (Libre):** Connected as 'usuario\_libre' (password 'PatataCaliente123'). The user changes directory ('cd LocalUser/usuario\_libre'), removes the file ('rm -rf hola.txt'), lists files ('ls'), and uploads the file ('put hola.txt'). The upload is successful, showing '373 bytes transferred'.

Both users then quit the session.

Figura 3: Demostración desde *host* donde se muestra una sesión desde un cliente FTP con dos usuarios, uno enjaulado y otro sin enjaular al servidor en Microsoft Windows Server 2025.

Para los registros, he copiado el archivo del directorio C:\inetpub\logs\LogFiles y lo renombrado a windows.csv.

## 2. Mensajería instantánea

«Mensajería instantánea (*ejabberd*). Se instalará un servidor de mensajería instantánea tipo *ejabberd* y se utilizarán dos clientes de mensajería logados en sendos clientes, manteniendo una conversación.»

---

El servicio de mensajería instantánea implementado se basa en **XMPP** (*Extensible Messaging and Presence Protocol* o Protocolo Extensible de Mensajería y Presencia). Es un protocolo abierto y basado en XML destinado a la mensajería instantánea en tiempo real, la información de presencia y la gestión de listas de contactos. Al igual que FTP, se ubica en la capa de aplicación del modelo TCP/IP.

El servicio opera bajo una arquitectura **cliente-servidor** descentralizada, similar a la del correo electrónico:

- **El Servidor (*ejabberd*):** Actúa como el núcleo de la comunicación. Gestiona el enrutamiento de mensajes XML, almacena las credenciales de usuario, administra las listas de contactos (*roster*) y gestiona la presencia (estado *online/offline/ocupado*).
- **El Cliente:** Se conecta al servidor (típicamente a través del puerto 52xx) para iniciar una sesión, intercambiar mensajes y actualizar su estado de presencia.

En cuanto a sus orígenes, XMPP comenzó su vida como el proyecto **Jabber**. Fue creado por Jeremie Miller en **1999** como una alternativa abierta y libre a los servicios de mensajería cerrados de la época (como AIM o ICQ). Posteriormente, el protocolo fue formalizado y estandarizado por la IETF (*Internet Engineering Task Force*) en el año **2004**, bajo los **RFC 6120** y **RFC 6121**. Esta estandarización ha permitido que cualquier cliente compatible con XMPP pueda conectarse a cualquier servidor que implemente el protocolo.

Sobre su uso, la elección de **ejabberd** para esta práctica no es trivial. Es un servidor XMPP escrito en **Erlang**, un lenguaje diseñado para sistemas distribuidos de alta disponibilidad. Esto lo convierte en un estándar industrial por su capacidad de tolerancia a fallos y escalabilidad masiva. Se utiliza no solo para chat entre personas (como en WhatsApp o Google Talk en sus inicios), sino también para señalización VoIP, Internet de las Cosas (IoT) y notificaciones *push*.

En el desarrollo de esta práctica, la implementación del servicio de chat persigue los siguientes objetivos técnicos:

1. **Despliegue de Infraestructura XMPP:** Instalación y configuración de un servidor robusto (**ejabberd**) capaz de gestionar dominios virtuales y autenticación segura.
2. **Interoperabilidad y Comunicación:** Demostrar la capacidad del servidor para gestionar sesiones concurrentes, permitiendo que dos usuarios distintos (clientes) inicien sesión simultáneamente y establezcan un flujo de comunicación bidireccional en tiempo real, validando así el correcto enrutamiento de los mensajes XML.

### 2.1. ☰ Debian GNU/Linux 13 (Trixie) con ejabberd

```
#!/bin/sh  
su -
```

```

#-----#
# Variables de entorno
#-----#
P3ASORC_SERVICIO=chat
P3ASORC_SISTEMA=linux

# NO TOCAR
P3ASORC_MEMORIA=/home/ivan/SUPERMEMORIA/$P3ASORC_SERVICIO
P3ASORC_CONFIG=$P3ASORC_MEMORIA/ficheros_configuracion
P3ASORC_LOG=$P3ASORC_MEMORIA/$P3ASORC_SISTEMA.log
P3ASORC_HISTORIAL=$P3ASORC_MEMORIA/history$P3ASORC_SISTEMA.txt

# ---
P3ASORC_CHAT_DOMAIN=$(hostname -f)
P3ASORC_CHAT_USER1=usuario_chat1
P3ASORC_CHAT_USER2=usuario_chat2
P3ASORC_CHAT_PASS=1

# Rutas de configuración y logs de ejabberd
P3ASORC_CHAT_CONFIG_FILE=/etc/ejabberd/ejabberd.yml
P3ASORC_CHAT_LOG_DIR=/var/log/ejabberd
P3ASORC_CHAT_LOG_FILE=$P3ASORC_CHAT_LOG_DIR/ejabberd.log

#-----#
# Servicio
#-----#
# PASO 1: Paquetes y limpieza
rm -rf $P3ASORC_CHAT_LOG_FILE
apt remove -y ejabberd
apt install -y ejabberd

# PASO 2: Configuración
cat << EOF > $P3ASORC_CHAT_CONFIG_FILE
loglevel: debug
log_rotate_count: 0
hosts:
  - $P3ASORC_CHAT_DOMAIN
  - localhost

certfiles:
  - "/etc/ejabberd/ejabberd.pem"

define_macro:
  'TLS_CIPHERS': "HIGH:!aNULL:!eNULL:!3DES:@STRENGTH"
  'TLS_OPTIONS':
    - "no_sslv3"
    - "no_tlsv1"
    - "no_tlsv1_1"
    - "cipher_server_preference"
    - "no_compression"

c2s_ciphers: 'TLS_CIPHERS'
s2s_ciphers: 'TLS_CIPHERS'
c2s_protocol_options: 'TLS_OPTIONS'
s2s_protocol_options: 'TLS_OPTIONS'

listen:
  -

```

```

port: 5222
ip: "::"
module: ejabberd_c2s
max_stanza_size: 262144
shaper: c2s_shaper
access: c2s
starttls_required: true
protocol_options: 'TLS_OPTIONS'

-
port: 5223
ip: "::"
module: ejabberd_c2s
max_stanza_size: 262144
shaper: c2s_shaper
access: c2s
tls: true
protocol_options: 'TLS_OPTIONS'

-
port: 5269
ip: "::"
module: ejabberd_s2s_in
max_stanza_size: 524288
shaper: s2s_shaper

-
port: 5443
ip: "::"
module: ejabberd_http
tls: true
protocol_options: 'TLS_OPTIONS'
request_handlers:
  /api: mod_http_api
  /bosh: mod_bosh
  /ws: ejabberd_http_ws

-
port: 5280
ip: "::"
module: ejabberd_http
tls: true
protocol_options: 'TLS_OPTIONS'
request_handlers:
  /admin: ejabberd_web_admin
  /.well-known/acme-challenge: ejabberd_acme

-
port: 5478
ip: "::"
transport: udp
module: ejabberd_stun
use_turn: true

-
port: 1883
ip: "::"
module: mod_mqtt
backlog: 1000

disable_sasl_mechanisms:
- "digest-md5"
- "X-OAUTH2"

```

```

s2s_use_starttls: required
auth_password_format: scram
acl:
  admin:
    user:
      - ""

local:
  user_regex: ""
loopback:
  ip:
    - 127.0.0.0/8
    - ::1/128

access_rules:
  local:
    allow: local
  c2s:
    deny: blocked
    allow: all
  announce:
    allow: admin
  configure:
    allow: admin
  muc_create:
    allow: local
pubsub_createnode:
  allow: local
trusted_network:
  allow: loopback

api_permissions:
  "console commands":
    from: ejabberd_ctl
    who: all
    what: "*"
  "webadmin commands":
    from: ejabberd_web_admin
    who: admin
    what: "*"
  "admin access":
    who:
      access:
        allow:
          - acl: loopback
          - acl: admin
      oauth:
        scope: "ejabberd:admin"
        access:
          allow:
            - acl: loopback
            - acl: admin
    what:
      - "*"
      - "!stop"
      - "!start"
  "public commands":
    who:

```

```

ip: 127.0.0.1/8
what:
- status
- connected_users_number

shaper:
normal:
rate: 3000
burst_size: 20000
fast: 200000

shaper_rules:
max_user_sessions: 10
max_user_offline_messages:
5000: admin
100: all
c2s_shaper:
none: admin
normal: all
s2s_shaper: fast

modules:
mod_adhoc: {}
mod_admin_extra: {}
mod_announce:
access: announce
mod_avatar: {}
mod_blocking: {}
mod_bosh: {}
mod_caps: {}
mod_carboncopy: {}
mod_client_state: {}
mod_configure: {}
mod_disco: {}
mod_fail2ban: {}
mod_http_api: {}
mod_last: {}
mod_mqtt: {}
mod_muc:
access:
- allow
access_admin:
- allow: admin
access_create: muc_create
access_persistent: muc_create
access_mam:
- allow
default_room_options:
mam: true
mod_muc_admin: {}
mod_offline:
access_max_user_messages: max_user_offline_messages
mod_ping: {}
mod_pres_counter:
count: 5
interval: 60
mod_privacy: {}
mod_private: {}

```

```

mod_pubsub:
  access_createnode: pubsub_createnode
  plugins:
    - flat
    - pep
  force_node_config:
    "eu.siacs.conversations.axolotl.*":
      access_model: open
      storage:bookmarks:
        access_model: whitelist
mod_push: {}
mod_push_keepalive: {}
mod_roster:
  versioning: true
mod_s2s_bidi: {}
mod_s2s_dialback: {}
mod_shared_roster: {}
mod_sic: {}
mod_stream_mgmt:
  resend_on_timeout: if_offline
mod_stun_disco: {}
mod_vcard:
  search: false
mod_vcard_xupdate: {}
mod_version: {}
EOF

# PASO 3: Permisos y aplicar cambios
chown ejabberd:ejabberd $P3ASORC_CHAT_CONFIG_FILE
chmod 640 $P3ASORC_CHAT_CONFIG_FILE
systemctl stop ejabberd
systemctl enable ejabberd
systemctl start ejabberd
sleep 20

# PASO 4: Registro de usuarios
ejabberdctl unregister $P3ASORC_CHAT_USER1 $P3ASORC_CHAT_DOMAIN
ejabberdctl unregister $P3ASORC_CHAT_USER2 $P3ASORC_CHAT_DOMAIN
ejabberdctl register $P3ASORC_CHAT_USER1 $P3ASORC_CHAT_DOMAIN
  $P3ASORC_CHAT_PASS
ejabberdctl register $P3ASORC_CHAT_USER2 $P3ASORC_CHAT_DOMAIN
  $P3ASORC_CHAT_PASS

#-----
# Valida servicio
#-----
systemctl status ejabberd --no-pager
# Verificar puertos (52** es cliente-servidor, 5280 es web admin)
netstat -tunlp | grep -E 'beam|epmd'

#-----
# Extraer logs, configs e historial
#-----
rm -rf $P3ASORC_MEMORIA
mkdir -p $P3ASORC_MEMORIA
mkdir -p $P3ASORC_CONFIG

cp $P3ASORC_CHAT_CONFIG_FILE $P3ASORC_CONFIG

```

```

cp $P3ASORC_CHAT_LOG_FILE $P3ASORC_LOG

history > $P3ASORC_HISTORIAL
chmod 777 -R $P3ASORC_MEMORIA

tree $P3ASORC_MEMORIA

#-----
# Comprobación desde host
#-----
sudo dnf install -y pidgin

pidgin -m

```

Fragmento de código 4: Configuración y comprobación de este servicio en  Debian GNU/Linux 13 (Trixie).

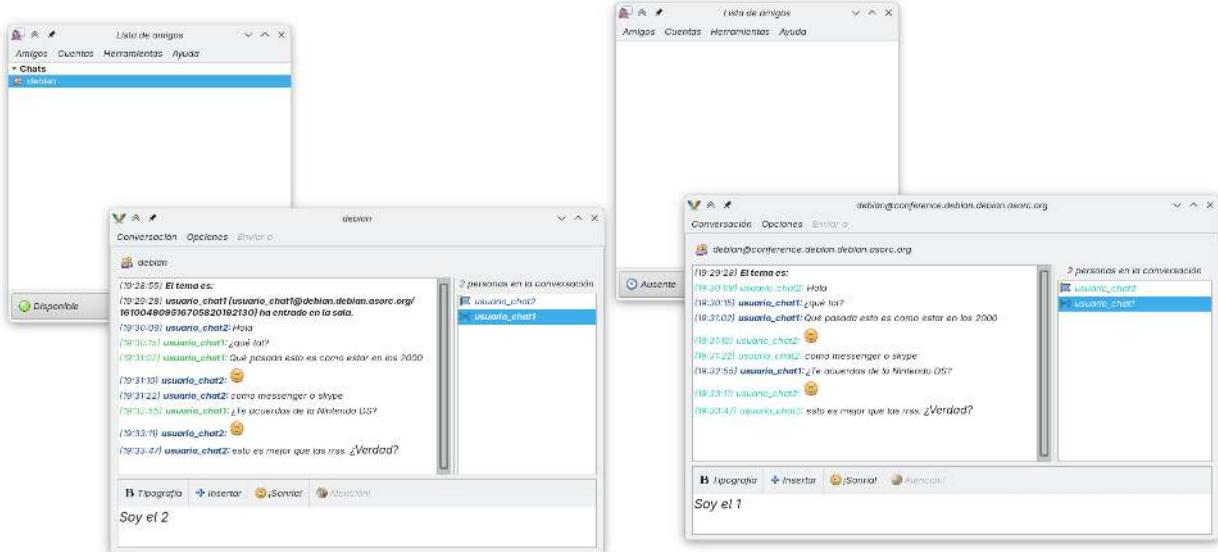


Figura 4: Demostración desde *host* donde se aprecian dos clientes de mensajería logueados en dos clientes, manteniendo una conversación nostálgica con un servidor alojado bajo  Debian GNU/Linux 13 (Trixie).

## 2.2. ● FreeBSD 14.3 con ejabberd

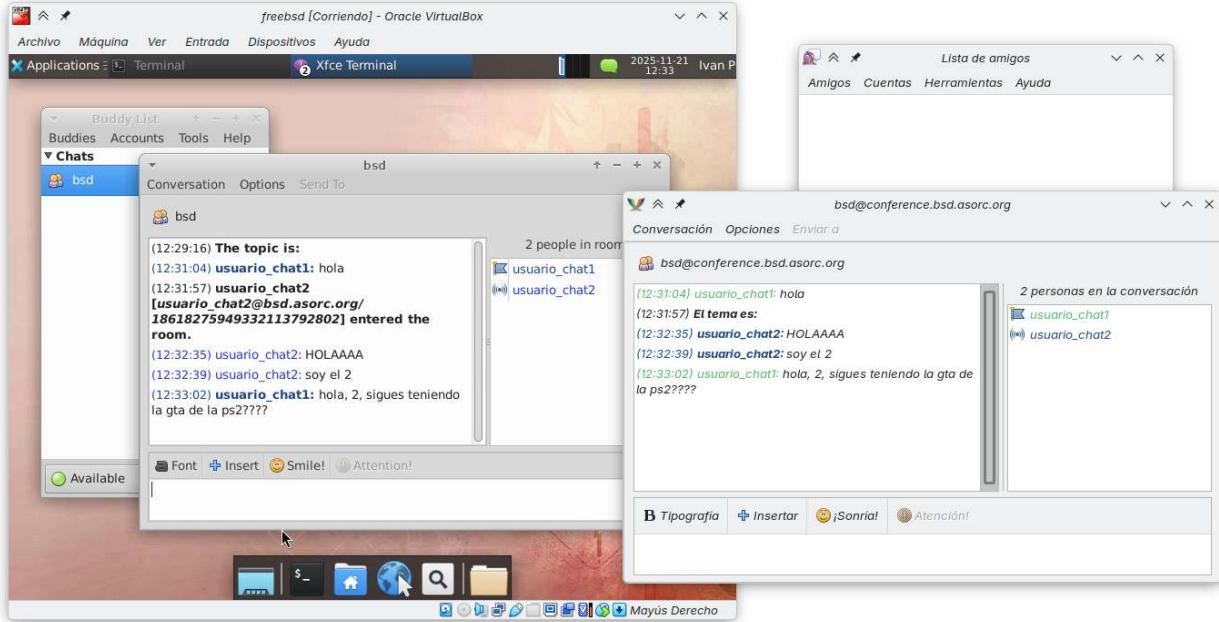


Figura 5: Demostración desde *host* y *guest* donde se aprecian dos clientes de mensajería logueados en dos clientes, manteniendo una conversación nostálgica con un servidor alojado bajo ● FreeBSD 14.3.

```
#!/bin/sh
su -

#-----
# Variables de entorno
#-----
P3ASORC_SERVICIO=chat
P3ASORC_SISTEMA=unix

# Rutas de entrega (NO TOCAR)
P3ASORC_MEMORIA=/home/ivan/SUPERMEMORIA/$P3ASORC_SERVICIO
P3ASORC_CONFIG=$P3ASORC_MEMORIA/ficheros_configuracion
P3ASORC_LOG=$P3ASORC_MEMORIA/$P3ASORC_SISTEMA.log
P3ASORC_HISTORIAL=$P3ASORC_MEMORIA/history$P3ASORC_SISTEMA.txt

# --- Variables específicas del servicio ---
P3ASORC_CHAT_DOMAIN=$(hostname)
P3ASORC_CHAT_USER1=usuario_chat1
P3ASORC_CHAT_USER2=usuario_chat2
P3ASORC_CHAT_PASS=1

# Rutas de configuración y logs de ejabberd
P3ASORC_CHAT_CONFIG_FILE=/usr/local/etc/ejabberd/ejabberd.yml
P3ASORC_CHAT_LOG_DIR=/var/log/ejabberd
P3ASORC_CHAT_LOG_FILE=$P3ASORC_CHAT_LOG_DIR/ejabberd.log

#-----
# Servicio
```

```

#-----#
# PASO 1: Limpieza y paquetes
# Borrar logs anteriores
service ejabberd stop
pkill -9 beam
pkill -9 epmd
pkill -9 ejabberd
rm -rf $P3ASORC_CHAT_LOG_FILE
rm -rf /var/spool/ejabberd

# Instalación limpia
pkg remove -y ejabberd
pkg install -y ejabberd

# Fichero de configuración
cat << EOF > $P3ASORC_CHAT_CONFIG_FILE
loglevel: 5
log_rotate_count: 0
hosts:
- "$P3ASORC_CHAT_DOMAIN"
- "localhost"

certfiles:
- "/usr/local/etc/ejabberd/ejabberd.pem"

listen:
-
  port: 5222
  ip: "0.0.0.0"
  module: ejabberd_c2s
  max_stanza_size: 262144
  shaper: c2s_shaper
  access: c2s
  starttls_required: true
-
  port: 5223
  ip: ":"
  module: ejabberd_c2s
  max_stanza_size: 262144
  shaper: c2s_shaper
  access: c2s
  tls: true
-
  port: 5269
  ip: ":"
  module: ejabberd_s2s_in
  max_stanza_size: 524288
  shaper: s2s_shaper
-
  port: 5443
  ip: ":"
  module: ejabberd_http
  tls: true
  request_handlers:
    /admin: ejabberd_web_admin
    /api: mod_http_api
    /bosh: mod_bosh
    /captcha: ejabberd_captcha

```

```

/upload: mod_http_upload
/ws: ejabberd_http_ws

-
port: 5280
ip: "::"
module: ejabberd_http
request_handlers:
/admin: ejabberd_web_admin
/.well-known/acme-challenge: ejabberd_acme

-
port: 5478
ip: "::"
transport: udp
module: ejabberd_stun
use_turn: true

-
port: 1883
ip: "::"
module: mod_mqtt
backlog: 1000

s2s_use_starttls: optional

acl:
local:
user_regexp: ""

loopback:
ip:
- 127.0.0.0/8
- ::1/128

access_rules:
local:
allow: local

c2s:
deny: blocked
allow: all

announce:
allow: admin

configure:
allow: admin

muc_create:
allow: local

pubsub_createnode:
allow: local

trusted_network:
allow: loopback

api_permissions:
"console commands":
from: ejabberd_ctl
who: all
what: "*"

"webadmin commands":
from: ejabberd_web_admin
who: admin
what: "*"

"adhoc commands":

```

```

from: mod_adhoc_api
who: admin
what: "*"

"http access":
from: mod_http_api
who:
access:
allow:
- acl: loopback
- acl: admin
oauth:
scope: "ejabberd:admin"
access:
allow:
- acl: loopback
- acl: admin
what:
- "*"
- "!stop"
- "!start"

"public commands":
who:
ip: 127.0.0.1/8
what:
- status
- connected_users_number

shaper:
normal:
rate: 3000
burst_size: 20000
fast: 100000

shaper_rules:
max_user_sessions: 10
max_user_offline_messages:
5000: admin
100: all
c2s_shaper:
none: admin
normal: all
s2s_shaper: fast

modules:
mod_adhoc: {}
mod_adhoc_api: {}
mod_admin_extra: {}
mod_announce:
access: announce
mod_avatar: {}
mod_blocking: {}
mod_bosh: {}
mod_caps: {}
mod_carboncopy: {}
mod_client_state: {}
mod_configure: {}
mod_disco: {}
mod_http_api: {}

```

```

mod_http_upload:
    put_url: https://@HOST_URL_ENCODE@:5443/upload
    custom_headers:
        "Access-Control-Allow-Origin": "https://@HOST@"
        "Access-Control-Allow-Methods": "GET,HEAD,PUT,OPTIONS"
        "Access-Control-Allow-Headers": "Content-Type"
mod_last: {}
mod_mam:
    assume_mam_usage: true
    default: always
mod_mqtt: {}
mod_muc:
    access:
        - allow
    access_admin:
        - allow: admin
    access_create: muc_create
    access_persistent: muc_create
    access_mam:
        - allow
    default_room_options:
        mam: true
mod_muc_admin: {}
mod_muc_occupantid: {}
mod_offline:
    access_max_user_messages: max_user_offline_messages
mod_ping: {}
mod_privacy: {}
mod_private: {}
mod_proxy65:
    access: local
    max_connections: 5
mod_pubsub:
    access_createnode: pubsub_createnode
    plugins:
        - flat
        - pep
    force_node_config:
        storage:bookmarks:
            access_model: whitelist
mod_push: {}
mod_push_keepalive: {}
mod_register:
    ip_access: trusted_network
mod_roster:
    versioning: true
mod_s2s_bidi: {}
mod_s2s_dialback: {}
mod_shared_roster: {}
mod_stream_mgmt:
    resend_on_timeout: if_offline
mod_stun_disco: {}
mod_vcard: {}
mod_vcard_xupdate: {}
mod_version:
    show_os: false
EOF

```

```

# Claves (REQUIERE su atención)
cat << EOF | openssl req -new -x509 -days 365 -nodes -out /usr/local/etc/
    ejabberd/ejabberd.pem -keyout /usr/local/etc/ejabberd/ejabberd.pem
AU
Some-State
Alicante
Ministerio de la Hacienda
seccion
$P3ASORC_CHAT_DOMAIN
laligadelosvengadores@gmail.ru

EOF

# PASO 3: Permisos
chown ejabberd:ejabberd $P3ASORC_CHAT_CONFIG_FILE
chmod 640 $P3ASORC_CHAT_CONFIG_FILE
chown ejabberd:ejabberd /usr/local/etc/ejabberd/ejabberd.pem
chmod 600 /usr/local/etc/ejabberd/ejabberd.pem

# PASO 4: Aplicar cambios
sysrc ejabberd_enable="YES"
service ejabberd restart
sleep 20

# PASO 5: Sincronizar galletas
cp /var/spool/ejabberd/.erlang.cookie /root/.erlang.cookie
chmod 400 /root/.erlang.cookie

# PASO 6: Registrar usuarios
EJABBERDCTL="/usr/local/sbin/ejabberdctl"
$EJABBERDCTL unregister $P3ASORC_CHAT_USER1 $P3ASORC_CHAT_DOMAIN
$EJABBERDCTL unregister $P3ASORC_CHAT_USER2 $P3ASORC_CHAT_DOMAIN
$EJABBERDCTL register $P3ASORC_CHAT_USER1 $P3ASORC_CHAT_DOMAIN
    $P3ASORC_CHAT_PASS
$EJABBERDCTL register $P3ASORC_CHAT_USER2 $P3ASORC_CHAT_DOMAIN
    $P3ASORC_CHAT_PASS

#-----
# Valida servicio
#-----

service ejabberd status
sockstat -4 -6 -l | grep beam
tail /var/log/ejabberd/ejabberd.log

#-----
# Extraer logs, configs e historial
#-----

rm -rf $P3ASORC_MEMORIA
mkdir -p $P3ASORC_MEMORIA
mkdir -p $P3ASORC_CONFIG

cp $P3ASORC_CHAT_CONFIG_FILE $P3ASORC_CONFIG
cp $P3ASORC_CHAT_LOG_FILE $P3ASORC_LOG

history > $P3ASORC_HISTORIAL
chmod 777 $P3ASORC_MEMORIA

tree $P3ASORC_MEMORIA

```

```
#-----#
# Comprobación desde host
#-----#
sudo dnf install -y pidgin

pidgin -m
```

Fragmento de código 5: Configuración y comprobación de este servicio en ● FreeBSD 14.3.

## 2.3. Microsoft Windows Server 2025 con Openfire

Para este sistema operativo se ha optado por una alternativa a ejabberd porque esta sugiere la instalación de docker, wsl y el sistema operativo Ubuntu. Si bien, el *host* (un mac intel de hace una década) puede con ello, lo cierto es que aparecen otra serie de problemáticas que desvirtúan el fin propio de este punto [poner en marcha el servicio de mensajería] que acompañan la práctica. Pensemos por un momento: ¿qué puede salir mal si instalamos wsl, un contenedor docker, un sistema operativo pesado como ubuntu dentro de una máquina virtual limitada a dos núcleos y 4 GB de RAM cuyo *host* es un MacBook prehistórico que ejecuta Fedora para poner en marcha un *messenger*? Nos quedaremos con las dudas pues el espacio no es ilimitado y tampoco lo es el tiempo.

El enunciado de la práctica no impone específicamente instalar pues dice «*Se instalará un servidor de mensajería instantánea tipo ejabberd*» dejando lugar a poder instalar una alternativa equivalente.

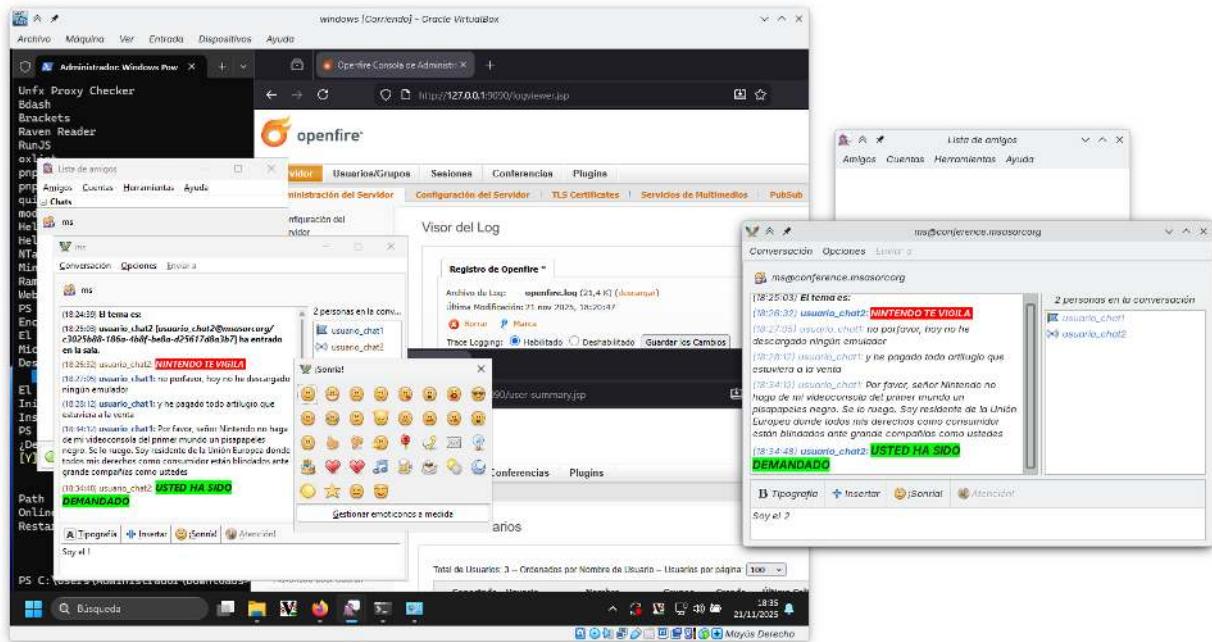


Figura 6: Demostración desde *host* y *guest* donde se aprecian dos clientes de mensajeríalogueados en dos clientes, manteniendo una conversación nostálgica con un servidor alojado bajo Microsoft Windows Server 2025.

Para la implementación del servicio de mensajería instantánea en el entorno Windows Server 2025, se optó por el SW **Openfire** debido a su estabilidad, instalador nativo y capacidad de gestión web, cumpliendo con los requisitos del protocolo XMPP.

### 2.3.1. Descarga e instalación

```
#!/bin/sh

#Enable-WindowsOptionalFeature -Online -FeatureName Containers
#wsl --install
#winget install pidgin Docker.DockerDesktop
# CAMBIAR HOST=msasorcorg
# CAMBIAR PASSWORD=1

winget install Oracle.JDK.25
# Descargar el instalador desde aquí https://www.igniterealtime.org/
#   downloads/
# Hacer lo típico de siguiente siguiente siguiente e instalar

#-----
# Comprobación desde host
#-----
sudo dnf install -y pidgin

pidgin -m
```

Fragmento de código 6: Instalación y comprobación de este servicio en Microsoft Windows Server 2025.

### 2.3.2. Configuración

Procede acceder al asistente de configuración web a través de <http://127.0.0.1:9090>, estableciendo los siguientes parámetros críticos:

- **Dominio y FQDN:** msasorcorg.
- **Idioma:** Español.
- **Base de Datos:** Base de datos interna (*Embedded Database*).
- **Perfil:** Por defecto.
- **Cuenta de Administrador:** Correo admin@example.com y contraseña 1.

Luego iniciar sesión con las credenciales de administrador de este servicio. Para cumplir con el requisito de registro detallado, se accedió a Server→Server Manager→Logs. En la sección **Log Settings**, se habilitó la opción **Trace Logging (Enabled)**. Esto fuerza al sistema a escribir toda la actividad en el archivo **all.log**.

Otro punto importante a recalcar en esta práctica para simplificar la puesta en servicio de este, se sugiere la absoluta inhabilitación del *firewall* de Microsoft Windows Server 2025.

### 2.3.3. Creación de usuarios

Por medio de la consola administrativa web, dentro de la pestaña **Usuarios/Grupos**, se perpetuará el alta los dos solicitados y requeridos usuarios:

- **usuario\_chat1** (Contraseña: 1)
- **usuario\_chat2** (Contraseña: 1)

#### **2.3.4. Extracción de *logs* y configuración**

Se ha copiado desde la ruta de instalación `logs\openfire.log` a la ruta de la memoria del servicio a `windows.csv`. También del directorio `conf` los archivos de interés como `openfire.xml` y `security.xml`.

### 3. Servidor de trabajo en grupo (*groupware*)

«*Servidor de trabajo en grupo. Se instalará un servidor de trabajo en grupo en cada uno de los tres servidores: (Zimbra, OpenXchange, Opengroupware, Microsoft Exchange, ...)*»

---

El **Groupware** (SW de trabajo colaborativo) es un conjunto de programas informáticos diseñados para facilitar el trabajo en equipo y la comunicación entre usuarios conectados a una red. Su función principal es integrar herramientas de mensajería, calendarios, gestión de contactos y documentos en una plataforma unificada, permitiendo la cooperación asíncrona o en tiempo real.

El servicio se basa habitualmente en una arquitectura **cliente-servidor** o basada en web:

- **El servidor:** Centraliza el almacenamiento de correos electrónicos, eventos de calendario y libretas de direcciones, gestionando los permisos y la sincronización entre dispositivos.
- **El cliente:** Puede ser uno pesado (como Outlook o Thunderbird) o, más comúnmente hoy en día, una interfaz web (*Webmail*) que permite al usuario acceder a los recursos compartidos desde cualquier lugar.

En cuanto a sus orígenes, el concepto de trabajo colaborativo asistido por ordenador se remonta a los años 60 con el sistema **PLATO**, pero el término *groupware* fue acuñado por Peter y Trudy Johnson-Lenz en **1978**. El hito comercial que definió el estándar moderno fue el lanzamiento de **Lotus Notes** en **1989**, que introdujo la replicación de bases de datos y el correo electrónico integrado. Posteriormente, soluciones como Microsoft Exchange y alternativas de código abierto como **Zimbra** u **OpenXchange** han evolucionado hacia suites completas de colaboración.

Sobre uso, estas herramientas son la columna vertebral de la comunicación empresarial moderna. Permiten no solo el envío de correos, sino la programación eficiente de reuniones mediante la comprobación de disponibilidad (*free/busy*), la delegación de tareas y la gestión compartida de recursos corporativos.

En el desarrollo de esta práctica, la implementación del servidor de trabajo en grupo persigue los siguientes objetivos técnicos:

1. **Despliegue de la suite colaborativa:** Instalación de una solución integral (tipo Zimbra u OpenXchange) que unifique servicios de MTA (*Mail Transfer Agent*) y bases de datos de colaboración.
2. **Funcionalidad cliente-servidor:** Verificar la correcta sincronización y acceso a los recursos compartidos mediante clientes web o de escritorio, asegurando la operatividad del flujo de trabajo en grupo.

### 3.1. Δ Debian GNU/Linux 13 (Trixie) con citadel

```
#!/bin/sh
su -

#-----
# Variables de entorno
#-----
P3ASORC_SERVICIO=grupo
P3ASORC_SISTEMA=linux

# NO TOCAR
P3ASORC_MEMORIA=/home/ivan/SUPERMEMORIA/$P3ASORC_SERVICIO
P3ASORC_CONFIG=$P3ASORC_MEMORIA/ficheros_configuracion
P3ASORC_LOG=$P3ASORC_MEMORIA/$P3ASORC_SISTEMA.log
P3ASORC_HISTORIAL=$P3ASORC_MEMORIA/history$P3ASORC_SISTEMA.txt

# ---
P3ASORC_GRUPO_CONFIG=/etc/citadel/citadel.rc
P3ASORC_GRUPO_LOG=/usr/local/citadel/data/log.0000000001

#-----
# Servicio (no backtrack)
#-----
# PASO 1: Instalar, compilar y configurar el servicio
curl https://easyinstall.citadel.org/install | bash
# <Enter>
# <Enter>
# <Enter>
# Y
# <Enter>
# 1
# <Enter>
# <Enter>
# <Enter> # Usará el puerto 504
# 0
# 8080
# 8443

#-----
# Valida servicio
#-----
systemctl status citadel
systemctl status webcit-http
systemctl status webcit-https
netstat -tunlp | grep -E 'webcit'

#-----
# Extraer logs, configs e historial
#-----
rm -rf $P3ASORC_MEMORIA
mkdir -p $P3ASORC_MEMORIA
mkdir -p $P3ASORC_CONFIG

cp $P3ASORC_GRUPO_CONFIG $P3ASORC_CONFIG
tr -cd '\11\12\15\40-\176' < $P3ASORC_GRUPO_LOG > $P3ASORC_LOG
systemctl status --no-pager -l citadel >> $P3ASORC_LOG
systemctl status --no-pager -l webcit-http >> $P3ASORC_LOG
```

```

systemctl status --no-pager -l webkit-https >> $P3ASORC_LOG

history > $P3ASORC_HISTORIAL
chmod 777 -R $P3ASORC_MEMORIA

tree $P3ASORC_MEMORIA

#-----
# Comprobación desde host
#-----
firefox 192.168.25.10:8080
# admin
# 1

# Desde guest
firefox localhost:8080
# usuari01
# 1

```

Fragmento de código 7: Configuración y comprobación de este servicio en  Debian GNU/Linux 13 (Trixie).

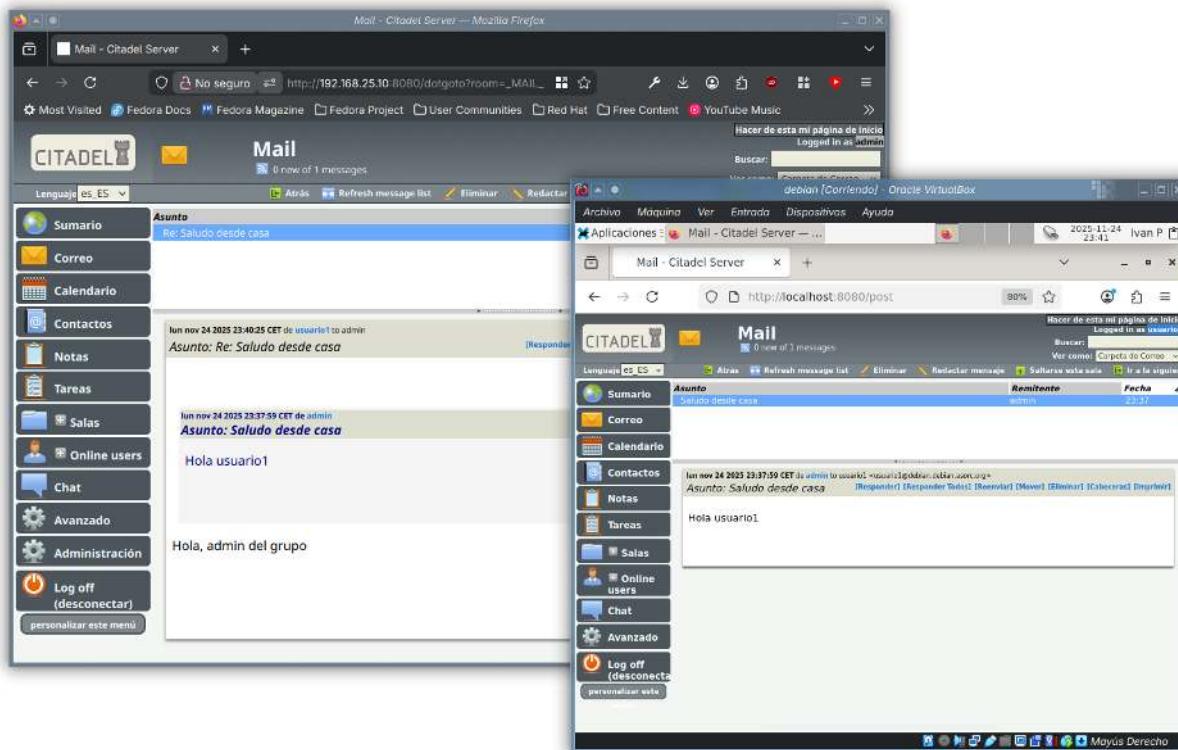


Figura 7: Demostración desde *host* (*logueado* como administrador) y *host* (*logueado* como *usuari01*) donde se muestra una sesión desde un cliente webmail entre estos usuarios empleando el servidor de este servicio alojado en  Debian GNU/Linux 13 (Trixie). *Guest* lee el correo de *host*, y *host* visualiza la respuesta de *guest*.

### 3.1.1. Motivo de la elección de esta alternativa

Si bien este no aparece entre la pequeña lista de la concretamente el enunciado habla, sí cumple con ser *groupware*. Se propone esta alternativa por la ligereza, la visibilidad a proyectos pequeños FOSS y su sencillez de configuración. Dado que no es una pieza de SW de renombre y como acto de transparencia aquí se exponen sus características a base de pantallazos. Pues esta suite unifica diversas utilidades utilizadas en el trabajo colaborativo como correo, notas, wiki, tareas e incluso calendario y entre otras... ya listos para usar sin mucho que retocar.

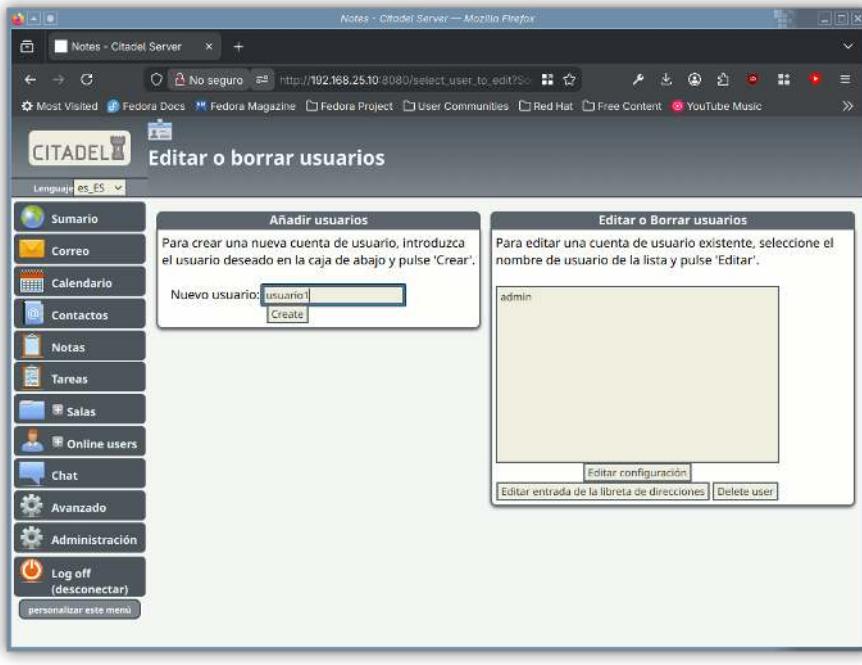


Figura 8: El servicio citadel permite el registro de usuarios de antemano o bien de forma ad-hoc registrándose por cuenta propia. También es posible modificar sus preferencias siendo administrador.

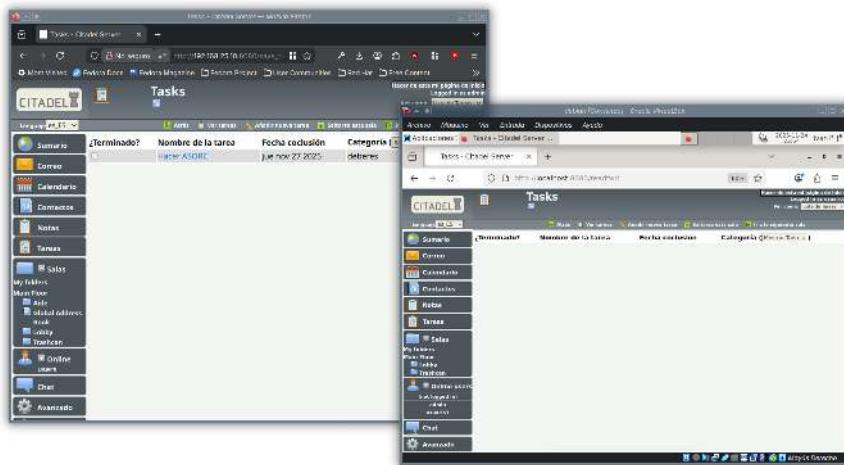


Figura 9: El servicio citadel permite un espacio privado para cada usuario. En este caso el usuario solo puede ver su propia lista de tareas.

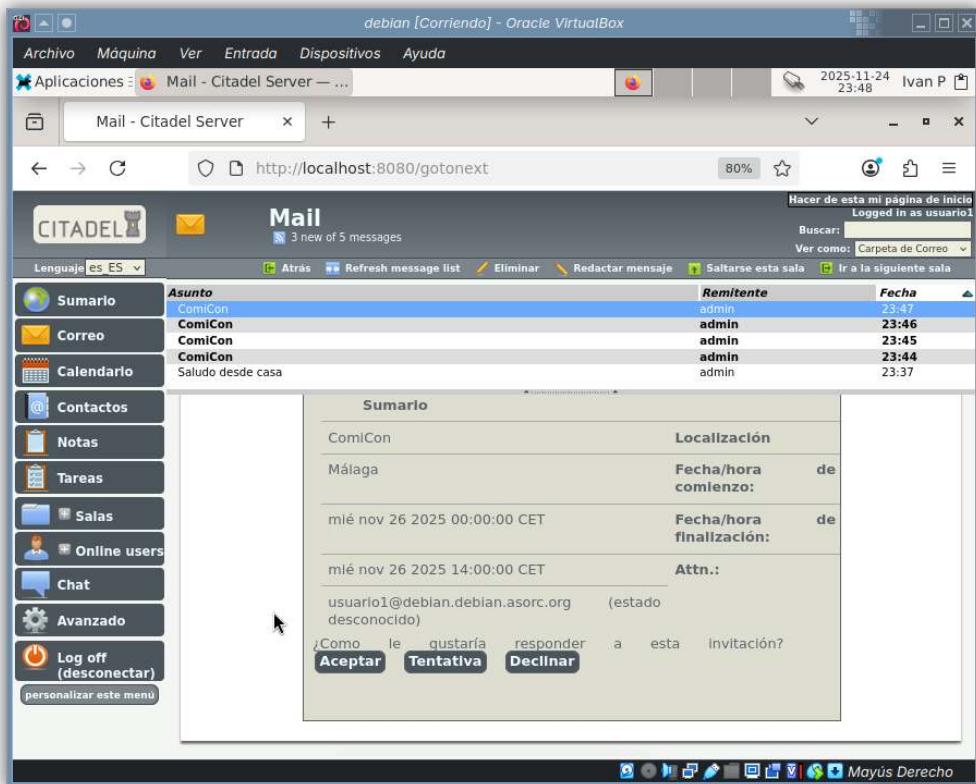


Figura 10: El servicio citadel permite invitar a usuarios a que compartan eventos del calendario. En este caso, el administrador ha enviado muchas invitaciones para la ComicCon.

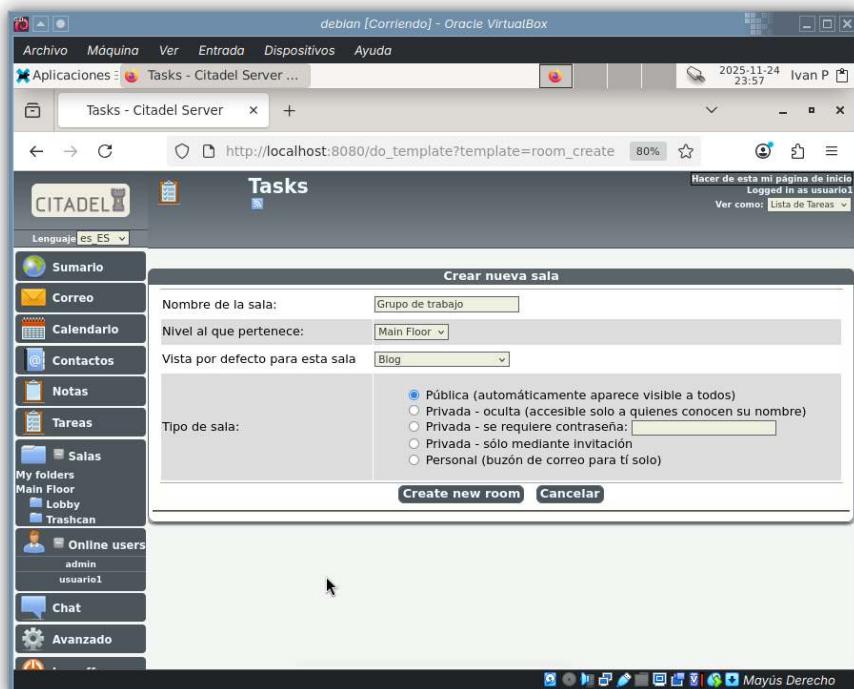


Figura 11: El servicio citadel mostrando las diferentes formas de compartir trabajo. Pudiendo elegir entre compartir notas, agendas de contacto, lista de tareas y más ítems.

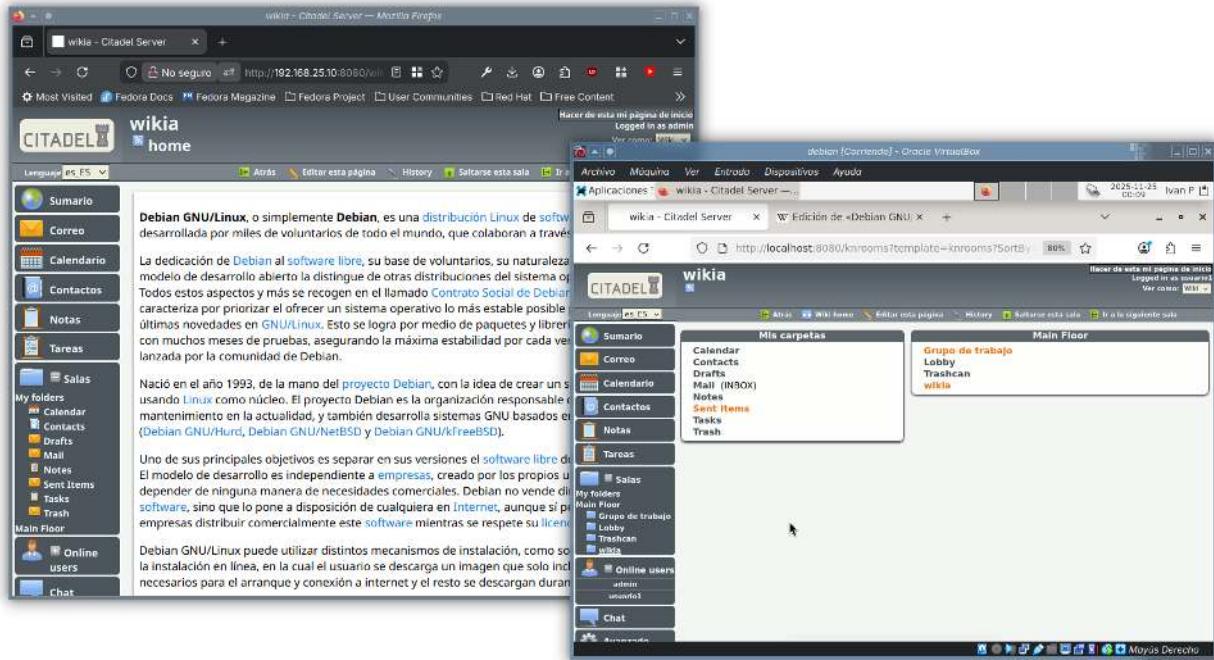


Figura 12: El servicio citadel permite ver la wiki creada por otro usuario.

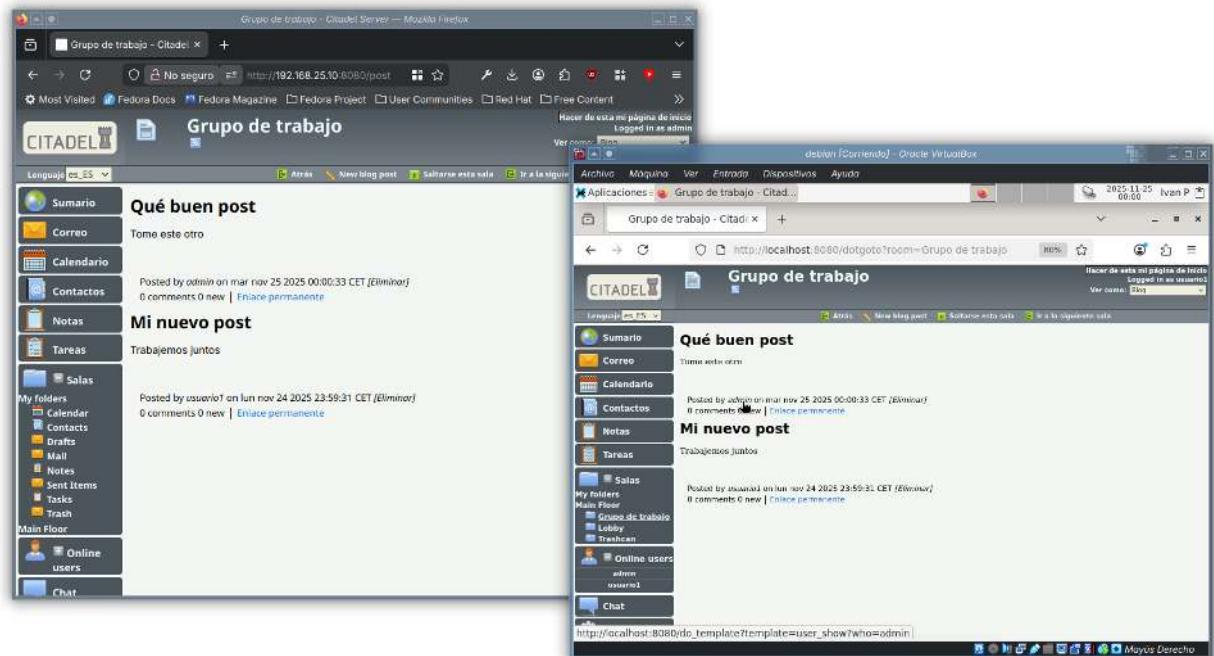


Figura 13: El servicio citadel permite la publicación de *posts* y ofrece la posibilidad de hacer *microblogging*.

### 3.2. ● FreeBSD 14.3 con SOGo

El *script* resultante es más largo porque en este se pretende mostrar las ocurrencias e ingeniería para conseguir este punto. Tomaron 4 intentos

```
#!/bin/sh
```

```

su -

#-----
# Variables de entorno
#-----
P3ASORC_SERVICIO=grupo
P3ASORC_SISTEMA=unix

# NO TOCAR
P3ASORC_MEMORIA=/home/ivan/SUPERMEMORIA/$P3ASORC_SERVICIO
P3ASORC_CONFIG=$P3ASORC_MEMORIA/ficheros_configuracion
P3ASORC_LOG=$P3ASORC_MEMORIA/$P3ASORC_SISTEMA.log
P3ASORC_HISTORIAL=$P3ASORC_MEMORIA/history$P3ASORC_SISTEMA.txt
P3ASORC_BSD_RC=/etc/rc.conf

# ---- Intento cuarto #4
P3ASORC_GRUPO_CONFIG=/usr/local/etc/apache24/Includes/sogo.conf
P3ASORC_GRUPO_CONFIG1=/usr/local/etc/sogo/sogo.conf
P3ASORC_GRUPO_FDIR=/usr/local/GNUstep/Local/Library/SOGO/WebServerResources
P3ASORC_GRUPO_INTERPHONE_NUMBERLESS_INTERFACES_ATTENDERS_AND_LISTENERS
    =127.0.0.1:20000
P3ASORC_GRUPO_DB=sogod
P3ASORC_GRUPO_DB_USER=sogo
P3ASORC_GRUPO_DB_PASS=1

#-----
# Servicio (no backtrack) Intento #1 (Alternativa)
#-----
# PASO 1: Instalar paquete
pkg install -y moregroupware apache24 mod_php82 mariadb106-server

# PASO 2: Configuración de BD
sysrc mysql_enable="YES"
service mysql-server start
sleep 5

mysql -u root << EOF
CREATE DATABASE IF NOT EXISTS moregroupware;
CREATE USER IF NOT EXISTS 'moregw_user'@'localhost' IDENTIFIED BY '1';
GRANT ALL PRIVILEGES ON moregroupware.* TO 'moregw_user'@'localhost';
FLUSH PRIVILEGES;
EOF

# PASO 3: Configuración Web (Apache)...
cat << EOF > $P3ASORC_GRUPO_APACHECONF
Alias /phpgroupware /usr/local/www/phpgroupware/

<Directory /usr/local/www/phpgroupware/>
    Options Indexes FollowSymLinks MultiViews
    AllowOverride None
    Require all granted
    DirectoryIndex index.php login.php
</Directory>

<Directory /usr/local/www/phpgroupware/setup/>
    Options Indexes FollowSymLinks MultiViews
    AllowOverride None
    Require all granted

```

```

        DirectoryIndex index.php
</Directory>
EOF

chown -R www:www $P3ASORC_GRUPO_PHPGWDIR
sysrc apache24_enable="YES"
service apache24 restart
service apache24 status
service mysql-server status

#-----
# Servicio (no backtrack) Intento #2 (La matrioska)
#-----

# Este Intento consistía en:
# 1. Instalar VirtualBox
# 2. Compartir la carpeta de la máquina virtual de mi debian (donde estaba
# el servicio)
# 3. La máquina funciona y arranca (ES COMO UNA MATRIOSKA, una VM dentro de
# otra VM)
# 4. El disco de debian, por haber osado utilizar las shapshots, pasaba a
# ser dinámico, y sus contenidos no eran actuales.
# 5. Copiar el disco a un espacio donde hubiera memoria requería tiempo,
# costes de lectura y escritura de al menos el doble que lo normal, y
# espacio (evitndemente)
# 6. Además iba lenta la VM
# 7. Idea era usar un tipo de conexión que permitiera enmascarar bajo la
# misma que el ahora host freebsd.
# Demasiada mente galaxia utilizada, muy enadeble a fallos y la lentitud
# hicieron echarme para atrás

#-----
# Servicio (no backtrack) Intento #3 (Contenedores)
#-----

P3ASORC_GRUPO_DB_ROOT_PASS="1"
P3ASORC_GRUPO_DB_USER_PASS="1"
P3ASORC_GRUPO_NETWORK_NAME="egroupware-net"

# PASO 1: Instalar podman
pkg install -y podman

# PASO 2: Declarar montaje
mount -t fdescfs fdesc /dev/fd
nano /etc/fstab
# meter esto
# fdesc          /dev/fd          fdescfs  rw      0      0

# PASO 3: Habilitación de servicio
service podman enable
sudo sysrc podman_enable="YES"

# PASO 4: Aplicar configuración por defecto
cp /usr/local/etc/containers/pf.conf.sample /etc/pf.conf
kldload pf
sysctl net.pf.filter_local=1

# REQUIERE PROCESO A MANO
nano /etc/sysctl.conf

```

```

# net.pf.filter_local=1
nano /etc/pf.conf
# nat-anchor "cni-rdr/*"
# v4egress_if = "em0"

# PASO 4: Habilitar e iniciar Linux
pkg install -y linux_base-c7
mkdir -p /compat/linux/proc
mkdir -p /compat/linux/sys
mkdir -p /compat/linux/dev
mount -t linprocfs linproc /compat/linux/proc
mount -t linsysfs linsys /compat/linux/sys
mount -t devfs devfs /compat/linux/dev
service pf start
sysrc linux_enable="YES"
service linux start

# PASO 5: Crear espacio para contenerización
truncate -s 4G /var/zfs_virtual.img
zpool create -f zroot /var/zfs_virtual.img
sysrc zfs_enable="YES"
service zfs start
zfs create -o mountpoint=/var/db/containers zroot/containers
# PASO 6: Habilitar docker hub
cat << EOF > /usr/local/etc/containers/registries.conf
# Configuración de registros para Podman en FreeBSD
# Formato TOML v2
# Lista de registros donde buscar cuando no se especifica el servidor
unqualified-search-registries = ["docker.io", "quay.io"]
# Configuración para Docker Hub
[[registry]]
location = "docker.io"
# Permitir conexiones HTTPS (seguro)
insecure = false
# Configuración para Quay.io
[[registry]]
location = "quay.io"
insecure = false
EOF
service podman start
# PASO 7: Crear red
podman network create $P3ASORC_GRUPO_NETWORK_NAME --subnet 10.5.0.0/24

# PASO 8: Correr imagen
podman pull --os linux docker.io/egroupware/egroupware:23.1
podman run -d \
--os linux \
--network $P3ASORC_GRUPO_NETWORK_NAME \
--name db-egroupware \
-e MARIADB_ROOT_PASSWORD=$P3ASORC_GRUPO_DB_ROOT_PASS \
-e MARIADB_USER=egroupware \
-e MARIADB_PASSWORD=$P3ASORC_GRUPO_DB_USER_PASS \
-e MARIADB_DATABASE=egroupware \
mariadb:lts

sleep 40
podman run -d \
--os linux \

```

```

-p 8080:80 \
--network $P3ASORC_GRUPO_NETWORK_NAME \
--name grupo \
-e EGW_DB_HOST=db-egroupware \
-e EGW_DB_ROOT_USER=root \
-e EGW_DB_ROOT_PASS=$P3ASORC_GRUPO_DB_ROOT_PASS \
-e EGW_DB_NAME=egroupware \
-e EGW_DB_USER=egroupware \
-e EGW_DB_PASS=$P3ASORC_GRUPO_DB_USER_PASS \
docker.io/egroupware/egroupware:23.1

sysrc bhyve_enable="YES"
sysrc docker_enable="YES"
docker-machine create -d virtualbox default
docker run -p 80:8080 --name grupo egroupware/egroupware:23.1

#-----
# Servicio (no backtrack) Intento #4 (La que no vi)
#-----

pkg install -y sogo-mysql mod_php83 php83-extensions php83-pdo sope2 php83-
    pecl-memcached php83-pecl-memcache memcached
sysrc mysql_enable="YES"
sysrc apache24_enable="YES"
sysrc sogod_enable="YES"
sysrc memcached_enable="YES"

sed -i '' 's/#LoadModule proxy_module/LoadModule proxy_module/' /usr/local/
etc/apache24/httpd.conf
sed -i '' 's/#LoadModule proxy_http_module/LoadModule proxy_http_module/' /
usr/local/etc/apache24/httpd.conf
sed -i '' 's/#LoadModule headers_module/LoadModule headers_module/' /usr/
local/etc/apache24/httpd.conf
sed -i '' 's/#LoadModule alias_module/LoadModule alias_module/' /usr/local/
etc/apache24/httpd.conf

service memcached start
service mysql-server start
service apache24 start
service memcached restart
service apache24 restart
service mysql-server restart

# Ojito
cat /var/db/mysql/*.err | tail -n 20
rm -rf /var/db/mysql/*

mysql -u root << EOF
CREATE DATABASE IF NOT EXISTS $P3ASORC_GRUPO_DB;
CREATE USER IF NOT EXISTS '$P3ASORC_GRUPO_DB_USER'@'127.0.0.1' IDENTIFIED BY
    '$P3ASORC_GRUPO_DB_PASS';
GRANT ALL PRIVILEGES ON $P3ASORC_GRUPO_DB.* TO '$P3ASORC_GRUPO_DB_USER'@'
    '127.0.0.1';
FLUSH PRIVILEGES;

USE $P3ASORC_GRUPO_DB;
CREATE TABLE IF NOT EXISTS sogo_view (
    c_uid VARCHAR(20) PRIMARY KEY,
    c_name VARCHAR(20),

```

```

    c_password VARCHAR(32),
    c_cn VARCHAR(128),
    mail VARCHAR(128)
);
INSERT IGNORE INTO sogo_view VALUES ('admin', 'admin', MD5('1'), ,
    'Administrador', 'admin@local.host');
INSERT IGNORE INTO sogo_view VALUES ('$P3ASORC_GRUPO_DB_USER', ,
    '$P3ASORC_GRUPO_DB_USER', MD5('$P3ASORC_GRUPO_DB_PASS'), ,
    '$P3ASORC_GRUPO_DB_USER', '$P3ASORC_GRUPO_DB_USER@bsd.asorc.org');
EOF

cat << EOF > $P3ASORC_GRUPO_CONFIG1
{
    SOGoProfileURL = "mysql://$P3ASORC_GRUPO_DB_USER:
        $P3ASORC_GRUPO_DB_PASS@127.0.0.1:3306/$P3ASORC_GRUPO_DB/sogo_user_profile
        ";
    OCSFolderInfoURL = "mysql://$P3ASORC_GRUPO_DB_USER:
        $P3ASORC_GRUPO_DB_PASS@127.0.0.1:3306/$P3ASORC_GRUPO_DB/sogo_folder_info"
        ;
    OCSSessionsFolderURL = "mysql://$P3ASORC_GRUPO_DB_USER:
        $P3ASORC_GRUPO_DB_PASS@127.0.0.1:3306/$P3ASORC_GRUPO_DB/
        sogo_sessions_folder";
    SOGoPageTitle = SOGo;
    SOGoLanguage = "Spanish";
    SOGoTimeZone = "Europe/Madrid";

    WOPort =
        "$P3ASORC_GRUPO_INTERPHONE_NUMBERLESS_INTERFACES_ATTENDERS_AND_LISTENERS";
    WLogFile = /var/log/sogo/sogo.log;

    SOGoUserSources = (
        {
            type = sql;
            id = directory;
            viewURL = "mysql://$P3ASORC_GRUPO_DB_USER:$P3ASORC_GRUPO_DB_PASS@127
                .0.0.1:3306/$P3ASORC_GRUPO_DB/sogo_view";
            canAuthenticate = YES;
            isAddressBook = YES;
            userPasswordAlgorithm = md5;
        }
    );
}
EOF

cat << EOF > $P3ASORC_GRUPO_CONFIG
Alias /SOGO.woa/WebServerResources/ $P3ASORC_GRUPO_FDIR/
Alias /SOGO/WebServerResources/ $P3ASORC_GRUPO_FDIR/

<Directory $P3ASORC_GRUPO_FDIR/>
    AllowOverride None
    Require all granted
</Directory>

ProxyRequests Off
SetEnv proxy-nokeepalive 1
ProxyPreserveHost On

```

```

ProxyPass /SOGO http://
$P3ASORC_GRUPO_INTERPHONE_NUMBERLESS_INTERFACES_ATTENDERS_AND_LISTENERS/
SOGO retry=0
ProxyPassReverse /SOGO http://
$P3ASORC_GRUPO_INTERPHONE_NUMBERLESS_INTERFACES_ATTENDERS_AND_LISTENERS/
SOGO
EOF

chmod -R 755 /usr/local/GNUstep/Local/Library/SOGO/WebServerResources/

service sogod restart
service apache24 restart

#-----
# Valida servicio
#-----
service apache24 status
service mysql-server status
service sogod status
sockstat -l | grep -E 'sogod'

#-----
# Extraer logs, configs e historial
#-----
rm -rf $P3ASORC_MEMORIA
mkdir -p $P3ASORC_MEMORIA
mkdir -p $P3ASORC_CONFIG

cp $P3ASORC_GRUPO_CONFIG1 $P3ASORC_CONFIG/sogo.conf
cp $P3ASORC_GRUPO_CONFIG $P3ASORC_CONFIG/apache_sogo.conf
cp /usr/local/etc/apache24/httpd.conf $P3ASORC_CONFIG/httpd.conf
cp $P3ASORC_BSD_RC $P3ASORC_CONFIG/rc.conf

touch $P3ASORC_LOG
echo "##### SOGO LOG #####" >> $P3ASORC_LOG
cat /var/log/sogo/sogo.log >> $P3ASORC_LOG
echo -e "\n##### APACHE ERROR LOG #####" >> $P3ASORC_LOG
cat /var/log/httpd-error.log >> $P3ASORC_LOG
echo -e "\n##### MYSQL ERROR LOG #####" >> $P3ASORC_LOG
cat /var/db/mysql/*.err >> $P3ASORC_LOG

history > $P3ASORC_HISTORIAL
chmod 777 -R $P3ASORC_MEMORIA
tree $P3ASORC_MEMORIA

#-----
# Comprobación desde host
#-----
firefox http://192.168.25.11/SOGO
firefox http://bsd.asorc.org/SOGO

```

Fragmento de código 8: Configuración y comprobación de este servicio en ● FreeBSD 14.3.

### 3.2.1. Prestaciones y características del servicio

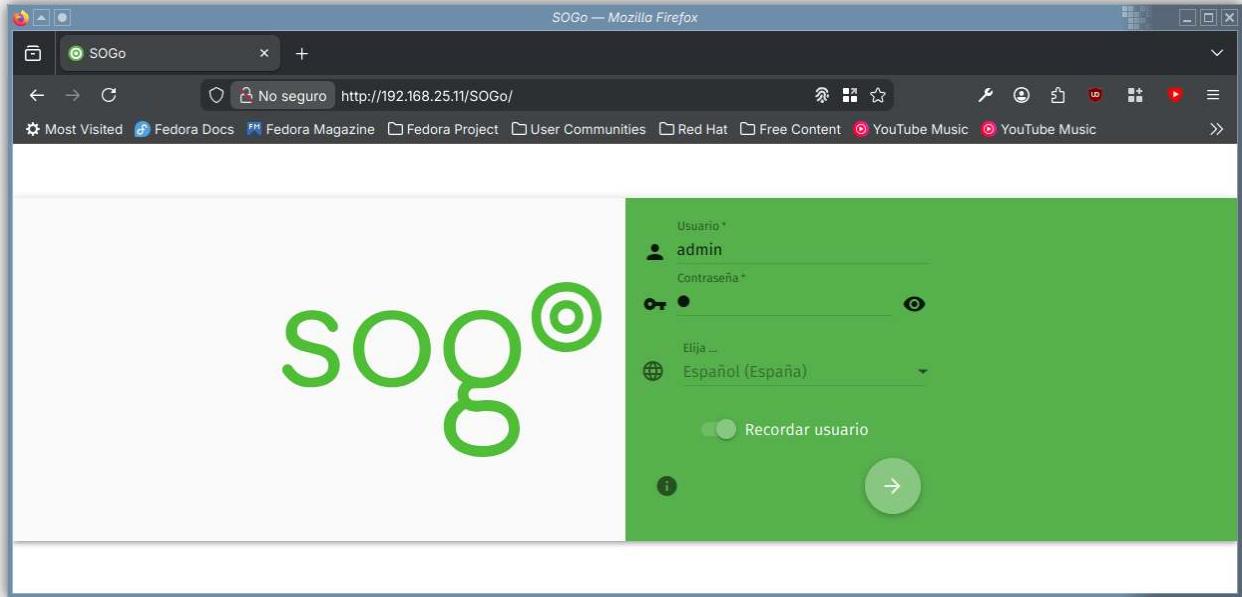


Figura 14: El servicio SOGo trae una interfaz web al destino de Material Design.

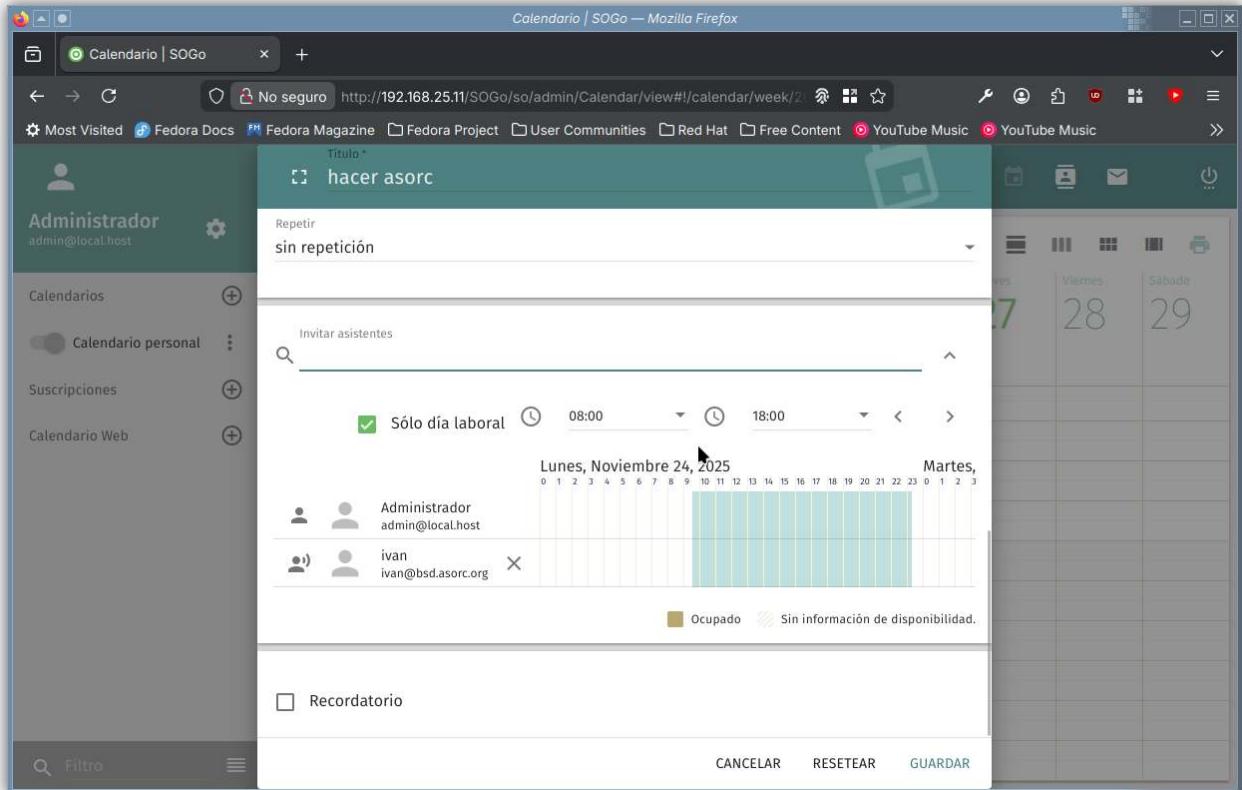


Figura 15: El servicio SOGo posibilita la suscripción de eventos en calendario, consultando su disponibilidad previamente. En este caso el administrador invita a ivan a «hacer asorc».

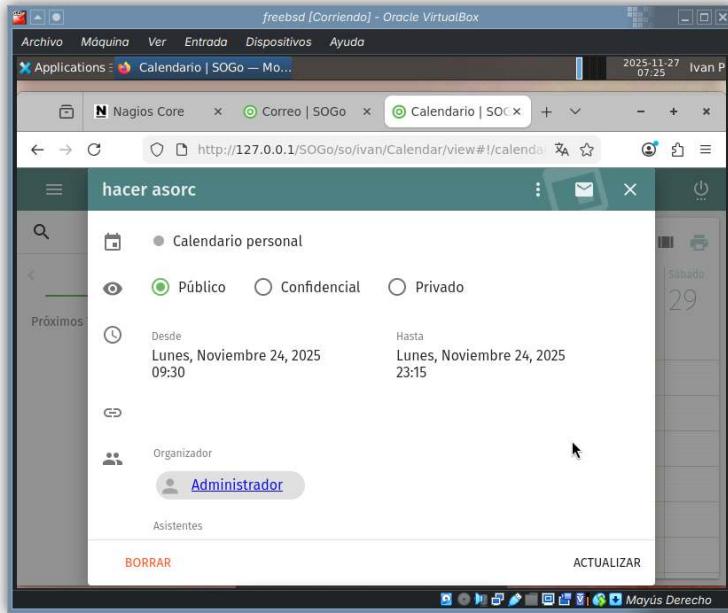


Figura 16: El servicio SOGo comparte el evento a usuarios de interés. Desde la perspectiva se examina el evento.

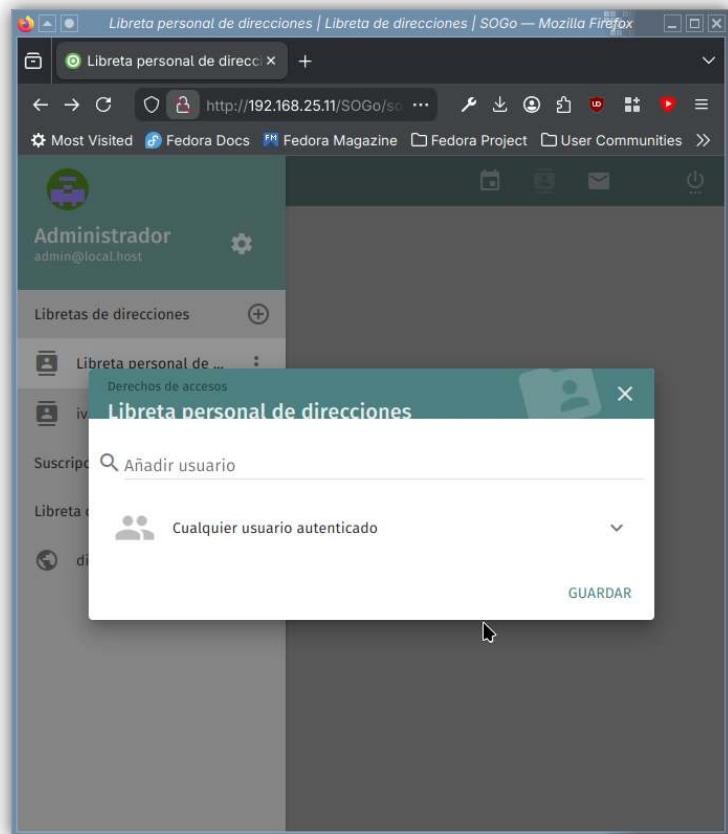


Figura 17: El servicio SOGo permite compartir agendas de contactos entre los usuarios de la plataforma.

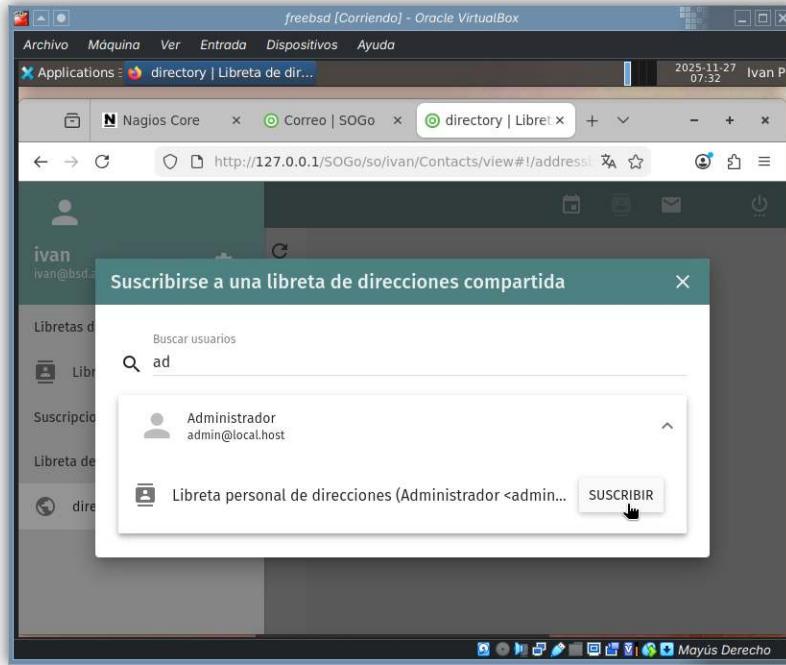


Figura 18: El servicio SOGo da libre albedrío de adoptar la agenda de contactos si es compartida.

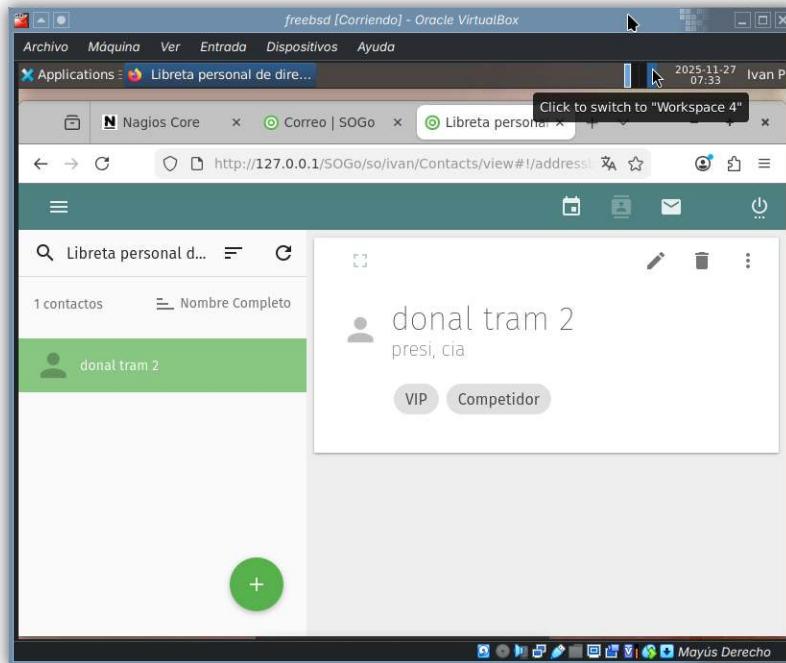


Figura 19: El servicio SOGo tiene previsto ver el contenido de un contacto, y con opciones de campos personalizados.

Además SOGo provee la posibilidad de una integración por correo. Este debe configurarse manualmente para su uso.

### 3.3. Microsoft Windows Server 2025 con SmarterMail

Enlace: <https://www.smartertools.com/smartermail/business-email-server>

Esta pieza de SW es un *freemium* que ofrece la posibilidad de usarlo sin tanto insistir con una prueba gratuita. Si bien no pide registro, invita a ello para extender funcionalidades de la plataforma. En este caso, prefiero reflejar en esta memoria la autenticidad del trabajo haciendo un paso a paso a base de pantallazos:

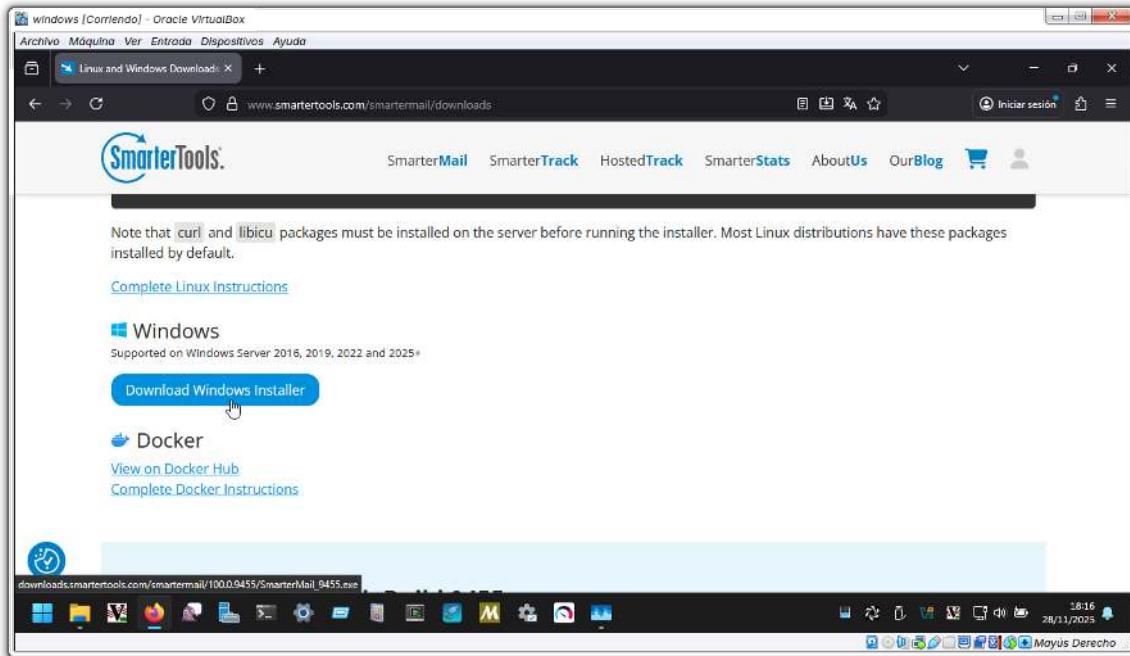


Figura 20: Descarga de SmarterMail en Microsoft Windows Server 2025.

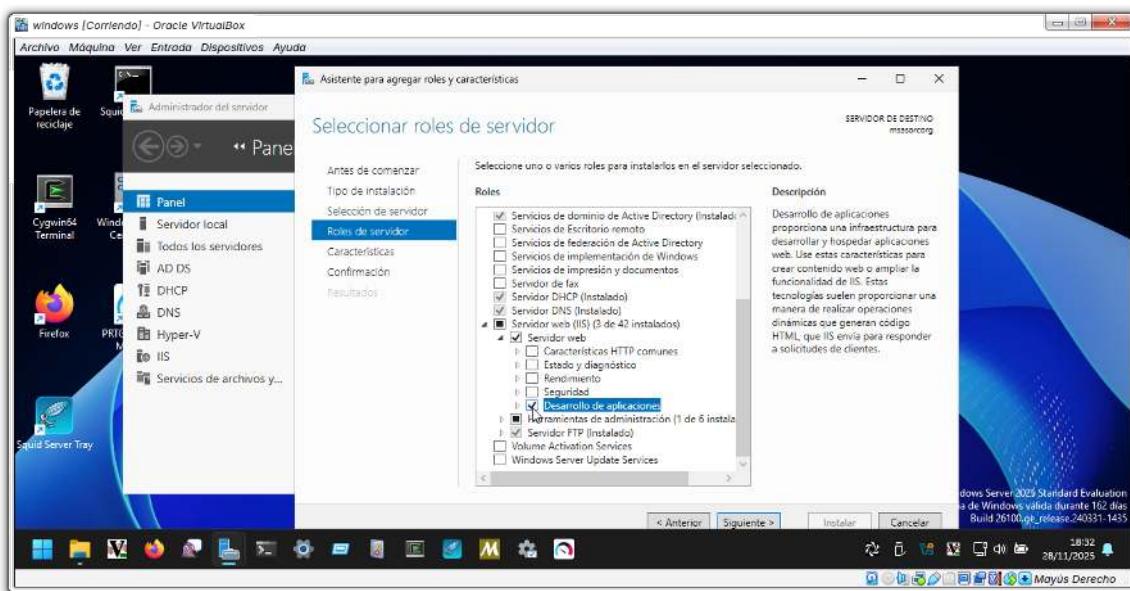


Figura 21: Mientras se descarga, es necesario agregar esta característica en Microsoft Windows Server 2025 desde la administración de servidor.

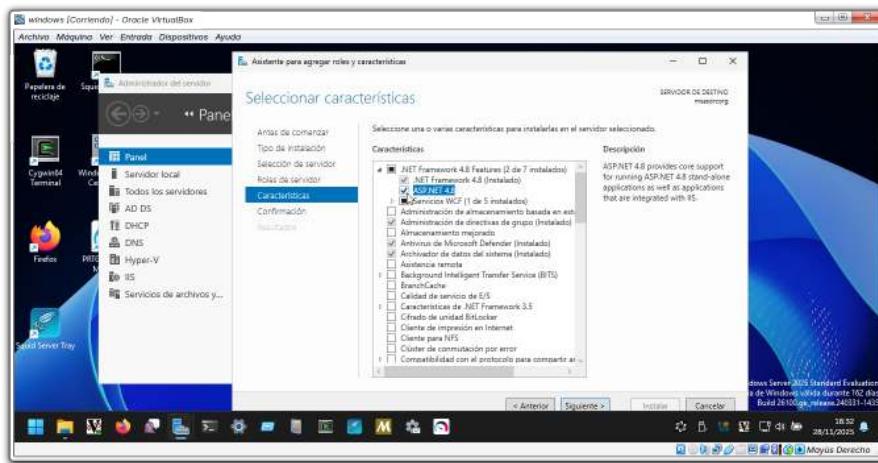


Figura 22: Habilitar ASP.NET 4.8 Framework en Microsoft Windows Server 2025 y esperar.

Todo ello puede simplificarse con `Install-WindowsFeature Web-Server, Web-Asp-Net45, NET-Framework-45-ASPNET`.

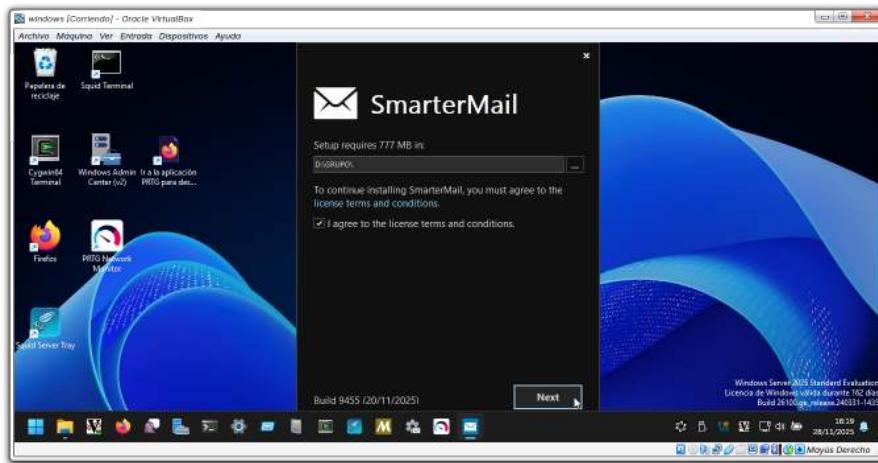


Figura 23: Asignar ubicación de instalación de SmarterMail.

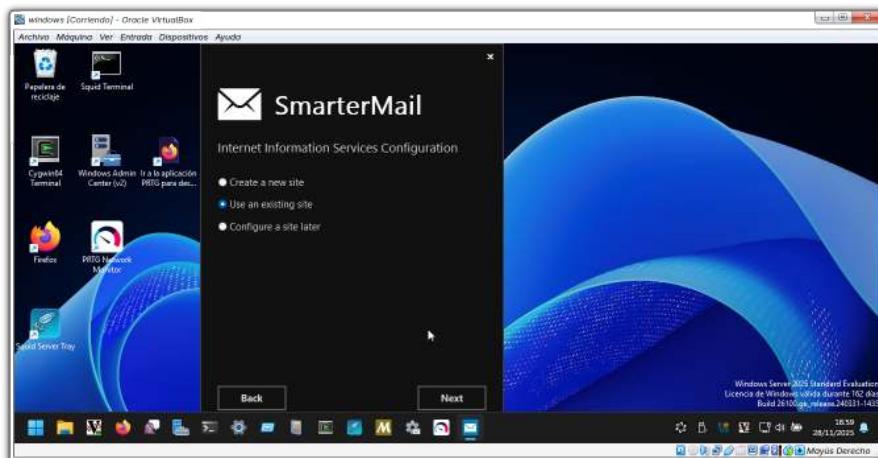


Figura 24: Configuración de parámetros en SmarterMail.

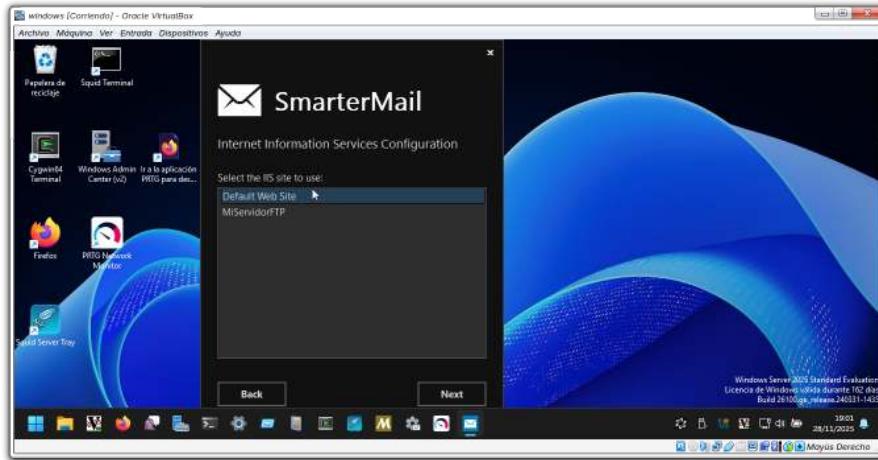


Figura 25: Usar el sitio por defecto de IIS SmarterMail. **Nota:** Se recomienda crear uno nuevo desde IIS, no usando este menú. Cosa la cual se hace (pero por cuestiones de tiempo aquí no se muestra) con el nombre «GRUPO» con escuchas al puerto 9999.

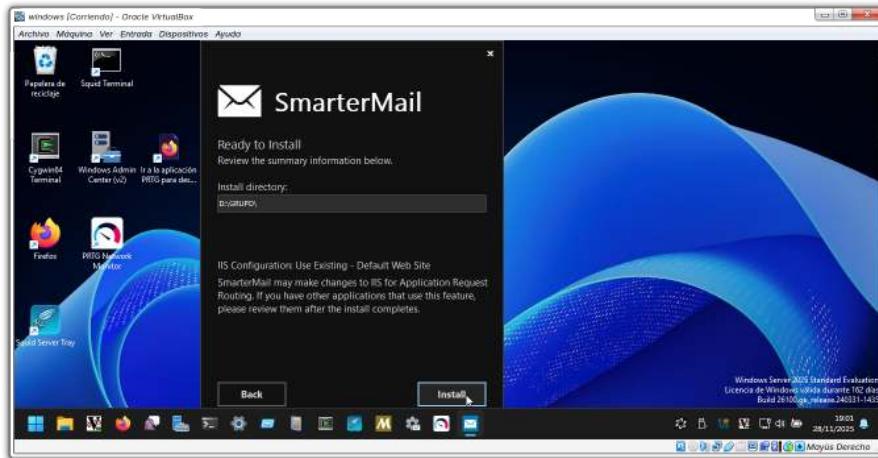


Figura 26: Pantalla de confirmación de instalación de SmarterMail.

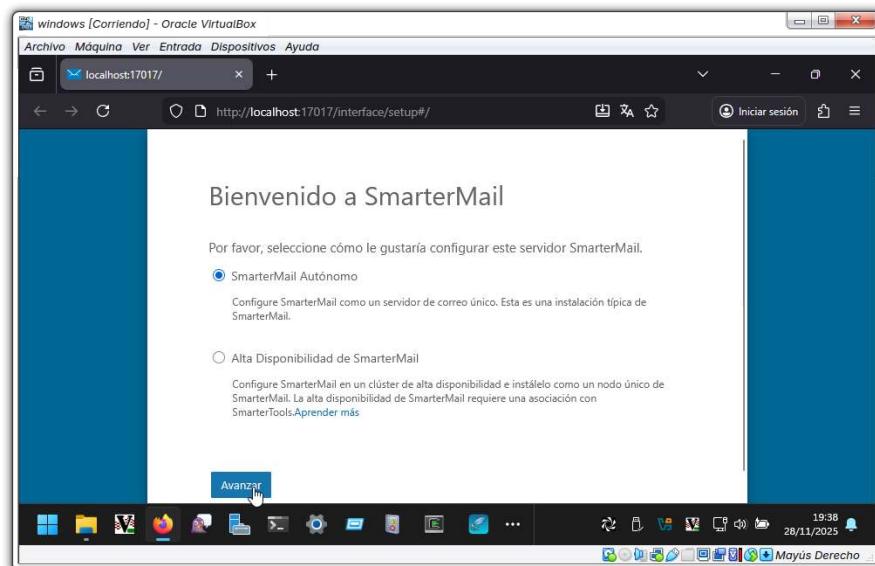


Figura 27: Pantalla de configuración inicial de SmarterMail.

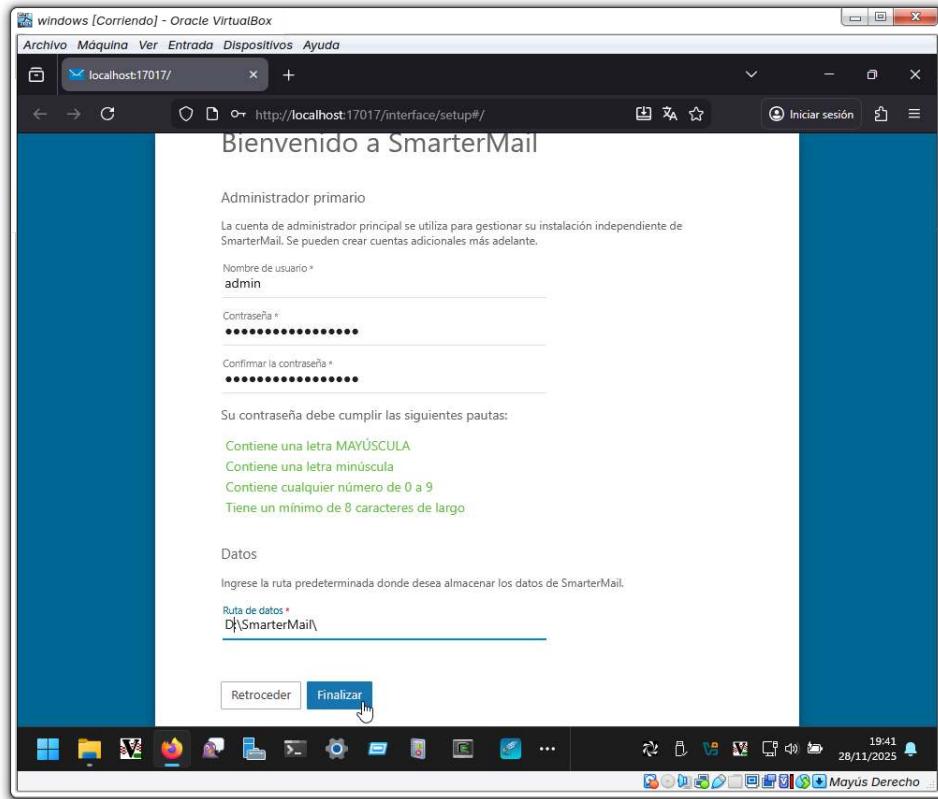


Figura 28: Pantalla para la creación del perfil administrador en SmarterMail.

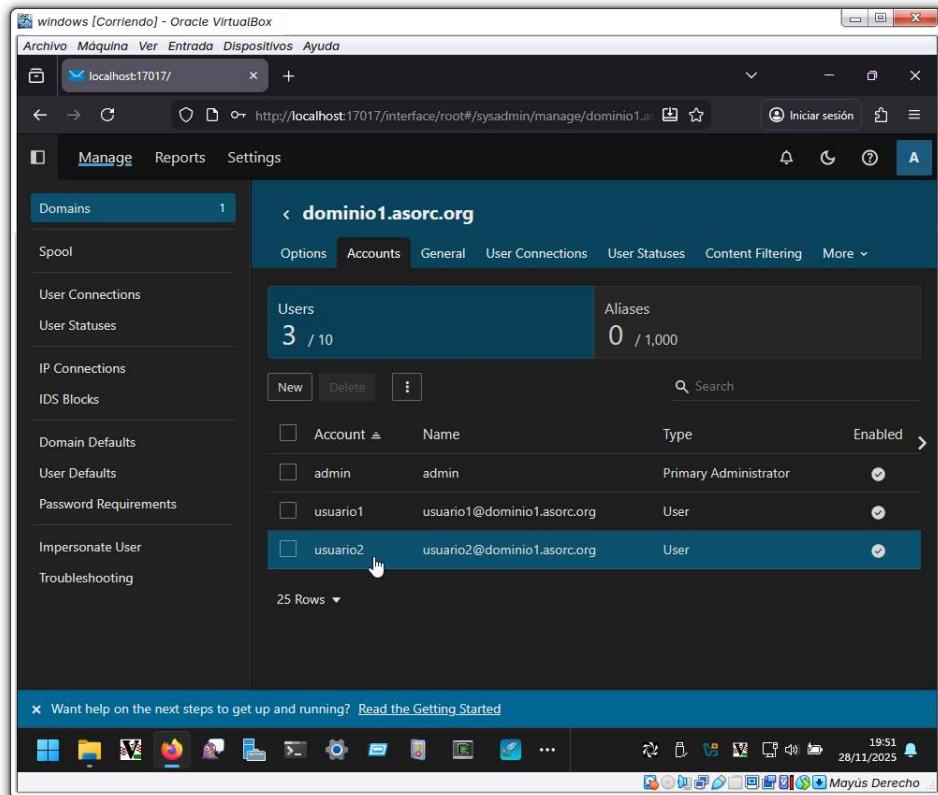


Figura 29: Desde la perspectiva de administrador en SmarterMail es posible modificar usuarios.

### 3.3.1. Características, justificación y registros del servicio propuestos

Pese a lo que pueda aparentar su nombre, SmarterMail no solo nos abstrae de configurar un correo sino que toda una plataforma colaborativa completa para el mundo empresarial. Pues ofrece lo típico de los *groupware* vistos anteriormente: una carpeta de archivos, tareas, notas, calendario. Además de otras integraciones con servicios de terceros como lo son con el acceso a LLM en línea, Google Workspace, compartición de archivos usando servidores de archivos típicos como NFS...

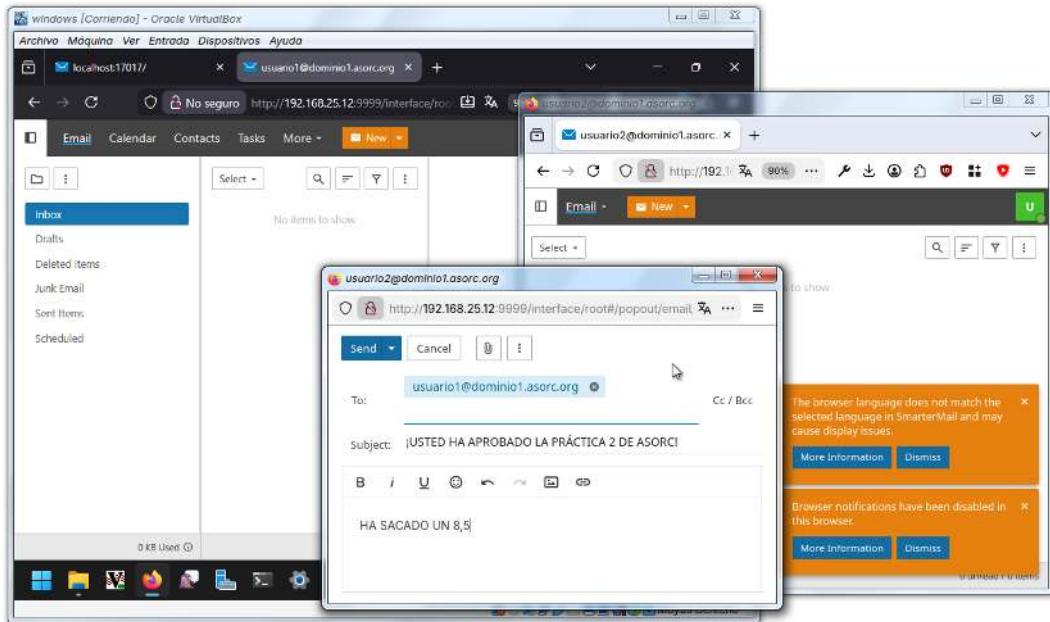


Figura 30: Envío de correos entre usuarios de la plataforma en SmarterMail (Servicio albergado en Microsoft Windows Server 2025).

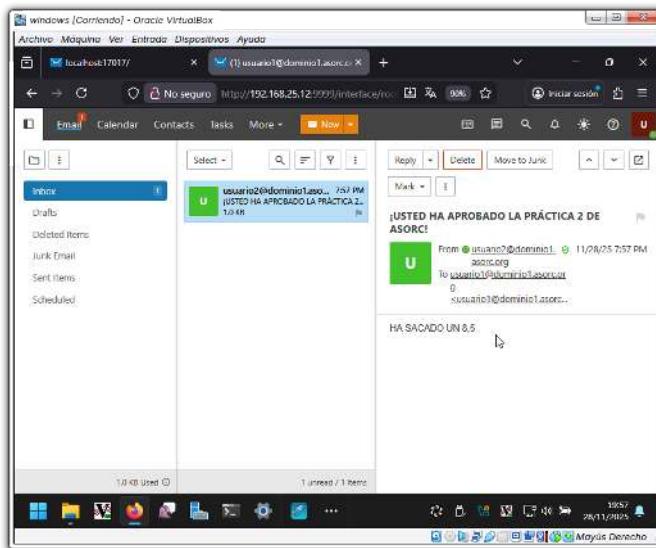


Figura 31: Visualización de la correcta recepción del correo enviado por otro usuario en SmarterMail.

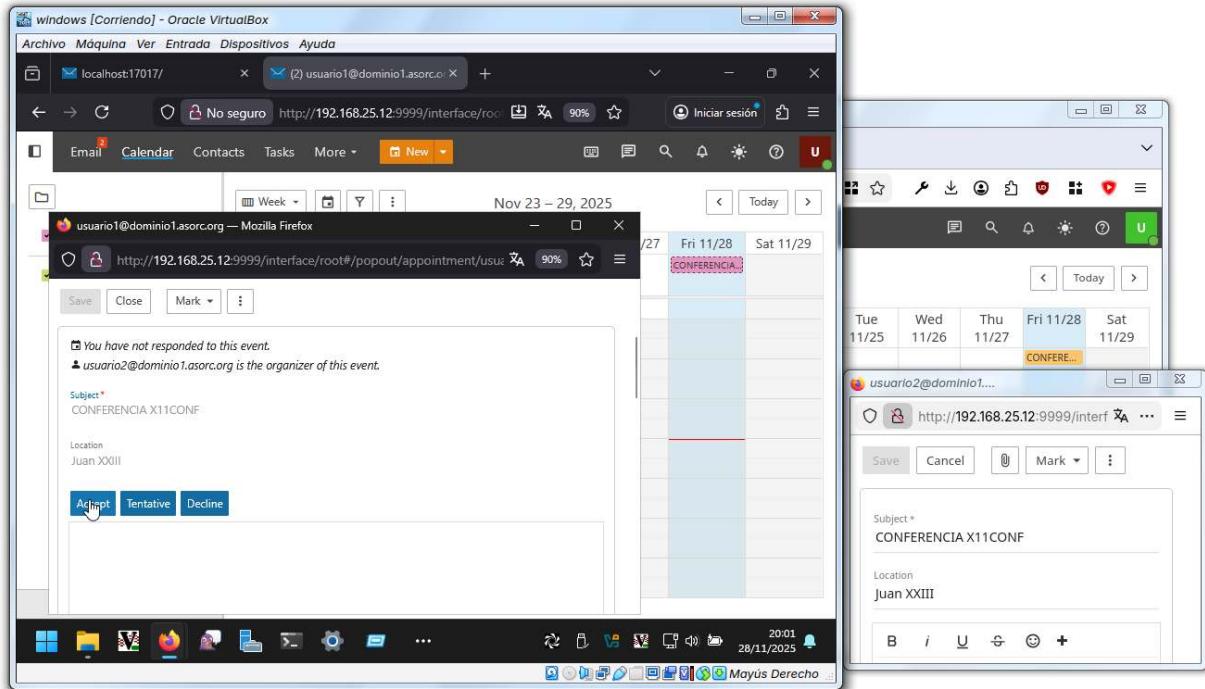


Figura 32: Es posible invitar a usuarios y consultar su disponibilidad para eventos del calendario en SmarterMail.

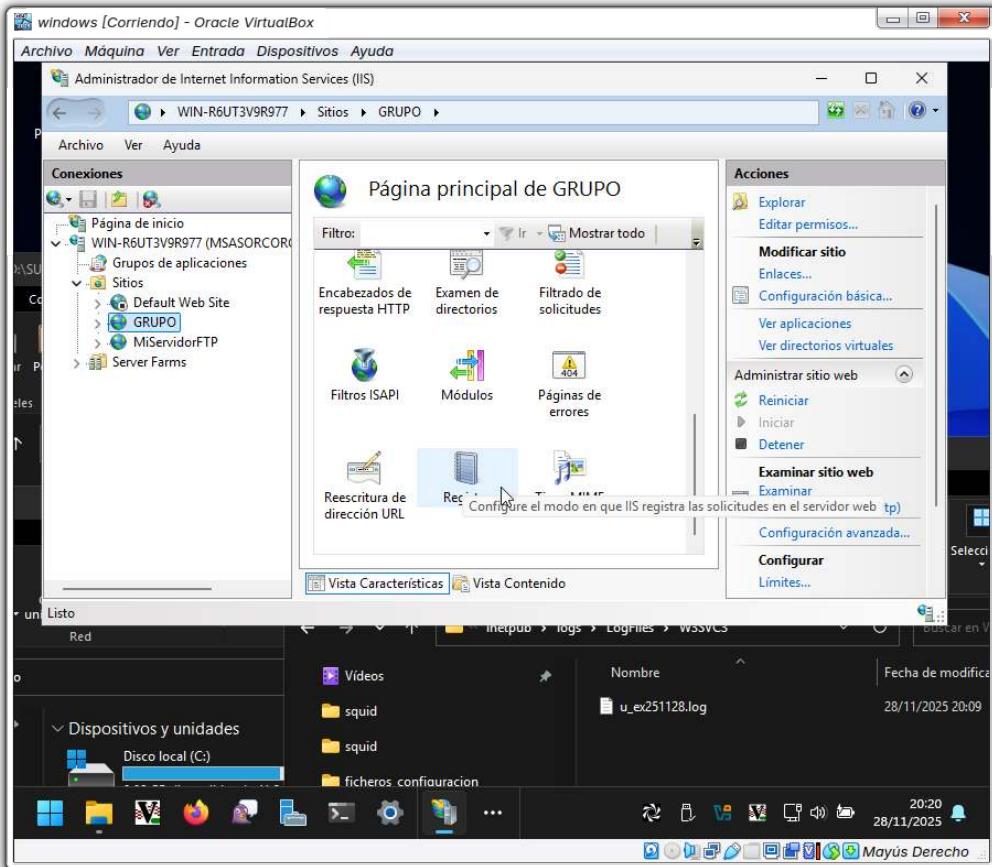


Figura 33: Extracción de *logs* a través de IIS para SmarterMail.

## 4. Proxy Caché (con Squid)

«*Proxy Cache (Squid): Se instalará un servidor proxy de manera transparente, es decir, sustituyendo al router como puerta de enlace predeterminada de los equipos de red. Se configurará la caché y se realizará restricción por páginas web y contenidos.*»

---

El **Proxy Cache** es un servidor intermediario que actúa como puerta de enlace entre la red local de una organización e Internet. Su función principal es interceptar las peticiones web de los clientes para mejorar el rendimiento mediante el almacenamiento en caché y reforzar la seguridad mediante el filtrado de contenidos.

El servicio opera interceptando el tráfico HTTP/HTTPS en la capa de aplicación:

- **El proxy (Squid):** Recibe la petición del cliente. Si el recurso está en su memoria (*hit*), lo devuelve inmediatamente ahorrando ancho de banda. Si no (*miss*), lo solicita al servidor de origen, lo almacena y lo entrega al cliente.
- **Modo transparente:** A diferencia del modo explícito, el cliente no necesita configuración; el tráfico se redirige a nivel de red (enrutamiento) hacia el puerto del proxy (típicamente 3128).

En cuanto a sus orígenes, la tecnología proxy deriva de los primeros cortafuegos y pasarelas de aplicación de ARPANET. Sin embargo, **Squid**, el SW de referencia en entornos UNIX, tiene su origen en el **Harvest Cache Daemon** desarrollado en la Universidad de Colorado en los años 90. El proyecto Squid nació oficialmente en **1996** financiado por la NSF (*National Science Foundation*), convirtiéndose en el estándar de facto para caché web de código abierto.

Sobre uso, es una herramienta crítica en la administración de redes corporativas y educativas. Se utiliza para reducir la latencia de navegación, optimizar el consumo de datos y, fundamentalmente, para aplicar políticas de uso aceptable mediante **ACLs** (*Access Control Lists*), bloqueando sitios maliciosos o no productivos.

En el desarrollo de esta práctica, la implementación del servicio Proxy Cache tiene un triple objetivo técnico:

1. **Configuración transparente:** Configurar el servidor para actuar como puerta de enlace predeterminada (*gateway*), interceptando el tráfico sin intervención del usuario final.
2. **Optimización:** Habilitar y dimensionar el almacenamiento en caché para acelerar la entrega de contenido web estático.
3. **Control de acceso:** Implementar reglas de restricción efectivas que filtren el acceso a determinados dominios y tipos de contenido, asegurando el cumplimiento de las políticas de red.

### 4.1. Debian GNU/Linux 13 (Trixie)

```
#!/bin/sh
su -
#-----
# Variables de entorno
```

```

#-----
P3ASORC_SERVICIO=proxy
P3ASORC_SISTEMA=linux

# NO TOCAR
P3ASORC_MEMORIA=/home/ivan/SUPERMEMORIA/$P3ASORC_SERVICIO
P3ASORC_CONFIG=$P3ASORC_MEMORIA/ficheros_configuracion
P3ASORC_LOG=$P3ASORC_MEMORIA/$P3ASORC_SISTEMA.log
P3ASORC_HISTORIAL=$P3ASORC_MEMORIA/history$P3ASORC_SISTEMA.txt

# ---
P3ASORC_PROXY_CONFIG=/etc/squid/squid.conf
P3ASORC_PROXY_LOG1=/var/log/squid/access.log
P3ASORC_PROXY_LOG2=/var/log/squid/cache.log

#-----
# Servicio
#-----
# PASO 0: Limpieza
systemctl disable squid
systemctl stop squid
rm -rf $P3ASORC_PROXY_CONFIG $P3ASORC_PROXY_LOG2 $P3ASORC_PROXY_LOG1
apt remove -y squid

# PASO 1: Instalar y copiar configuración por defecto
apt install -y squid
cp $P3ASORC_PROXY_CONFIG $P3ASORC_PROXY_CONFIG.bak

# PASO 2: Configurar servicio
sysctl -w net.ipv4.ip_forward=1
cat << EOF > $P3ASORC_PROXY_CONFIG
http_port 3128
http_port 3129 intercept
cache_dir ufs /var/spool/squid 4096 16 256
coredump_dir /var/spool/squid
acl mi_red src 192.168.25.0/24
acl sitios_prohibidos dstdomain .facebook.com .instagram.com .twitter.com .
    laliga.com
acl contenido_prohibido urlpath_regex -i laliga adult porn gambling
acl SSL_ports port 443
acl Safe_ports port 80 21 443 70 210 1025-65535
acl CONNECT method CONNECT
http_access allow localhost manager
http_access deny manager
http_access deny !Safe_ports
http_access deny CONNECT !SSL_ports
http_access deny sitios_prohibidos
http_access deny contenido_prohibido
http_access allow mi_red
http_access allow localhost
http_access deny all
visible_hostname debian.asorc.org
dns_nameservers 1.1.1.1
debug_options ALL,1 11,2 33,2
EOF

# PASO 3: Activación y arranque el servicio
systemctl restart squid

```

```

systemctl enable squid

#-----
# Valida servicio
#-----
systemctl status squid
netstat -tunelp | grep -E 'squid'

#-----
# Extraer logs, configs e historial
#-----
rm -rf $P3ASORC_MEMORIA
mkdir -p $P3ASORC_MEMORIA
mkdir -p $P3ASORC_CONFIG

cp $P3ASORC_PROXY_CONFIG $P3ASORC_CONFIG
cp $P3ASORC_PROXY_LOG1 $P3ASORC_LOG
cat $P3ASORC_PROXY_LOG2 >> $P3ASORC_LOG

history > $P3ASORC_HISTORIAL
chmod 777 -R $P3ASORC_MEMORIA

tree $P3ASORC_MEMORIA

#-----
# Comprobación
#-----
HTTPS_PROXY=192.168.25.10:3128 HTTP_PROXY=192.168.25.10:3128 curl -I https://www.facebook.com/
HTTPS_PROXY=192.168.25.10:3128 HTTP_PROXY=192.168.25.10:3128 curl -I https://nitter.net/

```

Fragmento de código 9: Configuración y comprobación de este servicio en  Debian GNU/Linux 13 (Trixie).

## 4.2. ● FreeBSD 14.3

```

#!/bin/sh
su -

#-----
# Variables de entorno
#-----
P3ASORC_SERVICIO=proxy
P3ASORC_SISTEMA=unix

# NO TOCAR
P3ASORC_MEMORIA=/home/ivan/SUPERMEMORIA/$P3ASORC_SERVICIO
P3ASORC_CONFIG=$P3ASORC_MEMORIA/ficheros_configuracion
P3ASORC_LOG=$P3ASORC_MEMORIA/$P3ASORC_SISTEMA.log
P3ASORC_HISTORIAL=$P3ASORC_MEMORIA/history$P3ASORC_SISTEMA.txt
P3ASORC_BSD_RC=/etc/rc.conf

# ---
P3ASORC_PROXY_CONFIG=/usr/local/etc/squid/squid.conf
P3ASORC_PROXY_LOG1=/var/log/squid/access.log
P3ASORC_PROXY_LOG2=/var/log/squid/cache.log

```

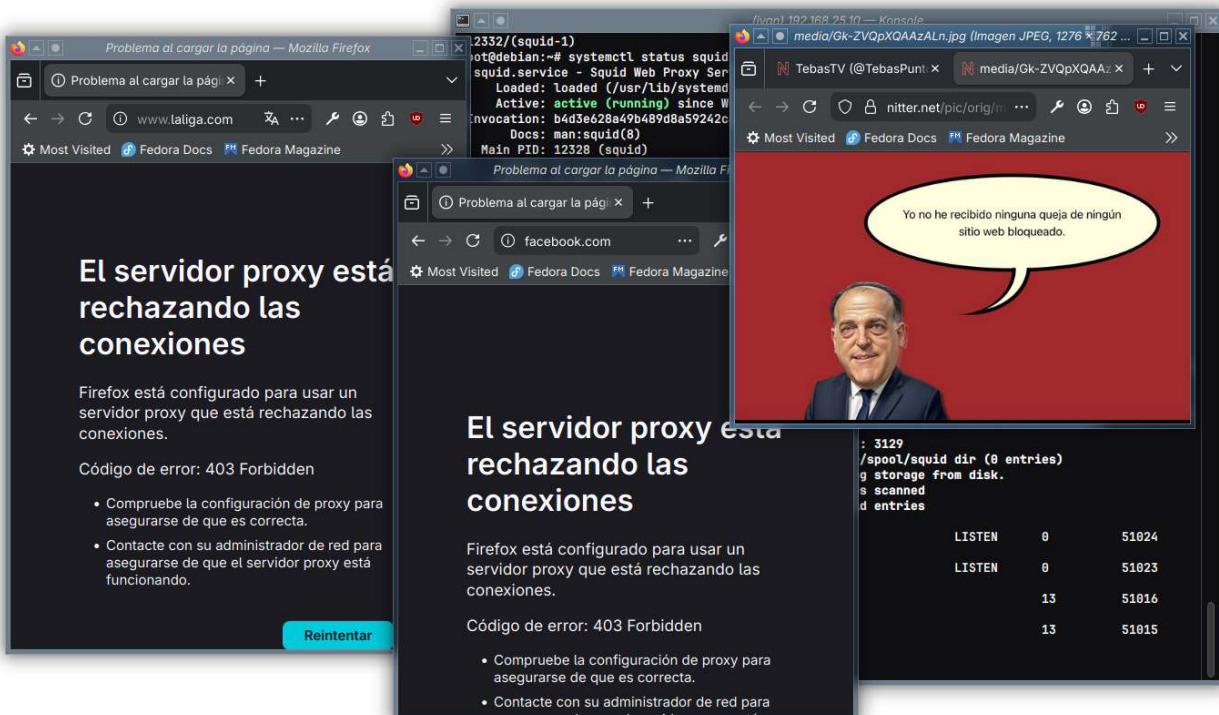


Figura 34: Comprobación del servicio proxy en Debian GNU/Linux 13 (Trixie) sobre ACLs desde el navegador Firefox en el *host*. Se aprecia la denegación de servicio por parte de Squid (*mensaje Access Denied*) en los portales de LaLiga y Facebook, mientras que la navegación hacia Nitter es exitosa.

```
#-----
# Servicio
#-----
# PASO 0: Limpieza
service squid stop
sysrc -x squid_enable
rm -rf $P3ASORC_PROXY_CONFIG $P3ASORC_PROXY_LOG2 $P3ASORC_PROXY_LOG1
pkg delete -y squid

# PASO 1: Instalar y copiar configuración por defecto
pkg install -y squid
if [ -f $P3ASORC_PROXY_CONFIG ]; then
    cp $P3ASORC_PROXY_CONFIG $P3ASORC_PROXY_CONFIG.bak
fi

# PASO 2: Configurar servicio
sysctl net.inet.ip.forwarding=1
sysrc gateway_enable="YES"

# Crear archivo de configuración
cat << EOF > $P3ASORC_PROXY_CONFIG
http_port 3128
http_port 3129 intercept
cache_dir ufs /var/squid/cache 4096 16 256
coredump_dir /var/squid/cache
acl mi_red src 192.168.25.0/24
EOF
```

```

acl sitios_prohibidos dstdomain .facebook.com .instagram.com .twitter.com .
    laliga.com
acl contenido_prohibido urlpath_regex -i laliga adult porn gambling
acl SSL_ports port 443
acl Safe_ports port 80 21 443 70 210 1025-65535
acl CONNECT method CONNECT
http_access allow localhost manager
http_access deny manager
http_access deny !Safe_ports
http_access deny CONNECT !SSL_ports
http_access deny sitios_prohibidos
http_access deny contenido_prohibido
http_access allow mi_red
http_access allow localhost
http_access deny all
visible_hostname freebsd.asorc.org
dns_nameservers 1.1.1.1
debug_options ALL,1 11,2 33,2
EOF

# PASO 3: Activación y arranque el servicio
squid -z
sysrc squid_enable="YES"
service squid start

#-----
# Valida servicio
#-----
service squid status
sockstat | grep squid

#-----
# Extraer logs, configs e historial
#-----
rm -rf $P3ASORC_MEMORIA
mkdir -p $P3ASORC_MEMORIA
mkdir -p $P3ASORC_CONFIG

cp $P3ASORC_PROXY_CONFIG $P3ASORC_CONFIG
cat $P3ASORC_PROXY_LOG1 > $P3ASORC_LOG
cat $P3ASORC_PROXY_LOG2 >> $P3ASORC_LOG
cp $P3ASORC_BSD_RC $P3ASORC_CONFIG

history > $P3ASORC_HISTORIAL
chmod 777 -R $P3ASORC_MEMORIA

tree $P3ASORC_MEMORIA

#-----
# Comprobación desde host
#-----
HTTPS_PROXY=192.168.25.11:3128 HTTP_PROXY=192.168.25.11:3128 curl -I https
://www.instagram.com/
HTTPS_PROXY=192.168.25.11:3128 HTTP_PROXY=192.168.25.11:3128 curl -I https
://nitter.net/

```

Fragmento de código 10: Configuración y comprobación de este servicio en ● FreeBSD 14.3.

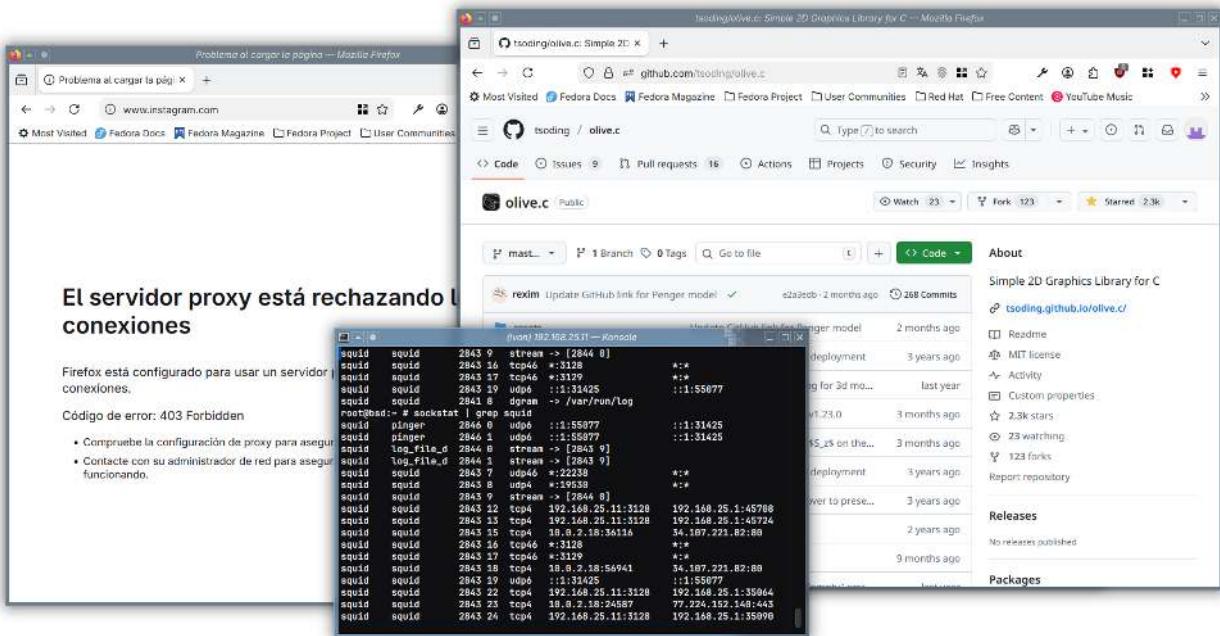


Figura 35: Comprobación del servicio proxy en  $\bullet$  FreeBSD 14.3 sobre ACLs desde el navegador Firefox en el *host*. Se aprecia la denegación de servicio por parte de Squid (mensaje *Access Denied*) en los portales de Instagram, mientras que la navegación hacia GitHub es exitosa.

## 4.3. Microsoft Windows Server 2025

Inicialmente, se intentó abordar la instalación de este servicio utilizando la versión de Squid portada a través de la capa de compatibilidad **Cygwin**. Sin embargo, dicha implementación presentó múltiples incompatibilidades, especialmente con las directivas de intercepción transparente y la gestión de permisos en el sistema de archivos NTFS, resultando en un servicio inestable incapaz de arrancar consistentemente. Por consiguiente, se optó por la utilización de un paquete MSI precompilado (Diladele) optimizado para sistemas Windows.

### 4.3.1. Instalación y configuración

- Descarga:** Se obtuvo el instalador MSI directamente desde el repositorio de paquetes: <https://packages.diladele.com/squid/4.14/squid.msi>.
- Instalación:** Durante el asistente de instalación, se modificó la ruta por defecto para ubicar el servicio en D:\Squid, centralizando así el servicio en la unidad de datos.
- Acceso a la configuración:** Una vez instalado el servicio, la gestión del archivo squid.conf se realiza de forma gráfica. Se debe localizar el ícono del servicio en la **bandeja del sistema** (área de notificaciones), hacer clic derecho y seleccionar la opción «**Open Configuration**».

El archivo de configuración resultante, adaptado para permitir el tráfico de la red local y aplicar las ACLs solicitadas, tiene en su lista negra sitios tan importantes para el progreso humano como las redes sociales o laliga.

#### 4.3.2. Comprobación y ACLs

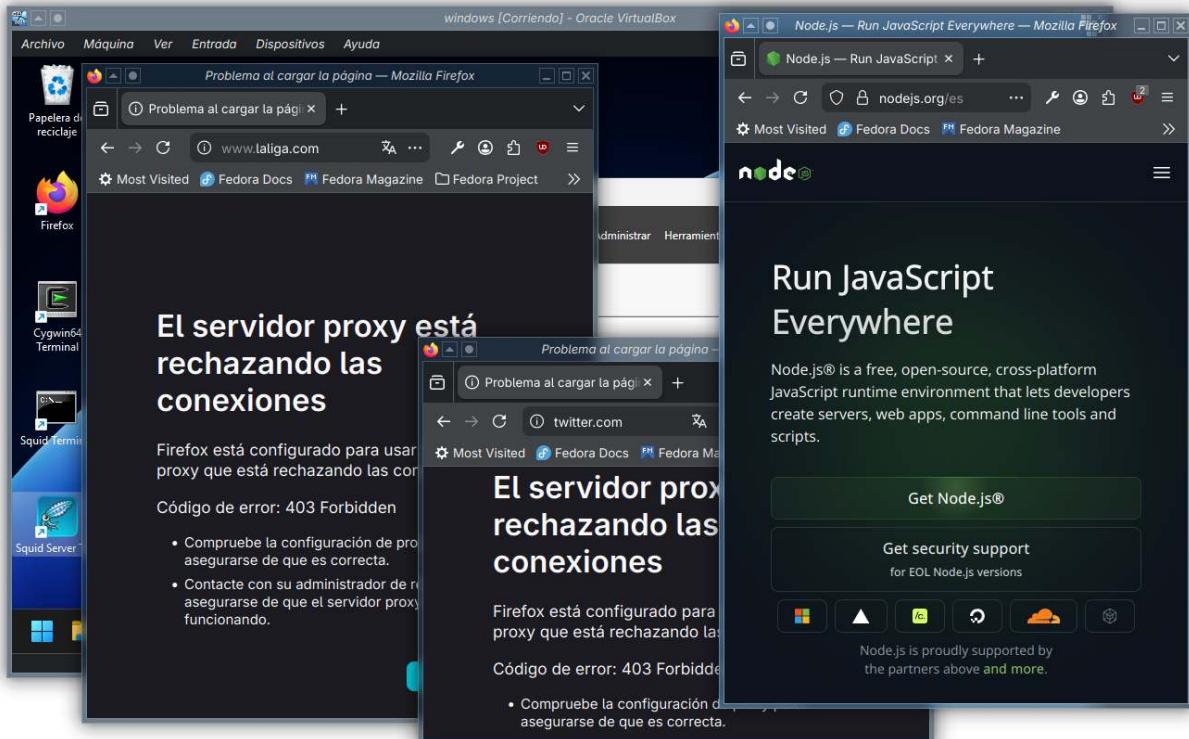


Figura 36: Comprobación del servicio proxy en Microsoft Windows Server 2025 sobre ACLs desde el navegador Firefox en el *host*. Se observa la denegación de servicio por parte de Squid (*mensaje Access Denied*) en los portales de LaLiga y Twitter, mientras que la navegación hacia NodeJS es exitosa.

```
#!/bin/sh
D:\Squid\bin\squid.exe -z
netsh interface portproxy add v4tov4 listenport=80 listenaddress
    =192.168.25.12 connectport=3129 connectaddress=127.0.0.1
netsh interface portproxy reset
netsh interface portproxy show all

#-----#
# Comprobación
#-----#
HTTPS_PROXY=192.168.25.12:3128 HTTP_PROXY=192.168.25.12:3128 curl -I https
://www.laliga.com
HTTPS_PROXY=192.168.25.12:3128 HTTP_PROXY=192.168.25.12:3128 curl -I https
://nodejs.org
```

Fragmento de código 11: Algunos de los comandos de utilidad empleados de prueba para este servicio en Microsoft Windows Server 2025.

Para los registros de actividad, se ha copiado el archivo generado en D:\Squid\var\logs\access.log y se ha renombrado a windows.csv para su inclusión en la entrega.

## 5. Rutado, cortafuegos y VPN

«Rutado, Cortafuegos, VPN: Se instalará uno de los tres servicios en cada sistema operativo de manera disjunta. Es decir, si en uno de ellos se instala Rutado, en ninguno de los otros se podrá utilizar»

---

Este punto de la práctica aborda la implementación de tres servicios de red fundamentales que, en entornos de producción, suelen converger en un único dispositivo de borde o perimetral (*Edge Device*). La integración de estas funciones permite gestionar el tráfico, asegurar las comunicaciones remotas y proteger la infraestructura interna simultáneamente.

El despliegue se basa en una arquitectura de **seguridad perimetral unificada**:

- **Router (Enrutamiento):** Actúa como el nodo central que interconecta dos o más redes lógicas (subredes) diferentes, tomando decisiones sobre hacia dónde enviar los paquetes de datos basándose en su dirección IP destino y la tabla de enrutamiento.
- **VPN (Red Privada Virtual):** Establece un túnel cifrado sobre una red pública (Internet) que permite a usuarios externos acceder a los recursos de la red interna como si estuvieran físicamente conectados a ella, garantizando la confidencialidad e integridad de la información.
- **Firewall (Cortafuegos):** Inspecciona y filtra el tráfico entrante y saliente según un conjunto de reglas predefinidas, permitiendo o denegando las conexiones para prevenir accesos no autorizados y ataques de red.

En cuanto a sus orígenes, el concepto de **router** se remonta a los primeros *gateways* de ARPANET (los IMP) en 1969. El **firewall** moderno surgió a finales de los 80 con la tecnología de filtrado de paquetes de DEC, evolucionando posteriormente hacia la inspección de estado (*Stateful Inspection*) patentada por Check Point en 1993. Por su parte, la tecnología **VPN** se popularizó a mediados de los 90 con el desarrollo del protocolo PPTP por un consorcio liderado por Microsoft y la posterior estandarización de IPsec, permitiendo la expansión segura de las redes corporativas a través de Internet.

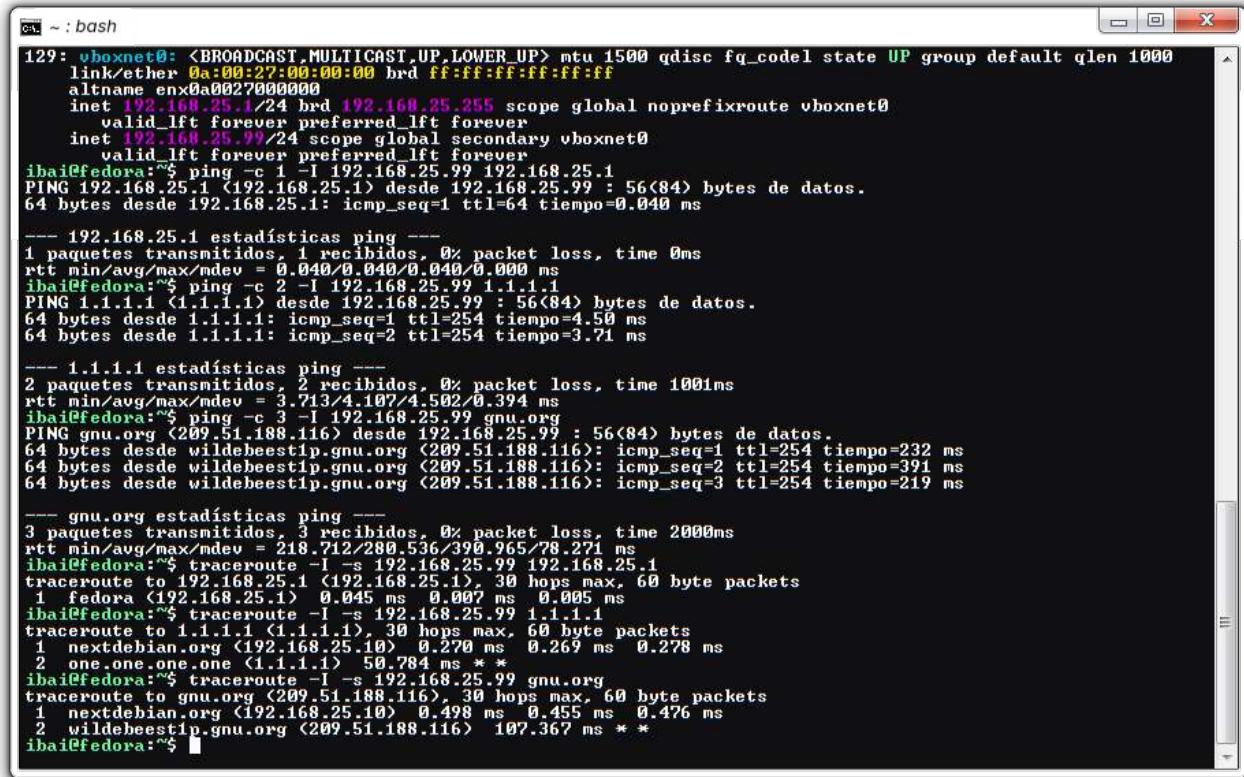
Sobre uso, la convergencia de estos tres servicios es el estándar en la administración de redes modernas. Permite optimizar el hardware y simplificar la gestión. Es la configuración típica tanto en pequeñas oficinas (SOHO) como en grandes corporaciones, donde un servidor o dispositivo dedicado gestiona la conectividad (enrutamiento), el teletrabajo seguro (VPN) y la defensa perimetral (Firewall). El uso de las VPN en el teletrabajo fueron muy recurridas durante el confinamiento a causa del covid ganando mayor importancia.

En el desarrollo de esta práctica, la implementación de este sistema integrado persigue los siguientes objetivos técnicos:

1. **Interconexión de redes:** Configurar el sistema para reenviar paquetes IPv4 entre dos interfaces de red distintas, habilitando la comunicación transparente entre equipos ubicados en subredes separadas.
2. **Acceso remoto seguro:** Desplegar un servidor VPN (empleando protocolos como WireGuard, OpenVPN o IPsec/L2TP) que autentique a clientes externos y les asigne una dirección IP virtual dentro de la red privada.

3. **Filtrado de tráfico:** Implementar políticas de seguridad mediante reglas de *iptables*, *nftables* o el *firewall* nativo del sistema operativo, restringiendo el acceso a servicios críticos y permitiendo únicamente el tráfico legítimo necesario.

## 5.1. Enrutado (*iptables*) en Debian GNU/Linux 13 (Trixie)



```

ct ~ : bash
129: vboxnet0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
link/ether 0a:00:27:00:00:00 brd ff:ff:ff:ff:ff:ff
  altname enx0a0027000000
  inet 192.168.25.1/24 brd 192.168.25.255 scope global noprefixroute vboxnet0
    valid_lft forever preferred_lft forever
  inet 192.168.25.99/24 scope global secondary vboxnet0
    valid_lft forever preferred_lft forever
ibai@fedora:~$ ping -c 1 -I 192.168.25.99 192.168.25.1
PING 192.168.25.1 (192.168.25.1) desde 192.168.25.99 : 56<84> bytes de datos.
64 bytes desde 192.168.25.1: icmp_seq=1 ttl=64 tiempo=0.040 ms

--- 192.168.25.1 estadísticas ping ---
1 paquetes transmitidos, 1 recibidos, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.040/0.040/0.040/0.000 ms
ibai@fedora:~$ ping -c 2 -I 192.168.25.99 1.1.1.1
PING 1.1.1.1 (1.1.1.1) desde 192.168.25.99 : 56<84> bytes de datos.
64 bytes desde 1.1.1.1: icmp_seq=1 ttl=254 tiempo=4.50 ms
64 bytes desde 1.1.1.1: icmp_seq=2 ttl=254 tiempo=3.71 ms

--- 1.1.1.1 estadísticas ping ---
2 paquetes transmitidos, 2 recibidos, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 3.713/4.107/4.502/0.394 ms
ibai@fedora:~$ ping -c 3 -I 192.168.25.99 gnu.org
PING gnu.org (209.51.188.116) desde 192.168.25.99 : 56<84> bytes de datos.
64 bytes desde wildebeestip.gnu.org (209.51.188.116): icmp_seq=1 ttl=254 tiempo=232 ms
64 bytes desde wildebeestip.gnu.org (209.51.188.116): icmp_seq=2 ttl=254 tiempo=391 ms
64 bytes desde wildebeestip.gnu.org (209.51.188.116): icmp_seq=3 ttl=254 tiempo=219 ms

--- gnu.org estadísticas ping ---
3 paquetes transmitidos, 3 recibidos, 0% packet loss, time 2000ms
rtt min/avg/max/mdev = 218.712/280.536/398.965/78.271 ms
ibai@fedora:~$ traceroute -I -s 192.168.25.99 192.168.25.1
traceroute to 192.168.25.1 (192.168.25.1), 30 hops max, 60 byte packets
  1  fedora (192.168.25.1)  0.045 ms  0.007 ms  0.005 ms
ibai@fedora:~$ traceroute -I -s 192.168.25.99 1.1.1.1
traceroute to 1.1.1.1 (1.1.1.1), 30 hops max, 60 byte packets
  1  nextdebian.org (192.168.25.10)  0.270 ms  0.269 ms  0.278 ms
  2  one.one.one (1.1.1.1)  50.784 ms  *  *
ibai@fedora:~$ traceroute -I -s 192.168.25.99 gnu.org
traceroute to gnu.org (209.51.188.116), 30 hops max, 60 byte packets
  1  nextdebian.org (192.168.25.10)  0.498 ms  0.455 ms  0.476 ms
  2  wildebeestip.gnu.org (209.51.188.116)  107.367 ms  *  *
ibai@fedora:~$ 

```

Figura 37: Demostración del esperado funcionamiento de este servicio en  Debian GNU/Linux 13 (Trixie).

```

#!/bin/sh
su -

#-----
# Variables de entorno
#-----
P3ASORC_SERVICIO=router_vpn_firewall
P3ASORC_SISTEMA=linux

# NO TOCAR
P3ASORC_MEMORIA=/home/ivan/SUPERMEMORIA/$P3ASORC_SERVICIO
P3ASORC_CONFIG=$P3ASORC_MEMORIA/ficheros_configuracion
P3ASORC_LOG=$P3ASORC_MEMORIA/$P3ASORC_SISTEMA.log
P3ASORC_HISTORIAL=$P3ASORC_MEMORIA/history$P3ASORC_SISTEMA.txt

# ---
P3ASORC_ROUTER_CONFIG=/etc/sysctl.conf
P3ASORC_ROUTER_ILAN=enp0s8
P3ASORC_ROUTER_IWAN=enp0s3

```

```

#-----#
# Servicio
#-----#
# Requisitos tener hecho el dhcp de a antes
# PASO 0: Instalar paquetes
apt install -y iptables-persistent

# PASO 1: Aplicar forwarding
cat << EOF > $P3ASORC_ROUTER_CONFIG
net.ipv4.ip_forward=1
EOF

# PASO 2: Usar NAT
# Habilitar NAT (Masquerade) en la interfaz de salida
iptables -t nat -A POSTROUTING -o $P3ASORC_ROUTER_IWAN -j MASQUERADE

# Permitir tráfico desde la LAN hacia INTERNET
iptables -A FORWARD -i $P3ASORC_ROUTER_ILAN -o $P3ASORC_ROUTER_IWAN -j
    ACCEPT

# Permitir que la respuesta de internet vuelva a entrar hacia la LAN
iptables -A FORWARD -i $P3ASORC_ROUTER_IWAN -o $P3ASORC_ROUTER_ILAN -m state
    --state RELATED,ESTABLISHED -j ACCEPT

# PASO 3: Garantir persistencia
netfilter-persistent save

#-----#
# Valida servicio
#-----#
sysctl -p
ss -tulpn | grep :53
ss -tulpn | grep :67

#-----#
# Extraer logs, configs e historial
#-----#
rm -rf $P3ASORC_MEMORIA
mkdir -p $P3ASORC_MEMORIA
mkdir -p $P3ASORC_CONFIG

journalctl --no-pager --since=today | grep iptables > $P3ASORC_LOG
cp $P3ASORC_ROUTER_CONFIG $P3ASORC_CONFIG

history > $P3ASORC_HISTORIAL
chmod 777 -R $P3ASORC_MEMORIA

tree $P3ASORC_MEMORIA

#-----#
# Comprobación desde host
#-----#

# HACK: Con esto añadimos otra ip a la misma interfaz
sudo ip addr add 192.168.25.99/24 dev vboxnet0
sudo ip route add default via 192.168.25.10 table 100
sudo ip rule add from 192.168.25.99 table 100
ip a

```

```
# TEST
ping -c 1 -I 192.168.25.99 192.168.25.1
ping -c 2 -I 192.168.25.99 1.1.1.1
ping -c 3 -I 192.168.25.99 gnu.org
traceroute -I -s 192.168.25.99 192.168.25.1
traceroute -I -s 192.168.25.99 1.1.1.1
traceroute -I -s 192.168.25.99 gnu.org
```

Fragmento de código 12: Configuración y comprobación del **enrutado** en  Debian GNU/Linux 13 (Trixie).

## 5.2. *Firewall (pf)* en ● FreeBSD 14.3

```
#!/bin/sh
su -

#-----
# Variables de entorno
#-----
P3ASORC_SERVICIO=router_vpn_firewall
P3ASORC_SISTEMA=unix

# NO TOCAR
P3ASORC_MEMORIA=/home/ivan/SUPERMEMORIA/$P3ASORC_SERVICIO
P3ASORC_CONFIG=$P3ASORC_MEMORIA/ficheros_configuracion
P3ASORC_LOG=$P3ASORC_MEMORIA/$P3ASORC_SISTEMA.log
P3ASORC_HISTORIAL=$P3ASORC_MEMORIA/history$P3ASORC_SISTEMA.txt
P3ASORC_BSD_RC=/etc/rc.conf

# -
P3ASORC_FIREWALL_CONFIG=/etc/pf.conf

#-----
# Servicio
#-----
# PASO 0. Crear configuración
# Permitir solo páginas y SSH.
cat << EOF > $P3ASORC_FIREWALL_CONFIG
block all
pass out all keep state
pass in proto tcp from any to any port 22
EOF

# PASO A: Activar
sysrc pf_enable="YES"
sysrc pf_rules="$P3ASORC_FIREWALL_CONFIG"
sysrc pflog_enable="YES"
service pf start
pfctl -sr

# PASO B: Desactivar
service pf stop
sysrc pf_enable="NO"
sysrc pf_rules="$P3ASORC_FIREWALL_CONFIG"
sysrc pflog_enable="NO"
```

```

#-----#
# Valida servicio
#-----#
service apache24 status
pfctl -sr

#-----#
# Extraer logs, configs e historial
#-----#
rm -rf $P3ASORC_MEMORIA
mkdir -p $P3ASORC_MEMORIA
mkdir -p $P3ASORC_CONFIG

cp $P3ASORC_FIREWALL_CONFIG $P3ASORC_CONFIG/
cp $P3ASORC_BSD_RC $P3ASORC_CONFIG/rc.conf

touch $P3ASORC_LOG
pfctl -sr > $P3ASORC_LOG
history > $P3ASORC_HISTORIAL
chmod 777 -R $P3ASORC_MEMORIA

tree $P3ASORC_MEMORIA

#-----#
# Comprobación desde host
#-----#
curl -I http://192.168.25.11/SOGo
ping -c 1 192.168.25.11

```

Fragmento de código 13: Configuración y comprobación del *firewall* en ● FreeBSD 14.3.

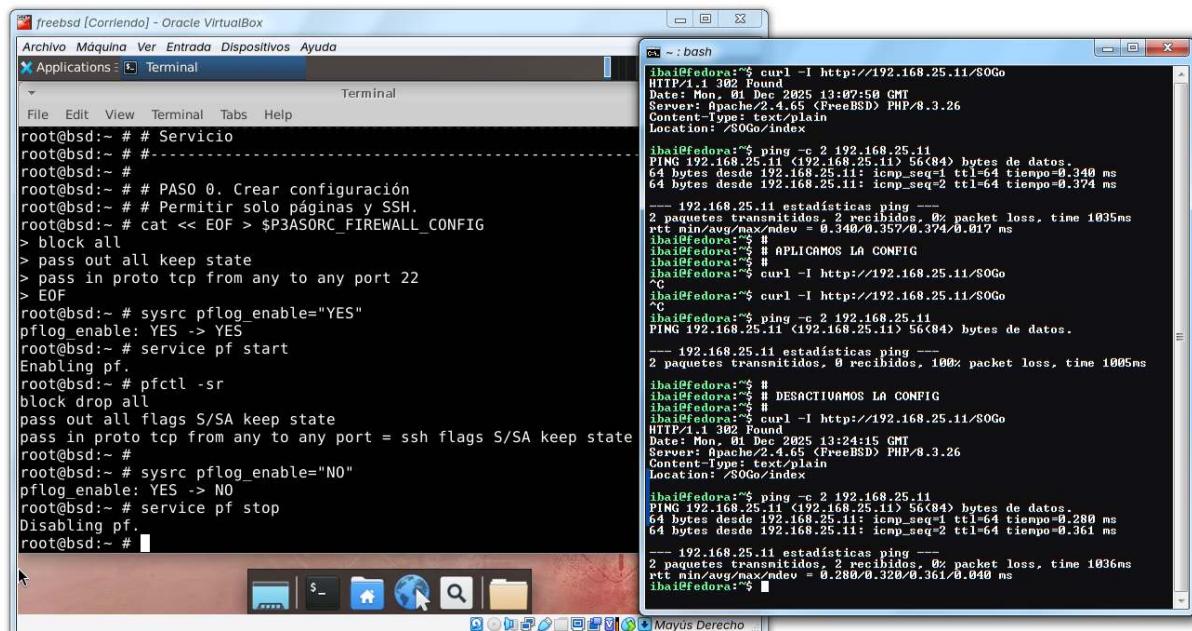


Figura 38: Demostración del esperado funcionamiento de este servicio en ● FreeBSD 14.3.

## 5.3. VPN (PPTP) en Microsoft Windows Server 2025

Es de vital recomendación seguir los pasos descritos aquí con el *firewall* desactivado y cualquier otra barrera que impidiera el correcto funcionamiento.

### 5.3.1. Instalación de característica opcional y configuración del servicio

Si bien la compañía de Windows nos garantiza una interfaz para todo, para ahorrarse navegar por menús de opciones infinitas, ejecutaremos lo siguiente para conservar cordura: `Install-WindowsFeature DirectAccess-VPN -IncludeManagementTools`. Luego seguir el procedimiento adjunto:

1. Ejecutar `rrasmgmt.msc`.
2. Clic derecho en el servidor → *Configurar y habilitar enrutamiento*.
3. Seleccionar **Configuración personalizada** → Marcar **Acceso VPN**.
4. Iniciar el servicio.

### 5.3.2. Configuración de los puertos de puertos

1. En `rrasmgmt.msc`, desplegar el árbol subyacente del servidor.
2. Clic derecho en **Puertos** → **Propiedades**.
3. Seleccionar **WAN Miniport (PPTP)** → *Configurar*.
4. Marcar: **Conexiones de acceso remoto (solo entrada)**.
5. Establecer puertos máximos: **5**.
6. Aplicar y aceptar todo cambio para que surja efecto.

### 5.3.3. Configuración de direcciones

Para evitar fallos de DHCP en entornos virtuales, procederemos con

1. Clic derecho en el servidor → **Propiedades** → Pestaña **IPv4**.
2. Seleccionar **Grupo de direcciones estáticas**.
3. Añadir rango (192.168.25.200 – 192.168.25.210).
4. Aplicar y aceptar todo cambio perpetrado.

### 5.3.4. Permisos de usuario + reglas de *firewall*

1. Correr `dsa.msc` (*Active Directory*) o `lusrmgr.msc` (Local).
2. Propiedades del usuario (Administrador o ivan) → Pestaña **Marcado (Dial-in)**.
3. Permiso de acceso a redes: Marcar **Permitir acceso**.
4. Aceptar destino.

### 5.3.5. Prueba empírica de éxito entre *host* y *guest*

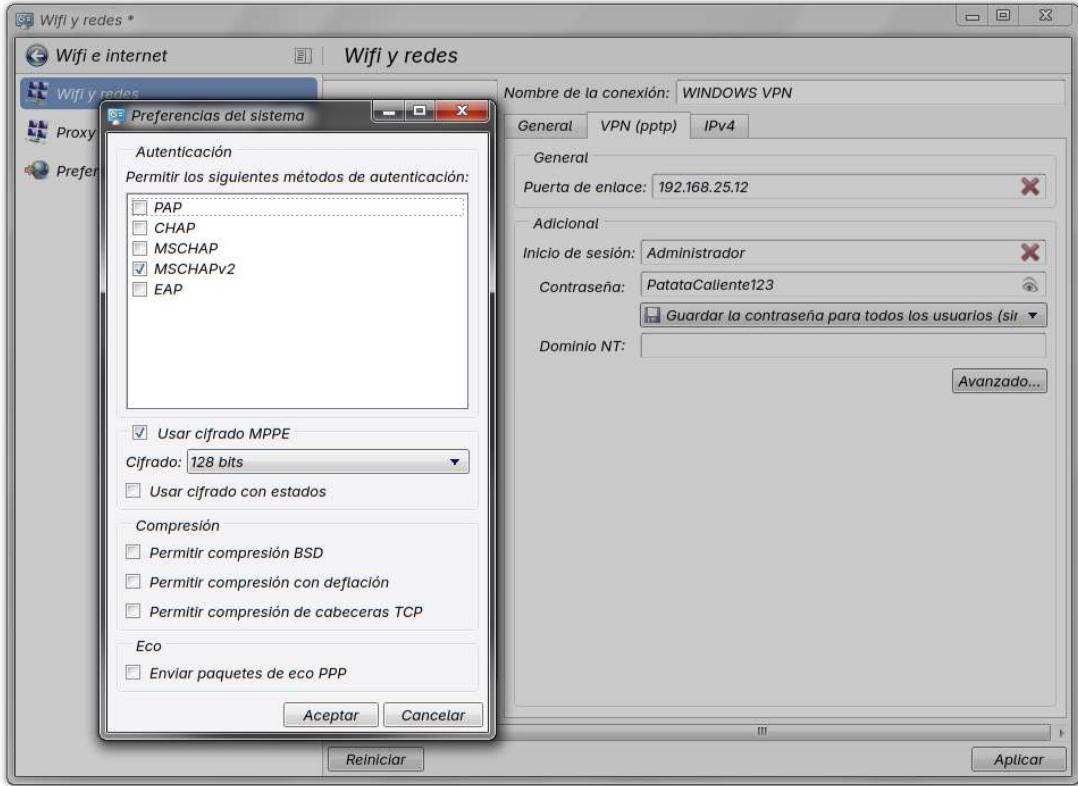


Figura 39: Configuración aplicada desde el panel de control de KDE Plasma (*host*).

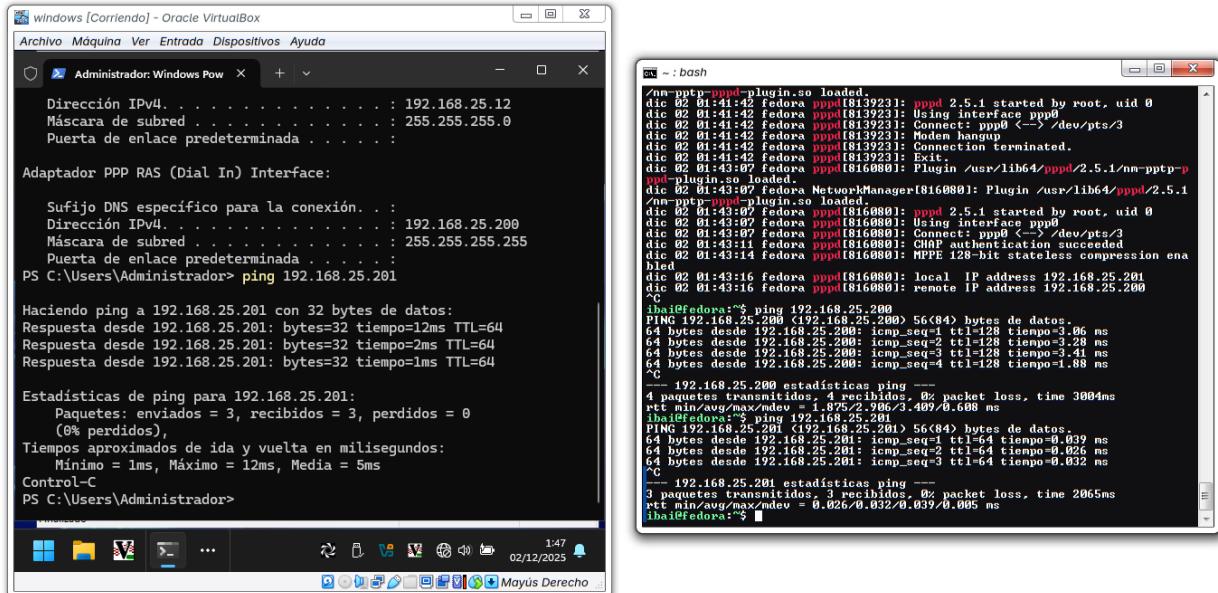


Figura 40: Demostración del esperado funcionamiento de este servicio en Microsoft Windows Server 2025.

### 5.3.6. Comandos útiles

```
#!/bin/sh

# Característica VPN
Install-WindowsFeature DirectAccess-VPN -IncludeManagementTools

# Manejador de enrutamiento y VPN
rrasmgmt.msc

# Administración de grupos y usuarios
lusrmgr.msc

# Mata al proceso que estorba (no es necesario hacer)
Stop-Process -Id $(Get-NetUDPEndpoint -LocalPort 123 | Select-
    ObjectOwningProcess).OwningProcess -Force

# Obtener info de dicho proceso
Get-NetUDPEndpoint -LocalPort 123

# Abrir puerto TCP 1723 (Control PPTP)
New-NetFirewallRule -DisplayName "Permitir PPTP-TCP" -Direction Inbound -
    Action Allow -Protocol TCP -LocalPort 1723

# Abrir Protocolo GRE 47 (Tunel de datos)
New-NetFirewallRule -DisplayName "Permitir GRE-47" -Direction Inbound -
    Action Allow -Protocol 123

# Ver actividad del servicio
netstat -an | findstr 1723

#-----#
# Comprobación desde host
#-----#

# En una terminal
sudo journalctl -f | grep pppd

# Sin conectarse
ping 192.168.25.200
ping 192.168.25.201

# Conectarse desde la bandeja
ping 192.168.25.200
ping 192.168.25.201
```

Fragmento de código 14: Configuración y comprobación de la **VPN** en  Microsoft Windows Server 2025.

## 6. Monitorización de servicios

«Monitorización de servicios (*Nagios*, *PRTG*). Se pide la instalación de un monitor de servicios tipo *Nagios* en cada una de las tres plataformas. En caso de no existir de manera nativa para Windows se pide la instalación de uno alternativo.»

---

La **Monitorización de Sistemas** es el proceso continuo de recolección y análisis de métricas sobre el estado de la infraestructura IT. Su función principal es asegurar la disponibilidad y el rendimiento de los servicios críticos, alertando a los administradores ante cualquier anomalía antes de que afecte a los usuarios finales.

El servicio se basa en una arquitectura de **gestor y agentes**:

- **El Servidor de monitorización (Nagios/PRTG):** Es el núcleo que programa los chequeos, procesa los resultados y envía notificaciones (alertas) vía email o SMS.
- **Plugins y agentes:** Pequeños programas (como NRPE en Linux o WMI/SNMP en Windows) que interrogan a los equipos remotos para obtener datos como uso de CPU, espacio en disco o estado de servicios (HTTP, SSH).

En cuanto a sus orígenes, la monitorización de redes ha evolucionado desde simples comandos *ping*. **Nagios**, el estándar en SW libre mencionado en el enunciado, fue creado por Ethan Galstad en **1999** bajo el nombre original de *NetSaint*. Fue renombrado a Nagios en **2002**. Su arquitectura modular basada en plugins definió el modelo moderno de supervisión de infraestructuras heterogéneas.

Sobre uso, es una pieza fundamental en cualquier centro de operaciones de red (NOC). Permite a los administradores pasar de un modelo reactivo (arreglar cuando se rompe) a uno proactivo, identificando cuellos de botella, planificando la capacidad (*capacity planning*) y garantizando el cumplimiento de los acuerdos de nivel de servicio (SLA).

En el desarrollo de esta práctica, la implementación del sistema de monitorización persigue los siguientes objetivos técnicos:

1. **Despliegue del monitor:** Instalación del núcleo de monitorización (**Nagios** en sistemas UNIX/Linux o una alternativa equivalente como **PRTG** o similar en Windows).
2. **Definición de objetos y alertas:** Configuración de *hosts* y servicios a supervisar, estableciendo umbrales de advertencia (*warning*) y crítica (*critical*) para verificar el estado de salud de la plataforma en tiempo real.

### 6.1. Debian GNU/Linux 13 (Trixie) con Nagios

```
#!/bin/sh
su -

#-----#
# Variables de entorno
#-----#
P3AS0RC_SERVICIO=nagios
P3AS0RC_SISTEMA=linux
```

```

# NO TOCAR
P3ASORC_MEMORIA=/home/ivan/SUPERMEMORIA/$P3ASORC_SERVICIO
P3ASORC_CONFIG=$P3ASORC_MEMORIA/ficheros_configuracion
P3ASORC_LOG=$P3ASORC_MEMORIA/$P3ASORC_SISTEMA.log
P3ASORC_HISTORIAL=$P3ASORC_MEMORIA/history$P3ASORC_SISTEMA.txt

# ---
P3ASORC_NAGIOS_USER=ivan
P3ASORC_NAGIOS_PASS=1
P3ASORC_NAGIOS_DIR=/etc/nagios4
P3ASORC_NAGIOS_CONFIG=$P3ASORC_NAGIOS_DIR/nagios.cfg
P3ASORC_NAGIOS_CGI_CONFIG=$P3ASORC_NAGIOS_DIR/cgi.cfg
P3ASORC_NAGIOS_HTPASSWD=$P3ASORC_NAGIOS_DIR/htpasswd.users
P3ASORC_NAGIOS_MAIN_LOG=/var/log/nagios4/nagios.log

#-----
# Servicio
#-----
# PASO 0: Limpieza
systemctl disable nagios4
rm -rf /etc/nagios4 /var/log/nagios4
apt remove -y nagios4 nagios-plugins-basic nagios-nrpe-plugin

# PASO 1: Instalar paquetes
apt install -y nagios4 nagios-plugins-basic nagios-plugins-contrib nagios-
    nrpe-plugin apache2 libapache2-mod-php unzip curl

# PASO 2: Autenticación apache
a2enmod cgi rewrite
htpasswd -c -b $P3ASORC_NAGIOS_HTPASSWD $P3ASORC_NAGIOS_USER
    $P3ASORC_NAGIOS_PASS
chown root:www-data $P3ASORC_NAGIOS_HTPASSWD
chmod 640 $P3ASORC_NAGIOS_HTPASSWD

# PASO 3: Configurar Nagios
sed -i "s/nagios@localhost/$P3ASORC_NAGIOS_USER@debian.asorc.org/g"
    $P3ASORC_NAGIOS_CONFIG
sed -i "s/nagiosadmin/$P3ASORC_NAGIOS_USER/g" $P3ASORC_NAGIOS_CGI_CONFIG

# PASO 4: Puesta en marcha del servicio
# Verificación de sintaxis
nagios4 -v $P3ASORC_NAGIOS_CONFIG
# Reiniciar servicios
systemctl restart apache2
systemctl restart nagios4
systemctl enable nagios4 apache2

#-----
# Valida servicio
#-----
systemctl status nagios4 --no-pager
systemctl status apache2 --no-pager
netstat -tunelp | grep -E 'nagios'
ss -tunelp | grep apache2

#-----
# Extraer logs, configs e historial
#-----

```

```

rm -rf $P3ASORC_MEMORIA
mkdir -p $P3ASORC_MEMORIA
mkdir -p $P3ASORC_CONFIG

cp $P3ASORC_NAGIOS_CONFIG $P3ASORC_CONFIG/
cp $P3ASORC_NAGIOS_CGI_CONFIG $P3ASORC_CONFIG/
cp /etc/apache2/conf-enabled/nagios4-cgi.conf $P3ASORC_CONFIG/apache_nagios.conf
cp $P3ASORC_NAGIOS_HTPASSWD $P3ASORC_CONFIG/
cp $P3ASORC_NAGIOS_MAIN_LOG $P3ASORC_LOG

history > $P3ASORC_HISTORIAL
chmod 777 -R $P3ASORC_MEMORIA

tree $P3ASORC_MEMORIA

#-----#
# Comprobación
#-----#
firefox http://192.168.25.10/nagios4
firefox http://debian.debian.asorc.org/nagios4

```

Fragmento de código 15: Configuración y comprobación de este servicio en  $\Delta$  Debian GNU/Linux 13 (Trixie).

Host	Service	Status	Last Check	Duration	Attempt	Status Information
localhost	Current Load	OK	11-26-2025 13:15:45	0d 0h 8m 9s	1/4	LOAD OK - total load average: 0.43, 0.95, 0.89
	Current Users	OK	11-26-2025 13:16:23	0d 0h 7m 31s	1/4	USERS OK - 2 users currently logged in
	HTTP	OK	11-26-2025 13:17:00	0d 0h 6m 54s	1/4	HTTP OK: HTTP/1.1 200 OK - 521 bytes in 0.008 second response time
	PING	OK	11-26-2025 13:17:38	0d 0h 6m 16s	1/4	PING OK - Packet loss = 0%, RTA = 0.63 ms
	Root Partition	OK	11-26-2025 13:18:15	0d 0h 5m 39s	1/4	DISK OK - free space: / 9807MiB (99.4% inode=86%).
	SSH	OK	11-26-2025 13:13:53	0d 0h 2m 41s+	1/4	SSH OK - OpenSSH_10.0p2 (protocol 2.0)
	Swap Usage	OK	11-26-2025 13:14:30	0d 0h 2m 41s+	1/4	SWAP OK - 100% free (1905MB out of 1905MB)
	Total Processes	OK	11-26-2025 13:15:08	0d 0h 2m 41s+	1/4	PROCS OK: 0 processes with STATE = RSZDT

Figura 41: Demostración desde *host* donde se aprecia la monitorización de los servicios hospedados en un servidor bajo  $\Delta$  Debian GNU/Linux 13 (Trixie).

## 6.2. ● FreeBSD 14.3 con Nagios

The screenshot shows the Nagios Core interface running in Mozilla Firefox. The title bar reads "Nagios Core — Mozilla Firefox". The address bar shows the URL "http://bsd.asorc.org/nagios/". The main content area displays the "Service Status Details For All Hosts" table. The table has columns: Host, Service, Status, Last Check, Duration, Attempt, and Status Information. There are 8 rows in the table, each corresponding to a service on the host "localhost". The services listed are Current Load, Current Users, HTTP, PING, Root Partition, SSH, Swap Usage, and Total Processes. Most services are in an "OK" state, except for HTTP which is "DOWN". The "Status Information" column provides detailed logs for each service.

Host	Service	Status	Last Check	Duration	Attempt	Status Information
localhost	Current Load	OK	11-26-2025 11:40:44	0d 1h 27m 37s	1/4	OK - load average: 0.40, 0.66, 0.63
localhost	Current Users	OK	11-26-2025 11:41:22	0d 1h 26m 59s	1/4	USERS OK - 2 users currently logged in
localhost	HTTP	DOWN	11-26-2025 11:42:45	0d 1h 26m 21s	1/4	HTTP OK: HTTP/1.1 200 OK - 1134 bytes in 0.002 second response time
localhost	PING	OK	11-26-2025 11:42:37	0d 1h 25m 44s	1/4	PING OK - Packet loss = 0%, RTA = 0.74 ms
localhost	Root Partition	OK	11-26-2025 11:38:54	0d 1h 25m 7s	1/4	DISK OK - free space: / 12206 MB (51.46% inode=93%).
localhost	SSH	DOWN	11-26-2025 11:38:52	0d 1h 24m 29s	1/4	SSH OK - OpenSSH_9.9 FreeBSD-20250219 (protocol 2.0)
localhost	Swap Usage	OK	11-26-2025 11:39:29	0d 1h 23m 52s	1/4	SWAP OK - 100% free (2047 MB out of 2047 MB)
localhost	Total Processes	OK	11-26-2025 11:40:08	0d 1h 23m 14s	1/4	PROCS OK: 29 processes with STATE = RSZDT

Figura 42: Demostración desde *host* donde se aprecia la monitorización de los servicios hospedados en un servidor bajo ● FreeBSD 14.3.

```
#!/bin/sh
su -

# -----
# Variables de entorno
# -----
P3ASORC_SERVICIO=nagios
P3ASORC_SISTEMA=unix

# NO TOCAR
P3ASORC_MEMORIA=/home/ivan/SUPERMEMORIA/$P3ASORC_SERVICIO
P3ASORC_CONFIG=$P3ASORC_MEMORIA/ficheros_configuracion
P3ASORC_LOG=$P3ASORC_MEMORIA/$P3ASORC_SISTEMA.log
P3ASORC_HISTORIAL=$P3ASORC_MEMORIA/history$P3ASORC_SISTEMA.txt
P3ASORC_BSD_RC=/etc/rc.conf

# ---
P3ASORC_NAGIOS_USER=ivan
P3ASORC_NAGIOS_PASS=1
P3ASORC_NAGIOS_OBJECTS_DIR=/usr/local/etc/nagios/objects
P3ASORC_NAGIOS_CONFIG=/usr/local/etc/nagios/nagios.cfg
P3ASORC_NAGIOS_MAIN_LOG=/var/spool/nagios/nagios.log
P3ASORC_NAGIOS_HOST_CONFIG=/usr/local/etc/nagios/objects/localhost.cfg
P3ASORC_NAGIOS_APACHE_CONF=/usr/local/etc/apache24/httpd.conf
P3ASORC_NAGIOS_APACHE_CONF_DIR=/usr/local/etc/apache24
P3ASORC_NAGIOS_APACHE_HTPASSWD=/usr/local/etc/nagios/htpasswd.users
```

```

P3ASORC_NAGIOS_APACHE_NAGIOS_CONF=/usr/local/etc/apache24/Includes/nagios.conf
P3ASORC_NAGIOS_CFG_SAMPLE_NAME="nagios.cfg-sample"
P3ASORC_NAGIOS_HOST_CFG_SAMPLE_NAME="localhost.cfg-sample"
P3ASORC_NAGIOS_SPOOL=/var/spool/nagios

#-----
# Servicio
#-----
# # PASO 0: Limpieza
service nagios stop
service apache24 stop
rm -rf $P3ASORC_NAGIOS_CONFIG $P3ASORC_NAGIOS_MAIN_LOG
$P3ASORC_NAGIOS_APACHE_HTPASSWD
pkg delete -y nagios
# Ojito que quizás no interese borrar
pkg delete apache php

# PASO 1: Instalar Nagios, plugins y Servidor Web (Apache)
# Instalamos los componentes principales
pkg install -y nagios nagios-plugins apache24 php83-extensions wget autoconf
automake gettext gcc openssl net-snmp p5-Net-SNMP-Util php83-zlib php83-ctype
php83-session php83-simplexml

# Archivos de objetos esenciales
cp /usr/local/etc/nagios/$P3ASORC_NAGIOS_CFG_SAMPLE_NAME
$P3ASORC_NAGIOS_CONFIG
cp /usr/local/etc/nagios/$P3ASORC_NAGIOS_CGI_SAMPLE_NAME
$P3ASORC_NAGIOS_CGI_CONFIG
cp /usr/local/etc/nagios/resource.cfg-sample /usr/local/etc/nagios/resource.cfg
cp $P3ASORC_NAGIOS_OBJECTS_DIR/commands.cfg-sample
$P3ASORC_NAGIOS_OBJECTS_DIR/commands.cfg
cp $P3ASORC_NAGIOS_OBJECTS_DIR/contacts.cfg-sample
$P3ASORC_NAGIOS_OBJECTS_DIR/contacts.cfg
cp $P3ASORC_NAGIOS_OBJECTS_DIR/timeperiods.cfg-sample
$P3ASORC_NAGIOS_OBJECTS_DIR/timeperiods.cfg
cp $P3ASORC_NAGIOS_OBJECTS_DIR/templates.cfg-sample
$P3ASORC_NAGIOS_OBJECTS_DIR/templates.cfg
cp $P3ASORC_NAGIOS_OBJECTS_DIR/$P3ASORC_NAGIOS_HOST_CFG_SAMPLE_NAME
$P3ASORC_NAGIOS_HOST_CONFIG

# PASO 2: Configurar apache y usuarios
sysrc apache24_enable="YES"
mkdir -p $(dirname $P3ASORC_NAGIOS_APACHE_HTPASSWD)
htpasswd -cb $P3ASORC_NAGIOS_APACHE_HTPASSWD $P3ASORC_NAGIOS_USER
$P3ASORC_NAGIOS_PASS

# Configuración PHP
cp /usr/local/etc/php.ini-production /usr/local/etc/php.ini
sed -i '' 's/DirectoryIndex index.html/DirectoryIndex index.php index.html
index.htm/g' $P3ASORC_NAGIOS_APACHE_CONF

# Carga forzada de módulos (CGI y PHP)
cat << EOF >> $P3ASORC_NAGIOS_APACHE_CONF

# --- AGREGADO POR SCRIPT NAGIOS ---
LoadModule cgi_module libexec/apache24/mod_cgid.so

```

```

LoadModule cgi_module libexec/apache24/mod_cgi.so

<FilesMatch "\.php$">
    SetHandler application/x-httpd-php
</FilesMatch>
<FilesMatch "\.phps$">
    SetHandler application/x-httpd-php-source
</FilesMatch>
# -----
EOF

# Configuración VirtualHost de Nagios
mkdir -p $(dirname $P3ASORC_NAGIOS_APACHE_NAGIOS_CONF)
cat << EOF > $P3ASORC_NAGIOS_APACHE_NAGIOS_CONF
ScriptAlias /nagios/cgi-bin "/usr/local/www/nagios/cgi-bin"

<Directory "/usr/local/www/nagios/cgi-bin">
    AllowOverride None
    Options ExecCGI
    AddHandler cgi-script .cgis
    AuthName "Nagios Access"
    AuthType Basic
    AuthUserFile $P3ASORC_NAGIOS_APACHE_HTPASSWD
    Require valid-user
</Directory>

Alias /nagios "/usr/local/www/nagios"

<Directory "/usr/local/www/nagios">
    Options None

    AuthName "Nagios Access"
    AuthType Basic
    AuthUserFile $P3ASORC_NAGIOS_APACHE_HTPASSWD
    Require valid-user
</Directory>
EOF

# PASO 3: Configurar Nagios Daemon y permisos de usuario
echo "--- CONFIGURACION NAGIOS Y PERMISOS ---"
sysrc nagios_enable="YES"
sysrc nagios_user="www"

sed -i '' 's/nagios$/www/g' $P3ASORC_NAGIOS_CONFIG
sed -i '' 's/nagios$/www/g' $P3ASORC_NAGIOS_HOST_CONFIG
sed -i '' "s/nagios@localhost/$P3ASORC_NAGIOS_USER@bsd.asorc.org/g"
$P3ASORC_NAGIOS_CONFIG

sed -i '' "s/nagiosadmin/$P3ASORC_NAGIOS_USER/g" $P3ASORC_NAGIOS_CGI_CONFIG
chmod 644 $P3ASORC_NAGIOS_CGI_CONFIG

# PASO 4: Otorgar permisos de sistema y comprobación
rm -f $P3ASORC_NAGIOS_SPOOL/nagios.log $P3ASORC_NAGIOS_SPOOL/status.sav
chown -R www:www $P3ASORC_NAGIOS_SPOOL
chmod -R 775 $P3ASORC_NAGIOS_SPOOL

# Permisos de ejecución CGI

```

```

chmod +x /usr/local/www/nagios/cgi-bin/*
chown root:www /usr/local/www/nagios/cgi-bin/*

# Verificación final
nagios -v $P3ASORC_NAGIOS_CONFIG

# Arranque
service apache24 start
service nagios start
service apache24 restart
service nagios restart

# FIX PARA PHP83
sed -i .bak 's/^nagios_user=www/nagios_user=nagios/' /usr/local/etc/nagios/
    nagios.cfg
sed -i .bak 's/^nagios_group=www/nagios_group=nagios/' /usr/local/etc/nagios/
    nagios.cfg
pw groupmod nagios -m www
chmod -R 775 /var/spool/nagios
chown -R nagios:nagios /var/spool/nagios
/usr/local/bin/nagios -v /usr/local/etc/nagios/nagios.cfg
service apache24 start
service nagios start
service apache24 restart
service nagios restart

#-----
# Valida servicio
#-----
service nagios status
service apache24 status
sockstat | grep -E 'nagios|httpd'

#-----
# Extraer logs, configs e historial
#-----
rm -rf $P3ASORC_MEMORIA
mkdir -p $P3ASORC_MEMORIA
mkdir -p $P3ASORC_CONFIG

cp $P3ASORC_NAGIOS_CONFIG $P3ASORC_CONFIG
cp $P3ASORC_NAGIOS_CGI_CONFIG $P3ASORC_CONFIG
cp $P3ASORC_NAGIOS_HOST_CONFIG $P3ASORC_CONFIG/localhost.cfg
cp $P3ASORC_NAGIOS_APACHE_CONF $P3ASORC_CONFIG/httpd.conf
cp /usr/local/etc/php.ini $P3ASORC_CONFIG/php.ini
cp $P3ASORC_APACHE_HTPASSWD $P3ASORC_CONFIG
cp $P3ASORC_NAGIOS_APACHE_NAGIOS_CONF $P3ASORC_CONFIG
cp $P3ASORC_BSD_RC $P3ASORC_CONFIG
cp $P3ASORC_NAGIOS_MAIN_LOG $P3ASORC_LOG

history > $P3ASORC_HISTORIAL
chmod 777 -R $P3ASORC_MEMORIA

tree $P3ASORC_MEMORIA

#-----
# Comprobación
#-----

```

```
firefox http://192.168.25.11/nagios/
firefox http://bsd.asorc.org/nagios/
```

Fragmento de código 16: Configuración y comprobación de este servicio en ● FreeBSD 14.3.

### 6.3. Microsoft Windows Server 2025 con Windows Admin Central (WAC)

En esta ocasión, para este servicio se optó por una solución hecha por la propia Microsoft, **WAC**. Este SW es conjunto de extensiones para el manejo y administración del servidor en cuestión. Está provisto de un *frontend* que únicamente está pensado para funcionar con navegadores chromium (cosas de Angular.js o manías de los diseñadores). Este complejo SW no solo es utilizado para la motorización de los servicios, sino que puede extenderse a una completa y robusta administración de servidores utilizando un *front* que medianamente funciona, porque en este o va lento o repites lo que quieras hacer 3 veces.

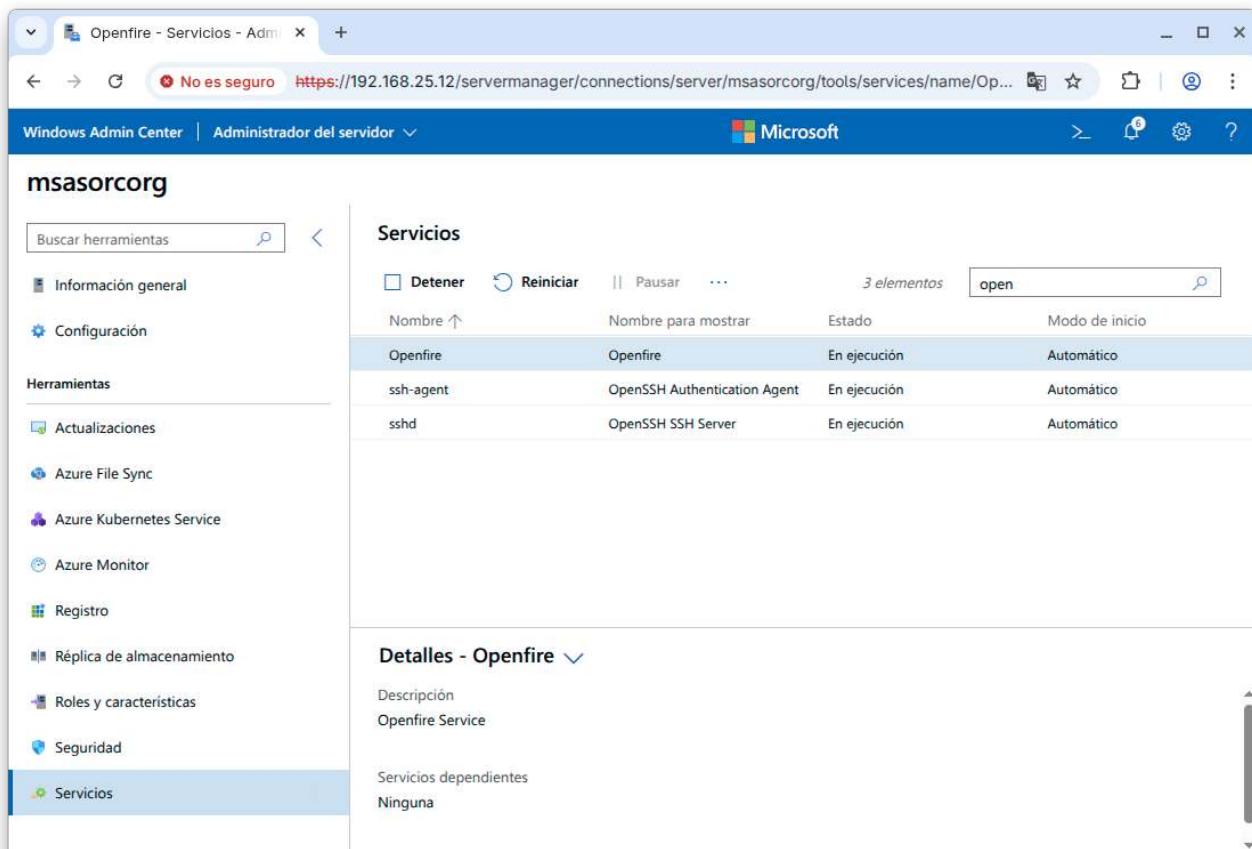


Figura 43: Demostración desde *host* donde se aprecia la monitorización de los servicios hospedados en un servidor bajo Microsoft Windows Server 2025.

```
#!/bin/sh

$Cert = New-SelfSignedCertificate -DnsName "msasorcorg" -KeyAlgorithm RSA -
    KeyLength 2048 -CertStoreLocation "Cert:\LocalMachine\My" -NotAfter (Get-
    Date).AddYears(10) -KeyExportPolicy Exportable -Provider "Microsoft
    Enhanced RSA and AES Cryptographic Provider"
```

```

$Thumbprint = $Cert.Thumbprint
$keyPath = $Cert.PrivateKey.CspKeyContainerInfo.UniqueKeyContainerName
$keyPathFull = "$env:ProgramData\Microsoft\Crypto\RSA\MachineKeys\$keyPath"
$NetworkServiceSid = [System.Security.Principal.SecurityIdentifier]::new([
    System.Security.Principal.WellKnownSidType]::NetworkServiceSid, $null)
$acl = Get-Acl $keyPathFull
$ar = New-Object System.Security.AccessControl.FileSystemAccessRule(
    $NetworkServiceSid, "Read", "Allow")
$acl.SetAccessRule($ar)
Set-Acl $keyPathFull $acl
Export-Certificate -Cert $Cert -FilePath "C:\msasorcorg.cer"
Import-Certificate -FilePath "C:\msasorcorg.cer" -CertStoreLocation "Cert:\LocalMachine\Root"

# Descargar WAC desde https://www.microsoft.com/es-es/evalcenter/download-windows-admin-center o bien buscar la aplicación Windows Center Admin (que posiblemente esté preinstalada)
# Aceptar términos y condiciones del servicio.
# Elegir instalación personalizada
# Establecer que el acceso a red es remoto también
# Establecer la autenticación por etiquetas HTML de formulario
# Puerto de escucha: 443
# Usar un certificado TLS y poner el contenido de la variable $RawCert
# Dejar FQDN por defecto
# Permitir el acceso desde cualquier ordenador
# Usar https
# Quitar las actualizaciones automáticas
# Establecer que la recolección espía de Microsoft sea solo de datos necesarios
# Instalar

Set-Item WSMan:\localhost\Client\TrustedHosts -Value '*' -Force
Add-Content -Path "C:\Windows\System32\drivers\etc\hosts" -Value "127.0.0.1 msasorcorg"
New-ItemProperty -Path "HKLM:\SYSTEM\CurrentControlSet\Control\Lsa\MSV1_0" -Name "BackConnectionHostNames" -Value "msasorcorg" -PropertyType MultiString -Force
$WacService = Get-Service | Where-Object {$__.DisplayName -like "*Windows Admin Center*" -or $__.Name -like "*Sme*"}

if ($WacService) {
    Write-Host "Servicio encontrado: $($WacService.Name)" -ForegroundColor Green
    Write-Host "Reiniciando..."
    Restart-Service -Name $WacService.Name -Force
    Write-Host "Servicio reiniciado correctamente." -ForegroundColor Green
} else {
    Write-Host "No he encontrado el servicio. Es posible que WAC no se instalara como servicio." -ForegroundColor Red
}

# Instalar certificado C:\msasorcorg.cer
# En el Equipo local (Local Machine) y "Colocar todos los certificados en el siguiente almacén".
# Para luego pulsar "Examinar" y elegir la segunda carpeta: Entidades de certificación raíz de confianza (Trusted Root Certification Authorities).

Stop-Service WindowsAdminCenter -Force -ErrorAction SilentlyContinue

```

```

Stop-Service ServerManagementGateway -Force -ErrorAction SilentlyContinue

$RutaCache1 = "$env:ProgramData\Microsoft\Windows\ServerManagementGateway\
    Cache"
$RutaCache2 = "$env:ProgramData\ServerManagementGateway\Cache"

if (Test-Path $RutaCache1) { Remove-Item "$RutaCache1\*" -Recurse -Force;
    Write-Host "Caché 1 borrada." }
if (Test-Path $RutaCache2) { Remove-Item "$RutaCache2\*" -Recurse -Force;
    Write-Host "Caché 2 borrada." }

Write-Host "Iniciando servicio..." -ForegroundColor Green
Start-Service WindowsAdminCenter -ErrorAction SilentlyContinue
Start-Service ServerManagementGateway -ErrorAction SilentlyContinue

#-----
# Comprobación
#-----
chromium https://msasorcorg
chromium https://192.168.25.12/servermanager/connections/server/msasorcorg/
    tools/services/name/FDResPub

# Iniciar sesión
# Autoagregar servidor 192.168.25.12

```

Fragmento de código 17: Configuración y comprobación de este servicio en Microsoft Windows Server 2025.

## 6.4. Microsoft Windows Server 2025 con PRTG

Bajo la petición del coordinador de esta asignatura de «échale un veo». la autoría de este kilométrico documento ve necesario añadir este punto. Este SW es un *freemium* y tiene una prueba gratuita para usar fácilmente. El proceso de instalación no requiere mayor complicación que otorgarle su correo electrónico a una compañía y ejecutar SW propietario en el sistema. Eso sí, es extremadamente fácil pues, solo es pulsar «Siguiente», «Siguiente», «Siguiente», «Siguiente»... Debida a la simplicidad sobre la instalación, omitiremos este proceso. Pero es necesario comentar que el correo entregado al sitio web para su descarga de «usar y tirar», de uso temporal. El programa, nos proporcionará una clave de licencia como 000023-PJ4BRG-C6BK7U-ZHUVTB-AMVXXN-RJCTEQ-SEGQ3D-KRIJF2-X5UC3S-IAUPCD que es utilizada en el proceso de instalación poniendo el correo que usemos para esta clave.

The screenshot shows a web browser window displaying the PRTG Network Monitor interface. The title bar reads '/Lista Top 10 | PRTG Network Monitor (MSASORCORG)'. The main content area is titled 'Lista Top 10' and features a table with the heading 'Mejor disponibilidad (mayor tiempo de actividad)'. The table lists various services and their availability percentages:

Tiempo activo [%]	Sensor	Dispositivo
100,0000%	Ping v2	DHCP: 10.0.2.2
100,0000%	Ping v2	192.168.25.1
100,0000%	Ping v2	Puerta de enlace: 10.0.2.1
100,0000%	Ping v2	DNS: one
100,0000%	Ping v2	msasorcorg
100,0000%	Ping v2	10.0.2.21
100,0000%	HTTP	Internet
100,0000%	Salud del servidor central (a...)	Servidor central de PRTG

At the bottom of the browser window, there is footer information: 'PAESSLER 25.4.114.1032+ Administrador de sistema PRTG 16:22 Actualizar en 29 s', 'Contacta con soporte', and 'Ayuda'.

Figura 44: Demostración desde *host* donde se aprecia la monitorización de los servicios hospedados en un servidor bajo Microsoft Windows Server 2025.

## 7. RAID 5

«RAID: Se pide la creación de un RAID 5 en cada uno de los sistemas con 4 discos. El raid deberá de estar formateado, montado y operativo en un directorio del sistema. Durante la corrección se generará un fallo en uno de los discos, se eliminará y se añadirá otro nuevo, mostrando cómo el sistema es capaz de recuperarse del fallo.»

---

El RAID (*Redundant Array of Independent Disks* o Matriz Redundante de Discos Independientes) es una tecnología de virtualización de almacenamiento que combina múltiples componentes de unidades de disco físicas en una o más unidades lógicas. Su función principal es la redundancia de datos, la mejora del rendimiento o ambas.

El servicio se basa en una arquitectura de **distribución de datos** (*data striping*) y redundancia:

- **Nivel Físico:** Conjunto de discos duros (en esta práctica, cuatro unidades de 256 MB) que operan como esclavos del controlador.
- **Nivel Lógico:** El sistema operativo percibe una única unidad de volumen (MD o *Multi-Device*) que abstrae la complejidad del almacenamiento subyacente.

En cuanto a sus orígenes, el término fue acuñado en **1987** por David Patterson, Garth Gibson y Randy Katz en la Universidad de California, Berkeley. Su publicación seminal, «*A Case for Redundant Arrays of Inexpensive Disks (RAID)*», argumentaba que un arreglo de discos económicos y de menor rendimiento podría superar a un único disco grande y costoso (*Single Large Expensive Disk* o SLED) tanto en rendimiento como en fiabilidad.

Sobre uso, el RAID 5 seleccionado para esta práctica es el estándar de facto para servidores de archivos y aplicaciones empresariales generales. Ofrece un equilibrio ideal entre rendimiento de lectura, tolerancia a fallos y eficiencia de almacenamiento. A diferencia del RAID 1 (espejo), que pierde el 50 % de la capacidad, el RAID 5 solo «pierde» la habilidad equivalente a un disco para la paridad distribuida, independientemente del número total de discos. Es comúnmente utilizado en servidores de bases de datos con alta carga de lectura, servidores web y sistemas de archivos corporativos.

En el desarrollo de esta práctica, la implementación del sistema RAID tiene un triple objetivo técnico:

1. **Creación y montaje:** Configurar un dispositivo de bloques virtual (`/dev/md0` en  Debian GNU/Linux 13 (Trixie)) uniendo cuatro dispositivos físicos y estableciendo un sistema de ficheros (*filesystem*) sobre él.
2. **Paridad distribuida:** Demostrar cómo el algoritmo RAID 5 distribuye la información de paridad entre todos los discos, permitiendo la reconstrucción de datos si falla una unidad.
3. **Recuperación ante desastres:** Simular un fallo crítico marcando un disco como defectuoso (*faulty*), retirándolo en caliente (*hot remove*) y reconstruyendo el arreglo (*rebuild*) con una nueva unidad sin pérdida de servicio.

Para ello, crearemos 4 discos por cada SO. Se recomienda marcar la casilla de «conectable en caliente» por cada nueva adición.

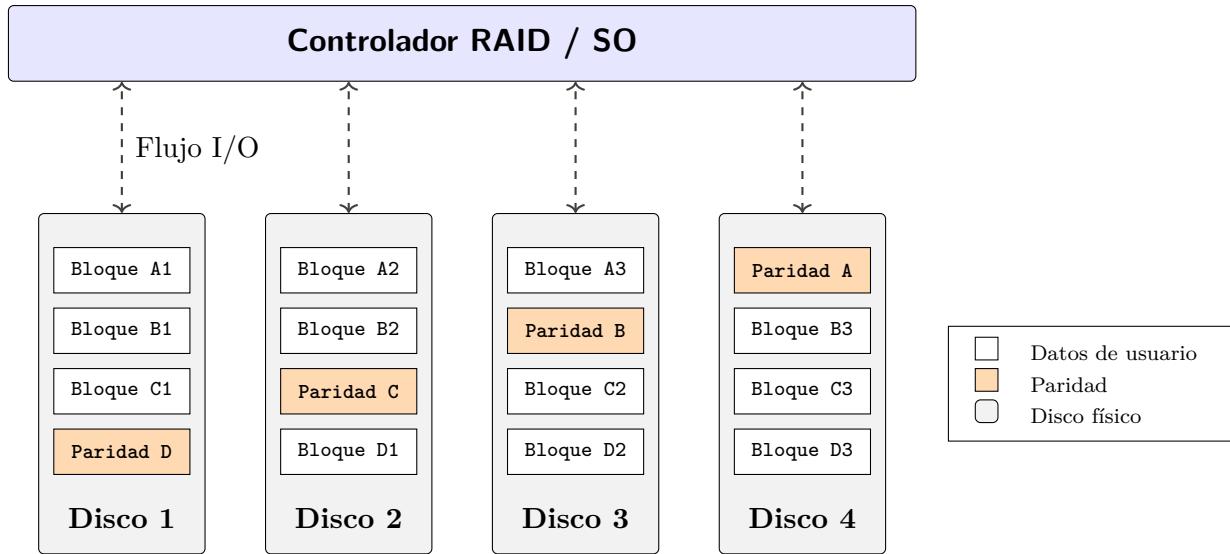


Figura 45: Arquitectura de interconexión lógica de RAID 5.

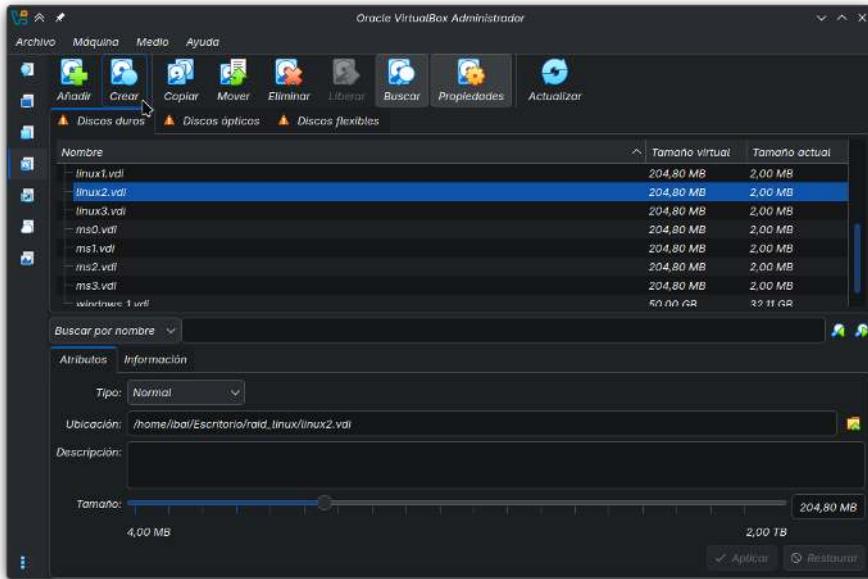


Figura 46: Pantalla de VBox desde donde administrar discos virtuales.

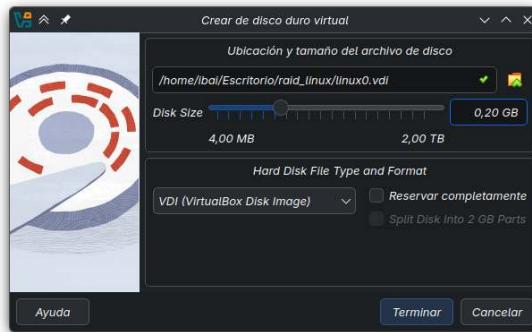


Figura 47: *Popup* con parámetros de creación de un disco virutal que asigna 0,20 GB. Con la excepción para Microsoft Windows Server 2025 que los discos deben ocupar 4,50 GB.

## 7.1. ☈ Debian GNU/Linux 13 (Trixie) con mdadm

```
#!/bin/sh
su -

#-----
# Variables de entorno
#-----
P3ASORC_SERVICIO=raid
P3ASORC_SISTEMA=linux

# NO TOCAR
P3ASORC_MEMORIA=/home/ivan/SUPERMEMORIA/$P3ASORC_SERVICIO
P3ASORC_CONFIG=$P3ASORC_MEMORIA/ficheros_configuracion
P3ASORC_LOG=$P3ASORC_MEMORIA/$P3ASORC_SISTEMA.log
P3ASORC_HISTORIAL=$P3ASORC_MEMORIA/history$P3ASORC_SISTEMA.txt

# ---
P3ASORC_RAID_LOGTMP=milogssobreraid.log
P3ASORC_RAID_BASE=/dev
P3ASORC_RAID_0=$P3ASORC_RAID_BASE/sdb
P3ASORC_RAID_1=$P3ASORC_RAID_BASE/sdc
P3ASORC_RAID_2=$P3ASORC_RAID_BASE/sdd
P3ASORC_RAID_3=$P3ASORC_RAID_BASE/sde
P3ASORC_RAID_01=$P3ASORC_RAID_BASE/sdb1
P3ASORC_RAID_11=$P3ASORC_RAID_BASE/sdc1
P3ASORC_RAID_21=$P3ASORC_RAID_BASE/sdd1
P3ASORC_RAID_31=$P3ASORC_RAID_BASE/sde1
P3ASORC_RAID_MD=/dev/md0
P3ASORC_RAID_ORDER_FILE=createraid.txt
P3ASORC_RAID_DELETE_FILE=deletepartition.txt
P3ASORC_RAID_MD_POINT=/mnt/miraid

#-----
# Servicio
#-----
# PASO 1: Instala paquetes y limpieza
umount -f $P3ASORC_RAID_MD_POINT
mdadm --remove -S $P3ASORC_RAID_MD
umount -f $P3ASORC_RAID_01
umount -f $P3ASORC_RAID_11
umount -f $P3ASORC_RAID_21
umount -f $P3ASORC_RAID_31
umount -f $P3ASORC_RAID_0
umount -f $P3ASORC_RAID_1
umount -f $P3ASORC_RAID_2
umount -f $P3ASORC_RAID_3

apt remove -y mdadm
apt install -y mdadm

# PASO 2: Formato
# Averiguar qué dispositivos a formatear
lsblk

# Archivos de órdenes de formateo
cat << EOF > $P3ASORC_RAID_DELETE_FILE
d
```

```

w

EOF

cat << EOF > $P3ASORC_RAID_ORDER_FILE
n
p
1

t
raid
w

EOF

# Formatea
fdisk $P3ASORC_RAID_0 < $P3ASORC_RAID_DELETE_FILE
fdisk $P3ASORC_RAID_1 < $P3ASORC_RAID_DELETE_FILE
fdisk $P3ASORC_RAID_2 < $P3ASORC_RAID_DELETE_FILE
fdisk $P3ASORC_RAID_3 < $P3ASORC_RAID_DELETE_FILE

fdisk $P3ASORC_RAID_0 < $P3ASORC_RAID_ORDER_FILE
fdisk $P3ASORC_RAID_1 < $P3ASORC_RAID_ORDER_FILE
fdisk $P3ASORC_RAID_2 < $P3ASORC_RAID_ORDER_FILE
fdisk $P3ASORC_RAID_3 < $P3ASORC_RAID_ORDER_FILE

rm -rf $P3ASORC_RAID_ORDER_FILE $P3ASORC_RAID_DELETE_FILE

# PASO 3: Inicialización
# Crear RAID
echo -e 'y\n' | mdadm --create --verbose $P3ASORC_RAID_MD -l 5 -n 4
$P3ASORC_RAID_01 $P3ASORC_RAID_11 $P3ASORC_RAID_21 $P3ASORC_RAID_31 >
$P3ASORC_RAID_LOGTMP 2>&1

# Sincronizar
mdadm --wait $P3ASORC_RAID_MD
mdadm --detail --verbose $P3ASORC_RAID_MD >> $P3ASORC_RAID_LOGTMP

# Montaje
mkfs.ext4 $P3ASORC_RAID_MD >> $P3ASORC_RAID_LOGTMP 2>&1
mkdir -p $P3ASORC_RAID_MD_POINT
mount $P3ASORC_RAID_MD $P3ASORC_RAID_MD_POINT
chmod 777 $P3ASORC_RAID_MD_POINT

#-----
# Valida servicio
#-----
mdadm --detail --verbose $P3ASORC_RAID_MD
cat /proc/mdstat
lsblk

#-----
# Extraer logs, configs e historial
#-----
rm -rf $P3ASORC_MEMORIA
mkdir -p $P3ASORC_MEMORIA
mkdir -p $P3ASORC_CONFIG

```

```

cp $P3ASORC_RAID_LOGTMP $P3ASORC_LOG
dmesg | grep -i "md" | tail -n 50 >> $P3ASORC_LOG
mdadm --detail --scan --verbose >> $P3ASORC_CONFIG/mdadm.conf

history > $P3ASORC_HISTORIAL
chmod 777 -R $P3ASORC_MEMORIA

tree $P3ASORC_MEMORIA

#-----
# Comprobación desde GUEST
#-----

# Ta bien
echo -e 'FLUJO NORMAL\n' >> $P3ASORC_RAID_LOGTMP
rm -f $P3ASORC_RAID_MD_POINT/hola.txt
ls $P3ASORC_RAID_MD_POINT
echo 'Hola desde ASORC' > $P3ASORC_RAID_MD_POINT/hola.txt
ls $P3ASORC_RAID_MD_POINT
cat $P3ASORC_RAID_MD_POINT/hola.txt
mdadm --detail --verbose $P3ASORC_RAID_MD >> $P3ASORC_RAID_LOGTMP

# Ups, disco roto
echo -e 'Ups, disco roto\n' >> $P3ASORC_RAID_LOGTMP
# Quitar >>* para la corrección
mdadm --verbose -f $P3ASORC_RAID_MD $P3ASORC_RAID_01 >> $P3ASORC_RAID_LOGTMP
2>&1
mdadm --detail --verbose $P3ASORC_RAID_MD
mdadm --detail --verbose $P3ASORC_RAID_MD >> $P3ASORC_RAID_LOGTMP 2>&1
ls $P3ASORC_RAID_MD_POINT
cat $P3ASORC_RAID_MD_POINT/hola.txt

# Limpieza
echo -e 'Limpieza\n' >> $P3ASORC_RAID_LOGTMP
mdadm --manage $P3ASORC_RAID_MD --remove $P3ASORC_RAID_01
mdadm --zero-superblock $P3ASORC_RAID_01

# Toca arreglarlo
echo -e 'Insertar\n' >> $P3ASORC_RAID_LOGTMP
mdadm --add --verbose $P3ASORC_RAID_MD $P3ASORC_RAID_01 >>
$P3ASORC_RAID_LOGTMP 2>&1
mdadm --detail --verbose $P3ASORC_RAID_MD
mdadm --detail --verbose $P3ASORC_RAID_MD >> $P3ASORC_RAID_LOGTMP 2>&1

# Espera a la reconstrucción
echo -e 'Espera a la reconstrucción\n' >> $P3ASORC_RAID_LOGTMP
mdadm --wait $P3ASORC_RAID_MD
mdadm --detail --verbose $P3ASORC_RAID_MD
mdadm --detail --verbose $P3ASORC_RAID_MD >> $P3ASORC_RAID_LOGTMP 2>&1

```

Fragmento de código 18: Configuración y comprobación de este servicio en  Debian GNU/Linux 13 (Trixie). Los *logs* pueden contrastar la comprobación de este servicio.

## 7.2. ● FreeBSD 14.3 con zpool

```
#!/bin/sh
su -

#-----
# Variables de entorno
#-----
P3ASORC_SERVICIO=raid
P3ASORC_SISTEMA=unix

# NO TOCAR
P3ASORC_MEMORIA=/home/ivan/SUPERMEMORIA/$P3ASORC_SERVICIO
P3ASORC_CONFIG=$P3ASORC_MEMORIA/ficheros_configuracion
P3ASORC_LOG=$P3ASORC_MEMORIA/$P3ASORC_SISTEMA.log
P3ASORC_HISTORIAL=$P3ASORC_MEMORIA/history$P3ASORC_SISTEMA.txt
P3ASORC_BSD_RC=/etc/rc.conf

# ---
P3ASORC_RAID_LOGTMP=milogssobreraid.log
P3ASORC_RAID_BASE=/dev
P3ASORC_RAID_0A=ada1
P3ASORC_RAID_1A=ada2
P3ASORC_RAID_2A=ada3
P3ASORC_RAID_3A=ada4
P3ASORC_RAID_0=$P3ASORC_RAID_BASE/$P3ASORC_RAID_0A
P3ASORC_RAID_1=$P3ASORC_RAID_BASE/$P3ASORC_RAID_1A
P3ASORC_RAID_2=$P3ASORC_RAID_BASE/$P3ASORC_RAID_2A
P3ASORC_RAID_3=$P3ASORC_RAID_BASE/$P3ASORC_RAID_3A
P3ASORC_RAID_MD=/dev/md0
P3ASORC_RAID_MD_POINT=/mnt/miraid
P3ASORC_RAID_POOL=miPOOL

#-----
# Servicio
#-----
# Paso 0: Instalar utilidades
pkg install -y lsblk

# PASO 1: Limpieza
umount $P3ASORC_RAID_MD_POINT
zpool destroy $P3ASORC_RAID_POOL

# PASO 2: Configuración y montaje
mkdir -p $P3ASORC_RAID_MD_POINT
zpool create -m $P3ASORC_RAID_MD_POINT $P3ASORC_RAID_POOL raidz1
    $P3ASORC_RAID_0A $P3ASORC_RAID_1A $P3ASORC_RAID_2A $P3ASORC_RAID_3A
chmod 777 $P3ASORC_RAID_MD_POINT
zpool status $P3ASORC_RAID_POOL > $P3ASORC_RAID_LOGTMP

#-----
# Valida servicio
#-----
df -h | grep $P3ASORC_RAID_POOL
camcontrol devlist

#-----
# Extraer logs, configs e historial
```

```

#-----
rm -rf $P3ASORC_MEMORIA
mkdir -p $P3ASORC_MEMORIA
mkdir -p $P3ASORC_CONFIG

cp $P3ASORC_RAID_LOGTMP $P3ASORC_LOG
dmesg | grep -i "pool" | tail -n 50 >> $P3ASORC_LOG

history > $P3ASORC_HISTORIAL
chmod 777 $P3ASORC_MEMORIA

tree $P3ASORC_MEMORIA

#-----
# Comprobación desde host
#-----

# Ta bien
echo -e 'FLUJO NORMAL\n' >> $P3ASORC_RAID_LOGTMP
rm -f $P3ASORC_RAID_MD_POINT/hola.txt
ls $P3ASORC_RAID_MD_POINT
echo -e "I'd just like to interject for a moment. What you're referring to
as FreeBSD, is in fact, GNU/FreeBSD, or as I've recently taken to calling
it, GNU plus FreeBSD. FreeBSD is not an operating system unto itself,
but rather another free component of a fully functioning GNU system made
useful by the GNU corelibs, shell utilities and vital system components
comprising a full OS as defined by POSIX. \n\nMany computer users run a
modified version of the GNU system every day, without realizing it.
Through a peculiar turn of events, the version of GNU which is widely
used today is often called FreeBSD, and many of its users are not aware
that it is basically the GNU system, developed by the GNU Project.\n\n
There really is a FreeBSD, and these people are using it, but it is just
a part of the system they use. FreeBSD is the kernel: the program in the
system that allocates the machine's resources to the other programs that
you run. The kernel is an essential part of an operating system, but
useless by itself; it can only function in the context of a complete
operating system. FreeBSD is normally used in combination with the GNU
operating system: the whole system is basically GNU with FreeBSD added,
or GNU/FreeBSD. All the so-called FreeBSD distributions are really
distributions of GNU/FreeBSD!" > $P3ASORC_RAID_MD_POINT/hola.txt
ls $P3ASORC_RAID_MD_POINT
cat $P3ASORC_RAID_MD_POINT/hola.txt
zpool status $P3ASORC_RAID_POOL >> $P3ASORC_RAID_LOGTMP

# Ups, disco roto
echo -e 'Ups, disco roto\n' >> $P3ASORC_RAID_LOGTMP
# Quitar >>* para la corrección
zpool offline $P3ASORC_RAID_POOL $P3ASORC_RAID_OA >> $P3ASORC_RAID_LOGTMP
2>&1
zpool status $P3ASORC_RAID_POOL
zpool status $P3ASORC_RAID_POOL >> $P3ASORC_RAID_LOGTMP 2>&1
ls $P3ASORC_RAID_MD_POINT
cat $P3ASORC_RAID_MD_POINT/hola.txt

# Toca arreglarlo
echo -e 'Insertar\n' >> $P3ASORC_RAID_LOGTMP
zpool replace $P3ASORC_RAID_POOL $P3ASORC_RAID_OA $P3ASORC_RAID_OA >>
$P3ASORC_RAID_LOGTMP 2>&1

```

```
zpool online $P3ASORC_RAID_POOL $P3ASORC_RAID_OA >> $P3ASORC_RAID_LOGTMP  
    2>&1  
zpool status $P3ASORC_RAID_POOL  
zpool status $P3ASORC_RAID_POOL >> $P3ASORC_RAID_LOGTMP 2>&1
```

Fragmento de código 19: Configuración y comprobación de este servicio en  Debian GNU/Linux 13 (Trixie). Los *logs* pueden contrastar la comprobación de este servicio.

(ivan) 192.168.25.11 — Konsole

```
Nueva pestaña Dividir vista Capilar Pegar Buscar...
```

```
errors: No known data errors
root@bodhi: # zpool status $#3ASORC_RAID_POOL >> $#3ASORC_RAID_LOGTMP 2>1
pool: $#3ASORC_RAID_POOL
  state: DEGRADED
    status: One or more devices has been taken offline by the administrator.
           Sufficient replicas exist for the pool to continue functioning in a
           degraded state.
  action: Online the device using 'zpool online' or replace the device with
         'zpool replace'.
config:
  NAME        STATE      READ WRITE CKSUM
  mpool      DEGRADED   0  0  0
  raidz1-0   DEGRADED   0  0  0
  ad51       ONLINE     0  0  0
  ad62       ONLINE     0  0  0
  ad63       ONLINE     0  0  0
  ad64       ONLINE     0  0  0

errors: No known data errors
root@bodhi: # zpool status $#3ASORC_RAID_POOL >> $#3ASORC_RAID_LOGTMP 2>1
pool: $#3ASORC_RAID_POOL
  state: DEGRADED
    status: One or more devices has been taken offline by the administrator.
           Sufficient replicas exist for the pool to continue functioning in a
           degraded state.
  action: Online the device using 'zpool online' or replace the device with
         'zpool replace'.
config:
  NAME        STATE      READ WRITE CKSUM
  mpool      DEGRADED   0  0  0
  raidz1-0   DEGRADED   0  0  0
  ad51       OFFLINE    0  0  0
  ad62       ONLINE     0  0  0
  ad63       ONLINE     0  0  0
  ad64       ONLINE     0  0  0

errors: No known data errors
root@bodhi: # zpool online $#3ASORC_RAID_POOL $#3ASORC_RAID_0A >> $#3ASORC_RAID_LOGTMP 2>1
pool: $#3ASORC_RAID_POOL
  state: ONLINE
    status: The pool was resilvered.
    scan: resilvered 1B in 00:00:00 with 0 errors on Sat Nov 22 14:29:03 2025
config:
  NAME        STATE      READ WRITE CKSUM
  mpool      ONLINE     0  0  0
  raidz1-0   ONLINE     0  0  0
  ad61       ONLINE     0  0  0
  ad62       ONLINE     0  0  0
  ad63       ONLINE     0  0  0
  ad64       ONLINE     0  0  0

errors: No known data errors
```

Figura 48: Extracto de la salida producida de la ejecución de órdenes provistas en esta sección sobre ● FreeBSD 14.3. Conexión remota por SSH desde *host* al *guest* (● FreeBSD 14.3).

## 7.3. Microsoft Windows Server 2025 con diskmgmt.msc

Para la implementación de este servicio en el sistema operativo propietario, se ha optado por la utilización de **discos dinámicos** nativos a través de la utilidad Administración de discos (diskmgmt.msc). Se han empleado 4 unidades virtuales de 4,5 GB cada uno creadas desde VBox.

### 7.3.1. Comandos útiles (Inicialización y extracción)

```
# Comandos útiles

### Averiguar si un disco puede ser usado para RAID
Get-PhysicalDisk | Where-Object DeviceId -ne 0 | Select-Object FriendlyName,
    CanPool, CannotPoolReason

### Inicialización
Get-Disk | Where-Object Number -ne 0 | Set-Disk -IsOffline $false
Get-Disk | Where-Object Number -ne 0 | Set-Disk -IsReadOnly $false
Get-Disk | Where-Object Number -ne 0 | Clear-Disk -RemoveData -RemoveOEM -
    Confirm:$false
Get-Disk | Where-Object Number -ne 0 | Initialize-Disk -PartitionStyle GPT

### Exportar eventos
Get-EventLog -LogName System -After (Get-Date).AddHours(-2) |
    Where-Object { $_.Source -match "Disk|VDS|Ntfs" -and ($_.EntryType -
        match "Error|Warning") } |
    Select-Object TimeGenerated, Source, Message |
    Export-Csv -Path "D:\SUPERMEMORIA\windows\raid\windows.csv" -
        NoTypeInformation -Encoding UTF8
```

Fragmento de código 20: Inicialización y extracción de *logs* sobre este servicio en Microsoft Windows Server 2025.

### 7.3.2. Creación del arreglo

- Ejecutar diskmgmt.msc.
- Clic derecho sobre el espacio no asignado (banda negra) de uno de los discos > **Nuevo volumen RAID-5...**
- En el asistente, agregar los 4 discos disponibles a la lista de seleccionados.
- Asignar la letra de unidad F:.
- Etiquetar el volumen como md0 y marcar **Dar formato rápido**.
- Confirmar la conversión de los discos básicos a **dinámicos**.

### 7.3.3. Simulación de fallo

El forzado del fallo consistirá en «tirar del cable» en caliente una unidad de disco.

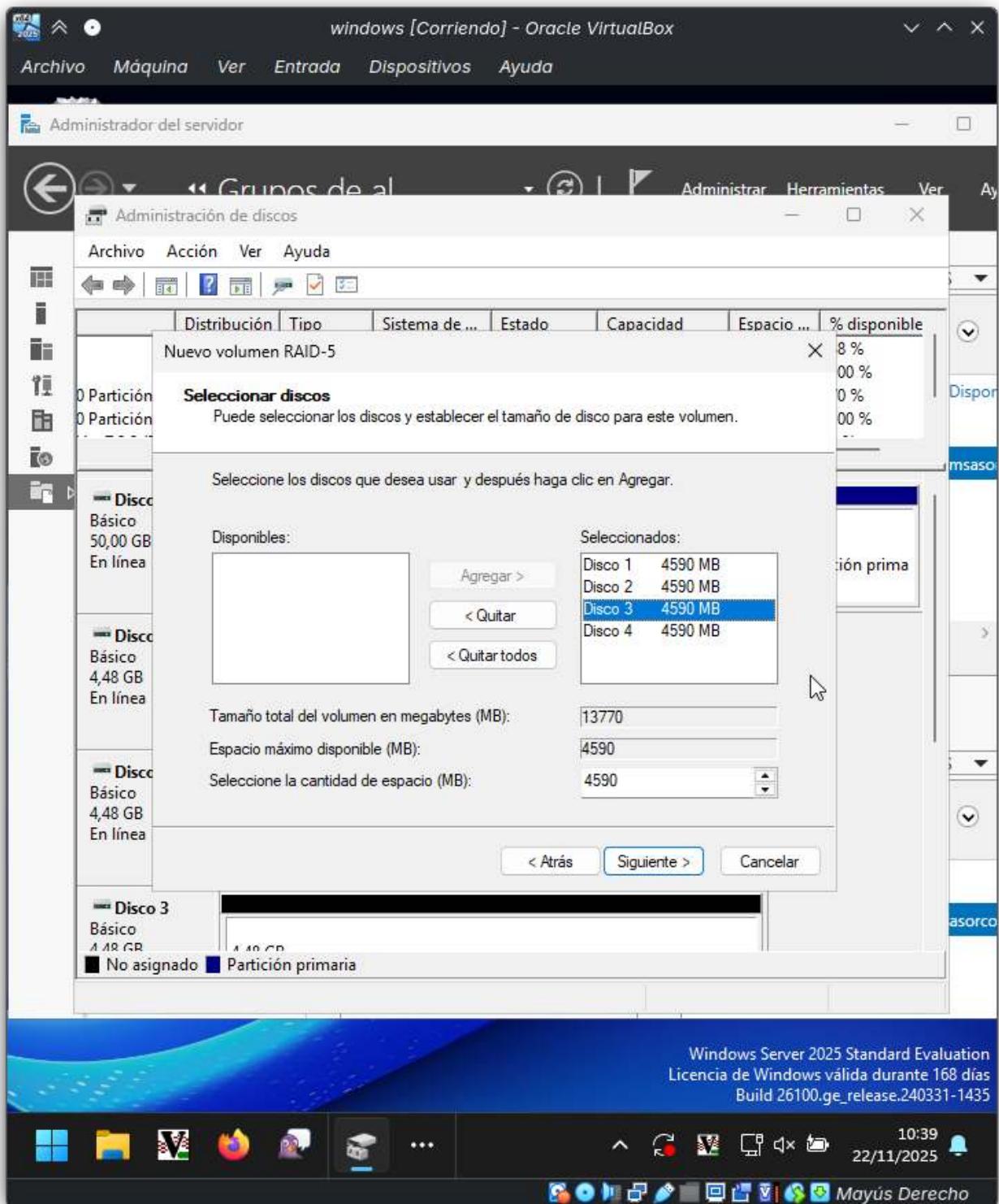


Figura 49: Asiste de RAID-5 en Microsoft Windows Server 2025 mostrando listado de unidades a convertir.

#### 7.3.4. Recuperación del fallo

Para esta demostración volveremos a conectar el dispositivo desde la pantalla de VBox mostrada anteriormente y a reactivarlo.

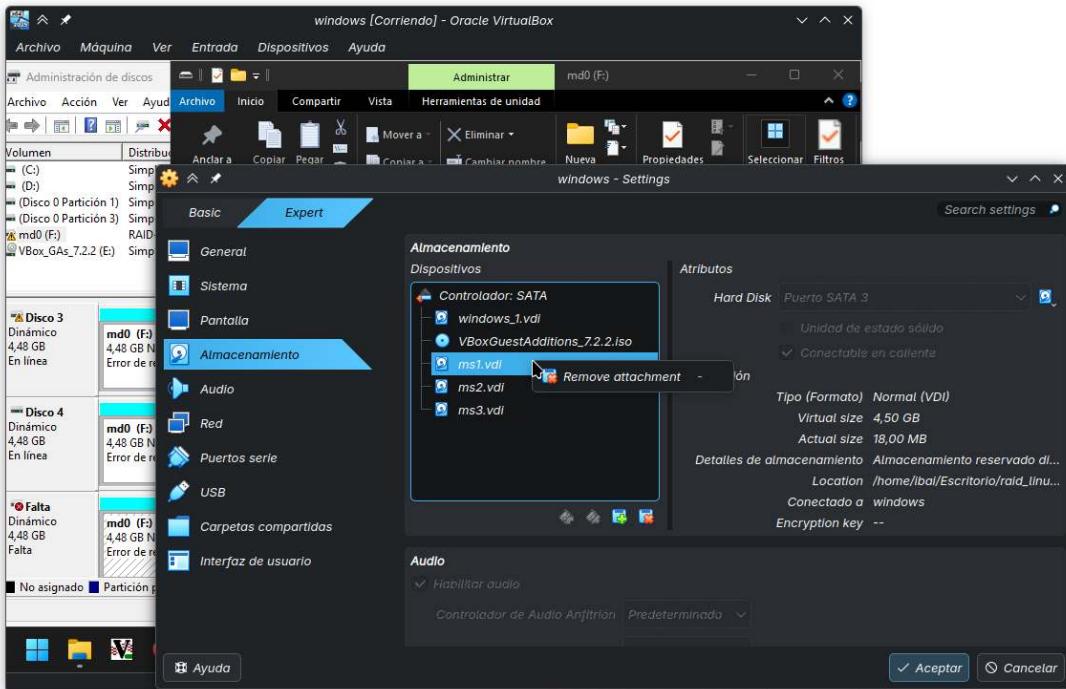


Figura 50: (Simulación) Captura de pantalla desde donde se aprecia la desconexión en caliente del dispositivo virtual. `ms0.vdi` ya ha sido desconectado.

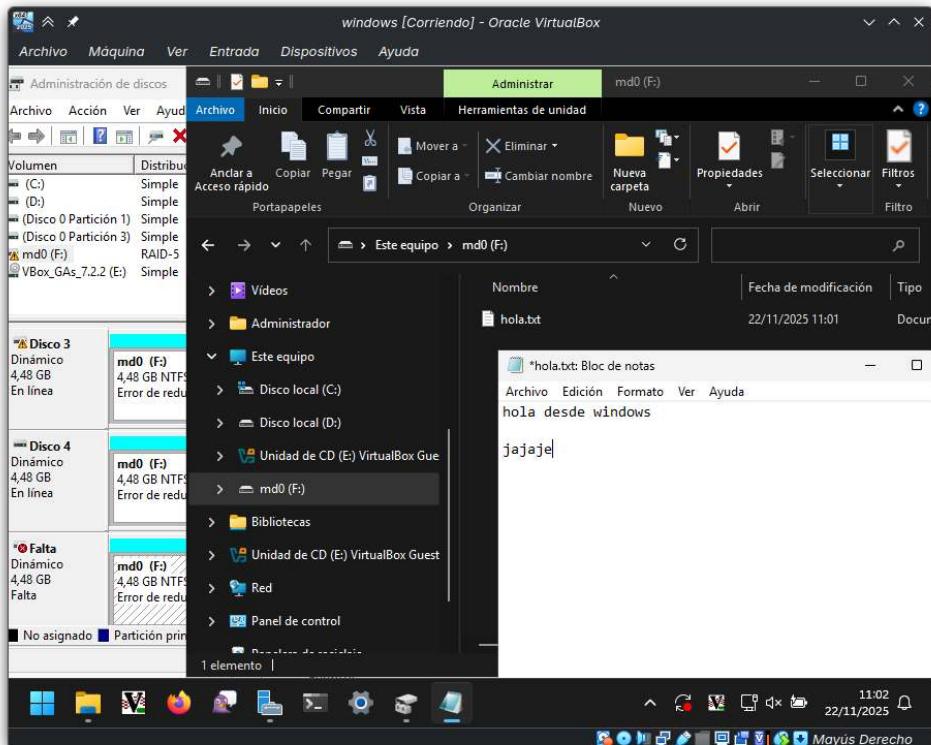


Figura 51: (Simulación) Incorrecto funcionamiento por errores de redundancia de RAID 5 en Microsoft Windows Server 2025.

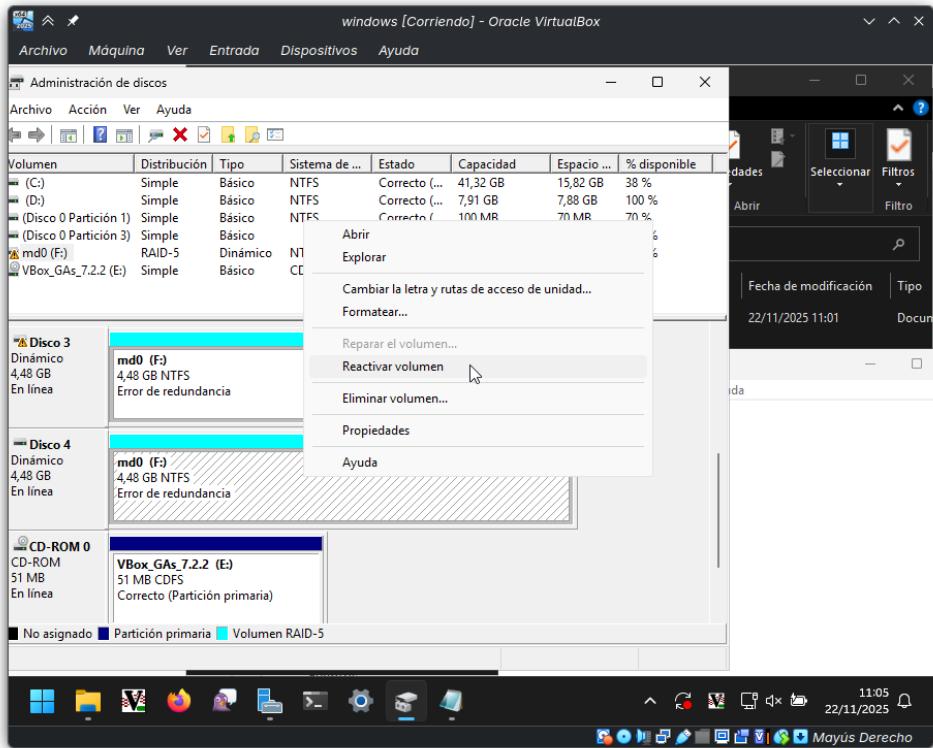


Figura 52: (Recuperación) Reactivación de disco dinámico de RAID 5 en Microsoft Windows Server 2025.

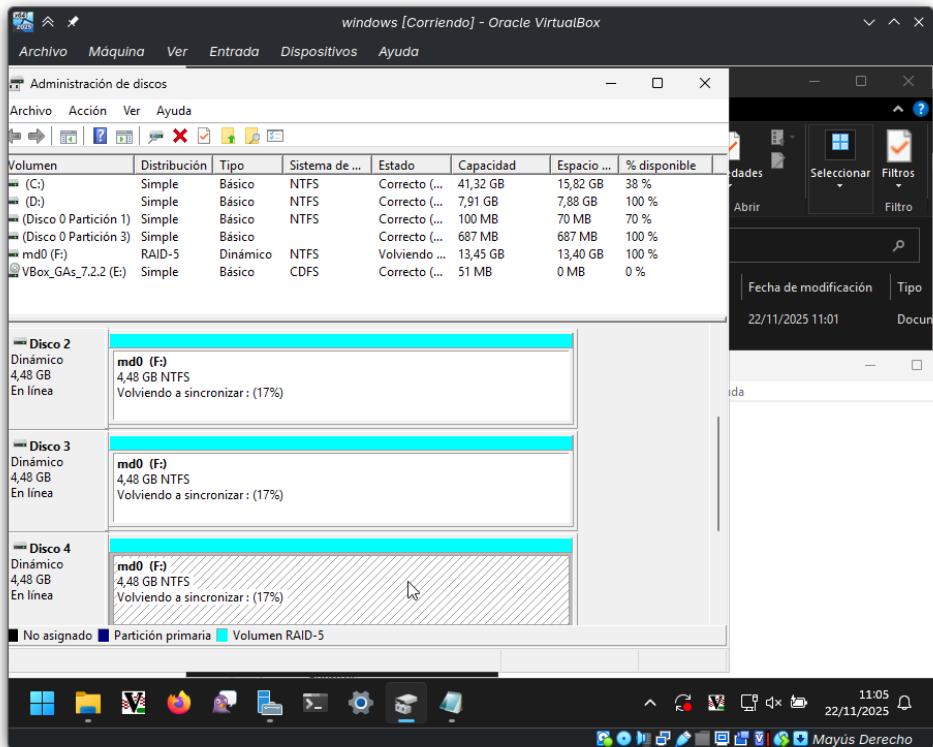


Figura 53: (Recuperación) Sincronización de volúmenes dinámicos en Microsoft Windows Server 2025.

## 8. Copias de seguridad

«*Backup (Absoluto, incremental, diferencial, sincronización). Se pide la creación de tres tipos de backup por sistema: El primero absoluto, el segundo incremental o diferencial y el tercero de sincronización utilizando rsync o equivalente en windows.*»

---

Las **Copias de Seguridad** (o *Backups*) son un proceso fundamental en la gestión de sistemas para garantizar la integridad y disponibilidad de la información ante desastres, fallos de HW, errores humanos o ataques informáticos. Consisten en la duplicación de datos en un medio de almacenamiento secundario para su posterior recuperación.

El servicio se basa habitualmente en una arquitectura **origen-destino** o centralizada:

- **El cliente (origen):** Es el sistema que contiene los datos originales. Ejecuta un agente o servicio que identifica los archivos modificados y los envía al servidor de respaldo.
- **El servidor de backup (destino):** Recibe los datos, gestiona el almacenamiento (rotación, compresión, deduplicación) y mantiene un catálogo de versiones para permitir la restauración (*restore*) en un punto temporal específico.

En cuanto a sus orígenes, la necesidad de respaldar información nació con la informática misma. En los años 50 y 60, se utilizaban tarjetas perforadas duplicadas. Con la llegada de las cintas magnéticas en los 70 y los discos duros masivos en los 80, surgieron protocolos como **rsync** (1996), creado por Andrew Tridgell y Paul Mackerras, que revolucionó la eficiencia al transferir solo las diferencias entre archivos.

Sobre uso, es una tarea crítica y obligatoria en cualquier entorno de producción. Se utiliza para proteger desde documentos de usuario hasta bases de datos completas. Estrategias como la regla 3-2-1 (3 copias, 2 medios distintos, 1 fuera de sitio) son estándar en la industria para asegurar la continuidad del negocio.

En el desarrollo de esta práctica, la implementación del sistema de copias de seguridad tiene un doble objetivo técnico:

1. **Automatización del respaldo:** Configurar una tarea programada que realice copias de seguridad incrementales o diferenciales de un directorio crítico a través de la red, asegurando la transferencia segura de los datos.
2. **Validación de restauración:** Demostrar la capacidad de recuperar la información desde el repositorio de backup ante una supuesta pérdida de datos, verificando la integridad de los archivos restaurados.

Para la corrección de este punto, se recomienda volver a ejecutar los *scripts*.

## 8.1. ☈ Debian GNU/Linux 13 (Trixie) con tar y rsync

```

(ivan) 192.168.25.10 — Konsole
└── a_dst
    └── COPIA_ASORC_2025-11-27.tar.gz
── d
    └── 1.txt
── d_dst
    └── COPIA_ASORC_2025-11-27.tar.gz
    └── personalizado_backup.log
── s
    └── 1.txt
── s_dst
    └── 1.txt

7 directories, 7 files
root@debian:~# rm -rf $P3ASORC_MEMORIA
mkdir -p $P3ASORC_MEMORIA
mkdir -p $P3ASORC_CONFIG

cp $P3ASORC_BACKUP_LOGFILE $P3ASORC_LOG
cp $P3ASORC_BACKUP_CRONFILE $P3ASORC_CONFIG/crontab_dump.txt

history > $P3ASORC_HISTORIAL
chmod 777 -$P3ASORC_MEMORIA

tree $P3ASORC_MEMORIA
/home/ivan/SUPERMEMORIA/backup
├── ficheros_configuracion
│   └── crontab_dump.txt
└── historylinux.txt
    └── linux.log

2 directories, 3 files
root@debian:~# systemctl status cron --no-pager
ls -LR $P3ASORC_BACKUP_PLACE | grep tar.gz
crontab -l
● cron.service - Regular background program processing daemon
    Loaded: loaded (/usr/lib/systemd/system/cron.service; enabled; preset: enabled)
    Active: active (running) since Thu 2025-11-27 09:19:32 CET; 1h 29min ago
      Invocation: fc91a0bc18e448cf8b99b7a564197812
        Docs: man:cron(8)
    Main PID: 2920 (cron)
      Tasks: 1 (limit: 4027)
        Memory: 980K (peak: 137.5M)
          CPU: 1.290s
        CGroup: /system.slice/cron.service
            └─2920 /usr/sbin/cron -f

nov 27 10:25:01 debian CRON[28229]: pam_unix(cron:session): session opened for us..id=0
nov 27 10:25:01 debian CRON[28231]: (daemon) CMD (test -x /usr/bin/udebsecan && /v...cron)
nov 27 10:25:02 debian CRON[28229]: pam_unix(cron:session): session closed for us..aemon
nov 27 10:30:01 debian CRON[30020]: pam_unix(cron:session): session opened for us..id=0
nov 27 10:30:01 debian CRON[30022]: (root) CMD (if [ -x /etc/init.d/anacron ] && ..; fi)
nov 27 10:30:01 debian CRON[30020]: pam_unix(cron:session): session closed for us.. root
nov 27 10:39:01 debian CRON[33333]: pam_unix(cron:session): session opened for us..id=0
nov 27 10:39:01 debian CRON[33335]: (root) CMD ( [ -x /usr/lib/php/sessionclean ..; fi)
nov 27 10:39:01 debian CRON[33333]: pam_unix(cron:session): session closed for us.. root
nov 27 10:49:01 debian cron[2920]: (root) RELOAD (crontabs/root)
Hint: Some lines were ellipsized, use -l to show in full.
-rw-rw-r-- 1 root root 164 nov 27 10:48 COPIA_ASORC_2025-11-27.tar.gz

```

Figura 54: Demostración desde *geust* empleando los diferentes mecanismos de copia de seguridad en ☈ Debian GNU/Linux 13 (Trixie).

```

#!/bin/sh
su -

# tar explicación
# -c: crear, -z: gzip, -v: verbose, -f: archivo, -p: preservar permisos
#-----
# Variables de entorno
#-----
P3ASORC_SERVICIO=backup
P3ASORC_SISTEMA=linux

P3ASORC_MEMORIA=/home/ivan/SUPERMEMORIA/$P3ASORC_SERVICIO
P3ASORC_CONFIG=$P3ASORC_MEMORIA/ficheros_configuracion
P3ASORC_LOG=$P3ASORC_MEMORIA/$P3ASORC_SISTEMA.log
P3ASORC_HISTORIAL=$P3ASORC_MEMORIA/history$P3ASORC_SISTEMA.txt

P3ASORC_BACKUP_PREFIX=COPIA_ASORC_
P3ASORC_BACKUP_PLACE=/var/COPIAS

```

```

P3ASORC_BACKUP_1=1.txt
P3ASORC_BACKUP_1_CONTENT=Hola
P3ASORC_BACKUP_2=2.txt
P3ASORC_BACKUP_2_CONTENT=Adios
P3ASORC_BACKUP_LOGFILE=$P3ASORC_BACKUP_PLACE/personalizado_backup.log
P3ASORC_BACKUP_CRONFILE=$P3ASORC_BACKUP_PLACE/crontab_backup
P3ASORC_BACKUP_A_SRC=$P3ASORC_BACKUP_PLACE/a
P3ASORC_BACKUP_A_DST=$P3ASORC_BACKUP_PLACE/a_dst
P3ASORC_BACKUP_D_SRC=$P3ASORC_BACKUP_PLACE/d
P3ASORC_BACKUP_D_DST=$P3ASORC_BACKUP_PLACE/d_dst
P3ASORC_BACKUP_S_SRC=$P3ASORC_BACKUP_PLACE/s
P3ASORC_BACKUP_S_DST=$P3ASORC_BACKUP_PLACE/s_dst

#-----
# Servicio
#-----
# PASO 0: Limpieza y datos iniciales

rm -rf $P3ASORC_BACKUP_PLACE
mkdir -p $P3ASORC_BACKUP_A_SRC
mkdir -p $P3ASORC_BACKUP_A_DST
mkdir -p $P3ASORC_BACKUP_D_SRC
mkdir -p $P3ASORC_BACKUP_D_DST
mkdir -p $P3ASORC_BACKUP_S_SRC
mkdir -p $P3ASORC_BACKUP_S_DST
echo "$P3ASORC_BACKUP_1_CONTENT" > $P3ASORC_BACKUP_A_SRC/$P3ASORC_BACKUP_1
echo "$P3ASORC_BACKUP_1_CONTENT" > $P3ASORC_BACKUP_D_SRC/$P3ASORC_BACKUP_1
echo "$P3ASORC_BACKUP_1_CONTENT" > $P3ASORC_BACKUP_S_SRC/$P3ASORC_BACKUP_1

#-----
# PASO 1: REALIZAR BACKUPS
#-----
# 1.a Copia Absoluta
tar -czvpf $P3ASORC_BACKUP_A_DST/$P3ASORC_BACKUP_PREFIX$(date +%F).tar.gz
$P3ASORC_BACKUP_A_SRC > $P3ASORC_BACKUP_LOGFILE 2>&1

# 1.b Copia Diferencial
tar -czvpf $P3ASORC_BACKUP_D_DST/$P3ASORC_BACKUP_PREFIX$(date +%F).tar.gz
$P3ASORC_BACKUP_D_SRC -N "$(date -d 'yesterday' +%F)" >>
$P3ASORC_BACKUP_LOGFILE 2>&1

# 1.c Copia Síncrona
rsync -av --delete $P3ASORC_BACKUP_S_SRC/ $P3ASORC_BACKUP_S_DST/ >>
$P3ASORC_BACKUP_LOGFILE 2>&1

#-----
# PASO 2: SIMULAR DESASTRE (Modificamos los datos)
#-----
rm -f $P3ASORC_BACKUP_A_SRC/$P3ASORC_BACKUP_1
rm -f $P3ASORC_BACKUP_D_SRC/$P3ASORC_BACKUP_1
rm -f $P3ASORC_BACKUP_S_SRC/$P3ASORC_BACKUP_1

echo "$P3ASORC_BACKUP_2_CONTENT" > $P3ASORC_BACKUP_A_SRC/$P3ASORC_BACKUP_2
echo "$P3ASORC_BACKUP_2_CONTENT" > $P3ASORC_BACKUP_D_SRC/$P3ASORC_BACKUP_2
echo "$P3ASORC_BACKUP_2_CONTENT" > $P3ASORC_BACKUP_S_SRC/$P3ASORC_BACKUP_2

#-----
# PASO 3: RESTAURAR

```

```

#-----#
rm -rf $P3ASORC_BACKUP_A_SRC/*
rm -rf $P3ASORC_BACKUP_D_SRC/*
rm -rf $P3ASORC_BACKUP_S_SRC/*

# 3.a Restauración Absoluta
P3ASORC_BACKUP_ARCHIVO_ABSOLUTO=$(ls -t $P3ASORC_BACKUP_A_DST/*.tar.gz |
    head -1)
tar -xzvf $P3ASORC_BACKUP_ARCHIVO_ABSOLUTO -C / >> $P3ASORC_BACKUP_LOGFILE
2>&1

# 3.b Restauración Diferencial
P3ASORC_BACKUP_ARCHIVO_DIFERENCIAL=$(ls -t $P3ASORC_BACKUP_D_DST/*.tar.gz |
    head -1)
tar -xzvf $P3ASORC_BACKUP_ARCHIVO_DIFERENCIAL -C / >>
$P3ASORC_BACKUP_LOGFILE 2>&1

# 3.c Restauración Síncrona
rsync -av --delete $P3ASORC_BACKUP_S_DST/ $P3ASORC_BACKUP_S_SRC/ >>
$P3ASORC_BACKUP_LOGFILE 2>&1

#-----#
# PASO 4: VER FINAL
#-----#
tree $P3ASORC_BACKUP_PLACE

#-----#
# PASO 5: CRON (Hacer backup los domingos a las 3am)
#-----#
crontab -l > $P3ASORC_BACKUP_CRONFILE 2>/dev/null
echo "0 3 * * 0 tar -czpf $P3ASORC_BACKUP_A_DST/FULL_\$(date +\%F).tar.gz
    $P3ASORC_BACKUP_A_SRC" >> mi_cron_temp
echo "0 3 * * 0 rsync -av --delete $P3ASORC_BACKUP_S_SRC/
    $P3ASORC_BACKUP_S_DST/" >> mi_cron_temp
crontab $P3ASORC_BACKUP_CRONFILE

# PASO 6 (OPCIONAL): Quitar cron
rm mi_cron_temp

#-----#
# Valida servicio
#-----#
systemctl status cron --no-pager
ls -lR $P3ASORC_BACKUP_PLACE | grep tar.gz
crontab -l

#-----#
# Extraer logs, configs e historial
#-----#
rm -rf $P3ASORC_MEMORIA
mkdir -p $P3ASORC_MEMORIA
mkdir -p $P3ASORC_CONFIG

cp $P3ASORC_BACKUP_LOGFILE $P3ASORC_LOG
cp $P3ASORC_BACKUP_CRONFILE $P3ASORC_CONFIG/crontab_dump.txt

history > $P3ASORC_HISTORIAL
chmod 777 -R $P3ASORC_MEMORIA

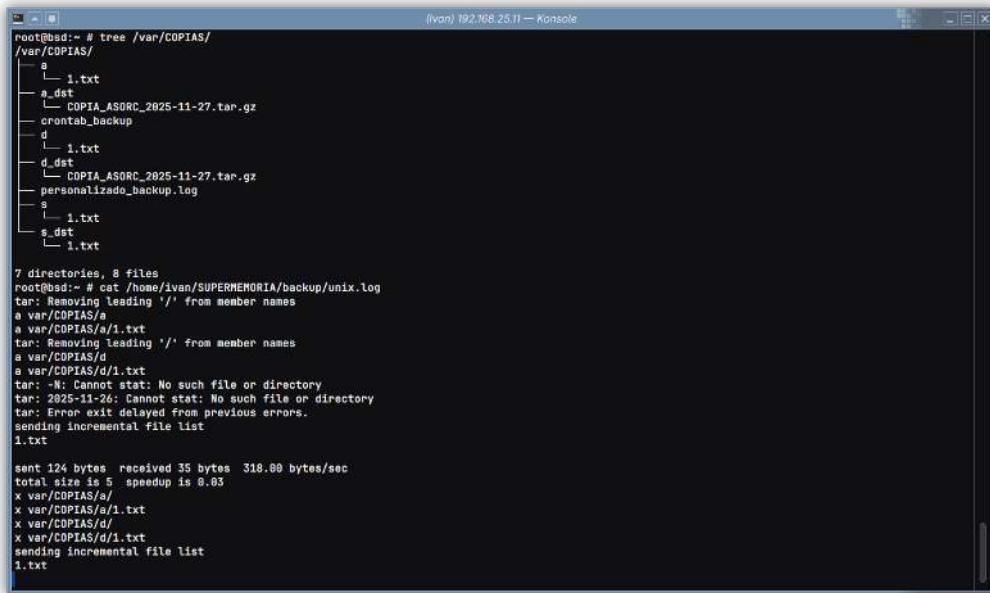
```

```
tree $P3ASORC_MEMORIA
```

Fragmento de código 21: Configuración y comprobación de este servicio en  Debian GNU/Linux 13 (Trixie).

## 8.2. ● FreeBSD 14.3 con tar y rsync

Procederemos con la misma mecánica que en  Debian GNU/Linux 13 (Trixie).



```
root@bsd:~# tree /var/COPIAS/
/var/COPIAS/
├── a
│   ├── 1.txt
│   └── s.dst
│       └── COPIA_ASORC_2025-11-27.tar.gz
├── crontab_backup
└── d
    ├── 1.txt
    └── d.dst
        ├── COPIA_ASORC_2025-11-27.tar.gz
        └── personalizado_backup.log
    └── s
        ├── 1.txt
        └── s.dst
            └── 1.txt

7 directories, 8 files
root@bsd:~# cat /home/ivan/SUPERMEMORIA/backup/unix.log
tar: Removing leading '/' from member names
a var/COPIAS/a
a var/COPIAS/a/1.txt
tar: Removing leading '/' from member names
a var/COPIAS/d
a var/COPIAS/d/1.txt
tar: -W: Cannot stat: No such file or directory
tar: 2025-11-26: Cannot stat: No such file or directory
tar: Error exit delayed from previous errors.
sending incremental file list
1.txt

sent 124 bytes received 35 bytes 318.00 bytes/sec
total size is 5 speedup is 0.03
x var/COPIAS/a/
x var/COPIAS/a/1.txt
x var/COPIAS/d/
x var/COPIAS/d/1.txt
sending incremental file list
1.txt
```

Figura 55: Demostración desde *geust* empleando los diferentes mecanismos de copia de seguridad en ● FreeBSD 14.3.

```
#!/bin/sh
su -

# tar explicación
# -c: crear, -z: gzip, -v: verbose, -f: archivo, -p: preservar permisos
#-----
# Variables de entorno
#-----
P3ASORC_SERVICIO=backup
P3ASORC_SISTEMA=unix

# NO TOCAR
P3ASORC_MEMORIA=/home/ivan/SUPERMEMORIA/$P3ASORC_SERVICIO
P3ASORC_CONFIG=$P3ASORC_MEMORIA/ficheros_configuracion
P3ASORC_LOG=$P3ASORC_MEMORIA/$P3ASORC_SISTEMA.log
P3ASORC_HISTORIAL=$P3ASORC_MEMORIA/history$P3ASORC_SISTEMA.txt
P3ASORC_BSD_RC=/etc/rc.conf

# ---
P3ASORC_BACKUP_PREFIX=COPIA_ASORC_
P3ASORC_BACKUP_PLACE=/var/COPIAS
P3ASORC_BACKUP_1=1.txt
```

```

P3ASORC_BACKUP_1_CONTENT=Hola
P3ASORC_BACKUP_2=2.txt
P3ASORC_BACKUP_2_CONTENT=Adios
P3ASORC_BACKUP_LOGFILE=$P3ASORC_BACKUP_PLACE/personalizado_backup.log
P3ASORC_BACKUP_CRONFILE=$P3ASORC_BACKUP_PLACE/crontab_backup
P3ASORC_BACKUP_A_SRC=$P3ASORC_BACKUP_PLACE/a
P3ASORC_BACKUP_A_DST=$P3ASORC_BACKUP_PLACE/a_dst
P3ASORC_BACKUP_D_SRC=$P3ASORC_BACKUP_PLACE/d
P3ASORC_BACKUP_D_DST=$P3ASORC_BACKUP_PLACE/d_dst
P3ASORC_BACKUP_S_SRC=$P3ASORC_BACKUP_PLACE/s
P3ASORC_BACKUP_S_DST=$P3ASORC_BACKUP_PLACE/s_dst

#-----
# Servicio
#-----

# PASO 0: Instalación, limpieza y datos iniciales
pkg install -y rsync tree
rm -rf $P3ASORC_BACKUP_PLACE
mkdir -p $P3ASORC_BACKUP_A_SRC
mkdir -p $P3ASORC_BACKUP_A_DST
mkdir -p $P3ASORC_BACKUP_D_SRC
mkdir -p $P3ASORC_BACKUP_D_DST
mkdir -p $P3ASORC_BACKUP_S_SRC
mkdir -p $P3ASORC_BACKUP_S_DST

echo "$P3ASORC_BACKUP_1_CONTENT" > $P3ASORC_BACKUP_A_SRC/$P3ASORC_BACKUP_1
echo "$P3ASORC_BACKUP_1_CONTENT" > $P3ASORC_BACKUP_D_SRC/$P3ASORC_BACKUP_1
echo "$P3ASORC_BACKUP_1_CONTENT" > $P3ASORC_BACKUP_S_SRC/$P3ASORC_BACKUP_1

#-----
# PASO 1: REALIZAR BACKUPS
#-----

# 1.a Copia Absoluta
tar -czvpf $P3ASORC_BACKUP_A_DST/$P3ASORC_BACKUP_PREFIX$(date +%F).tar.gz
$P3ASORC_BACKUP_A_SRC > $P3ASORC_BACKUP_LOGFILE 2>&1

# 1.b Copia Diferencial
FECHA_AYER=$(date -v-1d +%F)
tar -czvpf $P3ASORC_BACKUP_D_DST/$P3ASORC_BACKUP_PREFIX$(date +%F).tar.gz
$P3ASORC_BACKUP_D_SRC -N "$FECHA_AYER" >> $P3ASORC_BACKUP_LOGFILE 2>&1

# 1.c Copia Síncrona
rsync -av --delete $P3ASORC_BACKUP_S_SRC/ $P3ASORC_BACKUP_S_DST/ >>
$P3ASORC_BACKUP_LOGFILE 2>&1

#-----
# PASO 2: SIMULAR DESASTRE (Modificamos los datos)
#-----

rm -f $P3ASORC_BACKUP_A_SRC/$P3ASORC_BACKUP_1
rm -f $P3ASORC_BACKUP_D_SRC/$P3ASORC_BACKUP_1
rm -f $P3ASORC_BACKUP_S_SRC/$P3ASORC_BACKUP_1

echo "$P3ASORC_BACKUP_2_CONTENT" > $P3ASORC_BACKUP_A_SRC/$P3ASORC_BACKUP_2
echo "$P3ASORC_BACKUP_2_CONTENT" > $P3ASORC_BACKUP_D_SRC/$P3ASORC_BACKUP_2
echo "$P3ASORC_BACKUP_2_CONTENT" > $P3ASORC_BACKUP_S_SRC/$P3ASORC_BACKUP_2

#-----

```

```

# PASO 3: RESTAURAR
#-----
rm -rf $P3ASORC_BACKUP_A_SRC/*
rm -rf $P3ASORC_BACKUP_D_SRC/*
rm -rf $P3ASORC_BACKUP_S_SRC/*

# 3.a Restauración Absoluta
P3ASORC_BACKUP_ARCHIVO_ABSOLUTO=$(ls -t $P3ASORC_BACKUP_A_DST/*.tar.gz |
    head -1)
tar -xzvf $P3ASORC_BACKUP_ARCHIVO_ABSOLUTO -C / >> $P3ASORC_BACKUP_LOGFILE
2>&1

# 3.b Restauración Diferencial
P3ASORC_BACKUP_ARCHIVO_DIFERENCIAL=$(ls -t $P3ASORC_BACKUP_D_DST/*.tar.gz |
    head -1)
tar -xzvf $P3ASORC_BACKUP_ARCHIVO_DIFERENCIAL -C / >>
$P3ASORC_BACKUP_LOGFILE 2>&1

# 3.c Restauración Síncrona
rsync -av --delete $P3ASORC_BACKUP_S_DST/ $P3ASORC_BACKUP_S_SRC/ >>
$P3ASORC_BACKUP_LOGFILE 2>&1

#-----
# PASO 4: VER FINAL
#-----
tree $P3ASORC_BACKUP_PLACE

#-----
# PASO 5: CRON (Hacer backup los domingos a las 3am)
#-----
crontab -l > $P3ASORC_BACKUP_CRONFILE 2>/dev/null
echo "0 3 * * 0 tar -czpf $P3ASORC_BACKUP_A_DST/FULL_\$(date +\%F).tar.gz
    $P3ASORC_BACKUP_A_SRC" >> $P3ASORC_BACKUP_CRONFILE
echo "0 3 * * 0 rsync -av --delete $P3ASORC_BACKUP_S_SRC/
    $P3ASORC_BACKUP_S_DST/" >> $P3ASORC_BACKUP_CRONFILE
crontab $P3ASORC_BACKUP_CRONFILE

#-----
# Valida servicio
#-----
service cron status
ls -lR $P3ASORC_BACKUP_PLACE | grep tar.gz
crontab -l

#-----
# Extraer logs, configs e historial
#-----
rm -rf $P3ASORC_MEMORIA
mkdir -p $P3ASORC_MEMORIA
mkdir -p $P3ASORC_CONFIG

cp $P3ASORC_BACKUP_LOGFILE $P3ASORC_LOG
cp $P3ASORC_BACKUP_CRONFILE $P3ASORC_CONFIG/crontab_dump.txt
cp $P3ASORC_BSD_RC $P3ASORC_CONFIG/rc.conf

history > $P3ASORC_HISTORIAL
chmod 777 -R $P3ASORC_MEMORIA

```

```
tree $P3ASORC_MEMORY
```

Fragmento de código 22: Configuración y comprobación de este servicio en ● FreeBSD 14.3.

### 8.3. Microsoft Windows Server 2025 con tar y rsync

En este caso simularemos el flujo del *backup* instalando **MYSYS2** que emula los comandos usados en ☈ Debian GNU/Linux 13 (Trixie) y ● FreeBSD 14.3.

```
#!/bin/sh

##### DESCARGAR EL MYSYS2
powershell
C:/msys64/msys2_shell.cmd -defterm -here -no-start -ucrt64

# tar explicación
# -c: crear, -z: gzip, -v: verbose, -f: archivo, -p: preservar permisos
#-----
# Variables de entorno
#-----
P3ASORC_BACKUP_PREFIX=COPIA_ASORC_
# Nota: /var/COPIAS estará dentro de C:\msys64\var\COPIAS
P3ASORC_BACKUP_PLACE=/var/COPIAS
P3ASORC_BACKUP_1=1.txt
P3ASORC_BACKUP_1_CONTENT=Hola
P3ASORC_BACKUP_2=2.txt
P3ASORC_BACKUP_2_CONTENT=Adios
P3ASORC_BACKUP_LOGFILE=$P3ASORC_BACKUP_PLACE/personalizado_backup.log
P3ASORC_BACKUP_CRONFILE=$P3ASORC_BACKUP_PLACE/crontab_backup
P3ASORC_BACKUP_A_SRC=$P3ASORC_BACKUP_PLACE/a
P3ASORC_BACKUP_A_DST=$P3ASORC_BACKUP_PLACE/a_dst
P3ASORC_BACKUP_D_SRC=$P3ASORC_BACKUP_PLACE/d
P3ASORC_BACKUP_D_DST=$P3ASORC_BACKUP_PLACE/d_dst
P3ASORC_BACKUP_S_SRC=$P3ASORC_BACKUP_PLACE/s
P3ASORC_BACKUP_S_DST=$P3ASORC_BACKUP_PLACE/s_dst

#-----
# Servicio
#-----
# PASO 0: Limpieza y datos iniciales
pacman -S --noconfirm --needed rsync tree tar
rm -rf $P3ASORC_BACKUP_PLACE
mkdir -p $P3ASORC_BACKUP_A_SRC
mkdir -p $P3ASORC_BACKUP_A_DST
mkdir -p $P3ASORC_BACKUP_D_SRC
mkdir -p $P3ASORC_BACKUP_D_DST
mkdir -p $P3ASORC_BACKUP_S_SRC
mkdir -p $P3ASORC_BACKUP_S_DST

echo "$P3ASORC_BACKUP_1_CONTENT" > $P3ASORC_BACKUP_A_SRC/$P3ASORC_BACKUP_1
echo "$P3ASORC_BACKUP_1_CONTENT" > $P3ASORC_BACKUP_D_SRC/$P3ASORC_BACKUP_1
echo "$P3ASORC_BACKUP_1_CONTENT" > $P3ASORC_BACKUP_S_SRC/$P3ASORC_BACKUP_1

#-----
# PASO 1: REALIZAR BACKUPS
#-----
# 1.a Copia absoluta
```

```

tar -czvpf $P3ASORC_BACKUP_A_DST/$P3ASORC_BACKUP_PREFIX$(date +%F).tar.gz
$P3ASORC_BACKUP_A_SRC > $P3ASORC_BACKUP_LOGFILE

# 1.b Copia diferencial
tar -czvpf $P3ASORC_BACKUP_D_DST/$P3ASORC_BACKUP_PREFIX$(date +%F).tar.gz
$P3ASORC_BACKUP_D_SRC -N "$(date -d 'yesterday' +%F)" >>
$P3ASORC_BACKUP_LOGFILE

# 1.c Copia síncrona
rsync -av --delete $P3ASORC_BACKUP_S_SRC/ $P3ASORC_BACKUP_S_DST/ >>
$P3ASORC_BACKUP_LOGFILE

#-----
# PASO 2: SIMULAR DESASTRE (Modificamos los datos)
#-----
rm -f $P3ASORC_BACKUP_A_SRC/$P3ASORC_BACKUP_1
rm -f $P3ASORC_BACKUP_D_SRC/$P3ASORC_BACKUP_1
rm -f $P3ASORC_BACKUP_S_SRC/$P3ASORC_BACKUP_1

echo "$P3ASORC_BACKUP_2_CONTENT" > $P3ASORC_BACKUP_A_SRC/$P3ASORC_BACKUP_2
echo "$P3ASORC_BACKUP_2_CONTENT" > $P3ASORC_BACKUP_D_SRC/$P3ASORC_BACKUP_2
echo "$P3ASORC_BACKUP_2_CONTENT" > $P3ASORC_BACKUP_S_SRC/$P3ASORC_BACKUP_2

tree $P3ASORC_BACKUP_PLACE

#-----
# PASO 3: RESTAURAR
#-----
rm -rf $P3ASORC_BACKUP_A_SRC/*
rm -rf $P3ASORC_BACKUP_D_SRC/*
rm -rf $P3ASORC_BACKUP_S_SRC/*

# 3.a Restauración absoluta
P3ASORC_BACKUP_ARCHIVO_ABSOLUTO=$(ls -t $P3ASORC_BACKUP_A_DST/*.tar.gz |
head -1)
tar -xzvf $P3ASORC_BACKUP_ARCHIVO_ABSOLUTO -C / >> $P3ASORC_BACKUP_LOGFILE
2>&1

# 3.b Restauración diferencial
P3ASORC_BACKUP_ARCHIVO_DIFERENCIAL=$(ls -t $P3ASORC_BACKUP_D_DST/*.tar.gz |
head -1)
tar -xzvf $P3ASORC_BACKUP_ARCHIVO_DIFERENCIAL -C / >>
$P3ASORC_BACKUP_LOGFILE 2>&1

# 3.c Restauración síncrona
rsync -av --delete $P3ASORC_BACKUP_S_DST/ $P3ASORC_BACKUP_S_SRC/ >>
$P3ASORC_BACKUP_LOGFILE 2>&1

#-----
# PASO 4: VER FINAL
#-----
tree $P3ASORC_BACKUP_PLACE

echo "Copiar $P3ASORC_BACKUP_LOGFILE"

```

Fragmento de código 23: Configuración y comprobación de este servicio en Microsoft Windows Server 2025.

```

# 1.a Copia absoluta
tar -czvpf $P3ASORC_BACKUP_A_DST/$P3ASORC_BACKUP_PREFIX$(date +%F).tar.gz $P3ASORC_B
ACKUP_A_SRC > $P3ASORC_BACKUP_LOGFILE

# 1.b Copia diferencial
tar -czvpf $P3ASORC_BACKUP_D_DST/$P3ASORC_BACKUP_PREFIX$(date +%F).tar.gz $P3ASORC_B
tree $P3ASORC_BACKUP_PLACE-----P_S_SRC/ >> $P3ASORC_BACKUP_
advertiscencia: rsync-3.4.1-1 está actualizado -- omitiéndolo
advertiscencia: tree-2.2.1-1 está actualizado -- omitiéndolo
advertiscencia: tar-1.35-2 está actualizado -- omitiéndolo
...el sistema ya está actualizado.
tar: Eliminando la `/' inicial de los nombres
tar: Opción --after-date: Se trata la fecha '2025-11-26' como 2025-11-26 00:00:00
tar: Eliminando la `/' inicial de los nombres
/var/COPIAS
├── a
│   └── 2.txt
├── a_dst
│   └── COPIA_ASORC_2025-11-27.tar.gz
├── d
│   └── 2.txt
├── d_dst
│   └── COPIA_ASORC_2025-11-27.tar.gz
└── personalizado_backup.log
    └── s
        ├── 2.txt
        └── s_dst
            └── 1.txt

7 directories, 7 files
/var/COPIAS
├── a
│   └── 1.txt
├── a_dst
│   └── COPIA_ASORC_2025-11-27.tar.gz
├── d
│   └── 1.txt
├── d_dst
│   └── COPIA_ASORC_2025-11-27.tar.gz
└── personalizado_backup.log
    └── s
        ├── 1.txt
        └── s_dst
            └── 1.txt

7 directories, 7 files
Administrador@msasorcorg UCRT64 ~
#

```

Figura 56: Demostración desde *geust* empleando los diferentes mecanismos de copia de seguridad en Microsoft Windows Server 2025.

## 8.4. Microsoft Windows Server 2025 con Cobian Backup 11

Alternativamente, podemos hacer el *backup* utilizando un SW externo como lo es en esta ocasión Cobian. A continuación, se detalla la guía paso a paso.

### 8.4.1. Instalación del servicio

En un servidor, es crítico que el programa funcione aunque no haya ningún usuario *logueado*. Tras descargar Cobian desde su web oficial y durante la instalación de esta programa, será preciso marcar el tipo de la instalación seleccionando **Como servicio (As a service)**. Emplearemos las credenciales del administrador para simplificar el proceso.

### 8.4.2. *Backup* absoluto

Este respaldo copia **todo** cada vez que se ejecuta. Procediendo así:

1. Tarea → Nueva tarea.

2. **Pestaña General:**

- **Nombre:** «ABSOLUTO».
- **Tipo de respaldo:** Escoger **Completo**.
- **Crear respaldos separados:** MARCADO.

3. **Pestaña Ficheros:** Añadir el origen y el destino.

4. **Pestaña Horario:** Por ejemplo, Semanal (los viernes a las 22:00).

5. **Pestaña Archivo (Archive):** Seleccionar 7zip.

#### 8.4.3. *Backup* diferencial

1. Tarea → Nueva tarea.

2. **Pestaña General:**

- **Nombre:** «DIFERENCIAL».
- **Tipo de respaldo:** Escoger **Diferencial**.
- **Crear respaldos separados:** MARCADO.

3. **Pestaña Ficheros:** Añadir el origen y el destino.

4. **Pestaña Horario:** Por ejemplo, Semanal (los viernes a las 22:00).

5. **Pestaña Archivo:** Seleccionar 7zip.

#### 8.4.4. *Backup* de sincronización

Lo efectuaríamos de esta forma:

1. Tarea → Nueva tarea.

2. **Pestaña General:**

- **Nombre:** «SINCRONIZACIÓN».
- **Tipo de respaldo:** Escoger **Completo**.
- **Crear respaldos separados:** DESMARCADO.

3. **Pestaña Ficheros:** Añadir el origen y el destino.

4. **Pestaña Horario:** Por ejemplo, Semanal (los viernes a las 22:00).

5. **Pestaña Avanzado:** Marcar la casilla **Tarea de espejo**.

6. **Pestaña Archivo (Archive):** Sin compresión.

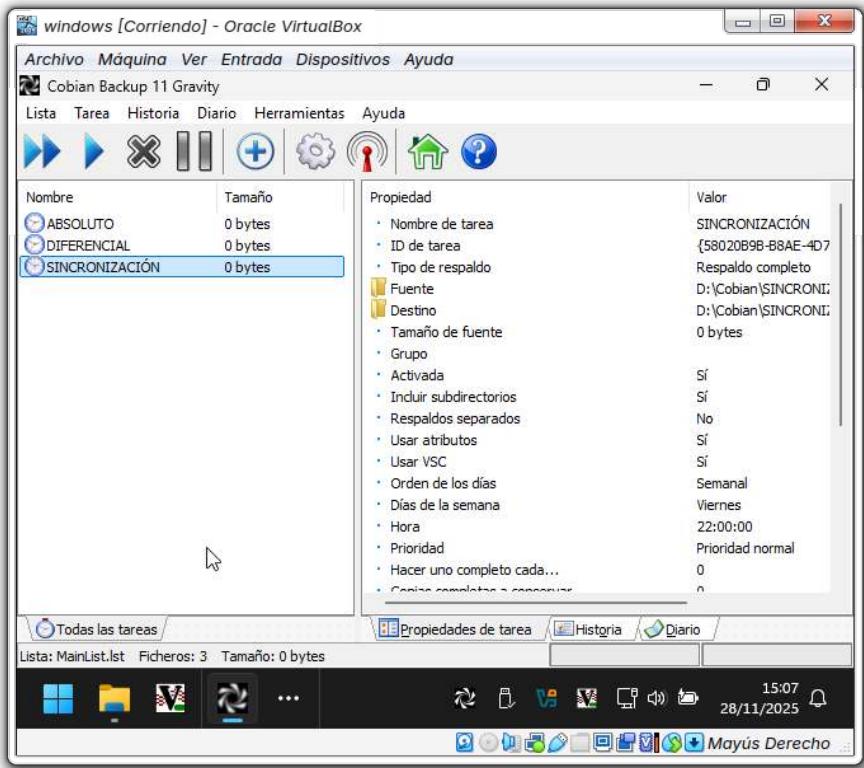


Figura 57: Demostración desde *geust* empleando los diferentes mecanismos de copia de seguridad en Microsoft Windows Server 2025.

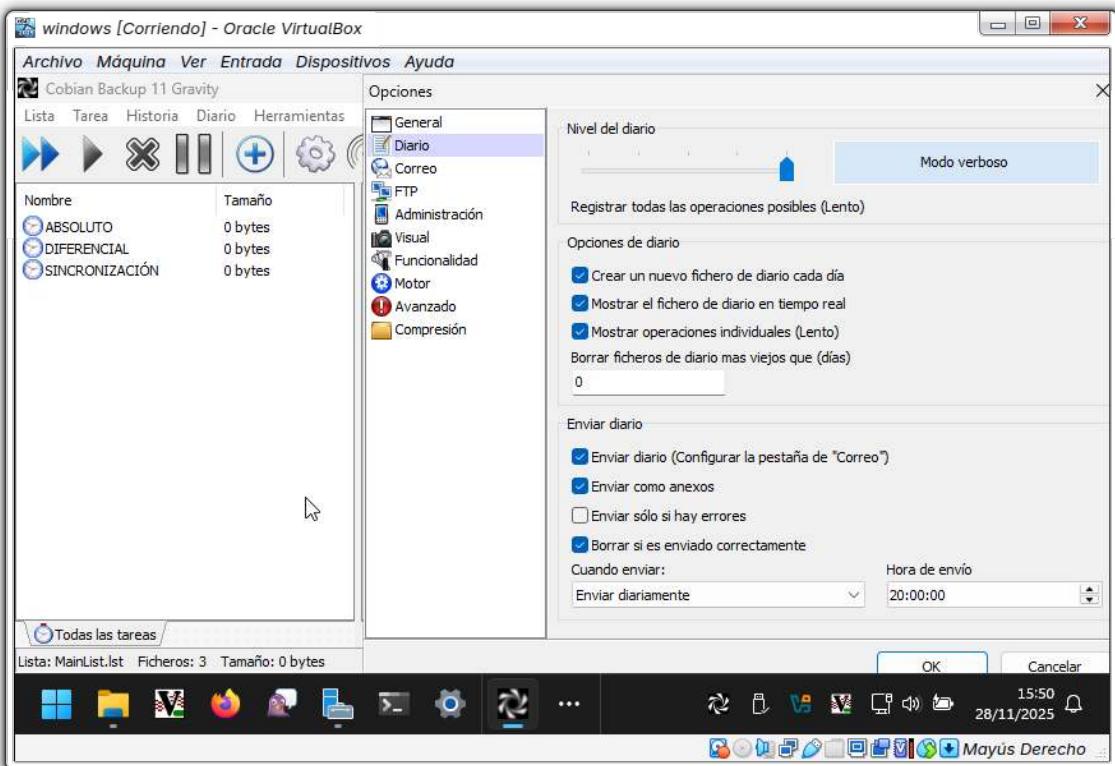


Figura 58: Activación del modo verbose en Cobian. Accediendo desde Herramientas → Opciones

## 9. Correo electrónico

«Servidor de correo electrónico. Se pide instalar un servidor de correo electrónico (SMTP) con servicio antivirus y anti-spam (implementando listas negras y grises). Así mismo se pide la configuración de un cliente web tipo Roundcube o Squirrelmail. En Linux/Unix podrá utilizarse bien Sendmail o bien Postfix.»

---

El **Servidor de Correo** es el componente de infraestructura encargado de enrutar, entregar y almacenar mensajes de correo electrónico. Es uno de los servicios más antiguos y esenciales de Internet, basado en protocolos estándar abiertos.

El servicio se estructura en dos roles principales que suelen operar conjuntamente:

- **MTA (*Mail Transfer Agent*):** Como Postfix, es el «cartero». Se encarga de enviar y recibir correos entre servidores utilizando el protocolo **SMTP** (*Simple Mail Transfer Protocol*).
- **MDA/MRA (*Mail Delivery Agent*):** Como Dovecot, es el «buzón». Almacena los mensajes y permite que los usuarios accedan a ellos mediante protocolos como **IMAP** o **POP3**.

En cuanto a sus orígenes, el correo electrónico antecede a Internet. El primer sistema de envío de mensajes entre usuarios de una misma máquina data de 1965 en el MIT. Sin embargo, el protocolo SMTP fue definido en **1982** (RFC 821) por Jonathan Postel, estableciendo el estándar de comunicación que perdura hasta hoy. **Postfix**, utilizado en esta práctica, fue creado por Wietse Venema en IBM en 1997 como una alternativa segura y rápida al clásico Sendmail.

Sobre uso, a pesar del auge de la mensajería instantánea, el correo electrónico sigue siendo el estándar oficial para la comunicación corporativa, notificaciones de sistemas y validación de identidades digitales.

En el desarrollo de esta práctica, la implementación del servidor de correo persigue los siguientes objetivos técnicos:

1. **Configuración SMTP/IMAP:** Desplegar un MTA funcional para el enrutamiento de mensajes y un servicio IMAP para el acceso de usuarios.
2. **Interoperabilidad de clientes:** Validar el ciclo completo de comunicación (envío y recepción) utilizando clientes de escritorio estándar (*MUA - Mail User Agent*) como Thunderbird, asegurando la correcta autenticación y entrega local.

**Nota:** Cabe recalcar que es posible que algunos puertos deban ser movidos.

### 9.1. Debian GNU/Linux 13 (Trixie)

```
#!/bin/sh
su -
#-----
# Variables de entorno
#-----
P3AS0RC_SERVICIO=mail
```

```

P3ASORC_SISTEMA=linux

# NO TOCAR
P3ASORC_MEMORIA=/home/ivan/SUPERMEMORIA/$P3ASORC_SERVICIO
P3ASORC_CONFIG=$P3ASORC_MEMORIA/ficheros_configuracion
P3ASORC_LOG=$P3ASORC_MEMORIA/$P3ASORC_SISTEMA.log
P3ASORC_HISTORIAL=$P3ASORC_MEMORIA/history$P3ASORC_SISTEMA.txt

P3ASORC_MAIL_HOSTNAME=debianasorc.org
P3ASORC_MAIL_MAIL_HOSTNAME=mail.$P3ASORC_MAIL_HOSTNAME
P3ASORC_MAIL_OSUSER=mailusuario1
P3ASORC_MAIL_OSUSER_PASS=1
P3ASORC_MAIL_OSUSER1=mailusuario2
P3ASORC_MAIL_OSUSER1_PASS=1
P3ASORC_MAIL_CONFIG=/etc/postfix/main.cf
P3ASORC_MAIL_CONFIG2=/etc/dovecot/dovecot.conf
P3ASORC_MAIL_CONFIG3=/etc/dovecot/conf.d/10-auth.conf
P3ASORC_MAIL_CONFIG4=/etc/dovecot/conf.d/10-mail.conf
P3ASORC_MAIL_CONFIG5=/etc/dovecot/conf.d/10-master.conf
P3ASORC_MAIL_CONFIG6=/etc/profile.d/mail.sh
P3ASORC_MAIL_CONFIG7=/etc/roundcube/config.inc.php
P3ASORC_MAIL_CONFIG8=/etc/apache2/conf-enabled/roundcube.conf
P3ASORC_MAIL_CONFIG9=/etc/mailname
P3ASORC_MAIL_CONFIG10=/etc/amavis/conf.d/15-content_filter_mode
P3ASORC_MAIL_CONFIG11=/etc/postfix/master.cf
P3ASORC_MAIL_CONFIG_PORT_MAIL=25
P3ASORC_MAIL_CONFIG_PORT_FILTER1=10024
P3ASORC_MAIL_CONFIG_PORT_FILTER2=10025

#-----
# Servicio
#-----
# PASO 0: Limpieza
apt purge -y postfix dovecot* roundcube* clamav* amavis* spamassassin* spamd
rm -rf /var/lib/roundcube /var/lib/dovecot /etc/dovecot

# PASO 1: Paquetería
apt update
apt install -y postfix clamav-daemon spamassassin amavis-new dovecot-imapd
    roundcube-core mailutils dovecot-core roundcube clamav roundcube-sqlite3
    nagios4* spamd

EEEEEEEEE=$DEBIAN_FRONTEND
export DEBIAN_FRONTEND=noninteractive
echo "postfix postfix/main_mailer_type select Internet Site" | debconf-set-
    selections
echo "postfix postfix/mailname string $P3ASORC_MAIL_MAIL_HOSTNAME" | debconf-
    set-selections
echo "roundcube-core roundcube/dbconfig-install boolean true" | debconf-set-
    selections
echo "roundcube-core roundcube/database-type select sqlite3" | debconf-set-
    selections
export DEBIAN_FRONTEND=$EEEEEEEEE

systemctl enable spamd
systemctl start spamd
systemctl enable clamav-freshclam
systemctl start clamav-freshclam

```

```

# PASO 2: Configurar IMAP de formas muy místicas
sed -i "s/^myhostname =.*$/myhostname = $P3ASORC_MAIL_MAIL_HOSTNAME/" $P3ASORC_MAIL_CONFIG

if grep -q "^mydomain =" $P3ASORC_MAIL_CONFIG; then
    sed -i "s/^mydomain =.*$/mydomain = $P3ASORC_MAIL_MAIL_HOSTNAME/" $P3ASORC_MAIL_CONFIG
else
    echo "mydomain = $P3ASORC_MAIL_MAIL_HOSTNAME" >> $P3ASORC_MAIL_CONFIG
fi

sed -i "s/^#*home_mailbox =.*$/home_mailbox = Maildir\//"
$P3ASORC_MAIL_CONFIG
sed -i "s/^mydestination =.*$/mydestination = \$myhostname, \$mydomain, localhost.\$myhostname, localhost.\$mydomain, localhost/"
$P3ASORC_MAIL_CONFIG
echo "$P3ASORC_MAIL_MAIL_HOSTNAME" > $P3ASORC_MAIL_CONFIG9
systemctl restart postfix

# PASO 3: Correspondiente buzón
sed -i 's|^#*mail_location =.*|mail_location = maildir:/Maildir|' $P3ASORC_MAIL_CONFIG4
if grep -q "^#*protocols =" $P3ASORC_MAIL_CONFIG2; then
    sed -i 's/^#*protocols =.*$/protocols = imap/' $P3ASORC_MAIL_CONFIG2
else
    echo "protocols = imap" >> $P3ASORC_MAIL_CONFIG2
fi

# PASO 4: Autenticación
systemctl restart dovecot
sed -i 's|^#Alias /roundcube|Alias /roundcube|' $P3ASORC_MAIL_CONFIG8
sed -i "s|\$\$config['smtp_host'].*|\$\$config['smtp_host'] = 'localhost
:25';|" $P3ASORC_MAIL_CONFIG7
sed -i "s|\$\$config['smtp_user'].*|\$\$config['smtp_user'] = '';|" $P3ASORC_MAIL_CONFIG7
sed -i "s|\$\$config['smtp_pass'].*|\$\$config['smtp_pass'] = '';|" $P3ASORC_MAIL_CONFIG7
systemctl restart apache2

# PASO 5: Permisos y usuarios
crear_usuario() {
    local USER=$1
    local PASS=$2
    userdel -r $USER 2>/dev/null
    adduser --disabled-password --gecos "" $USER
    echo "$USER:$PASS" | chpasswd
    mkdir -p /home/$USER/Maildir/{cur,new,tmp}
    chown -R $USER:$USER /home/$USER/Maildir/
    echo "Usuario $USER creado correctamente."
}

crear_usuario "$P3ASORC_MAIL_OSUSER" "$P3ASORC_MAIL_OSUSER_PASS"
crear_usuario "$P3ASORC_MAIL_OSUSER1" "$P3ASORC_MAIL_OSUSER1_PASS"

# PASO 6: Antivirus + SPAM
cat << 'EOF' > $P3ASORC_MAIL_CONFIG10
use strict;

```

```

@bypass_virus_checks_maps = (
    \%bypass_virus_checks , \@bypass_virus_checks_acl , \
    $bypass_virus_checks_re);

@bypass_spam_checks_maps = (
    \%bypass_spam_checks , \@bypass_spam_checks_acl , \$bypass_spam_checks_re);

1;
EOF

postconf -e "content_filter = smtp-amavis:[127.0.0.1]:
    $P3ASORC_MAIL_CONFIG_PORT_FILTER1"
cat << EOF >> $P3ASORC_MAIL_CONFIG11

smtp-amavis unix - - n - 2 smtp
  -o smtp_data_done_timeout=1200
  -o smtp_send_xforward_command=yes
  -o disable_dns_lookups=yes

127.0.0.1:$P3ASORC_MAIL_CONFIG_PORT_FILTER2 inet n - n -
  - smptd
  -o content_filter=
  -o local_recipient_maps=
  -o relay_recipient_maps=
  -o smtpd_restriction_classes=
  -o smtpd_client_restrictions=
  -o smtpd_helo_restrictions=
  -o smtpd_sender_restrictions=
  -o smtpd_recipient_restrictions=permit_mynetworks,reject
  -o mynetworks=127.0.0.0/8
  -o strict_rfc821_envelopes=yes
  -o smtpd_error_sleep_time=0
  -o smtpd_soft_error_limit=1001
  -o smtpd_hard_error_limit=1000
  -o receive_override_options=no_milters
EOF

usermod -aG amavis clamav
systemctl restart clamav-daemon amavis postfix

#-----
# Extraer logs, configs e historial
#-----

rm -rf $P3ASORC_MEMORIA
mkdir -p $P3ASORC_MEMORIA
mkdir -p $P3ASORC_CONFIG

cp $P3ASORC_MAIL_CONFIG $P3ASORC_CONFIG
cp $P3ASORC_MAIL_CONFIG2 $P3ASORC_CONFIG
cp $P3ASORC_MAIL_CONFIG3 $P3ASORC_CONFIG
cp $P3ASORC_MAIL_CONFIG4 $P3ASORC_CONFIG
cp $P3ASORC_MAIL_CONFIG5 $P3ASORC_CONFIG
cp $P3ASORC_MAIL_CONFIG6 $P3ASORC_CONFIG
cp $P3ASORC_MAIL_CONFIG7 $P3ASORC_CONFIG
cp $P3ASORC_MAIL_CONFIG8 $P3ASORC_CONFIG
cp $P3ASORC_MAIL_CONFIG9 $P3ASORC_CONFIG

```

```

systemctl status spamd amavis clamav-* postfix dovecot --no-pager -l >
$P3ASORC_LOG

history > $P3ASORC_HISTORIAL
chmod 777 -R $P3ASORC_MEMORIA

tree $P3ASORC_MEMORIA

#-----
# Comprobacion desde host
#-----
# http://192.168.25.10/roundcube P3ASORC_MAIL_OSUSER
#   P3ASORC_MAIL_OSUSER_PASS
# P3ASORC_MAIL_OSUSER -> P3ASORC_MAIL_OSUSER1@P3ASORC_MAIL_HOSTNAME
# http://192.168.25.10/roundcube P3ASORC_MAIL_OSUSER1
#   P3ASORC_MAIL_OSUSER_PASS1
# P3ASORC_MAIL_OSUSER1 -> P3ASORC_MAIL_OSUSER@P3ASORC_MAIL_HOSTNAME
# VER LOS METADATOS DEL CORREO en busca X-Virus, spam...
# Mostrar fichero P3ASORC_MAIL_CONFIG10

```

Fragmento de código 24: Configuración y comprobación de este servicio en  $\Delta$  Debian GNU/Linux 13 (Trixie).

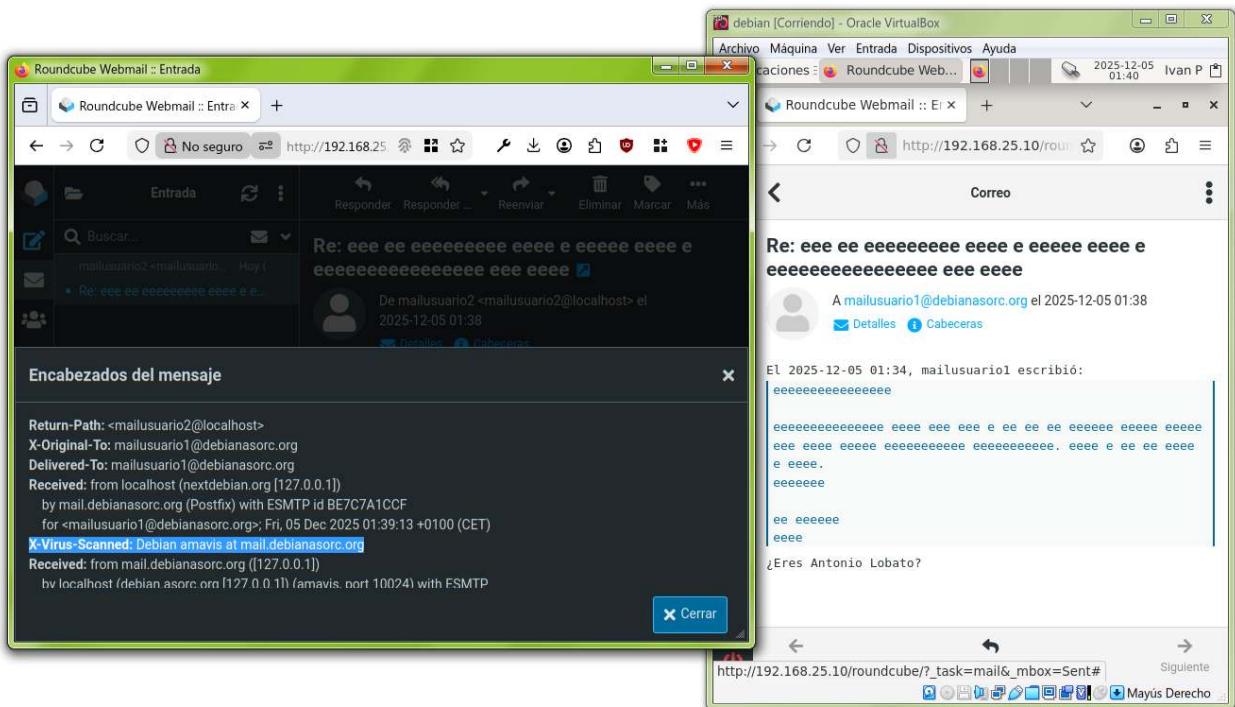


Figura 59: Envío de correo entre clientes desde  $\Delta$  Debian GNU/Linux 13 (Trixie).

## 9.2. ● FreeBSD 14.3

```

#!/bin/sh
su -

```

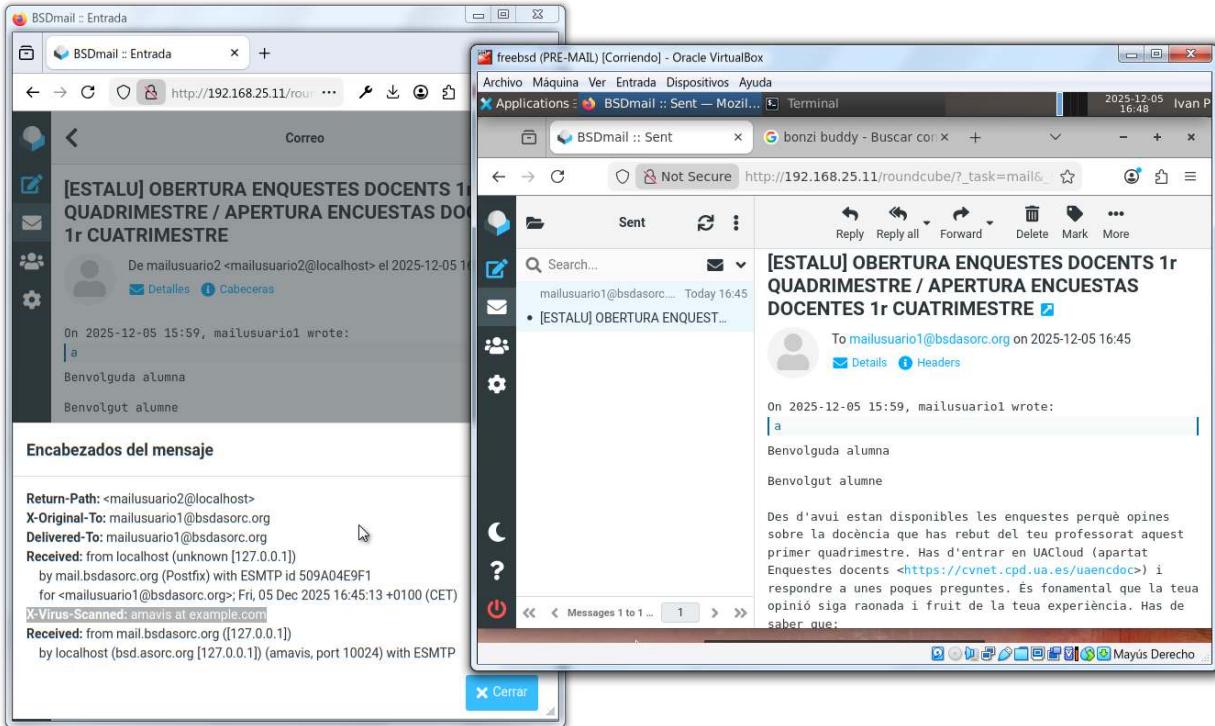


Figura 60: Envío de correo entre clientes desde ● FreeBSD 14.3.

```

#-----#
# Variables de entorno
#-----#
P3ASORC_SERVICIO=mail
P3ASORC_SISTEMA=unix

# NO TOCAR
P3ASORC_MEMORIA=/home/ivan/SUPERMEMORIA/$P3ASORC_SERVICIO
P3ASORC_CONFIG=$P3ASORC_MEMORIA/ficheros_configuracion
P3ASORC_LOG=$P3ASORC_MEMORIA/$P3ASORC_SISTEMA.log
P3ASORC_HISTORIAL=$P3ASORC_MEMORIA/history$P3ASORC_SISTEMA.txt
P3ASORC_BSD_RC=/etc/rc.conf

P3ASORC_MAIL_HOSTNAME=bsdasorc.org
P3ASORC_MAIL_MAIL_HOSTNAME=mail.$P3ASORC_MAIL_HOSTNAME
P3ASORC_MAIL_OSUSER=mailusuari01
P3ASORC_MAIL_OSUSER_PASS=1
P3ASORC_MAIL_OSUSER1=mailusuario2
P3ASORC_MAIL_OSUSER1_PASS=1
P3ASORC_MAIL_CUBO=/usr/local/www/roundcube
P3ASORC_MAIL_MIMARCA="BSDmail"
P3ASORC_MAIL_DB_SQLITE_DST_ROOT=/var/db/roundcube
P3ASORC_MAIL_DB_SQLITE_DST=$P3ASORC_MAIL_DB_SQLITE_DST_ROOT/cuboredondo.db
P3ASORC_MAIL_CUBOLOG_DST_COMMON=/var/log/roundcube-ui
P3ASORC_MAIL_CLE=$P3ASORC_MAIL_CUBOLOG_DST_COMMON-error.log
P3ASORC_MAIL_CLA=$P3ASORC_MAIL_CUBOLOG_DST_COMMON-access.log
P3ASORC_MAIL_SQL_BIN=/usr/local/bin/sqlite3
P3ASORC_MAIL_CONFIG=/etc/mail/mailer.conf
P3ASORC_MAIL_CONFIG1=/usr/local/etc/postfix/main.cf
P3ASORC_MAIL_CONFIG2=/usr/local/etc/dovecot/dovecot.conf

```

```

P3ASORC_MAIL_CONFIG3=/usr/local/etc/dovecot/conf.d/10-mail.conf
P3ASORC_MAIL_CONFIG4=/usr/local/etc/dovecot/conf.d/10-auth.conf
P3ASORC_MAIL_CONFIG5=$P3ASORC_MAIL_CUBO/config/config.inc.php
P3ASORC_MAIL_CONFIG6=/usr/local/etc/apache24/Includes/roundcube.conf
P3ASORC_MAIL_CONFIG7=/usr/local/etc/dovecot/conf.d/10-ssl.conf
P3ASORC_MAIL_CONFIG8=/usr/local/etc/php.ini
P3ASORC_MAIL_AMAVIS_CONF="/usr/local/etc/amavisd.conf"
P3ASORC_MAIL_MASTER_CF="/usr/local/etc/postfix/master.cf"
P3ASORC_MAIL_CLAMD_CONF="/usr/local/etc/clamd.conf"
P3ASORC_MAIL_FRESHCLAM_CONF="/usr/local/etc/freshclam.conf"
P3ASORC_MAIL_HOSTS=/etc/hosts
P3ASORC_MAIL_NALIASBIN=/usr/local/bin/newaliases
P3ASORC_MAIL_NALIASDB=/etc/aliases.db
P3ASORC_MAIL_HOSTINGS_etc="localhost localhost.my.domain nextbsd.org dbbsd.org web1bsd.org web2bsd.org bsd.bsd.asorc.org bsd.asorc.org"
P3ASORC_MAIL_CONFIG_PORT_FILTER1=10024
P3ASORC_MAIL_CONFIG_PORT_FILTER2=10025

#-----
# Servicio (no backtrack)
#-----

# PASO 0: Instalar paqueteria
pkg install -y postfix dovecot roundcube-php83 amavisd-new clamav
spamassassin mod_php83 php83-pdo_sqlite php83-filter php83-mbstring php83-
-iconv php83-session php83-ctype php83-dom php83-xml php83-simplexml
php83-sqlite3 php83-intl php83-zip

# PASO 1: Configurar hosts
cat << EOF > $P3ASORC_MAIL_HOSTS
::1           $P3ASORC_MAIL_HOSTINGS_etc
127.0.0.1     $P3ASORC_MAIL_HOSTINGS_etc
192.168.25.11 $P3ASORC_MAIL_HOSTINGS_etc
EOF

$P3ASORC_MAIL_NALIASBIN
chmod 644 $P3ASORC_MAIL_NALIASDB

# PASO 2: Inicializar postfix
cat << EOF > $P3ASORC_MAIL_CONFIG
#
# mailer.conf for use with dma(8)
#
# If sendmail is configured, an example of mailer.conf that uses sendmail
# instead can be found in /usr/share/examples/sendmail.

sendmail      /usr/local/sbin/postfix
send-mail     /usr/local/sbin/postfix
mailq         /usr/local/sbin/postfix
newaliases    /usr/local/sbin/postfix
EOF

sysrc sendmail_enable="NO"
sysrc sendmail_submit_enable="NO"
sysrc sendmail_outbound_enable="NO"
sysrc sendmail_msp_queue_enable="NO"
sysrc postfix_enable="YES"

# PASO 3: Configurar postfix

```

```

cat << EOF > $P3ASORC_MAIL_CONFIG1
compatibility_level = 3.10
#soft_bounce = no
queue_directory = /var/spool/postfix
command_directory = /usr/local/sbin
daemon_directory = /usr/local/libexec/postfix
data_directory = /var/db/postfix
mail_owner = postfix
#default_privs = nobody
myhostname = $P3ASORC_MAIL_MAIL_HOSTNAME
#myhostname = virtual.domain.tld
#mydomain = domain.tld
mydomain = $P3ASORC_MAIL_HOSTNAME
#myorigin = \$myhostname
myorigin = \$mydomain
inet_interfaces = all
#inet_interfaces = \$myhostname
#inet_interfaces = \$myhostname, localhost
#proxy_interfaces =
#proxy_interfaces = 1.2.3.4
mydestination = \$myhostname, localhost.\$mydomain, localhost, \$mydomain
#mydestination = \$myhostname, localhost.\$mydomain, localhost, \$mydomain
#mydestination = \$myhostname, localhost.\$mydomain, localhost, \$mydomain,
#local_recipient_maps = unix:passwd.byname \$alias_maps
#local_recipient_maps = proxy:unix:passwd.byname \$alias_maps
#local_recipient_maps =
unknown_local_recipient_reject_code = 550
#mynetworks_style = class
#mynetworks_style = subnet
mynetworks_style = host
#mynetworks = 168.100.3.0/28, 127.0.0.0/8
#mynetworks = \$config_directory/mynetworks
#mynetworks = hash:\$config_directory/network_table
#relay_domains =
relay_domains = \$mydestination
#relayhost = \$mydomain
#relayhost = [gateway.my.domain]
#relayhost = [mailserver.isp.tld]
#relayhost = uucphost
#relayhost = [an.ip.add.ress]
#relay_recipient_maps = hash:\$config_directory/relay_recipients
#in_flow_delay = 1s
#alias_maps = dbm:/etc/aliases
#alias_maps = hash:/etc/aliases
#alias_maps = hash:/etc/aliases, nis:mail.aliases
#alias_maps = netinfo:/aliases
#alias_database = dbm:/etc/aliases
#alias_database = hash:/etc/aliases
#alias_database = hash:/etc/aliases, hash:/opt/majordomo/aliases
#recipient_delimiter =
#home_mailbox = Mailbox
home_mailbox = Maildir/
#mail_spool_directory = /var/mail
#mail_spool_directory = /var/spool/mail
#mailbox_command = /some/where/procmail
#mailbox_command = /some/where/procmail -a "\$EXTENSION"
#mailbox_transport = lmtp:unix:/var/imap/socket/lmtp
#mailbox_transport = cyrus

```

```

#fallback_transport = lmtp:unix:/file/name
#fallback_transport = cyrus
#fallback_transport =
#luser_relay = \$user@other.host
#luser_relay = \$local@other.host
#luser_relay = admin+\$local
#header_checks = regexp:\$config_directory/header_checks
#fast_flush_domains = \$relay_domains
smtpd_banner = \$myhostname ESMTP \$mail_name
#smtpd_banner = \$myhostname ESMTP \$mail_name (\$mail_version)
#local_destination_concurrency_limit = 2
#default_destination_concurrency_limit = 20
debug_peer_level = 2
#debug_peer_list = 127.0.0.1
#debug_peer_list = some.domain
debugger_command =
    PATH=/bin:/usr/bin:/usr/local/bin:/usr/X11R6/bin
    ddd \$daemon_directory/\$process_name \$process_id & sleep 5
sendmail_path = /usr/local/sbin/sendmail
newaliases_path = /usr/local/bin/newaliases
mailq_path = /usr/local/bin/mailq
setgid_group = maildrop
html_directory = /usr/local/share/doc/postfix
manpage_directory = /usr/local/share/man
sample_directory = /usr/local/etc/postfix
readme_directory = /usr/local/share/doc/postfix
inet_protocols = ipv4
smtp_tls_CApth = /etc/ssl/certs
meta_directory = /usr/local/libexec/postfix
shlib_directory = /usr/local/lib/postfix
smtpd_use_tls = no
smtpd_tls_security_level = none
smtpd_tls_security_level = none
smtp_dns_support_level = disabled
content_filter = smtp-amavis:[127.0.0.1]:\$P3ASORC_MAIL_CONFIG_PORT_FILTER1
EOF

echo "ssl = no" > $P3ASORC_MAIL_CONFIG7

service postfix enable
service postfix start

postconf -e "disable_dns_lookups = yes"
service postfix restart

# PASO 4: Configurar dovecot
sysrc dovecot_enable="YES"
cat << EOF > $P3ASORC_MAIL_CONFIG2
protocols = imap
listen = *
#base_dir = /var/run/dovecot/
#instance_name = dovecot
#login_greeting = Dovecot ready.
#login_trusted_networks =
#login_access_sockets =
#auth_proxy_self =
#verbose_proctitle = no
#shutdown_clients = yes

```

```

#doveadm_worker_count = 0
#doveadm_socket_path = doveadm-server
#import_environment = TZ
dict {
    #quota = mysql:/usr/local/etc/dovecot/dovecot-dict-sql.conf.ext
}
!include conf.d/*.conf
!include_try local.conf
EOF

# PASO 5: Configurar buzones
cat << EOF > $P3ASORC_MAIL_CONFIG3
mail_location = maildir:~/Maildir
namespace inbox {
    #type = private
    #separator =
    #prefix =
    #location =
    inbox = yes
    #hidden = no
    #list = yes
    #subscriptions = yes
}

namespace {
    #type = shared
    #separator =
    #prefix = shared/%u/
    #location = maildir:%h/Maildir:INDEX=~/Maildir/shared/%u
    #subscriptions = no
    #list = children
}
#mail_shared_explicit_inbox = no
#mail_uid =
#mail_gid =
#mail_privileged_group =
#mail_access_groups =
#mail_full_filesystem_access = no
#mail_attribute_dict =
#mail_server_comment = ""
#mail_server_admin =
#mmap_disable = no
#dotlock_use_excl = yes
#mail_fsync = optimized
#lock_method = fcntl
#mail_temp_dir = /tmp
#first_valid_uid = 500
#last_valid_uid = 0
#first_valid_gid = 1
#last_valid_gid = 0
#mail_max_keyword_length = 50
#valid_chroot_dirs =
#mail_chroot =
#auth_socket_path = /var/run/dovecot/auth-userdb
#mail_plugin_dir = /usr/lib/dovecot
#mail_plugins =
#mailbox_list_index = yes
#mailbox_list_index_dirty_syncs = yes

```

```

#mailbox_list_index_include_inbox = no
#mail_cache_min_mail_count = 0
#mailbox_idle_check_interval = 30 secs
#mail_save_crlf = no
#mail_prefetch_count = 0
#mail_temp_scan_interval = 1w
#mail_sort_max_read_count = 0
protocol !indexer-worker {
    #mail_vsize_bg_after_count = 0
}
#maildir_stat_dirs = no
#maildir_copy_with_hardlinks = yes
#maildir_very_dirty_syncs = no
#maildir_broken_filename_sizes = no
#maildir_empty_new = no
#mbox_read_locks = fcntl
#mbox_write_locks = dotlock fcntl
#mbox_lock_timeout = 5 mins
#mbox_dotlock_change_timeout = 2 mins
#mbox_dirty_syncs = yes
#mbox_very_dirty_syncs = no
#mbox_lazy_writes = yes
#mbox_min_index_size = 0
#mbox_md5 = apop3d
#mdbox_rotate_size = 10M
#mdbox_rotate_interval = 0
#mdbox_preallocate_space = no
#mail_attachment_dir =
#mail_attachment_min_size = 128k
#mail_attachment_fs = sis posix
#mail_attachment_hash = %{sha1}
#mail_attachment_detection_options =
EOF

# PASO 5: Configurar autenticación
cat << EOF > $P3ASORC_MAIL_CONFIG4
disable_plaintext_auth = no
#auth_cache_size = 0
#auth_cache_ttl = 1 hour
#auth_cache_negative_ttl = 1 hour
#auth_realms =
#auth_default_realm =
#auth_username_chars =
    abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ01234567890.-_@
#auth_username_translation =
#auth_username_format = %Lu
#auth_master_user_separator =
#auth_anonymous_username = anonymous
#auth_worker_max_count = 30
#auth_gssapi_hostname =
#auth_krb5_keytab =
#auth_use_winbind = no
#auth_winbind_helper_path = /usr/bin/ntlm_auth
#auth_failure_delay = 2 secs
#auth_ssl_require_client_cert = no
#auth_ssl_username_from_cert = no
auth_mechanisms = plain login

```

```

#!include auth-deny.conf.ext
#!include auth-master.conf.ext

!include auth-system.conf.ext
#!include auth-sql.conf.ext
#!include auth-ldap.conf.ext
#!include auth-passwdfile.conf.ext
#!include auth-checkpassword.conf.ext
#!include auth-static.conf.ext
EOF

service dovecot start
service dovecot restart

# PASO 6: Añadir usuarios
pw userdel -n $P3ASORC_MAIL_OSUSER -y 2>/dev/null
pw userdel -n $P3ASORC_MAIL_OSUSER1 -y 2>/dev/null

echo "$P3ASORC_MAIL_OSUSER_PASS" | pw useradd -n $P3ASORC_MAIL_OSUSER -s /
    sbin/nologin -m -h 0
echo "$P3ASORC_MAIL_OSUSER1_PASS" | pw useradd -n $P3ASORC_MAIL_OSUSER1 -s /
    sbin/nologin -m -h 0

mkdir -p /home/$P3ASORC_MAIL_OSUSER/Maildir/{cur,new,tmp}
mkdir -p /home/$P3ASORC_MAIL_OSUSER1/Maildir/{cur,new,tmp}

chown -R $P3ASORC_MAIL_OSUSER:$P3ASORC_MAIL_OSUSER /home/
    $P3ASORC_MAIL_OSUSER/Maildir
chown -R $P3ASORC_MAIL_OSUSER1:$P3ASORC_MAIL_OSUSER1 /home/
    $P3ASORC_MAIL_OSUSER1/Maildir
chmod -R 700 /home/$P3ASORC_MAIL_OSUSER/Maildir
chmod -R 700 /home/$P3ASORC_MAIL_OSUSER1/Maildir

# PASO 7: Inicializar base de datos para frontend
rm -f $P3ASORC_MAIL_DB_SQLITE_DST
rm -rf $P3ASORC_MAIL_DB_SQLITE_DST_ROOT
mkdir -p $P3ASORC_MAIL_DB_SQLITE_DST_ROOT
chown www:www $P3ASORC_MAIL_DB_SQLITE_DST_ROOT

cp /usr/local/etc/php.ini-production "$P3ASORC_MAIL_CONFIG8"
sed -i .bak 's|^;*\ *session.save_path =.*|session.save_path = "/tmp"|' \
    $P3ASORC_MAIL_CONFIG8
sed -i .bak 's|^extension=|extension=/' "$P3ASORC_MAIL_CONFIG8"

$P3ASORC_MAIL_SQL_BIN $P3ASORC_MAIL_DB_SQLITE_DST < $P3ASORC_MAIL_CUBO/SQL/
    sqlite.initial.sql
chown www:www $P3ASORC_MAIL_DB_SQLITE_DST
chown www:www $P3ASORC_MAIL_DB_SQLITE_DST_ROOT
chmod 775 $P3ASORC_MAIL_DB_SQLITE_DST_ROOT
chmod 664 $P3ASORC_MAIL_DB_SQLITE_DST
chown -R www:www $P3ASORC_MAIL_CUBO/logs $P3ASORC_MAIL_CUBO/temp
chown -R 775 $P3ASORC_MAIL_CUBO/logs
chown -R 775 $P3ASORC_MAIL_CUBO/temp

# PASO 8: Habilitar frontend web
chown -R www:www $P3ASORC_MAIL_CUBO/
cat << EOF > $P3ASORC_MAIL_CONFIG5
<?php

```

```

\$config = [];
\$config['db_dsnw'] = 'sqlite:///P3ASORC_MAIL_DB_SQLITE_DST?mode=0646';
\$config['imap_host'] = 'localhost:143';
\$config['smtp_host'] = 'localhost:25';
\$config['smtp_user'] = '';
\$config['smtp_pass'] = '';
\$config['support_url'] = '';
\$config['product_name'] = '$P3ASORC_MAIL_MIMARCA';
\$config['des_key'] = 'rcmail-!24ByteDESkey*Str';
\$config['plugins'] = [
    'archive',
    'zipdownload',
];
\$config['skin'] = 'elastic';
EOF

cat << EOF > $P3ASORC_MAIL_CONFIG6
Alias /roundcube "$P3ASORC_MAIL_CUBO"

ErrorLog $P3ASORC_MAIL_CLE
CustomLog $P3ASORC_MAIL_CLA combined

<Directory $P3ASORC_MAIL_CUBO>
    Options Indexes FollowSymLinks
    AllowOverride All
    Require all granted
    DirectoryIndex index.php
</Directory>
EOF

service apache24 restart

# PASO 9: Antivirus + SPAM
sysrc clamav_clamd_enable="YES"
sysrc clamav_freshclam_enable="YES"
sysrc amavisd_enable="YES"
sysrc spamd_enable="YES"

cp /usr/local/etc/freshclam.conf.sample $P3ASORC_MAIL_FRESHCLAM_CONF
cp /usr/local/etc/clamd.conf.sample $P3ASORC_MAIL_CLAMD_CONF
sed -i .bak 's/^Example/#Example/' $P3ASORC_MAIL_FRESHCLAM_CONF
sed -i .bak 's/^Example/#Example/' $P3ASORC_MAIL_CLAMD_CONF
sed -i .bak 's|^#LocalSocket .*|LocalSocket /var/run/clamav/clamd.sock|' $P3ASORC_MAIL_CLAMD_CONF

mkdir -p /var/run/clamav
chown clamav:clamav /var/run/clamav

freshclam
sa-update

sed -i .bak 's/^#@bypass_virus_checks_maps/@bypass_virus_checks_maps/' $P3ASORC_MAIL_AMAVIS_CONF
sed -i .bak 's/^#@bypass_spam_checks_maps/@bypass_spam_checks_maps/' $P3ASORC_MAIL_AMAVIS_CONF
pw groupmod vscan -m clamav

cat << EOF >> $P3ASORC_MAIL_MASTER_CF

```

```

smtp-amavis unix - - n - 2 smtp
  -o smtp_data_done_timeout=1200
  -o smtp_send_xforward_command=yes
  -o smtp_dns_support_level=disabled

127.0.0.1:$P3ASORC_MAIL_CONFIG_PORT_FILTER2 inet n - n -
  - smtpd
  -o content_filter=
  -o local_recipient_maps=
  -o relay_recipient_maps=
  -o smtpd_restriction_classes=
  -o smtpd_client_restrictions=
  -o smtpd_helo_restrictions=
  -o smtpd_sender_restrictions=
  -o smtpd_recipient_restrictions=permit_mynetworks,reject
  -o mynetworks=127.0.0.0/8
  -o strict_rfc821_envelopes=yes
  -o smtpd_error_sleep_time=0
  -o smtpd_soft_error_limit=1001
  -o smtpd_hard_error_limit=1000
  -o receive_override_options=no_milters
EOF

# PASO 10: Reinicio de la pila de correo
service sa-spamd start
service clamav_freshclam start
service clamav_clamd restart
service amavisd restart
service postfix restart

#-----
# Extraer logs, configs e historial
#-----

rm -rf $P3ASORC_MEMORIA
mkdir -p $P3ASORC_MEMORIA
mkdir -p $P3ASORC_CONFIG

cp $P3ASORC_MAIL_CONFIG $P3ASORC_CONFIG
cp $P3ASORC_MAIL_CONFIG2 $P3ASORC_CONFIG
cp $P3ASORC_MAIL_CONFIG3 $P3ASORC_CONFIG
cp $P3ASORC_MAIL_CONFIG4 $P3ASORC_CONFIG
cp $P3ASORC_MAIL_CONFIG5 $P3ASORC_CONFIG
cp $P3ASORC_MAIL_CONFIG6 $P3ASORC_CONFIG
cp $P3ASORC_MAIL_CONFIG7 $P3ASORC_CONFIG
cp $P3ASORC_MAIL_CONFIG8 $P3ASORC_CONFIG
cp $P3ASORC_MAIL_CONFIG9 $P3ASORC_CONFIG
cp $P3ASORC_MAIL_AMAVIS_CONF $P3ASORC_CONFIG
cp $P3ASORC_MAIL_MASTER_CF $P3ASORC_CONFIG
cp $P3ASORC_MAIL_CLAMD_CONF $P3ASORC_CONFIG
cp $P3ASORC_MAIL_FRESHCLAM_CONF $P3ASORC_CONFIG

{
  echo "====="
  echo " ESTADO DE SERVICIOS"
  echo "====="
  service postfix status
  service dovecot status

```

```

service apache24 status
service amavisd status
service clamav_clamd status
service clamav_freshclam status
service sa-spamd status

echo ""
echo "=====PUERTOS ESCUCHANDO (sockstat)===="
echo "# Verificamos puertos críticos: 25(smtp), 143(imap), 10024(amavis),
10025(postfix-reentry), 3310(clamav), 783(spamassassin)
sockstat -4 -l | grep -E '25|143|10024|10025|3310|783'

echo ""
echo "===== /var/log/maillog ====="
echo "/var/log/maillog"
echo "===== "
tail -n 999 /var/log/maillog

echo ""
echo "===== ERRORES DE ROUND CUBE ====="
echo "===== "
tail -n 999 "$P3ASORC_MAIL_CLE"
tail -n 999 "$P3ASORC_MAIL_CLA"
} > $P3ASORC_LOG

history > $P3ASORC_HISTORIAL
chmod 777 -R $P3ASORC_MEMORIA

tree $P3ASORC_MEMORIA

#-----
# Comprobacion desde host
#-----
# http://192.168.25.11/roundcube P3ASORC_MAIL_OSUSER
#   P3ASORC_MAIL_OSUSER_PASS
# P3ASORC_MAIL_OSUSER -> P3ASORC_MAIL_OSUSER1@P3ASORC_MAIL_HOSTNAME
# http://192.168.25.11/roundcube P3ASORC_MAIL_OSUSER1
#   P3ASORC_MAIL_OSUSER_PASS1
# P3ASORC_MAIL_OSUSER1 -> P3ASORC_MAIL_OSUSER@P3ASORC_MAIL_HOSTNAME
# VER LOS METADATOS DEL CORREO en busca X-Virus, spam...
# Mostrar fichero P3ASORC_MAIL_AMAVIS_CONF

```

Fragmento de código 25: Configuración y comprobación de este servicio en ● FreeBSD 14.3.

### 9.3. Microsoft Windows Server 2025 SIN HACER

## 10. Servidor de terminales de usuario (PXE)

«*Servidor de terminales de usuario. Se pide instalar un servidor tipo LTSP o DRBL de manera que al arrancar un sistema diskless mediante PXE cargue el sistema operativo completo y de manera gráfica a través de la red.*»

---

El entorno **PXE** (*Preboot eXecution Environment*) es un estándar de la industria que permite a una estación de trabajo arrancar e instalar un sistema operativo a través de la red, independientemente de los dispositivos de almacenamiento de datos disponibles (como discos duros) o sistemas operativos instalados.

El servicio se basa en la orquestación de varios protocolos de red estándar:

- **DHCP**: Asigna una dirección IP al cliente y le indica dónde encontrar el archivo de arranque.
- **TFTP (*Trivial File Transfer Protocol*)**: Un protocolo ligero utilizado para transferir el *bootloader* inicial (como PXELINUX o WDS NBP) a la memoria RAM del cliente.
- **Servicios de archivos (HTTP/NFS/SMB)**: Utilizados posteriormente para transferir la imagen pesada del sistema operativo durante la instalación.

En cuanto a sus orígenes, PXE fue introducido por **Intel** en **1998** como parte de la iniciativa *Wired for Management* (WfM). Fue diseñado para reemplazar a las antiguas ROMs de arranque propietarias (como RPL), estableciendo un estándar universal que los fabricantes de tarjetas de red (NIC) y BIOS pudieran implementar.

Sobre uso, es una tecnología indispensable en la administración de grandes parques informáticos y centros de datos. Permite el despliegue masivo y desatendido de servidores y estaciones de trabajo, así como la ejecución de herramientas de diagnóstico o sistemas *live* sin necesidad de soportes físicos (USB/CD).

En el desarrollo de esta práctica, la implementación del servidor PXE tiene un doble objetivo técnico:

1. **Infraestructura de arranque de red**: Configurar correctamente los servicios DHCP y TFTP para entregar el código de arranque a un cliente "desnudo" (*bare-metal*).
2. **Despliegue de imagen**: Facilitar el acceso a los archivos de instalación del sistema operativo, permitiendo que el instalador tome el control tras el arranque inicial por red.

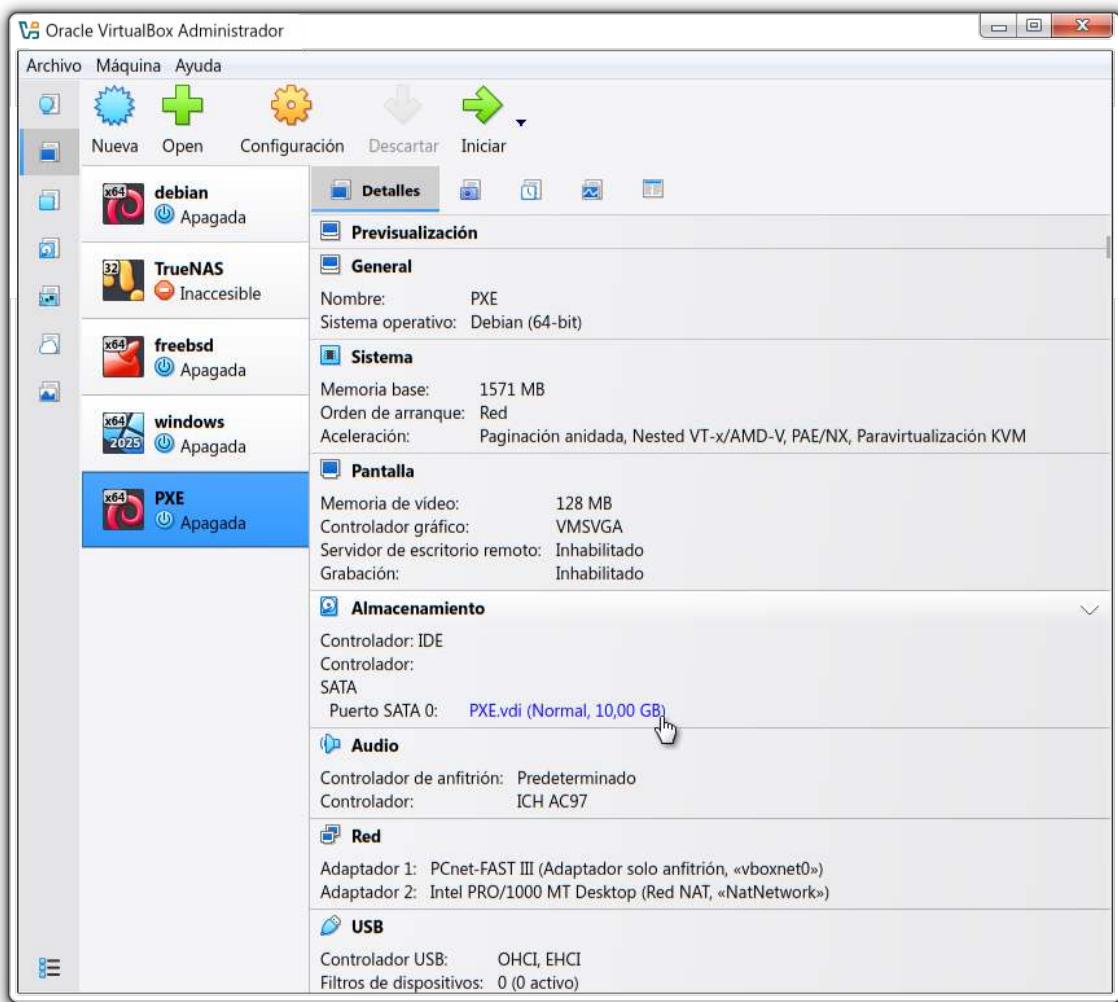


Figura 61: Configuración de la VM cliente PXE para los puntos 10 y 11 en VBox.

### 10.1. Debian GNU/Linux 13 (Trixie)

```
#!/bin/sh
su -

#-----
# Variables de entorno
#-----
P3ASORC_SERVICIO=pxe
P3ASORC_SISTEMA=linux

# NO TOCAR
P3ASORC_MEMORIA=/home/ivan/SUPERMEMORIA/$P3ASORC_SERVICIO
P3ASORC_CONFIG=$P3ASORC_MEMORIA/ficheros_configuracion
P3ASORC_LOG=$P3ASORC_MEMORIA/$P3ASORC_SISTEMA.log
P3ASORC_HISTORIAL=$P3ASORC_MEMORIA/history$P3ASORC_SISTEMA.txt

# --- VARIABLES ESPECIFICAS PXE ---
P3ASORC_PXE_IFACE=enp0s8
P3ASORC_PXE_IP=192.168.25.10
P3ASORC_PXE_MASK=255.255.255.0
P3ASORC_PXE_RANGO_INI=192.168.25.50
```

```

P3ASORC_PXE_RANGO_FIN=192.168.25.100
P3ASORC_PXE_DNS=1.1.1.1
P3ASORC_PXE_TFTP_DIR=/var/lib/tftpboot
# Usamos Slitaz Rolling: ~50MB con entorno grafico LXDE/Openbox incluido en
# el initrd
P3ASORC_PXE_URL_ISO=http://mirror.slitaz.org/iso/rolling/slitz-rolling.iso

# Rutas de configuracion para backup
P3ASORC_GRUPO_CONFIG_DHCP=/etc/dhcp/dhcpd.conf
P3ASORC_GRUPO_CONFIG_TFTP=/etc/default/tftpd-hpa
P3ASORC_GRUPO_CONFIG_PXE=$P3ASORC_PXE_TFTP_DIR/pixelinux.cfg/default

#-----
# Servicio (no backtrack)
#-----

# PASO 1: Instalar paquetes y limpieza profunda de puertos
apt update
apt install -y isc-dhcp-server tftpd-hpa pixelinux syslinux-common wget
psmisc

# Detener servicios para asegurar configuracion limpia
systemctl stop isc-dhcp-server
systemctl stop tftpd-hpa
systemctl stop inetd 2>/dev/null || true
systemctl stop xinetd 2>/dev/null || true

# Matar procesos huertos en puertos criticos (67 UDP, 69 UDP, 53 TCP/UDP)
echo "Liberando puertos..."
fuser -k -v 67/udp || echo "Puerto 67 OK"
fuser -k -v 69/udp || echo "Puerto 69 OK"
fuser -k -v 53/tcp || echo "Puerto 53 OK"

# Crear directorio antes de asignar permisos
mkdir -p $P3ASORC_PXE_TFTP_DIR
chown -R tftp:tftp $P3ASORC_PXE_TFTP_DIR
chmod -R 777 $P3ASORC_PXE_TFTP_DIR

# Iniciamos los servicios limpios
systemctl start isc-dhcp-server
systemctl start tftpd-hpa

# PASO 2: Configurar interfaz de escucha DHCP
# Vinculamos DHCP solo a la interfaz Host-Only
sed -i 's/^INTERFACESv4=.*$/INTERFACESv4="$P3ASORC_PXE_IFACE"/' /etc/
default/isc-dhcp-server

# PASO 3: Configurar DHCPD
# Definimos la subred y apuntamos al fichero de arranque
cat > $P3ASORC_GRUPO_CONFIG_DHCP <<EOF
option domain-name "pxelab.local";
option domain-name-servers $P3ASORC_PXE_DNS;
default-lease-time 600;
max-lease-time 7200;
authoritative;

subnet 192.168.25.0 netmask $P3ASORC_PXE_MASK {
    range $P3ASORC_PXE_RANGO_INI $P3ASORC_PXE_RANGO_FIN;
}
EOF

```

```

    option routers $P3ASORC_PXE_IP;
    next-server $P3ASORC_PXE_IP;
    filename "pxelinux.0";
}
EOF

# PASO 4: Configurar TFTP y Syslinux
sed -i 's|TFTP_DIRECTORY=.*|TFTP_DIRECTORY="$P3ASORC_PXE_TFTP_DIR"|'
$P3ASORC_GRUPO_CONFIG_TFTP
mkdir -p $P3ASORC_PXE_TFTP_DIR/pxelinux.cfg
cp /usr/lib/PXELINUX/pxelinux.0 $P3ASORC_PXE_TFTP_DIR/
cp /usr/lib/syslinux/modules/bios/ldlinux.c32 $P3ASORC_PXE_TFTP_DIR/
cp /usr/lib/syslinux/modules/bios/menu.c32 $P3ASORC_PXE_TFTP_DIR/
cp /usr/lib/syslinux/modules/bios/libutil.c32 $P3ASORC_PXE_TFTP_DIR/

# PASO 5: Descargar y preparar Thin Client (Slitaz)
if [ ! -f /tmp/slitz.iso ]; then
    wget -O /tmp/slitz.iso $P3ASORC_PXE_URL_ISO
fi

mkdir -p /tmp/iso_mount
mkdir -p $P3ASORC_PXE_TFTP_DIR/slitz
mount -o loop /tmp/slitz.iso /tmp/iso_mount

# Slitaz Rolling tiene multiples rootfs (rootfs1.gz, rootfs2.gz, etc)
# Los concatenamos todos en un solo archivo initrd.gz para cargar todo el
# entorno grafico
echo "Combinando sistemas de archivos de Slitaz..."
cat /tmp/iso_mount/boot/rootfs*.gz > $P3ASORC_PXE_TFTP_DIR/slitz/initrd.gz
cp /tmp/iso_mount/boot/bzImage $P3ASORC_PXE_TFTP_DIR/slitz/vmlinuz
umount /tmp/iso_mount

# PASO 6: Crear menu de arranque PXE
# Configuramos el arranque con los parametros especificos para Slitaz
cat > $P3ASORC_GRUPO_CONFIG_PXE <<EOF
DEFAULT menu.c32
PROMPT 0
TIMEOUT 50
MENU TITLE PXE Boot Menu - Debian 13 (Grafico)

LABEL slitz
    MENU LABEL ^1) Thin Client Grafico (Slitaz Rolling)
    KERNEL slitz/vmlinuz
    APPEND initrd=slitz/initrd.gz rw root=/dev/null autologin
EOF

# PASO 7: Reiniciar servicios finales
systemctl restart isc-dhcp-server
systemctl restart tftpd-hpa

#-----
# Valida servicio
#-----
systemctl status isc-dhcp-server --no-pager
systemctl status tftpd-hpa --no-pager
netstat -tunlp | grep -E '67|69'
ls -R $P3ASORC_PXE_TFTP_DIR/slitz

```

```

#-----#
# Extraer logs, configs e historial
#-----#
rm -rf $P3ASORC_MEMORIA
mkdir -p $P3ASORC_MEMORIA
mkdir -p $P3ASORC_CONFIG

# Copia de configuraciones
cp $P3ASORC_GRUPO_CONFIG_DHCP $P3ASORC_CONFIG/dhcpd.conf
cp $P3ASORC_GRUPO_CONFIG_TFTP $P3ASORC_CONFIG/tftpd-hpa
cp $P3ASORC_GRUPO_CONFIG_PXE $P3ASORC_CONFIG/default.pxe

# Extraccion de logs de estado
systemctl status --no-pager -l isc-dhcp-server > $P3ASORC_LOG
systemctl status --no-pager -l tftpd-hpa >> $P3ASORC_LOG

# Historial
history > $P3ASORC_HISTORIAL
chmod 777 -R $P3ASORC_MEMORIA

tree $P3ASORC_MEMORIA

#-----#
# Comprobacion desde host
#-----#
# Iniciar VM Cliente -> Menu PXE -> Seleccionar SliTaz.
# Pulsa <enter> 2 veces
# Deberias ver un escritorio grafico completo cargado en RAM.
# Ejecuta lxde-session si fuera necesario.
# Usuario: root password: root

```

Fragmento de código 26: Configuración y comprobación de este servicio en  $\Delta$  Debian GNU/Linux 13 (Trixie).

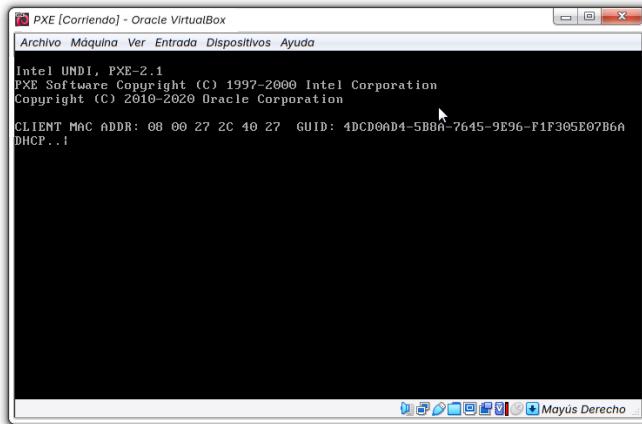


Figura 62: Arranque válido por red gracias a  $\Delta$  Debian GNU/Linux 13 (Trixie).

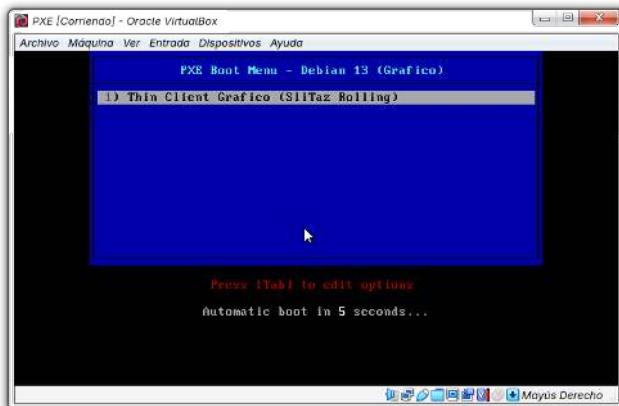


Figura 63: Menú PXE con la única entrada de iniciar en  $\Delta$  Debian GNU/Linux 13 (Trixie).

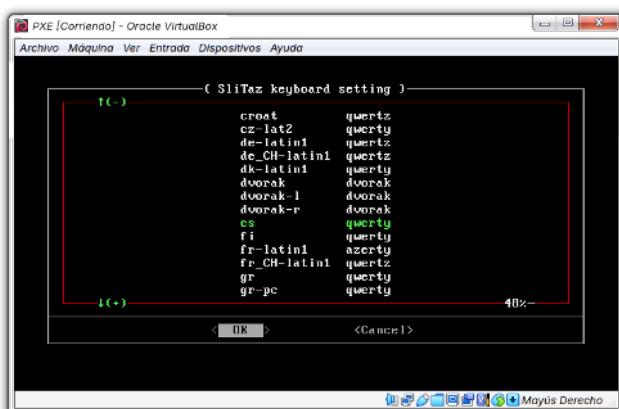


Figura 64: Menú de opciones de idioma y teclado.

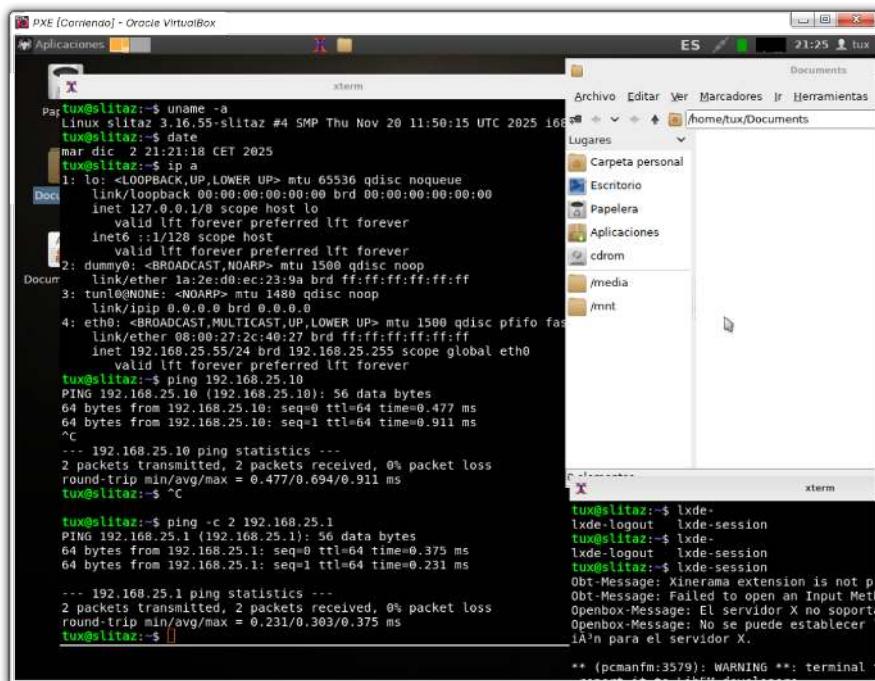


Figura 65: Sistema operativo completo con entorno gráfico `lxde-session` arrancado a través de red.

## 10.2. ● FreeBSD 14.3

```
#!/bin/sh
su -

#-----
# Variables de entorno
#-----
P3ASORC_SERVICIO=pxe
P3ASORC_SISTEMA=unix

# NO TOCAR
P3ASORC_MEMORIA=/root/SUPERMEMORIA/$P3ASORC_SERVICIO
P3ASORC_CONFIG=$P3ASORC_MEMORIA/ficheros_configuracion
P3ASORC_LOG=$P3ASORC_MEMORIA/$P3ASORC_SISTEMA.log
P3ASORC_HISTORIAL=$P3ASORC_MEMORIA/history$P3ASORC_SISTEMA.txt

# --- VARIABLES ESPECIFICAS PXE ---
# En FreeBSD (VirtualBox Intel Pro/1000), el adaptador 2 suele ser em1
P3ASORC_PXE_IFACE=em1
P3ASORC_PXE_IP=192.168.25.11
P3ASORC_PXE_MASK=255.255.255.0
P3ASORC_PXE_RANGO_INI=192.168.25.101
P3ASORC_PXE_RANGO_FIN=192.168.25.150
P3ASORC_PXE_DNS=1.1.1.1
P3ASORC_PXE_TFTP_DIR=/var/tftpboot
# Usamos Slitaz para demo rapida (Entorno Grafico ligero).
P3ASORC_PXE_URL_ISO=http://mirror.slitaz.org/iso/rolling/slitz-rolling.iso

# Rutas de configuracion FreeBSD
P3ASORC_GRUPO_CONFIG_DHCP=/usr/local/etc/dhcpd.conf
P3ASORC_GRUPO_CONFIG_PXE=$P3ASORC_PXE_TFTP_DIR/pxelinux.cfg/default
P3ASORC_INETD_CONF=/etc/inetd.conf

#-----
# Servicio
#-----

# PASO 1: Instalar paquetes y limpieza profunda
# Usamos pkg para instalar servidor DHCP y syslinux.
env ASSUME_ALWAYS_YES=yes pkg update -f
env ASSUME_ALWAYS_YES=yes pkg install isc-dhcp44-server syslinux wget

# Detener servicios y DESACTIVAR FIREWALLS (Causa principal de Timeouts)
service isc-dhcpd stop 2>/dev/null || true
service inetd stop 2>/dev/null || true
service pf stop 2>/dev/null || true
service ipfw stop 2>/dev/null || true
killall tftpd 2>/dev/null || true

# Limpieza de puertos
echo "Verificando puertos..."
sockstat -4 -l | grep -E ':67|:69' || echo "Puertos UDP 67/69 Libres"

# Crear directorio TFTP y asignar permisos
mkdir -p $P3ASORC_PXE_TFTP_DIR
chmod -R 777 $P3ASORC_PXE_TFTP_DIR
```

```

# PASO 2: Configurar servicios en rc.conf
# Habilitamos DHCPD e INETD
sysrc dhcpcd_enable="YES"
sysrc dhcpcd_ifaces="$P3ASORC_PXE_IFACE"
# Desactivamos firewalls permanentemente para la demo
sysrc pf_enable="NO"
sysrc firewall_enable="NO"
# Desactivamos tftpd standalone para evitar conflictos
sysrc tftpd_enable="NO"
# Activamos inetd con flags de escucha global
sysrc inetd_enable="YES"
sysrc inetd_flags="-wW -C 60"

# PASO 3: Configurar INETD para TFTP (Fix timeout)
# 1. Copia de seguridad
cp $P3ASORC_INETD_CONF ${P3ASORC_INETD_CONF}.bak
# 2. Borramos cualquier linea previa de tftp para evitar duplicados o
# errores
sed -i '' '/^tftp/d' $P3ASORC_INETD_CONF
sed -i '' '/^#tftp/d' $P3ASORC_INETD_CONF
# 3. Insertamos la linea limpia forzando udp4 y ruta correcta
# "udp4" asegura IPv4, "-l" logging, "-s" chroot seguro
echo "tftp dgram udp4 wait root /usr/libexec/tftpd tftpd -l -s
$P3ASORC_PXE_TFTP_DIR" >> $P3ASORC_INETD_CONF

echo "Configuracion inetd regenerada correctamente."

# PASO 4: Configurar DHCPD
cat > $P3ASORC_GRUPO_CONFIG_DHCP <<EOF
option domain-name "pxelab.local";
option domain-name-servers $P3ASORC_PXE_DNS;
default-lease-time 600;
max-lease-time 7200;
authoritative;

subnet 192.168.25.0 netmask $P3ASORC_PXE_MASK {
    range $P3ASORC_PXE_RANGO_INI $P3ASORC_PXE_RANGO_FIN;
    option routers 192.168.25.1;
    next-server $P3ASORC_PXE_IP;
    filename "pxelinux.0";
}
EOF

# PASO 5: Configurar ficheros Syslinux
SYSLINUX_PATH=/usr/local/share/syslinux/bios
mkdir -p $P3ASORC_PXE_TFTP_DIR/pxelinux.cfg

cp $SYSLINUX_PATH/core/pxelinux.0 $P3ASORC_PXE_TFTP_DIR/
cp $SYSLINUX_PATH/com32/elflink/ldlinux.c32 $P3ASORC_PXE_TFTP_DIR/
cp $SYSLINUX_PATH/com32/menu/menu.c32 $P3ASORC_PXE_TFTP_DIR/
cp $SYSLINUX_PATH/com32/libutil/libutil.c32 $P3ASORC_PXE_TFTP_DIR/

# PASO 6: Descargar y preparar Thin Client (ISO)
if [ ! -f /tmp/slitz.iso ]; then
    echo "Descargando ISO..."
    wget -O /tmp/slitz.iso $P3ASORC_PXE_URL_ISO
fi

```

```

mkdir -p /tmp/iso_mount
mkdir -p $P3ASORC_PXE_TFTP_DIR/slitz

echo "Montando ISO via mdconfig..."
MD_UNIT=$(mdconfig -a -t vnode -f /tmp/slitz.iso)
mount -t cd9660 /dev/$MD_UNIT /tmp/iso_mount

echo "Extrayendo kernel e initrd..."
cat /tmp/iso_mount/boot/rootfs*.gz > $P3ASORC_PXE_TFTP_DIR/slitz/initrd.gz
cp /tmp/iso_mount/boot/bzImage $P3ASORC_PXE_TFTP_DIR/slitz/vmlinuz

umount /tmp/iso_mount
mdconfig -d -u ${MD_UNIT#md}

# PASO 7: Crear menu de arranque PXE
cat > $P3ASORC_GRUPO_CONFIG_PXE <<EOF
DEFAULT menu.c32
PROMPT 0
TIMEOUT 50
MENU TITLE PXE Boot Menu - FreeBSD 14.3 (Grafico)

LABEL slitz
  MENU LABEL ^1) Thin Client Grafico (Slitaz/Xfce Compliant)
  KERNEL slitz/vmlinuz
  APPEND initrd=slitz/initrd.gz rw root=/dev/null autologin
EOF

# PASO 8: Reiniciar servicios finales
# Reiniciamos inetc (TFTP) y dhcpcd
service inetc restart
service isc-dhcpcd restart

#-----
# Validación
#-----
echo "Estado de servicios:"
service isc-dhcpcd status
service inetc status
echo "Escucha de puertos (Debe aparecer *:69):"
sockstat -4 -l | grep -E ':67|:69'
ls -R $P3ASORC_PXE_TFTP_DIR/slitz

#-----
# Extracción logs
#-----
rm -rf $P3ASORC_MEMORIA
mkdir -p $P3ASORC_MEMORIA
mkdir -p $P3ASORC_CONFIG

# Copia de configuraciones
cp $P3ASORC_GRUPO_CONFIG_DHCP $P3ASORC_CONFIG/dhcpcd.conf
grep tftp $P3ASORC_INETD_CONF > $P3ASORC_CONFIG/inetd_tftp_line
cp $P3ASORC_GRUPO_CONFIG_PXE $P3ASORC_CONFIG/default.pxe

# Logs (FreeBSD usa /var/log/messages)
tail -n 50 /var/log/messages | grep -E 'dhcpcd|inetd|tftpd' > $P3ASORC_LOG

# Historial

```

```

history > $P3ASORC_HISTORIAL
chmod 777 -R $P3ASORC_MEMORIA

echo "Despliegue completado en $P3ASORC_MEMORIA"
tree $P3ASORC_MEMORIA || ls -R $P3ASORC_MEMORIA

#-----
# Comprobación
#-----
# Iniciar VM Cliente -> Menu PXE -> Seleccionar SliTaz.
# Pulta <enter> 2 veces
# Deberias ver un escritorio grafico completo cargado en RAM.
# Ejecuta lxde-session si fuera necesario.
# Usuario: root password: root

```

Fragmento de código 27: Configuración y comprobación de este servicio en ● FreeBSD 14.3.

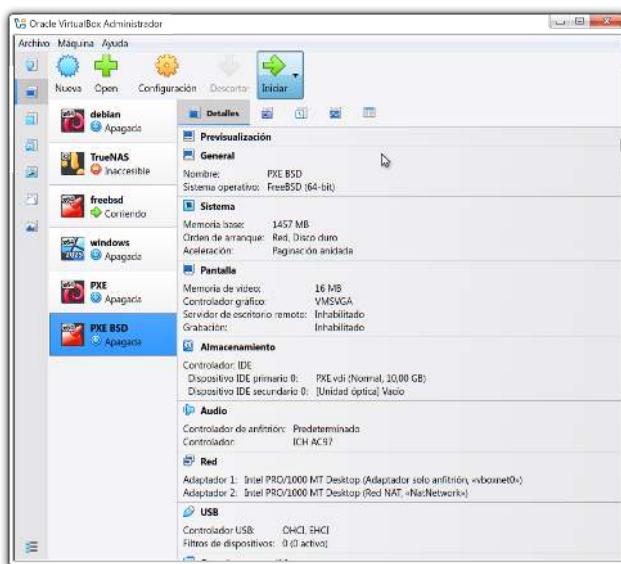


Figura 66: Configuración de la VM cliente empleada para arrancar por red de ● FreeBSD 14.3.

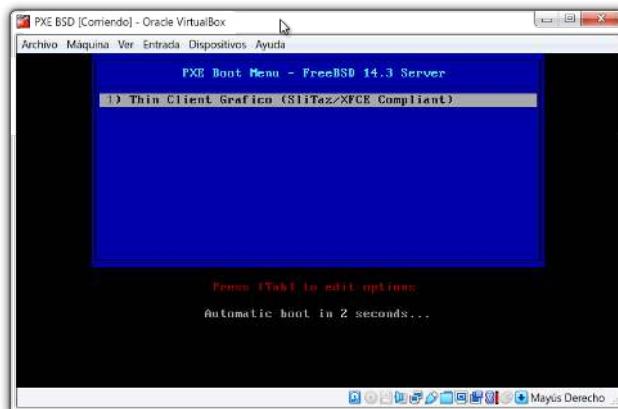


Figura 67: Menú PXE con la única entrada de iniciar un cliente ligero alojado en ● FreeBSD 14.3.

### 10.3. Microsoft Windows Server 2025 **SIN HACER**

## 11. Administración de instalaciones remotas + WoL

«Administración de instalaciones remotas. Se pide instalar un servidor de instalaciones remotas desatendidas. Utilizando el mismo método que en el punto 10, el cliente deberá de instalar uno de los tres sistemas operativos de manera desatendida mediante el paso del archivo de configuración como parámetro al instalador. Durante la demo se pedirá adicionalmente que el cliente esté apagado y se arranque mediante un paquete wol (wake on lan).»

---

**Wake-on-LAN** (WoL) es un estándar de redes Ethernet que permite encender remotamente un ordenador apagado mediante el envío de un paquete de red especialmente diseñado.

El servicio opera en la capa de enlace de datos (Capa 2 del modelo OSI):

- **Paquete mágico (*Magic packet*):** Es una trama de difusión (*broadcast*) que contiene 6 bytes con valor 255 (FF FF FF FF FF FF), seguidos de 16 repeticiones de la dirección MAC del equipo destino.
- **Recepción:** La tarjeta de red (NIC) del equipo apagado se mantiene en un estado de bajo consumo (*standby*), escuchando el tráfico de red en busca de esta secuencia específica para enviar la señal de encendido a la placa base.

En cuanto a sus orígenes, la tecnología fue desarrollada fruto de una alianza entre **Intel** e **IBM** en **1997**, presentándola como parte de las especificaciones de gestión avanzada de energía. Inicialmente requería un cable físico de 3 pines entre la tarjeta de red y la placa base, pero con la llegada del estándar PCI 2.2 y posteriores, la energía se suministra directamente a través del bus PCI/PCIe.

Sobre uso, WoL es esencial para la administración remota eficiente y el ahorro energético (*Green IT*). Permite a los administradores mantener los equipos apagados fuera del horario laboral y encenderlos bajo demanda para realizar tareas de mantenimiento nocturno, actualizaciones de seguridad o copias de seguridad, sin necesidad de desplazamiento físico.

En el desarrollo de esta práctica, la implementación de WoL persigue los siguientes objetivos técnicos:

1. **Configuración de HW y BIOS:** Habilitar las opciones de energía necesarias en la BIOS/UEFI y en el sistema operativo para permitir que la tarjeta de red despierte al equipo.
2. **Emisión del paquete mágico:** Utilizar herramientas de línea de comandos o gráficas desde un equipo controlador para generar y enviar la trama de encendido, verificando la reactivación exitosa del equipo objetivo.

**Nota:** Debido a las limitaciones de VBox, es muy probable que hacer `wakeonlan` (o derivados como `etherwake` o `ether-wake`) falle. Mis investigaciones concluyen con que es muy poco probable que exista forma nativa alguna en VBox de hacerlo *user friendly*. No obstante, existe un programa obsoleto y propietario para este uso. Sin embargo y debido al poco mantenimiento del proyecto junto a cuestiones de licencia evitaremos esta opción a toda costa. Es posible que exista algún *extension pack* que efectivamente trate esta clase de paquetería. Como ingeniero ante un problema que puede ser solucionado o bien con un reinicio o descargando un paquete creado por otras personas toca crear la una solución. La propuesta ingeniosa consiste en tratar las señales provenientes de los sockets tanto de la capa Ethernet como de UDP... Los detalles de implementación se encuentran en los anexos de este documento.

## 11.1. 🐧 Debian GNU/Linux 13 (Trixie)

```
#!/bin/sh
su -

#-----
# Variables de entorno
#-----
P3ASORC_SERVICIO=wol
P3ASORC_SISTEMA=linux

# NO TOCAR
P3ASORC_MEMORIA=/home/ivan/SUPERMEMORIA/$P3ASORC_SERVICIO
P3ASORC_CONFIG=$P3ASORC_MEMORIA/ficheros_configuracion
P3ASORC_LOG=$P3ASORC_MEMORIA/$P3ASORC_SISTEMA.log
P3ASORC_HISTORIAL=$P3ASORC_MEMORIA/history$P3ASORC_SISTEMA.txt

# --- VARIABLES ESPECIFICAS WOL / DESATENDIDA ---
P3ASORC_WOL_IFACE=enp0s8
P3ASORC_WOL_IP=192.168.25.10
P3ASORC_WOL_MASK=255.255.255.0
P3ASORC_WOL_RANGO_INI=192.168.25.50
P3ASORC_WOL_RANGO_FIN=192.168.25.100
P3ASORC_WOL_DNS=1.1.1.1
P3ASORC_WOL_TFTP_DIR=/var/lib/tftpboot
P3ASORC_WOL_WEB_DIR=/var/www/html
# URL del instalador de red de Debian 13
P3ASORC_WOL_URL_DEBIAN=http://ftp.debian.org/debian/dists/bookworm/main/
installer-amd64/current/images/netboot/debian-installer/amd64

# Rutas de configuracion
P3ASORC_GRUPO_CONFIG_DHCP=/etc/dhcp/dhcpd.conf
P3ASORC_GRUPO_CONFIG_TFTP=/etc/default/tftpd-hpa
P3ASORC_GRUPO_CONFIG_PXE=$P3ASORC_WOL_TFTP_DIR/pxelinux.cfg/default

#-----
# Servicio (no backtrack)
#-----

# PASO 1: Instalar paquetes (Añadido etherwake y apache2)
# apache2 servira el fichero preseed.cfg, etherwake hace el WOL
apt update
apt install -y isc-dhcp-server tftpd-hpa pxelinux syslinux-common wget
psmisc etherwake apache2

# Limpieza de procesos previos
systemctl stop isc-dhcp-server
systemctl stop tftpd-hpa
systemctl stop apache2

# Liberar puertos por seguridad
fuser -k -v 67/udp || echo "Puerto 67 limpio"
fuser -k -v 69/udp || echo "Puerto 69 limpio"
fuser -k -v 80/tcp || echo "Puerto 80 limpio"

# Asegurar directorios y permisos
mkdir -p $P3ASORC_WOL_TFTP_DIR
chown -R tftp:tftp $P3ASORC_WOL_TFTP_DIR
```

```

chmod -R 777 $P3ASORC_WOL_TFTP_DIR

# Iniciar servicios base
systemctl start isc-dhcp-server
systemctl start tftpd-hpa
systemctl start apache2

# PASO 2: Configurar interfaz DHCP
sed -i 's/^INTERFACESv4=.*$/INTERFACESv4="$P3ASORC_WOL_IFACE"/' /etc/
    default/isc-dhcp-server

# PASO 3: Configurar DHCPD
cat > $P3ASORC_GRUPO_CONFIG_DHCP <<EOF
option domain-name "pxewol.local";
option domain-name-servers $P3ASORC_WOL_DNS;
default-lease-time 600;
max-lease-time 7200;
authoritative;

subnet 192.168.25.0 netmask $P3ASORC_WOL_MASK {
    range $P3ASORC_WOL_RANGO_INI $P3ASORC_WOL_RANGO_FIN;
    option routers $P3ASORC_WOL_IP;
    next-server $P3ASORC_WOL_IP;
    filename "pxelinux.0";
}
EOF

# PASO 4: Configurar TFTP y Syslinux
sed -i 's|TFTP_DIRECTORY=.*|TFTP_DIRECTORY="$P3ASORC_WOL_TFTP_DIR"|
$P3ASORC_GRUPO_CONFIG_TFTP
mkdir -p $P3ASORC_WOL_TFTP_DIR/pxelinux.cfg
cp /usr/lib/PXELINUX/pxelinux.0 $P3ASORC_WOL_TFTP_DIR/
cp /usr/lib/syslinux/modules/bios/ldlinux.c32 $P3ASORC_WOL_TFTP_DIR/
cp /usr/lib/syslinux/modules/bios/menu.c32 $P3ASORC_WOL_TFTP_DIR/
cp /usr/lib/syslinux/modules/bios/libutil.c32 $P3ASORC_WOL_TFTP_DIR/

# PASO 5: Descargar Instalador Debian (Netboot)
mkdir -p $P3ASORC_WOL_TFTP_DIR/debian-installer
echo "Descargando kernel e initrd de Debian..."
wget -O $P3ASORC_WOL_TFTP_DIR/debian-installer/linux $P3ASORC_WOL_URL_DEBIAN
    /linux
wget -O $P3ASORC_WOL_TFTP_DIR/debian-installer/initrd.gz
    $P3ASORC_WOL_URL_DEBIAN/initrd.gz

# PASO 6: Crear Fichero Preseed (Instalacion Desatendida)
# Este fichero automatiza las preguntas del instalador.
cat > /etc/apache2/sites-available/000-default.conf <<EOF
<VirtualHost *:80>
    ServerAdmin dbdebian.org
    DocumentRoot /var/www/html
    ErrorLog \$\{APACHE_LOG_DIR\}/error.log
    CustomLog \$\{APACHE_LOG_DIR\}/access.log combined
</VirtualHost>
EOF

mkdir -p $P3ASORC_WOL_WEB_DIR
chown -R www-data:www-data $P3ASORC_WOL_WEB_DIR
chmod 755 $P3ASORC_WOL_WEB_DIR

```

```

mkdir -p /etc/apache2/sites-enabled
ln -sf /etc/apache2/sites-available/000-default.conf /etc/apache2/sites-
enabled/000-default.conf

rm -f $P3ASORC_WOL_WEB_DIR/preseed.cfg
cat > $P3ASORC_WOL_WEB_DIR/preseed.cfg <<EOF
# --- Localizacion ---
d-i debian-installer/locale string es_ES
d-i keyboard-configuration/xkb-keymap select es

# --- Red y DNS ---
d-i netcfg/choose_interface select auto
d-i netcfg/get_hostname string debian-pxe
d-i netcfg/get_domain string local
# CAMBIO: Forzar DNS 1.1.1.1
d-i netcfg/get_nameservers string 1.1.1.1

# --- Espejo ---
d-i mirror/country string manual
d-i mirror/http/hostname string ftp.es.debian.org
d-i mirror/http/directory string /debian
d-i mirror/http/proxy string

# --- Cuentas (root/root y usuario/usuario) ---
d-i passwd/root-login boolean true
d-i passwd/root-password password root
d-i passwd/root-password-again password root
d-i passwd/make-user boolean true
d-i passwd/user-fullname string Usuario PXE
d-i passwd/username string usuario
d-i passwd/user-password password usuario
d-i passwd/user-password-again password usuario

# --- Particionado ---
d-i partman-auto/method string regular
d-i partman-auto/choose_recipe select atomic
d-i partman-partitioning/confirm_write_new_label boolean true
d-i partman/choose_partition select finish
d-i partman/confirm boolean true
d-i partman/confirm_nooverwrite boolean true

# --- Finalizar ---
d-i grub-installer/only_debian boolean true
d-i grub-installer/with_other_os boolean true
d-i grub-installer/bootdev string /dev/sda
d-i finish-install/reboot_in_progress not
EOF

# Ajustar permisos para que Apache pueda leerlo
chmod 644 $P3ASORC_WOL_WEB_DIR/preseed.cfg
chown www-data:www-data $P3ASORC_WOL_WEB_DIR/preseed.cfg

# Permisos lectura web
chmod 644 $P3ASORC_WOL_WEB_DIR/preseed.cfg

# PASO 7: Crear Menu PXE con opcion Desatendida
rm -f $P3ASORC_GRUPO_CONFIG_PXE

```

```

cat > $P3ASORC_GRUPO_CONFIG_PXE <<EOF
DEFAULT menu.c32
PROMPT 0
TIMEOUT 100
MENU TITLE Bienvenido al menu de arranque por red de Debian 13!

LABEL install_auto
  MENU LABEL ^1) Instalar Debian (Desatendido) (item 11)
  KERNEL debian-installer/linux
  APPEND initrd=debian-installer/initrd.gz auto=true priority=critical url=
    http://$P3ASORC_WOL_IP/preseed.cfg interface=auto

LABEL slitaz
  MENU LABEL ^2) Thin Client Grafico (Slitaz Rolling) (item 10)
  KERNEL slitaz/vmlinuz
  APPEND initrd=slitaz/initrd.gz rw root=/dev/null autologin
EOF

# PASO 8: Reiniciar servicios
systemctl restart isc-dhcp-server
systemctl restart tftpd-hpa
systemctl restart apache2

# PASO 9: NAT
echo 1 > /proc/sys/net/ipv4/ip_forward
# Configurar IPTables para que haga de "puente" (NAT)
# Todo lo que venga de la red interna, que salga por la de internet (enp0s8)
# Asegurarse que enp0s8 es la que tiene internet. Si es al revés, cambia
# enp0s8 por enp0s3.
iptables -t nat -A POSTROUTING -o enp0s8 -j MASQUERADE

# Aceptar tráfico de reenvío
iptables -A FORWARD -i enp0s8 -o enp0s3 -m state --state RELATED,ESTABLISHED
  -j ACCEPT
iptables -A FORWARD -i enp0s3 -o enp0s8 -j ACCEPT

#-----
# Valida servicio
#-----
systemctl status isc-dhcp-server --no-pager
systemctl status apache2 --no-pager
# Comprobar que el preseed es accesible via web
curl -I http://localhost/preseed.cfg
ls -l $P3ASORC_WOL_TFTP_DIR/debian-installer/

#-----
# Extraer logs, configs e historial
#-----
rm -rf $P3ASORC_MEMORIA
mkdir -p $P3ASORC_MEMORIA
mkdir -p $P3ASORC_CONFIG

cp $P3ASORC_GRUPO_CONFIG_DHCP $P3ASORC_CONFIG/dhcpd.conf
cp $P3ASORC_GRUPO_CONFIG_TFTP $P3ASORC_CONFIG/tftpd-hpa
cp $P3ASORC_WOL_WEB_DIR/preseed.cfg $P3ASORC_CONFIG/preseed.cfg

systemctl status --no-pager -l isc-dhcp-server > $P3ASORC_LOG
systemctl status --no-pager -l apache2 >> $P3ASORC_LOG

```

```

history > $P3ASORC_HISTORIAL
chmod 777 -R $P3ASORC_MEMORIA

tree $P3ASORC_MEMORIA

#-----
# Comprobacion desde host
#-----
# 1. Preparacion:
#   - Apaga la VM Cliente.
#   - En VirtualBox/VMware, asegurate que la MAC del cliente es conocida.
#   - Configura la BIOS del cliente para aceptar Wake On LAN.

# 2. Despertar al cliente (WOL):
#   - Desde este servidor (Debian), ejecuta:
#     etherwake -i enp0s8 08:00:00:00:00:10
#     etherwake 08:00:00:00:00:10
#     wakeonlan 08:00:00:00:00:10

# 3. Instalacion:
#   - El cliente deberia encenderse solo.
#   - Cargara el menu PXE.
#   - Selecciona "Instalar Debian".
#   - No toques nada. Deberia particionar e instalarse solo (password root: root).

```

Fragmento de código 28: Configuración y comprobación de este servicio en  Debian GNU/Linux 13 (Trixie).

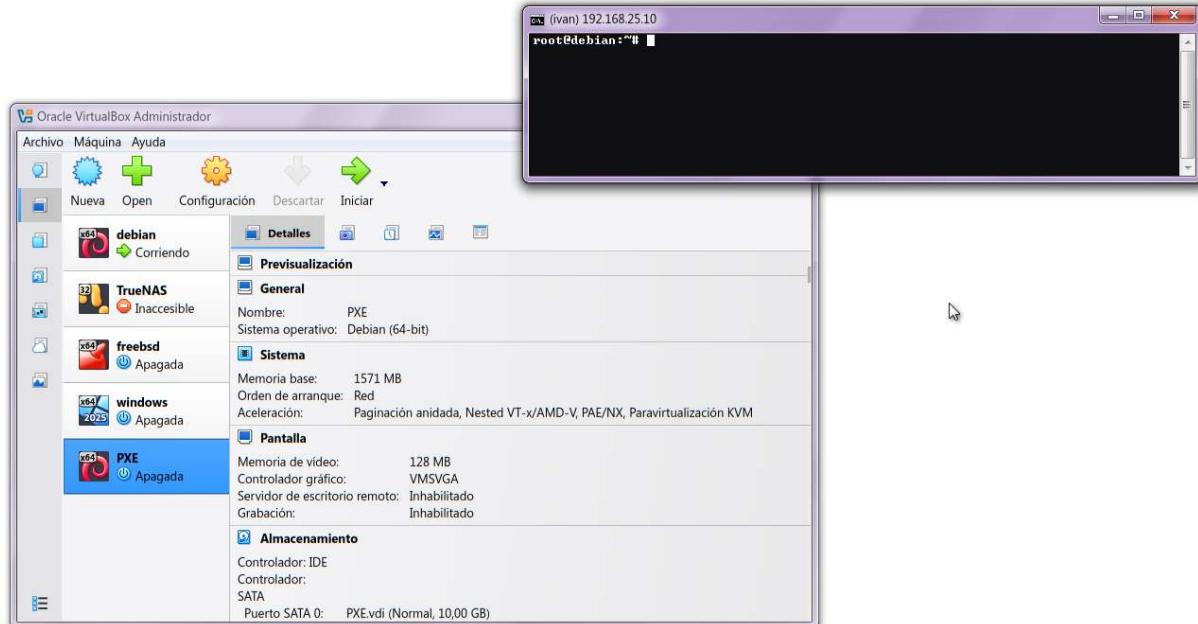


Figura 68: Situación inicial antes de proceder con la instalación desatendida.

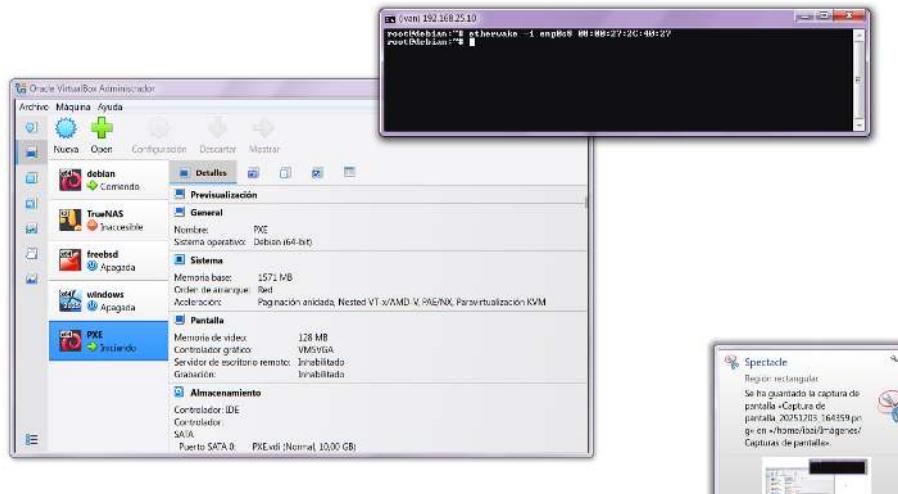


Figura 69: Desde la VM con Debian GNU/Linux 13 (Trixie) emite el paquete mágico que despierta la máquina virtual.

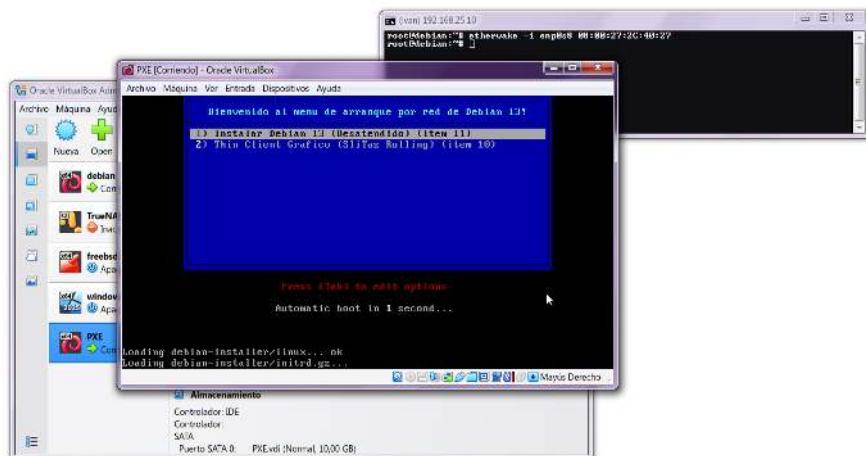


Figura 70: Menú PXE que iniciará la instalación desatendida por agotamiento del temporizador.

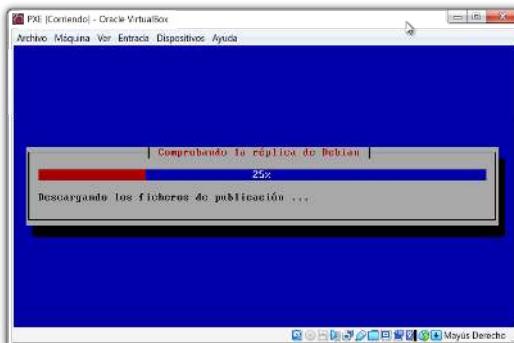


Figura 71: Instalador de Debian GNU/Linux 13 (Trixie) descargando paquetes de Debian GNU/Linux 13 (Trixie). Y el resto es historia.

11.2. ● FreeBSD 14.3 **SIN HACER**

11.3. ■ Microsoft Windows Server 2025 **SIN HACER**

# **Anexos**

---

## A. Definiciones de siglas y abreviaturas

**SW:** *software*, programario. **HW:** *hardware*, lo físico. **VBox:** Oracle VirtualBox. **VM:** Máquina virtual.

## B. Comparación de licencias

Una licencia de SW es un documento de carácter legal que dicta el cómo puede usarse y distribuirse dicho SW. Las licencias declaran qué derechos y limitaciones tienen determinados usuarios en relación con el SW.

La principal diferencia entre las licencias de Microsoft Windows Server 2025 y las de los sistemas *unix-like* (FreeBSD y Debian GNU/Linux) consiste en que son *Free and Open-Source Software* (en adelante, FOSS por sus siglas en inglés). Aunque los hay quienes prefieren referirse a este término como FLOSS.

El movimiento del software libre surgió formalmente en la década de 1980, impulsado en gran medida por Richard Stallman y la fundación GNU. El objetivo de este era crear un sistema operativo completo libre. Para una mayor cobertura legal de esta libertad, Stallman creó la Licencia Pública General de GNU (GPL), que introdujo el concepto de *copyleft*. Esta licencia garantiza que cualquier software derivado de un trabajo GPL debe, a su vez, ser distribuido bajo la misma licencia, preservando las libertades del usuario.

Mientras tanto, en la Universidad de California, Berkeley, se desarrollaban las licencias BSD. Este tipo de licencias de carácter permisivo, ofrecen un contraste al *copyleft*. Permitiendo la modificación e incluso la incorporación de productos propietarios (de código cerrado) sin la obligación de liberar sus propias modificaciones.

Por el contrario, la licencia de **Windows Server 2025** es un contrato comercial (EULA). Es un software propietario y de código cerrado; el usuario paga por el derecho de uso del SW, no por el SW en sí. Su modelo de licenciamiento va en función de la cantidad de núcleos de CPU del servidor y requiriendo Licencias de Acceso de Cliente (CALs) adicionales para cada usuario o dispositivo que acceda al servidor.

Por otro lado, **FreeBSD** y **Debian GNU/Linux** son gratuitos y de código abierto, pero sus licencias representan las dos principales ramas del FOSS:

- **FreeBSD** utiliza la licencia BSD. Es permisiva. Puede modificarse y usarse como quisiere.
- **Debian GNU/Linux** emplea la licencia GNU GPL. Permite modificaciones bajo la misma licencia.

A grandes rasgos y a modo de resumen:

- FreeBSD y Debian GNU/Linux son de código abierto (y también gratuitos) y utilizan las licencias BSD y GPL respectivamente.
- La licencia BSD es permisiva y permite todo tipo de modificaciones, e incluso hacerlo privativo. Muy usado en componentes de código abierto.
- La licencia GPL es libre y permite todo tipo de modificaciones siempre cuando le libere bajo la misma licencia. Muy usado en el SW libre.
- En cambio, Microsoft Windows Server 2025 está licenciado bajo una licencia privativa personalizada.

Característica	FreeBSD	Debian GNU/Linux	Windows Server 2025
Coste	Gratis	Gratis	Pago (núcleos, CALs)
Código Fuente	Disponible	Disponible	Propietario
Licencia Principal	BSD (2-Cláusulas)	GNU GPL	EULA (Propietaria)
Filosofía	Permisiva	<i>Copyleft</i>	Comercial
Modificaciones	Sí	Sí (misma licencia)	No

Cuadro 1: Comparativa resumida de licencias.

## C. Estructura de los ficheros de entrega

```

doc/documento.pdf
unix/ftp/unix.log
unix/ftp/historyunix.txt
unix/ftp/ficheros_configuracion/
unix/chat/unix.log
unix/chat/historyunix.txt
unix/chat/ficheros_configuracion/
unix/grupo/unix.log
unix/grupo/historyunix.txt
unix/grupo/ficheros_configuracion/
unix/proxy/unix.log
unix/proxy/historyunix.txt
unix/proxy/ficheros_configuracion/
unix/router_vpn_firewall/unix.log
unix/router_vpn_firewall/historyunix.txt
unix/router_vpn_firewall/ficheros_configuracion/
unix/nagios/unix.log
unix/nagios/historyunix.txt
unix/nagios/ficheros_configuracion/
unix/raid/unix.log
unix/raid/historyunix.txt
unix/raid/ficheros_configuracion/
unix/backup/unix.log
unix/backup/historyunix.txt
unix/backup/ficheros_configuracion/
unix/mail/unix.log
unix/mail/historyunix.txt
unix/mail/ficheros_configuracion/
unix/pxe/unix.log
unix/pxe/historyunix.txt
unix/pxe/ficheros_configuracion/
unix/wol/unix.log
unix/wol/historyunix.txt
unix/wol/ficheros_configuracion/
linux/ftp/linux.log
linux/ftp/historylinux.txt
linux/ftp/ficheros_configuracion/

```

linux/chat/linux.log  
linux/chat/historylinux.txt  
linux/chat/ficheros\_configuracion/  
linux/grupo/linux.log  
linux/grupo/historylinux.txt  
linux/grupo/ficheros\_configuracion/  
linux/proxy/linux.log  
linux/proxy/historylinux.txt  
linux/proxy/ficheros\_configuracion/  
linux/router\_vpn\_firewall/linux.log  
linux/router\_vpn\_firewall/historylinux.txt  
linux/router\_vpn\_firewall/ficheros\_configuracion/  
linux/nagios/linux.log  
linux/nagios/historylinux.txt  
linux/nagios/ficheros\_configuracion/  
linux/raid/linux.log  
linux/raid/historylinux.txt  
linux/raid/ficheros\_configuracion/  
linux/backup/linux.log  
linux/backup/historylinux.txt  
linux/backup/ficheros\_configuracion/  
linux/mail/linux.log  
linux/mail/historylinux.txt  
linux/mail/ficheros\_configuracion/  
linux/pxe/linux.log  
linux/pxe/historylinux.txt  
linux/pxe/ficheros\_configuracion/  
linux/wol/linux.log  
linux/wol/historylinux.txt  
linux/wol/ficheros\_configuracion/  
windows/ftp/windows.csv  
windows/ftp/ficheros\_configuracion/  
windows/chat/windows.csv  
windows/chat/ficheros\_configuracion/  
windows/grupo/windows.csv  
windows/grupo/ficheros\_configuracion/  
windows/proxy/windows.csv  
windows/proxy/ficheros\_configuracion/  
windows/router\_vpn\_firewall/windows.csv  
windows/router\_vpn\_firewall/ficheros\_configuracion/  
windows/nagios/windows.csv  
windows/nagios/ficheros\_configuracion/  
windows/raid/windows.csv  
windows/raid/ficheros\_configuracion/  
windows/backup/windows.csv  
windows/backup/ficheros\_configuracion/  
windows/mail/windows.csv  
windows/mail/ficheros\_configuracion/

```
windows/pxe/windows.csv
windows/pxe/ficheros_configuracion/
windows/wol/windows.csv
windows/wol/ficheros_configuracion/
```

## D. Esquema de particionado optado

```
root@debian:~# lsblk
NAME   MAJ:MIN RM  SIZE RO TYPE  MOUNTPOINTS
sda      8:0    0   48G  0 disk
|---sda1  8:1    0  953M 0 part  /boot
|---sda2  8:2    0     1K 0 part
|---sda5  8:5    0  1,9G 0 part  [SWAP]
|---sda6  8:6    0 14,9G 0 part  /
|---sda7  8:7    0  2,8G 0 part  /home
|---sda8  8:8    0  1,9G 0 part  /tmp
+-sda9  8:9    0 25,6G 0 part  /var
```

Fragmento de código 29: Esquema de particionado usado en la unidad principal bajo  Debian GNU/Linux 13 (Trixie).

```
root@bsd:~ # df -h
Filesystem      Size   Used  Avail Capacity  Mounted on
/dev/ada0s1a     25G   10G   13G   43%       /
devfs           1.0K    0B   1.0K    0%       /dev
/dev/ada0s1d     19G   3.5G   14G   20%       /var
```

Fragmento de código 30: Esquema de particionado usado en la unidad principal bajo  FreeBSD 14.3.

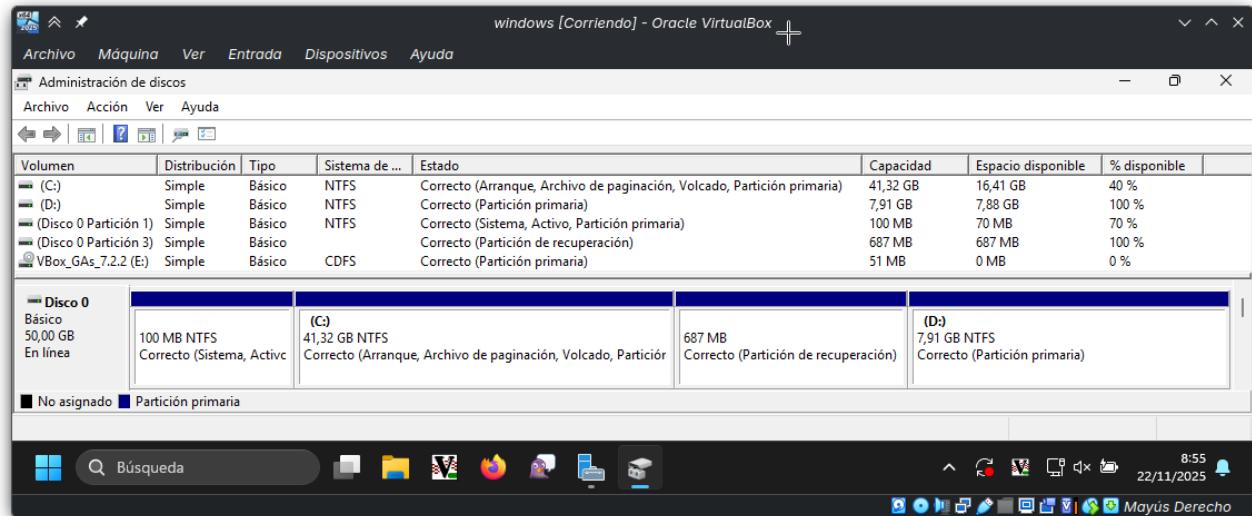


Figura 72: Esquema de particionado usado en la unidad principal bajo  Microsoft Windows Server 2025. La unidad D: estará a merced de los usuarios para administrar sus contenidos.

## E. Scripts de utilidad

### E.1. Preinicialización con nmcli

Es posible que para la validación de ciertos puntos (debido a la simplificación del proceso de verificación desde el anfitrión) se requiera, al menos en mi caso, la ejecución de un *script* en el *host* Fedora Linux para cumplir con los objetivos de la práctica.

```
#!/bin/bash

if [ "$EUID" -ne 0 ]; then
    echo "Se necesitan privilegios de administrador."
    pkexec "$0" "$@"
    exit $?
fi

echo '# Loopback entries; do not change.
# For historical reasons, localhost precedes localhost.localdomain:
127.0.0.1    localhost localhost.localdomain localhost4 localhost4.
               localdomain4
::1          localhost localhost.localdomain localhost6 localhost6.
               localdomain6
# See hosts(5) for proper format and other examples:
# 192.168.1.10 foo.example.org foo
# 192.168.1.13 bar.example.org bar
192.168.25.10 nextdebian.org dbdebian.org web1debian.org web2debian.org
                  debian.debian.asorc.org
192.168.25.11 nextbsd.org dbbsd.org web1bsd.org web2bsd.org bsd.asorc.org
192.168.25.12 nextms.org dbms.org web1ms.org web2ms.org msasorcorg' > /etc
               /hosts

CON_NAME="vboxnet0"
IF_NAME="vboxnet0"
IP_CONFIG="192.168.25.1/24"

if nmcli con show "$CON_NAME" &> /dev/null; then
    nmcli con modify "$CON_NAME" \
        ifname "$IF_NAME" \
        ipv4.method "manual" \
        ipv4.address "$IP_CONFIG" \
        ipv6.method "ignore"
else
    nmcli con add \
        type "ethernet" \
        con-name "$CON_NAME" \
        ifname "$IF_NAME" \
        ipv4.method "manual" \
        ipv4.address "$IP_CONFIG" \
        ipv6.method "ignore"
fi

nmcli con up "$CON_NAME"
```

Fragmento de código 31: *Script* que configura interfaces de red y el archivo que maneja la resolución de *hosts*.

## E.2. Tratamiento de paquetes mágicos naponlan

Este *script* de python habilita la posibilidad de usar las llamadas emitidas en los paquetes mágicos para encender máquinas de VBox aunque éstas estén completamente apagadas con VBox cerrado.

```
#!/usr/bin/python3
''

This program is free software: you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation, either version 3 of the License, or
(at your option) any later version.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with this program. If not, see <http://www.gnu.org/licenses/>.
''

import socket
import struct
import sys
import subprocess
import re
import os
import select
import time

VM_OWNER_USER = os.environ.get("SUDO_USER") or "ibai"

# Variables globales para los sockets
sock_udp = None
sock_raw = None
sockets_list = []

def cleanup_sockets():
    '''
    Cierra todos los sockets de escucha abiertos.
    '''

    global sockets_list
    if sockets_list:
        print("\n[INFO] Cerrando sockets...")
        for s in sockets_list:
            try:
                s.close()
            except Exception:
                pass

def get_vm_mac_map():
    '''
    Busca todas las máquinas virtuales registradas por el usuario
    VM_OWNER_USER,
    extrae sus MACs de red y crea un diccionario que mapea la MAC al nombre
    exacto de la VM,
    solo para las VMs que NO están en estado 'running'.
    '''
```

```

Retorna:
    dict: Diccionario donde la clave es la MAC y el valor es el nombre
de la VM.
    '',
mac_map = {}
try:
    # Usamos -H para asegurar que la variable HOME es la del usuario
    list_command = ["sudo", "-H", "-u", VM_OWNER_USER, "VBoxManage", "list", "vms"]
    result = subprocess.run(list_command, capture_output=True, text=True,
                           check=True)
    vm_lines = result.stdout.strip().split('\n')

    vm_names = []
    for line in vm_lines:
        match = re.search(r'^(.*?)"', line)
        if match and match.group(1) != "<inaccessible>":
            vm_names.append(match.group(1))

    print(f"[*] VMs registradas encontradas para {VM_OWNER_USER}: {', '.join(vm_names)}")

    for name in vm_names:
        info_command = ["sudo", "-H", "-u", VM_OWNER_USER, "VBoxManage",
                        "showvminfo", name, "--machinereadable"]
        info_result = subprocess.run(info_command, capture_output=True,
                                     text=True, check=True)

        state_match = re.search(r'VMState=(.*?)"', info_result.stdout)
        state = state_match.group(1) if state_match else "unknown"

        for line in info_result.stdout.strip().split('\n'):
            mac_match = re.search(r'macaddress(\d+)=(.*?)"', line)

            if mac_match:
                nic_number = mac_match.group(1)
                raw_mac = mac_match.group(2)
                formatted_mac = ":".join(raw_mac[i:i+2] for i in range
(0, len(raw_mac), 2)).upper()

                print(f"[*] VM '{name}' (NIC {nic_number}) -> MAC: {formatted_mac} (Estado: {state})"

                if raw_mac != "000000000000" and state != "running":
                    mac_map[formatted_mac] = name
                elif state == "running":
                    print(f"[*] MAC {formatted_mac} ignorada: VM '{name}' ya está ejecutándose.")

except subprocess.CalledProcessError as e:
    err_msg = e.stderr.strip() if e.stderr else "Sin detalles de error"
    print(f"[ERROR] No se pudo ejecutar VBoxManage como {VM_OWNER_USER}. {err_msg}")
    raise
except Exception as e:
    print(f"[ERROR] Error al parsear la informacion de las VMs: {e}")
    raise

```

```

    return mac_map

def start_vm(vm_name):
    """
    Ejecuta el comando VBoxManage para encender la máquina virtual
    especificada.
    """
    print(f"[WOL] Arrancando VM: {vm_name}")
    # Añadido -H para establecer HOME y capture_output para manejar errores
    start_command = ["sudo", "-H", "-u", VM_OWNER_USER, "VBoxManage", "startvm", vm_name, "--type", "headless"]

    # Ejecutamos con capture_output=True para obtener stderr si falla
    subprocess.run(start_command, check=True, capture_output=True, text=True)
    print(f"[WOL] VM {vm_name} arrancada.")

def setup_sockets():
    """
    Configura y devuelve la lista de sockets Capa 2 (RAW) y Capa 4 (UDP).
    """
    global sock_udp, sock_raw, sockets_list

    # 1. Socket UDP (Capa 4)
    try:
        sock_udp = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
        sock_udp.setsockopt(socket.SOL_SOCKET, socket.SO_BROADCAST, 1)
        sock_udp.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
        sock_udp.bind(('', 9))
        sockets_list.append(sock_udp)
        print("[*] Socket UDP vinculado al puerto 9 (wakeonlan).")
    except Exception as e:
        print(f"[ERROR] Fallo al crear socket UDP: {e}.") 

    # 2. Socket RAW (Capa 2)
    try:
        sock_raw = socket.socket(socket.AF_PACKET, socket.SOCK_RAW, socket.htons(3))
        sockets_list.append(sock_raw)
        print("[*] Socket RAW vinculado a ETH_P_ALL (ether-wake).")
    except Exception as e:
        print(f"[ERROR] Fallo al crear socket RAW (requiere root): {e}.") 

    return sockets_list

def listen_wol(initial_mac_map):
    """
    Bucle principal que escucha en ambos sockets y gestiona la activación de
    VMs.
    """
    global sockets_list
    current_mac_map = initial_mac_map

    print(f"[*] Monitorizando Capa 2 (Ethernet) y Capa 4 (UDP :9) en TODAS
    las interfaces...")

    while True:

```

```

try:
    readable, _, _ = select.select(sockets_list, [], [])

    for s in readable:
        try:
            data, _ = s.recvfrom(2048)
        except Exception as e:
            print(f"[ERROR] Error al recibir datos en el socket: {e}")
        continue

    sync_stream = b'\xff' * 6

    if sync_stream in data:
        idx = data.find(sync_stream)
        mac_part_start = idx + 6
        mac_part_end = mac_part_start + (16 * 6)

        if len(data) >= mac_part_end:
            mac_data = data[mac_part_start:mac_part_end]
            target_mac_bytes = mac_data[:6]

            if mac_data == target_mac_bytes * 16:
                mac_str = ':'.join(f'{b:02x}' for b in
target_mac_bytes).upper()
                print(f"\n{'-'*50}")
                print(f"[*] Paquete Mágico detectado para MAC: {mac_str}")
                print(f"[*] Actualizando mapa de VMs...")

                try:
                    current_mac_map = get_vm_mac_map()
                except Exception as e:
                    print(f"[ERROR] Fallo al actualizar el mapa: {e}.") 

                if mac_str in current_mac_map:
                    vm_name = current_mac_map[mac_str]
                    try:
                        start_vm(vm_name)
                    except subprocess.CalledProcessError as e:
                        # Aquí capturamos el error real gracias
a capture_output=True
                        err_out = e.stderr.strip() if e.stderr
else "Sin detalles"
                        print(f"[ERROR] Fallo al iniciar VM '{vm_name}': {err_out}")
                    except Exception as e:
                        print(f"[ERROR] Fallo inesperado al
iniciar VM '{vm_name}': {e}.")
                    else:
                        print(f"[WOL] MAC {mac_str} ignorada (no
mapeada o VM encendida).")
                        print(f"{'-'*50}\n")

                except select.error as e:
                    if e.errno == 4:
                        continue

```

```

        print(f"[ERROR] Error en select: {e}.")
        time.sleep(1)
    except KeyboardInterrupt:
        raise
    except Exception as e:
        print(f"[ERROR] Error bucle principal: {e}.")
        time.sleep(1)

if __name__ == "__main__":
    if os.geteuid() != 0:
        print("[WARN] Este programa requiere elevación (sudo).")

    print(f"[*] Usando al usuario '{VM_OWNER_USER}' para consultar y
arrancar VMs.")

    try:
        MAPPING = get_vm_mac_map()

        if not MAPPING:
            print("[WARN] No hay máquinas apagadas que mapear.")
            sys.exit(0)

        setup_sockets()
        listen_wol(MAPPING)

    except KeyboardInterrupt:
        cleanup_sockets()
        print("[INFO] Programa terminado.")
        sys.exit(0)
    except Exception:
        cleanup_sockets()
        sys.exit(1)

```

Fragmento de código 32: *Script naponlan.*

### E.3. Problema al establecer una red *Host-Only*

El siguiente escrito tiene el propósito de hacer frente a un problema vigente en las instalaciones de VBox en hosts Linux (al menos así ocurre en Fedora Linux y Debian GNU/Linux 13 (Trixie)).

```

su -

mkdir -p /etc/vbox
echo "* 10.0.0.0/8 192.168.0.0/16" | sudo tee /etc/vbox/networks.conf
systemctl reload vboxdrv
systemctl restart vboxdrv
VBoxManage hostonlyif ipconfig vboxnet0 --ip 192.168.25.1 --netmask
255.255.255.0

```

Fragmento de código 33: *Script* para solventar problemática de VBox.

### E.4. Makefile empleado para compilar T<sub>E</sub>X

```

TEX = pdflatex
PROJECT = document

```

```

OUT_LITE = $(PROJECT)_lite
BUILD_DIR = build
BULLSHIT = SUPERMEMORIA
L = $(BULLSHIT)/linux
U = $(BULLSHIT)/unix
W = $(BULLSHIT)/windows
USELESS = *.aux *.log *.out *.toc
COMPRESS = zip
RESULT = asorc_practica3.zip

all:
    $(TEX) $(PROJECT).tex
    $(TEX) $(PROJECT).tex
    $(TEX) $(PROJECT).tex

convert:
    gs -sDEVICE=pdfwrite -dCompatibilityLevel=1.4 -dPDFSETTINGS=/ebook -
        dNOPAUSE -dQUIET -dBATCH -sOutputFile=$(OUT_LITE).pdf $(PROJECT).pdf

tiny: convert
    mv $(OUT_LITE).pdf $(PROJECT).pdf

zip:
    rm -rf $(BUILD_DIR) $(RESULT)
    mkdir -p $(BUILD_DIR)/doc
    cp $(PROJECT).pdf $(BUILD_DIR)/doc
    cp -r $(L) $(BUILD_DIR)
    cp -r $(U) $(BUILD_DIR)
    cp -r $(W) $(BUILD_DIR)
    $(COMPRESS) -r $(RESULT) $(BUILD_DIR)/*
    rm -f $(USELESS) $(RESULT)

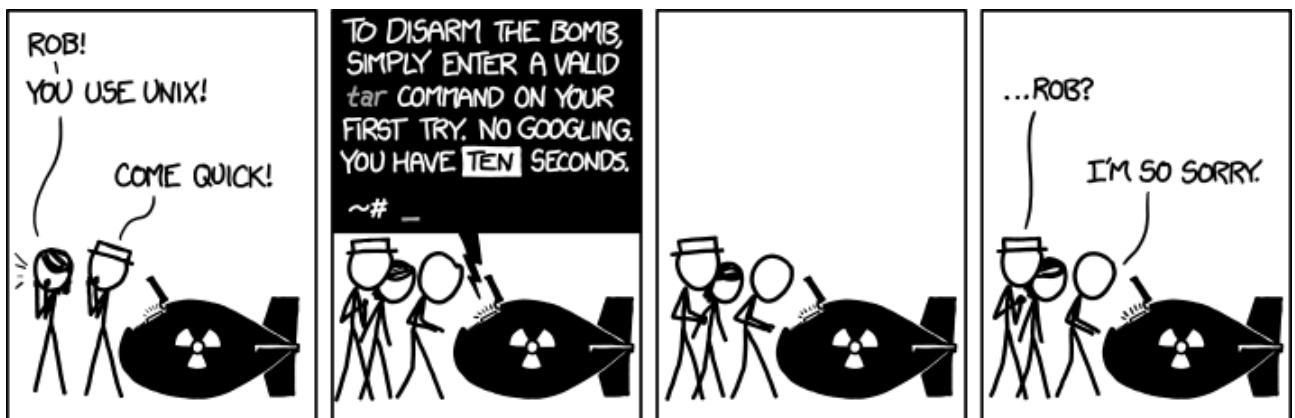
clean:
    rm -f $(USELESS) $(RESULT)

```

Fragmento de código 34: Escrito encargado de la compilación de este documento.

# FIN

*Gracias por su atención*



*Se nota que no se ha hecho el punto Copias de seguridad*