

## Iniciándonos en el Proyecto Linux Terminal Server

# A SU SERVICIO

El proyecto LTSP (Linux Terminal Server Project) ofrece un método exhaustivo para el servicio de terminal en Linux, incluyendo la facilidad de acceso a tarjetas de sonido locales, impresoras o memorias USB.

**POR CHRISTIAN KROLL**

Unix es una plataforma ideal para servidores de terminal. Los numerosos servicios con capacidad de red, como la interfaz gráfica X11, la terminal remota SSH o el sistema de archivos NFS, son estándares maduros y estables desde hace años. Pero una buena computación basada en servidores implica que la experiencia del usuario ha de ser idéntica a la de trabajar directamente con la aplicación.

Los programas deben reproducir su sonido en el sistema del cliente, aunque la aplicación se ejecute en el servidor; las memorias USB insertadas en el cliente deben ser accesibles desde las aplicaciones en ejecución en el servidor; y las impresoras conectadas al cliente deben ser visibles desde el sistema servidor de terminal.

El proyecto *Linux Terminal Project* (LTSP) [1] es un proyecto de código abierto comprometido con la construcción de un sistema avanzado de servidor de terminal para Linux. LTSP combina *X Window*, *SSH*, y *NFS* para ofrecer acceso a impresoras locales, dispositivos y tarjetas de sonido en configuraciones de arranque por red. Combina también de forma inteligente servicios de red esenciales para dar lugar a una solución de servidor de terminal muy completa, fácil de instalar y de mantener. La potencia y facilidad de configuración de LTSP le convierte en el favorito de las escuelas y los proyectos de ayuda al desarrollo.

### Una Mirada Interior

En el corazón de cualquier instalación de LTSP hay un entorno *chroot* que contiene todos los sistemas necesari-

os. Un servidor de X mínimo se inicia en el cliente ligero, y automáticamente conecta con el servidor de terminal. Hasta la versión 4 de LTSP, el entorno de *chroot* era una distribución separada con un juego de herramientas de administración. De cualquier modo, el entorno comprendía principalmente una combinación de proyectos como *Glibc* o *X.org*, lo que a menudo se convertía en un problema. Estos proyectos se encuentran en continuo desarrollo y reciben regularmente actualizaciones de seguridad, teniendo que ser portadas a LTSP.

### LTSP 5

Con LTSP 5 se optó por algo distinto. En lugar de publicar una recolección de proyectos, como se hacía anteriormente, los programadores se centraron en los componentes principales de LTSP. Ahora podemos disfrutar de acceso por parte del servidor a los periféricos conectados a los clientes ligeros, además de los scripts de inicio y las configuraciones personalizadas.

En LTSP 5 es la distribución del servidor quien proporciona el resto

de paquetes del programa, así como la mayor parte de las herramientas. Esto permite una emulación granular del entorno de *chroot*, la actualización de los programas instalados y la instalación de otros nuevos vía *apt-get*.

Aparte de Debian, actualmente es Ubuntu el mayor contribuyente al desarrollo de LTSP 5. Esto explica porqué son estas dis-

tribuciones, y sus ramificaciones *Edebuntu* y *Skolelinux*, quienes integran las implementaciones de LTSP 5 más avanzadas. Si nos encontramos configurando un servidor de terminal para LTSP, las estimaciones acerca de los requerimientos de hardware variarán enormemente dependiendo del uso que le vayamos a dar. En escenarios típicos de oficinas, el tamaño de memoria mínimo ronda los 256MB de RAM para el servidor, más 64MB de RAM por cada usuario registrado en él. Una CPU de gama media, corriendo a 2 GHz, es suficiente para 15 ó 20 usuarios en un entorno



SCOTT MAXWELL, FOTOLIA

con aplicaciones de oficina exclusivamente; las aplicaciones multimedia necesitan una CPU de gama alta con varios núcleos. Las aplicaciones Flash, Java, de vídeo y 3D someten al sistema a una tensión muy importante.

Si no se está seguro, es mejor observar detenidamente los requisitos de memoria y de CPU para el tipo de aplicaciones elegidas y multiplicarlos por el número de usuarios. Hemos de asegurarnos de prever un margen de seguridad; si el servidor se queda sin memoria principal, comenzará a hacer un uso intensivo de la memoria de intercambio, lo que afectará considerablemente al rendimiento total. La topología de red recomendada por Edubuntu describe una subred separada para los clientes. En ese caso el servidor de terminal debería tener dos tarjetas Ethernet —una de ellas en la subred y la otra conectada al resto de la LAN o a un router. Para diez o más clientes, lo más lógico es usar Gigabit Ethernet en la subred cliente. La mayoría de las veces, los discos duros, sometidos a múltiples accesos simultáneos, acaban siendo un cuello de botella en este tipo de sistemas, por lo que resulta conveniente decantarse por un sistema RAID con una buena velocidad de lectura.

Los discos SATA de bajo coste, con *Native Command Queuing* (NCQ) y 16MB de caché, son una buena opción como componentes RAID. El encolamiento nativo de comandos proporciona a estos discos la capacidad de modificar el orden de las peticiones para elevar el rendimiento en la lectura de los datos. Se trata de una gran ventaja cuando se tiene un número elevado de

peticiones de acceso simultáneas, aunque se necesita un controlador con soporte para NCQ.

Los requerimientos mínimos para un cliente LTSP ligero son: una CPU de 233 MHz, 64MB de RAM, una tarjeta de red Ethernet a 100Mb y un adaptador de gráficos con 2MB de memoria de vídeo. Se recomienda una CPU a 400 MHz y 128MB de RAM. Si pensamos reproducir vídeos o usar aplicaciones 3D, es necesario un adaptador de vídeo con *X video* o con soporte para *GLX*; de cualquier modo, se puede prescindir del disco duro local.

Se recomienda una tarjeta Ethernet con una ROM con capacidad PXE para arrancar los equipos a través de red. Si nuestra *boot ROM* no lo soporta, podemos descargar imágenes ROM *tailor-made*, disponibles para muchas tarjetas, desde el proyecto *Etherboot/gPXE* [2]. Asumimos, que se tenemos un dispositivo de grabación *EEPROM*; de lo contrario, aún podemos descargar una imagen de disco o de CDR desde el proyecto *Etherboot*. En líneas generales, es más práctico arrancar desde la red que desde un dispositivo antiguo, ya que el arranque por red reduce el número de problemas mecánicos y elimina la necesidad de conservar archivos de arranque.

## Instalando LTSP 5

Suponiendo que estamos usando la topología de red descrita, LTSP 5 estará listo para ser usado en cuanto completemos la instalación y los clientes ligeros sean capaces de arrancar sin más.

Añadir LTSP a un sistema Ubuntu es algo más costoso. Primero tenemos que asegurarnos de que las tarjetas Ethernet del terminal designado están configuradas correctamente. La configuración se encuentra en el fichero */etc/network/interfaces*. El Listado 1 muestra una configuración de ejemplo basada en la topología descrita por Edubuntu. La tarjeta *eth0* está conectada a la subred común y configurada automáticamente a través de la red; *eth1* sirve a la subred del cliente ligero y está configurada con la dirección IP estática *192.168.0.254*.

Para más detalles, ejecutar *man 5 interfaces*. Para prescindir de los asistentes y optimizar las configuraciones de nuestras interfaces de red, es buena idea desinstalar los paquetes *avahi-autoipd* y *network-manager*. Este comando aplica la configuración:

```
sudo invoke-rc.d \
networking restart
```

El siguiente paso es instalar el paquete *ltsp-server-standalone*, que incluye el script *ltsp-build-client* para la configuración del entorno *chroot* del sistema cliente.

El software tiene un par de dependencias con algunos servicios esenciales para la ejecución del servidor de terminal, aunque nuestro gestor de paquetes debería manejarlas sin necesidad de hacer nada más. Además de esto, el servidor de *SSH* debe estar instalado, puesto que *LTSP* se sirve de él para el mecanismo de autenticación y registro predeterminado.

Cuando llamamos al script *ltsp-build-client*, éste descarga automáticamente de los repositorios los paquetes que necesita para el *chroot* y ejecuta *debootstrap*, insta-

## Inseguridad del Sistema de Ficheros

*no\_root\_squash* demuestra una vulnerabilidad inherente a NFS: La versión 3, usada por LTSP, no soporta la autenticación. Esta carencia tan seria ha ocasionado que NFS se gane apellidos tan despectivos como 'No Filesystem Security' (*Sistema de Ficheros Inseguro* u otros aún peores). El cliente sólo tiene que decirle al servidor qué usuario e ID de grupo desea usar para que éste le brinde el pertinente acceso. Una vez el sistema ha sido comprometido, el atacante puede suplantar la identidad de cualquier usuario.

Para prevenir desastres, NFS suele dirigir al usuario *root* hacia *nobody*, pero la opción *no\_root\_squash* hace que sea inútil. Dado que el sistema de ficheros es la raíz del sistema desde el punto de vista del cliente, éste debe tener acceso al mismo con privilegios de *root*. El hecho de que el servidor exporte el directorio en modo de sólo lectura previene ataques al servidor de terminal; nunca se deben almacenar datos confidenciales en el *chroot*.

NFS tiene otro inconveniente. El rendimiento se desploma cuando se producen muchos accesos simultáneos a ficheros pequeños. Como ya sabemos, es justo eso lo que tienen que hacer los scripts de inicio al momento de arrancar la máquina. El administrador debería evitar que arranquen demasiadas máquinas simultáneamente. Su lugar, es más recomendable arrancar pequeños grupos de máquinas en intervalos de tiempo regulares.

### Listado 1: */etc/network/interfaces*

```
01 # interfaz de loopback
02 auto lo
03 iface lo inet loopback
04
05 # la tarjeta de la subred
   regular
06 subnet
07 auto eth0
08 iface eth0 inet dhcp
09
10 # la tarjeta del subred del
   cliente ligero
11 auto eth1
12 iface eth1 inet static
13 address 192.168.0.254
14 netmask 255.255.255.0
15 network 192.168.0.0
16 broadcast 192.168.0.255
```

lando un sistema Ubuntu mínimo en `/opt/ltsp/i386`. La principal diferencia con el sistema anfitrión, además de la espartana selección de paquetes instalados, radica en la inclusión del cliente `ltsp-client`, que incluye a su vez los scripts de inicio necesarios para las operaciones del cliente sin disco. Podemos administrar el sistema tal y como lo haríamos en una instalación normal de Ubuntu. La actualización por red tiene un aspecto como éste:

```
sudo chroot /opt/ltsp/i386
apt-get update
apt-get upgrade
exit
```

Antes de permitir a los clientes arrancar a través de la red, nos hacemos `root` y nos aseguramos de que los servicios necesarios están configurados correctamente y son accesibles.

## Conexión del Cliente

El cliente usa el protocolo DHCP (Dynamic Host Configuration Protocol) para recibir toda la configuración necesaria, como dirección IP, subred, pasarela predeterminada y otros parámetros. El demonio `dhcpd3`, que maneja todas las peticiones de la subred, se ejecuta en una de las máquinas (normalmente en el mismo servidor de terminal). La configuración se encuentra en el fichero `/etc/ltsp/dhcpd.conf`; en el Listado 2 tenemos un ejemplo con comentarios.

Si se está interesado en indagar en las opciones del demonio DHCP, puede consultarse *man 5 dhcpd.conf*. Afortunadamente, no suele ser necesario alterar la configuración proporcionada por el paquete `ltsp-server-standalone`. Lo principal es asegurarse de que la subred concuerda con la configuración de la tarjeta Ethernet en los clientes que estamos manejando. Las entradas `filename` y `option rootpath` también afectan a LTSP (ver las secciones *PXE* y *NFX*). En LTSP, es TFTP (Trivial File Transfer Protocol) quien proporciona al cliente el cargador de arranque y el núcleo. *TFTP* es una versión reducida de *FTP*. No soporta la autenticación de usuario ni ningún tipo de permisos de archivo. No lista directorios, y el tamaño máximo de fichero es de 32 MB. Por otro lado, la simplicidad del protocolo le hace idóneo para implementaciones sobrias, que caben dentro de la ROM de arranque de una tarjeta Ethernet. Edubuntu 7.04 usa por defecto el demonio `tftpd-hpa` como servidor TFTP. El demonio se inicia

mediante `inetd`. El directorio raíz de TFTP suele ser `/var/lib/tftpboot`, el lugar donde debemos guardar el cargador de arranque y el kernel junto con el disco RAM. Después de modificar el fichero de kernel, el script `ltsp-update-kernels` lo copia al entorno `chroot` y actualiza automáticamente los enlaces al cargador de arranque durante el proceso.

*PXE* (Preboot Execution Environment) es un estándar de Intel que dota a cualquier PC de la capacidad de arrancar automáticamente a través de la red desde sus tarjetas Ethernet. Para hacerlo, *PXE* conecta primero con el servidor DHCP, solicita una dirección IP y parsea las opciones del

fichero DHCP como se muestra en el Listado 2.

Esto le da al cliente una ruta TFTP al cargador de arranque que ha de descargar y ejecutar. Para ello, LTSP usa `Pxelinux`, del paquete `Syslinux` [4]; descarga el kernel y un disco RAM y arranca el cliente. El proyecto Etherboot ayuda a que los PCs sin tarjetas con capacidad para arranque PXE puedan arrancar por red. En lugar de un cargador de arranque, el cliente usa una imagen de kernel modificada por el programa `mknbi`. La configuración mostrada en el Listado 2 dota al servidor DHCP de la capacidad para detectar si hay algún cliente PXE, o de otro tipo, solicitando el cargador de

### Listado 2: `/etc/ltsp/dhcpd.conf`

```
01 # opciones obligatorias para
02 # todos los clientes
03 authoritative;
04
05 group {
06 # Parámetros típicos en una red IP
07 option domain-name "ejemplo.com";
08 option domain-name-servers 192.168.0.1;
09 option broadcast-address 192.168.0.255;
10 option routers 192.168.0.1;
11 option subnet-mask 255.255.255.0;
12
13 # Ruta al cargador de arranque en el
14 # servidor TFTP para PXE y Etherboot
15 if substring( option vendor-class-identifier, 0, 9 ) = "PXEClient"
16 {
17 filename "/ltsp/i386/pxelinux.0";
18 }
19 else{
20 filename "/ltsp/i386/nbi.img";
21 }
22
23 # La raíz de NFS a montar
24 option root-path "/opt/ltsp/i386";
25
26 # La máquina "foo", con dirección MAC
27 # "00:11:22:33:44:55"
28 # siempre recibirá la dirección IP
29 # "192.168.0.10"
30 host foo {
31 hardware ethernet 00:11:22:33:44:55;
32 fixed-address 192.168.0.10;
33 }
34
35 # El resto de máquinas de la
36 # subred "192.168.0.0/24"
37 # recibirán direcciones IP
38 # aleatorias entre
39 # "192.168.0.20" y "192.168.0.250"
40 subnet 192.168.0.0 netmask 255.255.255.0 {
41 range 192.168.0.20 192.168.0.250;
42 }
43 }
```

arranque y respondiendo con la ruta correspondiente.

GPXE es una implementación libre de PXE; ambos, Etherboot y GPXE, están disponibles en el sitio etherboot.org [2].

## Network File System

Como los clientes ligeros no suelen disponer de discos locales, sus sistemas de ficheros raíz son remotos. NFS (Network File System) es un protocolo útil para su manejo. El fichero `/etc/exports` contiene las configuraciones para el sistema de ficheros del cliente ligero.

Las entradas para las exportaciones están formadas por tres partes. La primera es la ruta absoluta hacia el directorio a exportar, seguida de un listado que limita las estaciones autorizadas. En ese listado se pueden incluir nombres de host o direcciones de subred (incluyendo comodines). El último componente define opciones específicas de NFS que determinan el comportamiento del servidor de NFS. El paquete `ltp-server` añade la siguiente entrada a `/etc/exports`:

```
/opt/ltpsp
*(ro,no_root_squash,async)
```

Esta entrada otorga a cualquier host, como indica el comodín `*`, permiso para acceder a la carpeta `/opt/ltpsp`. La opción `ro` (sólo lectura) impide el permiso de escritura. `async` indica al cliente que debe guardar un búfer a la hora de acceder para

### Listado 3: `/opt/ltpsp/i386/etc/lts.conf`

```
01 [default]
02 # con 16 bit de color se
   ahorra ancho de banda
03 X_COLOR_DEPTH=16
04 # paso de dispositivos
   locales a través del servidor
05 LOCALDEV=True
06 # Activamos servidor de
   sonido
07 SOUND=True
08 # distribución de teclado
   español
09 XKBLAYOUT=es
10
11 [00:11:22:33:44:55]
12 # Usar los drivers Vesa
13 XSERVER = vesas
14 # Integrar el ratón serie
15 X_MOUSE_DEVICE=/dev/ttyS0
16 X_MOUSE_
17 PROTOCOL=intellimouse
```

incrementar el rendimiento.

`no_root_squash` hace que cualquier usuario cliente, incluido `root`, pueda entrar en este directorio.

## Localización de Ficheros

Para acceder a su sistema de archivos, el cliente usa primero la opción `root-path` de DHCP (ver el Listado 2) para determinar la localización de su directorio raíz exportado. Es entonces cuando monta dicho directorio como su sistema de ficheros raíz, además de un directorio adicional que depende de su arquitectura, normalmente `i386`. El cliente monta sobre él un sistema de ficheros temporal, alojado en la RAM, para cada lugar con permisos de escritura.

Dicho de otro modo, cada cliente ligero comparte el mismo sistema de ficheros. Es en el archivo `/opt/ltpsp/i386/etc/lts.conf` donde el administrador tiene la opción de personalizar la configuración del cliente individual. Además de las configuraciones globales, podemos usar dicho fichero para las opciones individuales de cada máquina en base a su dirección MAC. El Listado 3 nos muestra un ejemplo con comentarios. Para una comprensión más detallada, así como una descripción de las opciones disponibles en `lts.conf`, consultar la documentación [5].

## Llevando a Local

Las APIs de audio en Linux (OSS, ALSA) suelen ser locales. Para poder enviar sonido de una máquina a otra necesitamos un servidor de sonido con una interfaz de audio transparente para la red. LTSP usa por defecto el servidor PulseAudio [6], desarrollado originariamente bajo el nombre de Polyaudio como sustituto del antiguo ESD (Enlightment Sound Daemon). Ver Figura 1.

PulseAudio soporta extensiones flexibles, en cuanto a que implementa una sofisticada arquitectura de plugins, emula las interfaces del extendido ESD y proporciona su propio plugin de ALSA. No sería demasiado útil sin un soporte para aplicaciones, pero cada vez son más los programas multimedia que soportan esta interfaz.

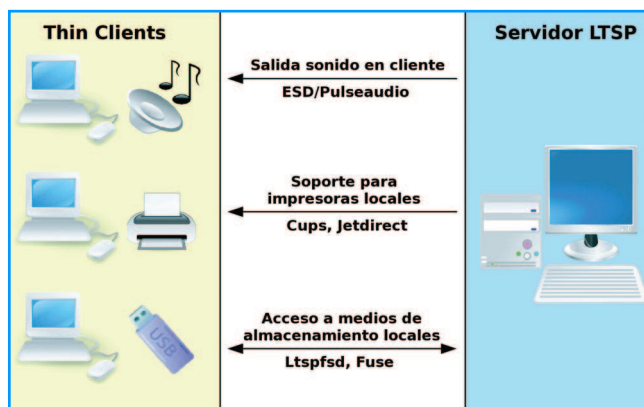


Figura 1: LTSP puede redireccionar las impresoras, el sonido y los dispositivos USB de los clientes.

La variable de entorno `PULSE_SERVER` para el protocolo nativo de PulseAudio nos dice dónde se encuentra el servidor de sonido. El formato de su contenido es `tcp:máquina:puerto`. Alternativamente, la variable `ESPEAKER` apunta a la interfaz emulada y su formato es similar (`máquina:puerto`). Si

## Gestor de Clientes Ligeros

Para que la administración de clientes sea lo más sencilla posible, Edubuntu integra *Thin Client Manager* en el paquete `thin-client-manager-gnome`, donde el administrador puede ver de una sola pasada las actividades que realizan los usuarios registrados.

Una relación de procesos, bastante parecida al monitor del sistema de Gnome, informa al administrador de la cantidad de memoria que están ocupando los procesos, así como de la carga que producen en el sistema. Además de todo esto, el administrador puede iniciar y terminar los procesos de cualquier usuario o enviarle mensajes de texto simples.

Si un usuario se comporta indebidamente, el administrador puede bloquear su actividad o incluso forzarle a salir.

El administrador de clientes ligeros incluye un visor de VNC, con el que se puede ver el escritorio del usuario en todo momento. Son necesarias algunas configuraciones adicionales para poder hacerlo, ya que los clientes no contienen servidores VNC por defecto. Pueden instalarse los paquetes necesarios usando el gestor de paquetes desde el `chroot` del servidor, por ejemplo, `x11vnc` desde el repositorio `universe`, ejecutando `apt-get install`.



queremos activar el soporte para el servidor de sonido en los clientes, todo lo que necesitamos es una entrada `SOUND = True` en `lts.conf`.

LTSP puede conectar los dispositivos locales del cliente ligero al servidor de terminal. En cada cliente se ejecuta un demonio llamado `ltspsd`, que monitoriza las memorias USB y los dispositivos conectados. Su homólogo, `ltspsf`, del mismo paquete, se ejecuta en el servidor. `Ltspsf` reacciona a los mensajes de evento del demonio `ltspsd`, montando cada dispositivo detectado. El demonio se basa en el marco `FUSE` [7] para el acceso a dispositivos a través de la red. Por defecto, sólo los usuarios incluidos en el grupo `fuse` pueden montar carpetas. Hay que activar esta función en el fichero `lts.conf`, añadiendo la línea `LOCALDEV = True`.

También es posible acceder a impresoras locales, aunque el soporte no es dinámico. Se comienza creando una entrada para el puerto de la impresora en el fichero `lts.conf` del cliente que la alberga. La entrada `PRINTER_0_DEVICE` incluye el nodo de la interfaz para el dispositivo, como por ejemplo `/dev/usb/lp0` para una impresora USB.

Con `PRINTER_0_TYPE` se define el tipo de interfaz, donde `U` significa USB, `P` paralelo y `S` indica que es serie. La documentación de Edubuntu [5] incluye otros parámetros, para quien desee precisar más.

Estos valores ayudan al script en Python `jet-pipe` a averiguar el puerto correcto. El script suele escuchar en el puerto TCP 9100 y pasar la entrada y la salida de este puerto a la interfaz de la impresora. También es posible usar CUPS en el servidor de terminal

para configurar la correspondiente impresora usando como interfaz un socket a la dirección IP del cliente. Las opciones de configuración de impresoras de Gnome para CUPS almacenan dicha configuración bajo el nombre `AppSocket/JetDirect`.

El ejemplo mostrado sólo funciona mientras no cambie la dirección IP del cliente. Para asegurarnos de que efectivamente hay una impresora conectada a una IP determinada, hemos de asignarle una dirección IP estática mediante el servidor DHCP. El Listado 2, desde la línea 25 a la 32, muestra un ejemplo de asignación estática usando el parámetro `fixed-address`.

### LDM

De forma predeterminada, LTSP 5 usa `LDM` (LTSP Display Manager) para la gestión del acceso gráfico. `LDM` no se ejecuta en el servidor, sino localmente en cada cliente. El acceso se ejecuta en segundo plano y está cifrado por SSH. Esto ocurre porque cuando el usuario ingresa comienza una sesión X; entonces se muestra Gnome, encapsulado en SSH, a través del servidor de las X del cliente. Además, `LDM` ajusta automáticamente la variable de entorno para el servidor de sonido. La salida de sonido de las aplicaciones es enviada directamente al cliente.

### Conclusiones

LTSP camina en la dirección correcta y está consiguiendo que los servicios de terminal basados en Linux sean más populares gracias a la facilidad de su mantenimiento y a una amplia gama de funcionalidades. El Proyecto de Servidor de Terminal en Linux ha dado un gran salto adelante en su viaje a

la versión 5, con unos resultados impresionantes. Sus desarrolladores han logrado un progreso notable respetando la flexibilidad y la facilidad de mantenimiento. Además, a LTSP le beneficia su integración en Edubuntu.

Por otro lado, los requerimientos de hardware son ahora mayores. La versión anterior, LTSP 4, no pedía tanto a los clientes y arrancaba más rápido.

Una vez más, la ayuda viene de camino. La futura versión de LTSP para Ubuntu 'Gutsy Gibbon', en estado de desarrollo, nos adelanta una serie de nuevas funcionalidades útiles. El cambio más significativo es el abandono de NFS como sistema de archivos, ya que circulan rumores sobre seguridad y vemos administradores pidiendo auxilio.

En el futuro, el `chroot` de LTSP se comprimirá en una imagen `Squash FS`, a la que los clientes accederán directamente a través de un dispositivo de bloque en red. Los administradores lo encontrarán más sencillo de proteger que NFS. Implicará también un avance y tiempos de arranque menores.

Gracias a SSH, los atacantes ya no podrán monitorizar el contenido de las sesiones X11, aunque este incremento en la seguridad tiene un coste en carga de CPU. Los clientes X con mayor volumen de tráfico, debido a la visualización de vídeos o animaciones complejas, dejarán al servidor exhausto rápidamente. Por suerte, alguna versión de LTSP venidera soportará conexiones X11 no cifradas, que combinadas con autenticación segura aumentarán el rendimiento en entornos en los que el hardware no sea tan potente. ■

**Tabla 1: Opciones de X11 en lts.conf**

Opción	Significado
XSERVER	Define el driver de la tarjeta gráfica ( <i>ati, nv, vesa, ...</i> )
X_MOUSE_DEVICE	Ruta al nodo del ratón
X_MOUSE_PROTOCOL	Protocolo para el ratón ( <i>ps2, intellimouse, ...</i> )
X_MOUSE_EMULATE3BTN	Emulación del tercer botón de ratón ( <i>True o False</i> )
X_COLOR_DEPTH	Bits de profundidad de color
USE_XFS	Usar servidor de fuentes tipográficas ( <i>True o False</i> )
XFS_SERVER	Dirección para el servidor de fuentes tipográficas
X_HORZSYNC	Rango de frecuencias soportado por el monitor
X_VERTREFRESH	Tasa de refresco soportada por el monitor
XF86CONFIG_FILE	Ruta al fichero de configuración de las X.org locales

### RECURSOS

- [1] Página de inicio de LTSP: <http://www.ltsp.org>
- [2] Etherboot y gPXE: <http://www.etherboot.org>
- [3] ROMs de arranque para tarjetas Ethernet: <http://www.rom-o-matic.net>
- [4] Syslinux y Pxelinux: <http://syslinux.zytor.com>
- [5] Documentación de Edubuntu para `lts.conf`: <http://doc.ubuntu.com/edubuntu/handbook/C/ltsp-client.html>
- [6] Página de inicio de PulseAudio: <http://www.pulseaudio.org>
- [7] Proyecto de FUSE: <http://fuse.sourceforge.net>