

# Servicios en red

Joaquin Andreu



<b>Unidad 1. Servicios DHCP</b>	ISBN 978-84-9003-083-7
<b>Unidad 2. Servicios DNS</b>	ISBN 978-84-9003-084-4
<b>Unidad 3. Servicios de acceso remoto</b>	ISBN 978-84-9003-085-1
<b>Unidad 4. Servicios FTP</b>	ISBN 978-84-9003-086-8
<b>Unidad 5. Gestión de servicios de correo electrónico</b>	ISBN 978-84-9003-087-5
<b>Unidad 6. Gestión de servidores web</b>	ISBN 978-84-9003-088-2
<b>Unidad 7. Interconexión de red</b>	ISBN 978-84-9003-089-9
<b>Unidad 8. Redes inalámbricas</b>	ISBN 978-84-9003-090-5
<b>Unidad 9. Voz IP</b>	ISBN 978-84-9003-091-2
<b>Servicios en red (obra completa)</b>	ISBN 978-84-9771-760-1

# 3

# Servicios de acceso remoto

## vamos a conocer...

1. Terminales en modo texto
2. Terminales en modo gráfico: escritorio remoto
3. VNC (*Virtual Network Computing*)
4. NX

### PRÁCTICA PROFESIONAL

Instalación y configuración de un servidor VNC en Linux y un cliente en Windows

### MUNDO LABORAL

Introducción al SO Google Chrome

## y al finalizar esta unidad...

- Instalarás servicios de acceso remoto en línea de comandos.
- Instalarás servicios de acceso remoto en modo gráfico.
- Probarás el acceso y la administración remotos entre sistemas de distinta naturaleza.



## CASO PRÁCTICO INICIAL

### situación de partida

Dulce es una ex administradora de red que, tras trabajar en una gran empresa, ahora ejerce como *freelance*, autónoma, llevando la gestión de redes de varias pymes (pequeñas y medianas empresas). Desde hace tiempo se dio cuenta de que muchas consultas, errores, configuraciones de usuarios, desinfecciones con antivirus, etc., podía resolverlas utilizando la conexión remota, pues aunque algunas empresas están a pocos kilómetros, así gestiona mejor su tiempo, ahorra costes, da un servicio más rápido y puede llevar una cartera mayor de empresas.

Recientemente se le han agregado una serie de empresas que tienen necesidades especiales:

- una asesoría-gestoría cuyas comunicaciones deben ser confidenciales;
- una academia que necesita visualizar una presentación en todos los equipos de la red;
- una empresa de transporte donde los transportistas deben actualizar los datos desde sus PDA en una base de datos en la red; y
- otros clientes sueltos que tienen las empresas en sitios poco accesibles en cuanto a comunicaciones, como en la huerta y ninguna compañía ISP les asegura más de 65 Kbits/s de conexión a internet.

### estudio del caso

Analiza cada punto de la Unidad de Trabajo, con el objetivo de contestar las preguntas de este caso práctico.

1. ¿Cómo puede asegurar mucha más confidencialidad en la gestión, copia y transmisión de archivos en la asesoría-gestoría?
2. Para la academia de estudios y formación, que podría tener teleformación en un futuro, pero en la que ahora los profesores necesitan, en determinados momentos, que todos los alumnos vean en sus ordenadores lo que hace el profesor en el suyo. ¿Qué programa le interesa instalar a Dulce para este caso?
3. ¿Qué software le conviene instalar para centralizar la base de datos de los transportistas y cubrir las necesidades de esta empresa?
4. ¿Cuál es la tecnología que es preferible para la administración remota de los clientes alejados, dispersos y con conexiones a internet con muy poco ancho de banda?
5. Existen muchos programas de acceso remoto con distintos tamaños, servicios, compatibilidades... ¿Qué programa, o programas, sería interesante que llevara Dulce en su pendrive para acceder remotamente, desde cualquier lugar, al mayor número de clientes?

# 1. Terminales en modo texto

## 1.1. Telnet

### saber más

#### Opciones telnet:

**?:** mostrar ayuda.

**close:** cerrar sesión Telnet.

**display:** mostrar la configuración de la conexión en pantalla.

**logout:** cerrar la sesión.

**mode:** cambia entre los modos de transferencia ASCII y los BINARIOS.

**open:** abre otra conexión de la actual.

**quit:** sale de la aplicación Telnet.

**set:** cambia la configuración de conexión.

**unset:** carga la configuración de conexión predeterminada.

### saber más

El protocolo Telnet está especificado en el RFC 854 y sus parámetros de configuración van del 855 al 861.

### vocabulario

**Sniffer:** Programa de captura de las tramas de red. Aunque viola la intimidad y no es legal usarlo para espionar, permite detectar problemas de comunicación, virus o troyanos.

Existen sniffer genéricos, específicos para Messenger, etc.

### recuerda

En modo texto, cuando hablamos de Windows nos referimos a toda la familia de Windows 9x, 2000, XP, Vista, Windows 7, etc.

Cuando hablamos de Linux nos referimos a Unix, Minix, Aix, Mac OS, etc.

Telnet es un protocolo de red (*Telecommunication Network*) que sirve para conectarse a una máquina de una red desde otra para manejárla remotamente, como si estuviésemos sentados delante de ella.

Es un protocolo cliente-servidor que se comunica por el puerto 23. Antiguamente este tipo de acceso tenía mucho sentido, cuando los terminales o clientes eran máquinas muy lentas y los servidores máquinas muy potentes, por ejemplo, un astrónomo podía hacer cálculos en servidores de la NASA.

Después, el sentido de esta herramienta se basó en poder administrar un ordenador remoto, configurarlo y solucionar errores.

También se usa para acceder a BBS, aunque se han sustituido en gran parte por foros, news, etc.

Cuando hacemos un telnet a una máquina (con su nombre de dominio o IP), la máquina remota nos pide el nombre de usuario y la contraseña con la que conectarnos (algunos programas presumen que lo hacemos con el mismo usuario y contraseña de la sesión que tengamos abierta en nuestro ordenador cliente o local). El nombre de usuario, cuando son servidores públicos, suele ser: guest, visitor, new-user, etc., y la contraseña es pulsar la tecla ENTER.

Este protocolo proporciona reglas básicas que permiten vincular a un cliente (sistema compuesto de una pantalla y un teclado) con un intérprete de comandos (del lado del servidor: aplicaciones, procesador, disco duro, etc.). Al acceder, se abre un terminal gráfico en modo texto (que suele ser un Terminal Unix estándar: vt100 «virtual Terminal»).

La sintaxis es:

Linux:	telnet IPNombre [puerto]
Windows:	telnet //IPNombre

### EJEMPLO

```
telnet 127.0.0.1 telnet //127.0.0.1 telnet locis.loc.gov
```

El telnet solo se usa en redes locales (de hecho la mayoría de sistemas operativos ya no lo permiten ni en LAN, como por ejemplo Windows), esto se debe a que las comunicaciones no están cifradas, es decir, no son seguras y se pueden descifrar con cualquier programa de sniffer. Además, permite conexión como superadministrador o root.

En Windows 9x está activado el cliente por defecto. En Windows XP y 2000 debes activarlo desde INICIO > PANEL DE CONTROL > HERRAMIENTAS ADMINISTRATIVAS > SERVICIOS, y si lo tienes instalado, en la lista debes activarlo con el botón secundario del ratón. En Windows Vista desde INICIO > PANEL DE CONTROL > PROGRAMAS, en la opción ACTIVAR o DESACTIVAR las características de Windows y buscar CLIENTE TELNET, se activa. Pruébalo desde un navegador con telnet://127.0.0.1.

## 1.2. SSH (*Secure Shell*)

El SSH o Intérprete de Comandos Seguro es el protocolo que viene a resolver los problemas de seguridad de telnet.

La gran diferencia con telnet es que sí cifra las conexiones.

Normalmente, el protocolo SSH cifra con claves de los algoritmos RSA, pero puede hacerlo con claves y algoritmos DSA (*Digital Signature Algorithm*).

Las **características** básicas del protocolo SSH son:

- Autenticación: se autentifica el usuario mediante nombres de usuario y contraseñas (de usuario o host, públicas y privadas).
- Confidencialidad: se cifran las conexiones.
- Integridad: si por el camino el paquete es alterado, se puede detectar.

Otras **ventajas** de este sistema son:

- No rechazo: si se contesta a esta comunicación no podemos negar quiénes somos.
- Evita programas sniffer.
- Evita la suplantación de host o *Man in The Middle*.
- Existen versiones para todos los sistemas operativos.
- Permite tunelización para FTP, SMTP, Messenger, etc.
- Permite la compresión de los paquetes.

El **funcionamiento** del SSH:

- Empieza cuando el cliente abre una conexión TCP en el puerto 22.
- El servidor y el cliente negocian la versión de SSH, el tipo de cifrado (RSA/DSA), etc.
- El servidor envía su clave pública al cliente.
- El cliente la compara con la lista de claves que tiene. Si es la primera vez, es el usuario el que indica si es válida o no. Esta fase es crítica, pues es el único momento en el que se puede suplantar a este equipo. Para evitarlo tenemos acceso a las claves en mano o desde intranet.
- El cliente genera una clave de sesión aleatoria, que es enviada al servidor dentro de un paquete cifrado con el algoritmo seleccionado y la clave pública.
- A partir de este momento, la comunicación se basa en el algoritmo simétrico de encriptación seleccionado.

## ACTIVIDADES

1. Prueba la orden telnet en Windows. ¿Qué ocurre? ¿Qué te recomienda usar?
2. Las claves para ADSL, WiFi, etc. suelen ser de 128 bits de longitud y se pueden «reventar» en pocos minutos. Las claves de 512 bits se pueden «romper» con cientos de ordenadores en pocas semanas:
  - a) Entra en <http://www.kriptopolis.org> y comprueba qué longitud de claves aconsejan para el cifrado.
  - b) ¿Por qué crees que en EE. UU. y Francia solo se permiten cifrados de hasta 512 bits?
  - c) ¿Cuánto miden las claves de los decodificadores de Canal Plus?

### saber más

Todo lo que se necesita saber de seguridad, criptografía, pictocriptografía, etc., lo puedes encontrar en: <http://www.kriptopolis.org>

### caso práctico inicial

Las características del SSH cumplen con las necesidades de confidencialidad que buscamos.

### vocabulario

**Tunelización:** Acción de encapsular un protocolo por otro. Se usa para saltarse los cortafuegos o encriptar protocolos inseguros.

**Spoofing:** Suplantación de personalidad (de usuario y/o host en este caso).

### saber más

La versión 1 de SSH se creó en 1995 pero ya no se utiliza porque emplea mucho procesador y solo admite codificación RSA.

El protocolo SSH-2 está especificado en el RFC 4651 y sus parámetros van del 4652 al 4656. Para saber más sobre encriptación de clave pública te aconsejamos que leas el RFC 4716 y para los tipos de encriptación el 4344.

### 1.3. Clientes SSH

#### saber más

Existe un proyecto de código abierto de programas SSH en:  
<http://www.openssh.org>

#### saber más

La sintaxis completa del SSH es:

```
ssh
[ -1246AaCfgKkMNnqsTtVvXxYy]
[ -b bind_address]
[ -c cipher_spec]
[ -D [ bind_address:] port]
[ -e escape_char]
[ -F configfile]
[ -i identity_file]
[ -L
  ind_address:] port:host:
host_port] [ -l login_name]
[ -m mac_spec]
[ -O ctl_cmd] [ -o option]
[ -p port]
[ -R
  [ bind_address:] port:host:
hostport] [ -S ctl_path]
[ -w
  local_tun[ :remote_tun]
[ user@] hostname [ command]
```

#### recuerda

El símbolo virgulilla (~) puede escribirse en la mayoría de sistemas operativos con **ALT+126** (pulsando estos dígitos en el teclado numérico).

En Windows y portátiles suele (si está configurado en español de España) aparecer pulsando **ALT GR+4**.

En Linux podemos instalar el cliente y servidor SSH con el paquete ssh:

```
#apt-get install ssh      o      #apt-get install openssh-client
```

Los tipos de clientes SSH permiten:

- SSH: encapsular otros protocolos, tunelizar, etc.
- SFTP: la transferencia de archivos seguros.
- SSMTMP: el envío seguro de correo electrónico.
- Genéricos: navegadores que permiten los protocolos HTTPS, SFTP, SSMTMP, etc.

### 1.4. Modo texto

Para conectar a un servidor SSH debemos ejecutar:

```
ssh [-p puerto] [usuario@]ip_del_servidor
```

Como mostramos a continuación:

#### EJEMPLO

```
ssh -p 22 admin@127.0.0.1           ssh root@127.0.0.1
```

Los parámetros básicos de **configuración** del cliente se encuentran en `~/.ssh/config` o `/etc/ssh/ssh_config` y son (primero especificamos los valores por defecto):

- **Host \*|127.0.0.1|nombre-dominio**: restringe a qué servidores podemos conectarnos, el comodín asterisco (\*) permite conectarnos a todos. Se puede especificar `* .org; 192.168.0.*` (toda esa subred) o `10.0.0.[ 1-9]` (solo de ese equipo, terminado en 1 hasta el 9, ambos inclusive).
- **Port 22|otro**: puerto que podemos cambiar.
- **Protocol 2|1**: versión del protocolo SSH.
- **PubkeyAuthentication yes|no**: autenticación por clave pública.
- **PasswordAuthentication yes|no**: autenticación por contraseña.
- **IdentifyFile ruta/archivo `~/.ssh/id_rsa` y `~/.ssh/id_dsa`**: por defecto, identifica al usuario con clave RSA o DSA.
- **ForwardX11 no|yes**: activar el sistema gráfico XWindows versión 11.
- **Compression no|yes**: activa o desactiva la compresión.
- **HostbasedAuthentication no|yes**: autenticación adicional por host (rhost).
- **RhostsRSAAuthentication no|yes**: autenticación adicional por host en la versión RSA.
- **RSAAuthentication yes|no**: activar la autenticación con el algoritmo de encriptación asimétrica RSA.

Los parámetros de control y autenticación anteriores se pueden configurar desde el terminal con la orden ssh:

```
ssh [-o orden parametro] [user@]nombre-host
```

Como mostramos a continuación:

### EJEMPLO

```
#ssh -o Compression yes root@127.0.0.1
```

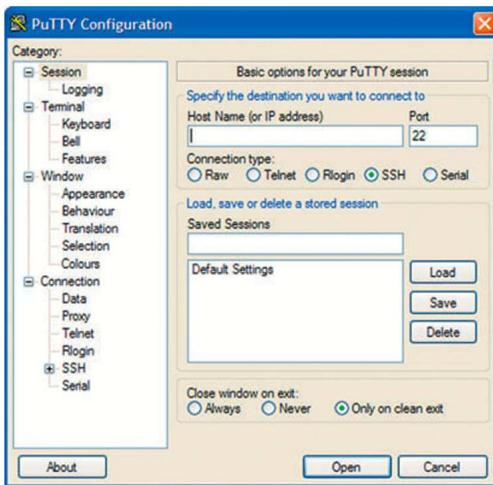
## 1.5. Modo gráfico

Para conectarnos a servidores SSH, aun en modo texto, debemos instalar, en los sistemas Windows, programas adicionales como: PuTTY (el más usado), FileZilla (para SFTP), OpenSSH, etc., aunque ya existen programas que lo soportan de forma opcional e integrada como GAIM, Google Messenger, Internet Explorer, Mozilla, etc., todos ellos intuitivos en su instalación (**SIGUIENTE, ACEPTAR CONDICIONES Y FINALIZAR** o **NEXT, ACCEPT, FINISH** o incluso se ejecutan directamente).

PuTTY permite realizar **TELNET**, **RLOGIN**, pero por defecto está configurado el SSH por el puerto 22, con el algoritmo IDEA y la versión 2.

### caso práctico inicial

PuTTY es una herramienta que sirve para la administración remota, en su versión portable, para llevarla en el pendrive USB.



Las configuraciones básicas que debemos hacer son:

- **HOST NAME:** que puede ser el nombre de dominio o la IP del servidor (ver imagen superior).
- **X11:** desplegando la CATEGORÍA SSH, aparece la OPCIÓN X11; en el lateral, en X11 FORWARDING podemos activar la casilla ENABLE X11 FORWARDING (activar el modo gráfico), tal y como vemos en la imagen de la izquierda.

## 1.6. Servidores SSH

### Instalación

#### saber más

Puedes bajarte los servidores SSH de:

- <http://www.OpenSSH.org>
- <http://freesshd.com>
- <http://sourceforge.net/projects/sshwindows/files/archivo%20setupssh381-20040709.zip>

En Windows debemos instalar OpenSSH, FreeSSHd, WinSSHd, etc. Al hacerlo se crean las claves y los archivos de configuración en C:\Archivos de programa\OpenSSH\etc.

```
#apt-get install ssh      o      #apt-get install openssh-server
```



### Arranque y parada

Para arrancar el servidor en Linux ejecutamos:

```
#/etc/init.d/ssh restart
```

Y para pararlo:

```
#/etc/init.d/ssh stop
```

Para iniciar en Windows, desde INICIO > PANEL DE CONTROL > HERRAMIENTAS ADMINISTRATIVAS > SERVICIOS, busca OPENSSH en la lista y con el botón secundario del ratón selecciona INICIAR. Para pararlo igual, pero pulsando DETENER.



Desde línea de comando:

```
net start opensshd
```

## Ficheros y parámetros de configuración

El fichero de configuración del servidor es `sshd_config`, y se encuentra en la carpeta `etc/ssh/` de Linux y en Windows en:

`C:\Archivos de programa\Open SSH\etc.`

Sus **parámetros de configuración básicos** son (omitimos los que coinciden con los del cliente, ver la referencia al caso práctico):

- `PermitRootLogin yes|no`: permite conectarse a la máquina remota como superusuario (root).
- `ListenAddress 0.0.0.0`: direcciones IP que escucha. `0.0.0.0` es todo internet con IPv4; `::` es todo internet con IPv6.
- `HostKey /etc/ssh/ssh_host_rsa_key`: lugar donde guarda las claves RSA o DSA (`ssh_host_dsa_key`).
- `KeyRegenerationInterval 1h`: si queremos forzar la regeneración de la contraseña, ponemos los segundos, minutos, horas, etc.
- `ServerKeyBits 768`: longitud en bits de la clave, 512 es insegura y 1024 eleva las necesidades de computación.
- `AuthorizedKeysFile .ssh/authorized_keys`: lista de claves de usuarios autorizados.
- `X11Forwarding no|yes`: activar el sistema gráfico XWindows versión 11.

En modo gráfico de Linux podemos usar Webmin:

- Autenticación (*Authentication*): permiso para conectarse como root, si solo deja conectarse a máquinas reconocidas, etc.
- En Red (*Networking*): direcciones IP que atenderá, si se activa el reenvío, etc.
- Control de Acceso (*Access Control*): usuarios permitidos, etc.
- Opciones Varias (*Miscellaneous Options*): activación de las X11, intervalo de regeneración de clave, etc.
- Opciones de máquina-cliente (*Client Host Options*): configuración particular para los usuarios (compresión, etc.).



Después de configurar, debemos aplicar cambios (**APPLY CHANGES**) y rearrancar el servidor.

Desde Webmin podemos arrancar y parar el servidor con **START SERVER** y **STOP SERVER**.

### caso práctico inicial

Los parámetros de configuración del servidor SSH son críticos para nuestros clientes de la asesoría.

### caso práctico inicial

Archivo `sshd_config` por defecto:

```
Protocol 2
PermitRootLogin yes
RSAAuthentication no
PasswordAuthentication yes
#Port 22
#ListenAddress 0.0.0.0
#ListenAddress ::

#HostKey
/etc/ssh/ssh_host_rsa_key
#HostKey
/etc/ssh/ssh_host_dsa_key
#KeyRegenerationInterval
1h
#ServerKeyBits 768
#PubkeyAuthentication yes
#AuthorizedKeysFile .ssh/
authorized_keys
#RhostsRSAAuthentication no
#HostbasedAuthentication no
#X11Forwarding no
#Compression yes
StrictModes yes
UsePrivilegeSeparation no
MaxStartups 10:30:60
Banner /etc/banner.txt
Subsystem sftp
/usr/sbin/sftp-server
```

## vocabulario

**Demonio:** Proceso que se ejecuta de forma residente y perpetua (requiere ser matado).



## Autenticación de usuarios

Las claves de usuario están en los archivos: `ssh_host_rsa_key`, `ssh_host_rsa_key.pub`, `ssh_host_dsa_key` y `ssh_host_rsa_key.pub`.

En el servidor, el archivo `ssh/authORIZED_KEYS` tiene las claves de los usuarios permitidos, que siempre serán un subconjunto de los usuarios locales del servidor.

En Windows debemos crear los usuarios, para ello desde la carpeta `\bin` utilizamos la orden `mkpasswd`:

```
mkpasswd -l [-u usuario] >> ..\etc\passwd
```

Como en:

### EJEMPLO

```
mkpasswd -l -u joaquin >> ..\etc\passwd
```

## Monitorización y logs

Según el Reglamento de la LOPD (Ley Orgánica de Protección de Datos), debemos tener estos registros almacenados 2 años en caso de aplicar medidas de seguridad de nivel alto a nuestros ficheros.

En el caso de Linux estos se crean y se actualizan automáticamente; en Windows conviene (por si la versión instalada no lo hace) crear los archivos vacíos, para que empiece a actualizarlos.

Dentro de la carpeta `\var`, crear el archivo de texto ASCII (con el bloc de notas, por ejemplo) `lastlog`.

## Agentes de autenticación

El SSH es tan seguro que no deja tiempo para escribir la contraseña y hacer algo antes de que vuelva a pedirla. En el caso de autenticación por clave pública, usamos un demonio que guarda las contraseñas privadas y las reenvía para poder trabajar cómodamente, es decir, solo debemos introducir la clave una vez. El agente debe inicializarse añadiendo al archivo de configuración `ssh_config` la siguiente línea:

```
ForwardAgent yes
```

Para lanzar el demonio `ssh-agent`:

```
$ eval `ssh-agent`
```

Pero debemos guardar las claves para que el agente las transmita:

```
$ ssh-add .ssh/id_rsa
```

## Túneles SSH

Los túneles encriptan otros protocolos (FTP, SMTP, etc.) y los convierten en seguros o los centralizan para pasarlo por el cortafuegos, etc. Desde línea de comandos podemos crearlo con:

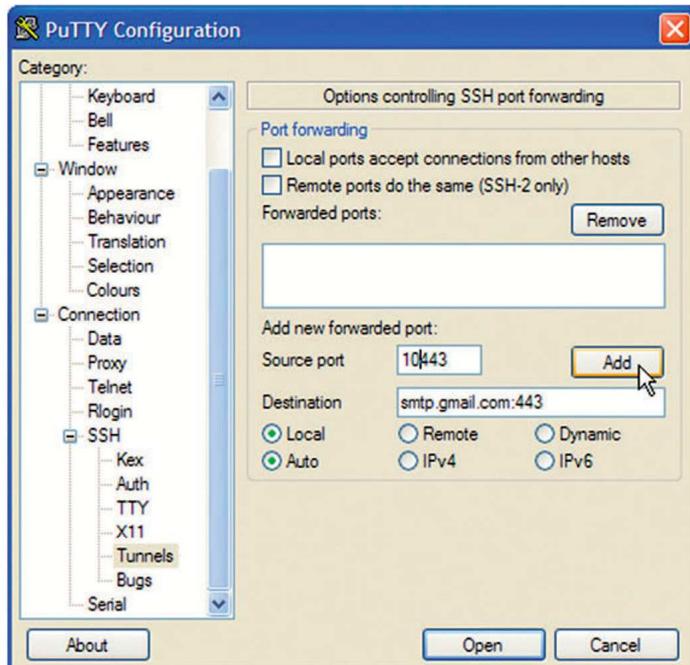
```
ssh -L puertolocal:servidor:puertoservidor [usuario@]servidor
```

Tal y como vemos en:

### EJEMPLO

```
ssh -L 10443:smtp.gmail.com:443 usuario@smtp.gmail.com
```

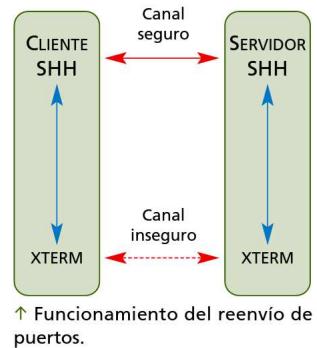
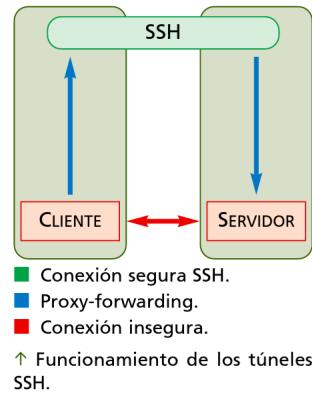
En PuTTY se configuran en la CATEGORÍA SSH, TUNNELS, donde debemos poner en SOURCE PORT el puerto local (siempre superior al 10000), y en DESTINATION el servidor: puerto. Pulsar el botón añadir (ADD).



## Reenvío X11

El reenvío de puertos (FORWARDING) es una característica que permite reenviar puertos para mandarlos por el puerto seguro SSL.

En el caso de las X11, hablamos del sistema gráfico (XTERM), terminal que envía por defecto las comunicaciones sin codificar y por ello se requiere reenviar la información por otro puerto distinto y seguro, el SSH. Para que esto funcione debemos tener configurada la negociación de este reenvío en el archivo `sshd_config` del servidor con la línea `X11FORWARDING` a YES; en el archivo `ssh_config` en el cliente la línea `FORWARDX11` debe ir también a YES. Si usamos PuTTY y/o Webmin también debemos activar esta opción, tanto en clientes como en servidores.



### Transferencia segura de archivos usando SSH (sftp y scp)

Cuando queremos transferir archivos de forma segura (cosa más que recomendable para evitar el envío de virus a servidores, por privacidad, etc.) utilizamos la orden sftp (existe en todos los sistemas operativos al instalar el cliente SSH). Su sintaxis es:

```
sftp [usuario@]servidor
```

Para copiar archivos podemos usar scp, cuya sintaxis es:

```
scp origen destino
```

Donde ORIGEN y DESTINO son el path de los archivos en el local o el servidor, por ejemplo en el local /etc/imagen1.jpg y en el remoto usuario@servidor:/path/archivo. El comando scp funciona igual que cp, con su sintaxis, comodines, etc.

Ejemplo que copia del local al servidor:

#### EJEMPLO

```
scp /etc/*.jpg joaquin@gmail.com:/images/
```

O al revés, si queremos copiar del servidor al cliente:

#### EJEMPLO

```
scp joaquin@gmail.com:/images/*.jpg /etc/
```

Para saber más sobre el comando sftp utiliza la ayuda en Linux:

```
$man sftp
```

y para el comando scp:

```
$man scp
```

Para servidores sftp, puedes consultar en internet el siguiente enlace:

<http://www.openbsd.org/cgi-bin/man.cgi?query=sftp-server>

Para sftp podemos usar las órdenes que ya conocemos y que ampliaremos en el siguiente tema: ls, chmod, cp, exit, get, put, help, ll, pwd, lpwd, mkdir, lmkdir, quit, rename, rm, rmdir, ?, etc.

## ACTIVIDADES

3. Usa ssh en modo texto para conectarte a:

locis.loc.gov

step.jbu.edu:7000

towel.blinkenlights.nl

4. Utiliza PuTTY en tu ordenador y un servidor SSH en el ordenador de un compañero (y luego al revés), para conectarlos.

5. Copia archivos de tu ordenador al de un compañero que tenga un servidor SSH, usa sftp y scp.

## 2. Terminales en modo gráfico: escritorio remoto

Existen diversos programas que nos permiten manejar un equipo desde otro, de forma remota. Unos envían como fichero de imagen (generalmente en formato jpg) lo que sucede en la pantalla del servidor y otros solo envían las coordenadas X e Y del cursor y simulan el sistema operativo anfitrión. Hay algunos muy versátiles y otros específicos (como los que se utilizan en teleformación, etc.).

### 2.1. Protocolo RDP (*Remote Desktop Protocol*)

Microsoft® Remote Desktop Protocol es un protocolo de escritorio remoto que permite la administración remota de los servidores Windows desde cualquier sistema operativo cliente. Es de los más utilizados ya que está integrado en varias de las versiones de XP, Vista y Windows 7. Se comunica por el puerto TCP 3389.

#### caso práctico inicial

Aunque vaya más lento por el ancho de banda tan bajo, para los clientes dispersos que tengan Windows, nos interesa este protocolo.

### 2.2. Clientes de escritorio remoto

Los clientes de escritorio remoto son programas que se conectan a un servidor para administrarlo, ejecutar programas en máquinas más potentes, etc. Se suelen utilizar para la administración remota y así evitar desplazamientos, para acceder a servidores que no tienen pantalla, para teleformación, para solucionar problemas de configuración, etc.

### 2.3. Instalación y conexiones

En algunas versiones de Windows, el cliente de escritorio remoto requiere el CD de instalación y añadir el complemento de Windows Escritorio remoto.

Para usar este servicio, primero debemos configurar el servidor desde INICIO > PANEL DE CONTROL > SISTEMA, en la solapa ACCESO REMOTO activamos PERMITIR ENVÍO DE INVITACIONES DE ASISTENCIA REMOTA DESDE ESTE EQUIPO. Luego, en CONFIGURACIÓN AVANZADA, activaremos PERMITIR QUE ESTE EQUIPO ESTÉ CONTROLADO REMOTAMENTE (ver imagen en el lateral).

Después, en el cliente, debemos ir a INICIO > TODOS LOS PROGRAMAS > ACCESORIOS > COMUNICACIONES > CONEXIÓN A ESCRITORIO REMOTO.



## recuerda

Es fácil confundirse entre el Inicio del servidor y del cliente, pues los verás al mismo tiempo en la pantalla al ejecutar el RDP.

## caso práctico inicial

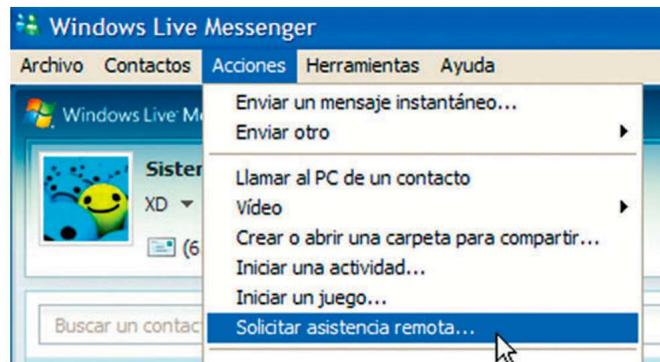
En muchos equipos puede que no hayamos podido configurar el servidor y la persona que esté delante no se aclare con nuestras indicaciones, es el momento de aprovechar MS Windows Live Messenger.



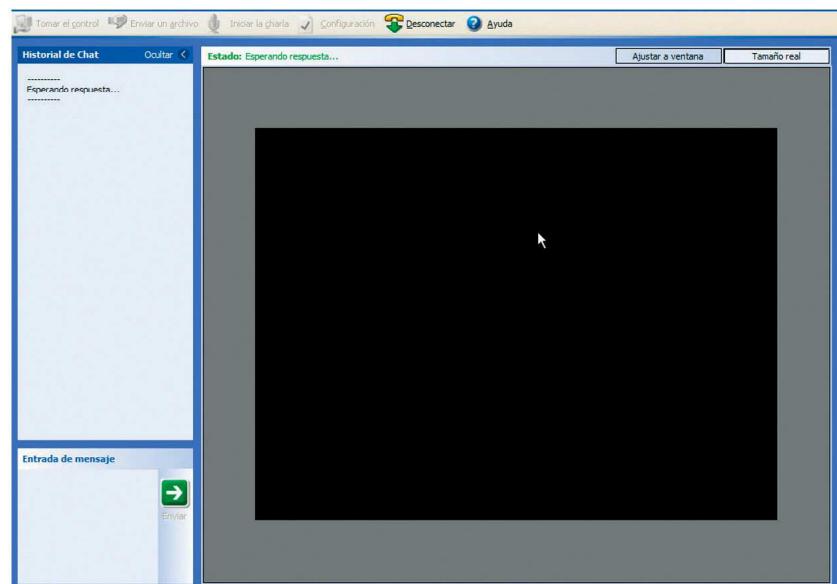
El botón **Opciones** permite configurar los colores de 8, 16, 24 y 32 bits; las últimas versiones usan cifrados de 128 bits con el algoritmo RC4 (medianamente vulnerable); permite redireccionamiento de audio; envía la imagen del remoto en un formato propio que permite compresión a partir de Vista y XP SPIII que permite HTTPS (con Windows 2008 Server), etc.

Lo importante es poner el nombre del equipo remoto o su dirección IP y pulsar **CONECTAR**. En este momento, el servidor recibe la petición y debe aceptarla. Si acepta, en el cliente aparecerá la pantalla del servidor y podremos manejarlo (nuestro usuario debe existir en el servidor).

Si no hemos configurado el servidor, lo ideal es usar Windows Live Messenger, que tiene integrada la asistencia remota (si el usuario tiene cuenta hotmail.com, hotmail.es, msn.com, msn.es, live.com, o cualquiera de Microsoft). Para ello abrimos Messenger, nos identificamos y, en el menú **ACCIONES**, pulsamos **SOLICITAR ASISTENCIA REMOTA**, y seleccionamos al usuario (al que debemos tener agregado en la agenda).



El usuario remoto debe autorizarlo. Le aparece: **NOMBREUSUARIO TE ESTÁ INVITANDO A QUE TE CONECTES A SU EQUIPO MEDIANTE ASISTENCIA REMOTA. ¿DESEAS ACEPTAR (ALT+W) O RECHAZAR (ALT+X)?** (tal y como puedes ver en la imagen de la izquierda). A partir de ahí aparece la pantalla del usuario:



## saber más

Puedes descargar un cliente gratuito para Windows y Mac en:

<http://www.teamviewer.com/es>

Y LogMeIn Free para Windows en:

<http://www.logmein.com>

Prueba en Linux el comando:

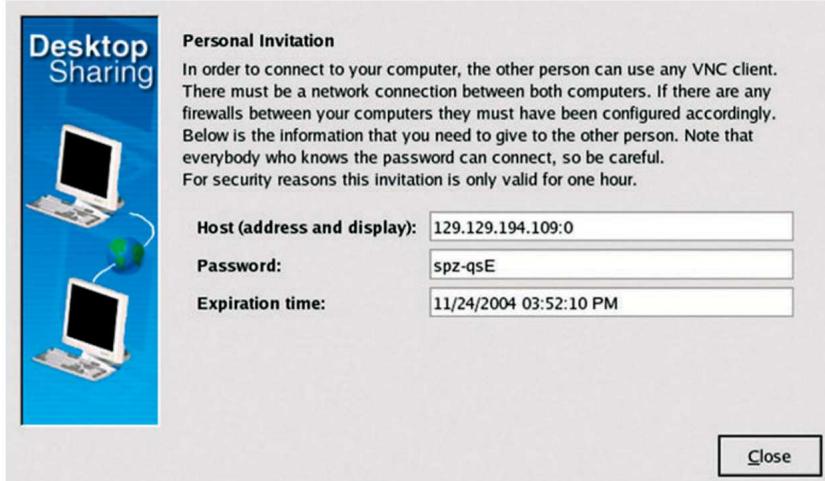
\$ krfb

Cuando se termina de conectar (suele tardar) aparece en el escritorio remoto, la posibilidad de chatear con el otro usuario o la de empezar la administración remota pulsando **TOMAR EL CONTROL**.

Existen clientes para Mac OS: por ejemplo, iRAPP de pago, o Remote Desktop Connection Client for Mac y TeamViewer, gratuitos.

En Remote Desktop Connection Client 2 para Mac OS, desde Archivo (**FILE**), Nueva Conexión (**NEW CONNECTION**), al introducir la IP del servidor podremos configurar la resolución de pantalla (**DISPLAY**), el sonido (**SOUND**), las impresoras remotas (**PRINTERS**), otras aplicaciones Windows (**APPLICATIONS**) y la seguridad por conexión cifrada (**SECURITY**), tal y como podemos observar en las imágenes.

Para Linux existen varios programas (Desktop Sharing, rDesktop), todos ellos muy intuitivos.



Existen versiones para iPhone o iPod touch (LogMeIn Ignition) o PDA (RDP para Windows Pocket).

Recomendamos utilizar los NX para Linux porque cifran las conexiones.

Algunas páginas web ofrecen sistemas operativos online, cuyo cliente puede ser cualquier navegador.

Desde <http://download.microsoft.com/download/> podemos bajar RDCC 2 para Mac OS.

Desde <http://www.rdesktop.org> podemos bajar rDesktop para clientes Linux. Puede dar errores al conectar a servidores Windows 2008 server SPII.

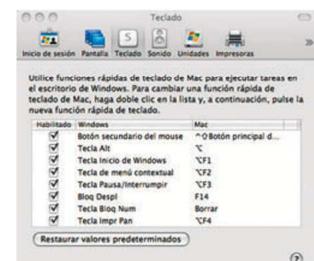
## 2.4. Servidores de escritorio remoto

Todos los sistemas operativos de Windows permiten comportarse como servidor, pero solo para un cliente. Si deseamos permitir varias conexiones a la vez (no para administración remota, sino para ejecutar aplicaciones) debemos utilizar Windows NT4, 2003 server, 2008 server o posterior.

En Linux se puede instalar un servidor en cualquier equipo (mínimo con procesador 386).



↑ Pantalla inicial de Remote Desktop Connection Client 2 (RDCC 2) para Mac OS X.



↑ Pantalla de configuración de teclado (RDCC 2).

## saber más

Sistemas operativos online:

<http://www.andrewmin.com/webx/>

<http://eyeos.org>

<http://www.mygoya.de>

<http://www.glidigital.com>

<http://www.goowy.com>

<http://www.youos.com>

<http://www.google.com>



## 2.5. Instalación y configuración básica

Para configurar el servidor de Escritorio remoto de Windows Server, desde INICIO > PANEL DE CONTROL > SISTEMA, en la solapa ACCESO REMOTO seleccionaremos PERMITIR A LOS USUARIOS CONECTARSE REMOTAMENTE A ESTE EQUIPO. Luego, en SELECCIONAR USUARIOS REMOTOS agregaremos los usuarios a los que permitiremos conectarse (ver imagen al margen).

Para las configuraciones básicas, desde INICIO > HERRAMIENTAS ADMINISTRATIVAS > CONFIGURACIÓN DEL SERVIDOR TERMINAL SERVER, pinchamos con el botón secundario del ratón y seleccionamos PROPIEDADES.

# 3. VNC (Virtual Network Computing)

## 3.1. Funcionamiento y características

El VNC o programa de Computación en Red Virtual es otro programa de administración remota de código libre. El servidor utiliza el puerto 5900, los clientes de Windows el puerto 5800, y los de Linux y Mac OS el terminal que no estén usando (por ejemplo, si solo hemos activado un terminal, el siguiente puede acceder desde el puerto 5801, si tenemos activados cuatro debemos conectarnos por el puerto 5804, etc.).

### caso práctico inicial

Para la academia que debe compartir una pantalla con varios clientes, podremos usar este software.

Las características más importantes de VNC son:

- Permite crear pantallas virtuales en Linux y Mac OS, pero solo puede compartir la pantalla actual en Windows.
- En algunas versiones puede compartir la pantalla con varios clientes a la vez.
- Permite codificación IDEA de hasta 128 bits para encriptar y RSA de hasta 2048 bits para autenticar.
- Permite compartir impresoras.
- Permite FTP seguro.
- Permite chat seguro.
- Puede compartir aplicaciones con servidores Windows.

## 3.2. Clientes VNC

Existen distintas versiones de clientes VNC (RealVNC, TightVNC, UltraVNC, etc.) que están diseñadas para todos los sistemas operativos. RealVNC es uno de estos programas, que podremos bajar de <http://www.realvnc.com> y sirve para todos los sistemas operativos.

Desde Linux podemos bajarnos el paquete vncviewer.

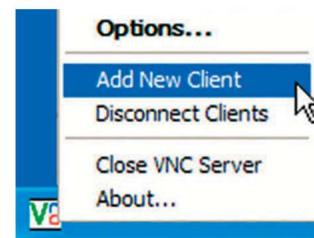
En el cliente RealVNC, nos pide el SERVER (nombre o IP), y en OPCIONES podemos elegir número de colores, pantalla completa, etc. (desde Linux podemos usar #vncviewer IP).



### 3.3. Servidores VNC

Desde Linux ejecutamos vncserver (debemos bajar el paquete vncserver):

```
# apt-get install vncserver
```



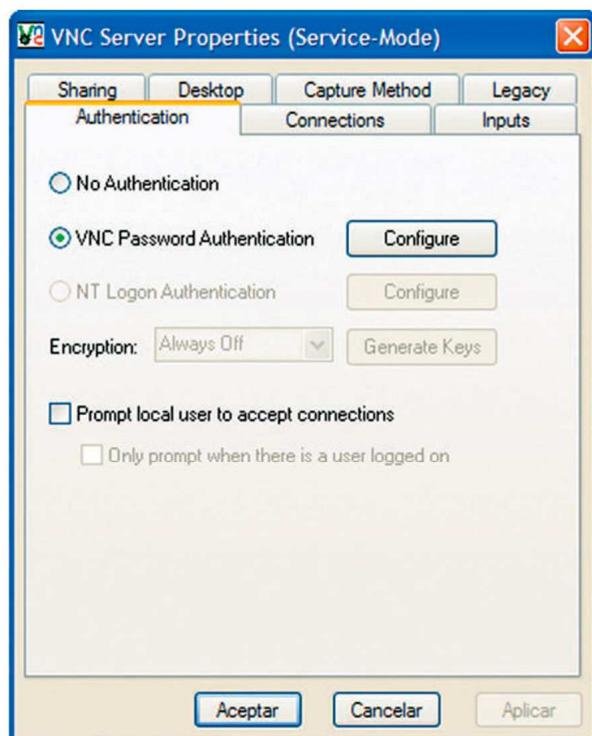
En el caso de Linux, Mac OS y Windows, RealVNC tiene una versión de servidor que se queda residente al instalar. Sobre el icono VNC (junto al de los programas residentes, junto a la hora, ver imagen al margen) podemos:

- ADD NEW CLIENT: añadir nuevos clientes.
- DISCONNECT CLIENTS: desconectar a los clientes.
- CLOSE VNC SERVER: parar el servidor.
- OPTIONS: modificar las opciones básicas (contraseñas, puertos, si se visualiza el escritorio o la opción «por defecto», etc.).

Como mínimo debemos entrar en **VNC PASSWORD AUTHENTICATION** y poner una contraseña.

#### caso práctico inicial

VNC podría ser un escritorio remoto útil para alguno de nuestros propósitos, ya que ocupa muy poco y nos sirve para cualquier sistema operativo.



Otras opciones interesantes son:

- DESKTOP: para conservar el fondo de escritorio, etc.
- CONNECTIONS: si queremos configurar los puertos de conexión.
- INPUTS: donde aceptamos entrada desde el cliente de teclados, ratón, imprimir pantallas, etc.

## 4. NX

### 4.1. Funcionamiento y características

#### caso práctico inicial

Para los clientes dispersos con poco ancho de banda que tengan Linux, nos interesa este software.

La tecnología NX es otro tipo de administración remota, solo que permite conexiones X11 muy rápidas a servidores UNIX (Unix, Aix, Linux, Mac OS, etc.) incluso con conexiones lentas como las de línea telefónica básica (desde 40 kbytes/s). Además, las conexiones las hace sobre SSH. El NX es un código abierto y funciona por el puerto 5000 (y el 22).

### 4.2. Clientes NX

#### saber más

Cliente FreeNX para todos los sistemas operativos en:

<http://www.nomachine.com>

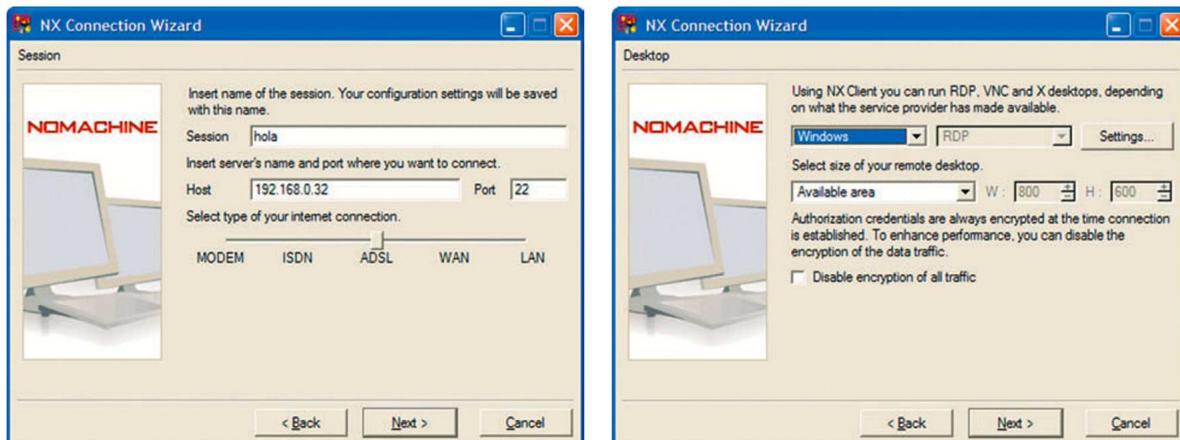
Desde Linux podemos instalar el paquete freenx (que podrá pedirnos los paquetes reenx, libxcomp, libxcompext, nxagent, nxdesktop, nxlibs, nxproxy y nxviewer).

Como la tecnología NX fue liberada por su empresa creadora, podemos encontrar muchas versiones. La más usada es FreeNX, que está implementada para todos los sistemas operativos, tanto en el modo texto, como en el entorno gráfico.

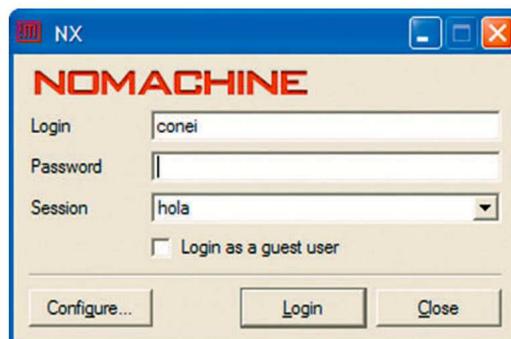
Para acceder desde el terminal:

```
$ nxclient usuario@servidor
o $ /usr/NX/bin/nxclient usuario@servidor
```

Desde el sistema gráfico creamos una sesión con la IP del servidor, el puerto y la velocidad de conexión, tamaño del terminal virtual, sistema operativo, etc.:



Otras veces nos aparece una ventana que solo nos pide el nombre de usuario (**LOGIN**), la contraseña (**PASSWORD**) y que seleccionemos una sesión (**SESSION**):



### 4.3. Servidores NX

La instalación del servidor NX solo está disponible para Unix, Linux y Solaris. Podemos instalar el paquete freenx:

```
#apt-get install freenx
```

O bajar el servidor desde: <http://www.nomachine.com>

A partir de la versión 3.6 ya está integrado en Linux KNOPPIX (es una versión de Linux de arranque con software GNU/Linux).

Solo si tienes problemas en la instalación, añade el repositorio de paquetes:

```
deb http://www.datakeylive.com/ubuntu/ gutsy main
```

Descomenta: borrar el símbolo almohadilla (#) que precede a las líneas de repositorios (empiezan por deb) en el archivo: /etc/apt/sources.list.

Ejecuta una actualización de paquetes:

```
#apt-get UPDATE
```

Ahora ya puedes instalar el paquete (y sus asociados) con:

```
#apt-get install openssh-server tcl8.4 nxlibs nxagent
nxproxy freenx nxclient
```

Modifica las siguientes líneas del archivo /etc/nxserver/node.conf:

```
#ENABLE_3_0_0_BACKEND=>>0>
#DISPLAY_BASE=1000
#AGENT_EXTRA_OPTIONS_X=>>>
```

Con las siguientes:

```
ENABLE_3_0_0_BACKEND=>>1>
DISPLAY_BASE=1001
AGENT_EXTRA_OPTIONS_X=>>-fp /usr/share/fonts/X11/misc/,/usr/
share/fonts/X11/Type1/,/usr/share/fonts/X11/75dpi
```

En todos los casos, la configuración básica necesaria pasa por renovar las claves, cosa que debemos hacer como superusuario o utilizar la orden sudo:

```
# gpg --keyserver subkeys.pgp.net --recv-keys 1135D466
# gpg --export --armor 1135D466 | sudo apt-key add
# nxsetup -install -clean
```

#### saber más

Si surgen problemas de visualización de clientes NX en Mac OS, modifica en el archivo: /etc/nxserver/node.conf del servidor, la línea AGENT\_EXTRA\_OPTIONS\_X=>>-fp/usr/share/fonts/X11/misc/,/usr/share/fonts/X11/Type1/,/usr/share/fonts/X11/96dpi

Es necesario añadir **usuarios** al servidor con:

```
# nxserver -adduser <usuario>
```

Después se le asigna una **contraseña** al usuario con:

```
# nxserv -passwd <usuario>
```

Para «lanzar» el servidor NX la primera vez, (que además «lanza» el servidor de SSH y genera las claves necesarias para la comunicación cifrada SSH) usaremos el comando nxsetup sin parámetros::

```
# nxsetup
```

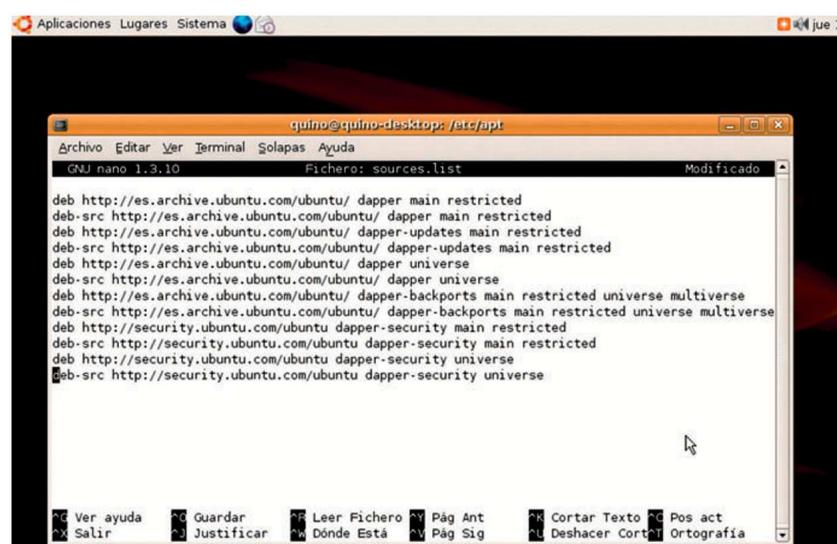
Para **arrancar** el servidor usaremos:

```
# nxserver -start
```

Y para **pararlo**:

```
# nxserver -stop
```

Debemos recordar que el servidor NX de la marca NoMachine, en modo gráfico, es gratuito solo para dos clientes a la vez. La marca 2X (ver 2x.com) tiene una versión gratuita (solo requiere registro) sin límite de clientes (se anuncia como «escritorios virtuales»). La versión que aconsejamos es la de FreeNx, gratuita, sin límite de usuarios y de código abierto.



→ Archivo de configuración de la imagen de repositorios /etc/apt/sources.list

## 4.4. NX sobre SSH

Si el servidor NX no funciona por defecto por el puerto 22, debemos realizar los siguientes pasos:

1. Modificar el fichero `/usr/NX/etc/server.cfg`, en las líneas:

```
#SSHD_PORT = «22»
#SSHD_AUTH_PORT = «22»
```

Sustituir el puerto 22 por el puerto de nuestro servidor SSH.

2. Modificar también el fichero `/usr/NX/etc/node.cfg`, en la línea:

```
#SSHD_PORT = «22»
```

### saber más

Es más fácil usar el servidor SSH sobre el puerto 22 y evitar esta edición. Al instalar el servidor este suele comprobar el funcionamiento del SSH y si ocurre algún error puede de ser tan solo por la mala elección del puerto.



← Aspecto del entorno EyeOS.

## ACTIVIDADES

6. Busca en internet información sobre el protocolo de encriptación simétrico IDEA. Localiza proyectos para «reventarlo» en los foros. ¿Qué te parece este algoritmo?
7. Prueba la asistencia remota de Microsoft Windows Live Messenger con un compañero de clase (primero actúa tú de servidor y él de cliente, y luego al revés). Si no tenéis cuenta Live o Hotmail debéis crearos una. Prueba los siguientes servicios desde el escritorio remoto:
  - a) El chat.
  - b) La pizarra.
  - c) Un editor de texto.
8. Prueba el sistema operativo online EyeOs (<http://eyeos.org>). ¿Qué te parece?

## ACTIVIDADES FINALES

- 1. Prueba desde el navegador la orden telnet, usando la URL telnet://telnet.coin.missouri.edu con el usuario guest y sin contraseña.
- 2. Entra con telnet en modo texto en Windows y Linux, y prueba las opciones: display, open, logout y quit.

Servidor	Usuario	Contraseña
sendit.nodak.edu	bbs	
seorf.ohiou.edu	guest	
scfn.thpl.lib.fl.us	visitor	
pen1.pen.k12.va.us	guest	Guest
freenet.msp.mn.us	visitor	
fnet.cc.utoledo.edu	visitor	Visitor
128.175.63.164		

- 3. Entra en <http://www.kriptopolis.org> y busca información sobre criptografía (introducción, criptografía clásica, criptografía simétrica, etc.).
- 4. Busca información sobre el significado de Spoofing, *Man in The Middle* y sniffer.
- 5. Escribe en una hoja la configuración del archivo de configuración del servidor SSH para crear:
  - Una conexión con codificación RSA, modo gráfico X11 y permitiendo la administración remota del superusuario.
  - Una conexión con codificación DSA, sin modo gráfico y sin permitir la administración remota del superusuario.
  - Una conexión lo más segura posible.
- 6. Instala y configura SSH para el modo texto, modifica los archivos para los casos del ejercicio anterior y pruébalos con un compañero a ver si funcionan.
- 7. Configura los archivos de configuración de un cliente y un servidor SSH para que se conecten por otro puerto que no sea el 22. Comprueba que funciona.
- 8. Prueba a conectarte con PuTTY a algún servidor de los indicados en el ejercicio 2.
- 9. Configura un servidor SSH en un ordenador y que tu compañero se conecte a tu dirección IP desde PuTTY (realizadlo luego al revés).
- 10. Instala el servidor SSH de Webmin, configura el servidor para que toda la subred de la clase pueda conectarse. ¿Lo ves poco intuitivo? ¿El problema es que está en inglés?
- 11. Busca el archivo lastlog en Linux, arranca el servidor y consulta qué está escribiendo.
- 12. Según el punto de este tema «Monitorización y logs», en Windows debemos crear el archivo lastlog. Comprueba que no existe y créalo.
- 13. Lanza en Linux el agente de autenticación. Realiza una conexión SSH. ¿Lo ves útil para no estar repitiendo las claves?

- 14. Crea un túnel para conexiones FTP en modo texto (consulta en el epígrafe 1 de este tema, el apartado dedicado a los «Túneles SSH»).
- 15. Crea el túnel para conexiones FTP en PuTTY. ¿Te parece más sencillo en modo gráfico o en modo texto? ¿Crees útil conocer ambos métodos sabiendo que la mayoría de los servidores están en modo texto?
- 16. Desde el escritorio remoto de Windows, comprueba que dos compañeros tuyos de clase no tienen errores o avisos en su configuración de hardware.
- 17. Desde una conexión de escritorio remoto de Messenger, cambia la resolución de pantalla de un compañero.
- 18. Entra en <http://www.youos.com>, date de alta y comprueba las posibilidades que tiene. ¿Ves útil un sistema operativo desde cualquier ordenador, PDA o móvil? ¿Qué aplicaciones añadirías? Busca sistemas operativos online en español. ¿Tienen menos aplicaciones?
- 19. Entra en <http://www.andrewmin.com/webx>. ¿Te gusta el entorno?
- 20. Entra en Google Docs (<http://docs.google.com>, debes tener una cuenta en Google o Gmail). Crea un archivo de texto, hoja de cálculo o presentación y compártela con al menos dos compañeros. ¿Te parece interesante que todos los que invites puedan modificar el archivo? ¿Para qué lo usarías?
- 21. Configura un servidor VNC en Windows, acepta tu IP como cliente y conéctate desde otro equipo. ¿Te parece rápido?
- 22. Edita en Linux los archivos de ssh-config y sshd-config y explica con tus palabras qué opciones tiene por defecto (obvia las opciones que estén comentadas, es decir, las que empiecen por el símbolo almohadilla #).
- 23. Busca en internet los protocolos (al menos tres) que sean susceptibles de hacer túneles.
- 24. Completa en tu cuaderno el siguiente cuadro:

Clientes que se conectan a servidores Windows	Clientes que se conectan a servidores Linux	Clientes que se conectan a servidores Mac OS	Clientes que se conectan a servidores mixtos

- 25. Ahora, rodea con un círculo los programas-cliente del ejercicio anterior que permiten conexiones seguras.
- 26. Añade una cruz, en la tabla que has copiado en tu cuaderno del ejercicio 24, delante de los clientes que aceptan conexión X, xterm, X11 o sistemas gráficos.
- 27. Busca información y pantallas de uso de los programas de acceso remoto especializados para formación (por ejemplo: NetOp School).
- 28. Busca vídeos en YouTube sobre cómo instalar y configurar los servidores y clientes estudiados.

**Nota:** Escribe cosas como: INSTALAR VNC Y CONTROLAR UN PC REMOTAMENTE, PuTTY, configurar Escritorio remoto, etc. (puede que salgan vídeos en otros idiomas).

## PRÁCTICA PROFESIONAL

### material

- PC con Linux Ubuntu, última versión (o máquina virtual, mínimo Ubuntu con kernel 2.6).
- Ordenadores con Windows.
- Conexión a internet.



### Instalación y configuración de un servidor VNC en Linux y un cliente en Windows

#### Objetivo:

Instalar un servidor VNC en un servidor Linux y un cliente VNC en Windows con las técnicas que hemos estudiado en la Unidad de Trabajo.

#### Desarrollo:

Supongamos que trabajamos en una empresa que no quiere tener el monitor del servidor Linux enchufado las 24 horas del día (por ahorro de energía, costes y para evitar contaminar el medio ambiente). Como nosotros estamos casi siempre en un PC con Windows dentro de esa misma LAN, sería interesante que lo administrásemos remotamente.

1. Instala el servidor VNC en Linux. Para ello necesitas un CD de Linux o una conexión a internet.

Desde la línea de comandos instala el paquete vnc4server (o alguno posterior).

```
# apt-get install vnc4server
o
$ sudo apt-get vnc4server
```

También puedes hacerlo a partir del entorno gráfico desde el instalador de paquetes (**SISTEMA [O EL LOGO DE LINUX] > ADMINISTRACIÓN > GESTOR DE PAQUETES SYNAPTIC**).

Busca el paquete vnc4server, pulsa y selecciona **MARCAR PARA INSTALAR**, y pincha en **APLICAR**.

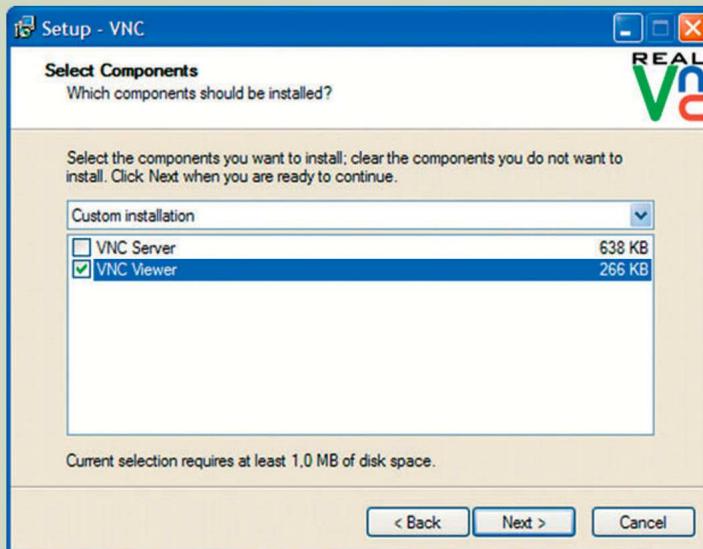
2. Luego configúralo desde el entorno gráfico:

**SISTEMA > PREFERENCIAS > ESCRITORIO REMOTO**

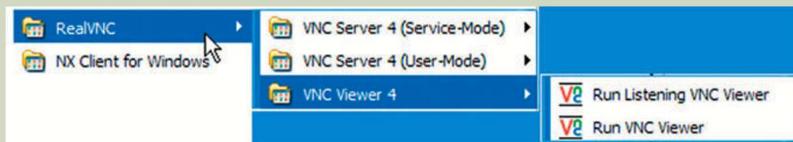
Activa **PERMITIR A OTROS USUARIOS VER MI ESCRITORIO**, y **REQUERIR QUE EL USUARIO INTRODUZCA CONTRASEÑA** (pon una difícil, pero que recuerdes). Desactiva **PEDIR CONFIRMACIÓN**.

3. Escribe la IP del servidor (usa en modo texto la orden ifconfig).
4. Ya puedes apagar la pantalla.
5. Cambia al PC Windows. Instala el cliente VNC (tienes versiones gratuitas en <http://www.realvnc.com> o TightVNC en softonic.es, sourceforge.net, utilidades-utiles.com, etc.).
6. Instala el programa.
7. Pulsa **NEXT** o **SIGUIENTE** hasta que aparezca el contrato. Pulsa en **ACCEPT**, **AGREE** o **ACEPTO**.

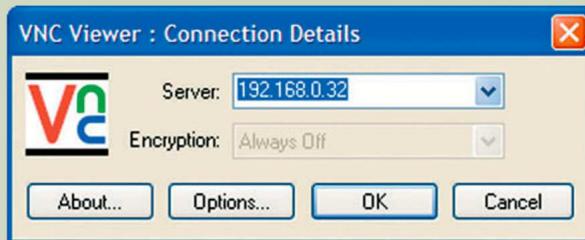
8. Sigue hasta que te permita seleccionar componentes: VNC VIEWER y VNC SERVER. Desactiva VNC SERVER. Termina la instalación.



9. Ejecuta VNC VIEWER (estará en INICIO > PROGRAMAS > REALVNC > VNC VIEWER > RUN VNC VIEWER).



10. Introduce la IP (supongamos que es 192.168.0.32, pero debes poner la que te apuntaste del servidor).



11. Pulsa OK.  
12. Espera a que se conecte. Te pedirá la contraseña que escribiste en el servidor.  
13. Ya puedes empezar a utilizar el servidor Linux desde Windows en la ventana que aparece.

## MUNDO LABORAL



### Google Chrome

↑ El sistema operativo Google Chrome es un navegador a pantalla completa sobre un kernel de Linux. Trabaja con computación en nube, es decir, es un cliente que aprovecha las aplicaciones online de grandes servidores en máquinas que pueden ser muy lentas, carecer de disco duro, etc. Este tipo de clientes son una alternativa a la administración remota para ejecutar aplicaciones.

### Introducción al SO Google Chrome

[...] Estamos anunciando un nuevo proyecto que sea una extensión natural de Google Chrome (el sistema operativo Google Chrome). Es nuestra tentativa re-pensar lo que deben ser los sistemas operativos.

El SO Chrome de Google Chrome es de código abierto, el sistema operativo ligero que será instalado inicialmente en los miniportátiles. A finales de este año publicaremos su código fuente, y los portátiles que funcionan con el SO del cromo de Google estarán disponibles para los consumidores en la segunda mitad de 2010. [...]

La velocidad, la simplicidad y la seguridad son los aspectos claves del SO Google Chrome. Estamos diseñando el SO para que sea rápido y ligero, para arrancar y para conseguir estar en internet en pocos segundos. La interfaz utilizadora es mínima [...], y la mayor parte de la experiencia del usuario ocurre en internet. Y como hicimos para el navegador Google Chrome, estamos volviendo a los fundamentos y estamos reajustando totalmente la arquitectura de seguridad subyacente del SO de modo que los usuarios no tengan que ocuparse de los virus, del malware ni de las actualizaciones de la seguridad. Debe apenas trabajar.

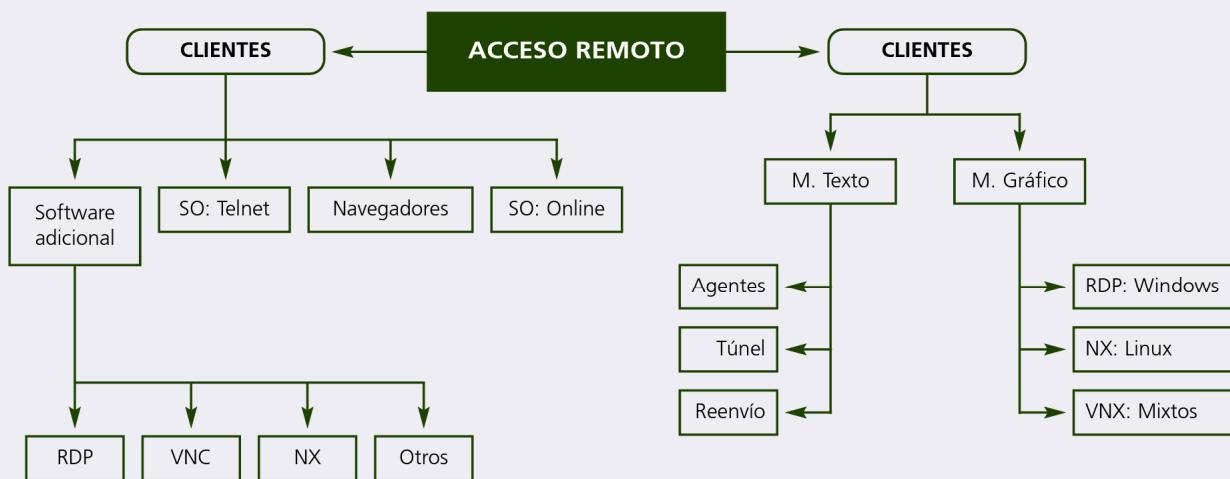
El OS del cromo de Google funcionará en ambos x86 (32 y 64 bits) [...] y estamos trabajando con los OEM múltiples para atraer un número de portátiles para poner el próximo año. La arquitectura de los programas es simple: Chrome de Google, que funciona dentro de un nuevo sistema de la visualización en una ventana encima de un núcleo de Linux. Para los reveladores de uso, la Red es la plataforma. Todos los usos en internet trabajarán automáticamente y las nuevas aplicaciones pueden ser escritas usando sus tecnologías preferidas de la Red. Y, por supuesto, estas aplicaciones funcionarán no solo en el SO Google Chrome, también en cualquier navegador estándar basado en Windows, Mac y Linux, revelando la base de usuarios más grande de cualquier plataforma. [...]

Publicado el 07/07/2009 en el blog oficial de Google:  
<http://googleblog.blogspot.com/2009/07/introducing-google-chrome-os.html>.

### Actividades

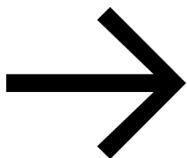
1. Los programas de administración remota se pueden usar para aprovechar los grandes servidores y aplicaciones. ¿Crees que Google Chrome y Google Docs son una alternativa a la administración remota con este fin?
2. ¿Qué te parece la idea de este tipo de sistemas operativos?
3. ¿Cuál crees que la gente usará más, Windows o Google Chrome?

## EN RESUMEN



## EVALÚA TUS CONOCIMIENTOS

1. La orden `scp` sirve para:
  - a) Transferir archivos.
  - b) Copiar archivos.
  - c) Enviar archivos por Messenger.
  - d) Hacer túneles seguros.
2. El acceso remoto se realiza:
  - a) Solo por IP.
  - b) Por IP, nombre de dominio o invitación (como en Messenger).
  - c) Solo por nombre de dominio.
  - d) Por MAC y contraseña RSA o DSA.
3. Indica la contraseña más segura:
  - a) 12312345456789325422
  - b) sd123!@#d4@#sdw23d17
  - c) saludcochedineroymas
  - d) Todas lo son, miden lo mismo.
4. Por defecto (se puede modificar), el puerto de comunicación del protocolo SSH es el:
  - a) 22.
  - b) 443.
  - c) 5800.
  - d) Las respuestas a), b) y c) son correctas.
5. NX es un cliente que permite conectarse a servidores NX de:
  - a) MS-DOS y terminal de Linux.
  - b) Windows y Linux.
  - c) Linux, Minix, FreeBSD, Unix.
  - d) Minix, Linux, FreeBSD, Unix, Aix y Mac OS.
6. Accediendo a telnet desde un navegador:
  - a) Se abre en el navegador, pues soporta telnet:// de forma nativa.
  - b) Se abre con la aplicación telnet del SO, pero en modo texto.
  - c) Se abre una aplicación telnet que debemos instalar.
  - d) No pasa nada.



**Redacción y selección de contenidos:** Joaquín Andreu

**Edición:** Montserrat Sánchez

**Diseño de cubierta:** Paso de Zebra

**Fotocomposición, maquetación**

**y realización de gráficos:** MT Color & Diseño, S. L.

**Fotografías:** Microsoft Corporation; Canonical Ltd.; Apple Inc.; Bind, licencia BSD; ICANN; PuTTY, Simon Tatham; OpenSSH, OpenBSD; Webmin; TeamViewer GMBH; RealVNC Limited; Medialogic; Google; Google Inc.; Mozilla Foundation; Filezilla-project.org; gFTP, Brian Masney ; GNU.org; OpenSight Software, LLC; cPanel Inc.; Cuerpo Nacional de Policía, Ministerio del Interior, Gobierno de España; Fábrica Nacional de Moneda y Timbre, Ministerio de Economía y Hacienda, Gobierno de España; Conselleria de Justicia i Administracions Públiques, Generalitat Valenciana; Agència Catalana de Certificació, Generalitat de Catalunya; Yahoo!; Adobe Systems Incorporated; Romain Bourdon; The Apache Software Foundation; Oracle Corporation; The PHP Group; GNU; 3Com Corporation; PLANET Technology Corporation; GSMA; Telefónica Móviles España, SAU; France Télécom; The Information Technology & Innovation Foundation; TeleAtlas; HISPAKSAT, SA; SES ASTRA, Grupo SES; NEO-SKY 2002, SA; Euskaltel, SA; Xfera Móviles, SA; EDIMAX Technology Co.; Check Point Software Technologies Ltd.; Agnitum Ltd.; Bluetooth SIG.; Wi-Fi Alliance; IEEE; D-LINK Europe Ltd.; Jinx, Inc.; Medion Iberia, SL; Sony Computer Entertainment Europe; Symantec Corporation; Bratel Co., Ltd.; Technicolor; Koninklijke Philips Electronics N.V.; Accton Technology Corporation; Skype Limited; Cisco System, Inc.; ITU; ISOC; Digium, Inc.; Peoplecall the callshop Co.; Vonage Marketing LLC; AOL Inc.; Telefonica USA Inc.; Jajah, Inc.; CounterPath Corporation; Internap Network Services Corporation; iDATE FR; Getty Images (Photos.com) y archivo Editex

**Dibujos:** Ángel Ovejero

**Dirección producción:** Teresa del Arco

**Preimpresión:** José Ciria

**Producción editorial:** Francisco Antón

**Dirección editorial:** Carlos Rodríguez

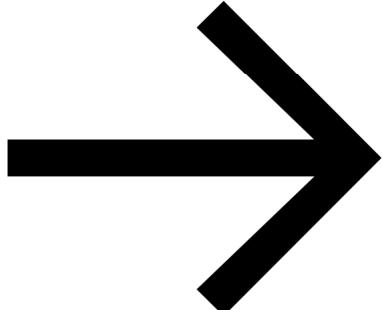
**Editorial Editex, S. A.** ha puesto todos los medios a su alcance para reconocer en citas y referencias los eventuales derechos de terceros y cumplir todos los requisitos establecidos por la Ley de Propiedad Intelectual. Por las posibles omisiones o errores, se excusa anticipadamente y está dispuesta a introducir las correcciones precisas en posteriores ediciones o reimpresiones de esta obra.



El presente material didáctico ha sido creado por iniciativa y bajo la coordinación de **Editorial Editex, S. A.**, conforme a su propio proyecto editorial.

© **Editorial Editex, S. A.**

Vía Dos Castillas, 33. C.E. Ática 7, edificio 3, planta 3<sup>a</sup>, oficina B  
28224 Pozuelo de Alarcón (Madrid)



Cualquier forma de reproducción, distribución, comunicación pública o transformación de esta obra solo puede ser realizada con la autorización de sus titulares, salvo excepción prevista por la ley. Diríjase a CEDRO (Centro Español de Derechos Reprográficos, [www.cedro.org](http://www.cedro.org)) si necesita fotocopiar o escanear algún fragmento de esta obra.