

ADMINISTRACIÓ DE SISTEMES OPERATIUS

U4: ADMINISTRACIÓ REMOTA

CFGS
ASIX

DPT INF

Administració remota

Vicent Benavent

CFGS ASIX

Mòdul: Administració de Sistemes Operatius

UD4: Administració remota i serveis d'impressió.



Tot el temari el pots trobar a Github: [ASO](#)

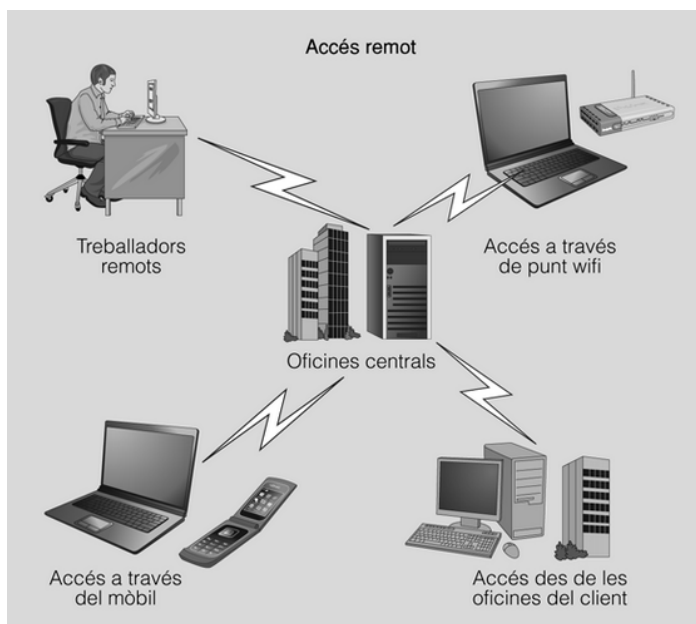
Aquest document és un resum del apunts de la IOC que podeu trobar [ací](#).

1 SERVEIS D'ACCÉS I ADMINISTRACIÓ REMOTA

La generalització de les comunicacions entre ordinadors, tant en xarxa local com mitjançant Internet, ha introduït canvis dràstics en la forma d'accedir als equips i en la seva administració, permetent que pugui fer-se de forma remota. Tant si aquest accés es basa en la línia d'ordres, en una aplicació gràfica o en l'ús del navegador com a interfície gràfica, l'accés i l'administració remota faciliten a l'administrador la tasca de configuració i gestió del sistema informàtic, tenint sempre en compte la capacitat i ample de banda de la comunicació, i la necessària atenció a la seguretat i a la privacitat.

1.1 Introducció a l'accés remot a equips

La capacitat d'accedir remotament a arxius i informació en ordinadors a través d'Internet és interessant tant des del punt de vista de l'administració de sistemes com de l'execució i explotació de qualsevol tipus d'aplicació remota. Pot ser una eina útil per recuperar un arxiu oblidat a l'ordinador o per permetre a un administrador de sistemes modificar la configuració d'un servidor. També és utilitzat per les companyies per accedir a la informació comercial i administrativa dels seus sistemes, ja sigui des de les oficines del client o des del domicili dels seus treballadors.



Aquest ús tan diversificat fa que hi hagi moltes tecnologies disponibles per permetre aquest tipus d'accés: des del sistema de fitxers compartit incorporat en la majoria de sistemes operatius fins a eines més específiques desenvolupades per empreses.

L'ús tan divers fa que també hi hagi una gran diversitat de maneres i mitjans d'accedir remotament a informació, tal com podeu veure a la figura. Per fer front a aquesta diversitat hi ha nombrosos protocols i eines que donen accés remot des d'un portàtil amb connexió Wi-Fi, un telèfon mòbil (3G o superior) o una connexió fixa d'Internet.

1.1.1 Usos més freqüents de l'accés remot a ordinadors

El programari que permet l'administració remota és cada vegada més comú i s'utilitza sovint quan és difícil o molt difícil estar físicament a prop d'un sistema per usar-lo o per tal d'accedir al material d'Internet que no està disponible en la mateixa ubicació.

Els servidors i altres equips de xarxa, per diverses raons, a vegades es distribueixen en distàncies considerables. Fins i tot quan estan relativament a prop, dins del mateix edifici o la mateixa planta, poden estar instal·lats en espais amb accés difícil o restringit. Per aquestes raons l'administració remota, és a dir l'administració d'un equip des d'un altre equip, és una necessitat quotidiana.

A continuació mostrem una llista dels usos més freqüents de l'accés remot a ordinadors:

- **Administració de sistemes:** gestionar, administrar i configurar equips i servidors de forma remota en xarxa local o mitjançant Internet.
- **Suport i assistència tècnica de forma remota (helpdesk):** solucionar problemes tècnics a domicili sense necessitat de desplaçaments físics (vegeu la [figura.2](#)).
- **Treball a distància:** accés des del domicili als recursos de la xarxa de l'oficina (arxius, escriptori, correus, impressores, etc.), cosa que possibilita el teletreball.
- **Reunions i presentacions en línia:** compartir escriptori amb assistents situats en diferents llocs de treball. Inclou prestacions addicionals com videoconferència i xats de text i de veu.
- **Aplicacions d'ensenyament:** el professor pot monitoritzar les activitats escolars, compartir el seu escriptori i donar ajuda de forma remota.
- **Supervisió d'activitats:** tasques de treballadors, supervisió parental, etc.

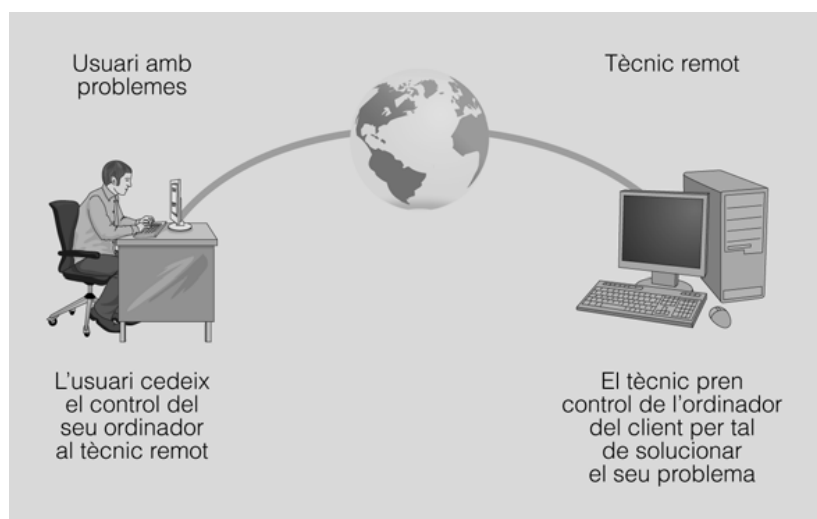


Figura Suport i assistència tècnica remota a usuaris

1.1.2 Tipus d'accés remot

Les tasques d'accés i administració remots es poden dur a terme amb mitjans diferents, però sempre estan condicionats per la xarxa que connecta l'equip que cal administrar amb l'estació on hi ha l'administrador.

La xarxa imposa condicions de:

- 1 **Capacitat:** si la connexió no té una amplada de banda suficient no serà pràctic treballar amb una interfície gràfica remota. Com més augmenti el retard a la xarxa, més frustrant serà el treball interactiu.
- 2 **Seguretat:** és obvi que la comunicació entre l'administrador i l'equip remot no ha de ser interceptada per altres usuaris de la xarxa.

Els diferents mitjans d'administració remota es poden agrupar en **tres categories** bàsiques:

- 1 Sessió de treball a la consola
- 2 Sessió de treball amb interfície gràfica
- 3 Client o eina d'administració local

Cada categoria té els seus punts forts i febles. Una sessió de **treball a la consola** mitjançant **SSH** o l'obsolet i insegur **Telnet** no exigeix grans capacitats a la xarxa. És el mètode més lleuger i fins i tot es pot utilitzar amb comoditat per administrar màquines molt distants o amb xarxes de capacitat escassa. A més, és relativament senzill automatitzar tasques mitjançant guions de programació per repetir les mateixes operacions en un conjunt de màquines.

L'administració remota amb **interfície gràfica** permet obrir a l'estació local finestres d'aplicacions que s'executen en el servidor remot. Fins i tot permet veure l'escriptori complet de l'estació remota. Tot i que pot ser un mètode de treball còmode, requereix capacitats de la xarxa que normalment només es troben disponibles dins d'una LAN.

Emprar una **eina d'administració local** feta a mida, com un assistent per configurar una impressora remota, o genèrica, com un navegador web, intenta conjugar punts forts de les dues tècniques anteriors. En aquest cas, l'administrador dialoga amb una aplicació local o una pàgina web i no necessita recordar ordres. A més, com que la representació gràfica es produeix localment, no s'exigeix gran capacitat a la xarxa. El problema de les eines específiques és que es tracta d'un programari que cal instal·lar a cada estació que l'administrador vulgui accedir. Sovint aquesta eina només està disponible per a un sistema operatiu concret o fins i tot per a una de les seves versions.

Aquest problema s'obvia amb les interfícies web, ja que un navegador web és un programari comú present en tots els equips. A la taula podeu veure un resum dels principals avantatges i inconvenients dels tres tipus d'administració remota.

Taula: Els tres principals mitjans d'administració remota

Mètode d'administració	Avantatges	Inconvenients
Treball a la línia d'ordres	Requisits mínims per a la xarxa. És flexible i es pot automatitzar.	Cal conèixer la sintaxi i recordar les ordres.
Interfície gràfica	Visual i flexible. No cal recordar ordres.	Imposa requisits de capacitat a la xarxa.

Client local	Visual i flexible. No cal recordar ordres. No consumeix gaires recursos de la xarxa.	Si no es tracta d'una interfície web (navegador), cal instal·lar programari.
---------------------	--	--

1.2 Administració remota basada en la línia d'ordres

Les connexions remotes mitjançant la línia d'ordres són l'opció que menys capacitats exigeix a la xarxa i per tant es poden fer servir fins i tot en els casos en el qual el canal de comunicació no té una gran amplada de banda disponible o quan el retard és important. Exigeixen conèixer la sintaxi pròpia de l'interpret d'ordres emprat i de les seves eines, però són un mecanisme molt flexible que permet fer moltes automatitzacions.

1.2.1 Introducció a SSH

SSH és un protocol de xarxa que permet l'intercanvi d'informació de manera segura. Utilitza xifrat i criptografia de clau pública per tal de fer l'autenticació de l'estació remota.

Una de les seves funcions més emprades és iniciar una sessió remota per tal d'executar ordres en substitució de Telnet. Però les seves capacitats són més àmplies, ja que també permet:

- Transmetre fitxers de manera segura.
- Fer còpies de seguretat de manera eficient i segura en combinació amb l'ordre rsync.
- Fer túnels per assegurar qualsevol servei que no es transmeti encriptat (HTTP, SMTP, VNC, etc.) o per travessar tallafocs que estiguin bloquejant el protocol.
- Reenviament automàtic de sessions X11 des d'un host remot (disposa d'aquesta funció openSSH però no altres implementacions d'SSH).
- Navegar pel web a través d'una connexió amb un servidor intermediari (proxy) xifrada amb programes clients que siguin compatibles amb el protocol SOCKS.
- Seguiment automatitzat i administració remota de servidors a través d'un o més dels mecanismes exposats anteriorment.
- Fent servir SSHFS (SSH File System), un sistema d'arxius basat en SSH que pot crear de manera segura un directori en un servidor remot i actuar com a sistema de fitxers en xarxa.

Les seves possibilitats es poden combinar de moltes maneres diferents. Ateses les seves característiques criptogràfiques, SSH és una eina fonamental per a l'administrador de xarxa. De les capacitats i funcions esmentades, les més utilitzades són les d'inici de sessió remota, la transferència d'arxius i la tunelització d'altres serveis mitjançant redreçament estàtic de ports o bé establint un servei SOCKS amb el redreçament dinàmic de ports.

1.2.2 Instal·lació d'SSH

SSH utilitza una arquitectura client-servidor, en què el client es connecta a una màquina remota, el servidor. En la major part de distribucions GNU/Linux el client ja hi és però, si es vol accedir a una màquina de manera remota, en aquesta màquina hi haurà d'haver el servidor SSH instal·lat.

La implementació més popular d'SSH és la desenvolupada per la fundació OpenSSH, el servidor openSSH-server. Per procedir a instal·lar-la, executem l'ordre següent des de la consola de la màquina a la qual ens connectarem (servidor).

```
# apt install openssh-server
```

Si no hi ha hagut cap error durant la instal·lació podreu veure un missatge en què es creen les claus necessàries per assegurar les comunicacions segures mitjançant l'encryptació.

```
S'està configurant openssh-server (1:5.5p1-6+squeeze1)...
```

```
Creating SSH2 RSA key; this may take some time ...
```

```
Creating SSH2 DSA key; this may take some time ...
```

```
Restarting OpenBSD Secure Shell server: sshd.
```

L'última part de la configuració bàsica es fa de manera automàtica quan intentem connectar un client per primera vegada. És la generació automàtica de la clau compartida que utilitzaran client i servidor per assegurar que les comunicacions són segures. Un cop s'ha comunicat la clau compartida entre totes dues estacions, el missatge s'encrypta de forma convencional.

1.2.2.1 Arxius de configuració del client SSH

El client d'OpenSSH es pot configurar de manera prou flexible com perquè l'administrador pugui definir una configuració general per a tot el sistema, perquè cada usuari pugui modificar els paràmetres adients per a les seves connexions o perquè pugui especificar opcions determinades per a cada connexió individual.

El client d'OpenSSH farà servir:

- Les opcions indicades a la línia d'ordres
- Els valors especificats en el fitxer de configuració de l'usuari: **\$HOME/.ssh/ssh_config**
- Els valors especificats en la configuració per a tot el sistema: **/etc/ssh/ssh_config**

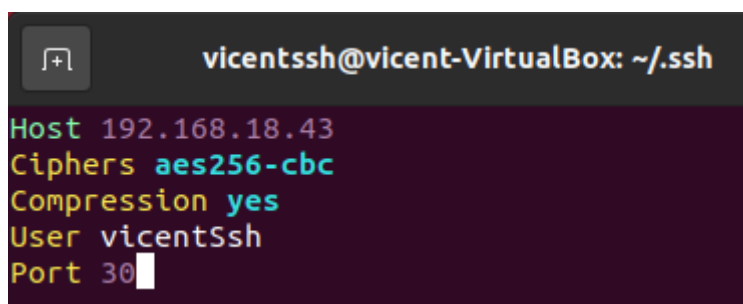
Per a cada paràmetre, el client farà servir el primer valor trobat. És a dir, si s'especifica un paràmetre a la línia d'ordres no se'n consultarà el valor en els fitxers de configuració. Dins dels fitxers de configuració és possible definir seccions per a diferents equips (mitjançant la paraula reservada *host*). Les línies buides i les que comencen amb un # (comentari) seran ignorades. A la taula podeu trobar una llista amb les opcions més bàsiques de configuració d'un client.

Taula: Algunes de les opcions més bàsiques de la configuració d'un **client**

Opció	Funció
Host <patró>	Permet especificar opcions que només s'aplicaran a les connexions amb l'amfitrió indicat. L'amfitrió s'indica mitjançant patrons amb els caràcters * i ?. Especifica el port de destinació per a la connexió, que per defecte és el 22.

CheckHostIP <yes no>	El seu valor predeterminat és <i>yes</i> . Si l'opció està activada, es comprovarà l'adreça de l'estació remota mitjançant el fitxer <i>known_hosts</i> per tal d'advertir un possible enverinament de DNS.
Cipher i Chipers	Permeten especificar respectivament l'algorisme d'encryptació per les a connexions SSH1 i la precedència d'algorismes que cal emprar en les connexions SSH2.
Compression <yes no>	Si la connexió és molt lenta, la compressió pot millorar els resultats. Si la xarxa té prou amplada de banda normalment no es recomana.
Port <port>	Especifica el port de destinació per a la connexió, que de manera predeterminada és el 22.
RekeyLimit <limit>	Especifica el volum màxim d'informació que es pot transmetre abans d'haver de renegociar la clau de sessió. Es poden fer servir els sufixos K, M o G.
User <usuari>	Especifica l'usuari per establir la connexió a l'estació remota.
SendEnv <variables>	Permet enviar el valor de les variables d'entorn especificades a l'estació remota.

A continuació podeu veure un exemple d'un arxiu **\$HOME/.ssh/ssh_config**, corresponent a la configuració d'un client SSH.



```

Host 192.168.18.43
Ciphers aes256-cbc
Compression yes
User vicentSsh
Port 30

```

En aquest cas, el fitxer indica que, quan establim una connexió amb el servidor amb la IP 192.168.18.43, s'ha d'utilitzar un algorisme d'encryptació del tipus aes256-cbc (on pot ser més menuda, ej 128, 196, però no més gran de 256 per llimitació, [més info](#)) i s'han de comprimir les dades que s'envien. A més, la connexió es farà utilitzant l'usuari vicentSsh i s'establirà la connexió amb el port 30 del servidor. Observeu que, perquè la connexió funcioni, el servidor SSH, al seu torn, haurà d'estar configurat per treballar al port 30.

1.2.2.2 Arxius de configuració del servidor SSH

La funció del servidor SSH és esperar les connexions dels clients (normalment al port TCP 22), dur a terme la seva autenticació i, si tot ha anat bé, obrir una sessió de treball, executar una ordre o bé redreçar ports.

Cada vegada que es rep un intent de connexió des d'un client, es fan en primer lloc totes les comprovacions i inicialitzacions criptogràfiques per garantir la seguretat. Després es tracta d'autenticar l'usuari i finalment se segueixen els passos d'un procés d'inici de sessió habitual.

1.2.2.2.1 Fitxer de configuració sshd_config

Malgrat que en el funcionament del servidor d'OpenSSH hi intervenen diversos fitxers, l'arxiu principal de configuració és **/etc/ssh/sshd_config**. Aquest fitxer conté diferents paraules clau amb el seu valor. Les línies que comencen amb # (comentaris) o les que estan en blanc són ignorades.


```

Include /etc/ssh/sshd_config.d/*.conf

#Port 22
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::

#HostKey /etc/ssh/ssh_host_rsa_key
#HostKey /etc/ssh/ssh_host_ecdsa_key
#HostKey /etc/ssh/ssh_host_ed25519_key

# Ciphers and keying
#RekeyLimit default none

# Logging
#SyslogFacility AUTH
#LogLevel INFO

# Authentication:

#LoginGraceTime 2m
#PermitRootLogin prohibit-password
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10

#PubkeyAuthentication yes

# Expect .ssh/authorized_keys2 to be disregarded by default in future.
#AuthorizedKeysFile .ssh/authorized_keys .ssh/authorized_keys2

#AuthorizedPrincipalsFile none

#AuthorizedKeysCommand none
#AuthorizedKeysCommandUser nobody

```

Com podeu veure el fitxer conté molts exemples comentats que ens poden servir, per defecte ve activa:

- ChallengeResponseAuthentication -> Versió “deprecated” de KbdInteractiveAuthentication
- UsePAM -> Especifica que es permet el login mitjançant PAM. Més info a [RedHat](#).
- X11Forwarding
- PrintMotd
- AcceptEnv
- Subsystem per sftp

A la taula podeu trobar els principals paràmetres de configuració que podem configurar al servidor SSH.

Taula: Paràmetres de configuració del servidor SSH

Opció	Funció
AcceptEnv	Especifica quines variables d'entorn enviades pel client es copiaran a l'entorn de la sessió(7). Vegeu SendEnv i SetEnv a ssh_config(5) per saber com configurar el client.

AllowGroups	Si s'especifica seguida d'una llista de grups (separats per espais), només els usuaris que tenen algun dels grups indicats com a grup principal o suplementari podran iniciar sessió. És possible emprar els patrons ? i * en la definició dels grups. Només es poden indicar els grups mitjançant el seu nom, no en format numèric (GID).
AllowUsers	Té la mateixa funció que <i>AllowGroups</i> , però per als usuaris. En aquest cas, a més, és possible indicar des de quins amfitrions d'origen s'acceptarà la connexió. Per exemple: <i>AllowUsers usuari1@192* usuari2</i> .
Banner <fitxer>	Envia el contingut del fitxer indicat al client abans de dur a terme l'autenticació.
Compression <yes delayed no>	Especifica si es farà servir la compressió. El valor <i>delayed</i> , l'opció predeterminada, indica que només es farà servir la compressió un cop s'hagi autenticat l'usuari.
DenyGroups	Permet especificar una llista de grups, separats per espais, als quals no es permetrà iniciar sessió. Es poden especificar els grups mitjançant el seu nom, emprant els patrons ? i * de manera opcional.
DenyUsers	Igual que <i>DenyGroups</i> , però per als usuaris. En aquest cas és possible indicar un equip (o subxarxa) per a cada usuari.
KbdInteractiveAuthentication	Especifica si es permet l'autenticació interactiva amb el teclat.
LoginGraceTime	Període de temps màxim per dur a terme l'autenticació. El valor 0 expressa que no hi ha límit.
MaxAuthTries	Nombre màxim d'intents d'autenticació que es poden fer.
MaxStartups	Nombre màxim de connexions simultànies que encara no han completat la seva autenticació.
PasswordsAuthentication <yes no>	Indica si s'accepta l'autenticació mitjançant contrasenya (<i>password</i>).
PermitEmptyPasswords <no yes>	Si s'utilitza l'autenticació mitjançant contrasenya, especifica si el servidor permet la connexió a comptes que tenen una contrasenya buida.
PermitRootLogin <yes without-password forced-commands-only no>	Especifica si s'accepta la connexió de superusuari mitjançant SSH. El valor <i>without-password</i> indica que el superusuari no podrà fer servir l'autenticació basada en contrasenya i el valor <i>forced-commands-only</i> , que només es permetrà l'autenticació de clau pública per executar certes ordres de manera remota (normalment per fer còpies de seguretat).
Port ListenAddress	Permeten especificar el port on el servidor escoltarà les connexions dels clients i les adreces on obrirà aquest port. De manera predeterminada s'utilitza el port 22 de qualsevol adreça local. És possible especificar múltiples vegades aquestes opcions, però convé que <i>Port</i> sempre aparegui abans que <i>ListenAddress</i> .
PrintMotd	Especifica si sshd(8) ha d'imprimir a /etc/motd quan un usuari inicia sessió de manera interactiva. (En alguns sistemes també s'imprimeix per l'interpret d'ordres, /etc/profile o equivalent.) El valor predeterminat és sí.
Protocol	Especifica quins protocols es podran fer servir en les connexions dels clients (1, 2 o tots dos). És important recordar que el protocol SSH2 és força més segur que l'SSH1.
PubkeyAuthentication <yes no>	Especifica si s'acceptarà l'autenticació de clau pública.
Subsystem	Configura un subsistema extern (per exemple, un dimoni de transferència de fitxers). Els arguments haurien de ser un nom de subsistema i una ordre (amb arguments opcionals) per executar-se a petició del subsistema. L'ordre sftp-server implementa el subsistema de transferència de fitxers SFTP. Alternativament, el nom internal-sftp implementa un servidor SFTP en procés..
X11Forwarding <no yes>	Especifica si s'acceptarà el reenviament X11 per tal que les aplicacions gràfiques executades en el servidor obrin la seva finestra en el servidor X del client.

Teniu més informació a la pàgina de man: [config sshd config](#).

1.2.2.2.2 Altres arxius de configuració

Hi ha altres arxius de configuració que permeten de forma genèrica el filtratge, control d'accés i mecanismes de protecció de diferents serveis (POP, Sendmail, Telnet, SSH, etc.) actuant de fet com un tallafocs bàsic.

Així, si volem habilitar o restringir l'accés a determinats equips i serveis podem editar els arxius de configuració **/etc/hosts.deny** i **/etc/hosts.allow** indicant en la directiva dintre de l'arxiu el servei que volem controlar, en aquest cas, el dimoni SSH. D'aquesta manera el sistema, davant d'una petició d'accés al servei, fa la cerca següent, que conclou en el moment de la primera coincidència:

- 1 Comprova l'arxiu `/etc/hosts.allow`. Si hi troba coincidència valida l'accés.
- 2 Comprova l'arxiu `/etc/hosts/deny`. Si hi troba coincidència no valida l'accés.
- 3 En cas de no trobar coincidència en cap dels arxius valida l'accés.

Exemples de directives d'aquests arxius:

- 1 `sshd: ALL` (permet/denega l'accés ssh a tothom)
- 2 `sshd: 192.168.18.43` (permet/denega l'accés SSH de la IP 192.168.18.43)

Recordeu que perquè qualsevol canvi tingui efecte s'ha de reiniciar el servei:

`/etc/init.d/ssh restart`

Per a informació completa sobre els arxius de configuració `/etc/hosts.allow` i `/etc/hosts.deny` consulteu la informació del sistema amb l'ordre: `man hosts_access`.

1.2.3 Connexió a una estació remota amb SSH

La funció més comuna per a SSH és establir una sessió de treball remota fent ús de tècniques criptogràfiques per transmetre la informació. L'ús del client SSH és força senzill.

```
$ ssh user@host
```

- user: és l'usuari que es connectarà a la màquina remota.
- host: representa la IP o el nom de domini del servidor SSH al qual ens volem connectar.

És tot el que hem d'escriure per iniciar una sessió remota com a usuari (user) en l'equip amfitrió (host). En executar l'ordre ens demanarà la contrasenya de l'equip remot i, si l'escrivim de manera correcta, podrem accedir a la sessió de treball remota per escriure ordres.

```
$ ssh vicentSsh@192.168.18.43
```

Per finalitzar la connexió escriurem *exit*.

```
$ exit
```

Si no s'especifica el nom d'usuari a l'hora d'invocar l'ordre SSH, intentarà fer la connexió amb l'usuari amb el qual estem connectats al terminal de GNU/Linux.

```
usuari1@vicent-VirtualBox$ ssh 192.168.18.43
```

```
usuari1@192.168.18.43's password:
```

Fixeu-vos que en aquest cas intenta connectar-se com a usuari1, ja que no s'ha especificat amb quin usuari s'ha de connectar.

El client d'OpenSSH disposa de diferents opcions que es detallen en el seu manual. Algunes de les més freqüents són les descrites a la taula

Taula: Opcions més freqüents del client OpenSSH

Opció	Funció
-1	Força l'ús de la versió 1 del protocol SSH. Només es recomana emprar SSH1 per connectar-se a servidors antics que no són compatibles amb SSH2.
-2	Força l'ús de la versió 2 del protocol: SSH2.
-4	Força l'ús de l'adreçament IPv4.
-6	Força l'ús de l'adreçament Ipv6.
-C	Activa la compressió gzip en la connexió. Es recomana activar la compressió si s'està emprant SSH amb un enllaç lent, com un mòdem. Si l'enllaç és de banda ampla es recomana treballar sense compressió.
-p port	Port al qual es connectarà en l'equip remot. De manera predeterminada el servidor SSH s'executa al port TCP 22, però si es tracta d'un servidor accessible des d'Internet és recomanable escollir un altre port per evitar els intents de connexió.
-q	No imprimeix els missatges d'avertència, només els errors. Amb una altra -q no imprimeix ni els errors.
-X	Activa la retransmissió X11 per tal que els programes gràfics llançats en l'estació remota obrin la seva interfície gràfica en el servidor X local.
-x	Desactiva la retransmissió X11.

Algunes vegades només es desitja executar una ordre a l'estació remota, no obrir un shell (intèrpret d'ordres) per treballar. En aquest cas, és possible indicar l'ordre que cal executar en la mateixa crida de SSH.

Per exemple, per tal de veure el final del fitxer de registre /etc/log/auth.log al servidor 192.168.18.43 farem:

```
$ ssh vicentSsh@192.168.18.43 tail /etc/log/auth.log
```

1.2.4 Transferència d'arxius entre equips

Entre les utilitats incloses en la distribució d'OpenSSH trobem les ordres scp i sftp. Aquestes ordres permeten transferir fitxers amb totes les garanties de seguretat d'SSH.

1.2.4.1 Còpies segures (scp)

La sintaxi bàsica de la instrucció secure copy (scp) és:

```
scp [opcions] [[user@]host1:]fitxer1 [[user@]host2:]fitxer2
```

- host1: representa la màquina d'origen.
- host2: representa l'estació de destinació.

L'ordre **scp** es pot veure com una versió estesa de l'ordre **cp**, que permet copiar fitxers fins i tot entre màquines diferents. De fet, és el substitut **d'SSH** per l'ordre rcp, que és antiga i insegura. En fer una còpia mitjançant scp, es pot escollir qualsevol combinació de fitxers locals o remots tant per l'origen com per la destinació.

En l'exemple que podeu veure a continuació copiem el fitxer local manual.pdf en una estació remota amb IP 192.168.18.43.

```
vicentSsh@vicent-VirtualBox:~$ scp manual.pdf 192.168.18.43:
```

```
vicentSsh@192.168.18.43's password:
```

```
manual.pdf 100% 179KB 179.5KB/s 00:00
```

Si la contrasenya és correcta es procedirà a la transferència de l'arxiu i s'indicarà de manera interactiva el percentatge que ja s'ha enviat, la grandària en kilobytes, l'índex de transferència i el temps transcorregut des del començament de la transmissió.

No només es poden copiar fitxers locals en una estació remota, sinó que es poden copiar fitxers de l'estació remota en la local i fins i tot entre dues estacions remotes. A la taula podeu trobar un exemple de cada tipus.

Taula: Exemples de diferents usos de la instrucció **scp**

Exemple	Ordre
Copiar fitxer local a equip remot	scp fitxer usuari@192.168.18.43:/home/usuari
Copiar fitxer remot a equip local	scp usuari@192.168.18.43:/home/usuari/fitxer ./
Copiar un fitxer d'un equip remot a un altre equip remot	scp usuari@192.168.56.12:/home/usuari/fitxer usuari@192.168.56.11:/home/usuari

La pàgina del manual d'scp ens mostrarà els seus paràmetres, la major part dels quals són els mateixos que té l'ordre ssh. No obstant això, té algunes opcions pròpies que són particularment útils i que podeu veure a la taula.

Taula: Opcions de la instrucció scp

Opció	Funció
-l limit	Establir un límit a l'amplada de banda en kbps.
-p	Preservar el temps de modificació, accés i els permisos dels fitxers copiats.
-r	Fer una còpia recursiva pels directoris.

1.2.4.2 Transferències segures de fitxers (sftp)

L'ordre sftp és un substitut per al client d'FTP, que és el tradicional però que és insegur. En emprar sftp s'estableix una connexió al sistema remot i després, de manera interactiva, es podran indicar ordres per explorar el sistema d'arxius remot i fer modificacions.

La sintaxi bàsica de la instrucció sftp és:

```
sftp [opcions][[user@]host1:]fitxer1 [[user@]host2:]fitxer2
```

- host1: representa la màquina d'origen.
- host2: representa l'estació de destinació.

Un cop establerta la connexió segura, podem executar les mateixes ordres que si ens haguéssim connectat mitjançant FTP. A la taula podeu trobar les ordres que sftp pot interpretar.

Taula: Ordres que poden ser interpretades per sftp

Ordre	Funció
bye, exit, quit	Finalitzar la sessió.
cd camí	Canviar el directori remot.
chgrp, chown, chmod	Canviar el grup, el propietari o els permisos en el sistema remot.
get fitxer	Transmetre el fitxer remot indicat al directori de treball local.
put fitxer	Transmetre el fitxer local indicat al directori de treball remot.
ls, mkdir, pwd, rename, rm, rmdir, ln	Elaborar llistats, crear directoris, consultar la ruta, canviar de nom, eliminar fitxers i directoris i crear enllaços simbòlics al sistema de fitxers remot.
lcd, lls, lmkdir, lpwd	Versions de les ordres <i>cd</i> , <i>ls</i> , <i>mkdir</i> i <i>pwd</i> per treballar al sistema de fitxers local.

Vegem un exemple de com fer transferències segures entre l'ordinador local i un equip amb la IP 192.168.100.10:

```
vicentSsh@vicent-VirtualBox:~$ sftp vicentSsh@192.168.100.10
vicentSsh@192.168.100.10's password:
Connected to 192.168.100.10.
sftp>
```

En aquest punt ja s'ha establert la connexió segura amb l'equip remot i ja es poden començar a utilitzar les ordres pròpies d'sftp. En l'exemple següent es transferirà el fitxer fitxer.pdf de l'equip remot al local.

```
sftp> cd Documents/
sftp> get manual.pdf
```

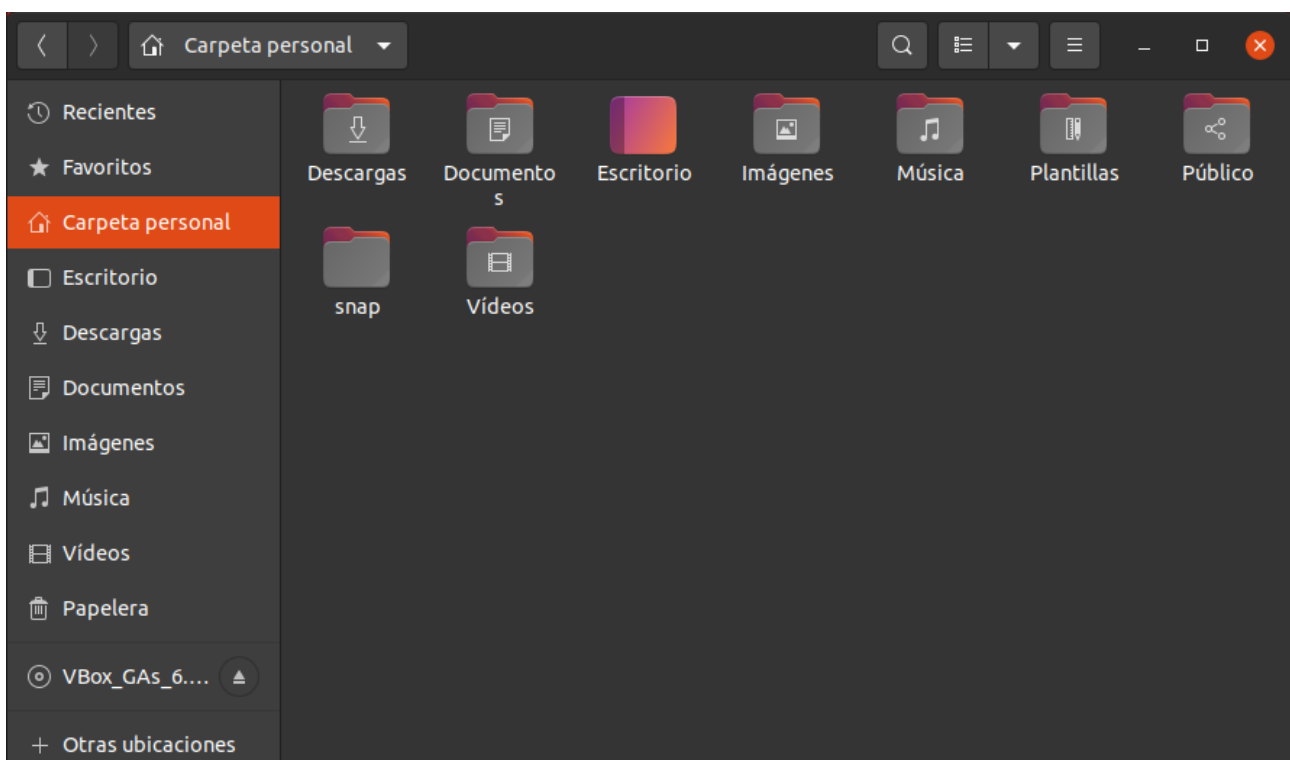
```
Fetching /home/vicentSsh/Documents/manual.pdf to manual.pdf
/home/vicentSsh/Documents/manual.pdf 100% 179KB 179.5KB/s 00:00
sftp>
```

1.2.4.3 sftp a l'escriptori GNU/Linux

Els escriptoris **GNOME** i **KDE** també proporcionen un accés senzill a servidors SSH per tal de treballar en xarxa. A GNOME, el navegador d'arxius Nautilus pot mostrar directoris remots com si fossin locals fent servir el GNOME VFS (sistema de fitxers virtual de GNOME).

Com que la distribució Debian utilitza GNOME per defecte, per tal d'utilitzar gràficament la instrucció sftp al visualitzador d'arxius Nautilus, només és necessari introduir a “Altres ubicacions” o “Otras ubicaciones” la mateixa ordre que s'utilitza per obrir una connexió SSH. Vegeu la [figura.4](#).

Nautilus:



```
sftp://nomUsuari@ipServidor
```

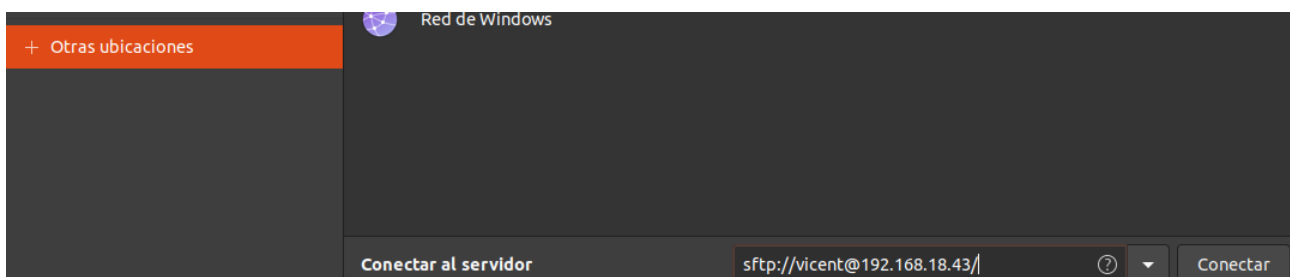


Figura Introduir a la barra d'adreces l'ordre de connexió ssh

Després d'uns segons per establir la comunicació, el navegador d'arxius Nautilus detecta que es vol fer una connexió segura i demana la contrasenya de l'usuari de l'equip remot amb una finestra similar a la de la figura

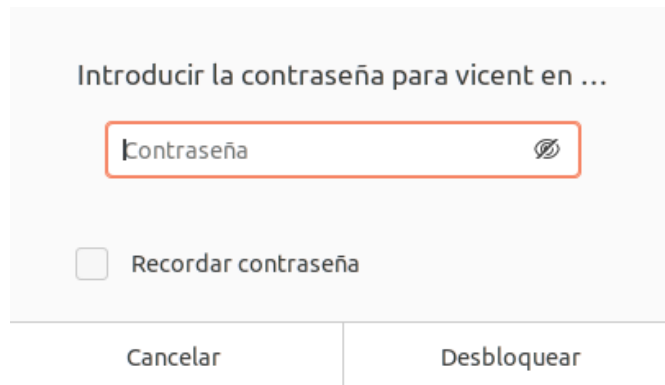


Figura Nautilus demana la contrasenya per connectar-se a l'equip remot

Si la contrasenya és correcta, s'estableix la connexió remota i Nautilus passa al mode de transferència segura d'arxius i la unitat remota queda muntada. A la [figura.6](#) es pot observar que es visualitzen els arxius de l'equip remot.

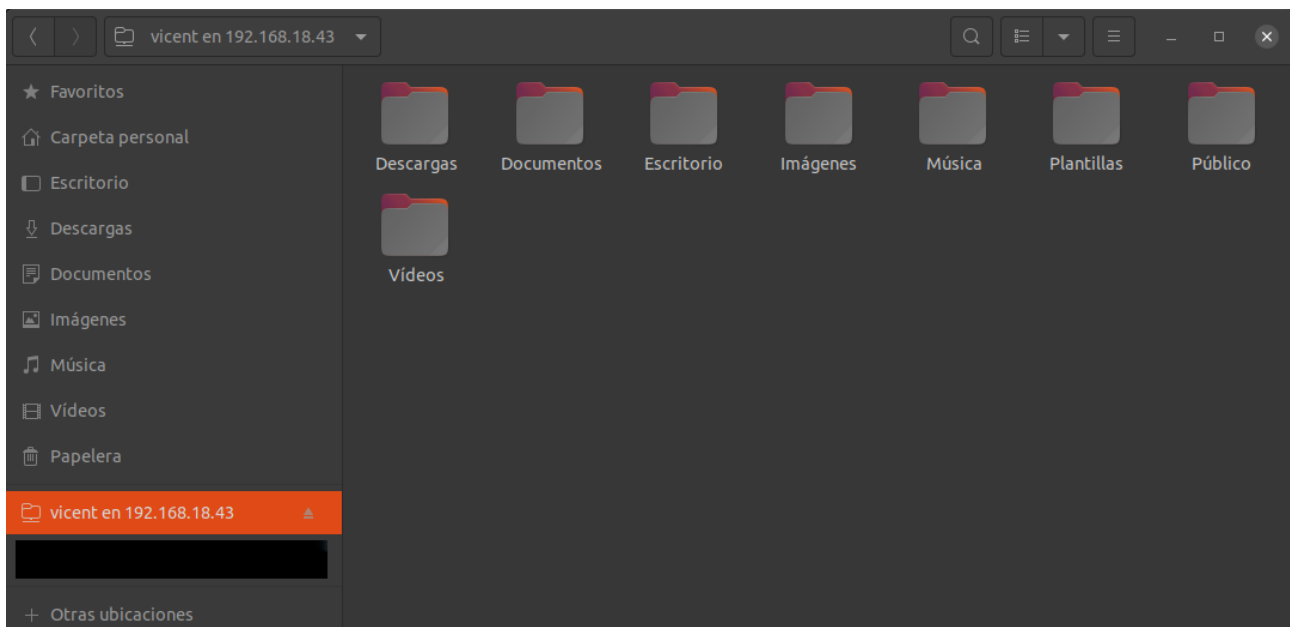


Figura Nautilus mostra la unitat remota per fer la transferència segura d'arxius

Utilitzant aquesta finestra, sempre que es tinguin els permisos necessaris, es poden copiar arxius, crear carpetes o fer qualsevol altra operació típica d'un navegador de fitxers.

1.2.5 Redreçament de ports TCP i túnels amb SSH

La major part dels protocols que utilitzem en les nostres comunicacions estan basats en dissenys de fa gairebé 40 anys, quan la seguretat en xarxes telemàtiques no era un problema. Telnet, FTP, POP3, protocols d'ús quotidià, descuiden la seguretat i confidencialitat de les dades que envien. No serveix de res protegir els servidors, implantar una bona política de contrasenyes i actualitzar les versions dels nostres dimonis si després, quan un usuari de POP3, per exemple, vol veure el seu correu electrònic des de la nostra xarxa, envia el seu usuari i contrasenya en text pla per la xarxa.

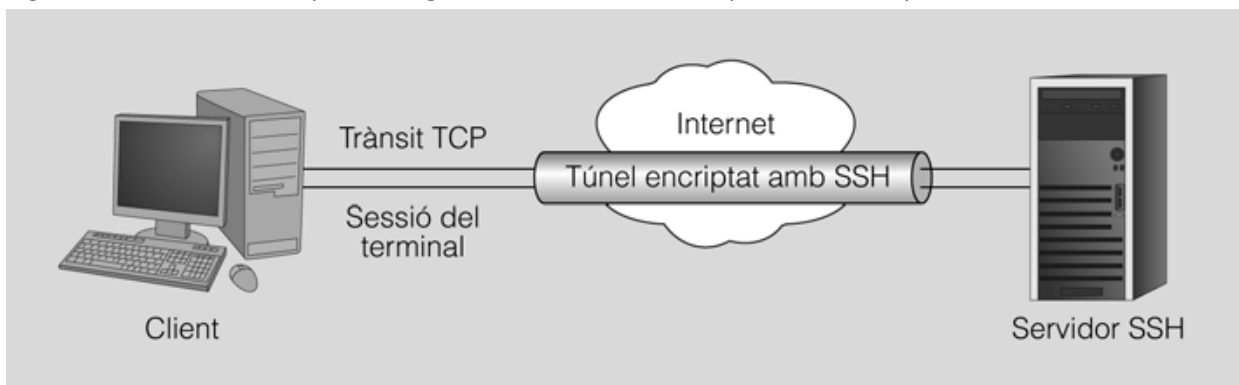
Fent servir SSH es poden establir túnels encriptats pels quals es pot transmetre qualsevol protocol que faci servir TCP. Així és possible, per exemple, emprar un túnel SSH per:

- descarregar el correu mitjançant POP3 i enviar-lo fent servir SMTP.
- accedir a un servidor web mitjançant HTTP.
- accedir a un sistema d'arxius remot mitjançant NFS.

Els protocols de l'exemple (POP3, SMTP, HTTP i NFS) no utilitzen tècniques criptogràfiques. El seu ús en una xarxa que no és de confiança no ens permet garantir la confidencialitat. Però com que tots aquests serveis utilitzen TCP com a protocol de transport, poden ser tunelitzats per SSH.

La idea en què es basa aquest procediment, i que es representa a la [figura.7](#), és la de fer un túnel pel qual viatjaran les dades de manera segura (*tunneling*). A cada un dels extrems del túnel hi ha les aplicacions estàndard (un dimoni POP3 estàndard, el nostre client de correu preferit, FTP, un navegador web, etc.) i la comunicació s'assegura fent ús de tota la potència criptogràfica d'SSH. SSH recull les dades que el client vol enviar i les reenvia pel túnel o canal segur. A l'altre costat del túnel es recullen les dades i es tornen a enviar al servidor corresponent.

Figura Túnel fet amb SSH per fer segures les comunicacions que utilitzen el protocol TCP



1.2.5.1 Redreçament estàtic de ports: túnel SSH per accedir al servei SMTP

El redreçament estàtic de ports ens permet crear un canal de comunicació segur i transparent a l'usuari entre un port de la màquina local i un altra port de la màquina remota, que pot ser un dels ports estàndards que fan servir qualsevol dels serveis de xarxa.

Per exemple, per establir un túnel SSH entre el port local 10025 i el port 25 (corresponent al servei SMTP) de l'estació 192.168.10.100 establint la connexió amb l'usuari *usuari* faríem:

```
$ ssh usuari@192.168.10.100 -L 10025:192.168.10.100:25
```

Un cop establerta la sessió SSH, i mentre es mantingui, existirà un túnel encriptat entre el port TCP local 10025 i el 25 de l'estació 192.168.10.100.

En altres paraules, per fer una connexió segura amb el servidor SMTP de l'estació remota, hauríem de connectar amb el port local 10025. SSH s'encarregarà de fer el transport de manera segura entre totes dues estacions.

1.2.5.2 Redreçament dinàmic de ports: Servidor SOCKS

Una altra de les possibles aplicacions del túnel SSH és accedir a un servidor web des d'una IP diferent de la de la nostra màquina. Com es pot veure a la [figura.8](#), el navegador utilitza el túnel creat per SSH per connectar-se al servidor i navegar per pàgines web utilitzant la IP externa del servidor.

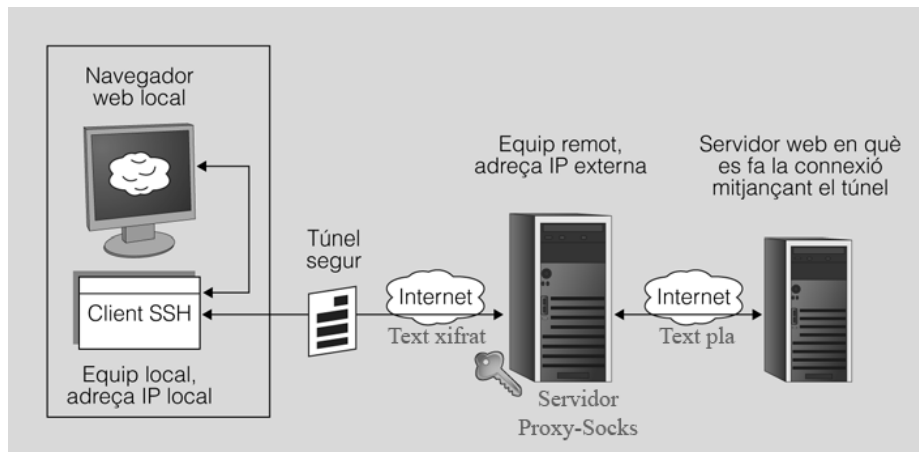


Figura Navegar per Internet utilitzant la IP del servidor

SSH permet doncs crear un túnel des de l'equip local fins a un servidor remot. Un cop connectats a aquest servidor remot utilitzem el túnel per fer la navegació per Internet. A la [figura.9](#) podeu veure les IP de cada un dels equips de l'exemple. Fixeu-vos que tant la màquina local com el servidor remot tenen la seva pròpia IP local i una **IP externa (NAT)**. Per connectar-nos al servidor remot haurem d'utilitzar l'adreça IP externa de la xarxa on està connectat, i només funcionarà si l'encaminador al qual ens connectem té el port 22 redreçat al servidor amb el qual farem el túnel. És a dir, quan l'encaminador remot rebí una connexió pel port 22 ha de redreçar aquest trànsit al port 22 del nostre servidor.

NAT (network address translation):
La traducció d'adreces de xarxa és una tècnica que amaga un espai d'adreces, que generalment consisteix en un conjunt d'adreces de xarxa privada, darrere d'una única adreça IP, sovint en l'espai d'adreces IP públiques. Aquest mecanisme s'implementa en un encaminador que utilitza unes taules per assignar les adreces "ocultes" a una sola adreça IP, de manera que els paquets a la sortida semblen originar-se en l'encaminador.

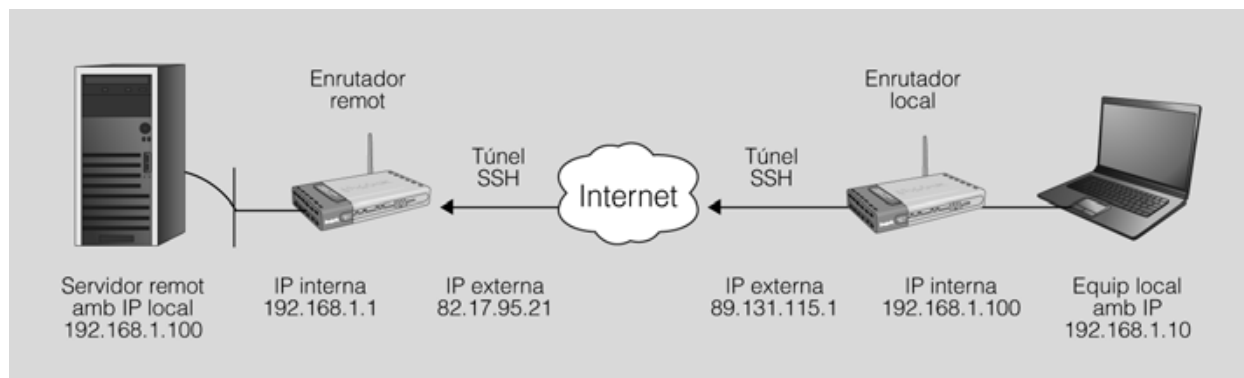


Figura Configuració d'equips per fer un túnel SSH

Abans de fer el túnel SSH, si es fa una connexió a Internet des de l'equip local amb un navegador qualsevol, es farà des de l'adreça IP 89.131.115.1.

A continuació veiem com fer el túnel amb l'equip remot:

```
ssh -D 1080 -p 22 -Nf usuari@82.17.95.21
```

L'opció **-D** habilita el **redreçament dinàmic de ports locals** que a la pràctica permet fer servir SSH com un servidor SOCKS.

Un **servidor SOCKS** dona a la intranet un servei similar al que proporciona un servidor web intermediari (*proxy*), però no està limitat al protocol HTTP/HTTPS, sinó que permet redreçar qualsevol tràfic TCP/IP. Fins i tot la versió 5 de SOCKS permet el tràfic UDP.

El servidor SOCKS funciona mitjançant l'assignació d'un sòcol per escoltar el port local. En el nostre cas hem triat el port 1080, ja que és l'estàndard per a aquest servei. Cada vegada que s'estableix una connexió a aquest port, la connexió es transmet pel canal segur. A continuació el protocol d'aplicació és l'encarregat de determinar on es fa la connexió a la màquina remota.

SOCKS: *Protocol d'Internet que facilita l'encaminament de paquets de xarxa entre les aplicacions client-servidor a través d'un servidor intermediari.*

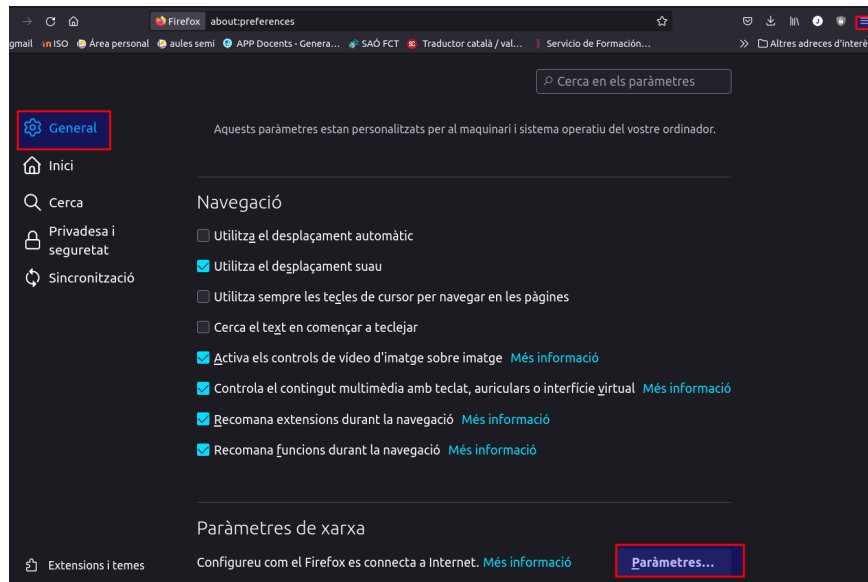
A la [taula.10](#) veiem amb detall les opcions emprades.

Taula: Opcions de redreçament dinàmic de l'ordre ssh

Opció	Descripció
-D 1080	Habilita el redreçament dinàmic de ports fent servir, en aquest cas, el port 1080 estàndard per a servidors SOCKS.
-p 22	Permet indicar el port pel qual es farà el túnel SSH, habitualment el número 22.
-N	Especifica que no es vol executar cap ordre remota i no s'obrirà cap terminal interactiu. És d'utilitat quan es fa servir SSH per al redreçament de ports.
-f	Com que no ens cal interactivitat, aquesta opció fa que SSH quedi en segon pla i es dissociï de la <i>shell</i> actual, cosa que converteix el procés en un dimoni (cal l'opció -N).

Un cop establert el túnel, cal configurar a nivell d'aplicació el programa que el farà servir. En aquest cas l'aplicació que hem de configurar és el navegador local (en el meu cas Mozilla Firefox 94.0) perquè utilitzi el túnel SSH per accedir a la xarxa d'Internet.

Si utilitzeu el navegador Firefox, heu d'anar a la icona de les tres línies a la part superior dreta > *General* > *Paràmetre de xarxa* i prémer el botó *Paràmetres*. S'ha de configurar, tal com mostra la captura. Servir la configuració manual del servidor intermediari (*proxy*), que l'ordinador central SOCKS sigui *localhost*, que utilitzi el port 1080 i activar el DNS proxy.



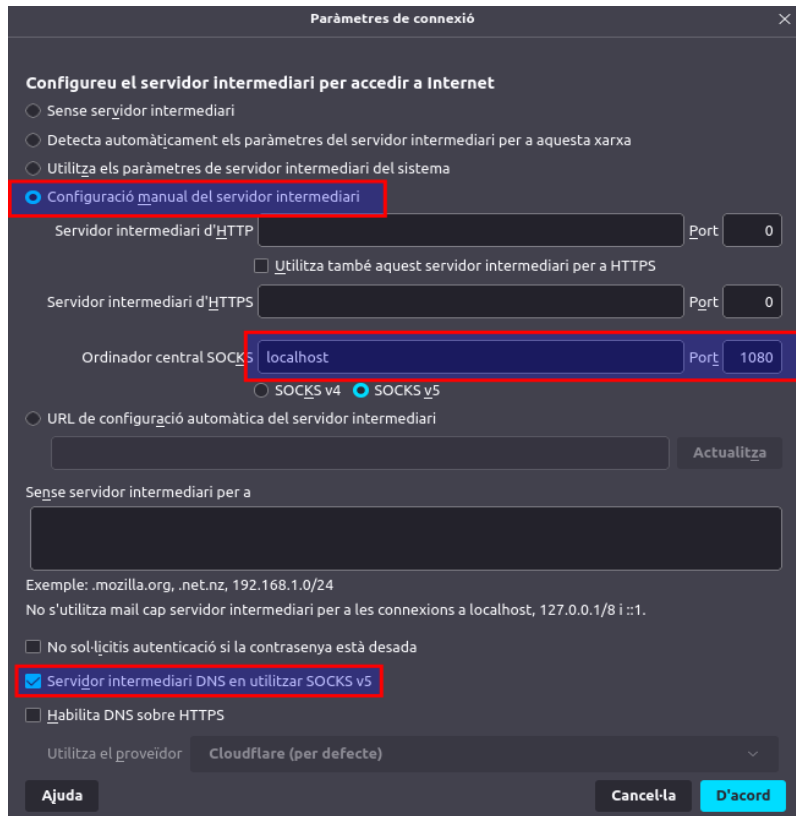


Figura Configuració del navegador Firefox per utilitzar el servidor SOCKS

Ara el navegador ja utilitza el túnel SSH i quan es fa una connexió a Internet des de l'equip local es fa des de l'adreça pública de l'encaminador remot (en aquest cas, amb IP 82.17.95.21).

Hi ha aplicacions que no admeten específicament l'ús del servidor SOCKS i que no disposen de la possibilitat de configurar aquesta opció. En aquest cas es pot carregar un programa addicional que detecti peticions a la pila TCP/IP per modificar-les i redreçar-les a través del servidor SOCKS sense que la aplicació hagi de tenir implementada aquesta possibilitat. Les més conegudes són *tsocks* i *proxychains*.

1.3 Administració remota amb interfície gràfica

Si la xarxa de comunicacions té una amplada de banda suficient i un retard raonable, es poden fer servir eines que permeten treballar amb aplicacions que s'executen en un equip remot, però que mostren la seva interfície gràfica en un terminal local. En tot cas, la seguretat ha de ser un punt fonamental a considerar, atès que la funcionalitat administrativa de l'aplicació farà que transportem per la xarxa dades importants.

Les opcions més conegudes per l'administració remota són:

- El sistema servidor X Window (transparència de xarxa)
- Computació virtual en xarxa VNC (*virtual network computing*)
- Webmin, eina gràfica per a l'administració remota de Linux

1.3.1 Protocols d'accés remot a interfícies gràfiques

Hi ha dues propostes tecnològiques principals quant a protocols d'accés remot a interfícies gràfiques:

- D'una banda, es poden transmetre directament les diferents directives que donen instruccions als motors gràfics dels servidors de finestres respectius.
- De l'altra, es pot transmetre informació relativa a cada píxel que permet fer una representació del mapa de memòria gràfica del gestor de finestres.

Alguns dels protocols poden incorporar les seves pròpies tècniques de seguretat i xifrat o bé incorporar una capa addicional de seguretat, per exemple fent servir túnels SSH.

1.3.1.1 Protocol X11

Protocol desenvolupat inicialment a mitjan anys vuitanta a l'Institut Tecnològic de Massachusetts (MIT) com a part del sistema de finestres X Window amb l'objectiu de proporcionar una interfície gràfica als sistemes Unix.

La versió del protocol que es fa servir des de 1987 i fins a l'actualitat és la v11. Per aquesta raó s'acostuma a anomenar X11 tant al protocol com als servidors i clients del sistema X Window.

Aquest protocol és independent del sistema operatiu i permet la interacció gràfica remota entre un client i un servidor fent transparent la xarxa per a l'usuari, que veu el sistema com si fos un terminal gràfic virtual.

1.3.1.2 Tecnologia NX

Més que un protocol de comunicació pròpiament dit, NX és una tecnologia que permet gestionar i millorar les connexions X Window fent servir compressió del tràfic de dades del mateix protocol X11 i afegint mecanismes de memòria cau per accelerar la comunicació i reduir el requeriments d'amplada de banda i latència. FreeNX És una implementació en codi lliure amb llicència GPL de la tecnologia client-servidor NX.

1.3.1.3 Remote framebuffer (RFB)

El protocol de xarxa RFB (remote framebuffer protocol) també és lliure i fa servir la tècnica d'enviar informació dels píxels que conformen la memòria d'imatge gràfica.

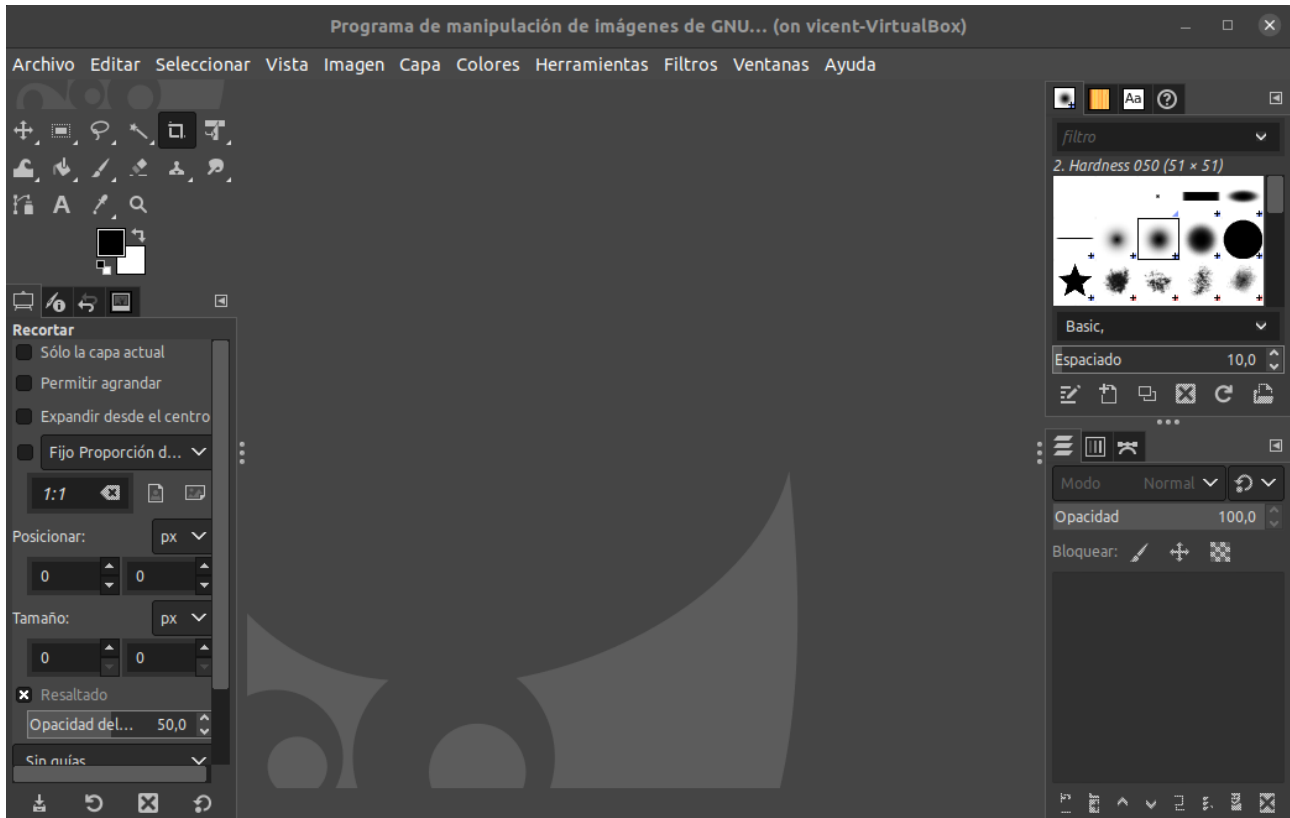
La memòria d'imatge (framebuffer) és l'àrea de la memòria que emmagatzema les dades que defineixen la imatge gràfica que apareix a la pantalla. Normalment, les pantalles actuals estan basades en píxels, és a dir, mostren mapes de bits en comptes d'imatges vectorials.

La memòria d'imatge conté la informació necessària per definir l'estat de cada píxel de la pantalla. La quantitat d'informació necessària dependrà de la resolució i la profunditat de color (la quantitat de bits necessaris per codificar el color, i potser la transparència, de cada píxel).

Els programes més coneguts que fan servir el protocol RFB són tota la família d'aplicacions **VNC (virtual network computing)**.

El protocol RFB fa servir per defecte el port 5900 del servidor per establir la comunicació. Alternativament, en cas de connexió a través d'un navegador es fa servir per defecte el port 5800.

Cal assenyalar que RFB no és un protocol segur. En fer servir VNC en una xarxa que no sigui de confiança, caldrà combinar el seu ús amb SSH o bé una altra tecnologia per crear una xarxa privada virtual (VPN).



L'aplicació GIMP s'està executant en l'ordinador servidor remot (en aquest cas vicent-VirtualBox). Aquest envia la informació gràfica mitjançant un canal segur SSH a l'ordinador client (que actua com a servidor gràfic X Window) i mostra les finestres resultants a l'usuari.

Cal indicar que el servidor SSH haurà de tenir activat el redreçament del protocol X (normalment està activat per defecte), és a dir que haurà de tenir el paràmetre següent en una línia de l'arxiu de configuració `/etc/ssh/ssh_config`:

```
// Activació del redreçament X en /etc/ssh/sshd_config
```

```
X11Forwarding yes
```


1.3.3 Virtual network computing (VNC)

VNC és un sistema per veure, i si s'escau també controlar, un escriptori remot.

Per aconseguir això, cal instal·lar dos programes: un servidor VNC a la màquina a la qual vulguem accedir i un visualitzador VNC a la màquina client. És possible utilitzar un visualitzador en un sistema operatiu determinat i un servidor en un de diferent. D'aquesta manera, és possible executar un sistema Windows complet des d'un Macintosh o GNU/Linux, o fer qualsevol combinació que puguem imaginar.

1.3.3.1 Funcionament de les aplicacions VNC

VNC fa servir el protocol RFB (*remote framebuffer*) que està basat en una memòria d'imatge i així pot treballar amb qualsevol sistema de finestres, ja sigui X Window, Windows o un altre.

En els sistemes GNU/Linux és possible exportar mitjançant VNC l'escriptori actual o fins i tot un terminal de text, o bé crear un escriptori virtual nou. Els sistemes Microsoft Windows no permeten la creació d'escriptoris virtuals nous, només permeten exportar en xarxa l'únic escriptori disponible. En el moment d'exportar un escriptori es poden definir dues contrasenyes diferents, una per a aquells usuaris que podran controlar de manera remota aquest escriptori i una altra per a aquells usuaris que el podran veure però no interactuar amb el teclat i el ratolí.

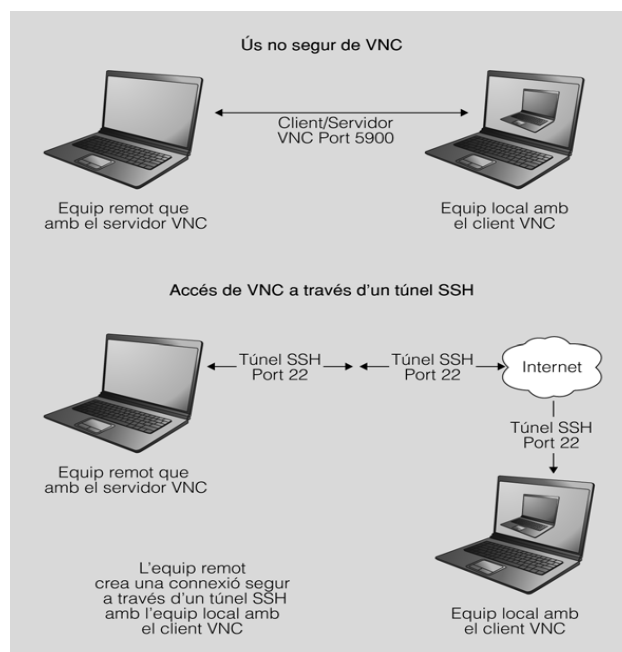


Figura Ús no segur de VNC i ús a través d'un túnel SSH

La major part de servidors VNC inclouen un petit servidor web que ofereix la descàrrega d'una miniaplicació Java que es pot fer servir com a client VNC. D'aquesta manera, en l'ordinador local només cal un navegador compatible amb Java per poder accedir a l'escriptori remot.

1.3.3.2 Implementacions VNC

A la taula podeu trobar un llistat dels productes més populars que implementen VNC.

Taula: Programaris que implementen VNC

Programari	Característiques
VNC	És el programari original. En l'actualitat només s'utilitza com a referència i per a les proves de compatibilitat.
TightVNC	Una versió que s'ofereix gratuïtament amb llicència GPL. S'han millorat significativament els algorismes de compressió de VNC i permet la transferència de fitxers.
RealVNC	És una versió desenvolupada per alguns dels desenvolupadors d'AT&T. L'equip de RealVNC també està oferint-ne una versió empresarial i ha elaborat un producte molt interessant que pretén ser accessible per a qualsevol client VNC.
UltraVNC	UltraVNC ofereix una funció de transferència d'arxius, una gestió millorada de la compressió de vídeo i una funció de xat. Recentment s'ha afegit l'opció de controlar una sola finestra del programa en lloc de controlar tot l'escriptori.
X11VNC	Ofereix una interface mitjançant l'entorn x11. Tot i que ja no està desenvolupat pel seu autor original Karl Runge, LibVNC i la comunitat GitHub s'han fet càrrec del desenvolupament i hui en dia és una de les millors opcions per la compatibilitat dels sistemes amb x11

Altres eines relacionades amb VNC i que cal destacar són **Vino**, una eina de GNOME per compartir l'escriptori i que acostuma a incloure Ubuntu. Tenim també diversos clients VNC com **Vinagre** i el potent client VNC que inclou la distribució Debian anomenat **Remmina**, que no només és compatible amb el protocol RFB de VNC sinó també amb la tecnologia NX o el protocol RDP de Microsoft Windows.

1.3.4 Programari d'accés remot TightVNC

El programari d'accés remot **TightVNC** és una de les implementacions de VNC més avançades. Es distribueix amb llicència GPL, la qual cosa permet un accés lliure al seu codi font, a la seva distribució i a la seva instal·lació.

Requisit: com que les connexions amb VNC poden no funcionar molt bé si no tenim una bona connexió a internet, recomane instal·lar i utilitzar l'entorn gràfic [xfce](#).

1.3.4.1 Instal·lació del servidor x11vnc

El paquet del servidor x11vnc s'instal·la només escrivint el següent a la línia d'ordres:

```
# apt install x11vnc
```

Un cop instal·lat ja podem arrencar el servidor VNC, *vncserver*, juntament amb una sèrie d'opcions.

```
# x11vnc -display :0 -geometry 1024x768 -forever &
```

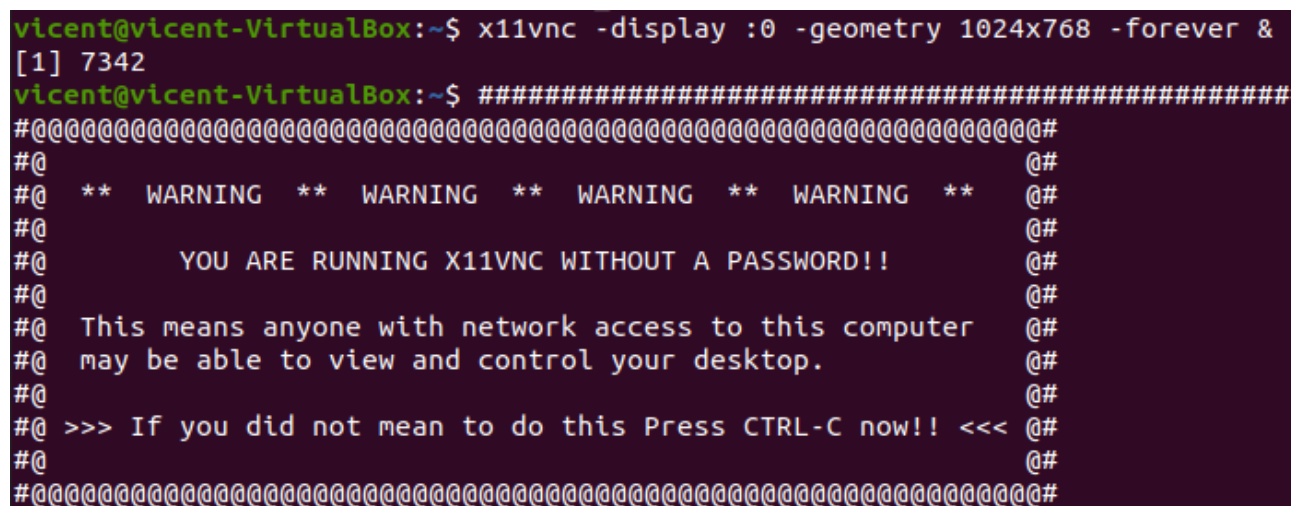
Opcions:

- `:0`: indica la pantalla en la qual s'establirà la sessió per fer les connexions remotes.
- `-geometry`: estableix les dimensions de la finestra amb què es farà la connexió.
- `-forever`: permet que quan tanquem la connexió des del client no es pare el servei VNC.
- `&` : l'et (o en castellà, ampersand), permet que l'execució del procés VNC continua en "background" mentre podem continuar utilitzant el terminal.

Podem obrir tantes pantalles com necessitem, simplement canviant `":1"` per `":2"`, `":3"`, `":4"` etc.

Teniu tota la informació per l'execució ací: [x11vnc](#)

A continuació, el programari ens demanarà dues contrasenyes (màxim 8 caràcters), la segona de les quals és opcional. La primera és la necessària per permetre l'accés total a l'equip servidor des de l'equip remot. La segona és per permetre només un accés de visualització de l'escriptori.



Per aturar el servidor **x11vnc** es pot fer servir la drecera de teclat **Ctrl+C** si no hem indicat l'et (**&**)

```
^Ccaught signal: 2
13/11/2021 09:50:34 deleted 38 tile_row polling images.
vicent@vicent-VirtualBox:~$
```

I si s'indica l'et (&), al terminal hem de buscar el procés del x11vnc actiu, per això podem fer ús de:

```
# ps -ef | grep x11vnc
```

```
vicent@vicent-VirtualBox:~$ ps -ef | grep x11vnc
vicent      7342      5950  0 09:55 pts/0    00:00:00 x11vnc -display :0 -geometry 1024x768 -forever
```

Com podem vore tenim el PID 7432 que correspon al x11vnc, ara només cal executar un kill per tancar-lo:

```
# kill 7342
```

Com podeu observar, quan arranquem el servei ens diu clarament que no tenim un contrasenya per la connexió, per tant qualsevol usuari podrà entrar sense restriccions.

Com arreglem això? Seguint els passos que ens dona el terminal:

```
#@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@#
#@
#@ You can create an x11vnc password file by running:                               @#
#@                                                                                     @#
#@      x11vnc -storepasswd password /path/to/passfile                             @#
#@ or    x11vnc -storepasswd /path/to/passfile                                     @#
#@ or    x11vnc -storepasswd                                                         @#
#@                                                                                     @#
#@ (the last one will use ~/.vnc/passwd)                                           @#
#@                                                                                     @#
```

Per emmagatzemar la contrasenya al propi usuari fem ús de:

```
# x11vnc -storepasswd
```

```
vicent@vicent-VirtualBox:~$ x11vnc -storepasswd
Enter VNC password:
Verify password:
Write password to /home/vicent/.vnc/passwd? [y]/n n
not creating password.
```

1.3.4.2 Instal·lació del client TightVNC a Ubuntu

El paquet que permet instal·lar el client TightVNC es diu *xtightvncviewer*.

```
# apt install xtightvncviewer
```

A continuació, un cop s'està executant el servidor de TightVNC, es pot fer la connexió des del client remot. S'ha d'indicar l'adreça IP del servidor i la pantalla on està funcionant el servidor.

```
# vncviewer 192.168.18.43:0
```

A la [figura.16](#) es pot veure com accedim des del terminal remot (vicent-MS-7C84) al servidor d'IP 192.168.18.43 i podem veure com s'obre l'escriptori d'aquest servidor (vicent-VirtualBox). Fixeu-vos que hem

introduït la contrasenya que permet accedir a la sessió. Si hi haguéssim accedit amb la contrasenya de només visualització, no hauríem pogut navegar pels menús del sistema.

```
vicent@vicent-VirtualBox: ~  
vicent@vicent-VirtualBox:~$ vncviewer 192.168.18.43  
vncviewer: ConnectToTcpAddr: connect: Connection refused  
Unable to connect to VNC server  
vicent@vicent-VirtualBox:~$ vncviewer 192.168.18.43  
Connected to RFB server, using protocol version 3.8  
No authentication needed  
Authentication successful  
Desktop name "vicent-VirtualBox:0"  
VNC server default format:  
 32 bits per pixel.  
Least significant byte first in each pixel.  
True colour: max red 255 green 255 blue 255, shift red 16 green 8 blue 0  
Warning: Cannot convert string "-*-helvetica-bold-r-*-16-*-16-*-*-*-*-*" to type FontStruct  
Using default colormap which is TrueColor. Pixel format:  
 32 bits per pixel.  
Least significant byte first in each pixel.  
True colour: max red 255 green 255 blue 255, shift red 16 green 8 blue 0
```

Figura Connexió remota mitjançant TightVNC

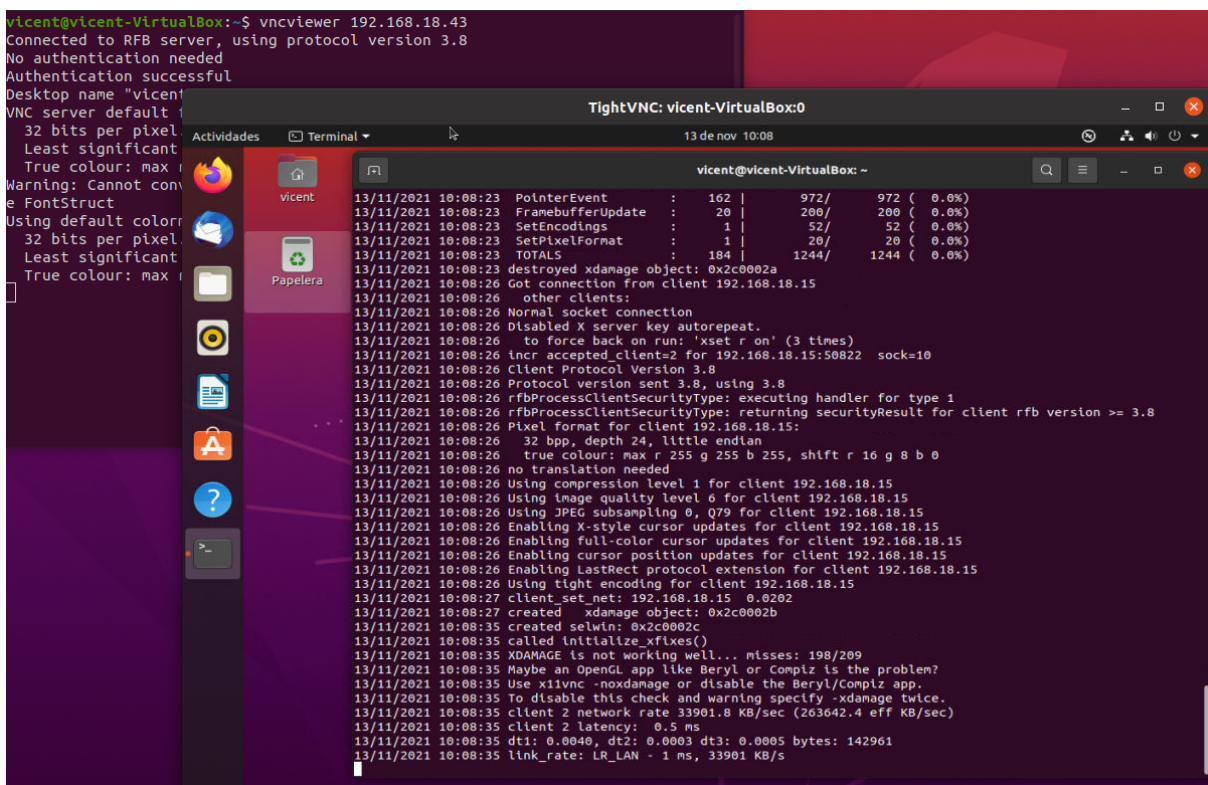
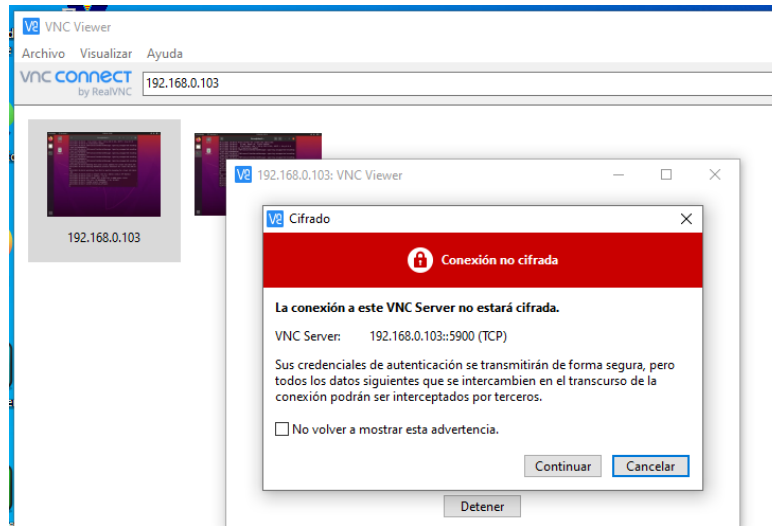


Figura Accés a l'escriptori remot mitjançant TightVNC

1.3.4.3 Instal·lació del client a Windows

Anem a instal·lar el client VNC Viewer al nostre Windows.

Una vegada fet, només hem de ficar la IP i a funcionar.



1.3.4.4 Connexió per VNC emprant un túnel SSH a Windows

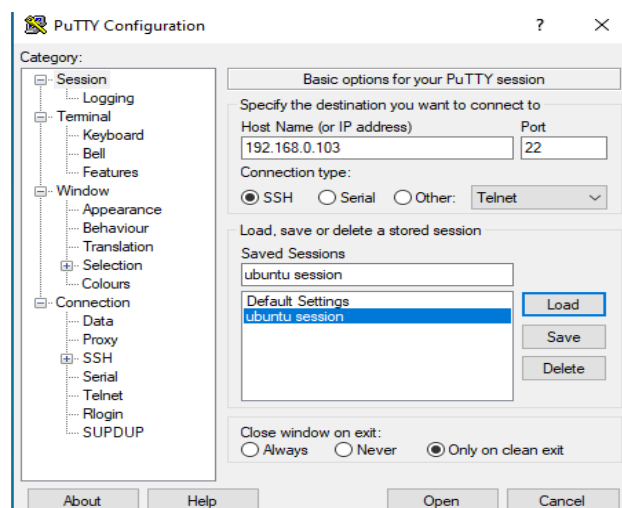
En aquest cas, a Ubuntu cal iniciar el servidor x11vnc indicant el port que anem a tunelitzar.

```
$ x11vnc -rfbport 5900 -shared
```

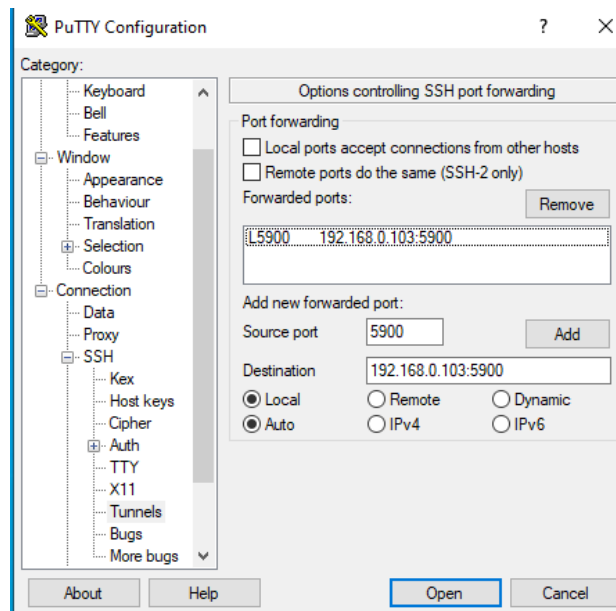
Ara crearem a Windows un túnel SSH amb redreçament estàtic de ports i posteriorment ens connectarem per VNC. D'aquesta forma les dades aniran encriptades.

Anem a seguir les passes del següent [vídeo](#).

Obrim Putty, fem la IP del servidor x11 i li fem un nom a la sessió.



Fem clic a connection > SSH > Tunnels. Introduïm el port origen i la IP destí amb el port i polsem Add

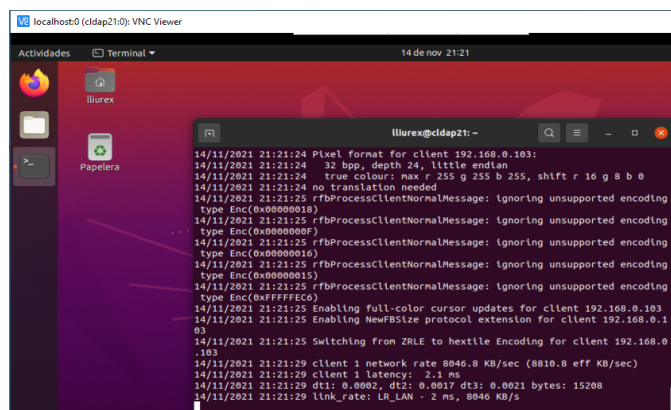
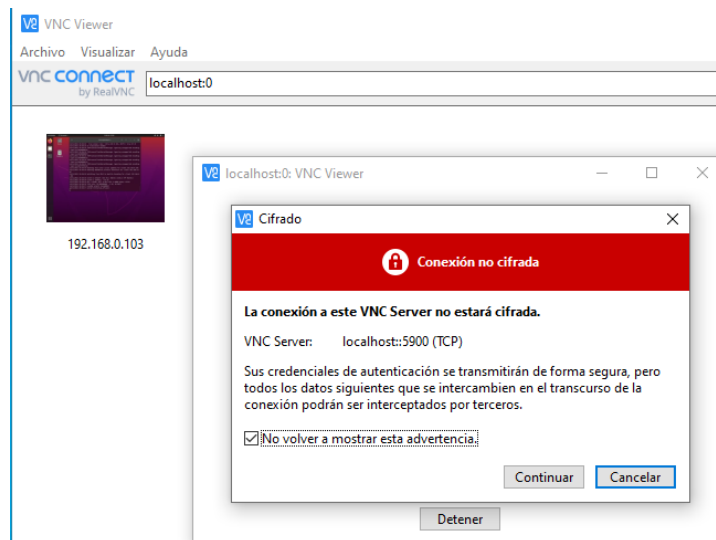


Tornem a sessions, desen els canvis i prenem *Open*. Introduïm l'usuari i contrasenya i ja tenim el túnel creat

```
lliurex@cldap21: ~  
login as: lliurex  
lliurex@192.168.0.103's password:  
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.11.0-38-generic x86_64)  
  
* Documentation:  https://help.ubuntu.com  
* Management:    https://landscape.canonical.com  
* Support:        https://ubuntu.com/advantage  
  
Se pueden aplicar 48 actualizaciones de forma inmediata.  
33 de estas son actualizaciones de seguridad estándares.  
Para ver estas actualizaciones adicionales ejecute: apt list --upgradable  
  
Your Hardware Enablement Stack (HWE) is supported until April 2025.  
Last login: Sun Nov 14 20:01:51 2021 from 192.168.0.105  
lliurex@cldap21:~$
```

Obrim VNCViewer. I en lloc de ficar la *IP:port*, com que tenim el túnel, ficarem *localhost:0*

Ens sortirà un altra vegada el missatge informant que les dades no estan encriptades, però realment si que ho estan.



1.3.4.5 Connexió per VNC emprant un túnel SSH a Linux

El procediment és molt paregut al que em vist a Windows

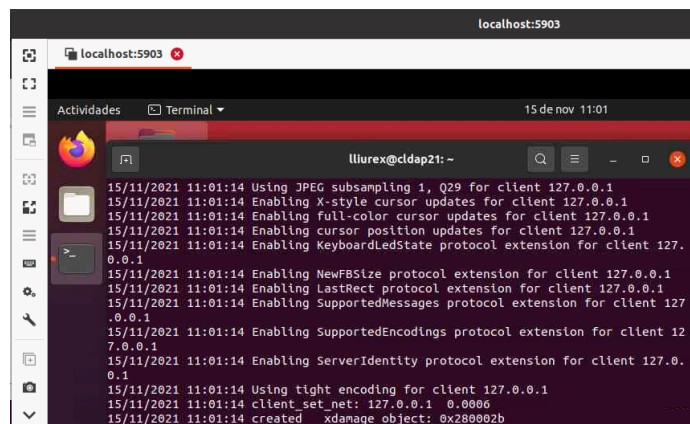
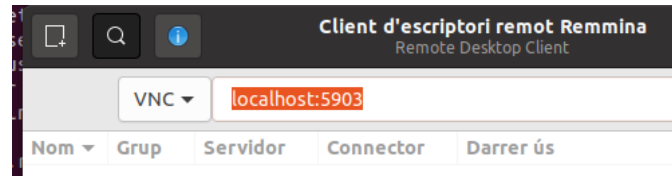
Al servidor cal iniciar el servidor x11vnc indicant el port que anem a ‘tunelitzar’. Hem de triar **un port que estigui lliure**.

```
$ x11vnc -rfbport 5903 -shared
```

Al Linux que farà de **client** crearem el túnel

```
$ ssh -L 5903:localhost:5903 lliurex@192.168.0.103
```

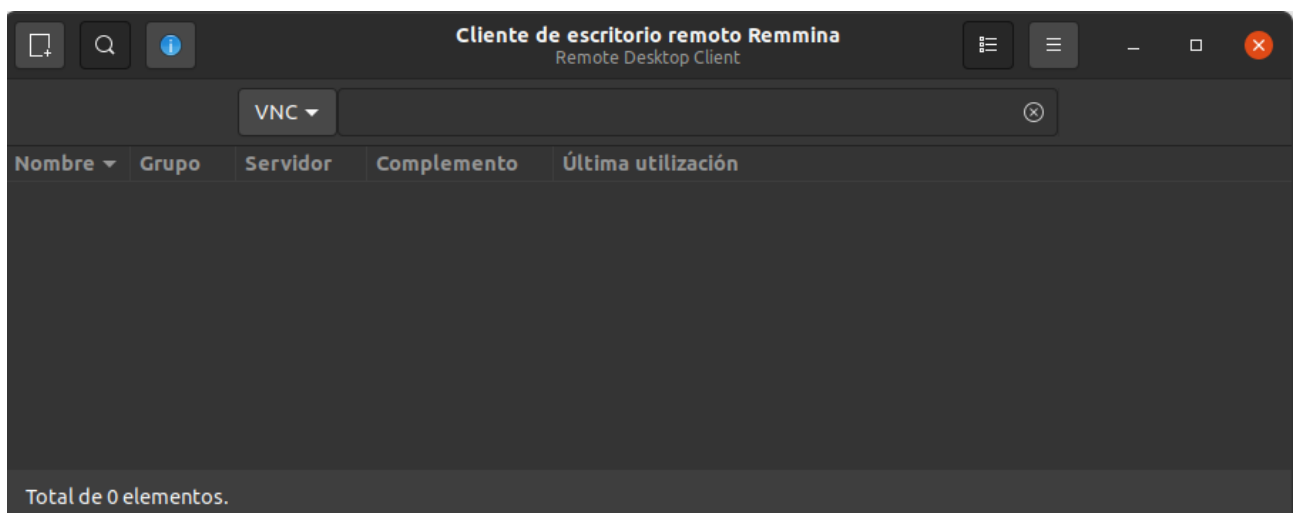
Obrim el nostre client VNC, al meu cas Remmina, i farem la connexió



1.3.4.6 *Nous clients d'escriptori remot multiprotocol*

La tendència actual és que els clients d'accés a escriptoris remots funcionin amb diferents protocols i que per tant es pugui accedir a diferents plataformes i sistemes operatius. És el cas del projecte Remmina, desenvolupat en GTK+ (i per tant compatible amb els escriptoris GNOME i XFCE), que permet fins i tot mantenir diferents connexions de manera simultània.

Remmina és un client d'escriptori remot que està incorporat per defecte en les darreres versions de Linux Debian i que és compatible amb els protocols RDP (Microsoft Windows), RFB (plataforma VNC), XDMCP (protocol de gestió de finestres X), NX i SSH, entre d'altres.



1.3.5 Gestió remota mitjançant una aplicació gràfica local

Una aproximació diferent a l'administració d'equips remots consisteix a executar una aplicació local amb interfície gràfica que permeti gestionar l'estació remota.

Aquesta solució s'utilitza amb molta freqüència per administrar petits equips de xarxa domèstics, als quals el fabricant de l'equip proporciona un programari de gestió remota. En aquest cas el fabricant del dispositiu implementarà una interfície web d'administració que permetrà la gestió del dispositiu per a tots els seus usuaris. Avui dia la major part de dispositius de xarxa, com ara els routers, les impressores o els punts d'accés, incorporen una interfície web per a la seva administració.

Fins i tot és possible administrar servidors complets a través del web, fent servir eines com **Webmin**.

1.3.5.1 Introducció a Webmin

Webmin és una aplicació que permet administrar un servidor Unix/Linux de manera remota mitjançant qualsevol navegador web modern.

Es tracta de programari lliure amb llicència GPL, escrit en Perl i que es pot executar en sistemes operatius com GNU/Linux, OpenSolaris i FreeBSD.

Un cop instal·lat i en funcionament, Webmin proporciona una interfície web (normalment accessible al port 10000) per administrar un gran ventall d'opcions en l'equip i els seus serveis. Es tracta d'una aplicació modular que permet la gestió de diferents àrees en funció dels mòduls instal·lats. En la distribució estàndard s'hi inclouen mòduls que permeten gestionar des del sistema operatiu (usuaris, quotes de disc, processos) fins als serveis de xarxa (Apache, Bind, Samba, etc.). I tot, amb una interfície web molt intuïtiva per a l'usuari.

Perl

És un llenguatge de programació dinàmic, dissenyat per Larry Wall i disponible en moltes plataformes diferents. Es tracta d'un llenguatge de propòsit general amb el qual es poden implementar tot tipus d'aplicacions. Però la construcció d'aplicacions web dinàmiques i l'administració de sistemes són dues especialitats que no es poden discutir. En molts sentits Python és una alternativa més moderna a Perl, malgrat que no treu cap vigència a Perl. Tots dos són eines excel·lents.

1.3.5.2 Instal·lació de Webmin

A la pàgina de Webmin (www.webmin.com) es pot descarregar l'aplicació en diferents formats inclòs Debian (.deb). Per nosaltres: [Webmin Debian](#).

La instal·lació es fa automàticament al directori /usr/share/webmin i, un cop instal·lat, la interfície web estarà disponible al port TCP 10000. L'usuari administrador de Webmin serà el superusuari (*root*) amb la seva contrasenya.

Webmin pot funcionar mitjançant HTTP i HTTPS; després de la instal·lació inicial, només estarà disponible l'accés mitjançant HTTPS.

Podem treballar en mode local fent servir Webmin per administrar el propi equip. Per fer això només caldrà obrir un navegador web i dirigir-lo a l'adreça `https://localhost:10000` (o bé `https://127.0.0.1:10000`). Si volem accedir de forma remota des de un altre ordinador connectat en xarxa cal substituir *localhost* per l'adreça IP de l'equip que volem administrar, sempre, lògicament, que aquest tingui iniciat el servei Webmin.

A la [figura.19](#) podem observar la pàgina d'inici de sessió per a Webmin que s'està executant de manera local, després d'haver-nos identificat com a superusuari.

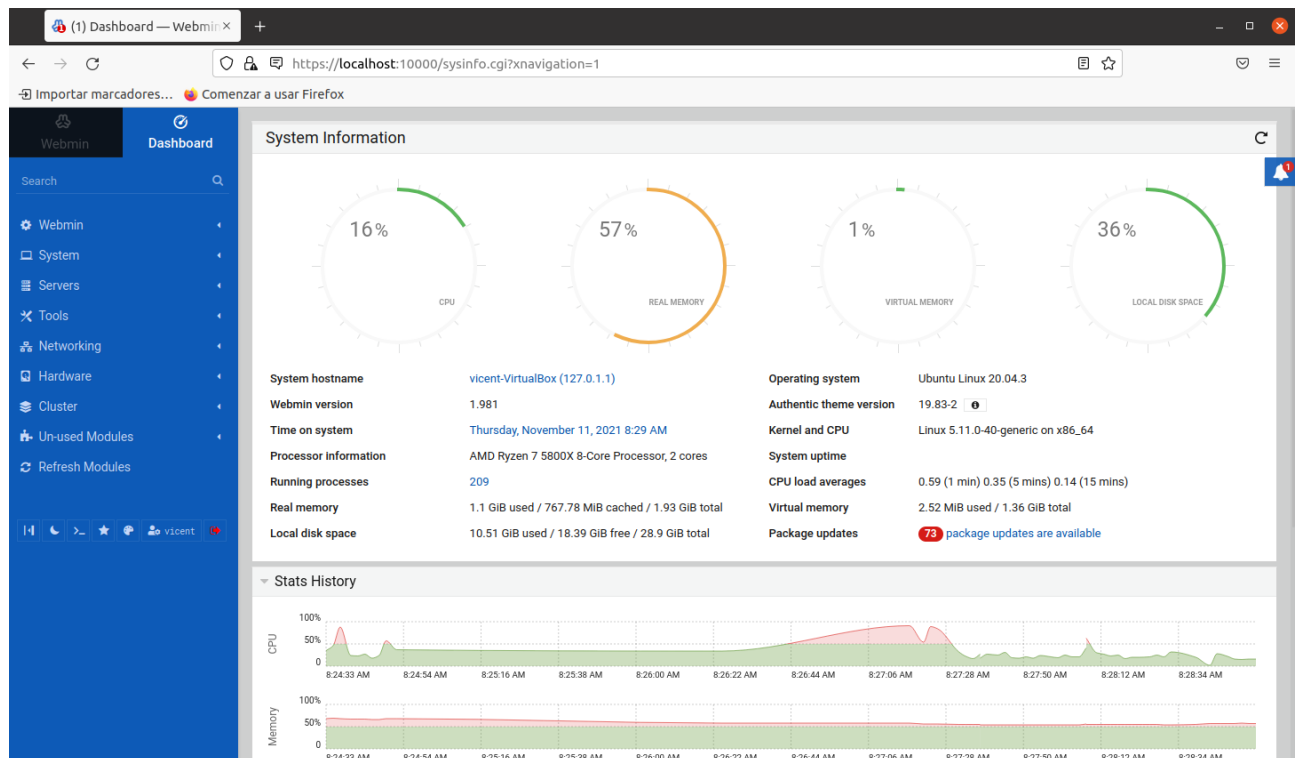
HTTP

i

HTTPS:

La diferència entre aquests dos protocols és que el segon consisteix a fer servir HTTP combinat amb SSL (secure socket layer), per tal d'encriptar tota la informació transmesa.

Figura Pàgina d'entrada a Webmin



Un cop instal·lat Webmin, el fitxer `/etc/init.d/webmin` ens permetrà engegar i aturar l'aplicació com qualsevol altre servei (directives `start`, `stop`, `restart`).

1.3.5.3 Funcions de Webmin

Webmin és una aplicació modular que aporta una gran funcionalitat. Les diferents opcions s'exposen mitjançant una interfície clara i intuïtiva disponible en diferents idiomes.

Les opcions de la interfície estan agrupades dins de pestanyes generals, que podeu trobar a la taula.

Taula: Opcions de Webmin

Pestanya	Funcions
Webmin	Configuració general de l'aplicació: idioma, aspecte, usuaris i fitxers de registre de Webmin. Permet gestionar configuracions exportant un fitxer i tornant-lo a carregar.
Sistema	Administració del sistema informàtic en el qual s'executa Webmin. Des del control d'usuaris, les quotes de disc, les tasques periòdiques i la gestió de paquets fins a les còpies de seguretat.
Servidors	Mòduls per administrar els diferents serveis que es poden executar en l'equip. Alguns dels més corrents són: Samba, Postfix, OpenSSH, Squid, etc.
Altres	Permet gestionar fitxers, comprovar el funcionament dels serveis, instal·lar/desinstal·lar programari i fins i tot establir connexions Telnet/SSH des del mateix navegador al servidor.
Xarxa	Configurar tots els paràmetres de xarxa de l'equip i els seus adaptadors. A més, permet configurar, entre d'altres, el tallafocs o la monitorització dels adaptadors de xarxa.
Maquinari	Permet gestionar tot el maquinari: impressores, particions, RAID, gestor d'engegada GRUB, etc.

1.4 Tendències actuals de l'accés i administració remota d'equips

El món globalitzat i intercomunicat mitjançant Internet ha fet possible pensar en els programes d'aplicació com a serveis disponibles no en el propi ordinador sinó en el “núvol”, entès com a metàfora de la xarxa global.

Quan es parla de computació en el núvol es refereix a l'accés a recursos i serveis des de qualsevol lloc, fer servir programari específic d'aplicació de forma remota mitjançant la interfície del mateix navegador d'Internet i emmagatzemant les dades en servidors externs.

Tot això fa que l'accés i l'administració remota prenguin una nova dimensió on els recursos, programes, potència de càlcul, sistemes d'emmagatzematge, etc. poden estar distribuïts i intercomunicats en qualsevol lloc de la xarxa.

Dos exemples que il·lustren aquesta tendència:

El projecte eyeOS va néixer de la mà d'un grup de joves programadors d'Olesa de Montserrat (Baix Llobregat) l'agost de 2005 i ha obtingut un gran èxit i reconeixement internacional.

Chrome Remote Desktop: Es tracta d'una extensió de Chrome, el navegador de Google que permet accedir a l'escriptori des de qualsevol lloc. L'usuari pot connectar-se i gestionar diferents equips, sigui quina sigui la seva plataforma i el seu sistema operatiu, mentre disposin de navegador Chrome. Per tant, no cal la instal·lació de cap programa addicional.



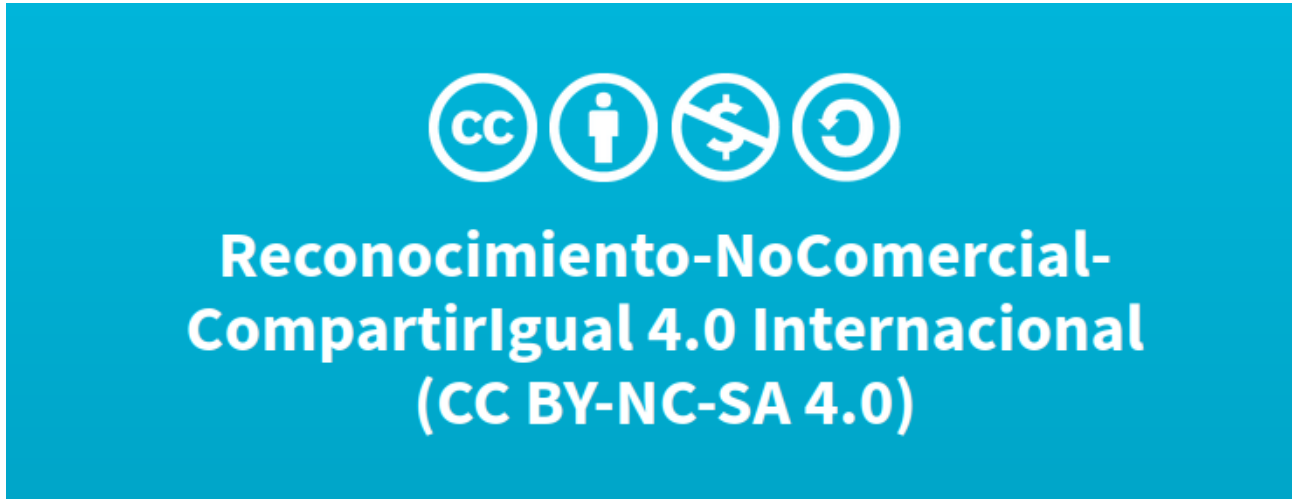
eyeOS és un escriptori virtual, multiplataforma i de codi lliure que inclou tota la estructura d'un sistema operatiu i algunes aplicacions ofimàtiques. El continguts, arxius i programes que conformen aquest sistema operatiu es troben en servidors privats o públics a la xarxa i es pot accedir als seus recursos des de qualsevol navegador web modern.

2 BIBLIOGRAFIA

Temari original baix llicència Creative Commons Reconeixement-NoComercial-CompartirIgual 4.0 Internacional:

Javier Martínez (IES María Enríquez)

Vicent Benavent



Modificacions per Vicent Benavent:

- Afegides característiques dels fitxers de configuració de SSH
- Ampliada la taula d'opcions disponibles de configuració
- Afegit fitxer de configuració ssh per un usuari específica i tipus de paràmetres
- Actualitzat servei sftp i captures de funcionament a Ubuntu 20.04
- Actualitzat execució de serveis i aplicacions de forma remota
- Afegit funcionament del servei d'escriptori remot mitjançant VNC (x11vnc)
- Actualitzada la versió de webmin