

---

# Servidor de Mail i CVS



Marc Serra <nandelbosc@s3os.net>

Copyright © 2005 Solucions i Serveis amb Sistemes Open Source (www.s3os.net)

Dins la peixera no pot faltar un peix encarregat de fer-nos arribar el correu electrònic...  
En aquest cas serà el host amb el nom de BACALLÀ.

A més del correu del nostre domini (s3os.net) tindrem la possibilitat de crear dominis virtuals, d'aquesta manera podem oferir als nostres clients de pàgines web (amb el corresponen domini) comptes de correu virtuals. Virtuals? Doncs si, en cap moment tindrà un usuari real al nostre sistema!

Aprofitant que farà "poca" feina, hi afegirem un CVS (Control Version System), que no és res més que un control de versions concurrent destinat, principalment, als programadors per poder tenir diferents branques d'un mateix codi, és a dir, una mica d'organització ;)

## Table of Contents

Instal·lació de postfix .....	2
Configuració .....	2
Provant postfix: .....	4
Courier-IMAP .....	4
Instal·lació i configuració .....	5
Integrar cyrus-sasl i SSL a Postfix i Apache .....	7
Cyrus-sasl .....	7
Creem certificats SSL .....	7
SSL a Apache i Postfix .....	8
MySQL .....	9
Instal·lació i configuració .....	9
Apache i PHPMyAdmin .....	10
Instal·lació dels paquets necessaris .....	10
Configuracions .....	11
L'usuari vmail .....	12
Autenticació MySQL i dominis virtuals .....	12
Autenticació .....	12
Postfix .....	12
Canvi de permisos .....	14
Recarreguem postfix .....	14
Squirrelmail .....	14
L'instal·lem... .....	14

El configurem...	14
RESUM del Virtual Mail System .....	14
Serveis necessaris .....	15
Afegir usuaris .....	15
CVS .....	16
Instal·lació .....	16
Definim el repositori, al servidor .....	16
Exemples d'utilització, al client .....	16

## Important

Aquest document està escrit per a postfix-2.1.x. Si utilitzes <postfix-2 moltes de les variables d'aquest document seràn diferents. És recomanable que t'actualitzis. Sinó pots consultar la documentació de postfix paral·lelament amb aquest document per a veure les diferències.

## Important

Aquest document te en compte que el sistema on instalem el soft és un Snod amb configuració SnodServer. Amb configuracions SnodDesktop o amb gentoos tingueu en compte que els següents USE flags (Relatiu a postfix, o sigui que postfix te en compte) estan actius en la configuració de SnodServer:

```

Configuració de USE Flags per a postfix
+ipv6 +ldap -mailwrapper -mbox +mysql +pam +postgres -sasl (-selinux) \\  

+ssl -vda

```

# Instal·lació de postfix

Postfix és un “mailer”, una alternativa a “Sendmail”, molt més fàcil de configurar.

Per defecte amb, Gentoo i Snod ens vé instal·lat el paquet mail-mta/ssmtp. Aquest entra en conflicte amb el paquet de postfix per això l'hem de desinstal·lar.

```
# emerge -p postfix
```

These are the packages that I would merge, in order:

```
Calculating dependencies ...done!
[blocks B      ] mail-mta/ssmtp (from pkg mail-mta/postfix-2.1.5-r1)
[ebuild N     ] mail-mta/postfix-2.1.5-r1
```

```
# emerge unmerge ssmtp
```

Lavors ja podem instal·lar postfix:

```
# emerge -k postfix
```

# Configuració

Un cop instal·lat postfix, anem-lo a configurar.

En el següent exemple configurarem postfix amb les següents característiques:

- Acceptant mails entrants de totes les màquines de la xarxa i de les possibles subxarxes connectades.

File: /etc/postfix/main.cf

```
myhostname = pop.s3os.net
mydomain = s3os.net
inet_interfaces = all
mydestination = $myhostname, localhost.$mydomain, localhost, $mydomain
mynetworks = 192.168.0.0/24, 127.0.0.0/8
default_destination_concurrency_limit = 10
```

Anem a comentar una mica el significat dels anteriors paràmetres:

- `inet_interfaces`: Indica les interfícies (Dispositius) de red que s'utilitzaran per a rebre correu.
- `mydestination`: Especifica la llista de dominis que el servidor de correu considerarà que ell n'és el destinatari final
- `mynetworks`:  
Indica una llista de clients SMTP "trusted". O sigui que hi indicarem una llista de hosts amb permisos per a enviar mails per a SMTP. Per exemple 192.168.0.0/24 indica totes les màquines de la subxarxa 192.168.0. El 24 indica que els primers 24 bits són la subred.
- `default_destination_concurrency_limit`: Nombre per defecte d'enviaments en paral·lel al mateix destí.

1

Ara habilitem el mode verbose per a poder depurar postfix. Editem el fitxer /etc/postfix/master.cf i simplement afegim "-v" al final de la línia per el servei smtp

```
# service type private unpriv chroot wakeup maxproc command + args
#          (yes)   (yes)   (yes)   (never) (50)
#
=====
smtp      inet  n       -       n       -       -       smtpd -v
```

Ara hem de generar la base de dades de 'aliases'. Si volem afegir un àlies nou, ara és el moment. Ho podem fer editant el fitxer /etc/mail/aliases. Per exemple jo he afegit el de 'admin'.

```
# Basic system aliases -- these MUST be present.
MAILER-DAEMON:      postmaster
postmaster:         root

# General redirections for pseudo accounts.
adm:                root
```

```
bin:                root
daemon:            root
exim:              root
lp:                root
mail:              root
named:             root
nobody:            root
postfix:           root

# Well-known aliases -- these should be filled in!
# root:
# operator:

# Standard RFC2142 aliases
abuse:             postmaster
ftp:               root
hostmaster:        root
news:              usenet
noc:               root
security:          root
usenet:            root
uucp:              root
webmaster:         root
www:               webmaster

# trap decode to catch security attacks
# decode:          /dev/null

admin:             root@domini.com
```

Llavors actualitzem la base de dades de aliases. Això ho hem de fer el primer cop que instal·lem postfix i cada cop que modifiquem els aliases.

```
/usr/bin/newaliases
```

Ens creará el fitxer `/etc/mail/aliases.db`

## Provant postfix:

Engeguem el servei:

```
# /etc/init.d/postfix start
```

Ara engeguem un client de mail per a consola, per exemple 'mutt', i enviem-nos un mail a nosaltres mateixos. Podem enviar un mail a `root@s3os.net`. Seguidament comprovem si postfix despatxa correus a la màquina local. També podem comprovar el funcionament dels nous aliases si és que n'hem afegit.

Si tot ha anat bé rebrem els mails que ens autoenviem. Sinó podem depurar els problemes.

## Courier-IMAP

Courier-IMAP [<http://www.courier-mta.org/imap/>] és un servidor de IMAP [<http://www.imap.org/>] i de

POP3 que són protocols d'internet per a accedir a correus electrònics. IMAP a diferència de POP permet que els missatges llegits resideixin en el servidor, s'entén que podem manipular i llegir mails d'un servidor compartit des de vàris clients sense haver de tornar-los a transferir en el servidor.

## Instal.lació i configuració

Instal.lar-lo és fàcil :)

```
# emerge courier-imap
```

Si volem la funcionalitat de servidor segur amb SSL haurem de crear certificats.

Per a fer-ho editarem el fitxer `/etc/courier-imap/pop3d.cnf` i el fitxer `/etc/courier-imap/imapd.cnf` per la configuració del servei POP3 i IMAP respectivament. Canviant els següents paràmetres:

```
[ req_dn ]
C=ES
ST=GI
L=Spain
O=Courier Mail Server
OU=Automatically-generated POP3 SSL key
CN=localhost
emailAddress=postmaster@s3os.net
```

Això són dades que apareixeràn als corresponents certificats. Per a crear-los executem:

```
# mkpop3dcert
# mkimapdcert
```

Ja podem engegar el serveis:

```
# /etc/init.d/courier-imapd start
# /etc/init.d/courier-imapd-ssl start
# /etc/init.d/courier-pop3d start
# /etc/init.d/courier-pop3d-ssl start
```

Óbviament no cal que engegum cap servei que no necessitem. Per exemple si només volem servir imap sense ssl, només cal engegar `courier-imapd`

A continuació anem a provar si podem llegir correu des de un client remot. Per a fer-ho jo he creat un usuari de proves i li he enviat un mail des de l'administrador utilitzant mutt.

Cal que ens assegurem que les connexions per al port corresponent a cada servei estiguin obertes. Per exemple si tenim un enrutador que ens enllaça la nostre LAN a internet és possible que haguem de confirmar que els ports de cada servei estan oberts i s'enruten cap al servidor de correu.

Els ports per a cada servei són:

```
# useradd -m -g users mailtest
# passwd mailtest
New UNIX password:
Retype new UNIX password:
passwd: password updated successfully
```

Llavors he configurat el kmail perque accedeixi al meu servidor.

**Add Account - Kontact**

**Account Type: IMAP Account**

General Security Filtering

Name: mailtest

Login: mailtest

Password: \*\*\*\*\*

Host: pop.s3os.net

Port: 143

Prefix to folders:

Store IMAP password in configuration file

Automatically compact folders (expunges deleted messages)

Show hidden folders

Show only subscribed folders

Load attachments on demand

List only open folders

Exclude from "Check Mail"

Enable interval mail checking

Check interval: 1 min

Trash folder: trash

Help OK Cancel

Configuració de kmail per provar el servidor IMAP

Si tot ha anat bé, arribat a aquest punt, ja tenim un servidor de correu al que li podem consultar els correus remotament i que accepta correus entrants. Els usuaris del servidor de correu son els usuaris locals

del servidor i els correus es guarden al directori `.maildir` del home de cada usuari. Ténen permís per a utilitzar SMTP, o sigui per a enviar correus utilitzant aquest servidor de correu, tots els usuaris de la lan Local (Recordeu `mynetworks = 192.168.0.0/24, 127.0.0.0/8` al configurar postfix).

Si per el contrari, la cosa no funcionan. Consulteu els logs a `/var/log/mail/current` segurament allà trovareu una bona pista per a solucionar el problema.

## Integrar cyrus-sasl i SSL a Postfix i Apache

Si volem assegurar el servidor de correu és important encriptar les contrasenyes, aquesta és la funció de SSL (Secure Socket Layer), d'aquesta manera cap 'mala persona' que estigui escoltant connexions a la nostra xarxa podrà accedir als nostres comptes de correu. Per altra banda, el cyrus-sasl ens ajudarà amb la comucació contra la base de dades MySQL i la gestió d'usuaris, tot de forma segura.

## Cyrus-sasl

### Instal·lació

Com sempre...

```
# emerge cyrus-sasl
```

### Configuració i arrencada

Editem un parell de fitxers i arrenquem el servei

```
# nano -w /etc/sasl2/smtpd.conf
    mech_list: PLAIN LOGIN
    pwcheck_method: saslauthd

# nano -w /etc/conf.d/saslauthd
    SASLAUTHD_OPTS="{SASLAUTH_MECH} -a pam -r"

# /etc/init.d/saslauthd start
```

## Creem certificats SSL

Per crear els certificats modifiquem els següents fitxers:

```
# cd /etc/ssl/
# nano -w openssl.cnf
    countryName_default
    stateOrProvinceName_default
    localityName_default
    0.organizationName_default
    commonName_default
    emailAddress_default.
(si les variables no hi són, les afegim)
```

Editem el fitxer `CA.pl` i busquem les cadenes `"#create certificate"` i `"#create certificate request"`, per deixar-les com segueix...

```
# cd misc
# nano -w CA.pl
# create a certificate
system ("$REQ -new -nodes -x509 -keyout newreq.pem -out newreq.pem \\  
$DAYS");
# create a certificate request
system ("$REQ -new -nodes -keyout newreq.pem -out newreq.pem $DAYS");

# ./CA.pl -newca
# ./CA.pl -newreq
# ./CA.pl -sign
# cp newcert.pem /etc/postfix
# cp newreq.pem /etc/postfix
# cp demoCA/cacert.pem /etc/postfix
```

Ara fem el mateix per apache

```
# openssl req -new > new.cert.csr
# openssl rsa -in privkey.pem -out new.cert.key
# openssl x509 -in new.cert.csr -out new.cert.cert -req -signkey new.cert.key \\  
-days 365
```

De moment (i fins que instalem apache) deixem els certificats aquí.

## SSL a Apache i Postfix

### Addició

Editarem el fitxer de configuració de Postfix per afegir-li el suport SSL i sasl. Afegirem els següents paràmetres al final del fitxer, així seràn de bon trobar ;) Finalment el reiniciem...

```
# nano -w /etc/postfix/main.cf
smtpd_sasl_auth_enable = yes
smtpd_sasl2_auth_enable = yes
smtpd_sasl_security_options = noanonymous
broken_sasl_auth_clients = yes # per a Outlook
smtpd_sasl_local_domain = # millor deixar-lo en blanc
smtpd_recipient_restrictions = permit_sasl_authenticated, permit_mynetworks, \\  
reject_unauth_destination
smtpd_use_tls = yes
#smtpd_tls_auth_only = yes # comentada per testing, \\  
després la podem activar
smtpd_tls_key_file = /etc/postfix/newreq.pem
smtpd_tls_cert_file = /etc/postfix/newcert.pem
smtpd_tls_CAfile = /etc/postfix/cacert.pem
smtpd_tls_loglevel = 3
smtpd_tls_received_header = yes
smtpd_tls_session_cache_timeout = 3600s
tls_random_source = dev:/dev/urandom

# postfix reload
```

### Comprobació

```
# telnet localhost 25

Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
220 mail.domain.com ESMTTP Postfix
EHLO domain.com
250-mail.domain.com
250-PIPELINING
250-SIZE 10240000
250-VERFY
250-ETRN
250-STARTTLS
250-AUTH LOGIN PLAIN
250-AUTH=LOGIN PLAIN
250-XVERP
250 8BITMIME
^]
telnet> quit
```

SI NO FUNCIONA, ASSEGUREM-NOS QUE TENIM EL SERVEI "fam" (file alteration monitor) ENEGAT.

## MySQL

Ja podem instal·lar la base de dades, en aquest cas MySQL. Necessitarem el fitxer genericmailsql.sql [./genericmailsql.sql].

## Instal·lació i configuració

### Recuperar contrasenya de 'root' a mysqld

Aquest apartat és només per si estem actualitzant una base de dades anterior i no sabem la paraula de pas de l'usuari root. Per fer-ho seguirem els següents passos:

- Matem tots els processos mysqld
- Engueguem MySQL en mode segur
- Connectem al servidor MySQL
- Triem la BDD
- Actualitzem la paraula de pas
- Reiniciem privilegis
- Sortim de MySQL
- Parem MySQL
- Engueguem MySQL normalment

```
# kill `cat /var/lib/mysql/hostname.pid`
# /usr/bin/safe_mysqld --skip-grant-tables&
# /usr/bin/mysql

mysql> use mysql;
mysql> update user set password = password('.....') where user = \\
      'root' and host='localhost';
mysql> flush privileges;
mysql> quit

# /etc/init.d/mysql stop
# /etc/init.d/mysql start
```

Ara si, anem a instal·lar...

```
# emerge mysql
# /usr/bin/mysql_install_db
# /etc/init.d/mysql start
# mysqladmin -u root -p create mailsql
# mysql -u root -p mailsql < genericmailsql.sql
# mysql -u root -p mysql

mysql> GRANT SELECT,INSERT,UPDATE,DELETE
->      ON mailsql.*
->      TO mailsql@localhost
->      IDENTIFIED BY 'paraula.de.pas';
->      quit
```

Verifiquem que l'usuari 'mailsql' es pot connectar a la base de dades...

```
# mysql -u mailsql -p mailsql
```

La nova base de dades té els valors per defecte i taules configurades per dos dominis. Les següents taules estan incluídes:

- alias - àlies de mails locals i informació d'àlies mailman.
- relocated - mapes de les adreces d'usuari
- transport - transport de mail per defecte de tots els dominis
- users - tota l'inforació de les comptes d'usuari
- virtual - mapes dels dominis virtuals

## Apache i PHPMyAdmin

Instal·larem el servidor web Apache per poder comunicar-nos amb phpMyAdmin. phpMyAdmin és un interfcie web interactiva per a MySQL.

## Instal·lació dels paquets necessaris

També hi hem afegit el mòdul per a php, així ens mostrarà les pàgines fetes en aquest llenguatge

```
# emerge apache mod_php phpmyadmin
```

## Configuracions

### Afegir suport PHP a Apache

Afegim les següents línies als fitxers de configuració d'apache...

```
# nano /etc/apache2/conf/apache2.conf
    LoadModule php4_module      /usr/lib/apache2-extramodules/libphp4.so
# nano /etc/conf.d/apache
    APACHE2_OPTS="-D SSL -D PHP4"
```

També hem aprofitat per preparar apache per SSL, ara actualitzem la base de dades de MySQL:

```
# mysql -u root -p < /usr/share/webapps/phpmyadmin/2.6.2-r2/sqlscripts/\\
mysql/2.6.2-r2_create.sql
```

### Certificats SSL per a Apache

Abans hem deixat els certificats per apache "mig penjats"...

```
# cp /etc/ssl/misc/new.cert.cert /etc/apache/conf/ssl/
# cp /etc/ssl/misc/new.cert.key /etc/apache/conf/ssl/
# nano -w /etc/apache/conf/vhosts/ssl.default-vhost.conf
    ServerName host.domain.name
    ServerAdmin elteu@email.address
    SSLCertificateFile /etc/apache/conf/ssl/new.cert.cert
    SSLCertificateKeyFile /etc/apache/conf/ssl/new.cert.key

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

### Configurem phpMyAdmin

Fa un moment hem instal·lat aquesta aplicació (# emerge phpmyadmin), anem a configurar-la...

```
# nano -w /home/httpd/htdocs/phpmyadmin/config.inc.php

$config['Servers'][$i]['host'] = 'localhost';           //
    # MySQL hostname
$config['Servers'][$i]['controluser'] = 'mailsql';     //
    # MySQL informació de l'usuari de control (usuari de només lectura)
$config['Servers'][$i]['controlpass'] = 'paraula.de.pas'; //
    # a "mysql/user" y "mysql/db")
$config['Servers'][$i]['user'] = 'mailsql';           //
    # Usuari MySQL
$config['Servers'][$i]['password'] = 'paraula.de.pas'; //
    # Password
```

## L'usuari vmail

Hem de crear un usuari al cual correran els usuaris virtuals

```
# adduser -d /home/vmail -s /bin/false vmail
# uid=`cat /etc/passwd | grep vmail | cut -f 3 -d :`
# groupadd -g $uid vmail
# mkdir /home/vmail
# chown vmail. /home/vmail
```

Així els usuaris virtuals faràn servir la 'home' de l'usuari vmail i l'uid i gid de vmail. Per a comptes locals farem servir 'homes' d'usuari, el mateix per a l'uid i el gid.

## Autenticació MySQL i dominis virtuals

Ara intentarem que courier-imap i postfix utilitzin mysql ;)

### Autenticació

```
# nano -w /etc/courier-imap/authdaemonrc
    authmodulelist="authmysql authpam"

# nano -w /etc/courier-imap/authmysqlrc
    MYSQL_SERVER          localhost
    MYSQL_USERNAME        mailsql
    MYSQL_PASSWORD        paraula.de.pas
    MYSQL_DATABASE        mailsql
    MYSQL_USER_TABLE      users
    (assegurem-nos que la següent línia està comentada)
    #MYSQL_CRYPT_PWFIELD  crypt
    MYSQL_CLEAR_PWFIELD   clear
    MYSQL_UID_FIELD       uid
    MYSQL_GID_FIELD       gid
    MYSQL_LOGIN_FIELD     email
    MYSQL_HOME_FIELD      homedir
    MYSQL_NAME_FIELD      name
    MYSQL_MAILDIR_FIELD   maildir

# /etc/init.d/courier-authlib restart
# /etc/init.d/saslauthd restart
```

### Postfix

```
# nano -w /etc/postfix/mysql-aliases.cf
    user          = mailsql
    password      = paraula.de.pas
    dbname        = mailsql
    table         = alias
    select_field  = destination
    where_field   = alias
    hosts        = unix:/var/run/mysqld/mysqld.sock

# nano -w /etc/postfix/mysql-relocated.cf
    user          = mailsql
    password      = paraula.de.pas
```

```

dbname          = mailsql
table           = relocated
select_field    = destination
where_field     = email
hosts           = unix:/var/run/mysqld/mysqld.sock

# nano -w /etc/postfix/mysql-transport.cf
user            = mailsql
password        = paraula.de.pas
dbname          = mailsql
table           = transport
select_field    = destination
where_field     = domain
hosts           = unix:/var/run/mysqld/mysqld.sock

# nano -w /etc/postfix/mysql-virtual-gid.cf
user            = mailsql
password        = paraula.de.pas
dbname          = mailsql
table           = users
select_field    = gid
where_field     = email
additional_conditions = and postfix = 'y'
hosts           = unix:/var/run/mysqld/mysqld.sock

# nano -w /etc/postfix/mysql-virtual-maps.cf
user            = mailsql
password        = paraula.de.pas
dbname          = mailsql
table           = users
select_field    = maildir
where_field     = email
additional_conditions = and postfix = 'y'
hosts           = unix:/var/run/mysqld/mysqld.sock

# nano -w /etc/postfix/mysql-virtual-uid.cf
user            = mailsql
password        = paraula.de.pas
dbname          = mailsql
table           = users
select_field    = uid
where_field     = email
additional_conditions = and postfix = 'y'
hosts           = unix:/var/run/mysqld/mysqld.sock

# nano -w /etc/postfix/mysql-virtual.cf
user            = mailsql
password        = paraula.de.pas
dbname          = mailsql
table           = virtual
select_field    = destination
where_field     = email
hosts           = unix:/var/run/mysqld/mysqld.sock

# nano -w /etc/postfix/main.cf
alias_maps      = mysql:/etc/postfix/mysql-aliases.cf
relocated_maps  = mysql:/etc/postfix/mysql-relocated.cf

local_transport = local
local_recipient_maps = $alias_maps $virtual_mailbox_maps \\
    unix:passwd.byname

virtual_transport = virtual

```

```
virtual_mailbox_domains = virt-bar.com, un.altre.domini

virtual_minimum_uid = 1000
virtual_gid_maps = static:valor.gid.usuari.vmail
virtual_mailbox_maps = mysql:/etc/postfix/mysql-virtual-maps.cf
virtual_alias_maps = mysql:/etc/postfix/mysql-virtual.cf
virtual_uid_maps = static:valor.uid.usuari.vmail
virtual_mailbox_base = /
#virtual_mailbox_limit =
```

## Canvi de permisos

Canviem els permisos a uns quants fitxers... (per seguretat)

```
# chmod 640 /etc/postfix/mysql-*.cf
# chgrp postfix /etc/postfix/mysql-*.cf
```

## Recarreguem postfix

```
# postfix reload
```

Arribats a aquest punt ja tenim el servidor preparat. Els usuaris ja es poden autenticar contra la base de dades MySQL fent servir la direcció de mail completa per a POP3, SMTP i IMAP!

## Squirrelmail

Squirrelmail és una aplicació que ens permet revisar el correu amb qualsevol navegador web.

## L'instal·lem...

```
# emerge squirrelmail
```

## El configurem...

```
# webapp-config -I -h localhost -d /mail squirrelmail 1.4.3a-r2
# cd /var/www/localhost/htdocs/mail/config
# perl ./conf.pl
```

Canviem la configuració de l'organització, servidor i carpeta per squirrelmail. Ja ens podem loggejar amb la direcció de correu completa.

## RESUM del Virtual Mail System

Arribats a aquest punt tot hauria de funcionar perfectament, però... Quins serveis necessitem engegar? Com afegim usuaris?

## Serveis necessaris

Amb aquests dos serveis és suficient: courier-imapd i postfix. Per engegar:

```
# /etc/init.d/courier-imapd start
```

Parar:

```
# /etc/init.d/courier-imapd stop
```

Reiniciar:

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

```
    o (només per a postfix)
```

```
# postfix reload
```

Parar un procés 'zombie':

```
# /etc/init.d/postfix zap
```

## Afegir usuaris

### Usuaris virtuals

Usuaris sense compte al nostre sistema, utilitzaran l'usuari 'mailsql' creat anteriorment

```
# nano /etc/postfix/main.cf
```

Busquem l'apartat 'virtual\_mailbox\_domains' i hi afegim el nostre domini. Ara reiniciem postfix:

```
# postfix reload
```

Amb qualsevol navegador web apuntem a: <http://localhost/phpmyadmin> (o <http://bacalla/phpmyadmin>), accedim a la Base de dades 'mailsql', dins aquesta la taula 'transport' i comprovem que hi ha el domini desitjat (si no hi és l'afegim). Ara busquem la taula 'users', clic a 'add new row' (o afegir nova fila, si ho tenim en català) i utilitzem el gid, uid i directori personal de l'usuari mailsql. Finalment enviem un mail al nou usuari, d'aquesta manera crearem el seu directori 'mailbox'.

Per loggejar-nos amb aquest usuaris a squirrelmail, per exemple, hem de fer-ho amb la direcció de correu completa: `usuari@domini.xxx`

### Usuaris locals

Usuaris amb compte a la nostra màquina (amb directori personal, shell, ...)

```
# adduser usuari
```

Si, simplement afegint un usuari normalment ja tindrà compte de correu. Podem passar-li els paràmetres que volguem (tipus de shell, directori personal,...)

Aquests usuaris no han de posar el domini a continuació del nom.

## CVS

Primer instal·larem l'aplicació, després només haurem de definir quin directori serà el repositori. Anem-ho a veure...

## Instal·lació

```
# emerge cvs
```

## Definim el repositori, al servidor

En aquest cas '/home/cvs/' serà el directori arrel on actuarà el control de versions

```
# mkdir /home/cvs/  
# mkdir /home/cvs/directori.programa.1  
# cvs -d /home/cvs/directori.programa.1 init
```

La feina al servidor ja està acabada.

## Exemples d'utilització, al client

Farem servir el protocol SSH, quines possibilitats tenim...

## Definim el protocol

```
# export cvs_rsh="ssh"
```

## Repositori local i l'opció 'import'

La primera inserció de fitxers al servidor hem de fer-la de la següent manera...

```
# mkdir directori.programa.1  
# cd directori.programa.1  
# cvs -d nandelbosc@bacalla.s3os.net:/home/cvs/directori.programa.1 import \<\  
-m "first import" directori.programa.1 vendor nom.branca  
# cd ..  
# rm -f modulename
```

Què hem fet...

- Hem creat el directori on hi haurà els fitxers de configuració

- Entrem al directori
- Fem el primer enviament de dades al servidor, l'opció '-m' és per al log de canvis del cvs, 'nom.branca' és per si tenim pensat crear diferents branques del programa (un clar exemple: els desenvolupadors d'Apache, desenvolupen la branca 1.x i 2.x)
- Sortim del directori
- El borrem, ja tenim la còpia al repositori

## L'opció 'checkout'

Per recuperar els fitxers enviats al servidor

```
# cvs -d nandelbosc@bacalla.s3os.net:/home/cvs/directori.programa.1 co \  
    directori.programa.1  
# cd directori.programa.1
```

- Quin directori volem recuperar (/home/cvs/directori.programa.1) i a quin directori el volem desar (directori.programa.1)
- Entrem al directori
- Podem tornar a programar

## Actualitzar fitxers ja existents al servidor, 'commit'

```
# cvs -d nandelbosc@isildur.s3os.net:/home/cvs/s3os commit -m "first change"
```

Aquí no hi ha gaire secret, només canvia l'opció 'commit' (podem traduir-la a actualitza), el missatge, una vegada més, és per a tenir un registre al log dels canvis fets.