

# Sistema de videovigilància basat en programari lliure i material reciclat

Agustí Fontquerni

Escola Universitària Salesiana de Sarrià (EUSS) [1]  
Passeig Sant Joan Bosco, 74  
08017 Barcelona  
afontquerni@euss.es

Juny de 2007

## Resum

En aquesta ponència es presenta un projecte de videovigilància basat en programari lliure, implementat a partir de la integració de projectes existents i components electrònics reutilitzats. És un sistema actualment en producció en una instal·lació municipal de recollida selectiva o deixalleria.

El sistema operatiu instal·lat és el GNU/Linux basat en una distribució Debian. Específicament s'han instal·lat i configurat diversos projectes de programari lliure en el camp de la captura i el tractament d'imatges (motion, ffmpeg, mplayer, virtualdub, ffdshow-tryouts).

S'ha desenvolupat una aplicació totalment lliure sobre una base de dades MySQL i tecnologia web PHP sobre servidor Apache. Aquesta permet gestionar tot el sistema de forma integrada, desassistida, autònoma, remota i amigable a l'usuari.

## 1 Introducció

En aquesta ponència es presenta un projecte de videovigilància basat en programari lliure, implementat a partir de la integració de projectes existents i components electrònics reutilitzats.

Per tal d'aconseguir el sistema de captura i processament d'imatges obtingudes per diverses càmeres, i alhora disminuir el cost de la instal·lació, s'ha optat per l'ús de diversos projectes lliures disponibles existents en la comunitat de programari lliure i d'elements provinents del mateix centre de recollida selectiva (deixalleria) objecte de monitorització. Així aquests elements es reutilitzen en aquest projecte enlloc de procedir al seu cicle estàndard de reciclatge, aprofitant-ne el seu potencial encara totalment funcional.

En els següents apartats es presenta la motivació del projecte, la descripció de les seves parts i components, l'aplicació integradora desenvolupada, una millora al codi font del projecte lliure Motion (nucli del sistema), i finalment les conclusions.

## **2 Antecedents**

L'origen del projecte rau en la necessitat de l'equipament municipal de la deixalleria de Navàs de detectar conductes incíviques (conductes de dipositar tot tipus de deixalles en l'accés a la instal·lació fora d'horari de servei). Addicionalment detectar intrusions en la instal·lació normalment fora hores de servei i sovint de forma nocturna.

Una altra motivació fou l'aplicació pràctica i social, en el municipi on resideix l'autor, dels coneixements de noves tecnologies, programari lliure i comunicacions per tal realitzar el projecte final de carrera d'enginyeria industrial.

Aquests fets anteriors i la col·laboració en diversos temes sobre noves tecnologies, el coneixement de la problemàtica de la instal·lació municipal anteriorment comentada, junt amb la conscienciació que la millor forma d'eliminació de residus és la reutilització front a l'ús d'abocadors, porten al desenvolupament i integració d'aquesta aplicació de programari lliure.

## **3 Descripció de la instal·lació**

El nucli del sistema és un ordinador personal reutilitzat, que s'empra com a servidor local. Aquest gestiona i emmagatzema les imatges de forma continua procedent diverses càmeres CCD, algunes reciclades.

Les comunicacions entre els diversos elements són USB, TCP/IP amb xarxa cablejada i TCP/IP sense fils (wireless). S'ha utilitzat una connexió ADSL a Internet i una connexió a la xarxa sense fils metropolitana.

La ubicació de les diverses càmeres, s'ha situat per tal de cobrir la major àrea de la deixalleria i l'accés a la instal·lació.

La protecció ambiental és mínima, per tal de reduir costos i pel fet que els elements sensors a protegir no són molt valuosos ja que la majoria procedeixen del cicle de reciclatge del mateix centre o són de cost reduït.

A la figura 1 es presenten els diversos elements del sistema i la seva interconnexió. La relació dels diversos elements és:

1. Ordenador servidor
2. Router ADSL Zyxtel 660 HW (connexió Internet)
3. Router Wi-Fi Linksys WRT54GL (connexió xarxa metropolitana sense fils)
4. Càmera 1 Conceptronic CNETCAM (connexió cable Ethernet)
5. Càmera 2 Conceptronic C54NETCAM (connexió sense cable Wi-Fi)
6. Càmera 3 Veo Observer (connexió cable Ethernet)
7. Càmera 4 Logitech QuickCam Express USB (connexió cable USB)

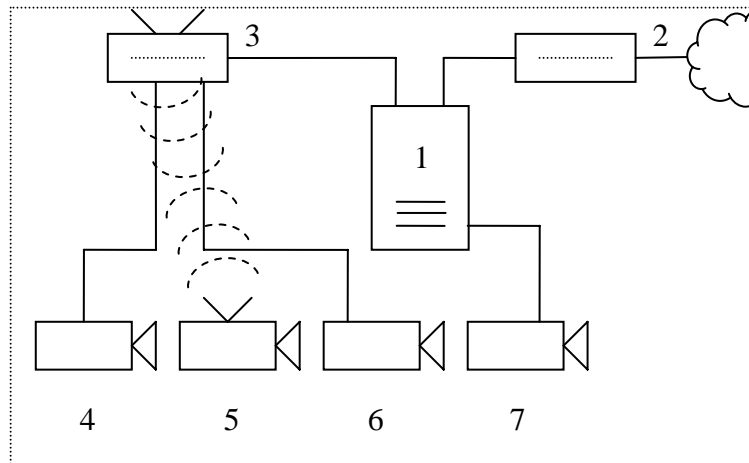


Figura 1: Esquema interconnexió equips

## 4 Descripció de l'aplicació

El programari instal·lat és el sistema operatiu GNU/Linux basat en una distribució Debian. Específicament s'han instal·lat i configurat diversos projectes de programari lliure en el camp de la captura i el tractament d'imatges (motion, ffmpeg, mplayer, virtualdub, ffdshow-tryouts)

S'ha desenvolupat una aplicació totalment lliure (documents i codi desenvolupat) sobre una base de dades MySQL i tecnologia web PHP sobre servidor Apache. Aquesta permet gestionar tot el sistema de forma integrada, desassistida, autònoma, remota i amigable al usuari.

Per integrar els diversos projectes i adaptar-los a les necessitats del projecte, s'han desenvolupat diverses tasques per tal de generar els vídeos horàriament, purga d'arxius antics i tasques de manteniment.

A continuació es detallen els diversos blocs, una breu descripció i les seves relacions (vegeu figura 2)

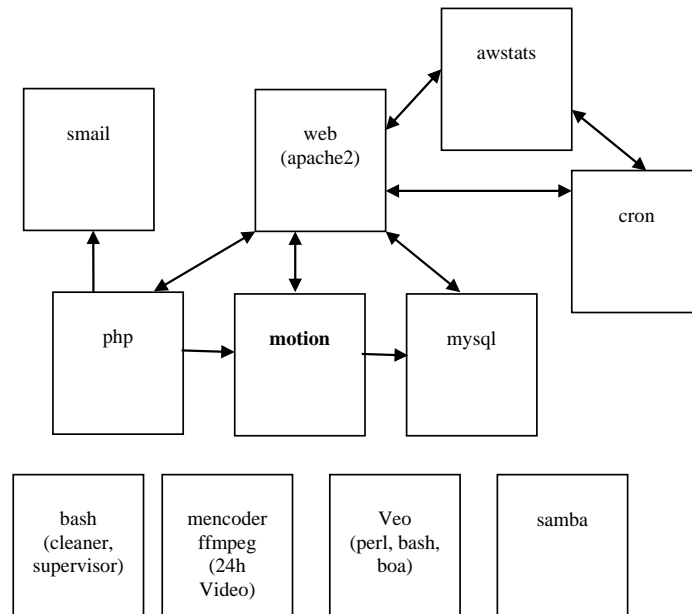


Figura 2: Diagrama de blocs de l'aplicació

**Motion:** bloc referent al projecte lliure Motion, utilitzat com a nucli d'adquisició.

Processa les imatges en format JPEG de les diverses càmeres per emmagatzemant-les contínuament, 24 hores, de forma estructurada (directori / hora). Disposa d'un algoritme de detecció d'alarmes a partir d'aquest processat i una conseqüent generació de vídeos associats amb una notificació per correu electrònic.

**Apache2:** bloc servidor web. Permet seguretat SSL (HTTPS) i autenticació d'usuari.

**Php:** intèrpret del llenguatge del mateix nom, PHP, originalment per crear web dinàmiques però que també permet llista d'ordres de comandes des de sistema.

**MySQL:** bloc servidor de base de dades amb suport per llenguatge SQL

**Smail:** servei de correu electrònic instal·lat per enviar notificacions d'alarmes.

**Awstats:** bloc de l'aplicació d'anàlisis dels registres de consultes del servidor web Apache2 i generació d'estadístiques i informació vària associada a la navegació web per part del usuaris.

**Cron:** bloc relatiu servei del sistema per executar tasques periòdiques.

**Bash:** bloc que simbolitza els diversos scripts o fitxers d'ordres de sistema per executar tasques de sistema per a l'aplicació desenvolupada (codificació de vídeos horaris de gravació contínua 24 hores, borrar d'imatges i vídeos antics, supervisió dels diversos processos en cas de caiguda, ... )

**Mencoder / ffmpeg:** bloc referent a la utilitat i la llibreria de codificació de vídeo a partir de les imatges individuals JPEG emmagatzemades en directoris per cada hora completa.

**Veo:** bloc que conté diversos elements que permeten accedir al Motion a les imatges de la càmera Veo Observer que no té un mecanisme estàndard d'accés.

**Samba:** servei de publicació d'arxius per xarxa de només lectura. Permet accedir directament als vídeos generats a equips Windows o GNU/Linux amb samba.

## 5 Interfície d'usuari

La interfície d'usuari és amb tecnologia web, accessible exclusivament de forma remota.

A la figura 3 o pàgina principal portal de l'aplicació, es pot observar la pantalla centre de comandament del sistema. Depenen de l'usuari correctament validat, pren diverses formes i funcionalitats actives segons el nivell d'accés atorgat a cada usuari (Administrador, Operador o Usuari Bàsic).

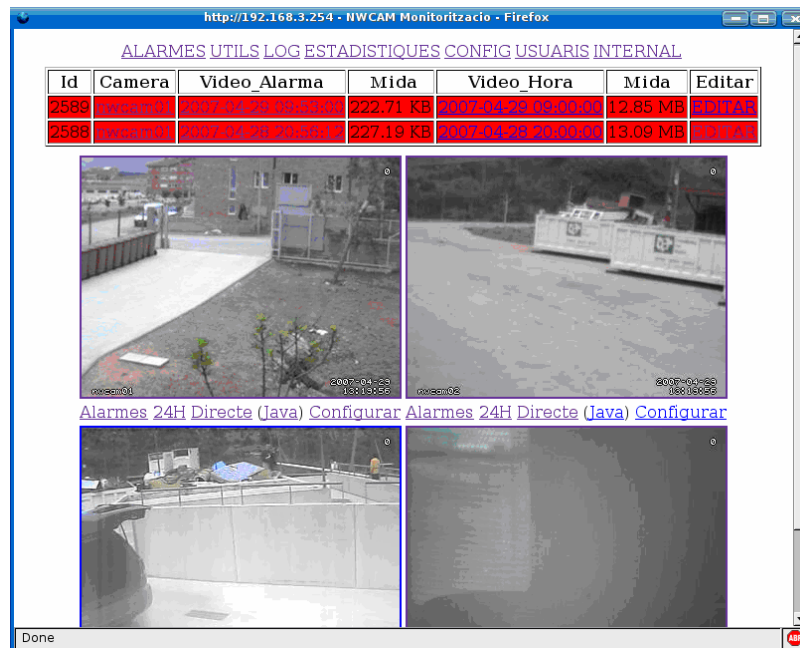


Figura 3: Pàgina principal amb alarmes amb nivell accés administrador

A la part superior i en una sola línia, hi ha un menú amb les opcions d'accés a les altres funcionalitats de l'aplicació: gestió alarmes ([ALARMES](#)), descàrrega utilitats client ([UTILS](#)), registre d'operacions realitzades en l'aplicació ([LOG](#)), dades estadístiques de visites i consultes de les pàgines web ([ESTADISTIQUES](#)), configuració dels paràmetres aplicació ([CONFIG](#)), gestió usuaris ([USUARIS](#)) i altres funcions només vàlides pel propi desenvolupament del projecte ([INTERNAL](#)).

Immediatament després, opcionalment, pot aparèixer una taula amb les alarmes no editades. En aquesta taula se'ns mostra el detall de l'alarma (càmera, hora, pes de l'arxiu) així com, si és disponible, tot el vídeo horari complet de la gravació contínua 24 hores; i finalment la possibilitat d'edició i qualificació de l'alarma.

A la part central (si no hi ha excessives alarmes pendents), es disposa d'una quadrícula amb una imatge recent de les 4 càmeres instal·lades. Aquesta pàgina té un refresc automàtic de les imatges cada 10 segons i així obtenir noves imatges més actualitzades. Sota cada imatge es disposa de diverses opcions: Editar alarmes específiques de la càmera ([Alarmes](#)), accés a l'arxiu de vídeos de gravació contínua 24 hores ([24H](#)), monitorització d'imatges de la càmera en temps real ([Directe \(Java\)](#)) i configuració horària de la activació de la detecció ([Configurar](#)).

La aplicació web sencera s'ha estructurat en diversos arxius especialitzats per a cada funcionalitat. A la figura 4 es pot observar les diverses pàgines que formen l'aplicació i seguidament una breu descripció de cada pàgina.

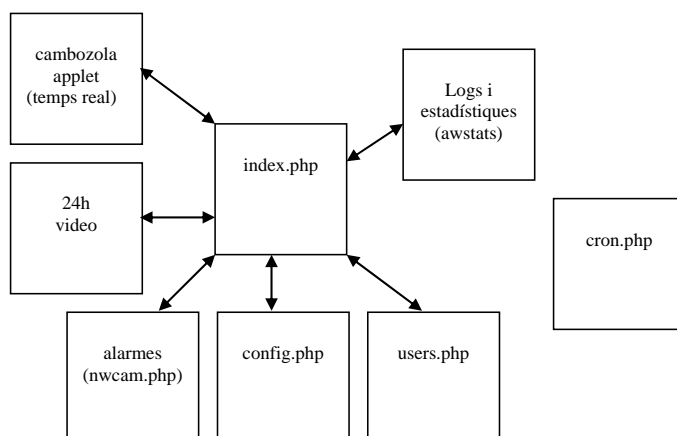


Figura 4: Diagrama de blocs de la aplicació

**index.php:** verifica nivell d'accés usuari, consulta les alarmes a la base de dades no editades i que cal mostrar. Crea la quadrícula d'imatges amb la última captura disponible de cada càmera a través del fitxer enllaç simbòlic "lastsnap.jpg" que crea el Motion. A partir d'aquesta pàgina web dinàmica en codi PHP s'accedeix a la resta de pàgines de la interfície d'usuari de l'aplicació.

**nwcam\_cambozola.php:** genera la pàgina de descàrrega i visualització en el navegador client de l'applet java cambozola disponible en l'arxiu cambozola.jar.

**awstats:** Amb el projecte awstats instal·lat (desenvolupat amb Perl) es disposa d'una completa informació estadística cronològica d'accés i navegació per les pàgines i imatges descarregades directament dels registres del servidor web Apache2.

**24h vídeo:** Vista directori d'arxius de vídeos dividits per cada hora de la gravació contínua 24 hores de videovigilància per a les quatre càmeres. La vista és la proporcionada pel mateix servidor web Apache2, la qual ha calgut configurar adequadament perquè l'ordre de visualització sigui segons la data i per ordre de més recent a més antic. Així els primers elements de la part superior són els últims vídeos més nous disponibles. Els vídeos codificats es generen per un procés bash (nw-encoding.sh) que executa la utilitat mencoder per codificar el directoris d'hores senceres d'imatges JPEG. Aquest són generades i desades estructuradament en un camí i nom adequat pel procés Motion cada segon.

**nwcam.php:** genera el llistat d'alarmes a través de la base de dades MySQL segons el filtre de càmera seleccionat i nivell d'usuari. La inserció dels registres i les dades bàsiques associades a l'alarma recau al procés Motion. Les dades de qualificació, la acció escollida (borrar/guardar) i comentaris es generen a través de la web. També implementa l'edició de la qualificació i acció a realitzat en cada alarma. Inclou altres fitxers externs per a la funcionalitat de configuració horària.

**nwcam\_timetable.php:** implementa la interfície d'usuari per la configuració horària de la detecció per cada càmera i la seva programació per detectar alarmes fora de l'horari d'obertura al públic de la instal·lació de deixalleria, pròpiament inclòs per l'arxiu nwcam.php. Les dades s'emmagatzemen en paràmetres a la taula CONFIG de la base de dades.

**config.php:** s'implementa la consulta a la base de dades i el formulari d'edició corresponent per gestionar configuració dels paràmetres de l'aplicació.  
**users.php:** s'implementa la consulta a la base de dades i el formulari d'edició corresponent per gestionar els usuaris i les dades associades.  
**cron.php:** El servei periòdic de sistema Cron executa l'arxiu cron.php que configurar-la en calent el procés Motion a través de la seva API HTTP de control, segons la configuració existent a la base de dades.

## 6 Col·laboració en el projecte lliure Motion

En el desenvolupament del projecte, s'ha detectat una necessitat que no cobria el projecte *Motion* originalment i s'ha creat una millora. Els líders desenvolupadors del projecte han incorporat en la branca principal de desenvolupament i formarà part de la propera versió *Motion 3.2.8* disponible per a tota la comunitat. [2]

### 6.1 Descripció i justificació de la necessitat

El projecte *Motion* actualment permet adquirir imatges de dispositius suportats pel kernel Linux amb la part proporcionada pel projecte *V4L (Video For Linux)*; o bé mitjançant protocols HTTP o FTP per descarregar-se fitxers d'imatges des de dispositius de xarxa com càmeres IP.

Aquesta última característica vers càmeres IP amb protocols HTTP o FTP es configura amb la opció `netcam_url` en el fitxer `motion.conf.thread0x` corresponent [3]. En el següent exemple es detalla la configuració de *Motion* utilitzat en aquest projectes per accedir a aquests tipus de càmeres IP com les que es disposen a la instal·lació.

```
netcam_url http://192.168.3.40/IMAGE.JPG
```

Observant aquesta opció que utilitza una URL com a paràmetre, es va creure interessant que permetés un nou tipus de protocol `file://` i així obtenir les imatges d'un mateix arxiu local d'imatge. La necessitat està relacionada amb problemàtica de la càmera *3 Veo Observer* que no permet un accés estàndard (HTTP o FTP). Així s'evita el pas de simular un servidor web local per accedir a aquesta imatge i amb la conseqüent sobrecàrrega inútil del sistema. Per tant, amb la nova millora estarà disponible utilitzar l'anterior opció de configuració de la següent forma:

```
netcam_url file:///var/www/veo/imatge.jpg
```

### 6.2 Detall de la implementació

En els fils de discussió en el *wiki* de la web de desenvolupament del projecte *Motion*, es comenta la necessitat que la futura gran línia de versió 4.x.x reimplementar, reestructurar i millorar la part de permetre disposar de “*drivers*” específics addicionals i desenvolupats fora del propi projecte *Motion*. Drivers que la implementació pròpia del projecte no suporta.

El codi font està desenvolupat en codi C. L'anàlisi d'aquest s'encamina a agafar com a base i exemple de desenvolupament les funcions implementades per donar suport al protocol `ftp://`. Els detalls són:

- Els fitxer afectats són `netcam.h` i `netcam.c`.

- Es modifica la estructura “*struct netcam\_context*” afegint-hi el nou camp “*struct file\_context \*file;*” que és un punter a una nova estructura creada (*struct file\_context*) amb la informació pròpia per aquest protocol nou [file://](#) . (afecta fitxer *netcam.h* )
- Es creen les noves funcions “*netcam\_read\_file\_jpeg()*”, “*file\_new\_context()*”, “*file\_free\_context()*” i “*netcam\_setup\_file()*”. Es modifica la funció del bucle principal de procés d’adquisició d’una càmera per incloure el nou protocol “*file:*” en la funció “*netcam\_handler\_loop()*”, així com la funció “*netcam\_url\_parse()*” d’anàlisi del paràmetre “*netcam\_url*” de configuració (fitxer *netcam.c* ).

A continuació es descriu cronològicament el passos d’execució que afecten a la millora:

1. En la inicialització a través de la configuració del fitxers *motion.conf* i *motion.conf.thread0x*, es carreguen les variables a partir dels valors introduïts. Entre aquestes la variable “*netcam\_url*”, la qual en la funció “*netcam\_url\_parse()*” s’anàliza la cadena de text introduïda que sigui un valor i format URL vàlid, i suportat per a l’aplicació. Per tant, en aquest punt, ha calgut ajustar la comprovació per tal que accepti el nou protocol [file://](#). A més l’expressió regular que avalua els formats existents no s’adequa al nou format i cal una nova expressió per aquesta nova forma de URL vàlida. Però alhora cal que sigui compatible i similar a l’anterior, per mantenir les estructures de paràmetres ja existents en la funció i en l’aplicació. Així s’utilitza la mateixa estructura amb les funcions de sistema *regcomp()* i *regex()* existents. A continuació el detall les dues expressions regulars de verificació de URLs (la existent “[http://](#) + [ftp://](#)” i la nova per a [file://](#)) :

**(http|ftp)://(((.\*):(.\*))@)?([^\:][^-.a-z0-9]+):([0-9]+)?(\$|([^\:]\*))**

**(file)://(((.\*):(.\*))@)?([^\:][^-.a-z0-9]\*):([0-9]\*)?(\$|([^\:][^-.a-z0-9]+))**

2. A continuació les funcions “*file\_new\_context()*”, “*file\_free\_context()*” i “*netcam\_setup\_file()*” són auxiliars, imatge de les funcions existents per protocol [ftp://](#) i base utilitzada per a la millora de desenvolupada. La seva funcionalitat és simplement la petició de memòria dinàmica per a variables d’aquest protocol a partir del paràmetre “*netcam\_url*”, la seva inicialització i la seva alliberació al finalitzar.
3. El punt d’entrada del codi realment a executar és la funció “*netcam\_handler\_loop()*”. Aquesta s’executa per a cada fil o *thread* d’execució creat per a cada càmera configurada en el procés pare de *Motion* a través dels fitxers de configuració. En aquesta funció es comprova quin tipus de protocol s’utilitza ( [http://](#) o [ftp://](#) ) i s’utilitza un punter a funció pròpi i diferent per a cadascú. En aquest punt és també on s’ha afegit una comprovació més pel protocol [file://](#) i per tant s’utilitza el punter a funció a una nova funció pròpia (“*netcam\_read\_file\_jpeg()*”). Aquesta funció implementa la millora d’aquesta nova funcionalitat de suport a protocol [file://](#).
4. El nucli de la millora que permet la obtenció de la fitxer imatge del sistema d’arxiu es troba en la funció “*netcam\_read\_file\_jpeg()*”. Aquesta s’ha creat a partir de la base de la funció “*netcam\_read\_file\_jpeg()*” existent per a protocol [ftp://](#). S’ha canviat la part de petició per xarxa d’una nova imatge, per una espera activa de noves característiques (la data de modificació) del fitxer imatge especificat en “*netcam\_url*”. S’ha utilitzat la funció de sistema “*stat()*”. Quan les característiques canvien, l’arxiu ha estat sobreescrit o s’ha canviat l’enllaç simbòlic, es surt del bucle de espera activa i es fa la lectura de les dades de la nova imatge.

Immediatament és notifica al bucle principal de processat que hi ha una nova imatge disponible.

5. Després del processat, torna a executar-se altre cop la funció nucli d'adquisició i de la millora "*netcam\_read\_file\_jpeg()*", la qual queda monitoritzant, en una espera activa, una nova data de modificació del fitxer imatge especificat (tornar al punt 4 anterior).

### 6.3 Publicació de la millora en la branca principal del projecte Motion (patch)

Actualment el control de versions de desenvolupament del codi font del projecte es realitza mitjançant *SVN* (Subversion). L'accés de descarrega de només lectura és obert a tothom (lliure), però les modificacions d'aquest és limitat als desenvolupadors. La direcció del servidor *SVN* del codi font és <http://www.lavrsen.dk/svn/motion/trunk/>.

Els passos seguits a enviar la publicació de la millora van ser:

1. La lectura de la documentació de les webs del projecte (<http://www.lavrsen.dk/> i <https://sourceforge.net/projects/motion/>)
2. La descàrrega de la última versió de codi font en desenvolupament del projecte disponible.  
*svn co http://www.lavrsen.dk/svn/motion/trunk/*
3. Aplicar les modificacions per tal de tenir la nova funcionalitat
4. Generar un arxiu amb les diferències entre la versió última oficial i la modificada (*patch*)  
*svn diff netcam.c netcam.h > path\_fileurl\_netcam.diff*
5. Enviar un missatge el 22 d'abril de 2007 al desenvolupador líder actual del projecte, Kenneth Lavrsen, a través del portal de desenvolupament *SourceForge* (<http://sourceforge.net>)
6. Registrar-se i enviar un correu electrònic a la llista de desenvolupament ([motion-devel@sourceforge.net](mailto:motion-devel@sourceforge.net)) amb la diferència que inclou la millora (*patch*).
7. El mateix dia 22 d'abril de 2007, 14 minuts després del primer correu, contesta Kenneth Lavrsen el correu enviat anteriorment amb la millora, proporcionant informació vària, acceptant la millora i obrint una nova pàgina amb la informació sobre aquesta a la mateixa web de desenvolupament del projecte lliure Motion (<http://www.lavrsen.dk/twiki/bin/view/Motion/NetcamFileURLPatch>)
8. El dia 23 d'abril de 2007, el desenvolupador Angel Carpintero responsable de la propera versió 3.2.8 del projecte inclou la millora en la revisió r180 del *SVN* de codi font del mateix (missatge [\[Motion-devel\] \[SVN\] AngelCarpintero r180 - trunk](#) a la llista [motion-devel@sourceforge.net](mailto:motion-devel@sourceforge.net))

## 7 Conclusions

L'aplicació de videovigilància desenvolupada ha aconseguit amb escreix els objectius inicials. Ha estat instal·lada i en funcionament durant varis mesos en la instal·lació municipal de recollida selectiva o deixalleria. A més de l'enregistrament continu, és disposa d'un mecanisme de detecció i avís d'incidència o alarma amb validació remota. Això ha permès corregir conductes i procediments enregistrats per les càmeres durant períodes fora horari públic i sense personal en servei.

L'ús de material reciclat dificulta la instal·lació per no disposar lliurement del material més adequat ni tenir sempre disponibilitat ni homogeneïtat. Malgrat aquesta dificultat d'obtenir material específic, la variabilitat i la no garantia, el cost és molt reduït i el cost mediambiental encara més baix.

A nivell de desenvolupament i programació, destacar la facilitat de crear processos bash de sistema per a tasques específiques. Cal, però, limitar i simplificar el seu ús tenint present la seva lentitud per obtenir una aplicació àgil i eficient. Per altre banda, la programació en llenguatge PHP i els seus mòduls existents disponibles, faciliten el desenvolupament web dinàmic i vinculat a la base de dades. En la fase final, s'ha migrat cap a PHP alguns procediments per unificar la tecnologia de les diverses parts de l'aplicació. Així es facilita el manteniment, el reaprofitament del coneixement i de les rutines desenvolupades, en tasques anteriorment implementades en intèrpret de comandes bash. S'ha constatat la simplificació i facilitat que aporta l'ús d'estàndards de connexió, i més si són oberts.

Una altra conclusió important és la disponibilitat tecnològica i avançada de programari lliure que ja es disposa actualment, i que les noves tecnologies permeten fàcilment accedir-hi. Alhora, de forma individual, es poden realitzar millores puntuals al disposar del codi font i que aquestes millores es reverteixin i quedin per sempre al coneixement comú, lliurement distribuïble gràcies a la seva llicència.

## Referències

- [1] Escola Universitària Salesiana de Sarrià. <http://www.euss.es>.
- [2] Motion. <http://motion.sourceforge.net/> , desembre 2006 a maig 2007
- [3] Motion. <http://www.lavrsen.dk/twiki/bin/view/Motion/>, desembre 2006 a maig 2007
- [4] FFmpeg <http://ffmpeg.mplayerhq.hu/>, 1 abril 2007
- [5] Mplayer. <http://www.mplayerhq.hu/>, 1 abril 2007
- [6] Applet Java Cambozla. <http://www.charliemouse.com/code/cambozola/config.html>, 4 marc 2007
- [7] FFdShow. <http://ffdshow-tryout.sourceforge.net/> , 27 desembre 2007
- [8] VirtualDub. <http://www.virtualdub.org/> , 1 abril 2007
- [9] Ajuntament de Navàs. <http://www.navas.cat>