

Cámaras digitales bajo Linux

La Caja de Fotos

Las cámaras digitales progresivamente van reemplazando a las tradicionales en nuestras casas. La forma típica en Linux de transferir imágenes desde la cámara al disco duro es con gPhoto o mediante el módulo basado en kernel de memoria masivos USB. Los entornos gráficos como digikam prometen un uso mejorado.



POR ANDREA MÜLLER

Comprar carretes, revelarlos, ordenar y almacenar las copias, comprar marcos y montar las copias en álbumes. Todas estas tareas que nos ocupan tanto tiempo son ahora cosas del pasado para fotógrafos aficionados que han pasado de un equipo de fotografía tradicional a uno digital. Aquellos que han hecho el cambio se encuentran con diferentes aspectos como encontrar el mejor software para su ordenador y transferir las imágenes digitales desde la cámara hasta el PC de su casa. La respuesta para usuarios de Linux depende de la marca y modelo de la cámara digital.

Los dispositivos que se conectan por sí mismo al sistema como si fueran discos duros son los más sencillos, permitiendo al usuario la simple introducción del comando *mount* para tener acceso al dispositivo. Si tenemos una cámara que usa un protocolo especial para comunicarse

con el PC y es un modelo reciente, Puede que podamos utilizar el software producido por el proyecto gPhoto [1]. Este artículo describe ambos procedimientos y nos introduce a software diseñado específicamente para interactuar con cámaras digitales.

Montando la Cámara

Algunas cámaras, como las series Exilim de Casio o Aito, se comunican con el sistema como un disco duro SCSI de la misma forma que un dispositivo de memoria USB se conecta al equipo. Podemos usar el comando *mount* para montar la cámara, comprobando el archivo */var/log/messages* para descubrir el dispositivo adecuado:

```
Feb 1 19:27:18 aries kernel: SCSI device sdb: 20208
512-byte hdwr sectors (10 MB)
Feb 1 19:27:18 aries kernel:
```

```
sda: Write Protect is off
Feb 1 19:27:18 aries kernel:
sda: sda1
```

El siguiente comando ejecutado desde la raíz monta la cámara detectada que ha detectado como un *sda 1*:

```
mount -t vfat /dev/sda1
/mnt/usb -o umask=0
```

en */mnt/usb*. Los dispositivos son formateados con el sistema de archivos VFAT típicamente usado en Windows 9x. Como este sistema de archivos no tiene ningún permiso, el administrador puede asignarlos usando la opción *umask*. 0 es el valor más permisivo y asigna acceso de lectura y escritura a cualquier directorio de la cámara a cualquier usuario.

Una entrada en */etc/fstab* permitirá a usuarios sin privilegios el montaje de la cámara. Las opciones de montaje per-

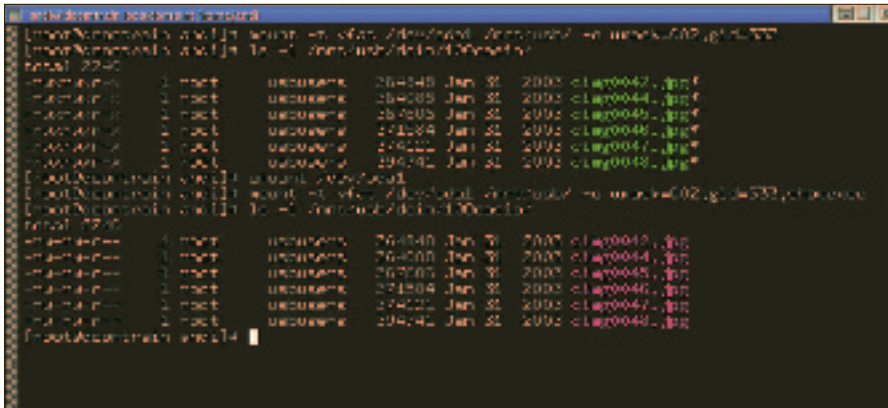


Figura 1: La opción de montaje "showexec" copia los permisos de archivos de Windows bajo Linux.

miten métodos para los permisos como asignar a un grupo *usbusers* permisos para borrar los archivos de la cámara. Tras crear el grupo y definir el punto de montaje, la entrada puede ser algo como:

```
/dev/sda1 /mnt/usb auto users,☞
noauto,gid=333,umask=002,☞
showexec 0 0
```

La entrada *gid* es seguida por una ID numérica como la almacenada en el archivo */etc/group*. Todos los miembros del grupo tienen acceso de lectura y escritura a la cámara. La entrada *users*, al contrario que *user*, permite a usuarios no privilegiados desmontar la cámara aún si ellos mismos no la han montado.

showexec le resuelve el problema resultante de que VFAT no disponga de privilegios de sistema. Se debe asignar permisos de ejecución a los usuarios en *umask* para permitirles el acceso a la estructura del directorio. Un efecto secundario es que los archivos normales son etiquetados como ejecutables. Algunos programas de Linux como las versiones más antiguas del gestor de archivos Rox rehusan mostrar una previsualización de imágenes ejecutables. Podemos usar *showexec* para solucionar este problema puesto que emula el comportamiento de Windows. Los archivos con extensiones como *.exe*, *.bat*, y *.com* son etiquetados como ejecutables pero no otras (ver figura 1). La entrada *fstab* permite a usuarios normales montar la cámara en */mnt/usb* usando *mount/dev/sda1*.

Los usuarios de Suse Linux pueden evitar este trabajo manual. El demonio *hotplug* crea automáticamente la *fstab* apropiada tan pronto se detecta la conexión de un dispositivo de memoria masiva USB.

Podemos usar casi cualquier gestor de archivos para transferir archivos de imágenes desde nuestra cámara al disco duro. Si preferimos un programa especializado debemos probar *Digikam* [2]. Es el único que funciona con *gPhoto 2* y también soporta dispositivos de almacenamiento masivo de memoria USB.

El proyecto gPhoto.

gPhoto 2, que alcanzó la versión 2.1.4 en enero del 2004 consiste en dos componentes: la librería *libgphoto2*, que soporta más de 400 tipos de cámaras, y el programa línea de comandos *gphoto2* que permite a los usuarios el acceso a las cámaras soportadas.

El proyecto tienen sus raíces en *gphoto0.4.x*. La versión antigua tenía una interfaz gráfico integrado y una arquitectura monolítica. *gphoto2* tienen una estructura modular y fue completamente reescrito por sus desarrolladores. La primera versión estable se comenzó a poder descargarse en febrero del 2002. Las nuevas versiones de este software están disponibles cada pocos meses con versiones beta y preestrenos para acortar el espacio entre versiones estables y permitir a los usuarios probar cámaras nuevas.

A medida que las versiones incluidas por los distribuidores tienden a quedarse atrás respecto al actual estado de desarrollo, los usuarios con nuevos modelos necesitan recopilar nuevos recursos. No obstante, esto también nos obliga a recompilar nuevas interfaces gráficas y *gtkam*, puesto que esta herramienta no funcionará habitualmente con la última versión si esta conectada a una librería antigua.

Antes de probar todas estas cosas debemos introducir el comando *gphoto2 --list-cameras* para buscar que dispositi-

tivos se soportan en la versión suministrada con nuestra distribución. Si no encontramos nuestro modelo debemos navegar hasta [3] para verificar el listado de compatibilidades de la versión actual de *gphoto2*.

Últimamente, *gPhoto 2* ha mejorado el soporte a modelos nuevos. Muchas de las cámaras actuales usan el Protocolo de Transferencia de Fotografías (Picture Transfer Protocol -PTP) [4] que asegura una transmisión simple de archivos entre dispositivos PTP. Esto elimina la necesidad de controladores especiales y permite a cualquier ordenador el intercambio de datos con cualquier otro dispositivo PTP sin la necesidad de instalar ningún software adicional.

De hecho este tipo de dispositivo no tiene porque estar limitado a cámaras y ordenadores. El intercambio de datos con teléfonos móviles o impresoras es igual de fácil. Las cámaras PTP son "plug and play" en Windows XP y MacOS X.

Soporte Mejorado PTP

Si bien el estándar fue publicado en el año 2000, muy pocos fabricantes estaban preparados para adoptarlo desde el principio, siendo Kodak y Sony unos de los pioneros. Afortunadamente, otros fabricantes comenzaron a apreciar las ventajas, por ejemplo, el del ahorro de costes, puesto que ya no necesitaban producir sus propios controladores. *gPhoto 2* introdujo el soporte de cámaras PTP en junio del 2002. Al principio era bastante espartano, siendo solo capaz de transferir archivos de imágenes a disco. Otros comandos como sacar el listado de los archivos de imágenes, pistas destinadas a la reproducción de errores I/O o interfaces basados en GUI solo funcionaban a ratos. La estabilidad se incrementó dramáticamente con la versión 2.1.3. No obstante, una cámara explícitamente listada en la lista de dispositivos soportados estaba mejor soportada que una cámara de clase genérica *USB PTP Class Camera*. Esta característica sigue en fase de pruebas.

Permisos

Si nuestra cámara está soportada, aún tenemos que lidiar con el problema de los permisos a no ser que nuestro distribuidor se haya preocupado de permitir

a otros usuarios al margen de *root* el acceso a la cámara. Las cámaras USB se detectan por el sistema como entradas debajo del punto de montaje */proc/bus/usb*. Aquí es donde está localizado el sistema de archivos USB. El kernel genera el sistema de archivos de forma genérica de forma similar al pseudo sistema de archivos */proc* [5]. Los cambios en permisos se pierden cada vez que se reinicia el sistema.

Hay dos formas de permitir acceso a la cámara a otros usuarios:

- Cambiando los opciones de montaje para el sistema de archivos USB.
- Ajustando dinámicamente los permisos usando el demonio *hotplug*.

El primer método es el más simple de los dos. La siguiente entrada en */etc/fstab*

```
none /proc/bus/usb usbfs
devmode=0664,devgid=333 0 0
```

monta en */proc/bus/usb* con permisos de lectura y escritura para usuarios y grupos con la identidad (ID) de 333 (estos son los usuarios USB en nuestro sistema). Los usuarios que no sean parte de este grupo solo tienen acceso de lectura.

Estas opciones de montaje son válidas para cualquier dispositivo USB y no sólo para cámaras.

El demonio *hotplug* proporciona una forma más elegante de resolver este problema. Monitoriza los puertos USB del sistema y reconoce cualquier dispositivo USB conectado por sus respectivos IDs, los cuales se almacenan en archivos */etc/hotplug*. Dependiendo de nuestra distribución, el demonio analiza el archivo */etc/hotplug/usb.usermap* o el archivo

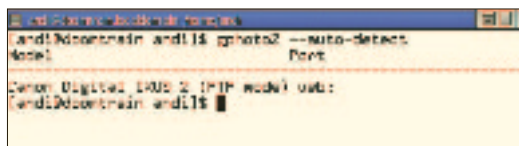


Figura 2: gPhoto 2 reconoce nuevas cámaras conectadas automáticamente por conexión USB.

/etc/hotplug/usb/usbcam.usermap para la cámara.

Scripts

Cuando el demonio conecta con un dispositivo llama al código */etc/hotplug/usb/usbcam*, que ajusta los permisos para la cámara conectada. *libgphoto2* proporciona cuatro códigos de ejemplo. Si construimos el código nosotros mismos, los encontraremos en el directorio de código fuente bajo *packaging/linux-hotplug*. La mayoría de las distribuciones los almacenan bajo */usr/share/doc/libgphoto2-versionnumber*.

Los cuatro scripts de ejemplo suministrados consisten en: *usbcam.group*, donde el administrador introduce el grupo al que asignar los permisos correctos, *usbcam.user*, el cual espera el nombre del usuario al que asignar permisos, *usbcam.x11-app*, el cual lanzará inmediatamente gPhoto y *usbcam.console*.

El código requiere PAM [6], instalado por defecto en Red Hat o Mandrake Linux. Éste asigna todos los permisos de la cámara al usuario conectado a la consola. Si construimos nuestra propia versión de gPhoto, necesitaremos actualizar *usbcam.usermap* usando el siguiente comando:

```
/usr/local/lib/libgphoto2/
print-usb-usermap >
/etc/hotplug/usb/usbcam.usermap
```

Si nuestra distribución analiza *usb.usermap* deberemos borrar cualquier

archivo que comience con *usbcam* y luego añadir el ID de la cámara al final del archivo:

```
/usr/local/lib/libgphoto2/
print-usb-usermap >>
/etc/hotplug/usb.usermap
```

La configuración de las cámaras de esta forma garantiza el acceso a usuarios normales. Un interfaz gráfico no es absolutamente necesario. La herramienta de la línea de comandos de gPhoto 2 proporciona un amplio abanico de funciones. La siguiente sintaxis:

```
gphoto2 --auto-detect
```

Muestra nuestra cámara tal y como es detectada por el programa (ver figura 2).

Cámara en la Línea de Comandos

Solo se requieren unos pocos parámetros para el acceso básico. *-l* lista los directorios, *-L* añade los nombres de los archivos. Los usuarios pueden hacer referencia a los números de serie en estos resultado (ver figura 3) usando la sintaxis *-p number(n)* para filtrar archivos específicos. *-P* hace referencia a todas las imágenes encontradas en el directorio actual de trabajo.

Si solo queremos transferir archivos de imagen específicos al disco duro podemos indicar *gphoto2 -p 2,4,5* para descargar las imágenes 2, 4 y 5 o especificar un rango, por ejemplo *gphoto2 -p 2-4*, es bastante fácil excluir archivos de un rango. Por ejemplo *gphoto -p 2-6,4* descarga las imágenes 2, 3, 5 y 6, pero no la número 4.

Nombres Intuitivos

El programa puede hacer muchas más cosas. Las cámaras digitales asignan

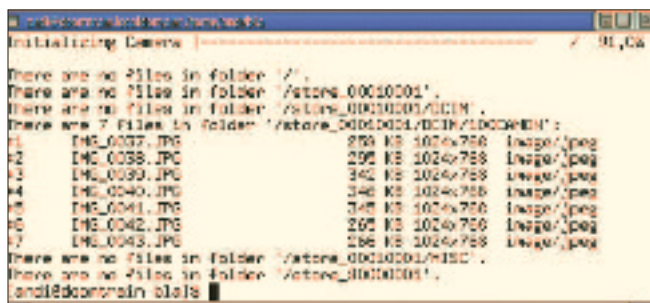


Figura 3: gPhoto 2 muestra una lista de las imágenes en la cámara.

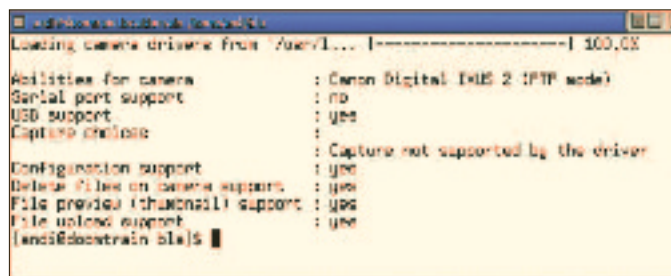


Figura 4: El parámetro *-a* lista las capacidades de la cámara, indicando si la cámara soporta miniaturas, por ejemplo.mentación.

nombres de archivo a las imágenes que son francamente poco intuitivos, pero *gphoto2* puede cambiar el nombre cuando descarga los archivos. Para hacer esto debemos indicar el parámetro `--filename` seguido por un nombre. `--filename` acepta formatos típicos de fechas (*date*) como `%y`, `%m`, y `%d` para años, meses y días. Adicionalmente, `%n` especifica un número, `%C` el sufijo de un archivo y finalmente `%f` especifica un nombre de archivo sin sufijo. *gphoto2 -P --filename cumpleaños_juanito_%n.%C* descarga las fotografías de la cámara y construye el nombre del archivo con la ristra *cumplazos_juanito_*, el número de la imagen, un punto y el sufijo del archivo.

Los parámetros `-T` y `-t número(s)` descargan cualquier o una selección de miniaturas de nuestra colección de fotos. Si disponemos de una cámara que puede almacenar clips de vídeo, al margen de imágenes, podemos especificar `--get-audio-data número(s)` o `--get-all-audio-data` para acceder a estos archivos. Para hacer sitio a nuevos archivos introducimos `--delete-files número(s)` y `--delete-all-files`.

Las funciones que *gPhoto 2* ofrece dependen en parte de la cámara que conectemos. *gphoto2 -a* nos mostrará las funciones de la dispositivo conectados (ver figura 4).

Algunas cámaras permiten que les subamos archivos de imágenes. El parámetro `-m` seguida de un nombre crea un directorio en la cámara y *gphoto2 -u nombrefichero* sube archivos. Los resultados de *gphoto2 -a* no son completamente fiables. Si bien se supone que el controlador para nuestra cámara de ensayo no soporta *capture* según el programa, es posible controlar la cámara remotamente y *gphoto2 --capture-image* capturará una imagen. Desafortunadamente, la opción `--capture-sound` nos llevó a errores entrada / salida en nuestro laboratorio Linux.

El parámetro `--shell` permite al usuario navegar por los directorios de la cámara igual que haría por los sistemas de archivos locales. Los usuarios pueden introducir *gphoto2 --shell* para desplegar

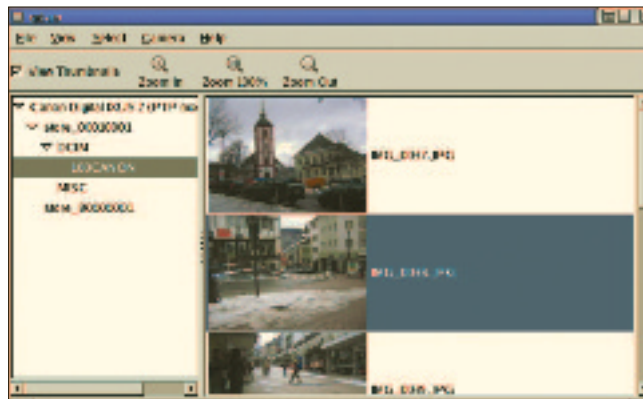


Figura 5: Claro y funcional: el GUI de GTKam.

la línea de comandos interna del programa. Si alguna vez hemos usado un cliente basado en texto FTP, no debemos tener problemas en encontrar el camino.

Funciones Adicionales

help muestra lista de los comandos disponibles para *gPhoto2*, *help nombre-comando* muestra la sintaxis de comandos individuales. *ls* lista los archivos. *cd* cambia a otro directorio de la cámara. *delete file* borra un archivo. *get file* inicia la descarga de una fotografía al directorio actual, y *get-thumbnail* hace lo mismo con la miniatura de la imagen actual. Si abrimos el shell de *gPhoto* en un directorio para el que no tenemos privilegios de escritura, podemos escribir *lcd directory* para cambiar a otro directorio local. El comando *show-exif fichero* está diseñado para mostrar los meta-datos del archivo de una imagen, pero no funcionó en nuestro laboratorio. *q*, *quit* o *exit* finaliza el programa de la línea de comandos.

Entornos gráficos

Si no estamos contentos con la línea de comandos, podemos probar uno de los entornos gráficos de *gphoto*. Hay muchos compitiendo actualmente por la posición de mejor amigo del fotógrafo amateur.

El competidor más antiguo es *gtkam* [7], competidor fuerte (ver figura 5) que solo provee funcionalidades básicas. Seleccionando *Camera | Add | Camera | Detect* indica al programa que acceda a la función de auto reconocimiento de *gPhoto 2*. Podemos usar una vista tipo árbol del gestor de archivos en una cámara que lo soporte. *GTKam* muestra imágenes en la parte derecha del panel.

La vista miniaturizada proporciona una vista general inicial. El menú *File* nos permite descargar todas las imágenes o solo las seleccionadas y borrar de la cámara cualquier imagen que hemos transferido correctamente a nuestro disco duro.

digikam proporciona muchas más ventajas para el usuario. En lugar de abandonar al usuario tras la descarga, la herramienta también gestiona imágenes en el

disco duro. Las imágenes se organizan en álbumes, los cuales con en realidad simples directorios. El programa mostrará los álbumes en lugar de simplemente ofrecer una vista general del sistema de archivos. Las vistas miniaturizadas se muestran en la zona derecha del panel (ver figura 6). Podemos arrastrar y soltar una imagen para copiarla o moverla a otro álbum en la vista árbol.

Digikam, la estrella emergente

Digikam muestra utiliza una método abierto a la hora de montar una cámara. Al margen de los modelos soportados por *gPhoto 2*, ofrece además acceso a dispositivos de almacenamiento masivo USB. Podemos seleccionar *USB Mass Storage* e introducir un pinto de montaje (ver figura 7) para ajustar este tipo de dispositivos. Seleccionando *Camera | Connect* montamos el dispositivo en el archivo del sistema.

Además de las funciones de descarga y borrado de imágenes, *Digikam* también tiene la función de subir imágenes. Para ver imágenes a escala completa podemos abrirlas opcionalmente en un visor externo en lugar de en el visor interno. Los programas KDE como *kuickshow* y *kview* son algunos ejemplos de visores externos. Una función de presentación de diapositivas que opcionalmente soporta etiquetas con nombres de archivos redondea el programa.

La nueva versión 0.7 acaba de salir. Esta versión tendrá funcionalidades adicionales. *Digikam* usa una arquitectura plug-in en la nueva versión, permitiendo a los desarrolladores extender la funcionalidad del programa. Los candidatos de versión 1 y 2 ya disponían de una

colección de funciones plug-in que proporcionaban las siguientes ampliaciones entre otras:

- Presentaciones de diapositivas con efectos.
- Codificador MPEG que puede usar la herramienta *mjpeg* o *Imagemagick* para crear vídeos para VCDs, SVCDs o DVDs.
- Galería HTML, una extensión de los plug-ins de Konqueror Gallery
- Editor de comentarios.
- Ayudante de impresión de fotografía para ordenar múltiples imágenes en la misma hoja.
- Archivo de CD (necesita K3b).
- Funciones de correo electrónico con opciones para cambiar el tamaño y ratio de compresión de los archivos. Funciona con KMail, Sylpheed y Mozilla.

En lo que se refiere a la gestión y archivo de imágenes digitales, Digikam ha desarrollado funciones que atienden a todas las necesidades de los usuarios. *flphoto* [8] usa un método distinto. El programa es realmente una herramienta de gestión de imágenes que ofrece acceso a la cámara como un añadido. La herramienta carece de la opción de borrar imágenes de la cámara. El diálogo *Album | Import | Camera* proporciona una solución fácil no obstante. Cuando arrancamos este programa, FLPhoto detecta automáticamente la cámara y muestra previsualizaciones en miniatura de las imágenes en la ventana de transferencia (ver imagen 8).

Las funciones de gestión de imágenes son muy avanzadas. El programa trabaja con álbumes, que en realidad son archivos de texto que contienen los nombres de las imágenes del álbum. Si

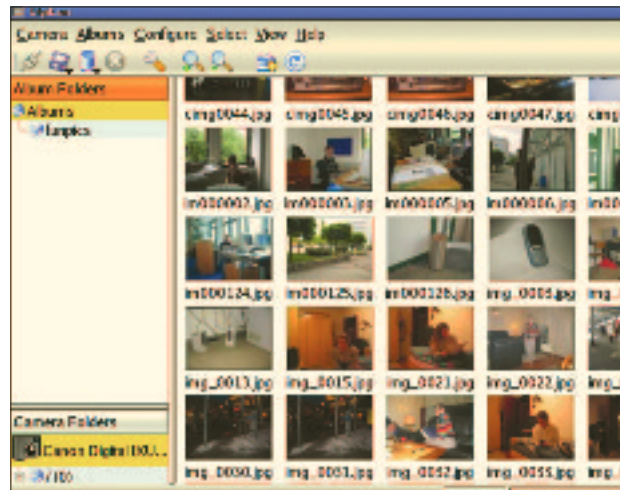


Figura 6: Vista de árbol, vista cámara y previsualización en Digikam.

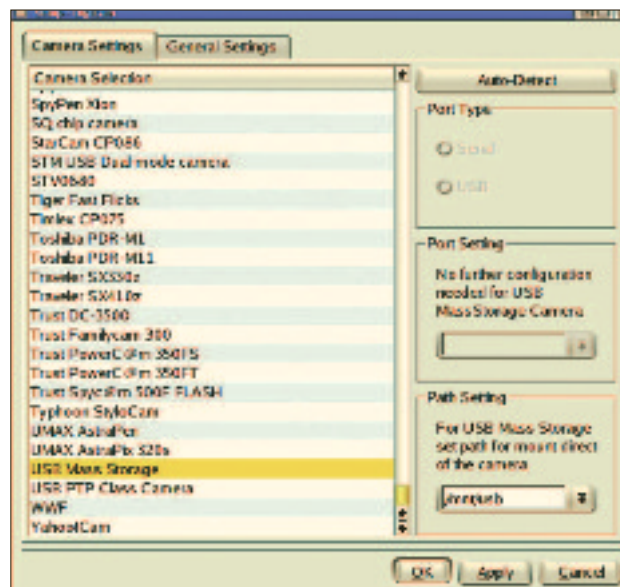


Figura 7: Digikam también soporta cámaras que el sistema detecta como sistemas de almacenamiento masivo USB.

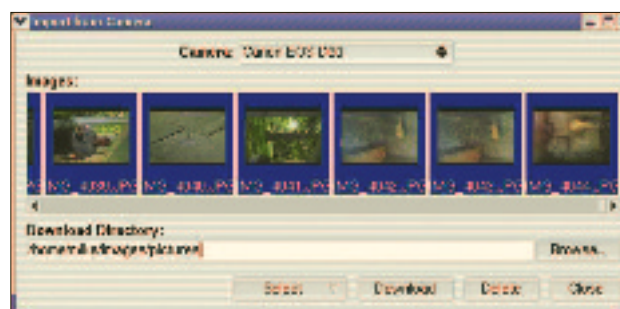


Figura 8: Importación de imágenes desde una cámara digital en FLPhoto.

seleccionamos *Image | Remove* para borrar una imagen del álbum, no borramos la imagen físicamente de nuestro disco. La barra de previsualización en la parte inferior de la ventana es más práctica

que la típica vista en forma de árbol, puesto que deja más espacio para la imágenes en el escritorio (ver figura 9).

Si preferimos usar el mismo programa para todas las operaciones con nuestros archivos, puede que lo preferible sea optar por usar el gestor de archivos KDE Konqueror para transferir imágenes desde nuestra cámara al disco.

Gracias a la Kioslave *camera*, la herramienta de gestión de archivos de KDE puede usar la librería *libphoto2* para acceder a cualquier cámara digital soportada como si estuvieran localmente conectada al sistema. Escribiendo *camera:/* en la línea de direcciones URL, el proceso de auto detección le indica a Konqueror que debe mostrar el directorio raíz de la cámara.

Información Exif

Las imágenes de algunas cámaras digitales incluyen el llamado Exif (Formato de intercambio de archivos de imágenes.- Exchangeable Image File Format) [9], que almacena información como la fecha y hora, la distancia focal, el modo del flash o el tipo de cámara. Los usuarios que deseen mantener esta meta-información mientras procesan las imágenes, no pueden usar herramientas de manipulación de imágenes tradicionales, puesto que la meta-información se pierde al guardar las imágenes de la cámara al disco duro.

Se está buscando una solución a este problema en la próxima versión de Gimp, puesto que es capaz de guardar la información Exif. Mientras tanto debemos optar por el comando llamado *jhead* [10].

jhead analiza la cabecera de Exif, la almacena y luego manda la imagen a otro programa donde pueda ser editada. La siguiente sintaxis ejecutará exactamente esta tarea:

```
jhead -cmd "jpegtran -rot 270
&i > &o" foto_retrato.jpg
```

jpegtran rota la imagen 90 grados a la izquierda (*-rot 90* rotará el mismo ángulo pero hacia la derecha). Esto es algo que los fotógrafos aficionados necesitan habitualmente cuando hacen fotografías con la cámara en posición vertical. *jhead &i* con el nombre del archivo original. *&o* se refiere a un archivo temporal donde los resultados serán almacenados. *jhead* inserta la cabecera Exif en este archivo y reemplaza la imagen original con la rotada. Como el programa automáticamente asigna el nombre del archivo correcto podemos usar comodines como **.jpg* para procesar múltiples imágenes en un mismo lote.

JPegtran: Mejor que Imagemagick

Comparado con *convert* del paquete Imagemagick, el programa JPegtran usado en nuestro ejemplo tiene la ventaja de crear archivos jpeg sin pérdidas. Si usamos *convert*, el almacenaje de imágenes recomprimará la imagen y perderá datos de la misma. JPegtran es parte de *libjpeg* y debe estar disponible con la mayoría de las distribuciones.

La solución más directa para visualizar cabeceras Exif actualmente en GTK es un programa llamado *gexif*, disponible para su descarga desde la

página Web del proyecto *libexif* [11]. Usa pestañas para mostrar los campos individualmente e incluso muestra explicaciones de algunos tipos de contenidos (ver figura 10). La última versión de Digikam, a partir de la 0.6, dispone de la misma función.

exiftags [12] es útil para mostrar información de Exif en la línea de comandos. Separa el resultado en la cámara y las partes específicas de la imagen. El mismo paquete incluye adicionalmente *exifcom*, que puede ser utilizado para rellenar el campo de comentarios del usuario (UserComment) de la cabecera de Exif (se supone que la propia cámara fija este campo). Llamar a la herramienta sin ningún tipo de indicador o parámetro mostrará un comentario, si está disponible.

```
exifcom -w "my
comment"
imagefile.jpg
```

fijará o sobrescribirá el comentario. Si deseamos prevenir que *exifcom* nos solicite permiso antes de sobrescribir, simplemente debemos añadir el parámetro *-f*.

Con Buena Voluntad...

Algunas personas aún mantienen que Linux no implementa multimedia como otros sistemas. Obviamente hace tiempo que no prestan atención al mundo de las

cámaras digitales, puesto que se han producido unos

| INFO | |
|---|---|
| ff[1]gphoto2: | http://www.gphoto.org/ |
| [2] Digikam: | http://digikam.sourceforge.net/ |
| [3] lista de las cámaras compatibles con gPhoto2: | http://www.gphoto.org/proj/libgphoto2/support.php |
| [4] Software gratuito con soporte PTP: | http://ptp.sourceforge.net |
| [5] Sistemas de archivos de dispositivos USB: | http://www.linux-usb.org/USB-guide/x173.html |
| [6] PAM: | http://www.kernel.org/pub/linux/libs/pam/ |
| [7] Gtkam: | http://www.gphoto.org/proj/gtkam/ |
| [8] flphoto: | http://www.easysw.com/~mike/flphoto/ |
| [9] Exif: | http://www.exif.org/ |
| [10]jhead: | http://www.sentex.net/~mwandel/jhead/ |
| [11] gexif: | http://sourceforge.net/projects/libexif/ |
| [12] exiftags: | http://johnst.org/sw/exiftags/ |

progresos tremendos en esta área a lo largo de los últimos meses.

El proyecto KDE, Digikam, es un buen ejemplo esta afirmación: es un paquete potente y cómodo de utilizar, que hace que el procesado de imágenes en Linux sea un verdadero placer. Incluso si no tenemos suerte y Digikam no soporta nuestra cámara, aún podemos utilizar como último recurso un lector de tarjetas basado en USB que nos garantizará ser capaz de transferir nuestra fotografías a nuestro disco duro

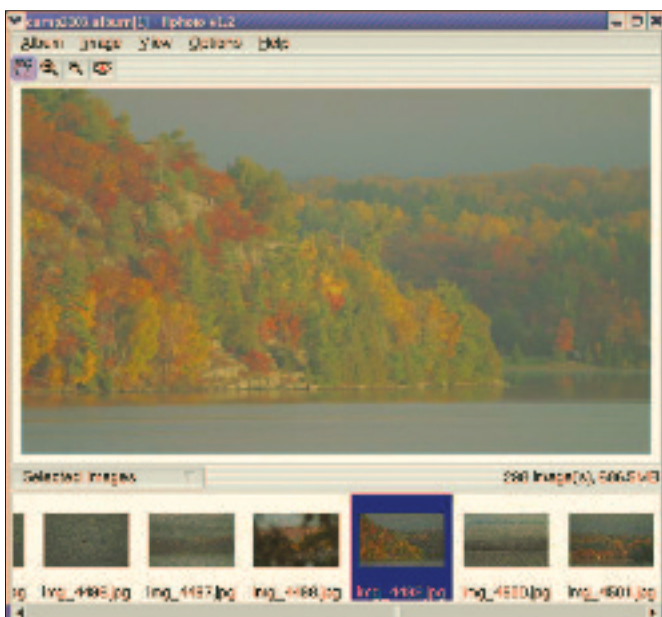


Figura 9: Ventana de previsualización y visualizador de imágenes de FLPhoto.

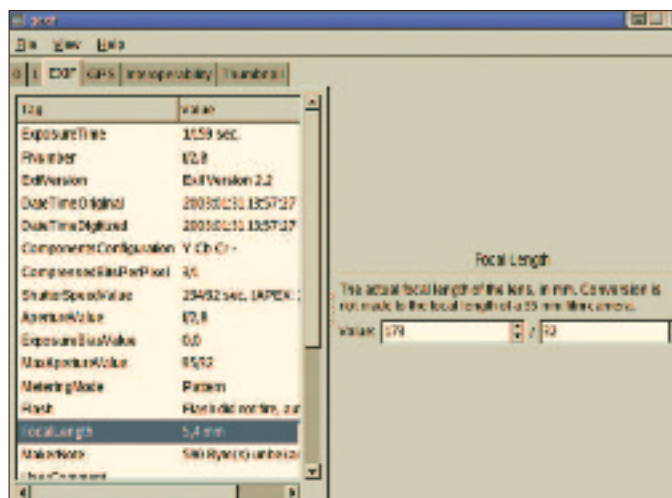


Figura 10: Gexif muestra toda la gama de información Exif, desde la configuración de la cámara, hasta miniaturas.