ImageMagick (tutorial).

Recientemente tuvimos la estupenda oportunidad de asistir al Congreso de Tecnologías Libres 2016 y tuvimos la necesidad (madre de las invenciones) de publicar las fotografías que capturamos en el evento. En un principio redimensionamos unas pocas para nuestra cuenta Twitter, pero pronto nos dimos cuenta que la tarea es tediosa y debemos aligerar la carga con herramientas del Software Libre. No hace mucho tiempo uno de nuestros faros en GNU/LINUX -en lengua castellana- **Ubunlog** publicaron un artículo sobre ImageMagick: instalación y usos básicos del mismo. Pero como nos percatamos que el proceso masivo de 300 imágenes en una sola linea de comando ***puede*** "colgar" nuestra computadora por largo tiempo decidimos publicar esta entrada con el valor agregado de nuestros anteriores temas publicados y además unos "scripts" que tal vez les puedan ser útiles a ustedes, amén de la recomendación de un "plugin" para Wordpress con el cual escribimos estas líneas a la fecha (quien sabe, tal vez algún día evolucionemos hacia otra plataforma de *blogging*).

ImageMagick.

Breve historia.

Bien lo retrata en su página web la historia de ImageMagick que pasamos a traducir y resumir, contada en idioma inglés por **John Cristy** (Principal ImageMagick Architect):

Corría el año de 1987 cuando el **Dr. David Pensak**, supervisor de John Cristy en la empresa de productos químicos llamada **Dupont**, le solicitó poder mostrar imágenes de 24 bits (color verdadero) en los nuevos -y costosos- monitores de 256 colores ya que hardware de aquella época tenía muy poca potencia -y por ende debían ser convertidos a 256 colores-. Es por ello que John Cristy utilizó el buscador de moda para ese entonces: Usenet. Obtuvo respuesta de **Paul Ravelin** donde le indicaba no una, sino varias soluciones de software para la tarea encomendada y puso a su disposición un servidor FTP del "Information Sciences Institute" (ente adscrito a la Universidad del Sur de California) con el código fuente de numerosas aplicaciones. Tras varios años de conseguir muchas de las respuestas, en lo que a computación se refiere, en su trabajo para la empresa Dupont -y el exigente Dr. David Pensak- él se decidió a mejorar y retribuir todo el software utilizado y decidió igualmente liberar las herramientas de procesamiento de imágenes para que otros -¡ejem! nosotros por ejemplo- nos beneficiriamos de ello (de hecho nosotros contribuimos en esta entrada con un "script" en "bash" y otro en lenguaje **PHP**, así que la historia ¡sigue y sigue!).

Pero como del "dicho al hecho hay enorme trecho" él primero tenía que solicitar permiso a la empresa Dupont en la cual laboraba, ya que en horas de trabajo fue que él desarrolló dichas herramientas. Es así que de nuevo interviene el Dr. David Pensak y convence a sus superiores de

otorgar permiso de "copyleft" a John Cristy ya que no era ni un producto químico ni biológico y ellos no tenían noción del valor del software para entonces. **Es así que el 1° de agosto de 1990 ImageMagick ve la luz en Usenet en el grupo "comp.archives" (gracias de nuevo Dr. Pensak).**

A mediados de los años 1990, y con miles de usuarios en el mundo entero, ImageMagick versión 4.2.9 fue incluido en un nuevo sistema operativo que era distribuido libremente: **GNU/Linux**.

Es así que luego de su distribución junto a GNU/Linux el sr. **Bob Friesenhahn** contacta a John Cristy a fin de "normalizar" la aplicación para que sea compatible con el resto de las herramientas de dicho sistema operativo (más adelante veremos que gracias a esto es que hoy en 2016 nosotros podemos desarrollar "scripts" o guiones funcionales y compatibles en otros idiomas de programación).

A partir de la versión 5 de ImageMagick se incorpora de esta manera el lenguaje C++ y se unen al desarrollo los siguientes programadores:

- Bob Friesenhahn (proponente)
- Glenn Randers-Pehrson.
- William Radcliffe.
- Leonard Rosenthol.

Ya eran decenas de miles de usuarios de ImageMagick cuando sucedió lo impensable: el desarrollo evolucionó tanto que en un momento dado la nueva versión era incompatible con una API existente e hizo que los usuarios reaccionaran bruscamente y exigieron paralizar la programación mientras que los desarrolladores quería seguir adelante. John Cristy no dio su brazo a torcer así que ImageMagick -de la mano de Bob- recibe su primera bifurcación de código y nace Magick++, *el primer "fork" (como se conoce en el idioma inglés).* Recordemos que precisamente esta es una de las normas de la licencia que rige el software libre, así que John Cristy continuó solo su camino.

Pero no trabajó solo por mucho tiempo: **Anthony Thyssen** le indicó ciertas fallas en la linea de comandos, los cuales no solo se corrigieron sino que también se mejoraron hasta tal punto que vieron que era necesario emitir una nueva versión: ImageMagick 6.0.

Tan lejos llegaron las librerías de Anthony Thyssen que el mismo John Cristy quedó sorprendido de la capacidad del código fuente original, y que públicamente reconoce la labor hecha en el avance de la colaboración en proyectos de software libre. A continuación, y en honor de quienes contribuyeron (y respetando las normas de la licencia GNU bajo la cual está concebida ImageMagick) nombramos a:

- **Fred Weinhaus** (cientos de "scripts" *que son libres para uso no comercial, caso contrario*

contactar a Fred Weinhaus para su autorización).
- **Glenn Randers-Pehrson** (gurú del formato PNG).
- **Dirk Lemstra** (desarrollo en ambiente "Windows" bajo .NET)

ImageMagick tiene ya una edad de 25 años al momento de escribir este artículo, y rumbo a los siguientes 25 años se desarrolló la versión 7.0 con importantes novedades descritas en este enlace web. Además, ustedes pueden encontrar la licencia que rige a ImageMagick en este otro enlace.

Instalación de ImageMagick en Ubuntu.

La instalación es común a las distros GNU/Linux basados en Debian:

```
apt-get install imagemagick
```

Recuerden que deben tener derechos de usuario raíz, para mayores detalles al trabajar la línea de comandos consulten nuestro tutorial al respecto.

Finalmente, para verificar si está correctamente instalado en nuestro ordenador, podemos ejecutar las siguientes lineas de comando con las cuales "crearemos" el logotipo de ImageMagick, visualizaremos sus especificaciones con el comando **identify** y luego lo abriremos en una ventana gráfica con el comando **display**:

```
convert logo: logo.gif identify logo.gif display logo.gif
```

> Al ejecutar el comando **display** tal vez recibiréis un mensaje un tanto singular: el reporte de unas fuentes de texto faltantes. La explicación rápida es que son fuentes privativas, no libres, y no acompañan a las distribuciones GNU/Linux. Más información en este enlace web.

Tal vez, cuando estéis más avezado o avezada con ImageMagick, necesitareís instalar las librearías avanzadas (una de tantas que existen) con el siguiente comando:

```
apt-get install graphicsmagick-imagemagick-compat
```

Como vosotros podéis ver, de primero utilizamos el comando convert el cual pasamos a describir en la siguiente sección.

Comando "convert".

El comando que nos interesa para redimensionar de manera masiva -y a nuestra manera- una gran cantidad de imágenes es el comando "**convert**". Específicamente para redimensionar lo acompañamos del argumento "**-resize**" y de seguido los dos valores de ancho y alto deseados. Sin embargo, debemos conocer un poco más acerca de algunos de los otros argumentos disponibles:

- Lo más básico: renombrar imágenes de manera masiva seleccionando un patrón de búsqueda y un prefijo que automáticamente numerará el comando. Por ejemplo si introducimos la orden "**convert *.jpg fotos.jpg**" ImageMagick renombrará todos los archivos jpg en la carpeta donde estemos ubicados en la linea de comandos de la siguiente manera: **foto-1.jpg** , **foto-2.jpg** , **foto-3.jpg** , etc.
- Ya vimos cómo renombrar masivamente un grupo de imágenes pero para convertir una sola solo debemos, desde luego, indicarle su nombre específico, y si queremos o necesitamos, otro nombre específico de salida para mantener el original; es decir, si omitimos el segundo nombre ImageMagick reemplazará el archivo de imagen original -cuidado con esto-. Las siguientes opciones soportan ambas maneras en este párrafo descritas y renombran masivamente según el párrafo anterior.
- Para rotar una imagen utilizamos el argumento "**-rotate**" seguido del ángulo a rotar, por ejemplo "**convert imagen.jpg -rotate 90 nueva_imagen_rotada.jpg**".
- Si queremos convertir a otro formato de archivo simplemente especificamos el o los archivos deseados acompañado del nombre con la extensión deseada. Por ejemplo "**convert imagen.jpg imagen.png**" o si queremos convertir todas las imágenes jpg en una carpeta: "**convert *.jpg imagen.png**" (recordad que ImageMagick agregará un sufijo numerado a cada archivo convertido: **imagen-1.png** , **imagen-2.png** , **imagen-3.png** , etc.)
- También podemos bajarle calidad a una imagen utilizando el argumento "**-quality**" acompañado del porcentaje deseado -formato jpg-.
- Si necesitamos redimensionar utilizamos, por ejemplo, "**convert imagen.jpg -resize 1024x768**" con lo cual obtendremos una imagen de tamaño 1024 píxeles de ancho por 768 píxeles de alto *sin conservar el archivo original.* Para obtener un archivo nuevo (otro ejemplo) emplearíamos "**convert imagen.jpg -resize 1024x768 imagen_redimensionada.jpg**".
- Por último podemos combinar los diferentes argumentos, teniendo en cuenta el problema

con el que nosotros nos tropezamos: el redimensionamiento masivo de imágenes puede hacer que nuestro ordenador quede bloqueado durante un buen tiempo, por eso decidimos utilizar un "script" que procesa uno a uno cada archivo.

- *Actualizado el miércoles 18 de septiembre de 2019:* el comando **convert** es compatible con gif animados como el siguiente que rotamos 180° para mostrar el signo de interrogación de apertura:

Uso de ImageMagick en un "bash script".

Ya en una entrada anterior hablamos procesar una serie de imágenes y aplicarle Reconocimiento óptico de caracteres con el programa **Tesseract** y vamos a reutilizar el "script bash" o proceso por lotes allí muy bien explicado, así que si os gusta id, leedlo y volved.

```
#!/bin/sh ####Licencia de uso### # Copyright 2016 Jimmy Olano at ks7000.net.ve # # Licensed under the Apache License, Version 2.0 (the "License"); # you may not use this file except in compliance with the License. # You may obtain a copy of the License at # # http://www.apache.org/licenses/LICENSE-2.0 # # Unless required by applicable law or agreed to in writing, software # distributed under the License is distributed on an "AS IS" BASIS, # WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. # See the License for the specific language governing permissions and # limitations under the License. ###################### patron="P*.JPG" nomarch="lista.txt" clear ls $patron > $nomarch while read linea do echo "Procesando "$linea convert $linea -resize 1024x768 $linea echo "aplicando logotipo a "$linea php funde_logo.php $linea rm $linea done
```