



REVISTA
Software Libre

Número de Aniversario
Enero / Febrero 2008

Richard Stallman platica con Revista SL

Revista SL a 3 años de trabajo:
Conoce la magia tras pantomimas

- * El proyecto OLPC en América Latina, conócelo
- * Columna MySQL
- * Introducción al desarrollo de interfaces con GTK# y Mono
- * Columna PERL





Eres libre de copiar, distribuir y comunicar públicamente RevistaSL

Eres Libre de hacer obras derivadas de RevistaSL



Atribución. Debes reconocer la autoría de RevistaSL y sus colaboradores en los términos especificados por el equipo editorial.

Al reutilizar o distribuir RevistaSL, tienes que dejar bien claro los términos de la licencia Creative Commons



Alguna de estas condiciones puede no aplicarse si se obtiene el permiso del equipo editorial

Nada en esta licencia menoscaba o restringe los derechos morales de los autores.

RevistaSL

El software libre hecho revista



~# Editorial SL



Parece increíble que ya han pasado tres años desde que iniciamos con esta aventura, tal vez ninguno de los que colaboramos en esta publicación llegamos a pensar que seguiríamos tanto tiempo con este proyecto que tan sólo pretendía difundir lo que a nosotros nos gusta y en lo que creemos: El Software Libre.

Varias personas han colaborado para esta revista durante el tiempo que ha durado el proyecto, algunas colaboran de forma esporádica, otras por única vez pero al tiempo que crece el interés por esta publicación tenemos el gusto de contar con mas colaboradores.

Aunque los artículos que aquí se publican siempre se han caracterizado por ser técnicos, en este número tenemos la oportunidad de ver artículos no tan técnicos; sobre todo por que el tema que queríamos resaltar era el aniversario del proyecto, por ello dedicamos un gran número de páginas a la historia y estado actual del proyecto, sobre todo para que conozcan que hay detrás del PDF que tienen en sus computadoras en este instante, y más aun que se animen a enviar un artículo, colaborar o simplemente contarle a todos sus amigos lo buena que es RevistaSL.

Dado que es un número especial, quisimos mostrar a un invitado especial: a Richard Stallman, quien platica algunas de sus ideas relacionadas al Software Libre; pero como sabemos que los rollos filosóficos a veces no nos entretienen igual que ver líneas de código, hemos incluido artículos bastante interesantes como el desarrollo de aplicaciones en C# que es uno de los lenguajes con gran crecimiento y que puede ser usado sin problema alguno en plataformas libres gracias al proyecto Mono, la acostumbrada Columna PL para esos amantes de Perl y sus bondades y una nueva columna, la Columna MySQL que estará escribiendo Jorge Medrano y muchos artículos mas.

Y por si las sorpresas fueran pocas, este número se liberará en el Congreso Nacional de Software Libre 2008, escenario donde se anunció por primera vez la existencia del proyecto en su edición 2005.

No nos queda mas que agradecer los tres años en que nos han leído, los tres años en los que nos han hecho llegar sus colaboraciones, los tres años en los que nos han abierto las puertas en sus eventos, los tres años en donde los patrocinadores han confiado en nosotros y sobre todo los tres años en que la comunidad de Software Libre nos ha acogido como parte de si misma.

Equipo Editorial SL
buzon@revista-sl.org

Editor en Jefe SL

Gonzalo Javier González Rodríguez

Coordinador Editorial SL

Carlos Augusto Lozano Vargas

Equipo de Redacción SL

Julio Mauricio Acuña Carrillo

Sonia Sánchez Díaz

Andres Bernardo Vargas Rodríguez

Victor Hugo Cordova Madrid

José Luis Galicia Sánchez

Jesús Antonio Balam Jiménez

Julio César Sosa Yeladaqui

Jesus Antonio Alvarez Cedillo

Enrique Alfonso Sánchez Montellano

Monica Aisha Aguilar Gómez

Diseño & Imágen SL

Edgar Guerra Rey

Josue Gutierrez Hernández

Ivan Alfredo Zenteno Aguilar

Eyden Barboza Barela

Héctor Leal Morales

Miriam ElizBeth Cacique Acevez

Equipo de Corrección SL

Martín Trejo Chávez

Luis Fernando Peniche Novelo

Vanessa Johanna Salcedo Weeber

Alejandra Fierro

Caricatura SL

Humberto Morales

www.revista-sl.org

contenido

Software Libre Hecho Revista

```
revistasl:~#nmap -A -O -p0-37 Revista SL
```

```
Starting Nmap 4.20 ( http://insecure.org ) at 2007-07-12 17:21 CDT
```

```
Interesting ports on Revista SL (index):
```

```
Not shown: 2 closed ports
```

PORT	STATE	SERVICE
5/tcp	OPEN	¿Por qué uso VIM?
7/tcp	OPEN	Pequeña "vista" a Motion
8/udp	OPEN	Una visión general de nuevas distribuciones a partir de Knoppix
10/tcp	OPEN	Portatiles para mejorar la educación
12/tcp	OPEN	Modelas = comunicar
13/tcp	OPEN	Instalación de Xilinx ISE Explorer
17/tcp	OPEN	CIDEFORM: Procesamiento de imagenes usando morfología con GAMBAS
22/udp	OPEN	Crea tu sitio wen en 10min usando Django
25/tcp	OPEN	Introducción al desarrollo de interfaces
29/tcp	OPEN	La Columna MySQL
32/tcp	OPEN	La Columna PL
34/tcp	OPEN	RevistaSL: ¿Proyecto de divulgación o locura colectiva?
39/tcp	OPEN	En entrevista: Richard Stallman
45/tcp	OPEN	Tips & tricks SL

```
Device type: Revista de Software Libre
```

```
Running: Revista SL
```

```
OS details: Software Libre Hecho Revista V.: 10
```

```
revistasl:~#
```

Nervios & Casaca

No cabe duda, entre mas consumimos tecnología, mas bites eructamos al hablar, por Humberto Morales Sánchez (HUMO)



¿Por qué uso VIM?

Andres Bernardo Vargas Rodríguez
andres.vargas@revista-sl.org

Vim es una derivación (fork) del famoso editor para servidores UNIX, Vi. VIM es el Vi mejorado. Su creador Bram Moolenaar recién compró su computadora Amiga, la cual tenía un Sistema Operativo distinto a los UNIX. Bram quería seguir usando el Vi de los servidores UNIX por lo que empezó a desarrollar su propia versión de Vi para el Sistema Operativo AMIGA.

Que tal amigos lectores de está su RevistaSL. Esta vez les traigo una pequeña revisión y comentarios sobre el uso de unos de mis programas favoritos: VIM

VIM fue para mi un editor que me atrajo mucho por su sintaxis similar al lenguaje C. Yo venía de usar Turbo C de Borland con su editor azul horrible en aquellos tiempos al inicio de mi carrera.

Uso vim por muchas razones, por ejemplo, el resaltado de código y el auto-completado.

Split/tabs

Vim como editor de terminal puede utilizar pestañas, nosotros no estamos acostumbrados a ver algo así en un editor de terminal por eso cuando las usé por primera vez me pareció algo genial.

Para utilizar las pestañas tenemos que entender como funcionan. Cuando usamos Vim en la consola y abrimos un archivo; Vim crea un buffer (podríamos decir un archivo temporal) del archivo en el que se realizan los cambios y modificaciones y cuando se guarda el archivo se guardan también los cambios. Este buffer es lo que se nos muestra en pantalla.

Primero abrimos el vim con un archivo.

```
$ vim foo.txt
```

Abrimos una pestaña presionando ESC para ir al modo comando y tecleamos

```
:tabnew
```

Con eso se nos abre la opción para meter otro buffer a Vim. Para cambiar entre pestañas utilizamos

```
:tabnext
```

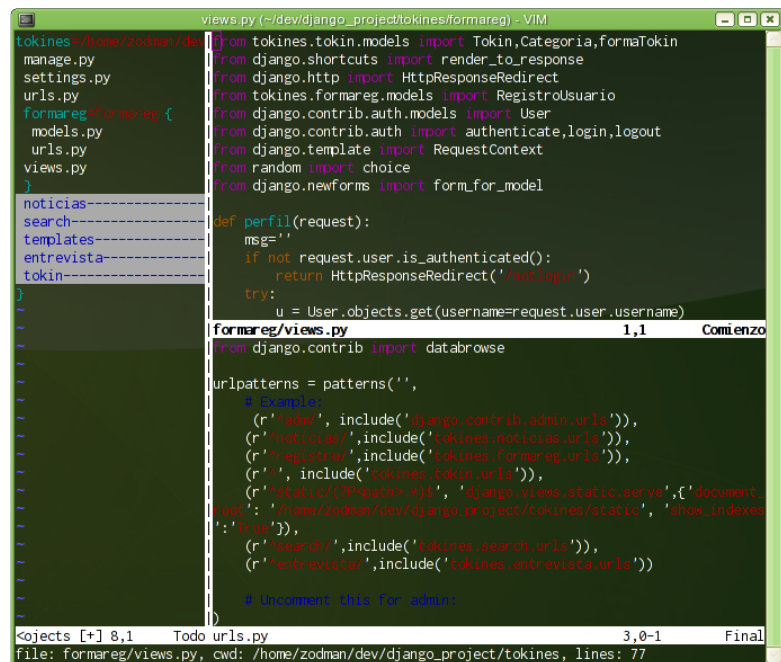
o con

```
:tabprevious
```

Para cerrar el tab usamos

```
:tabclose
```

Pero todo esto se nos haría mas fácil si agregamos a nuestro ~/.vimrc o a tu /etc/vim/vimrc (en Debian y Ubuntu)



Uso de pestañas en Vim

```
:nmap Z :tabprevious<cr>
:nmap X :tabnext<cr>
:map Z :tabprevious<cr>
:map X :tabnext<cr>
```

Con esto presionas SHIFT+z o SHIFT+X puedes cambiarte de manera rápida entre pestañas.

Y si aun quieres más potencia tu actual buffer, puedes dividirlo entre 2 o más divisiones dentro, usando lo siguiente:

```
:split
```

Para una división horizontal

```
:vsplit
```

Para una división vertical

Y con CONTROL+w+w

Cambias entre splits. ¡Cómo me encanta! Puedes tener y ver varios archivos al mismo tiempo sin moverte de buffer.

Auto-completar bajo conocimiento

Esto es lo que más me gusta de vim: puede auto-completar lo que quieras.

Esto quiere decir que si mientras tecleas un código vim puede auto-completar algo que ya hayas tecleado antes o si te regresas y quieres auto-completar algo que escribiste después.

Esto se hace así

Escribimos digamos código en Python (¿Por qué Python? Porque me gusta).

```
def sumar_dos_numeros_x_y(x,y):  
    return x+y
```

y quiero llamar la función, escribirlo es muy largoooo así que si estamos debajo de la definición de esa función mientras tecleamos(en modo edición, no hay que presionar ESC) presionamos CONTROL+n

Ejemplo teclea lo siguiente:

```
sumCONTROL+n
```

Y se auto-completa el nombre de la función (sumar_dos_numeros_x_y). Esta característica es como el auto-completar de cualquier IDE comercial. Y si tienes varias palabras que empiecen con las mismas letras, vim te presentará una lista de opciones para que escojas la correcta.

Folding

El folding en vim es algo muy interesante, es lograr contraer y expandir funciones o bloques de texto, haciendo de tu código espagueti a la bolognesa, un espagueti alfredo, con tiras de pollo empanizado.

Primero habilitamos folding

```
:set foldenable  
:set foldmethod=marked
```

(puedes agregarlo a tu vimrc)

Para empezar puedes crear foldings contando líneas de código presionando ESC+z+f+(número de líneas)j

Esto hace que las marcas de folding se agreguen ({{{ }}}) Estas determinaran el número de líneas que debe contraerse.

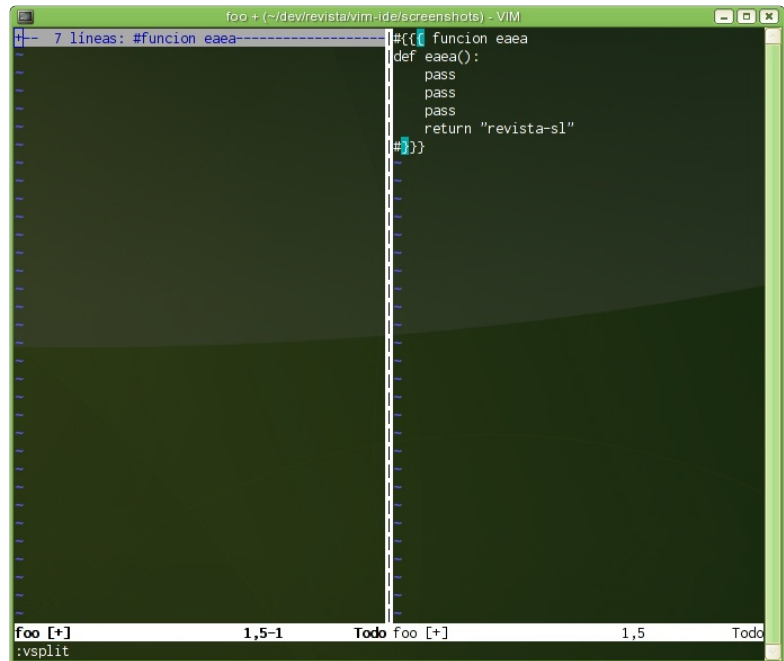
Para abrir el folding con ESC+z+o

Y para cerrar el folding con ESC+z+c

Para no estar contando las líneas de código, nada más agrega las marcas al código y después cierra el folding para ver que funcione

```
#{{{ funcion eaea  
def eaea():  
    pass  
    pass  
    pass  
    return "revista-sl"  
#}}}
```

Hay mucha información interesante en la red, así que si has pensado que vim no puede hacer eso que tu quieres, ¡no te imaginas! si buscas en la red, seguro encontrarás que sí lo puede hacer.



Uso de "folding" en Vim

Para buscar más información les dego uns ligas interesantes sobre Vim

<http://www.linux.com/articles/114138>

<http://www.pablasso.com/2007/05/13/3-caracteristicas-indispensables-de-vim>

<http://vim.org>

<http://tinyurl.com/2p6osu>

<http://tinyurl.com/2plbae>

Bueno estas sólo han sido algunas cosas, por que hay más... Se despidió su amigo Zodman.

```
:quit 
```

Vídeo vigilancia bajo Linux: Pequeña “vista” a Motion

Gonzalo Javier González Rodríguez
gonzalo.gonzalez@revista-sl.org

Motion, es un software que monitorea señales de vídeo desde una o más cámaras, con la particularidad de detectar significantes cambios en la imagen del vídeo; en pocas palabras, detecta movimiento. Escrito en C y creado para el Sistema Operativo Linux, esta herramienta es para la línea de comandos y su salida puede ser a archivos jpg, ppm o secuencias de vídeo mpeg.

Buscando opciones para crear un sistema de vídeo vigilancia para la empresa en la que laboro, me topé en la web con un software que se llama “Motion”, un software que me permitiría grabar vídeo y almacenarlo para su consulta posterior.

Entre sus características está el uso de una base de datos MySQL/PostgreSQL para guardar los eventos, compatibilidad con dispositivos V4L y cámaras ethernet, manejar múltiples dispositivos de vídeo, envío de un sms o e-mail cuando se detecte movimiento, streaming en vivo de vídeo, bajo consumo de CPU y de memoria, entre otras.

En la página web de Motion, podemos consultar la lista de dispositivos, tarjetas controladoras y cámaras v4l o ethernet, compatibles.

Al estudiar las características de Motion, entendí que este software es lo que estaba buscando o por lo menos me arrojaría los resultados que estaba buscando para el sistema de vídeo vigilancia.

Ya tenemos el Software, nos hace falta la cámara para hacer una prueba de campo. Nos decidimos por una D-Link DCS-900, con una resolución máxima de 640x480, 30 cuadros por segundo, diferentes niveles de compresión, acceso vía web, vídeo en vivo, panel de control, control de acceso de usuarios, entre otras características.

Para instalar motion, en mi caso bajo Ubuntu, fue mediante la ejecución de:

```
# apt-get install motion
```

Y luego instalé el “codec” para poder grabar vídeo en mpeg:

```
# apt-get install ffmpeg
```

El archivo de configuración de Motion, /etc/motion/motion.conf, está dividido en secciones para su mejor entendimiento. Si no encontramos este archivo podemos copiar un ejemplo de configuración de /usr/share/doc/motion/examples/motion-dist.conf.

Al modificar el archivo de configuración, verificamos que Motion arranque como “daemon”, le especificamos el origen de la señal de vídeo en “Capture device options”: netcam_url http://cam-01.dqr.com.mx/video.cgi?showlength=2

Lo anterior especificamos el origen de la señal de vídeo, para el caso de la D-Link DCS-900 el streaming lo obtenemos desde http://ip-camara/video.cgi?showlength=2.

En “Motion Detection Settings”, podemos configurar el número de píxeles que cambia de un frame a otro para considerar movimiento. En “Film (mpeg) File Output - ffmpeg based”, especificamos el codec a utilizar para la grabación de los vídeos. Las opciones para especificar el directorio de destino y el nombre de los archivos que se guardaran, están en “Target Directories and filenames For Images And Film”.

Ejecutando lo siguiente:

```
#motion -c /etc/motion/motion.conf
```

Podemos arrancar el “daemon” si nuestro archivo de configuración está correcto.

Un script de inicio y parada de nuestro “daemon”, lo podemos encontrar en /usr/share/doc/motion/examples/motion.init-Debian.

Verifiquemos si nuestro motion está funcionando, al pasar delante de la cámara y en nuestro servidor vemos el archivo de vídeo guardado.

Les dejo algunas referencias para que profundicen:

Página de Motion:

<http://www.lavrsen.dk/twiki/bin/view/Motion/WebHome>

Dlink DCS-900

<http://www.dlink.com.mx/products/?prodName=DCS-900> 

En la actualidad es posible disponer de varios sistemas operativos cuya licencia no restringe su uso a unos cuantos usuarios, esto en particular, es gracias a la existencia de la licencia GNU1 GPL2 (GNU General Public License)

Como es sabido GNU/Linux es el sistema operativo GNU con el núcleo Linux. De esta fusión se han desprendido una innumerable cantidad de variantes conocidas como distribuciones. Cada distribución ha sido pensada para un propósito particular; algunas son de uso más general que otras y se debe seleccionar una en función de que sea la más adecuada para el uso que se le piense dar, por lo tanto, no existe distribución buena o mala, sino “apropiada”.

Al principio, las distribuciones eran compartidas en el espíritu del software libre y bajo la licencia GNU GPL-, mediante discos magnéticos o a través de la Internet. Actualmente, la gente sigue compartiendo sus distribuciones bajo la misma licencia, pero también mediante dispositivos de almacenamiento como los discos compactos en formato CD o DVD e incluso en memorias USB.

Desde el punto de vista de instalación, existen las distribuciones que necesitan ser instaladas para funcionar y las que funcionan sin necesidad de ello, conocidas como versión “Live”.

Dentro de las diversas distribuciones que funcionan de forma “Live”, está la llamada KnoppixGNU/Linux LiveCD, una de las principales distribuciones en su tipo disponible en un CD, fue creada por el ingeniero alemán Klaus Knopper³. Su popularidad radica en la gran cantidad de dispositivos que puede identificar y configurar de manera automática, la diversidad de usos que se le pueden dar, así como en su “descompresión transparente y al vuelo”⁴. Knoppix es conocida como “el LiveCD original basado en Debian”.

En otras palabras, Knoppix es un sistema operativo contenido en un disco el cual funciona tan pronto es insertado en una unidad de CD, una vez reiniciada la máquina que lo contenga, sin necesidad de instalarlo, siempre y cuando se pueda arrancar desde esta unidad.

Knoppix es una distribución de propósito general porque contiene aplicaciones para distintas tareas (e.g. desarrollo de software, Internet, multimedia o gráficos) por lo que es probable que las que se necesiten estén incluidas, sin embargo, en ocasiones ésta no es la situación. Gracias a las propiedades del software libre y a los scripts que contiene (los cuales no serán expuestos en el presente artículo), es posible generar una nueva distribución derivada de ésta.

Dicho de otra forma, Knoppix, es adaptable a propósitos particulares, por lo que de ella se pueden derivar distribuciones con programas y un aspecto que cubran las necesidades de un determinado usuario o grupo de usuarios, hasta donde lo permita la versión de Knoppix con la que se trabaje.

Knoppix es portátil ya que puede almacenarse y ejecutarse en medios como el CD, el DVD, la memoria USB o el disco duro.

Son tres las etapas básicas del procedimiento de recreación de la distribución y cuatro opcionales. A continuación, se describen brevemente estas etapas.

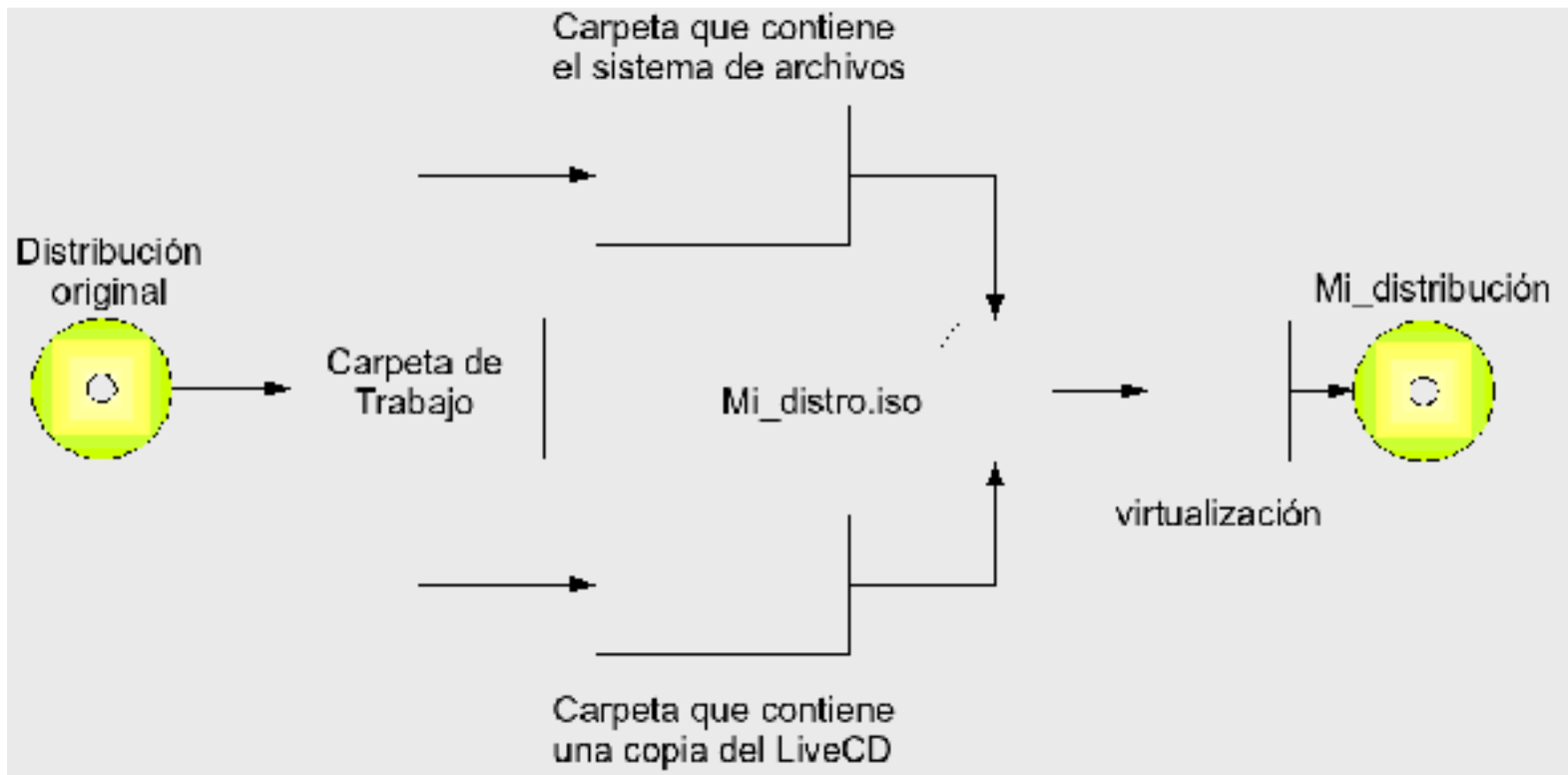
Etapas básicas.

1. **Preparación:** Se copian desde la distribución original todos los archivos necesarios.
2. **Personalización:** Se instalan y desinstalan programas, se personaliza el aspecto de la nueva distribución.
3. **Compresión:** Se crea un nuevo archivo comprimido que representa la nueva distribución lista para ser almacenada en algún dispositivo portátil.


Etapas opcionales.

4. **Emulación:** Se emplea alguna herramienta de virtualización de sistemas operativos como “qemu” para verificar que el sistema funcione correctamente antes de ser transferido definitivamente a un medio de almacenamiento.
5. **Grabación:** Se genera una copia preliminar de la nueva distribución para ser probada en el medio correspondiente.
6. **Personalización del nuevo escritorio:** Dado que al instalar programas, el escritorio tiende a actualizarse y por tanto a adquirir nuevas propiedades, es recomendable personalizar el escritorio a partir de la copia preliminar.
7. **Compresión final:** Se copian los archivos correspondientes a la particularización del nuevo escritorio en la carpeta correspondiente, se genera el archivo final que corresponde a la nueva distribución. se recomienda emular antes de transferir el archivo a algún dispositivo portátil de almacenamiento.

El procedimiento descrito se empleó para Knoppix, pero es semejante para otras distribuciones cuando se trata de trabajar a partir de una existente que funciona como LiveCD, como Ubuntu por ejemplo. En la figura se resume el proceso. Como se puede observar, existen dos carpetas que se crean a partir de la distribución original.



Proceso general para realizar una nueva distribución a partir de una existente.

En una se encuentra el sistema de archivos descomprimido y en la otra una copia de todo el disco; ambas dentro de una carpeta de trabajo contenida en disco duro. En la primera carpeta se instalan y desinstalan programas y se modifican algunos archivos de configuración mientras que en la segunda se agrega información adicional como por ejemplo una página html que describa nuestra distribución y que deseamos se abra al inicio de la sesión. Ambas se conjuntan para generar un archivo con extensión *.iso que es la imagen que será grabada en un CD. Para verificar que todo funciona como esperamos, es recomendable virtualizar nuestro sistema empleando este archivo, hecho esto solo falta grabarlo y volver a compartirlo como lo hizo el primer autor. 

Algunas ligas de interes:

- [1] Hacking Knoppix, Granneman Scott, JohnWiley & Sons, 2006 (336 páginas).
- [2] Anónimo, "Knoppix Remastering Howto", Knoppix.net, 15 de Noviembre, 2007: [En línea], Disponible: http://www.knoppix.net/wiki/Knoppix_Remastering_Howto.
- [1] <http://www.gnu.org/>
- [2] <http://www.gnu.org/licenses/licenses.html>
- [3] http://en.wikipedia.org/wiki/Klaus_Knopper
- [4] <http://en.wikipedia.org/wiki/Knoppix>

Portátiles para revolucionar la educación

María del Pilar Sáenz R.
pilar.saenz@colombialpc.org

Gloria Patricia Meneses P.
gloria.meneses@colombialpc.org

El proyecto de Un Portátil por Niño (OLPC, One Laptop Per Child) es una realidad. Ya se realizaron pilotos en algunos países de Latinoamérica. Algunos de estos países adoptaron los portátiles como parte de su política educativa.

La academia y la industria se unen para crear un proyecto de gran impacto en la educación, un computador portátil de aproximadamente US\$200. Este está siendo vendido a los gobiernos de cada país para iniciar una revolución educativa. Un cambio fundamental, que permitirá a las nuevas generaciones de los países en vías de desarrollo ser partícipes de la revolución tecnológica que estamos viviendo.

El MIT Media Labs con el apoyo de Google, AMD, Red Hat, News Corp, Brighstar y el PNUD (Programa de las Naciones Unidas para el Desarrollo) ha desarrollado un proyecto llamado *Un Portátil Por Niño (OLPC, One Laptop Per Child)*. El objetivo de este proyecto es entregar a cada niño un computador portátil que le permita experimentar, expresarse, utilizarse como ventana al mundo y como herramienta con la cual pensar.

Características de Hardware y diseño

El laptop de OLPC, la XO, es agradable a la vista, liviana, pequeña, resistente a golpes y a la lluvia. Tiene cierres resistentes al polvo y a la humedad. Sus dimensiones son 242mm x 228mm x 30mm y su peso menor a 1,5 kg. Para la conexión a internet y a la red en malla (mesh) utiliza una tarjeta Marvell Libertas, compatible con 802.11b/g. Tiene antenas dobles coaxiales ajustables y giratorias. Posee integrada una Cámara de video con resolución de 640x480 y frecuencia de 30 cuadros por segundo. La cámara soporta la desactivación del control automático de ganancia y el balance automático de color, permitiendo su utilización como un sensor fotométrico para aplicaciones educativas. Aunque tiene micrófono y parlantes internos también tiene una entrada estándar de micrófono con un conector de 3.5mm. Esta entrada también se puede utilizar como un sensor de CC con 2.5 V de polarización (impedancia de 3 kilo ohm) para pulsadores, teclas, sensores resistivos, etc. o como sensor de CC sin polarización para mediciones de voltaje (0 - 3 V). Su batería es de larga duración y bajo consumo, con un ciclo de vida de mínimo 2,000 ciclos de carga/descarga, una configuración de 4 celdas que provee 6V en serie y una capacidad máxima de 22 watt-hora. Posee una pantalla de cristal líquido de 7.5" (19.05 cm) con un modo dual (Blanco/negro de alta resolución en modo reflectivo y Color con muestreo quincunx en modo transmisivo) que permite leer a plena luz del sol. El área visible es de 152,4 mm x 114,3 mm con una resolución de 1200x900 (200 dpi). El consumo de la pantalla es de 0.1 Watt sin luz de fondo y entre 0.2 y 1.0 Watt con iluminación de fondo. Cuenta con memoria RAM de 256M, memoria flash de 1G, 3puertos USB, un procesador AMD Geode LX-700 de 433MHz y ranura para tarjetas de expansión. El teclado es de 70+ teclas con un recorrido de 1.2mm. Está sellado por una membrana de goma. El teclado se puede adaptar a diferentes idiomas, entre ellos: inglés, español, portugués, tailandés, árabe y urdu entre otros. Tiene un touchpad que soporta modo de entrada de escritura. La optimización de recursos fue el criterio de diseño de hardware más importante para brindar a los niños la mejor herramienta educativa,



Equipo del proyecto OLPC

debido a que muchos de ellos no cuentan con las condiciones ideales en sus hogares ni en sus escuelas.

Características de Software

Todo el software que utiliza la XO es libre y gratuito. Desde la BIOS hasta el sistema operativo tienen licencia GPL. La interface gráfica de usuario es Sugar, que se encarga de la interacción entre el humano y la XO. Controla desde los eventos de teclado y mouse hasta la generación del entorno gráfico para las actividades que se ejecutan en la XO. En las actividades que se pueden realizar en el portátil existe procesador de textos, programas de dibujo, música, browser, lector de noticias, diario y juegos diseñados especialmente para esta máquina. También tiene ambientes de desarrollo como Pippy, una actividad para explorar python en la XO, donde incluso es posible leer el código de varias actividades de la misma XO. Otro ambiente de desarrollo es Etoys, desarrollado en Smalltalk, basado en una imagen de squeakland.org que corre con la última máquina virtual de Squeak.

El uso de software libre permite que los gobiernos y escuelas no tengan que pagar licencias de software. Pero más importante aún, hace que la XO sea una máquina "abierto" que admite hacer mejoras y nuevas construcciones del sistema operativo, el software y los contenidos. De

esta forma, el portátil de OLPC se puede adaptar a las necesidades de cada centro educativo o país. Sin desconocer la cultura e historia que le son propias.

Propuesta Pedagógica

OLPC es un proyecto sobre la transformación de la educación. Sobre dar a los niños que no lo tienen la oportunidad de aprender. Pero no aprender de cualquier forma. Tanto el Hardware como el Software están diseñados alrededor de la idea de compartir el conocimiento y de construirlo a medida que se descubre, se utiliza y se modifica. La propuesta pedagógica que subyace a la olpc está sustentada en el trabajo de Seymour Papert y la corriente del construccionismo que él creó a partir de su trabajo con Jean Piaget. Según el construccionismo, una persona aprende al interactuar dinámicamente con el mundo físico, social y cultural en el cual está inmerso. El conocimiento es un fruto del trabajo propio y el resultado de las vivencias del individuo. Por tanto el niño desarrolla competencias, es decir aprende haciendo, pero también desarrolla su propio estilo de aprendizaje y un interés en algunas áreas particulares del conocimiento. Esto lo vuelve un pequeño experto que es capaz de compartir su experiencia y su conocimiento a otros iguales a él.

Para poder potenciar al máximo las capacidades de los niños, la xo del proyecto olpc posee características de software y hardware que facilitan la creación de comunidades de práctica, grupos de estudio e intercambio de información soportadas en primera instancia por la red en malla (Mesh) que se establece de forma automática entre las xo. Esta red y el acceso a internet, posibilita interacciones entre ellos a nivel local, entre niños de una misma aula de clase, y a niveles superiores en su misma escuela, ciudad o país.

Estas características soportan un real aprendizaje colaborativo, que entre muchas otras ventajas, logra convertir a los niños en protagonistas de su propio proceso de formación y no sólo en receptores pasivos de la información que les entregan los maestros.



Niñas beneficiadas por el proyecto en Arahua



Niños beneficiados en Galadima

En cuanto a su implementación, mayoritariamente se ha propuesto el proyecto a los gobiernos de los países en vías de desarrollo. Específicamente, en Latinoamérica, Uruguay, Brasil y Perú cuentan con programas piloto en Villa Cardal, Porto Alegre y Arahua, respectivamente. Uruguay fue el primer país en hacer un pedido oficial de portátiles para sus niños, de los cuales, los primeros fueron ya entregados en Florida y Villa Cardal. Perú también realizó un pedido de máquinas y además del piloto en Arahua ya repartió equipos en La Jota, en Tumbes.

En los demás países de la región se han entablado diálogos con los gobiernos nacionales y en algunos casos como Argentina, Chile y Colombia hay grupos o personas que como colaboradores voluntarios están desarrollando contenidos para las laptops y evaluando la usabilidad de estos. En Argentina hay dos grupos de desarrollo, Tuquito y Ututo. En Colombia, el grupo Olpc en Colombia que hace trabajo de divulgación del proyecto y desarrollo de contenidos para el mismo. Y en Chile, el grupo de Un Computador por Niño, que buscan promover el acceso permanente y libre a computadores por los niños de su país.

En otras regiones del mundo también se están desarrollando pruebas: Ulaanbaatar en Mongolia, Khairat en la India, Ban Samkha en Tailandia y Galadima en Nigeria.

De ser implementado de forma satisfactoria este proyecto permitiría a países en vías de desarrollo reducir la brecha digital con los países desarrollados. Potenciaría la apropiación tecnológica y disminuiría la dependencia económica y cultural con los países industrializados.

Información adicional sobre el proyecto en

<http://laptop.org/index.es.html>

<http://colombialpc.org>

Si está interesado en el proyecto contáctenos

contacto@colombialpc.org 

Estado del proyecto olpc en Latinoamérica

El proyecto olpc desde su creación ha contado con el apoyo no solo de empresas e instituciones sino de un nutrido grupo de voluntarios que contribuyen con su trabajo a la implementación y puesta en marcha del proyecto. Existe un grupo de soporte alrededor del mundo, que está mejorando las características de hardware y software de estos portátiles.

En el mundo del Software Libre, en donde uno de los principios básico es el acceso libre al código de los productos, a fin de que la comunidad y la tecnología evolucionen sin las trabas del "secreto comercial" o del "copyright", tenemos un reto importante: comunicar.

¿Por qué es un reto comunicar? La esencia de un desarrollo maduro de software es que todos los involucrados -desde el usuario final hasta el desarrollador, pasando por el arquitecto, el analista, el personal de calidad, etcétera.- comprendan el objetivo y funcionalidad del producto. Y en dicha esencia, la comunicación formal juega un papel sumamente crítico. Ya que se deben de establecer vías más allá del lenguaje verbal que garanticen la igualdad y entendimiento de los términos entre los involucrados.

Por otra parte, uno de los retos a los que nos enfrentamos, como desarrolladores, al intentar involucrarnos en algún proyecto de Software Libre, es a la falta de elementos formales que nos permitan entender rápidamente los detalles técnicos del funcionamiento de un producto, dejando esta labor (el involucramiento en los proyectos de desarrollo de Software Libre) a aquellos que tienen bastante tiempo, paciencia y capacidad técnica de entrar directo al código de alguna aplicación y poco a poco entender los recovecos del "código ajeno", y en su caso modificarlo, mejorarlo, corregir algún bug, e incluso hacer algún producto derivado. En cualquiera de los casos se requiere de una curva de aprendizaje en donde el esfuerzo, la mayoría de las veces, está en entender directamente el código.

Este reto, en el mejor de los casos, es usualmente atendido por los desarrolladores, al incluir extensos comentarios en los segmentos de código que modifican o crean. Sin embargo sigue habiendo un espacio entre el usuario final y el equipo de desarrollo en donde los canales de comunicación, en el peor de los casos, son bastante arcaicos, como la típica frase o excusa "yo pensé que el usuario necesitaba esto".

Estas son áreas de oportunidad que el modelado atiende.

Modelar es comunicar. Modelar es adoptar un estándar de elementos gráficos, para detallar y comprender mejor un problema. Modelar facilita el entendimiento del producto entre los involucrados de un proyecto, además de que nos ayuda a documentar el producto de software que se desarrolla. Documentar por documentar es inútil, lo que las diversas técnicas de modelado buscan es expresar en un "lenguaje común" para distintas audiencias (usuarios, desarrolladores, administradores de proyectos) los "qué" y los "cómo".

Cuando hablamos de "modelo" hablamos de abstracción del objeto real, y de su comprensión gracias a la precisión y sencillez con que se exprese. Un modelo nos ayuda a validar que las necesidades reales del negocio las satisfaga un producto de software, nos permite inclusive "simular" lo que sucederá dentro de los sistemas a fin de que los equipos de desarrollo sepan claramente que codificar.

Existen distintas estrategias de modelado estándar, dependiendo del nivel de abstracción (Alto Nivel, Ej. Procesos de Negocios, Requerimientos, etc. vs Bajo Nivel, Ej. Componentes, Código, etc.) y de

las necesidades específicas del proceso de desarrollo de software a atender: Desde las etapas tempranas del análisis (Ej. Modelado de Procesos de Negocios, BPMN -Business Process Modeling Notation-), hasta elementos más identificados según ciclo de vida utilizado (Ej. Lenguaje Unificado de Modelado, UML -Unified Modeling Language-, Diseño de Bases de Datos, ER -Entidad/Relación- etc.) o arquitectura a utilizar (SOA, Capas, MDA -Model Driven Architecture, MDD -Model Driven Development- etc.)

El valor que nos da el modelar nuestras aplicaciones es comunicar. Si estamos por liberar una aplicación como Software Libre y queremos invitar a otras personas a involucrarse en nuestro proyecto, resultará mucho más fácil si contamos con documentos estándar que permitan entender el producto. Si retomamos el concepto de "la curva de aprendizaje", esta será siempre menor si hay un modelo documentado, en lugar de ir directamente a entender todo el código. Inclusive, los involucrados en un proyecto pueden tomar y/o mejorar el modelo para hacer crecer los productos de forma ordenada y con mayor alcance que una simple lista de tareas.



Me gustaría plantear el caso teórico de una empresa que esta en la disyuntiva de elegir entre un ERP (o cualquier tipo de sistema de amplio impacto organizacional) de la comunidad de Software Libre o uno propietario. El proceso de implantación de un sistema de estas características conlleva muchos cambios, tanto organizacionales (procesos de negocios) como adaptaciones de la herramienta. Entonces, si tuvieran que elegir (dadas las condiciones de premura y competitividad de los negocios actuales, costos y tiempos de desarrollo / adaptación) si tomar un ERP con escasa documentación o uno con amplia documentación estándar, ¿cuál elegirían? ¿será cierto que los ERP propietarios están ampliamente documentados?

Me gustaría conocer sus experiencias, sobre cómo ha sido el proceso de involucramiento con los actuales proyectos de Software Libre a los que se han acercado. En siguientes artículos me gustaría comentar un poco de las técnicas de modelado anteriormente descritas y las herramientas que la misma comunidad de Software Libre nos provee para hacer de nuestros desarrollos productos más profesionales.

Instalación del Xilinx ISE Explorer en Ubuntu

Jesús Antonio Alvarez Cedillo
jesus.antonio@revista-sl.org

La herramienta ISE de Xilinx, es un entorno integrado de aplicaciones que se utiliza para sintetizar diseños y prototipos electrónicos con el fin de generar los bitstream para su descarga en FPGAs, fue portada a Linux. Sin embargo, oficialmente sólo se ofrece soporte para distribuciones Red Hat Enterprise.

¿Que es un FPGA?

Una FPGA (del inglés Field Programmable Gate Array) es un dispositivo semiconductor que contiene bloques de lógica cuya interconexión y funcionalidad se puede programar. La lógica programable puede reproducir desde funciones tan sencillas como las llevadas a cabo por una puerta lógica o un sistema combinacional hasta complejos sistemas en un chip. Ver figura 1.



Figura 1. Fpga SPARTAN de XILINX

Las FPGAs se utilizan en aplicaciones similares a los ASICs, sin embargo son más lentas, tienen un mayor consumo de potencia y no pueden abarcar sistemas tan complejos como ellos. A pesar de esto, las FPGAs tienen las ventajas de ser reprogramables (lo que añade una enorme flexibilidad al flujo de diseño), sus costes de desarrollo y adquisición son mucho menores para pequeñas cantidades de dispositivos y el tiempo de desarrollo es también menor.

Ciertos fabricantes cuentan con FPGAs que sólo se pueden programar una vez, por lo que sus ventajas e inconvenientes se encuentran a medio camino entre los ASICs y las FPGAs reprogramables.

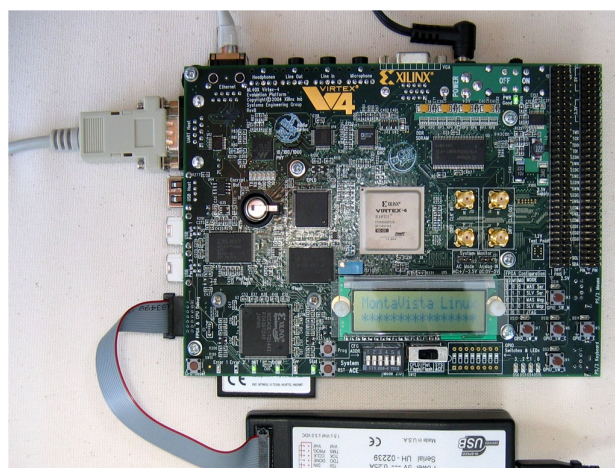
Históricamente las FPGAs surgen como una evolución de los conceptos desarrollados en las PLAs y los CPLDs.

XILINX ISE NAVIGATOR

Existen dos modalidades de Desarrollo, estas son:

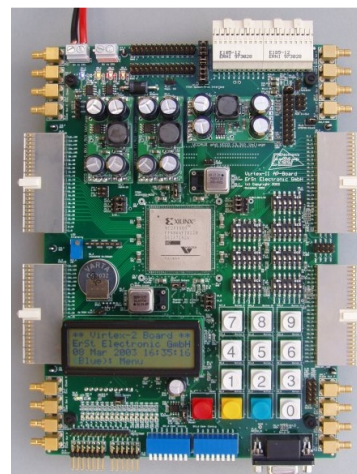
a) Programación virtual. - El dispositivo para esto usa una tablilla de entrenamiento, donde se puede modificar el diseño y la programación del dispositivo hasta que éste se encuentre óptimo para su inserción definitiva en el CHIP, en la siguiente figura se muestra una tablilla de entrenamiento, nótese que en si es un completo laboratorio con salida

VGA, salidas SERIALES, USB, Salida de RED, display LCD y led de prueba. La siguiente Figura Muestra una tablilla de entrenamiento para el Programa ISE.



Aquí se muestran las conexiones de Proyectos de Pruebas.

b) Programación Real. - En este tipo de programación el programa antes realizado y probado es insertado en la programación del dispositivo para siempre, una vez "quemado", el dispositivo no puede ser utilizado de nuevo y este es insertado en una aplicación electrónica. La siguiente figura muestra el circuito en una aplicación electrónica.



La figura Muestra una tablilla de entrenamiento para el Programa ISE

Ahora explicaremos cómo es posible instalar el ISE y el EDK 7.1.

Requisitos previos

Instalar el paquete libmotif3, que está en non-free. (apt-get install libmotif3).

Si no estuviese instalado este paquete, aparecería un error como el siguiente:

```
/cdrom/platform/lin/xilsetup: error while loading
shared libraries: libXm.so.3: cannot open shared
object file: No such file or directory
```

Instalar el paquete libcurl3 (apt-get install libcurl3).

Ejecutar los siguientes comandos, como root:

```
# ln -s /usr/lib/libcurl.so.3 /usr/lib/libcurl.so.2
# ldconfig
```

Si no están establecidos o no está instalado el paquete libcurl3 aparecerá un error como el siguiente:

```
/cdrom/platform/lin/xilsetup: error while loading
shared libraries: libcurl.so.2: cannot open shared
object file: No such file or directory
```

Instalación del CD 1

Una vez cumplidos los requisitos previos abrimos un terminal de gnome y nos metemos como root:

```
$ su
Password:
#
```

Introducir el CD 1: "ISE 7.1i – Linux 32-bit. Disc 1 of 3. Design Environment and Documentation". Es posible que el CD se automonte (en el gnome que tengo instalado se automonta nada más introducirlo). Si es así hay que desmontarlo y montarlo manualmente con permisos de ejecución, o de lo contrario no nos dejará ejecutar el script de arranque setup.

Ejecutar el comando mount para ver el estado actual de cómo está montado el CDRROM.

```
# mount
[...]
/dev/hda on /media/cdrom0 type iso9660
(ro,noexec,nosuid,nodev,user={usuario})
```

En este caso el CD se ha automontado y el usuario no tiene permisos de ejecución. En este caso hay que desmontar el CD y montarlo como root:

```
# umount /media/cdrom0
# mount /dev/hda -t iso9660 /cdrom
mount: dispositivo de bloques /dev/hda está
protegido contra escritura; se monta como sólo
lectura
```

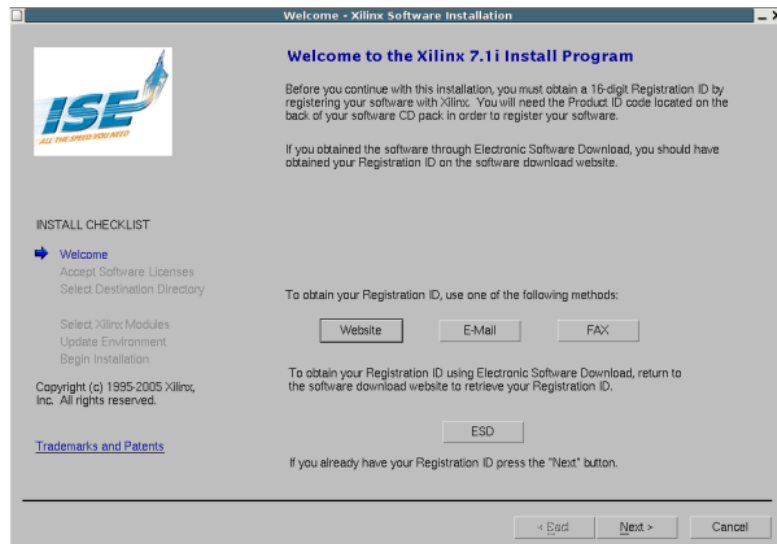
En mi ordenador el CD es el dispositivo /dev/hda y lo estoy montando sobre el directorio /cdrom. (un enlace simbolico a /dev/cdrom0) Si ahora ejecutamos el comando mount aparecerá lo siguiente:

```
# mount
[...]
/dev/hda on /media/cdrom0 type iso9660 (ro)
```

Ahora ya sí tenemos permisos de ejecución.

```
Exportar el display:
/home/juan# export DISPLAY=:0
Entrar en el CD y ejecutar el script setup:
# cd /cdrom
# ./setup
/cdrom/platform/lin/bin/lin
Wind/U Error (294): Unable to install Wind/U ini
file (/cdrom/platform/lin/data/WindU).
See the Wind/U manual for more details on the
".WindU" file and the "WINDU" environment variable
```

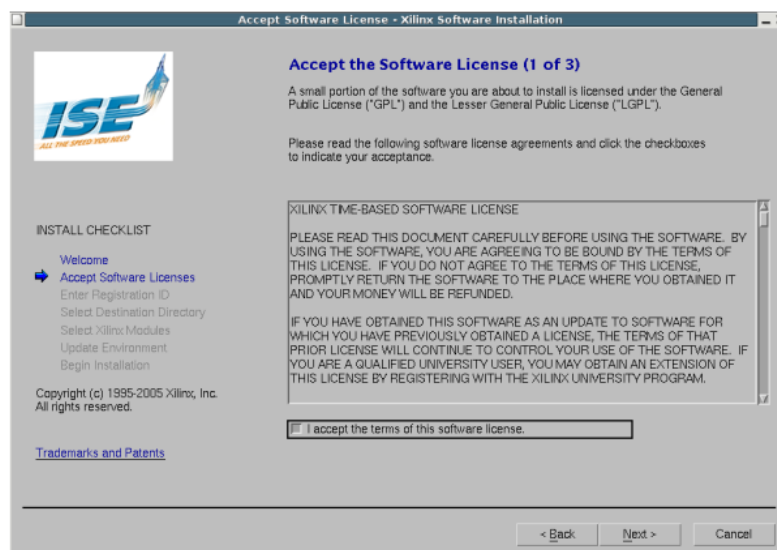
Después de unos segundos aparecerá la ventana de bienvenida del ISE 7.1:



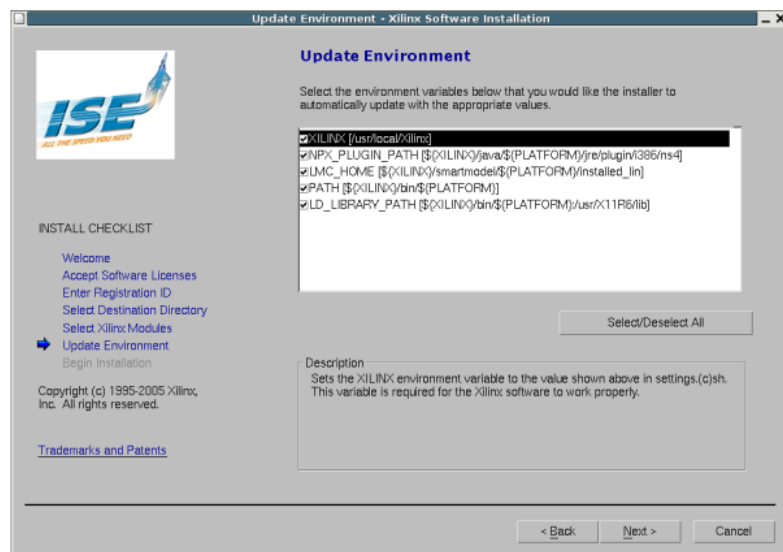
Si no estuviese exportado el display, aparecería el siguiente mensaje de error:

```
montu:/cdrom# ./setup
/cdrom/platform/lin/bin/lin
Wind/U X-toolkit Error: wuDisplay: Can't open
display
```

Pinchar en Next. Aparecerán tres pantallas con las licencias, que hay que aceptar y pinchar en Next. Aquí pongo un pantallazo de la primera de ellas:

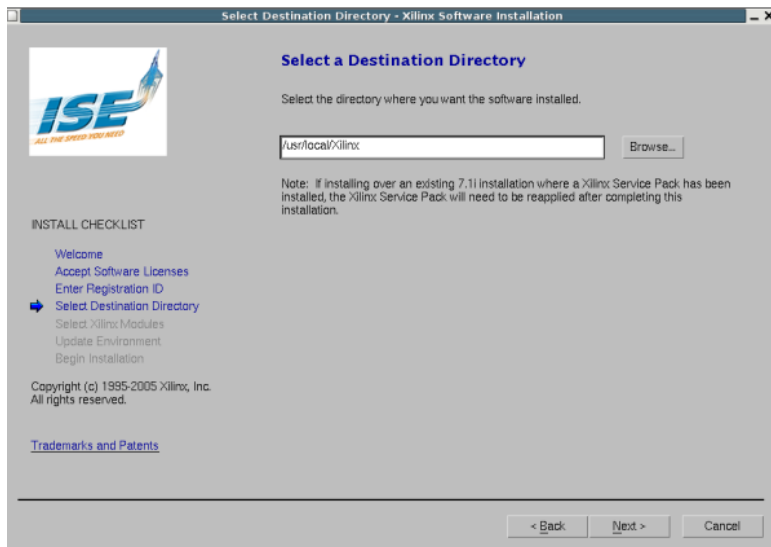


Introducir el número de registro, facilitado al adquirir el ISE.



Seleccionar el directorio dónde instalar el ISE. Yo lo pondré en /usr/local/Xilinx

Comprobación final. Aparece una ventana con todo lo que hemos seleccionado. Pulsamos en Install.



Comienza la instalación del CD 1. Se empiezan a instalar todos los ficheros en el disco duro. La duración depende del ordenador. En mi caso ha tardado unos 4 minutos.

Al llegar la barra de progreso al 100% comienza a instalar los drivers de los cables para descargar los diseños en la FPGA.

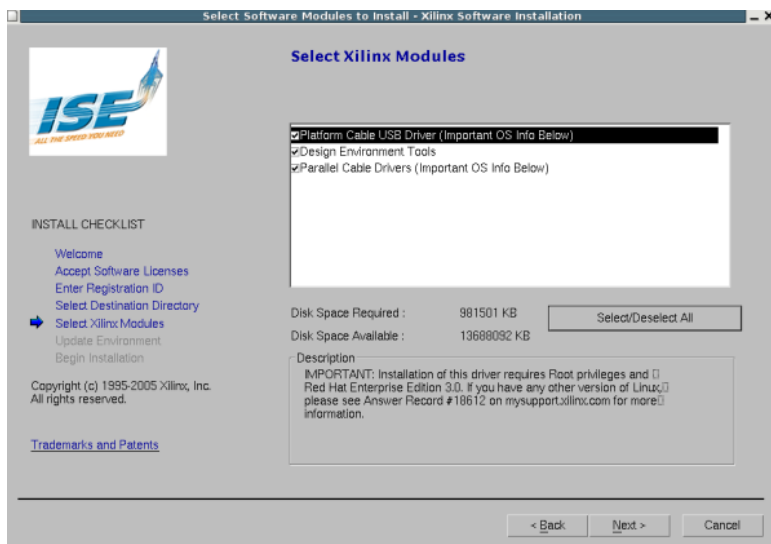
Fin de la instalación del CD 1. Pregunta si queremos descargar las actualizaciones disponibles.

Por último, UBUNTU no instala ninguna aplicación asociada, ni las variables de entorno, se deberá de concluir como root usando la siguiente secuencia:

```
cp /opt/Xilinx/settings.sh /usr/local/bin/startise
echo "export DISPLAY=:0" >> /usr/local/bin/startise
echo "exec ise" >> /usr/local/bin/startise
chmod +x /usr/local/bin/startise
```

Por último se inicia la aplicación (\$startise). 

Seleccionar qué módulos se instalarán. Yo he puesto todos.



Seleccionar las variables de entorno que se quieren instalar. Usualmente se instalan todas.

Sandino
Networks



TEQUILA, MEZCAL, CHARANDA, VODKA O UN BUEN BOURBON.
Servicios de hosting a la carta.



Desde hosting compartido hasta servidores dedicados. En Sandino Networks nos adaptamos a las necesidades de nuestros clientes.

Contamos con un gran soporte técnico y servicios de consultoría especializada que la experiencia del staff nos brinda

Sandino Networks es una compañía formada por personas que viven en Internet y han crecido con él.

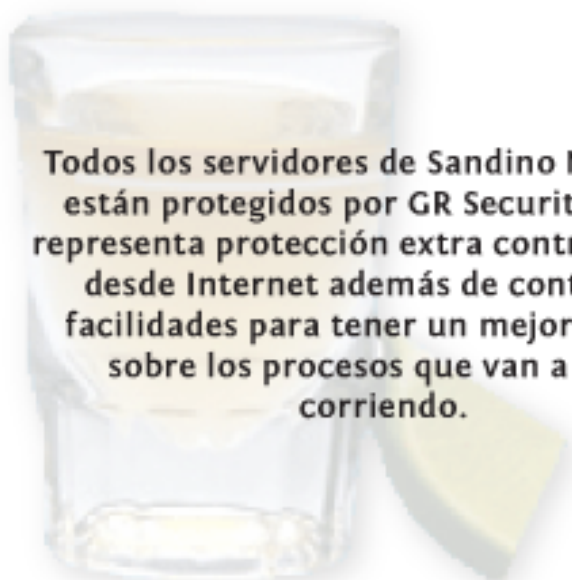
Emergency Recovery.
El mejor staff de soporte técnico a su disposición

Sistemas inutilizables, pérdida de datos, regreso a producción urgente. Nosotros lo ayudaremos a que regrese todo a la normalidad con nuestros servicios de consultoría emergente. Aunque ¿para qué sufrir, si puede prevenir todo esto?

Todos los servidores de Sandino Networks están protegidos por GR Security lo que representa protección extra contra ataques desde Internet además de contar con facilidades para tener un mejor control sobre los procesos que van a estar corriendo.

La seguridad es nuestra mas grande preocupación.

informes@sandino.net
www.sandino.net



La Morfología Matemática tiene diversas aplicaciones en el procesamiento de imágenes, representa un claro ejemplo del procesamiento no lineal. En el siguiente trabajo se mostrará el programa CIDEFORM, resultado de las técnicas aplicadas en la morfología matemática, el programa ofrece al usuario un ambiente simple para experimentar y explotar los diferentes aspectos usando operadores clásicos de esta área, el programa se desarrollo usando software libre como una herramienta open source realizada en Gambas, este software podría ser utilizado por investigadores en General.

La descripción básica de la Morfología Matemática descansa en la 'teoría de conjuntos' cuyos primeros trabajos se deben a Minkowski [Minkowsky, 1897], [Minkowsky, 1901] y Hadwiger [Hadwiger, 1957], [Hadwiger, 1959]. La continuación de estos trabajos de investigación, bajo la impulsión y reformulación de Matheron y Serra, se darían posteriormente a conocer bajo la denominación de Morfología Matemática, como una técnica no lineal de tratamiento de señales.

La mayor parte de esta teoría ha sido desarrollada en el Centre de Morphologie Mathématique (CMM) de l'Ecole des Mines de Paris.

Actualmente, el ámbito y alcance de los procesamientos morfológicos es tan amplio como el propio procesamiento de imágenes. Se pueden encontrar aplicaciones tales como la segmentación, restauración, detección de bordes, aumento de contraste, análisis de texturas, compresión, etc. En este capítulo se describen nociones básicas de la Morfología Matemática que serán útiles de repasar, ya que se hará referencia a ellas en el procesamiento cromático. Las obras [Matheron, 1967], [Matheron, 1975], [Serra, 1981] y [Serra, 1988], contienen en detalle la definición de los conceptos básicos en morfología a tratar a continuación.

Nociones sobre teoría de conjuntos

Se notarán los conjuntos con mayúsculas (X,Y,Z,...) y los elementos que contienen con minúsculas (p, q, r,...).

Definición 3.1: Dos conjuntos son iguales si están formados por los mismos elementos:

$$X = Y \iff (\forall p \mid p \in X \implies p \in Y \wedge \forall p \mid p \in Y \implies p \in X)$$

La igualdad de conjuntos es reflexiva, simétrica y transitiva. El conjunto vacío será denotado por \emptyset .

Definición 3.2: X es subconjunto de Y si todos los elementos de X pertenecen a Y:

$$X \subseteq Y \iff (\forall p \mid p \in X \implies p \in Y)$$

La inclusión es reflexiva, antisimétrica y transitiva.

Definición 3.3: La intersección de dos conjuntos X e Y es el conjunto de los elementos que pertenecen a ambos conjuntos:

$$X \cap Y = (\{ p \mid p \in X \wedge p \in Y \})$$

La intersección es conmutativa, asociativa e idempotente. Esta última propiedad es importante en morfología y significa que $X \cap X = X$.

Definición 3.4: La unión de dos conjuntos se constituye por los elementos que pertenecen a uno o al otro:

$$X \cup Y = (\{ p \mid p \in X \vee p \in Y \})$$

Al igual que la intersección, la unión de conjuntos es conmutativa, asociativa e idempotente.

Definición 3.5: La diferencia entre conjuntos X e Y, la componen los elementos que pertenecen a X pero que no están incluidos en Y:

$$X \setminus Y = (\{ p \mid p \in X \wedge p \notin Y \})$$

Esta operación no es en general conmutativa. Una variante de diferencia se conoce como diferencia simétrica de conjuntos, definida por los elementos que pertenecen a uno u a otro conjunto, pero nunca a ambos. La diferencia simétrica es conmutativa y asociativa.

Definición 3.6: La complementación de un subconjunto X, perteneciente a un conjunto Y (conjunto de referencia) se define como:

$$X^c = (\{ p \mid p \in Y \wedge p \notin X \})$$

Definición 3.7: Sean X e Y dos conjuntos pertenecientes al conjunto Z. Para todo elemento $x \in X$ e $y \in Y$, es posible hacer corresponder una suma algebraica $x+y$. De esta manera se forma un nuevo conjunto denominado adicción de Minkowski y denotado por $X \oplus Y$:

$$X \oplus Y = \{ x + y \mid x \in X, y \in Y \}$$

Definición 3.8: Dado un conjunto no vacío X, una relación binaria ' \leq ' en X es un orden parcial si cumple las siguientes propiedades:

- $x \leq x$ (reflexiva).
- $x \leq y, y \leq x$, implica que $x = y$ (anti simétrica).
- $x \leq y, y \leq z$, implica que $x \leq z$ (transitiva).

Para cualquier $x, y, z \in X$. Un conjunto con una relación de este tipo

será un conjunto que presenta un orden parcial y se denotará como (X, \leq) . El conjunto será totalmente ordenado si todos los elementos que lo componen son comparables, es decir: $x \leq y$ ó $y \leq x$, para cualquier par $(x,y) \in X$ [Avellanás, 1990].

Definición 3.9: Sea (X, \leq) un conjunto ordenado y $S \subseteq X$, un conjunto no vacío de X :

- Un elemento $x \in S$, es el menor elemento de S (mínimo) si $x \leq y$, para todo $y \in S$.
- Un elemento $y \in S$, es el mayor elemento de S (máximo) si $x \leq y$, para todo $x \in S$.
- Un elemento $x \in X$, es cota inferior de S si $x \leq y$, para todo $y \in S$.
- Un elemento $y \in Y$, es cota superior de S si $x \leq y$, para todo $x \in S$.
- Un elemento $x \in X$, es extremo inferior o ínfimo de S si y sólo si es cota inferior de S y para toda cota inferior i de S se verifica que $i \leq x$. (Es la mayor de las cotas inferiores). Si este elemento existe es único y se denota por $\inf S$.
- Un elemento $y \in X$, es extremo superior o supremo de S si y sólo si es cota superior de S y para toda cota superior s de S se verifica que $y \leq s$. (Es la menor de las cotas superiores). Si este elemento existe es único y se denota por $\sup S$.

Definición 3.10: Un conjunto ordenado (X, \leq) es un retículo completo si todos los subconjuntos de X poseen un ínfimo y un supremo.

Definición 3.11: Sean X e Y dos retículos completos. La relación f es una anamorfosis si y sólo si f es una biyección que conserva el ínfimo y supremo:

$$f(\inf \{x_i \mid i \in I\}) = \inf \{f(x_i) \mid i \in I\}$$

$$f(\sup \{x_i \mid i \in I\}) = \sup \{f(x_i) \mid i \in I\}$$

Para cualquier familia $\{x_i \mid i \in I\}$ en X , donde I es un conjunto de índices.

Propiedades de base de las transformaciones morfológicas

Toda operación morfológica es el resultado de una o más operaciones de conjuntos (unión, intersección, complementación...)[Díaz2003] haciendo intervenir dos conjuntos X, Y , ambos subconjuntos de un conjunto espacio Z . De los dos subconjuntos, Y recibe el nombre de elemento estructurante que, para operar con X , se desplazará a través del espacio Z .

Las operaciones morfológicas, transformaciones de conjuntos (X) , satisfacen, entre otras, las siguientes propiedades:

Invariabilidad a traslación

$$(X \oplus p) = (X) \oplus p$$

Donde p es el factor de traslación del conjunto.

Compatibilidad con las homotecias

Supongamos que X es una homotecia de un conjunto de puntos X , por tanto, las coordenadas de cada punto del conjunto se multiplican por alguna constante positiva k . Esto es equivalente a cambiar de escala con respecto a algún origen. Si k no depende de la escala, es invariante al cambio de escala:

$$(kX) = k(X)$$

La transformación morfológica \oplus posee el principio de conocimiento local si para cualquier conjunto de puntos M , subconjunto del dominio N , la transformación del conjunto X restringido al dominio de M , y después restringido al dominio N , es equivalente a aplicar la transformación $\oplus(X)$ y restringir el resultado en M :

$$(\oplus(X|_M)|_N) = (\oplus(X)|_M)$$

Continuidad

En forma resumida, este principio afirma que la transformación morfológica \oplus no exhibe ningún cambio abrupto. La noción de continuidad depende de la noción de vecindad, es decir, de la topología.

Dependiendo del tipo de operación morfológica se cumplirán otra serie de propiedades, como la idempotencia, extensividad, antiextensividad, etc.

Desarrollo

Transformaciones morfológicas elementales

El objetivo de las transformaciones morfológicas es la extracción de estructuras geométricas en los conjuntos sobre los que se opera, mediante la utilización de otro conjunto de forma conocida denominado elemento estructurante. El tamaño y la forma de este elemento se escoge, a priori, de acuerdo la morfología del conjunto sobre el que va a interactuar y de acuerdo a la extracción de formas que se desean obtener. Ejemplos básicos de elementos estructurantes utilizados en la práctica se muestran en la figura 1.



Figura 1. Ejemplo de formas básicas de elementos estructurantes planos.

Erosión

En 'teoría de retículos', una erosión es una operación que conmuta con el ínfimo. Dado un retículo completo X , una erosión es una función $\ominus : X \rightarrow X$ en la que:

$$\inf_{i \in I} (\ominus(x_i)) = \ominus(\inf_{i \in I} (x_i))$$

Donde I es cualquier conjunto de índices y $\{x_i\}$ es una colección arbitraria de valores $x_i \in X$.

La transformación de erosión es el resultado de comprobar si el elemento estructurante Y está totalmente incluido dentro del conjunto X . Cuando esto no ocurre, el resultado de la erosión es el conjunto vacío.

En la figura 2 se muestra la erosión de una señal unidimensional de función f . Se puede observar el efecto de la intersección de traslaciones definidas por el elemento estructurante Y . El resultado es el mínimo valor de todos los presentes bajo la definición del elemento estructurante.

Dilatación

La dilatación es la operación dual de la erosión. En 'teoría de retículos', un operador $\oplus : X \rightarrow X$ se denomina dilatación en el caso que conmute con el supremo de una colección de valores:

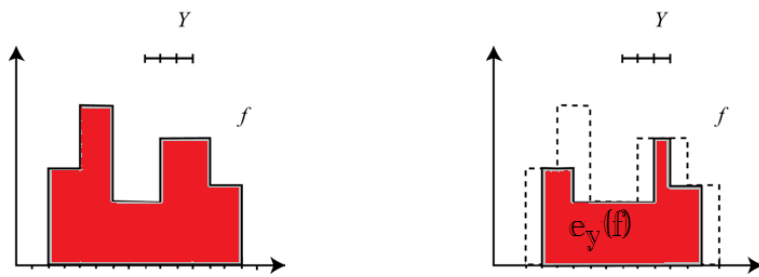


Figura 2. Erosión de una señal unidimensional definida por la función f . Elemento estructurante de tamaño 3. El resultado es el mínimo de todos los valores presentes bajo la definición del elemento estructurante.

$$Y(x_i) = \min_{i \in I} (x_i)$$

Donde I es cualquier conjunto de índices y $\{x_i\}$ es una colección arbitraria de valores $x_i \in X$.

El resultado de la dilatación es el conjunto de puntos origen del elemento estructurante Y tales que el elemento estructurante contiene algún elemento del conjunto X , cuando el elemento se desplaza por el espacio que contiene a ambos conjuntos:

$$Y(X) = \{x \mid Yx \cap X \neq \emptyset\}$$

Esta última ecuación puede describirse como una unión de conjuntos trasladados. Las traslaciones vienen definidas por el dominio del elemento estructurante:

$$Y(X) = \bigcup_{s \in Y} X - s$$

El efecto de una operación de dilatación puede observarse en la figura 3, en donde un elemento estructurante Y de forma de disco circular aumenta la definición del objeto X .

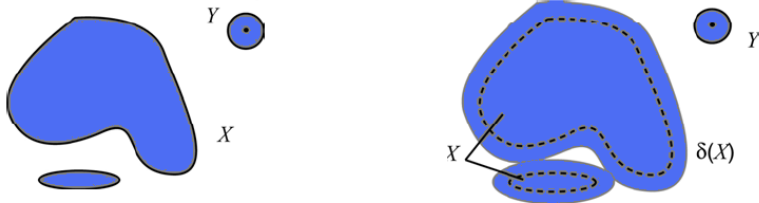


Figura 3. Dilatación de X por el elemento estructurante Y . El conjunto X aumenta su definición.

Se puede extender la última definición de dilatación a imágenes binarias o de escala de grises f , interpretando la dilatación como el máximo valor de las traslaciones de f (definidas por la forma del elemento estructurante) en cada punto de la imagen:

$$Y(f) = \max_{s \in Y} f - s$$

En otras palabras, el valor de dilatación de un píxel (x,y) es el máximo valor de la imagen en la ventana de vecindad definida por el elemento estructurante cuando su origen se sitúa en (x,y) :

$$Y(f)(x, y) = \max_{(s, t) \in Y} f(x - s, y - t)$$

Dualidad

La erosión y la dilatación son operaciones duales con respecto a la complementación. Esto significa que una erosión es equivalente a la complementación de la dilatación de la imagen complementada con el mismo elemento estructurante y viceversa, es decir:

$$Y = C \circ Y \circ C$$

Crecientes

Las operaciones de erosión y dilatación son crecientes, respetan el orden presente en la estructura de retículo. Para dos imágenes f y g :

$$\text{Si } f \leq g \text{ entonces } Y(f) \leq Y(g)$$

$$\text{Si } f \leq g \text{ entonces } Y(f) \leq Y(g)$$

Gracias a esta propiedad, las operaciones morfológicas pueden extenderse del caso binario a la escala de los grises, puesto que estas últimas señales pueden expresarse como una suma de imágenes binarias (una superposición de conjuntos apilados). Cada una de las imágenes binarias corresponde al conjunto de píxeles cuyo nivel de gris es superior a un umbral determinado. La señal de grises se define como:

$$f = \sum_{t=1}^{t \max} CSt(f)$$

Donde $CSt(f)$ es la imagen binaria que contiene los píxeles con valor en f mayor o igual a t .

El que la erosión y la dilatación sean crecientes implica que las relaciones de orden entre los diferentes conjuntos apilados de una imagen sean respetadas por estos operadores. La erosión y la dilatación son, pues, invariantes a la descomposición por umbrales:

$$Y \left(\sum_{t=1}^{t \max} CSt(f) \right) = \sum_{t=1}^{t \max} Y(CSt(f))$$

$$Y \left(\sum_{t=1}^{t \max} CSt(f) \right) = \sum_{t=1}^{t \max} Y(CSt(f))$$

Apertura y cierre

Generalmente, en un retículo completo X , la dilatación $Y(X)$ y la erosión $Y(X)$ son operaciones que no admiten inversa, por lo tanto, no hay manera de determinar el origen X desde las imágenes $Y(X)$ o $Y(X)$. Sin embargo, es posible, mediante una adjunción de operadores básicos aproximarse a la forma original en base a la dualidad que poseen. Se puede intentar recuperar el elemento original X erosionado si, una vez efectuada la erosión, se realiza una dilatación, adjunción de operaciones conocida como apertura. Lo mismo ocurre si se efectúa una dilatación seguida de una erosión, proceso conocido como cierre.

Apertura morfológica

La apertura de una señal f por un elemento estructurante Y se denota por $Y(f)$ y se define como la erosión de f por Y , seguida de la dilatación por el mismo elemento estructurante:

$$Y(f) = Y(Y(f))$$

La apertura de una imagen es independiente del origen del elemento estructurante, puesto que si la erosión se corresponde con una intersección de traslaciones, la dilatación que sigue es una unión de traslaciones en dirección opuesta.

En el caso general de conjuntos se puede definir la apertura como la unión de los elementos estructurantes Y que se encuentran totalmente dentro del conjunto X:

$$Y(X) = \cup \{Y \mid Y \subset X\}$$

El efecto de una operación de apertura puede observarse en la figura 4, en la que un elemento estructurante Y, en forma de disco circular, provoca en la erosión la desaparición de una estructura que, en la operación de dilatación, no se puede recuperar.

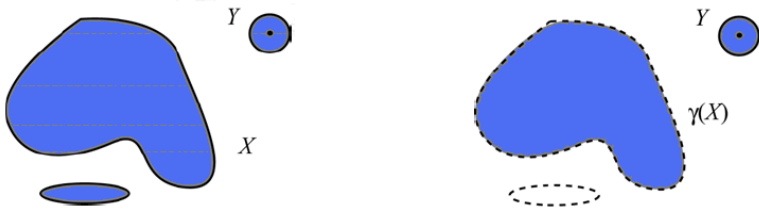


Figura 4. Apertura morfológica del conjunto X por el elemento estructurante Y. Eliminación de objetos menores en tamaño al elemento estructurante. La apertura redondea las convexidades importantes.

Cierre morfológico

El cierre de una señal f por un elemento estructurante Y se denota por $\phi(f)$ y se define como la dilatación de f por Y, seguida de la erosión por el mismo elemento estructurante:

$$\phi(f) = Y(Y(f))$$

Por la propiedad de complementación de las operaciones básicas, se puede expresar el cierre como la intersección de todas las traslaciones del complemento del elemento estructurante, tal que éste contiene a X:

$$\phi(X) = \bigcap \{Y_c \mid X \subset Y_c\}$$

Al igual que la apertura, el cierre de una imagen es independiente del origen del elemento estructurante. El cierre de un conjunto X por un elemento estructurante con forma de disco se ilustra en la figura 5. El cierre es el espacio descrito por el elemento estructurante cuando es forzado a estar fuera de los conjuntos.

En el cierre de la señal unidimensional presente en la figura 5.a se observa como la dilatación une estructuras que la erosión no puede separar. En una imagen de niveles de grises el cierre morfológico ayuda a eliminar estructuras oscuras menores en tamaño al elemento estructurante. La dilatación maximiza los valores de forma que se atenúan los objetos oscuros (figura 5.b). La erosión minimiza la señal y sólo los elementos no eliminados quedan presentes en la imagen final.

Gambas

Gambas [Gamproect] es un lenguaje de programación libre derivado de BASIC. Es similar al producto de Microsoft Visual Basic y se

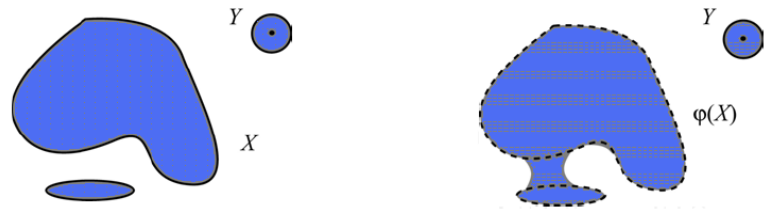


Figura 5. Apertura morfológica del conjunto X por el elemento estructurante Y. El cierre redondea las concavidades importantes.

distribuye con licencia GNU GPL. Aunque desde otro punto de vista, está muy inspirado también por Java.

Permite crear formularios, botones de comandos, cuadros de texto o enlazar bases de datos como MySQL, PostgreSQL o SQLite.

Gambas nació como respuesta a la necesidad de tener un entorno de desarrollo rápido de aplicaciones o RAD, además cumple la necesidad de muchos programadores del lenguaje de Microsoft de tener un lenguaje de programación sencillo para plataformas libres (como GNU/Linux y *BSD).

A pesar de estar basado en el lenguaje BASIC y poseer gran similitud con Visual Basic, Gambas no es un clon de este último.

Es importante señalar que prácticamente todos los proyectos desarrollados en este entorno son compatibles con versiones nuevas de Gambas, no así los hechos por M\$ Visual Basic, es decir, por ejemplo un proyecto hecho en ms Visual Basic 5.0 difícilmente correrá en la última versión, ya que no es orientado a Objetos, por lo tanto carece de herencia.

La principal diferencia con respecto a Visual Basic 6.0 es el soporte para orientación a objetos. En Gambas, no existen los arreglos de controles pero dispone de un elemento similar llamado group, que utilizado conjuntamente con arreglos de objetos ofrecen resultados análogos.

Gambas añade algunas propiedades nuevas en algunos de sus controles, como la alineación vertical/horizontal en etiquetas, o algunas propiedades gráficas en el elemento "Button" (Equivalente a CommandButton de Visual Basic).

El acceso a bases de datos es sumamente sencillo y potente. Gambas incluye, al igual que Visual Basic, su propio empaquetador para la distribución de aplicaciones, que permite crear incluso paquetes deb.

Gambas es una excelente y potente alternativa a la programación visual ofrecida en Visual Basic para entornos Linux. Su talón de Aquiles actualmente es su escasa documentación.

En la figura 6. Se muestra el Logotipo de este programa.



Figura 6. Logotipo Oficial del proyecto GAMBAS

En la Figura 7 se muestra el sistema completo de manipulación de imágenes, al cual se le llamo CIDEFORM, este nombre se le dio con la combinación de CIDETEC y MORFOLOGIA, este sistema será publicado para su libre distribución bajo la licencia GLP en los sitios SOURCE FORGE y FRESHMEAT.

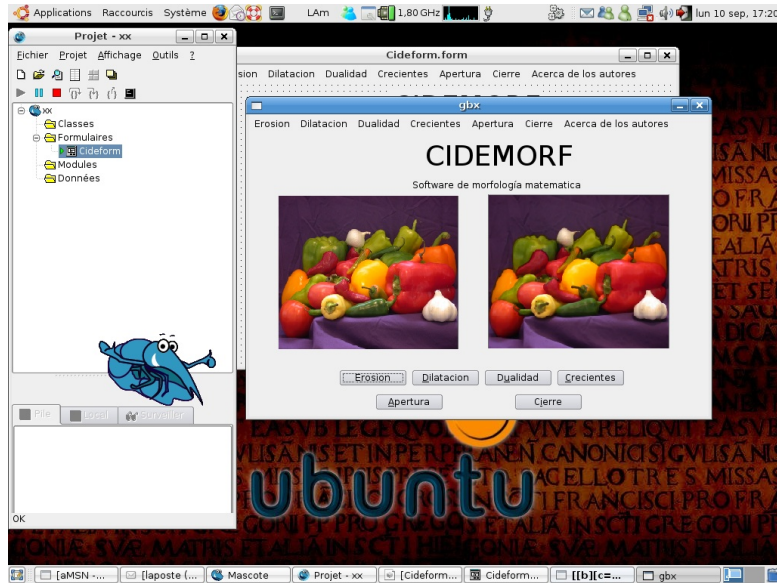


Figura 7. Se muestra la interfaz del programa CIDEFORM.

Conclusiones

El reconocimiento de patrones se puede definir desde un punto de vista cualitativo como “La asignación de un objeto o evento físico a una de las categorías posibles especificadas previamente” o desde un punto de vista cuantitativo como “Un problema de estimación de la función densidad en un espacio de alta dimensión y división del espacio en las regiones de las categorías o clases”

En la figura 8 se muestra el esquema general de análisis de Imágenes.



Figura 8- Esquema General de Análisis de Imágenes.

Como es posible analizar en la figura 8, dentro de las técnicas de nivel bajo encontraremos la sección de mejoramiento y restauración de la imagen, la cual inicia con la adquisición de las imágenes (esta adquisición podrá ser el resultado de una fotografía, el archivo resultante del uso de un Escáner o bien el resultado directo de una cámara digital fotográfica o cámara de video con capacidades de almacenamiento digital o binario.) y que termina con el mejoramiento de estas, la restauración o su conversión. (En este punto la creación de un software usando diferentes operaciones morfológicas, permitirá realizar cualquiera de los tres casos).

El software CIDEFORM presento un buen desempeño de optimización de imágenes utilizando la morfología matemática. El desarrollo de sistemas basados en herramientas de software libre permite la explotación de estas herramientas para el acceso libre para investigadores y científicos en los países de habla hispana, la siguiente versión tendrá las mejoras en los algoritmos de pre-procesamiento y soportará múltiples lenguajes.

Crea tú sitio web en 10 minutos usando Django

Andres Bernardo Vargas Rodríguez
andres.vargas@revista-sl.org

Como recordarán en la edición pasada, Django es un framework para desarrollo web escrito en Python con el cual se pueden tener aplicaciones web sumamente funcionales en muy poco tiempo. Ahora veamos como crear un sitio web de forma rápida y sencilla, aprovechando todo el poder de este framework.

Si ya sabemos algo sobre desarrollo con Django en tan solo 10 minutos podremos tener nuestro sitio web, si no es así lee el artículo presentado en el número pasado y veras como no te va a costar mucho tiempo construir tu sitio web.

El sitio consta de hacer paginas estaticas con contenido dinamico algo asi como que tengas varias páginas html con links a ellas cada uno.

Primero debemos empezar a crear el proyecto con django-admin.

```
$ django-admin.py startproject sitio10min
$ cd sitio10min/
```

Ahora hacemos las configuraciones necesarias al archivo settings.py

```
DATABASE_ENGINE = 'sqlite3'
DATABASE_NAME = 'base.db'

INSTALLED_APPS = (
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.sites',
    'django.contrib.admin', # agregamos el \
        administrador de django
)
```

Con esto ya tenemos lo básico. Ahora creamos las tablas que nos van a servir para el sitio.

```
$ python manage.py syncdb
Creating table auth_message
Creating table auth_group
Creating table auth_user
Creating table auth_permission
Creating table django_content_type
Creating table django_session
Creating table django_site
Creating table django_admin_log
You just installed Django's auth system, which means
you don't have any superusers defined.
Would you like to create one now? (yes/no): yes
Username (Leave blank to use 'zodman'): admin
E-mail address: admin@admin.com
Password:
Password (again):
Superuser created successfully.
Installing index for auth.Message model
Installing index for auth.Permission model
Installing index for admin.LogEntry model
```

Con esto listo ya funciona el Django

```
$ python manage.py runserver
Validating models...
0 errors found
```

```
Django version 0.97-pre-SVN-unknown, using settings
'sitio10min.settings'
Development server is running at
http://127.0.0.1:8000/
Quit the server with CONTROL-C.
```

Pero no tenemos nada aun, ahora configuremos el acceso al administrador de Django, en el archivo urls.py

```
from django.conf.urls.defaults import *

urlpatterns = patterns('',
    # Example:
    # (r'^sitio10min/', \
        include('sitio10min.foo.urls')),

    # Descomentemos la linea de abajo de esta forma
    (r'^admin/', \
        include('django.contrib.admin.urls')),
)
```

Lo que acabamos de hacer es agregar la URL para que podamos entrar al administrador de Django, ahora si entramos al navegador veremos: Page not found.

Si entramos en el navegador: <http://localhost:8000/admin/>

El login del administrador que por cierto esta bien bonito. Por ahora no tenemos mas que el administrador de usuarios y de sitios. Pues bien iniciemos ahora con el sitio.

Para empezar usaremos 2 addons del Django: flatpages [1] y los templatetag [2].

Instalemos los flatpages, en el archivo settings.py agregamos a las aplicaciones que usa para nuestro proyecto las flatpages.

```
INSTALLED_APPS = (
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.sites',
    'django.contrib.admin',
```

```
'django.contrib.flatpages' # aplicacion flatpages
```

Y al middleware también:

```
MIDDLEWARE_CLASSES = (  
    'django.middleware.common.\br/>        CommonMiddleware',  
    'django.contrib.sessions.middleware.\br/>        SessionMiddleware',  
    'django.contrib.auth.middleware.\br/>        AuthenticationMiddleware',  
    'django.middleware.doc.XViewMiddleware',  
    'django.contrib.flatpages.middleware.\br/>        FlatpageFallbackMiddleware'  
    #middleware de las flatpages  
)
```

Con esto ya tenemos instalado las flatpages ahora crearemos sus tablas en la base de datos.

```
$ python manage.py syncdb  
Creating table django_flatpage  
Installing index for flatpages.FlatPage model
```

Para que veamos si funcionó entramos a nuestro navegador en <http://localhost:8000/admin>

Y vemos que el administrador ya agrego las flatpages. Las flatpages son páginas ya programadas para Django en ellas puedes poner contenido estático como el FAQ o el about de tu sitio. Estas páginas estan compuestas de título, nombre y contenido.

Ya vamos por la mitad del sitio. Ahora ya que tenemos nuestras páginas estáticas. Hagamos el template para mostrarla creamos un directorio en el directorio donde estamos trabajando.

```
$ mkdir templates
```

Y dentro de ella un directorio para las flatpages (que por default tiene que ser el directorio flatpages).

```
$ mkdir templates/flatpages
```

Dentro de este directorio un archivo donde toda la información es concentrada (renderizada por Django) por default tenemos que llamarla default.html

```
<html>  
<body>  
  
<h1>{{ flatpage.title }}</h1>  
{{ flatpage.content }}  
  
</body>  
</html>
```

Tú directorio quedaria asi:

```
templates/  
  |-- flatpages  
    |-- default.html
```

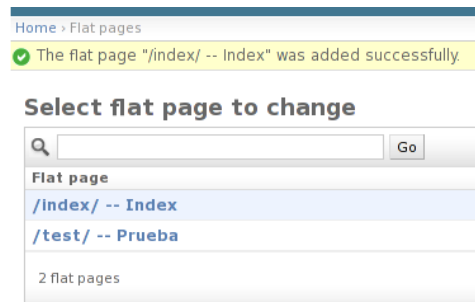
Ya que agregamos esto debemos agregar al settings.py el directorio con nuestro template, ayudate con el comando: pwd para poner el directorio.

```
TEMPLATE_DIRS = (  

```

```
# Put strings here, like \  
    "/home/html/django_templates" or \  
    "C:/www/django/templates".  
# Always use forward slashes, even on Windows.  
# Don't forget to use absolute paths, not \  
    relative paths.  
    '/home/zodman/sitio10min/templates'  
)
```

Ahora bien entremos al administrador de Django (<http://localhost:8000/admin/>) y agregamos las flatpages.



Flatpages en Django

Ya que tengamos nuestras paginas entremos a verlas

```
http://localhost:8000/index/  
http://localhost:8000/test/
```

Ahore pongamos un menú para ellas. Con esto utilizaremos las templatetags, podriamos también en el default.html agregar las href pero para hacerlo más automatizado.

Así que como dice la documentación de Django[2], creen una aplicación, con cualquier nombre, es lo de menos importancia.

```
$ python manage.py startapp util
```

Ya que creamos la aplicación instalemosla al Django en el archivo settings.py, hay que hacer este paso para que Django importe la aplicación se fije del directorio templatetags para que pueda importar también su contenido.

```
INSTALLED_APPS = (  
    'django.contrib.auth',  
    'django.contrib.contenttypes',  
    'django.contrib.sessions',  
    'django.contrib.sites',  
    'django.contrib.admin',  
    'django.contrib.flatpages',  
    'sitio10min.util'  
)
```

En este directorio útil creamos otro directorio que se llame templatetags (tiene que ser llamado así para que Django pueda importarlo).

```
$ mkdir util/templatetags
```

Y dentro este directorio creamos un archivo `__init__.py`

```
$ touch util/templatetags/__init__.py
```

Ahora dentro tempatetags creamos un archivo generamenu.py y ponemos el siguiente código:

```
from django.template import Library  
from django.contrib.flatpages.models import FlatPage
```

```
register=Library()
```

```
# generamos un menú para las paginas estáticas
```

```
def flatpagemenu():  
    pages=FlatPage.objects.all()  
    menu=""  
    for i in pages:  
        menu+=" <a href='"+i.url+"\  
            ' title='"+i.title+"'>" + i.title + "</a>"  
    return menu
```

```
#registramos el tag para el template  
register.simple_tag(flatpagemenu)
```

Este archivo hace que importe las flatpage existentes y las ponga en el menú.

Ahora modifiquemos el archivo default.html en nuestro directorio templates/flatpages.

```
<html>  
<body>  
{% load generamenu %}  
{% flatpagemenu %}  
  
<h1>{{ flatpage.title}}</h1>  
{% flatpage.content %}  
  
</body>  
</html>
```

Actualizamos nuestro navegador y ya tenemos lista la página con su menú dinámico.

Pero hagamos las aun mejor, configuremos más las URL. Por que si entramos a <http://localhost:8000/> nos aparece que la página no fue encontrada.

Así que utilizemos algo para redireccionar las direcciones usando de las vistas genéricas el `redirect_to` [3]

En nuestro archivo `urls.py` agreguemos esto:

```
from django.conf.urls.defaults import *  
  
urlpatterns = patterns('',  
    # Example:  
    # (r'^sitio10min/', \  
    include('sitio10min.foo.urls')),  
  
    # Uncomment this for admin:  
    (r'^admin/', \  
    include('django.contrib.admin.urls')),  
    (r'^$', 'django.views.generic.simple.redirect_to\  
    , {'url': '/index/'}) # agregar esto  
)
```

Así que cuando abramos la dirección raíz `^$` = <http://localhost:8000/> nos mandará a la página del index.

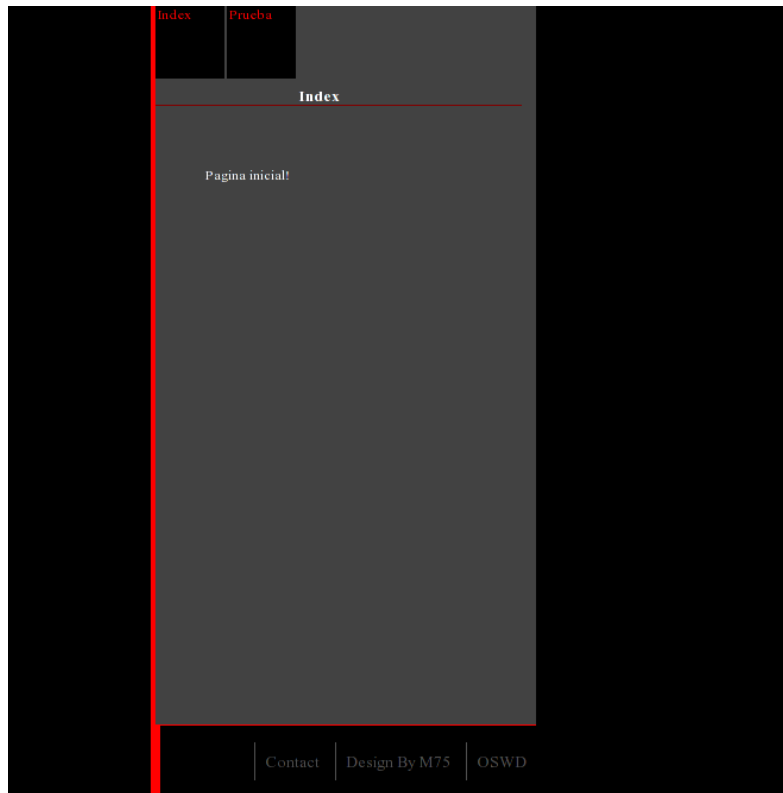
Ahora no te compliques la vida poniendolo guapo. entra a www.oswd.org y buscale un bonito template a tu sitio.

Para mas información te dejo las referencias que he consultado para este artículo:

[1]<http://www.djangoproject.com/documentation/flatpages/>

[2]http://www.djangoproject.com/documentation/templates_python/#extending-the-template-system

[3]http://www.djangoproject.com/documentation/generic_views/#django-views-generic-simple-redirect-to



Muestra del sitio web que puedes desarrollar con Django.

Introducción al desarrollo de interfaces gráficas

Eyden Barboza Varela
eyden.barboza@revista-sl.org

Gtk# (gtk sharp) es el binding de .Net para las bibliotecas de GTK + , si alguna vez has programado en él verás que hacerlo en Gtk# resulta muy fácil , aquí lo utilizaremos usando el lenguaje de programación C#.

Cualquier aplicación escrita en Gtk# podrá ser ejecutada en diversos sistemas operativos Linux, MacOS, Windows. Para tenerlo mas claro , observemos el siguiente ejemplo:

```
using System;
using Gtk;

public class Prueba
{
    static Gtk.Entry texto;

    //definimos el punto de entrada de nuestro programa

    public static void Main()
    {
        //se inicializa GTK
        Application.Init();

        //creamos una ventana
        Window ventana = new Window("prueba");

        //establecemos el borde de la ventana
        ventana.BorderWidth = 10;

        //establecemos el tamaño por defecto que tendrá
        //la ventana
        ventana.SetDefaultSize(450,150);

        //creamos un boton con la etiqueta Aceptar
        Button boton = new Button("Aceptar");

        //creamos un Label con la etiqueta prueba
        Label etiqueta = new Label("prueba");
        texto = new Gtk.Entry();

        //asociamos el evento DeleteEvent del objeto ventana
        //a un manejador de eventos

        ventana.DeleteEvent += new
        DeleteEventHandler(window_cerrar);

        //asociamos el evento Clicked del objeto boton a
        //un manejador de eventos pasando como parámetro
        //el nombre del método
        boton.Clicked += new EventHandler(button_click);

        //creamos nuestras cajas empaquetadoras
```

```
VBox vbox = new VBox();
HBox hbox = new HBox();

//hacemos el empaquetado
vbox.PackStart(hbox,false,false,5);
hbox.PackStart(etiqueta,false,false,5);
hbox.PackStart(texto,false,false,5);
hbox.PackStart(boton,false,false,5);

ventana.Add(vbox);
ventana.ShowAll();
//la aplicación entra en un bucle a la espera
//de eventos
Application.Run();
}

//definimos el método que se ejecutara cuando
//nuestra ventana emita un DeleteEvent
static void window_cerrar(object o,DeleteEventArgs
args)
{
    Application.Quit();
}

//definimos el método button_click el cual sera
//llamado cuando el botón sea pulsado
static void button_click(object sender , EventArgs
e)
{
    Console.WriteLine("{0}",texto.Text);
}
}
```

y compilamos de la siguiente forma

```
$mcs -pkg:gtk-sharp-2.0 prueba.cs
```

y ejecutamos mono prueba.exe

El resultado sería el mostrado en la figura 1.

Voy a comentar unos puntos que considero importantes; qué pasaría si

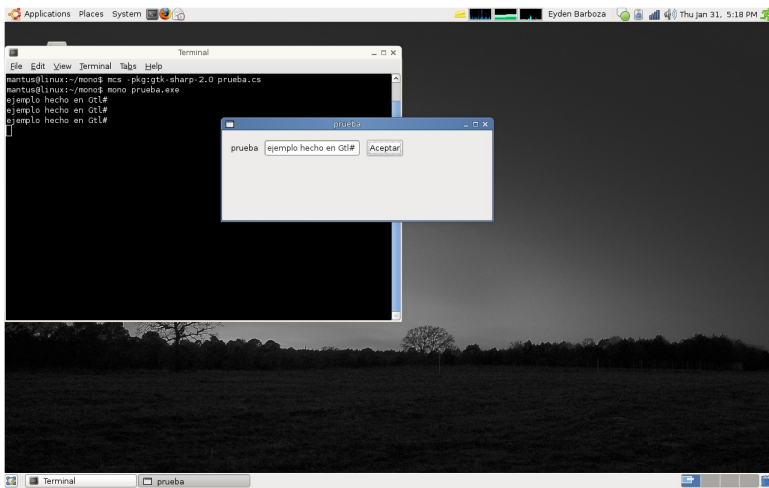


Figura 1: Resultado del primer ejemplo

en nuestra aplicación presionamos un botón y nosotros esperamos que pase algo y sin embargo esto no sucede, nuestro programa no responde, esto pasa por que no hemos establecido nuestros manejadores de eventos en GTK se usa el tipo de programación orientado a eventos esto quiere decir que haremos una asociación de manejadores de eventos a su respectivos métodos a ejecutar, dichos eventos son el resultado de la interacción de el usuario con algún widget de nuestra interfaz , como por ejemplo cuando un usuario hace click en un botón es lanzado el evento clicked, de esta forma se conectaría el evento.

```
boton.Clicked += new EventHandler (boton_pulsado);
```

Aquí lo que hacemos es asociar el evento clicked del objeto botón a un manejador de eventos, como vemos es pasado como parámetro al constructor el nombre del método que contendrá las sentencias que queremos que se ejecuten al ocurrir dicho evento.

Cajas Empaquetadoras

En Gtk#, sólo nos permite tener un Widget en la ventana, si quisieramos agregar mas nos mostraría un error, allí es donde tenemos que recurrir a las cajas empaquetadoras estos son controles invisibles que nos ayudan a organizar todos nuestros widgets son de dos tipos verticales y horizontales y los objetos se insertan de izquierda a derecha o de derecha a izquierda, según como le especifiquemos, para las cajas verticales usamos VBox y para las horizontales Hbox, para insertar los objetos dentro de los contenedores usamos PackStart() y PackEnd() el método PackStart() irá de izquierda a derecha en un hbox y de arriba hacia abajo en un vbox con PackEnd() el orden iría en sentido contrario un ejemplo de declarar un VBox en C# sería de la siguiente forma:

```
VBox vbox = new VBox();
```

Ahora vamos a ver cual sería la forma de ir empaquetando los objetos de nuestra interfaz

```
contenedor.PackStart(objeto, expand, fill, padding);
contenedor.PackEnd(objeto, expand, fill, padding);
```

Donde objeto es el widget que queremos agregar al contenedor, expand se encarga de controlar si el widget ocupara todo el espacio disponible de la caja y de esta forma se expande hasta ocupar el espacio que resta , Fill controla si el espacio extra del contenedor va hacer rellenado con nuestro objeto, y padding pues es el espacio que se va a dejar entre el objeto que estamos empaquetando y el de a lado.

```
contenedor.PackStart(boton, true, true, 5);
```

Ahora vamos hacer el mismo ejemplo pero diseñando nuestra interfaz con la ayuda de Glade.

Este es un diseñador de interfaces para GTK+ ,una de las ventajas que nos da el usar Glade, es que diseñamos nuestra interfaz de usuario de una forma totalmente gráfica lo cual nos agiliza el proceso de desarrollo de nuestra GUI al terminar, Glade nos generará un archivo .Glade este archivo está escrito en XML el cual contendrá todas las definiciones y propiedades de todos los widgets que formen parte de nuestro proyecto y por medio de la biblioteca libglade podremos usar nuestra GUI en nuestro Proyecto hecho en C#.

Ventajas que nos ofrece el hecho de usar Glade

- Limpieza en nuestro código .
- Agilizar el proceso de desarrollo de nuestra interfaz gráfica.

Para este ejemplo creamos un nuevo proyecto en glade en donde los widgets importantes a insertar son los siguientes:

```
GtkEntry texto;
GtkButton boton
```

```
using System;
using Gtk;
using Glade;
```

```
class Example
```

```
{
[Glade.Widget] Gtk.Entry texto;
```

```
public Example()
```

```
{
```

```
Application.Init();
Glade.XML gxml = new
Glade.XML(null, "gui.glade", "window1", null);
gxml.Autoconnect(this);
Application.Run();
}
```

```
}
```

```
public static void Main(String[] args)
```

```
{
```

```
new Example();
```

```
}
```

```
public void on_boton_clicked(object sender ,
EventArgs a)
```

```
{
```

```
Console.WriteLine("{0}", texto.Text);
```

```
}
```

```
public void on_window1_delete_event (object o,
DeleteEventArgs args)
```

```
{
```

```
Application.Quit ();
args.RetVal = true;
```

```
}
```

```
}
```

Ahora compilamos y ejecutamos nuestro ejemplo:

```
$ mcs -pkg:glade-sharp-2.0 -resource:gui.glade
example.cs
$ mono example.exe
```

Lo que se muestra en la figura 2 sería el resultado.

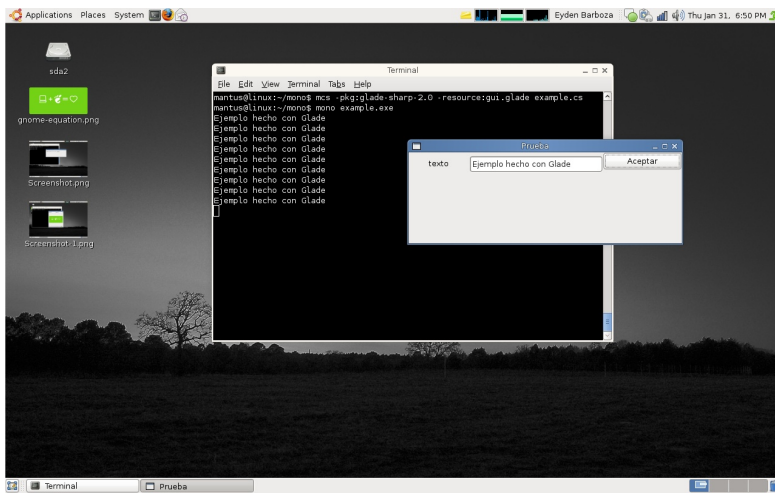


Figura 2: Resultado del ejemplo

De esta forma se reducirían nuestras líneas de código, ya que la definición de las propiedades de nuestra interfaz se encuentran en el archivo gui.glade. Ahora vamos hacer un ejercicio mas, el cual lo que hace es abrir una imagen y crear un thumbnail, para eso de nuevo creamos un nuevo proyecto en glade en donde los widgets importantes a insertar son los siguientes:

```
GtkSpinButton ancho
GtkSpinButton alto
GtkImage imagen
GtkLabel nombre
GtkButton button
```

De esta forma quedaría nuestro código

```
using System;
using Glade;
using Gdk;
using Gtk;

class thumbnail{

    [Glade.Widget] Gtk.Image imagen;
    [Glade.Widget] Label nombre;
    [Glade.Widget] Gtk.SpinButton ancho;

    [Glade.Widget] Gtk.SpinButton alto;

    static FileSelection file;

    public thumbnail ()
    {
        Application.Init();
        Glade.XML xml = new Glade.XML
        (null, "ventana.glade", "ventana", null);
        xml.Autoconnect (this);
        Application.Run();
    }

    public static void Main(String[] args)
```

```
{
    new thumbnail();
}

public void event_ok( object obj, EventArgs args)
{
    try {
        Gdk.Pixbuf pixbuf = new Pixbuf (file.FileName);

        imagen.Pixbuf =
        pixbuf.ScaleSimple(ancho.ValueAsInt, alto.ValueAsInt,
        Gdk.InterpType.Hyper);
        pixbuf.Dispose();

        nombre.Text = "Archivo: " + file.FileName;
        file.Destroy();

    } catch (Exception e) {
        Console.WriteLine (e);
    }
}
```

```
public void event_cancel(object obj ,EventArgs args)
```

```
{
    file.Destroy();
}

public void on_button1_clicked (object o, EventArgs
args)

{
    file = new FileSelection("Seleccionar archivo");
    file.OkButton.Clicked +=new EventHandler (event_ok);
    file.CancelButton.Clicked +=new EventHandler
(event_cancel);

    file.FileName = " ";

    file.Show ();
}

public void on_ventana_delete_event (object o,
DeleteEventArgs args)
{
    Application.Quit ();
    args.RetVal = true;
}
}
```

De nuevo compilamos y ejecutamos

```
$ mcs -pkg:glade-sharp-2.0 -resource:ventana.glade
image.cs
$ mono image.exe
```

Nuestro programa se debería ver como en la figura 3.

Como en el ejemplo pasado, mencionaré algunos puntos que considero importantes.

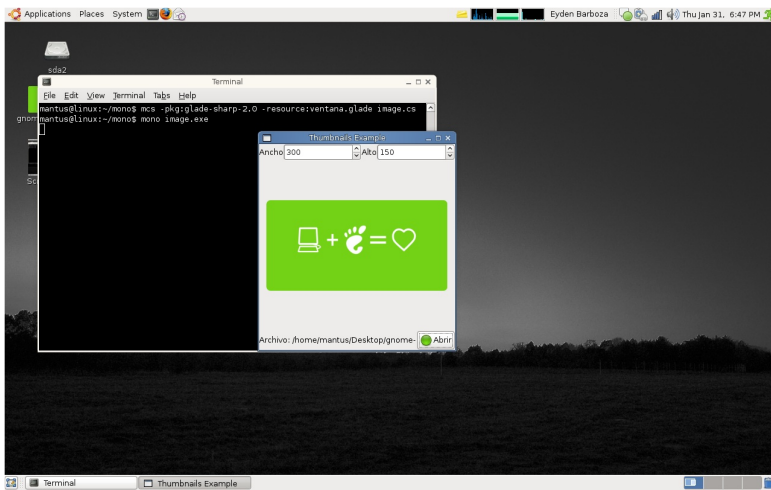


Figura 3: Resultado del ejemplo

PIXBUF

GdkPixbuf es una biblioteca que se encarga de lo que en si es la carga y manipulación de imágenes, acepta diversos formatos de archivo de imagen (png, tiff, jpg, gif, xpm, pcx, ico, xpm, xbm), ésta se encuentra dentro del espacio de nombres Gdk, esta es la forma de construir el objeto.

```
pixbuf = new Gdk.Pixbuf (filew);
```

Aquí escalamos la imagen original, para generar el Thumbnail y posteriormente la propiedad Dispose para liberar el Pixbuf que hemos creado

```
imagen.Pixbuf =
pixbuf.ScaleSimple(200,200,Gdk.InterpType.Hyper);
pixbuf.Dispose();
```

En el ejemplo para obtener el nombre del archivo utilizamos un FileSelection el cual es una caja de diálogo con dos botones uno de cancelar y el otro de aceptar, de nuevo lo construimos de la misma forma

```
file = new FileSelection("File selection");
```

Para obtener el nombre de la imagen , tuvimos que hacer uso de la propiedad Filename del mismo FileSelection esto con la necesidad de obtener el nombre de la imagen donde file es el nombre de nuestro FileSelection


```
file.Filename
```

En nuestro programita mostrábamos el nombre del archivo a través de un Label la sentencia quedaría de la siguiente forma

```
nombre.Text = "Archivo: " + file.Filename;
```

Como comenté antes nuestro FileSelection contiene un boton OK y el otro Cancel dichos botones tenemos que conectarlos con sus señales a sus respectivos métodos

```
file.OkButton.Clicked +=new EventHandler (event_ok);
file.CancelButton.Clicked +=new EventHandler
(event_cancel);
```

Bueno como podemos ver , es realmente rápido y sencillo , el desarrollo de aplicaciones gráficas utilizando C# con mono y Gtk# , también observamos que Glade nos reduce mucho el tiempo de desarrollo de ésta. 

La Columna MySQL: Recuperación de la contraseña de root y triggers

Jorge de Jesús Medrano Rodríguez
h1pple@h1pple.net

A partir de ahora escribire una columna dedicada a MySQL. Esta columna será a manera de mini tutoriales para solucionar diversas problemáticas como va desde hacer un trigger, un cursor, un procedimiento almacenado, una función, utilizar vistas, realizar indexaciones, optimizar bases de datos, entre otras muchas cosas que tengo alojadas entre mis dos orejas.

Recuperando la contraseña del root.

Primero es necesario tener root o algún usuario en el servidor con los privilegios suficientes para detener e iniciar el servidor.

Después hay que detener el proceso actual que se este ejecutando de MySQL.

Una vez detenido crear un archivo de texto el cual va a contener la siguiente instrucción:


```
SET PASSWORD FOR 'root'@'localhost' =  
PASSWORD('password_nuevo_de_root');
```

Guarde el archivo con cualquier nombre. para casos de ejemplo utilizaremos el nombre de: mysql-restaura-password

Por último reinicie el servidor MySQL con la opción `-init-file=/ruta/archivo/mysql-restaura-password`

```
mysqld_safe --init-file=/ruta/archivo/mysql-restaura-  
password &
```

Los contenidos del archivo son ejecutados al inicio del servidor, cambiando la contraseña de root.

 No olvide borrar el archivo `mysql-restaura-password` ya que no hacerlo puede ser un problema de seguridad.

Triggers (disparadores)

A partir de la versión 5.0.2 de MySQL se incorporo el soporte básico para los triggers. Los triggers son un objeto dentro de la base de datos el cual esta asociado a una tabla y este es ejecutado al ocurrir un evento en la tabla. Los eventos en los que es ejecutados los trigger son: INSERT, UPDATE, DELETE pero se debe especificar si queremos que sea antes "BEFORE" o después "AFTER" de alguna de estas tres acciones. Sin mas rollo vayamos directo al código, la sintaxis para crear un trigger es la siguiente:

```
CREATE TRIGGER nombre_trigger  
momento_ejecucion_evento ON nombre_tabla  
FOR EACH ROW BEGIN  
sentencias_trigger  
END
```

nombre_trigger: es el nombre con el cual será identificado el objeto en la base de datos.

momento_ejecucion _ evento: puede ser BEFORE (antes) o AFTER (después) para indicar al trigger que se ejecute antes o después del evento que lo activa.

nombre_tabla: Indica el nombre de la tabla a la que estará asociado el trigger. Esta debe ser una tabla permanente, no puede ser una tabla temporary o una vista. Veamos algunos ejemplos sencillos de las tablas.

Primero crearemos un par de tablas para pruebas.

```
CREATE TABLE suma_multiplica(n1 INT, n2 INT, suma INT,  
multiplicacion INT);
```

Al agregar 2 números (n1, n2) a la tabla `suma_multiplica` esta deberá ser capaz de realizar la suma de los dos números, así como de la multiplicación de los mismos e insertar los resultados en sus respectivos campos suma, multiplicación.

Solución:

```
DELIMITER //  
CREATE TRIGGER tgr_suma_multiplica BEFORE INSERT ON  
suma_multiplica  
FOR EACH ROW BEGIN  
    SET NEW.suma = NEW.n1+ NEW.n2;  
    SET NEW.multiplicacion = NEW.n1 * NEW.n2;  
END//  
DELIMITER ;
```

Una vez creada la tabla y el trigger realicemos un par de inserciones de datos para ver el resultado:

```
INSERT INTO suma_multiplica (n1, n2) VALUES  
(2,5),(10,4),(4,5),(6,6),(8,20);
```

Ahora hagamos un select a la tabla `suma_multiplica` para ver el resultado generado:


```
SELECT * FROM suma_multiplica;
```

Los triggers nos permiten tener acceso a las columnas de las tablas mediante los prefijos NEW y OLD , donde NEW hace referencia a los datos nuevos que serán ingresados en la tabla y donde OLD hace referencia a los datos actuales que serán modificados.

INSERT tenemos disponible solamente NEW . columna .
UPDATE cuenta con OLD . columna y NEW . columna .

DELETE sólo cuenta con OLD . columna .

Con estos prefijos podremos realizar manipulación de datos.

Como nota final hay que recordar que no podemos asignar dos triggers en un mismo tiempo de ejecución y mismo evento, y aunque actualmente no están soportados. 

CONSOL

CONGRESO NACIONAL DE
SOFTWARE LIBRE

Una Nueva Visión Sobre las Tecnologías

www.consol.org.mx

2008
UACM

Universidad Autónoma
de la Ciudad de México

Nada humano me es ajeno

DEL **19** AL **22** DE
FEBRERO DEL **2008**
PLANTEL SAN LORENZO
TEZONCO

TEMAS

COMUNIDAD, FILOSOFÍA Y NEGOCIOS

DESARROLLO DE SOFTWARE

APLICACIONES

ADMINISTRACIÓN, SEGURIDAD Y REDES

EDUCACIÓN Y ACADEMIA

INVITADOS INTERNACIONALES

MARTIN LANGHOFF.

ROBERTO OLIVARES.

ALVARO LÓPEZ.

FEDERICO HEINZ.

LESLIE VOGT.

ALCIDES MORALES.



Entre algunas pláticas, trabajo y otras cosas, pareciera que el tiempo simplemente se esfuma tan, tan rápidamente.

Como sea, ya estamos en una entrega más de La Columna.pl. En esta ocasión quiero tocar un tema bien interesante que son esas pequeñas herramientas que nos pueden llenar de una manera muy especial alguna de nuestras necesidades. ¿Hay algo por ahí que quieras automatizar pero no te decides? Inténtalo.

El ejemplo que voy a poner aquí es acerca de una pequeña utilidad que hice para revisar el estado de la batería de mi laptop. Como es bien sabido por algunos de ustedes, amables lectores, utilizo una máquina PowerBook G4, sobre Debian GNU/Linux. En esta máquina hago la mayoría de mi trabajo, que se basa principalmente en desarrollo de software y algo de administración de sistemas. La mayoría de mi trabajo lo hago en esta máquina que es casi tan estable como una piedra. En fin, las computadoras Apple, como ésta, utilizan un microcontrolador que gestiona las funcionalidades de energía, tales como el control de suspensión, el nivel de batería y otras similares, llamado PMU. Lamentablemente las aplicaciones actualmente existentes que monitorean los atributos de PMU, en Linux sobre estas computadoras no me satisfacen del todo; Algunas son unas aplicaciones en GTK verdaderamente estorbosas; otras están integradas al panel de control de GNOME o KDE, del cual no utilizo ninguno. Opté por utilizar el control de batería de GKrellM, que es algo modesto, pero que de todas formas a veces no quería tener siempre en mi escritorio - que para empezar, no es que tenga un escritorio como tal, pues de un escritorio sólo utilizo algunos escritorios virtuales y algo de soporte de systray. Finalmente no estaba muy seguro de lo que necesitaba. Lo único seguro era mi inseguridad en el caso :-)

¡Una pequeña aplicación en Perl! claro, desde un emulador de terminal podría llamar esta aplicación y me diría qué porcentaje tengo de batería restante en mi máquina, sería lo más simple, sencillo y a mi gusto, sería perfecto. Pues vamos a poner las manos a la obra.

En mi máquina, la información de la batería se encuentra en `/proc/pmu/battery_0`, que luce así:

```
flags      : 00000011 charge      : 3535
max_charge : 3537
current    : 0
voltage    : 12495
time rem.  : 0
```

El archivo indica diferentes valores para amperaje y voltaje. Los que me interesan son los valores de `charge` y `max_charge` que indican miliamper/hora de la carga actual que tiene la pila y la máxima.

Lo que hay que hacer es bastante simple. Obtener el valor de `charge`,

el de `max_charge` y obtener un porcentaje del primero con respecto al segundo. En esta columna tocaré algunos temas básicos pero que pueden ser de utilidad para entender algunos conceptos más adelante en otras columnas.

```
#!/usr/bin/perl use warnings;
use strict;
```

Mando llamar al intérprete de Perl que en la mayoría de las distribuciones con binarios precompilados irá a dar a `/usr/bin/perl` e inmediatamente mando llamar dos pragmas de Perl: El primero, `warnings`, para controlar advertencias y el segundo, `strict`, para restringir construcciones inseguras. Es bueno siempre llamar ambos pragmas para educarnos un poco al programar. Un pragma es únicamente un indicador para el momento de la compilación, podremos estudiar un poco los pragmas en futuras columnas.

```
my $pmufile = '/proc/pmu/battery_0';
```

Declaro la variable `$pmufile` que indicará la ruta absoluta del archivo que mencionaba anteriormente.

```
my $filecont; {
    local $/ = undef;
    open(PMU, $pmufile) or die $!;
    $filecont = <PMU>;
}
```

Aunque el uso de `local` ya no se aconseja mucho, es bueno a veces para utilizarlo en bloques específicos donde sólo tengamos que modificar, quizás, una variable especial, como en este caso, la variable especial `$/`. Esta variable nos sirve para definir el input record separator, que por default, es una línea nueva. Modificarlo nos permite hacer cosas interesantes, sobre todo cuando efectuamos un ciclo `while()` hacia un `filehandle`. En este caso, indefinimos su valor, por lo cual, luego de abrir el archivo `$filename` a lectura, vaciamos todo el contenido del archivo en una sola variable, `$filecont`, que habíamos declarado antes del bloque. En alguna columna futura tengo pensado tratar a fondo las variables especiales, que son algo muy interesante dentro de Perl.

```
my($charge) = $filecont =~ /charges+:s(d+)/;
my($max_charge) = $filecont =~ /max_charges+:s(d+)/;
```

Aquí creamos las variables `$charge` y `$max_charge` y les asignamos el

valor que nos arroja nuestra expresión regular como la variable especial \$1, en ambas líneas.

```
my $porc = int (($charge/$max_charge) * 100);
```

Dividimos \$charge entre \$max_charge y multiplicamos el resultado por cien, para obtener el porcentaje. Sin embargo, este resultado nos arrojaría más de diez decimales, después del punto. Por ello, utilizamos la función de Perl, int, que nos regresa el valor entero de su parámetro (mi incipiente naturaleza matemática me hace hacer notar que un número entero no es necesariamente un número natural :-)).

```
print "t", "Restante: $porc %", "n";
```


print en realidad requiere una lista como argumento, lo que generalmente hacemos en enviarle un sólo parámetro (por ejemplo, print "Hola mundo");, sin embargo, para una notación mucho más intuitiva, ponemos el salto de línea y el tabulador como primer y tercer parámetro, una lista, precisamente.

¡Y listo! Le agrego un bit de ejecución al script, lo renombro a pila, lo pongo en /usr/local/bin o algo similar y puedo correrlo fácilmente:

```
damog@cochina:~$ pila           Restante: 86 %
damog@cochina:~$
```

Ya luego podría usar ese comando para hacer otras cosas: parsear el valor y podría enviarme un correo avisándome que ya se me va a acabar la pila, o mandar un mensaje al systray o quizás conectarme al IRC y avisarle a mi nick al respecto; hay muchas cosas más que Perl nos facilita hacer, ¡tu imaginación es el límite!

De otras muchas formas habríamos podido abrir el archivo y leer el contenido de charge y max_charge. ¿Tú de qué forma lo hubieras hecho? ¿Quizás ciclar el filehandle hasta toparse con /^max_charge/? ¿Quizás usar alguna combinación con ejecución externa de cat o de grep?

Te invito a que me envíes tus comentarios. Nos leemos en el próximo número. 

Referencias útiles

<http://perldoc.perl.org/index-pragmas.html>

<http://perldoc.perl.org/perlvar.html>

<http://perldoc.perl.org/functions/int.html>

<http://perldoc.perl.org/perlop.html#Integer-Arithmetic-integer>

RevistaSL: ¿Proyecto de divulgación o locura colectiva?

Staff SL
buzon@revista-sl.org

Hace tres años RevistaSL "La revista mexicana del Software Libre" se presentaba en el Congreso Nacional de Software Libre" ... hoy RevistaSL "El Software Libre hecho revista" regresa al mismo congreso a agradecer y mostrar los logros alcanzados gracias al incondicional apoyo de la comunidad internacional de Software Libre.

El inicio

En la cálida ciudad de Chetumal, localizada al sureste de México, la Comunidad Linux Chetumal (CLC) inició un ambicioso proyecto de divulgación a nivel local. La elaboración de un columna dedicada a la computación, las nuevas tecnologías y el Software Libre; publicada en el Diario de Quintana Roo.

Los "chetulinixeros", queriendo alcanzar un mayor nivel de difusión, buscaron el apoyo de la comunidad mexicana de Software Libre mediante sitios Web de impacto nacional atrayendo nuevos entusiastas con ganas de apoyar la idea de la columna.

Durante varias semanas la Columna DQR, como se le llamó al proyecto, se publicó en el periódico local y en el sitio Web de la CLC. Sin embargo, y a pesar del éxito, el proyecto dejó de publicarse.

La CLC y los colaboradores de la Columna DQR, no desistieron. Gonzalo González, presidente de la CLC, con el apoyo de la Fundación de Software Libre de México ideó un nuevo proyecto de divulgación, que tenía como objetivo contar con alcance a nivel nacional.

Después de meses de planeación, y sin dar noticia alguna a los colaboradores de la Columna DQR, Gonzalo publicó el 17 de Febrero de 2005 el número 0 del nuevo proyecto denominado "RevistaSL, la revista mexicana del Software Libre". Ante la sorpresa de los colaboradores de la Columna DQR, quienes no tenían idea de este nuevo proyecto, Gonzalo les incluyó como editores de RevistaSL.

RevistaSL tuvo un gran aceptación por parte de la comunidad de Software Libre en México, lo que le ganó varios cientos de descargas, decenas de comentarios y felicitaciones y la integración de más personas al proyecto. Para dar a conocer mas aún el proyecto, una semana más tarde, Augusto Lozano, quien había colaborado en la Columna DQR, presentó el proyecto RevistaSL, en el marco del Congreso Nacional de Software Libre 2005, a desarrolladores, usuarios y entusiastas del Software Libre en México.

Un mes más tarde, RevistaSL publicaba un nuevo número y anunciaba el cambio a publicación bimestral en lugar de mensual. Este número nuevamente tuvo la gran aceptación por parte de la comunidad de Software Libre, lo que lo acomodaba entre las preferencias de geeks y no tan geeks.

A pesar del notorio éxito del proyecto, el precario equipo editorial tuvo problemas para darle continuidad al siguiente número, esto debido a la falta de colaboradores externos, múltiples ocupaciones de los editores y escasa comunicación que existía entre los editores, que por su



alejamiento geográfico, contaban con una mala coordinación. A pesar que durante varios meses el equipo editorial no contó con el suficiente material para liberar un nuevo número, los ánimos no decaían.

Poco más de un año después de la liberación del primer número, la CLC organizó la primera edición del Congreso de Software Libre de la Frontera Sur (FS), que reunió a desarrolladores, investigadores, usuarios y entusiastas del Software Libre de México y Belice en Chetumal. Entre los asistentes se encontró el equipo editorial de RevistaSL.

Entre las múltiples convivencias de los ponentes, el tema de "la desaparecida" RevistaSL se hizo presente y después de algunos vodkas y cervezas, los ánimos por retomar el proyecto habían inundado a los editores.

Lamentablemente, sin el efecto causado por el vodka y la cerveza, editar un nuevo número de RevistaSL se tornaba todo un reto, el equipo editorial hechó mano de las nuevas amistades formadas durante el FS para conformar un nuevo equipo editorial, lo suficientemente capaz para aceptar este gran reto... esta gran locura.

Una semana más tarde, "RevistaSL, la revista mexicana del Software Libre" quedaba en olvido, en los principales sitios Web referentes al Software Libre, se anunciaba un proyecto de divulgación más grande, tan grande que el adjetivo "mexicana" no lo describía, se anunciaba: "RevistaSL, El Software Libre Hecho Revista".

Este nuevo número superó a los anteriores, y a pesar de la gran cantidad de errores gramaticales, ortográficos, de estilo y algunos incluso desconocidos, llevo a la primer gran meta de RevistaSL, superar las mil descargas.

Todo este éxito no hizo más que aumentar las ganas de trabajar en el equipo editorial que no terminaba de contar las descargas, cuando ya pensaba en su próximo número, el siguiente, el siguiente... y el siguiente.

“RevistaSL es un registro de lo que un grupo de locos puede llegar a hacer para dar a conocer a los demás lo que hay en el mundillo del software libre” - Julio Acuña

La situación actual

A tres años de trabajo, el equipo editorial de RevistaSL ha publicado diez números, que si bien no han sido estrictamente bimestrales, han tenido la suficiente continuidad para ganarse el afecto de lectores en México, España, Colombia, Venezuela, Perú, Argentina, Cuba, Chile, Bolivia, Uruguay y otros tantos países más.

A tres años de trabajo, por las páginas de RevistaSL han pasado importantes desarrolladores como José Carlos Nieto, David Moreno, Gerardo Horvilleur, Marcela Tiznado, Fabien Mannessier, Sandino Araico, Alvaro López, Matt Dillon, Richard Stallman y muchos, muchos más.

A tres años de trabajo, los cientos de descargas se han convertido en miles, tantos miles que han superado las veinte mil descargas en un sólo número...

A tres años de trabajo, RevistaSL ha fungido como medio asociado en congresos en México, Belice, España y Brasil como es el caso del Congreso de Software Libre de la Frontera Sur (México - Belice), Congreso Nacional de Software Libre (México), Encuentro Nacional Linux (México), el Festival de Software Libre (México), FLOSS International Conference (España), Latinoware (Brasil), it1k1: Hacker Conference (México), por mencionar algunos.

A tres años de trabajo, RevistaSL sigue siendo un proyecto entre amigos, abierto a todo aquel que quiera colaborar, que atrae a los desarrolladores a escribir sobre sus proyectos...A tres años de trabajo, “habemus” RevistaSL para mucho, mucho rato más ...

“Es un medio en el cual se difunde el Software Libre de manera directa a sus lectores, en ella se dan a conocer los temas de mucha actualidad en este mundo del SL” - Ivan Zenteno

Los culpables

RevistaSL cuenta con un sólido equipo editorial conformado por desarrolladores, diseñadores, difusores, usuarios, investigadores y adictos a las computadoras. Todos y cada uno de ellos son quienes sacrifican horas de sueño por hacer esta locura, realidad:



Gonzalo Javier González Rodríguez
gonzalo.gonzalez@revista-sl.org

El "Jefe" y culpable de que pasemos noches enteras sin poder dormir al tener que trabajar sobre la revista.

De día es un padre de familia que trabaja en el Diario de Quintana Roo resolviendo problemas relacionados a las TI; de noche asigna tareas, administra los recursos de la revista y embriaga a sus editores en el paradisíaco Chetumal.

Carlos Augusto Lozano Vargas
augusto.lozano@revista-sl.org

Calificado como irreverente y alternativo, un aficionado a la acuariofilia, el cortometraje y el teatro, que pasa su vida escuchando canciones de Belinda y viendo telenovelas.

De día se hace pasar como estudiante de ingeniería en el Instituto Politécnico Nacional de México; de noche toma el mouse como su mejor herramienta y se sienta a maquetar el PDF, recibe los regaños de la gente experimentada del equipo editorial y sin látigo ni bat intenta organizar el trabajo editorial.

Julio Mauricio Acuña Carillo
julio.acunya@revista-sl.org

Un amante de las bellas mujeres que pasa su vida leyendo o escuchando jazz y rock progresivo, o en sus ratos libres programando aplicaciones en Python, Django, PostgreSQL todo sobre su distribución GNU/Linux favorita: Debian.

De día, Julio trabaja en el gobierno, atendiendo las múltiples necesidades en tecnologías de información que surgen; de noche toma su grabadora y sale a entrevistar a las personalidades más importantes del mundo del Software Libre, y cuando no lo está haciendo, corrige artículos, traduce o simplemente comparte el látigo de la organización con Sonia.

"Lugar donde puedes desahogar y expresar tus conocimientos para los demás" - Andres Vargas

Sonia Sánchez Díaz
sonia.sanchez@revista-sl.org

Una amante de la música y los conciertos, a los cuales asiste aun cuando se rompa las piernas en ello o sufra de daños extremos al transbordar en la línea 9 del metro de la Ciudad de México rumbo al Foro Sol o el Palacio de los Deportes. Alguien quien adora viajar, visitar museos, leer y que siempre se le puede encontrar bebiendo un café en Starbucks o en un vagón del metro, con maleta en mano, rumbo a la estación de autobuses TAPO para viajar a Puebla, o sentada en primera fila del teatro o del cine para pedir un autógrafo de Diego Luna o paseando en Ciudad Universitaria, su alma máter...

Sonia de día colabora de forma profesional en una publicación completamente mexicana de tecnologías de información; de noche pone orden al trabajo en RevistaSL, hace revisión de artículos, envía recordatorios de los pendientes y saca el látigo para que los integrantes del equipo editorial hagan su mejor trabajo.

Andrés Bernardo Vargas Rodríguez
andres.vargas@revista-sl.org

Un carismático programador, recién egresado de la Licenciatura en Ciencias de la Computación, que nunca se pierde un congreso de computación, pues siempre está divulgando el uso de Software Libre, y cuando no lo hace, se le puede hallar sentado frente a su computadora escuchando música punk rock y desarrollando alguna nueva aplicación en Python,

Andrés, de día termina sus últimos deberes escolares; de noche escribe artículos para RevistaSL o planea la forma de hacer llegar aun mas allá, el alcance del proyecto

Mónica Aisha Aguilar Gómez
monica.aguilar@revista-sl.org

Recien graduada de la facultad de Derecho ayuda a su novio Andrés Vargas a escribir y corregir sus articulos para la Revista SL. No usa ninguna distribucion de gnu linux pero sus proyectos los hace en el openoffice.

Victor Hugo Cordova Madrid
victor.cordova@revista-sl.org

Un fanático del fútbol que gusta de ver los partidos en la televisión, tapizar su recámara con pósters y afiches de su equipo favorito o simplemente tomar una cerveza en la bahía de Chetumal.

De día trabaja en el gobierno y estudia su carrera universitaria; de noche lee las preguntas, comentarios y retroalimentación que los lectores envían a RevistaSL

“Criticar el proyecto que tu mismo elaboras es tarea difícil. Como se titula este artículo, RevistaSL es una locura colectiva, una locura que es seguida por miles alrededor del mundo, una locura que traspasa fronteras, una locura que deja un granito de arena en la difusión del código abierto, y que a modo personal, forja grandes amistades... creo que voy a llorar” - Augusto Lozano

“Es una forma de dar a conocer lo que un grupo de aficionados al software libre puede llegar hacer” - Eyden Barboza

José Luis Galicia Sánchez
jose.galicia@revista-sl.org

Un "chetulanuxero" amante del PHP y el futbol quién de día se dedica a brindar soluciones Web, mientras de noche colabora con RevistaSL.

Jesús Antonio Balam Jiménez
jesus.balam@revista-sl.org

Un programador de Quintana Roo que quién fue de los primeros editores con lo que conto RevistaSL

Julio Cesar Sosa Yeladaqui
julio.cesar@revista-sl.org

Jesús Antonio Alvarez Cedillo
jesus.alvarez@revista-sl.org

Un profesor estricto quien gusta de cantar, salir con los amigos por cervezas y simplemente entrar a las salas de chat a platicar con sus amigos.

De día es un respetable investigador del Instituto Politécnico Nacional en México; de noche escribe, revisa y traduce artículos para la revista.

Enrique Alfonso Sánchez Montellano
enrique.sanchez@revista-sl.org

Amante de los videojuegos, el anime y la inteligencia artificial

De día busca fallos de seguridad que explotar; de noche se sienta a escribir artículos para la revista y aportar nuevas ideas a seguir.

Edgar Guerra Rey
edgar.guerra@revista-sl.org

Diseñador gráfico y animador por computadora que pasa su vida con su novia, o comiendo comida mexicana en los restaurantes de las calles de Bogotá.

De día se dedica a la edición gráfica de animaciones; de noche colabora con el arte de RevistaSL, el es autor de dos de los logos que han representado el proyecto.

Josué Gutierrez Hernandez
josue.gutierrez@revista-sl.org

Un artista con amor a los bytes, rockstart supremos del Guitar Hero dedicado al diseño y a la programación.

De dia trabaja desarrollando aplicaciones de software; de noche diseña las portadas y el arte gráfico de RevistaSL.

Ivan Alfredo Zenteno Aguilar
ivan.zenteno@revista-sl.org

Iván, alias el k001operator egresado de la carrera de Ingeniería en Sistemas Computacionales hace ya 3 años, es supuestamente programador en el día, actualmente desempleado por convicción y necesidad psicológica. De noche, supuestamente colabora con la

“RevistaSL, es uno de los mejores medios electrónicos de difusión de proyectos, noticias, tips, etcétera sobre Software Libre, me siento orgullosa de poder colaborar con todo este equipo de personas que trabajamos en diferentes puntos del país y que nos une algo en común: un pingüino llamado Tux.” - Sonia Sánchez

RevistaSL, pero lo que sí es seguro, es un entusiasta del Software Libre donde quiera que se encuentra.

[Eyden Barboza Barela](#)
eyden.barboza@revista-sl.org

Eyden mejor conocido como Mantus, originario del ya antes mencionado Chetumal y sin dudar alguna un fanático del software libre.

De día es estudiante de Ing. en Sistemas Computacionales en el Tecnológico de Cancún; de noche mantiene el sitio Web del proyecto.

[Héctor Leal Morales](#)
hector.leal@revista-sl.org

Ing. en Comunicaciones y Electrónica del IPN en México, amante del basketball, las fiestas y de los buenos negocios.

Héctor de día dirige su propia empresa de desarrollo Web; de noche se encarga de llevar RevistaSL al mundo, buscar patrocinadores y distribuir la Revista entre sus clientes.

[Miriam Elizabeth Cacique Acevez](#)
miriam.cacique@revista-sl.org

Diseñadora gráfica quien desde hace unos meses trabaja en una empresa de desarrollo Web.

De día trabaja en la creación de imágenes, publicidad y cuanto cosa rara se le ocurre a su jefe; de noche colabora en la creación de portadas del proyecto.

[Martín Trejo Chávez](#)
martin.chavez@revista-sl.org

Martín Trejo es un desarrollador autodidácta que trabaja actualmente como consultor de TI. Le gusta escribir código, aunque no es necesariamente el desarrollador más productivo. La elegancia y la inteligencia las ve como partes inherentes de una aplicación, desde la arquitectura, hasta el diseño de la base de datos, pasando por el modelo del dominio y tocando ligeramente la experiencia del usuario. Se siente a disgusto con las metodologías formales y burocráticas, tampoco se siente a gusto con las técnicas en extremo ágiles y novedosas. La metodología ideal es aquella que consigue hacer soluciones sólidas, reutilizables, confiables y que además consiguen que los participantes crezcan con cada iteración. Desafortunadamente, todavía no conoce tal metodología.

[Luis Fernando Peniche Novelo](#)
luis.peniche@revista-sl.org

Sofador profesional. Activista que paga la deuda de identidad que tiene con el software libre apoyando la revista. Radica desde hace seis meses en comunidades mayas del oriente de Yucatán, donde sufre de

donde sufre de síndrome de abstinencia tecnológica, pero se apura a difundir la tecnología inalámbrica y los centros comunitarios como un medio para abatir el rezago tecnológico de la región (Y reducir los síntomas de su síndrome).

Los fines de semana, armado con una laptop con Debian GNU/Linux, firmada por Richard Stallman y adornada con una calcomanía del Subcomandante Marcos, construye el otro mundo posible, apoyando con alojamiento y administración de sitios web a organizaciones sociales, de derechos humanos y medios de comunicación independientes; por la noche, ya muy noche, cuando los entiende, corrige la ortografía y redacción de los artículos de la RevistaSL.

[Vanessa Johanna Salcedo Weber](#)
vanessa.johanna@revista-sl.org

Estudiante universitaria de idiomas, que en su lado oscuro pasa el tiempo participando en la metadistro para computadoras de bajo rendimiento llamada Ikim o colaborando con el proyecto Debian Women.

De día pasa su vida en la escuela obteniendo buenas calificaciones; de noche corrige artículos, traduce o simplemente toma el bat para presionar a que la maquetación se haga de forma más rápida.

[Alejandra Fierro Chacón](#)
alejandra.fierro@revista-sl.org

Estudiante que gusta pasarla bien con sus amigos, y que incluso se integro al equipo editorial durante una fiesta.

Alejandra de día es estudiante de una universidad en Chihuahua; de noche se dedica a la corrección de artículos para la Revista.

[Humberto Morales](#)
humerto.morales@revista-sl.org

Un artista digital quién pone humor al proyecto con una tira comica. Además Humberto nos brinda ideas sobre el diseño de la revista.

"Algo difícil intentar definir Revista SL: Creo que me atrevería a decir que es uno de los proyectos más maravillosos donde trabajamos personas comunes y corrientes, con el único objetivo de hacer que el Software Libre sea difundido a todas las fuentes a las que pueda hacer llegar el ciberespacio" - Johanna Salcedo

Detrás de cámaras...

Al público RevistaSL es un proyecto coordinado, fruto de dos meses bien planeados de trabajo. La realidad puede ser muy diferente.

RevistaSL es un proyecto que dista mucho de la descripción anterior, pero no por ello cae en el desastre... o tal vez sí.

La magia comienza en una junta de organización, dado las grandes distancias geográficas entre los participantes, la junta se realiza en el canal de IRC; una junta en donde en momentos se comienza con un tema y se termina tratando otros cinco simultáneamente, donde los "slaps" dominan, las teclas no bastan y la confusión reina. En esta complicada junta se analizan los resultados pasados, número de

descargas, comentarios y se platican las anécdotas extrañas que pudieran haber ocurrido. Es aquí en donde se decide el tema central del próximo número, fechas de entrega de artículos e incluso, alguno que otro valiente se atreve a prometer artículos.

De esta junta se extrae una lista de las actividades a realizar, actividades que se envían a una lista de correo y se discuten a fondo. Los correos diarios llegan a ser en momentos demasiados, lo que provoca que muchas veces se opine sin querer de algo que se había resuelto en otro mensaje. Para ello, se anexa un log de la junta de IRC, log que rara vez es legible dada la moderación de la junta.


Todo parece marchar bien, se tienen fechas planteadas, así como invitados a colaborar; la magia termina unos días antes del cierre de recepción de artículos momento en que el equipo editorial ve la cuenta de Google Docs, así como la bandeja de entrada del correo electrónico dedicado a las colaboraciones, vacías. El nerviosismo reina hasta el último día de recepción, sin embargo y casi como ley natural, a media noche los artículos comienzan a llegar, la participación de lectores y editores se vuelve notable, y los corazones del equipo editorial nuevamente toman su ritmo normal de latido.

Con artículos recibidos, y la mayoría de las veces otros en redacción, el equipo de corrección ortográfica asume la monstruosa tarea de la minuciosa caza de faltas gramaticales y ortográficas, varias veces los artículos son revisados por distintas personas, sin embargo, el toque SL esta presente, el equipo editorial ávido de hallar errores corrige lo corregido y edita lo editado. Sin embargo no se puede demeritar el trabajo de corrección que número a número mejora la calidad editorial de la revista. ¿Qué seríamos sin sus regaños?

Una vez corregidos la mayor parte de los artículos, es tiempo de maquetar, los diseñadores envían su propuesta de portada, propuestas que son votadas de forma "estratégica", a pesar de ser nuestros amigos, las portadas son criticadas fuertemente con el fin de brindar una agradable primera vista al proyecto. Con portada o sin portada, la revista se empieza a crear.

RevistaSL deja a un lado los equipos de múltiples procesadores que facilitan el diseño en los medios impresos, a cambio, la vieja "turbo bob" hace su aparición. RevistaSL es un proyecto libre y como tal es maquetado con Software Libre, la creación de la revista se hace sobre una vieja computadora que corre un sistema Slackware GNU/Linux, y en su momento se hace uso de GIMP, y Scribus, este último el software mes el más importante en todo el proceso.

Días antes de la liberación e incluso algunos días después, el equipo editorial sigue su trabajo, los archivos fuente de Scribus son enviados por correo para correcciones de última hora, cuando regresan nuevamente, se corrige el diseño, el cual fue alterado por el manejo de diferentes versiones.

Después de meses de trabajo, sin jugo ni comida, y con la tensión en el canal de IRC, se lee la frase que todos esperan: "Habemus RevistaSL", frase que marca el fin de una nueva edición, frase que marca el resultado de muchos días de trabajo, frase que marca el esfuerzo de un grupo de locos que de la noche a la mañana pasaron de ser personas normales que disfrutaban de los placeres de la vida, a convertirse en parte del equipo editorial más desordenado y más exitoso del Software Libre. 

Durante el Simposio "Privatización del conocimiento" celebrado el pasado mes de Noviembre en las instalaciones de la Universidad Nacional Autónoma de México, fué del agrado de los asistentes la participación de Richard Stallman, uno de los grandes iconos del movimiento de Software Libre.

RevistaSL no podía perder la oportunidad de asistir a esta conferencia.

RSL: Muy bien. He estado leyendo bastante del Software Libre y entiendo que es más una filosofía que otra cosa. ¿Cómo haces para que el usuario común que no sabe nada de Software Libre se interese?

RMS: Primero antes de explicar "cómo" tengo que decir qué es la idea, de otra forma no funciona. Entonces, El Software Libre quiere decir un software que respeta la libertad del usuario y la solidaridad social de su comunidad. Hay cuatro libertades esenciales que el programa debe de llevar para todos sus usuarios. Son, la libertad 0, la libertad de ejecutar el programa como quieras; la libertad 1, la libertad de estudiar el código fuente del programa y cambiarlo para que haga lo que quieras; la libertad 2, que es la libertad de ayudar a tu prójimo, la libertad de distribuir copias a los demás cuando quieras; y la libertad 3, que es la libertad de contribuir con tu comunidad, es la libertad de distribuir copias de tus versiones cambiadas cuando quieras. Con estas cuatro libertades tu programa es libre, es decir que cada usuario puede usarlo, puede hacer con él lo que quiera y los usuarios tienen el control, si una de estas libertades falta, es decir que el programador tiene el control de los usuarios, es decir que los mantiene en un estado de división o impotencia, y nuevamente los dos al mismo tiempo, entonces usando Software Libre podemos tener una sociedad libre, usando software no libre, es decir .

software privativo que limita la libertad de sus usuarios, hacemos la informática bajo la dictadura de los dueños de los programas. Es una lección entre libertad y democracia a un lado, y la dictadura al otro lado. El software privativo es un sistema colonial, un sistema de colonización electrónica, algunas empresas han colonizado las computadoras de mucha gente y otras, menos computadoras, pero su meta es colonizar. Mantienen, como todo sistema colonial, a la gente en sus sistemas de división e impotencia.

Entonces como sabemos que no practican la informática y no aprecian la libertad tanto o no ven cómo se aplica la libertad a este aspecto de la vida. Puedo usar una analogía entre los programas y las recetas de cocina, porque los dos tienen una similitud. Los dos son obras de uso práctico para hacer obras en la vida, un programa es una serie de etapas para llegar a un resultado deseado, también una receta y por lo tanto las mismas cuatro libertades son esenciales para las recetas, pero el estado no intenta negarnos las cuatro libertades para las recetas, los cocineros pueden disfrutar de las cuatro libertades en las recetas. Uno puede cocinar libremente la receta, uno puede estudiar y cambiar la receta es normal, cambiarlo muchos lo hacen, a veces uno hace cambios pequeños pero algunas saben hacer cambios más importantes; también el copiar y distribuir copias de la receta. Los cocineros lo hacen cuando lo quieren, y también hay copias cambiadas de la receta. Hay dos recipientes, tendré que usar dos, pero funciona...

Usare dos ¡ah! si, ¡ah! ordenadamente tengo dos, pero sería un



Conferencia de Richard Stallman, en la foto César González, Odin Mojica, Rommen Sánchez, Eduardo Ruiz entre otros.

desperdicio usar éste para la pequeña cantidad que queda, tengo uno aquí.

RSL: Si, no puedes tener 5 minutos aquí y 5 en otra

RMS: No funciona bien eso, entonces imagina que algún día el estado diga a los cocineros: desde hoy si copias o cambias una receta te llamaremos pirata y te pondremos en prisión, imagina el enojo que habrá al ser negada esta libertad, ahora todos los cocineros, salvo algunos cocineros naturales que no necesitan las recetas, comprenderán por que el software debe de ser libre.

RSL: Muy bien, ya entiendo la analogía, es muy indicada para comprender mejor el Software Libre, pero digamos que yo también quiero ser libre pero no sé nada de programación, ni sé nada de informática

RMS: Todavía puedes ejecutar como quieras un programa que quieres usar, todavía puedes copiar el programa y distribuir copias, entonces una gran libertad de esas libertades te sirve directamente. Puedes aprovechar estas libertades, las otras dos, la libertad de cambiar y estudiar el código fuente y la libertad de distribuir copias de tus versiones cambiadas no sabes ejercer, pero todavía podrías ejercerlas indirectamente, podrías convencer a tu primo programador que cambie un programa por ti o podrías pagar a un programador que cambie el programa como quieras. Entonces es muy útil para ti que los usuarios



tengan estas libertades, y también otros cambian el programa y los cambios que les gustan a los usuarios son aceptados y otros cambios se rechazan, entonces los usuarios colectivamente tienen el control del desarrollo y del futuro del programa.

RSL: Y nadie lucra con el Software Libre...

RMS: No, no es verdad, no es verdad, si pagas a tu primo o al programador que cambie el programa por ti, él sí lucra, no estamos en contra del negocio, estamos por la libertad y la solidaridad social, entonces no aceptamos que nadie tenga poder para mantenernos impotentes, pero que alguien gane dinero aprovechando la libertad de los demás no nos molesta.

RSL: Richard, ¿Tú sabías que en Estados Unidos patentaron posiciones del Yoga?

RMS: ¿Posiciones de qué?

RSL: Yoga

RMS: No sé

RSL: Patentaron como 2mil posiciones de yoga y el gobierno de la India denunció al gobierno de Estados Unidos, porque es absurdo patentar posiciones del cuerpo de uno

RMS: Sí, es absurdo, el sistema de patentes es absurdo

RSL: Exacto, y también el gobierno de la India levantó el mismo ejemplo que tu indicas. Indicó que las posiciones del yoga son tan antiguas y tienen tantas tradiciones...

RMS: Pero es otra cosa

RSL: No por que...

RMS: Las cosas de la gente no pueden poder ser patentadas, ni siquiera en los Estados Unidos, es un ejemplo de la función incorrecta del sistema de patentes pero, es un error enfocar las patentes absurdas, porque una patente que no es absurda puede ser igualmente dañina, es un error enfocar como ejemplo las patentes que no siguen la ley.

RSL: Pero de hecho siguen la ley porque fueron patentadas

RMS: No, no, no, no... ya eran conocidas, entonces...

RSL: Pero fueron patentadas, los que patentaron hicieron variaciones, entonces la ley las tomó como originales y entonces cumplían el requisito de la originalidad, algunas de las posiciones. Entonces pusieron el mismo ejemplo de la comida, el gobierno de la India, otros pusieron otro ejemplo. Y dijeron, sí, es verdad que las recetas no se pueden patentar

RMS: Es que las recetas no se pueden patentar...

RSL: Si, es que la receta de la Coca Cola está patentada...

RMS: No, no lo es, es secreta, no es patentada

RSL: ¿No está patentada?

RMS: No

RSL: Pero nadie puede compartir, o sea, no tiene las cuatro libertades.

RMS: No, no comprendes bien. Hay una sola entidad que tiene la copia de la receta, es la empresa de Coca Cola y ella sí tiene las cuatro libertades, pero la libertad de distribuir copias no es una obligación.

RSL: Como Microsoft

RMS: No lo comprendo, no, no veo como Microsoft sea similar, no, no lo comprendo, no puedo decir que es verdad o falso por que no lo comprendo. No veo como pueda haber similaridad. Microsoft distribuye copias de Windows, hay muchos que tienen copias de Windows pero Microsoft los restringe en el uso de sus copias, es completamente diferente. Si alguien tiene un programa y no lo distribuye, nadie tiene una copia, entonces pierde una comodidad pero no libertad.

RSL: Mmm... ya, pero entonces...

RMS: Si tu amigo tiene una receta, pero no quiere darte una copia, no ataca tu libertad o sea, no es amable que es otra cosa. Sí podrías decirle que no es amable, que no es un buen amigo sino comparte la receta contigo.

RSL: Pero, me vende una copia

RMS: Es lo mismo

RSL: Como la Coca Cola, me vende una lata...

RMS: No, Coca Cola no vende copias de latas, hay que distinguir las copias, si mezclas todo no comprendes nada. Una botella de Coca Cola no es la copia de la receta es el producto de la receta, es otra cosa, es como la salida de un programa. La receta es como el programa

RSL: Pues entonces ahí esta el punto, porque el usuario entiende muy bien las ventajas del Software Libre

RMS: Porque con el Software Libre puede compartir, si tiene una copia y es Software Libre puede compartirlo, puede hacer otras copias y darlas o venderlas a los demás cuanto quiera, incluso puede compartir con sus amigos y también puede estudiar lo que le hace el programa. La receta de Coca Cola no la usamos por que no tenemos copias, Windows podríamos usar porque podemos tener copias, pero si las usamos, Microsoft tendría control completo de lo que hagamos, es mejor no tener Windows y no lo tengo. No digo que todo mundo debe toda ayuda que pueda pedir o todo favor, pero respetar mi libertad sí, todos deben respetar mi libertad

RSL: ¿Sabes cuál es el problema? Que la gente no siente que tenga esa libertad

RMS: No reconocen el control que las empresas tienen sobre ellos, pero de verdad, muchos programas privativos contienen utilidades para vigilar al usuario o para restringir al usuario.

RSL: ¿Entonces cómo me controlan como usuario de software?

RMS: Conozco unos ejemplos, por ejemplo, Windows le dice a Microsoft lo que el usuario hace, Windows Media Player dice todas las cosas que el usuario mira, Microsoft no tiene el derecho de saber, pero lo sabe porque han puesto esta funcionalidad en el programa, los usuarios no pueden quitarlo, entonces si usan Windows Media Player siguen diciéndole a Microsoft todas estas cosas que el usuario mira. El usuario puede querer que cese de decirlo a Microsoft, pero aquí nota que no tiene la libertad número 1, es decir, no puede cambiar el programa, Microsoft tiene el control, el usuario no, con Windows Vista es peor aún, porque fue concebido para restringir más al usuario. Windows Vista no funciona con varios dispositivos por que Microsoft quiere imponer el remplazo de esos dispositivos por otros concebidos para restringir más al usuario.

Windows Vista podría funcionar con esos dispositivos, pero muchos tienen copias pero nadie puede cambiar su software para funcionar con esos dispositivos, y ¿Por qué?, ¿Por qué Microsoft lo hace?... son dispositivos para poner imagen en la pantalla, y usan pantallas y Microsoft quiere restringir más que nunca el acceso a las imágenes en la pantalla, entonces ha decidido que Windows Vista no funcione con esos dispositivos, entonces exigen que el usuario compre un dispositivo nuevo que funciona peor. También Microsoft tiene la capacidad de enviar a cualquier computadora con Windows Vista un comando de apagarse y no funcionar más.

RSL: ¿Sí?, ¿Qué comando es?

RMS: No sé. Pero no es secreto, entonces también Microsoft puede cambiar su sistema en cualquier momento. El usuario de Windows Vista no tiene la opción de negar el cambio. Con otras versiones de Windows sí, el usuario puede decir no quiero que se instalen actualizaciones, el usuario de Windows Vista no puede, tampoco con Mac OS, Apple puede cambiar el sistema en cualquier momento como quiera, tiene el control completo.

RSL: ¿Crees que Windows Vista tendrá éxito?

RMS: No puedo ver el futuro (Risas...)
Tenemos una campaña contra la adopción de Windows Vista para disminuir su éxito.

RSL: ¿Y en qué consiste la campaña?

RMS: Oh esa es buena. Pueden ver en el sitio en el sitio badvista.org Porque es una nueva vista.

RSL: Y el ejemplo que estabas poniendo de Windows Media Player, que cuando tu actualizas tu Windows XP, te baja un software para autenticar la licencia, eso también es un espía.

RMS: Sí, es

RSL: Pero entonces el cambio hacia el uso del Software Libre tiene que ser individual

RMS: No comprendo, individual en este contexto ¿Qué quiere decir?

RSL: Es decir, yo tengo que adoptar el uso de Software Libre por mi sola, sin un organismo grande que me esté empujando.

RMS: Si hay organismos grandes en varias partes del mundo.

RSL: ¿Dónde?

RMS: En Extremadura, en una parte de India.

RSL: Esa parte de India es muy interesante, tiene mucho movimiento.

RMS: Pero la pregunta no tiene sentido, si nadie tiene poder sobre lo que haces, obviamente tu decisión es una decisión individual.

RSL: No, si vosotros os dirigís a cada uno de los individuos para que tome esa decisión o también hay otra...

RMS: Hacemos todo lo posible, cuando hay un grupo nos dirigimos a todo el grupo. Donde hay un individuo nos dirigimos a todo un individuo.

RSL: Richard, en Berkeley recuerdo que dijiste que ahora tu labor es política.

RMS: Sí

RSL: Pero es una manera de hacer política..

RMS: ¿Te sorprende?. Es una cuestión política, es una cuestión de si el usuario tiene libertad o el desarrollador tiene poder y siempre era un tema político, entonces cuando digo que mi participación es política es decir que ya no es técnica, mi trabajo ya no es desarrollar programas libres. No, la lucha misma por la libertad de los usuarios, porque las empresas como Microsoft ya intentan prohibir el Software Libre, por lo tanto, ya no basta solo desarrollar Software Libre, tenemos que luchar por el derecho de distribuir Software Libre

RSL: Richard, yo estuve pensando sobre la dimensión política de como impedir políticamente y además tu tienes un nombre, en todo el mundo eres muy conocido, muy respetado.

RMS: No tanto, no tan conocido...

RSL: Pero en los círculos de poder.

RMS: No como una estrella de rock...

RSL: No, no, pero los políticos y los que tienen cargos importantes si te conocen.

RMS: No, no.. la mayoría no

RSL: Pero si te respetan mucho



RMS: Quizá, en algunos países, pero en la mayoría no.

RSL: Y lo que tú dices es importante para ellos.

RMS: Será para algunos, pero el gobierno de España aun no ha aceptado migrar al Software Libre, obviamente no me respetan tanto.

RSL: Lo que quiero decir es ¿cuál es tu labor?, o sea, conocemos tu labor con la gente, hablando directamente, eso si lo conocemos bien porque lo estamos viviendo, te gusta dirigirte directamente al público y que tu mensaje.

RMS: Pero esa pregunta, high politics.

RSL: High politics.

RMS: Pues quien quiere hablar conmigo lo hago

RSL: Pero tu no vas.

RMS: No puedo, no puedo

RSL: ¿No tocas las puertas?

RMS: No, sería inútil, ellos tienen guardias, tienen sus secretarios, es muy difícil concertar una reunión si no tienen el derecho, la reunión sería inútil, entonces lo hago cuando ellos vienen a mi

RSL: ¿Has hablado con muchos gobiernos?

RMS: No muchos, no muchos



RevistaSL en la conferencia de Richard Stallman

RSL: ¿Con cuáles?

RMS: Con dos o tres Presidentes, el Expresidente de India, el Presidente actual de Ecuador y no hablé con Chávez sobre Software Libre, pero lo apoya

RSL: ¿Sabes que van a sacar el ordenador Bolivariano?

RMS: No

RSL: Se supone que Hugo Chávez ha dependido, y quiere detener

la dependencia total, tecnológica sobre Estados Unidos

RMS: Exacto, que es un peligro

RSL: Si, por ello van a migrar a Software Libre, van a sacar un ordenador y bueno... pero bueno luego dicen, si todos los ordenadores.

RMS: No oigo...

RSL: Que luego dicen, que los ordenadores que van a vender, los ordenadores bolivarianos tienen un sistema de Intel, por lo tanto la dependencia tecnológica sobre Estados Unidos no se va a dar.

RMS: Pero hay solo dos opciones, hay Intel y hay AMD, es difícil una independencia, pero esto no ridiculiza el intentar una independencia en otros niveles

RSL: Quizá los más importantes.

RMS: De todos modos he hablado con oficiales de otros países, de Brasil, España, de India, no recuerdo más...

RSL: En el estado español ¿con quién?

RMS: No recuerdo, pero hubo una reunión de la agencia tributaria que me invitaron a dar la conferencia sobre el Software Libre y no sé que haya pasado, pero estos detalles no tienen mucha importancia, no cambian mucho

RSL: La pregunta creo que iba más a si no es más accesible para expandir el Software Libre dirigirse a un representante de gobierno que a un usuario.

RMS: Cuando quiera escucharme lo hago, pero exigir que me escuchen no sirve y tienen mucha advertencia. Si no quieren conocerme no harán una reunión conmigo, no es útil, los ciudadanos del país si ellos no pueden convencerlo a reunirse conmigo claro que no será útil molestarlo, no debería

RSL: Mucha gente está peleando para que se genere ese interés en el Software Libre, para que los políticos se interesen y quizá te parezca una tontería pero que a tí como máximo representante de este movimiento...

RMS: Para ellos no tengo ninguna importancia

RSL: Pero si te dieran el premio Nobel, eso cambiaría mucho las cosas.

RMS: Quizás pero...

RSL: Eso cambiaría mucho.

RMS: No tiene sentido hacer claves sobre esta posibilidad, no tiene sentido hacer claves basados en sucesos improbables...

RSL: ¿Crees que es improbable?

RMS: Si

RSL: ¿Y a qué aspiras entonces en todo esto?

RMS: Mi aspiración es la liberación del ciberespacio, es una ambición muy grande y no sé si llegaré a lograrlo, pero si intento, hago lo máximo. No intento por caminos que me parecen inútiles como tener una reunión con un oficial que no parezca interesarse. ¿Por qué hacerlo?, parece un desperdicio de tiempo, si ustedes consiguen una reunión con un oficial mexicano lo haré porque esto demuestra que me escuchará.

RSL: ¿Has pensado alguna vez que alguien tiene que tomar tu relevo?

RMS: Si, pero aún no he encontrado, sería muy bueno que otros, pero necesita años de trabajo en el movimiento buscando qué funciona y qué no, dirigiéndose a las cuestiones de hasta qué punto uno puede hacer un compromiso, donde está la línea que no debe cruzar nunca, es difícil, por ejemplo: Hago compromisos cada día, pero hay compromisos que no hago por que negarían las metas éticas del movimiento, no puedo hacer un compromiso que niegue las metas políticas del movimiento.

RSL: ¿Cómo por ejemplo?

RMS: Promover una combinación de programas que incluya programas privativos por ser mejor. Muchos me sugieren promover una distribución del sistema GNU/Linux por su capacidad de atraer más gente mientras contiene programas privativos

RSL: Como Ubuntu

RMS: Exacto

RSL: Y lo que paso con FLISOL.

RMS: Exacto

RSL: Pero el relevo no sólo es difícil porque es mantener la coherencia con los principios del movimiento, sino también físicamente por que hoy estás en México y mañana te vas a Brasil.

RMS: No es tan difícil, no tengo que andar..

(Risas)

RSL: No, pero la gente siempre sabe que andas alrededor del mundo.

RMS: Si, pero no lo hacía hace 15 años

RSL: ¿Cuántas conferencias has dado en total?

RMS: No sé, no he contado todo

RSL: Pero más o menos, ¿500?

RMS: Más, no me sorprendería, pero no puedo saber

RSL: Vaya, ¿en cuántos países has estado en este mes?

RMS: He estado en Colombia, Costa Rica y Japón, en este mes estaré en México, Brasil y Ecuador.

RSL: ¿Hay algún país o estado al que no has ido pero te gustaría ir?

RMS: ¿Por razones políticas o por el movimiento?

RSL: No, por el movimiento

RMS: No, porque voy a donde me invitan, porque cuando me invitan, es decir que hay interés. Porque tendría que ganar de promover el Software Libre en ese país si no hay condiciones locales, tendría un deseo especial de ayudar a ese país

RSL: ¿Y cuál ha sido el más receptivo hasta ahora con el Software Libre?

RMS: No hay uno más receptivo, pero podría decir que en América Latina los países más receptivos son Venezuela, Colombia, Ecuador, Brasil y Argentina.

RSL: ¿Cuba no?

RMS: Cuba no mucho, pero en Cuba Microsoft y Apple no pueden demandar a nadie, entonces uno puede libremente distribuir copias de Windows. Es decir que ahí tienen la libertad 2 y la libertad 0, pero no tienen las libertades 1 y 3, entonces no es Software Libre y es un peligro para la seguridad nacional que use Windows, y los ministros lo han reconocido, pero los informáticos no, no reconocen el problema

RSL: Son ilegales, por distribuir copias...

RMS: No, en Cuba no son ilegales, nadie puede prohibir distribuir esas copias

RSL: Pero yo tenía entendido que en Cuba es en donde más se ha desarrollado el Software Libre.

RMS: No pienso, no es lo que vi, lo que vi es que en la Universidad de la Habana hice una conferencia y a los alumnos les dio mucho entusiasmo por el Software Libre pero a los profesores no, ellos despreciaron el Software Libre y el entusiasmo de los alumnos. Para ellos seguir usando Windows les es fácil

RSL: ¿Los expertos en marketing de Microsoft no usan eso contra el movimiento del Software Libre?, Que quienes lo apoyan son Hugo Chávez, Correa...

RMS: Pudiera ser, no voy a negar su apoyo por miedo de ser criticado por nuestros enemigos. A Microsoft le gusta llamarnos comunistas y en el movimiento hay algunos comunistas, pero también hay empresarios. El movimiento de Software Libre no es comunista, por ejemplo el Software Libre lleva un mercado libre para todos aquellos que den soporte y servicio mientras el soporte al software privativo es un monopolio porque el desarrollador, el dueño es quien tiene el código fuente, únicamente es él quien puede hacer cambios, entonces ¿Quién apoya el mercado libre?, nosotros, ellos no, y también ¿Quién respeta la propiedad privada? no ellos, porque cada copia les pertenece, Microsoft dice que cada copia de Windows le pertenece, entonces no respeta la propiedad privada de los usuarios

RSL: Y el gran enigma, entonces ¿Richard Stallman es liberal en ese sentido o es comunista?

RMS: Soy liberal en el sentido de los estados libres.

RSL: O sea que te defines en esa tradición, no en la de los estados comunistas.

RMS: No, ni anarquista. Tengo alguna simpatía al anarquismo pero no soy anarquista porque apoyo el estado del bienestar.

RSL: Pero acabas de mencionar "nuestros enemigos", los "enemigos del Software Libre"

RMS: Si

RSL: ¿Quiénes son los enemigos del Software Libre?

RMS: Microsoft, Apple, Adobe, mas o menos... también las empresas de cine, porque intentan prohibir el uso de Software Libre y la distribución del Software Libre para mirar las películas. Es asunto de la Gestión Digital de Restricciones

RSL: DRM

RMS: Es decir DRM, en inglés, Digital Restrictions Manager o las esposas digitales. Es crear programas para restringir al usuario, para tener la funcionalidad de funcionar y obviamente esos programas son

son privativos porque los desarrolladores no quieren que el usuario pueda cambiar el programa para escaparse de las esposas, entonces saltan leyes de muchos países corruptos como los Estados Unidos para prohibir la distribución de Software Libre que pueda acceder a las obras encriptadas, pero lo hacemos, legalmente o no, porque es la libertad.

RSL: Bueno Richard, muchas gracias por tu tiempo. 

Evitando que inicien tu equipo como root desde el GRUB

Gonzalo Javier González Rodríguez

Cuando usamos un "bootloader", es común que nos cree la opción de iniciar con un kernel en Recovery Mode (Modo de recuperación). Esta opción nos permite cargar nuestro kernel y nos da acceso como root directamente, sin preguntar password. Esto lo podemos evitar borrando las líneas en el archivo de configuración del bootloader(en este caso /boot/grub/menu.lst), donde se haga referencia al Recovery Mode o donde veamos que en la opción kernel tenga el parámetro "single".

```
title          Kernel 2.6.20 (recovery mode)
root           (hd0,0)
kernel         /boot/vmlinuz-2.6.20
root=UUID=94f94fc0-6d1f-4c07-9a46 ro single
initrd        /boot/initrd.img-2.6.20
```

Hasta este punto ya tenemos parcialmente solucionado el problema, ya que el propio grub te puede dar la opción de iniciar una shell especial donde te permita pasarle los parámetros de inicio del kernel, entre esos parámetros el "single" y, por consecuencia, acceso root.

Para tener totalmente protegido el grub es necesario instalarle una contraseña mediante la configuración de la opción "password" en el archivo de configuración. Hay dos formas de pasarle el password:

```
password "contraseña en texto plano"
```

ó

```
password --md5 "contraseña encriptada en md5"
```

Para obtener una contraseña para el grub encriptada en md5 es necesario ejecutar "grub-md5-crypt", este comando nos solicitará una contraseña para cifrar.

Si reiniciamos nuestro equipo y tratamos de modificar alguna opción, en el bootloader nos mostrará una leyenda diciendo "Presiona la tecla p para digitar tu password y modificar el grub".

Códigos, ¿herramienta de estudio o de destrucción?

Alfonso Franco

En una ocasión, mientras navegaba, me dio curiosidad saber que tan difícil era encontrar información sobre virus informáticos, así que decidí hacer la prueba. Al buscar algunas paginas para documentarme, me dí cuenta que algunas de éstas mostraban el código del virus, y me pregunté, ¿Qué tipo de personas utilizará esta información?.

El encontrar este tipo de artículos o líneas de texto en la web, no es del todo sorprendente, sin embargo para mí, es un poco preocupante, pues si llegaran a ser utilizados con fines inadecuados, estos textos son potencialmente peligrosos. Por ejemplo; si cae en manos de alguna persona la cual no conoce mucho del tema, las consecuencias podrían ser fatales, no solo para la persona, sino posiblemente puede ocasionar daños a terceros, pues en caso de que esta información dañe alguna máquina o borre alguna información, entre otras cosas, puede propagarse a través de otros medios y seguir creciendo su infección.

En mi opinión esta información debería de ser reservada para algunas personas que se identifiquen a través de formularios, sin embargo esto es difícil, pues algunas personas podemos caer en la mentira, ya que llenamos datos falsos o simplemente decimos que si a todo, sin que nos importe mucho el contenido de los formularios. Cabe mencionar que en estas páginas se dice que esa información es únicamente con fines educativos, y esperemos que así sea, pues en el último de los casos los únicos que decidimos que hacer somos nosotros.

Aunque algunos estudiosos de la conducta humana opinan que por naturaleza el ser humano tiende a la destrucción y estos códigos son más fáciles de conseguir que chocolate en la tienda de la esquina, no quiere decir que debemos utilizarlos con fines destructivos, mas bien, deben ser empleados para lo que son, simplemente para conocer a lo que nos podríamos enfrentar y para no ignorar lo que debemos combatir. A ti, ¿De qué lado te gustaría estar?.

RevistaSL

El software libre hecho revista



Colaboradores SL:

Alfonso Franco
Iratxe Esnaloa
Ekal Txpartgi
Odin Mojica
Richard Stallman
David Moreno
Jorge Medrano
Josias Galván
María del Pilar Reyes
Gloria Patricia Meneses
Osvaldo López

Eventos SL:

CONSOL 2008

Proyectos SL:

OLPC Colombia
Columna PL
Columna MySQL

~# Guide Lines SL

RevistaSL es una revista creada por la comunidad de Software Libre para la comunidad de Software Libre, por lo que la participación de la comunidad es fundamental.

Es por ello que RevistaSL invita a la comunidad de Software Libre a participar; puedes hacerlo de diferentes formas:

1. Escribe un artículo

La mayor parte del contenido de RevistaSL es aportado por colaboradores que desean compartir sus conocimientos, experiencias, opiniones, etc. Te recomendamos sigas las recomendaciones para que tu artículo quede perfecto y sea publicado de inmediato.

a) Debes liberar tu artículo bajo alguna licencia libre, en caso de no hacerlo adoptará la licencia usada por RevistaSL (Creative Commons 2.5 MX).

b) Existen diferentes tipos de artículos:

*Artículo: Un artículo abarca un tema, software o tecnología en específico. Consta de una breve introducción, desarrollo del tema, recomendaciones, etc. y generalmente viene acompañado de capturas de pantalla, gráficos, imágenes, etc. Consta de una longitud entre 450 y 600 palabras

*Tutorial. Los tutoriales son textos explicativos, acompañados de capturas de pantalla, código e instrucciones detalladas sobre cierto tema. No consta de un límite de palabras debido al detalle que se llega a tener en este tipo de artículos.

*Reseñas: Las reseñas refieren a eventos ocurridos como congresos, festivales de instalación, talleres, ciclos de conferencias, etc. Constan de una longitud de entre 150 a 200 palabras.

c) El artículo debe ser enviado en formato Open Document para su mejor manejo.

d) En caso de requerirse el equipo editorial realizará modificaciones mínimas al artículo, si el equipo editorial considera que se requieren modificaciones mayores se pondrá en contacto con el autor para pedirle su opinión.

2. Patrocina

Si ofreces algún tipo de producto o servicio, RevistaSL es un excelente medio para publicarlo. Ponte en contacto con nosotros para más información.

3. Divulga

Cuéntale a todo mundo del proyecto RevistaSL, anímalos a escribir y aprendernos.

*Dedicado a todos aquellos que han pasado por nuestras páginas, y en especial a mis grandes amigos, el Staff SL, sin ustedes nada de esto existiría...
- Augusto Lozano*