

SL

EL SOFTWARE LIBRE
HECHO REVISTA

www.revista-sl.org

Marzo - Abril 2007





Atribución 2.5 México

Eres libre de:



- copiar, distribuir y comunicar públicamente la obra
- hacer obras derivadas

Bajo las condiciones siguientes:



Atribución. Debes reconocer la autoría de la obra en los términos especificados por el propio autor o licenciente.

- Al reutilizar o distribuir la obra, tiene que dejar bien claro los términos de la licencia de esta obra.
- alguna de estas condiciones puede no aplicarse si se obtiene el permiso del titular de los derechos de autor

Advertencia 

Los derechos derivados de usos legítimos u otras limitaciones reconocidas por ley no se ven afectados por lo anterior.
Esto es un resumen fácilmente legible del [texto legal \(la licencia completa\)](#).



RevistaSL

El software libre hecho revista

~# Editorial SL



El hecho de hacer negocios con Software Libre causa controversia en todos los países del mundo. Muchas personas, instituciones, organismos y empresas, se cuestionan por el hecho de la posibilidad de hacer negocios con Software Libre, la mayoría en posición de escépticos; sin embargo, es muy viable y esta demostrada la posibilidad de éxito en los negocios con Software Libre.

Vivimos en un mundo que tiene cambios constantes, muy marcados en el área de la tecnología, pero no es la única área de cambios, pues estos cambios también son notorios en materias de economía y negocios. Anteriormente se creía fuertemente que para ser una persona exitosa en la vida, el único camino eran los estudios universitarios, pero los tiempos actuales nos muestran casos de éxito diferentes. En la actualidad existen grandes y exitosas empresas creadas por personas que dejaron sus estudios inconclusos, y que sólo necesitaron de esfuerzos, dedicación y una visión diferente de las cosas. Muchas de las personas mas exitosas hoy en día, no siguieron el tradicional modelo de negocios, sino que se adaptaron a un modelo nuevo y un mundo cambiante para lograr el éxito.

El Software Libre sin lugar a duda, presenta un modelo de negocios con muchas ventajas e innovaciones para los gobiernos, los clientes y proveedores, y cualquier otro particular. Existen muchas empresas dedicadas a los negocios con Software Libre, así como instituciones educativas que buscan la formación independiente de tecnología utilizando Software Libre como base fundamental del conocimiento. Los beneficios del Software Libre están siendo utilizados con mayor fuerza día a día, pues la falsa creencia de que por ser libre es gratis ya esta desapareciendo, y las empresas que comienzan a comprender las ventajas del uso de tecnologías basadas en Software Libre, realizan migraciones cada vez con mayor velocidad, y en ocasiones prefieren aplicaciones de Software Libre sobre otro tipo de herramientas.

ICE Héctor Leal Morales
a nombre de Tec. Javier González Rodríguez

Dirección Editorial SL

Tec. Gonzalo J. González Rodríguez

Coordinación Editorial SL

Carlos Augusto Lozano Vargas

Equipo Editorial SL

Lic. Julio Acuña Carrillo

Tec. Victor H. Cordova Madrid

Lic. Sonia Sánchez

Lic. José L. Galicia Sánchez

Lic. Jesus A. Balam Jiménez

Tec. Julio César Sosa Yeladaqui

Diseño & Imagen SL

DG Edgar Guerra

TSU Josue Gutierrez Hernández

Tec. Eyden Barboza Barela

ICE Héctor Leal Morales

Corrección SL

LIA David Arroyo Arias

Lic. Luis F. Peniche Novelo

Lic. Mauro Parra Miranda

Lic. Raquel Hernández Padilla

Caricatura SL

Humberto Morales

Buzón SL

buzon@revista-sl.org

Colaboraciones

colaboraciones@revista-sl.org

Publicidad

publicidad@revista-sl.org

contenido

Software Libre y Negocios



Leo Utskot en entrevista... Pag. 40



VirtualBox... Pag. 28



Proyecto ViveLinux... Pag. 13



Libera tu software... Pag. 5

¿Sabias qué?... Pag. 3
Eventos SL... 42
Buzón SL... 43

Comparativa Windows vs
Linux... Pag. 8

El "Che" y el Software
Libre... Pag. 10
Los 8 del Software Libre...
Pag. 32

Desarrollo LAMP.NET (3/3)...
Pag. 15

BULMAGES... Pag. 20
FacturaLUX... Pag. 22
Ofimática libre... Pag. 24
BitlBee... Pag. 26

.FIC 2007... Pag. 36
CONSOL 2007... Pag. 37

¿Sabías qué?

Datos curiosos de SL y algo más...

Julio César Sosa

julio.cesar@revista-sl.org



1.- ¿Sabías que el IRC fue utilizado durante la guerra del Golfo por ciudadanos kuwaitíes para comunicarse con el resto del mundo?

IRC, para muchos de nosotros el IRC pudiera ser sinónimo de charlas amenas y diversión para otros significa trabajo o compartir información, hace alusión al medio por el cual muchas personas unen su esfuerzo y conocimiento para llevar a buen puerto un determinado proyecto si de software hablamos. Sin embargo no debemos olvidar que el ser humano no solo se comunica por diversión, cuestiones laborales o educativas también lo hacemos en las circunstancias mas críticas de nuestra existencia en infinidad de formas y por los medios menos pensados.

El 16 de enero de 1991 dio comienzo la denominada Guerra del Golfo también llamada "Operación tormenta del desierto" en la que una coalición al mando del ejército estadounidense comenzó los ataques sobre Irak en represalia a la invasión iraquí e injustificada de la nación de Kuwait quienes de manera urgente y desesperada solicitaron la ayuda internacional.

Ante estos sucesos de espanto, desesperación y crisis algunos kuwaitíes decidieron optar por el IRC como el medio por el cual se comunicaban con el mundo exterior informando sobre lo que acontecía tanto en Kuwait como en las acciones llevadas en Irak tocando temas como las posibles movilizaciones aliadas, los frentes protegidos por el régimen iraquí así como también informando prácticamente al momento en que acontecían innumerables bombardeos sobre la ciudad de Bagdad.

También algunos corresponsales de importantes consorcios noticiosos aprovecharon el IRC para realizar reportes de primera mano, difundir información y comentar lo que sucedía tanto con personas de su misma cadena encontrados en Kuwait reportando los acontecimientos así como con civiles y público en general. Entre esas cadenas televisivas podemos enumerar la CNN, CBS, NBC entre otras quienes mantenían corresponsales de tiempo completo.

Aquí tenemos un pequeño extracto de una conversación entre ciudadanos kuwaitíes, gente de otras partes del mundo y corresponsales. Dicha información fue tomada de los registros históricos de <http://www.ibiblio.org/> sitio dedicado a recolectar información relacionada con la ciencia las artes, historia, economía, política, software entre muchas otras

```
//-----  
This is IRC CHANNEL +peace  
( till ca. Jan 17 02:22 )  
( jnweiger@immd4.informatik.uni-erlangen.de )  
IRC Log started Thu Jan 17 01:03  
//-----  
  
<nova:+report> "cnn reporters wont go t o  
bomb shelter"  
<NBC> Quzae: Only if you have IRC II  
<lynx:+report> none.  
<CNN> Security personell have found the 3 CNN  
reporters in the Bahgdad hotel.  
<nova:+report> "the second alert in Saudi-  
Arabia was false"  
<CNN> All but one apparently hit under the  
bed in the adjacent room while  
<Mark:+report> i think we have dead cnn  
reporters (personal opinion)  
<CNN> the one conned the security forces into  
leaving. They confiscated some  
<VOA_:+report> I think that too  
<Mark:+report> i think we have dead cnn  
reporters (personal opinion)  
<CNN> the one conned the security forces into  
leaving. They confiscated some  
<VOA_:+report> I think that too  
<CNN> video tape of the fighting, saying, "It  
will be returned tomorrow."  
<nova:+report> "saudi mil sources: no counter  
attact from Iraq"  
<CNN> The security personell then left, but  
the CNN people have closed their  
<NBC> 18 minutes to Bush  
<Finland-i> any cnn footage from s-arabia  
<CNN> window so are no longer supplying audio  
of the bombing. Finland: No... They have no  
way to get anything but audio back to CNN.  
<Finland-i> targests in kuwait also under  
attack
```

<Mark:+report> military personnel in saudi
aurabi just informed of conflict statted..sigh
<Finland-i> telcom tower hit
<CNN> They have recorded footage,, tho, and
are going to try and get it back for
broadcast.
<Mark:+report> i hope saddam hussien dont
have a son in irc

-----//

Los servidores de IRC utilizados para el envío de información relacionada con el conflicto en el golfo pérsico se encontraban en Finlandia lugar en donde fue creado el IRC y donde era muy popular.

Debido a situaciones como esta entre las que podemos comentar un breve periodo de agitación política y represión de los medios en Rusia donde también el IRC fue utilizado como medio alterno para el intercambio de información. Aunado a lo atractivo que resultaba el charlar y compartir con personas de diferentes países, el IRC comenzó a tomar mayor auge y popularidad a nivel mundial.

<http://www.ibiblio.org/pub/academic/communications/logs/Gulf-War/desert-storm/>

<http://es.wikipedia.org/wiki/IRC>

2.- ¿Sabías que México esta desarrollando misiles de guerra?

Así es, la Marina Armada de México con apoyo del CONACYT está desarrollando prototipos de misiles y directores de tiro con un alcance de 6 km aproximadamente aunado al desarrollo de un sistema de anaveaje y un sistema de vigilancia visual con un alcance aproximado de 20 km.

Todos estos proyectos están siendo desarrollados en los laboratorios del INAOE, "Instituto Nacional de Astrofísica, Óptica y Electrónica" el cual investiga en las áreas de Astrofísica, Ciencias Computacionales, Electrónica y Óptica. Dicho instituto se encuentra en Tonantzintla, Puebla a 15 km de la capital. Incluso ya han realizado pruebas de lanzamiento del cohete sin embargo el misil aun esta en investigación haciendo referencia al poder destructivo del mismo.

El sistema de anaveaje, el anaveaje es el aterrizaje de una aeronave ya sea helicópteros o aviones sobre una embarcación. Dicho sistema permitiría al piloto de la aeronave realizar un anaveaje de una manera más segura y controlada. Esto pudiera dar pie a pensar que en futuro podríamos hablar del desarrollo de portaviones mexicanos pudiera ser una aseveración aventurada sin embargo desarrollar un sistema de tal propósito implica tener el conocimiento y la tecnología para construir embarcaciones de guerra aun mas complejas.

Ante tales hechos es muy común escuchar el cuestionamiento de si somos un país pacífico, ¿por qué estamos desarrollando armamento de guerra? estos artificios bélicos pudieran ser usados para llevar a cabo acciones militares sin embargo también pueden utilizarse en tiempos de aparente paz ya que somos un país que cuenta con un sin número de problemas muestra clara de que aún somos una nación en desarrollo, un ejemplo de ello es el crimen organizado que trafica sustancias prohibidas por mar y aire esos grupos delictivos tienen los recursos y los medios para estar incluso aun mejor armados que el propio ejército entonces dichas armas se usarían de igual manera en el combate al narcotráfico el trafico de armas entre otros. Por otro lado no olvidemos que estamos viviendo en un mundo donde esa paz aparente se encuentra en un estado de fragilidad muy seria, bien dice el dicho latino "Si vis pacem Parabellum", Si quieres la paz prepárate para la guerra, si tanto anhelamos tener paz debemos estar seguros y preparados para defenderla y luchar por ella. También comentando con alguien muy especial le expuse la siguiente situación. ...imagina que alguien invade nuestro hogar aquel te puede hacer daño a ti y a nuestros hijos, el esta armado y yo también..., ¿crees que dudarías en apretar el gatillo? ¿Ustedes que harían? El mismo concepto y sentimiento por defender lo que es nuestro, nuestra paz, nuestra integridad y esa libertad que todo hombre sin acepción adora se aplica también en un plano más grande, en este caso como nación.

Por otro lado podríamos estar admirando los primeros esfuerzos serios de desarrollo aeronáutico que tal vez en un futuro podrían dar cabida a investigaciones de índole aeroespacial. n_n

<http://www.inaoep.mx/>

<http://ccc.inaoep.mx/~labvision/proy.htm>

<http://www.youtube.com/watch?v=p-jkF1WCSMs>

<http://es.wikipedia.org/wiki/Misil>

¿Tienes algo interesante que compartir?,
¿algún dato curioso?...

Envíalo a julio.cesar@revista-sl.org

¿Cómo escribir Software Libre?

Licencias, libertades, filosofías... ¿cómo?

Exal de Jesus García Carrillo

exal@gnu.org



Si somos de las personas que seguimos los ideales del Software Libre, conocemos y estamos de acuerdo con su filosofía quizá en algún momento querríamos que algún trabajo nuestro sea concebido bajo esos principios de libertad, en esta ocasión veremos como hacerlo.

Quizá la forma mas sencilla de hacer que un programa sea libre sea ponerlo bajo el dominio público, es decir, dejar el programa completamente libre, sin ningún tipo de restricción de código en cuanto a su uso, estudio, redistribución, modificación y también bajo ningún tipo de derechos sobre el propio programa, esto quizás, a simple vista pareciera ser lo mejor, nuestro programa sería totalmente libre, pero no, este es realmente un problema, ¿y cómo es esto?, bien, imagínese que una persona podría tomar su programa, hacerle cambios, muchos, pocos o inclusive ninguno y distribuirlo de manera no libre o peor aún privativamente, imagínese, ¡quizá algún día alguien le podría ofrecer su propio programa!.

Como vemos esta alternativa podría acarrear resultados no deseados para nuestro programa, y en sí para la comunidad entera, entonces, piense en poder dejar casi de la misma manera en cuanto a libertad se refiere a nuestro programa, es decir, sin ningún tipo de restricciones en cuanto a uso, estudio, modificación y redistribución del mismo, pero, hacerlo bajo los lineamientos de una licencia que lo proteja ante situaciones como las arriba mencionadas, una licencia en la que se explique claramente que ese programa es libre y que así se mantendrá, una licencia que le diga al desarrollador que su programa es y se mantendrá libre.

Como vemos, es necesario el uso de derechos reservados en nuestro trabajo, ahora bien, hay que tener cuidado a la hora de elegir bajo qué tipo de licencia pondremos nuestro software, por ejemplo, los programadores de software privativo utilizan los derechos reservados que les otorga su licencia para quitarle la libertad a los usuarios, en cambio nosotros utilizaremos estos derechos reservados para garantizar esta libertad.

La Free Software Foundation, a través del proyecto GNU ha escrito la licencia GPL (General Public Licence, o licencia pública general) para esto, la GPL, actualmente en su versión dos, lanzada en

junio de 1991, entre otras cosas establece que la gente tendrá el derecho de usar, estudiar, modificar y redistribuir el código, prohibiendo a toda persona la facultad de quitarle estas libertades redistribuyéndolo bajo cualquier circunstancia ajenas a la libertad, menos aún bajo una licencia privativa.

Ahora bien, nos agrada la idea, es ahora cuando surge la pregunta: ¿cómo puedo proteger mi programa, poniéndolo bajo los lineamientos de la licencia GPL de GNU?

Bien, para hacer esto, usted deberá incluir una copia completa de la licencia, misma que puede encontrar en la página del proyecto GNU en este enlace <http://www.gnu.org/licenses/gpl.txt>, con su programa, adicionalmente, agregar las siguientes líneas a cada uno de los archivos que conformen dicho programa, además de esto cada archivo deberá tener al menos una línea con el copyright y una indicación de dónde se puede encontrar el anuncio completo de licencia, que obviamente tiene que venir incluido con el programa.

Aquí debe ir una línea en donde se muestre el nombre del programa y una idea de qué es lo que hace.

```
Copyright (C) AÑO nombre del autor
```

```
This program is free software; you can
redistribute it and/or modify it under the
terms of the GNU General Public License
as published by the Free Software Foundation;
either version 2 of the License, or any later
version.
```

```
This program is distributed in the hope that
it will be useful, but WITHOUT ANY WARRANTY;
without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the GNU General Public License
for more details.
```

```
You should have received a copy of the GNU
General Public License along with this
program; if not, write to the Free Software
Foundation, Inc., 51 Franklin Street,
```

También es muy importante añadir información de cómo ponerse en contacto con usted vía electrónica y postal.

Si su programa es interactivo, agregue un mensaje como este cuando el software se inicie en el modo interactivo:

```
Programa version 2, Copyright (C) año nombre
del autor
Programa comes with ABSOLUTELY NO WARRANTY;
for details type `show w'. This is free
software, and you are welcome to redistribute
it under certain conditions; type `show c'
for details.
```

En donde los comandos `show w' y `show c' deberán indicar, el primero, que el programa se distribuye sin ningún tipo de garantía y el segundo, las condiciones bajo las cuales se podrá redistribuir el programa, obviamente estos comandos son hipotéticos, usted podrá utilizar los que desee o inclusive que sean elementos de un menú, clicks del ratón, ect., todo depende de su programa.

Como podemos ver, la licencia está en idioma inglés, algunas personas han traducido la licencia GPL a otros idiomas, pero estas traducciones no son oficiales, esto principalmente por el riesgo de error que pudiera existir en dicha traducción y cambiar así el sentido de la licencia.

No obstante, si usted insistiera en que el lenguaje de su licencia estuviera en un idioma que no sea el inglés, usted es libre de incluirla, para lo que podrá hacer lo siguiente:

Incluir la traducción no oficial haciendo mención de que se trata de una traducción no oficial de la licencia, incluyendo un mensaje como éste en la misma:

```
This translation of the GPL is informal, and
not officially approved by the Free Software
Foundation as valid. To be completely sure
of what is permitted, refer to the original
GPL (in English).
```

¿Le parece que se ha hecho una violación a la GPL?

Si usted cree que en algún lado se ha hecho una violación a la GPL, lo primero que debe de hacer es cerciorarse de lo siguiente:

- ¿El programa contiene una copia de la licencia?
- ¿Se menciona claramente qué programa está cubierto bajo la licencia?, ¿dice algo confuso, quizá diciendo que algo está bajo la licencia cuando realmente no lo está?

- ¿Está incluido el código fuente con el programa?
- ¿El código es distribuido solo como binarios?
- ¿Está completo el código fuente, o está diseñado para enlazar con otros módulos no libres?

Si después de esto, todavía cree que se trata de una violación a la GPL deberá tomar nota de los siguiente:

- El nombre exacto del producto.
- El nombre de la persona u organización que lo está distribuyendo.
- Direcciones de correo, direcciones postales y números de teléfono para contactar con el/los distribuidor/es.
- El nombre exacto del paquete cuya licencia se viola.

De qué manera se viola la licencia:

- ¿Está incluida la declaración del copyright del titular.
- ¿Falta todo el código fuente del programa?.
- El código fuente del programa, si existe, ¿solamente da un sitio web o FTP para descargarlo?.
- ¿Está incluida con la distribución del programa una copia de la licencia?.
- ¿Está disponible parte del código fuente, pero no todo?, si es así, ¿qué partes faltan?.

Mientras más detalles pueda reunir al respecto, más fácil le será al titular del copyright seguir el caso.

Una vez que hayamos reunido todos los detalles posibles, se deberá enviar ese informe al titular del copyright sobre los paquetes que están siendo utilizados de manera inapropiada, cabe mencionar que solamente el titular del copyright es el único que puede realizar acciones que hagan cumplir con la licencia.

Si el titular del programa fuera la Free Software Foundation, por favor envíe ese informe a <license-violation@gnu.org>

Se aclara que la GPL, como otras, es una licencia con copyright, esto quiere decir que únicamente los titulares del copyright están facultados para tomar acciones ante situaciones de este tipo. La FSF actúa sobre todas las violaciones que se le notifican sobre código perteneciente a la FSF y ofrece asistencia a otras personas que quieran hacer lo mismo.

Cabe destacar que actualmente se está trabajando en lo que será la tercera versión de la GPL de GNU, usted puede revisar los borradores que se han ido publicando, estos se encuentran en <http://gplv3.fsf.org>, por favor, lea los y comente al respecto allí mismo en la página, para que así sus opiniones puedan ser consideradas para la versión final de la licencia.

CONOCE
PIENSA
SORPRENDETE
TU MEJOR ELECCION

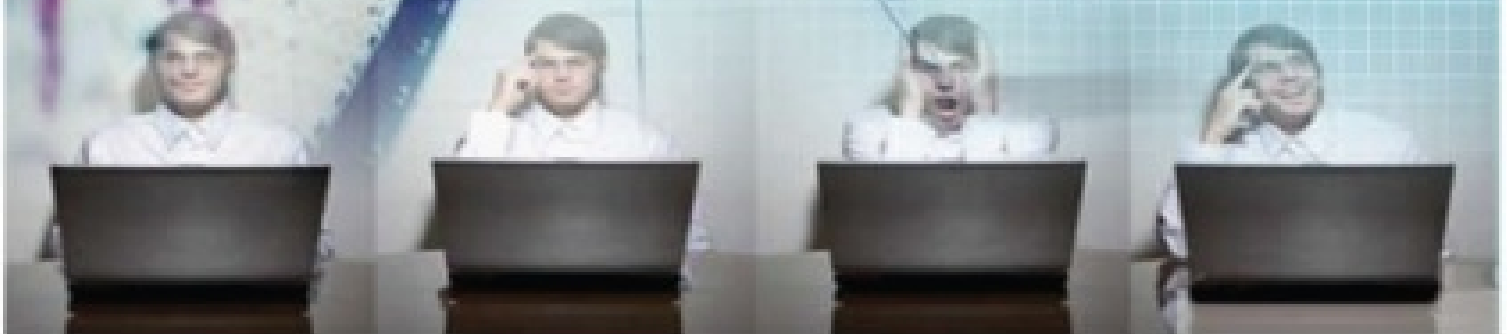
INFO-@

INNOVACIONES INFORMATICAS DEL SURESTE

www.info-e.com.mx

- Servidores dedicados GNU/Linux
- Resellers
- Instalacion de servidores GNU/Linux
- Hospedaje web

LEVANDOLGATECNOLOGIA A OTRO NIVEL



Info-e (FAR Systems) ofrece a los lectores de RevistaSL

100 MB de espacio en disco, 100 cuentas email, BD ilimitadas MySQL/PostgreSQL, cuentas FTP ilimitadas, parqueo de dominios ILIMITADOS, subdominios ilimitados, 10 GB de transferencia mensual, dominio .net/.com/.org, acceso a Fantastico. Todo esto por \$ 900.00 MXN + IVA

Envia un e-mail a ventas@farsystems.com.mx y di que leiste este anuncio en RevistaSL "El Software Libre hecho revista"

¿Linux o Windows Vista?

¿Cuál pondrias en tu empresa?

Ozram Intriago

palaguan@gmail.com



Hoy en día Microsoft está presente en la empresa y el hogar, siendo Windows su mayor producto. A finales de enero del 2007, Windows Vista llega al mercado doméstico para quien pueda aguantar sus características. El 25 de octubre de 2001 se ponía a la venta Windows XP. Casi seis años después llega Vista (Longhorn) con muchas críticas. Una de ellas, es la copia de funciones que escritorios como KDE o Gnome realizaban ya en Linux, u otras más novedosas que realiza Mac OS X de Apple. Incluso el Internet Explorer 7 tiene varias similitudes con el navegador Firefox 2.0

Windows Vista es esencialmente un Windows XP pero con mas colorcitos, efectos en ventanas, consume mucho mas recursos y lo mejor de todo ¡no es muy compatible con el hardware de hace mas de 1 año!. Si estás contento actualmente con Windows XP SP2, no hay ninguna razón para la actualización. Sin embargo, no vale la pena salir a comprarlo a toda prisa, en caso contrario necesitarás comprar un nuevo PC que tenga soporte a Windows Vista, ya que solo podremos disfrutar verdaderamente el Sistema Operativo con máquinas que tengan mucha "galleta" (sus características sean altas en cuanto a un gran rendimiento) que han sido fabricados durante el último año. Comparado con Mac OS X 10.4, Windows Vista se ve más bien tosco y poco intuitivo, casi como si estuviese todavía basado en DOS.

Durante la instalación muchos de los drivers necesarios los descarga directamente desde Internet de forma que si no contamos con una conexión a Internet nuestra instalación se puede quedar incompleta o bien te instale drivers genéricos o incompatibles con tu equipo.

Mas información de Windows Vista visita: <http://www.microsoft.com/latam/windows/products/windowsvista/default.msp>

Linux es un Sistema Operativo tipo Unix diseñado para aprovechar al máximo las capacidades de las computadoras PC basadas en el microprocesador i386 y posteriores. Es un SO con capacidades de multiprocesamiento, multitarea y multiusuario. Sin embargo, a diferencia de otros sistemas Unix para PC, usted no tiene que pagar cuantiosas licencias por el uso de Linux.

Como Sistema Operativo, GNU/Linux es muy eficiente y tiene un excelente diseño. En las plataformas Intel corre en modo protegido; protege la memoria para que un programa no pueda hacer caer al resto del sistema; carga sólo las partes de un programa que se usan; comparte la memoria entre programas aumentando la velocidad y disminuyendo el uso de memoria; usa un sistema de memoria virtual por páginas; utiliza toda la memoria libre para cache; permite usar bibliotecas enlazadas tanto estática como dinámicamente; se distribuye con código fuente; usa hasta 64 consolas virtuales; tiene un sistema de archivos avanzado pero puede usar los de los otros sistemas; y soporta redes tanto en TCP/IP como en otros protocolos.

Linux fue desarrollado desde sus comienzos para trabajar en red. Su protocolo principal es TCP/IP, aunque soporta una gran variedad de protocolos como SLIP/PPP, PLIP, NFS, Telnet, TNP, SMTP, IPX, AppleTalk, etc. Además es capaz de mediar entre todo tipo de redes, permitiendo trabajar en red con equipos que utilicen Sistemas Operativos como Windows 98 o XP sin ningún problema

Es capaz de responder, simultáneamente, a las solicitudes de varios usuarios que empleen el mismo ordenador, incluso con necesidades distintas. Además proporciona los elementos necesarios para garantizar la seguridad y privacidad de los datos entre los diferentes usuarios.

Es robusto, por lo que si un programa falla no interrumpirá el trabajo de los demás. Entraremos al sistema, desbloquearemos el programa y podremos seguir utilizando el sistema sin ningún problema. Esta característica permite que el sistema funcione durante periodos muy largos de tiempo sin necesidad de parar y volver a arrancar.

Actualmente es soportado por los sistemas informáticos independientemente del microprocesador que lleven instalado (386, 486, Pentium, Pentium Pro, Pentium II, Pentium III, Pentium 4, AMD 64, Amiga, , Atari, Alpha, PowerPC, SPARC, RISC, etc...).

Desde mi punto de vista prefiero poner Linux en mi empresa ya que resultaría mucho mas económico

tenerlo, porque no pagaría costosas licencias por el sistema operativo y sobre todo el equipo no se gastaría mucho en él, aunque la verdad la mayoría de la gente esta muy acostumbrada al uso de Windows, pero es mejor invertir en nuestro país y sobre todo en nuestra empresa contratando personal que de servicio y capacitación a los empleados y equipo, que regalando dinero a un gringo que ya es millonario y encima de eso contratar personal para que les den servicio y capacitación a los empleados y equipo.

Aquí les pongo los requerimientos para la instalación de ambos sistemas operativos para que hagan una comparación y vea que computadoras pueden adquirir de acuerdo a su presupuesto.

Requerimientos Minimos

Sistemas Operativos	GNU/Linux (Modo consola)	Windows Vista
Procesador	Pentium 100Mhz	Pentium III 800 Mhz
RAM	24 MB	512 MB
Disco Duro	450 MB	15 GB
Tarjeta de video	4 MB	Tarjeta aceleradora gráfica compatible con DirectX 9 y 64 MB

Requerimientos Recomendadas

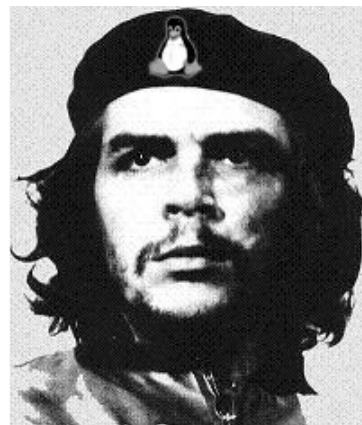
Sistema Operativo	GNU/Linux (Servidor)	GNU/Linux (Escritorio)	Windows Vista
Procesador	Pentium 100 Mhz	Pentium II 300 Mhz	Pentium III 1 Ghz
RAM	64 MB	128 MB	1 GB
Disco Duro	1 GB	4 GB	15 GB
Tarjeta de video	8 MB	4 MB	128 MB

Che Guevara

... Y Software Libre

Oscar Mondragón

oscar.mondragon@correo.de



El `desktop` de Linux está muerto. También el Che Guevara. Pero como veremos en este artículo, ambas no son razones para declarar la muerte de la **Revolución**.

Primero, hagamos un forense de la muerte del `desktop` de Linux.

Después de años de continuos éxitos tecnológicos, las alternativas GUI para usar Linux han alcanzado en su mayoría el mismo nivel de usabilidad y preponderancia de características de otros GUIs en el mundo privativo. Más allá, la disponibilidad de aplicaciones de escritorio para Linux es tan amplia hoy en día que la mayor parte de los usuarios debería fácilmente encontrar la posibilidad de llevar a cabo la totalidad de sus tareas ordinarias de cómputo usando un `desktop` de Linux. Sin embargo, las tasas de adopción del `desktop` de Linux son decepcionantes y muchos CIOs alrededor del mundo son cada día más escépticos acerca de las posibilidades de que **Linux algún día se convierta en una alternativa ampliamente utilizada en el escritorio**.¹

Atrás han quedado los días en que el `desktop` de Linux era tan amigable para el usuario de Windows como un coche sin volante: algo donde aparentemente todas las partes estaban allí, pero aun así era imposible de usar. De hecho, el `look and feel` de Linux hoy en día puede hacerse de tal modo que quizás algunos usuarios de Windows podrían ser convertidos a Linux **sin que siquiera se enteraran**.² Así que la pregunta evidéntísima es, ¿por qué los usuarios no están inundando los servidores de los distintos distros de Linux para obtener una copia gratuita de un súper-capaz sistema operativo?

Así que comencemos con las conclusiones de este estudio forense. Primero que nada, el hecho de que Linux sea igual de bueno que Windows, sencillamente no es suficiente. La abrumadora presencia de Windows no puede ser batida con algo 'igual de bueno'. Un hábito de 16 años es muy difícil de dejar sin una justificación de mayor peso.

Enfrentemos la realidad, algunas personas dejan de fumar porque acaban convenciéndose de los peligros que se ocultan en el cigarrillo luego de obtener información al respecto del cáncer

pulmonar y otras enfermedades provocadas por dicho hábito; pero aun así, sigue habiendo fumadores entre nosotros. Igualmente, usar Windows puede ser un factor de riesgo para que los usuarios de computadoras contraigan úlceras y gastritis, pero la mayor parte de ellos no dejarán de usar su Windows, simplemente porque existe algo igual de bueno (o malo) por allí.

Ya en serio, los dolores que acompañan la experiencia del usuario de cómputo en nuestros días poco tienen que ver con lo que un escritorio Linux podría hacer por ellos. Por ejemplo, con ambos SOs, Windows y Linux, el usuario es cruelmente sometido a rutinas y tareas que no están relacionadas con el trabajo que pretende llevar a cabo usando la computadora, sino con la computadora misma. Los usuarios sufren por todas esas instalaciones, reinstalaciones, actualizaciones, y demás tareas que deben llevar a cabo aun y cuando tan sólo pretendan usar la computadora para escribir sus recetas de cocina y compartirlas con sus amigos. Esta enorme carga que los sistemas operativos actuales imponen al usuario no puede ser aliviada con un sistema operativo que es tan bueno o tan malo como el otro.

Luego, pensemos en lo más obvio. Quizás la mayor plaga que afecta a los usuarios de computadoras hoy en día tiene que ver con cosas como spam y phishing, que los `desktops` de Linux ayudan a resolver tanto como lo hace Windows: ambos fallan por completo al usuario en este sentido. Concedido, virus, adware y otro tipo de malware puede no ser tan común en el mundo de Linux, pero un disco duro fallido o ser víctima del robo de la laptop, puede arruinarnos más de un día en la vida... y ninguno de estos asuntos es resuelto satisfactoriamente por ninguno de los SO de escritorio actuales.

Si en los aspectos en los que Linux compite exitosamente contra Windows no se encuentran muchas razones para que los usuarios migren, la asimetría entre ambos contrincantes se hace abismal cuando factoramos otras cuestiones. Por ejemplo, los usuarios especializados, desde diseñadores gráficos hasta sociólogos, no encontrarán su Photoshop o su SPSS disponible para Linux; y las oportunidades de que ese amigo que al día de hoy les ayuda con sus problemas de

computadora deje de proveerles de consejos si se mudan a Linux, son muy grandes.

Ante estos retos infranqueables parece lógico declarar la muerte de Linux en el desktop para emprender la retirada hacia el campo de los servidores y las aplicaciones de línea de comando, donde Linux es fuerte y sigue creciendo. Pero esperemos un poco, hay esperanzas. La guerra asimétrica puede ser ganada y la **Revolución** puede aun cristalizarse si se sigue el consejo de los expertos.

Ernesto 'Che' Guevara fue un revolucionario marxista que tuvo una sola victoria militar relevante durante su vida y muchas derrotas. Fue un ministro gubernamental mediocre, un policía cruel, un verdugo despiadado y, para muchos, un idealista impráctico e insaciable. Pero escribió un manual de guerrillas que llegó a convertirse en la biblia de los movimientos de guerra asimétrica (guerrilla) alrededor del mundo, algunos de los cuales resultaron sumamente exitosos en conseguir sus objetivos, como los Sandinistas en Nicaragua.

Si Linux quiere pelear esta guerra asimétrica contra Windows y ganarla en el desktop, donde el rival parece invencible, sus proponentes podrían aprender mucho de los principios de guerra de guerrillas del Che Guevara:

Comienza el movimiento desde los sitios menos vigilados por el enemigo

El campo y los sitios rurales montañosos han sido los puntos favoritos para iniciar guerrillas alrededor del planeta. En el mundo de TI el mercado de servidores fue donde Linux y el software libre fueron capaces de capturar espacios de ventaja desde un inicio, cuando Microsoft no prestaba mucha atención a este mercado. El servidor es aun el fuerte de Linux y otro Software Libre, de modo que podemos decir que la primera misión ha sido cumplida exitosamente. De ese modo, el único consejo del Che en este rubro sería: Mantengan sus posiciones.



Captura los almacenes de armas y parque de enemigo

En términos de TI esto debería interpretarse como una orden para no intentar destruir Windows y sus aplicaciones -como suelen proponer quienes viven en el Software Libre. Más que ello, el software propietario debe ser usado en favor del Software Libre. Sólo hay que pensar que si cada computadora que no puede correr aplicaciones de Gnome tiene un Explorer instalado, es tiempo entonces de dejar de insistir al usuario que adopte el Gnome y enviar al Explorer las aplicaciones desde un servidor de Linux.

Adopta una estrategia de golpes rápidos en cualquier blanco posible, independientemente de su magnitud

Una vez que el mundo del Software Libre esté dispuesto a usar Windows y Explorer como el medio para llegar al usuario, deberá buscar crear todo tipo posible de aplicaciones para con ello ir desplazando al demás software propietario de escritorio que corra en dicha máquina. Para ello es necesario replicar la funcionalidad y el look de las aplicaciones de escritorio, pero al día de hoy tecnologías libres como Ajax o propietarias como Flash/Apollo de Adobe hacen posible presentar al usuario aplicaciones basadas en XML que proveerán al usuario con una experiencia prácticamente indistinguible de las aplicaciones de desktop.

Este rubro demanda la existencia de aplicaciones que superen por años luz a lo que hay hoy. Mientras el mundo propietario comienza a presentar aplicaciones de Webmail, por ejemplo, de altas capacidades gráficas, como el Windows Live Mail, en el mundo del software libre parecemos no poder pasar del Squirrelmail, que era un gran competidor de Hotmail modelo 1996, pero que hoy queda muy corto.

La oportunidad de tener un cliente de correo, de IM, de calendario o hasta un procesador de palabras que corran todos desde servidores de Linux sobre el desktop de Windows y con todas las posibilidades gráficas y de usabilidad que el usuario espera de una aplicación de escritorio, está hoy más viva que nunca y representa el punto nodal de la guerra actual.

Apeala al soporte popular

Si puedes hacer que Linux le quite al usuario la totalidad de las tareas de mantenimiento e instalación de software, le estarás dando, ahora sí, argumentos para desinstalar su Outlook sin que tenga que pasar por el estrés de deshacerse de Windows.

Permitir que los usuarios sigan haciendo sus rutinarias tareas de cómputo con la misma usabilidad a la que están acostumbrados con sus

aplicaciones de escritorio, pero sin tener que llevar a cabo las tareas de mantenimiento de aplicaciones y sin riesgos de perder su información, traerá apoyo popular a Linux y de ahí se originará la victoria final para el software libre.

Windows en el desktop como software puede ser que sea invencible, pero si el desktop mismo se torna en un servicio basado en Linux, el juego cambia diametralmente. Esto se conseguirá cuando cada aplicación de usuario que hasta hoy se ha hecho en Gnome o KDE sea convertida a una aplicación de Web con look and feel de escritorio. Windows como servicio no tiene inercia alguna y presumiblemente ninguna otra ventaja sobre Linux, por tanto puede ser derrotado.

Microsoft ve al SO como un **puente hacia el usuario**³ sobre el que tiene total control. Si Linux puede capturar ese puente a través de aplicaciones de Web con apariencia y usabilidad de escritorio, la dominancia de Microsoft terminará, aun y cuando los usuarios no puedan ser convencidos de dejar su hábito de Windows.

Quizás ciertos usuarios, como quienes usan algún tipo específico de software de CAD o el propio Photoshop, estén aun lejos de llenar todas sus necesidades de cómputo con un desktop basado en Web. Pero la vasta mayoría de desktops **están en una categoría**⁴ donde las necesidades del usuario pueden ser cumplidas fácilmente con aplicaciones basadas en Web. Así como hace 15 años los animadores profesionales y otros usuarios altamente sofisticados necesitaban estaciones de trabajo de Silicon Graphics porque las PCs carecían en ese entonces del poder para cumplir con la necesidades de estos usuarios, del mismo modo es posible que en un futuro las máquinas con Windows estén reservadas para unos cuantos nerds mientras que el resto de nosotros usaremos "simples" estaciones Web con capacidad de almacenamiento ilimitada y miles de aplicaciones a un clic de distancia. En ese mundo futuro, no sería extraño que hasta algo como Knoppix acabara siendo la plataforma de escritorio más común del mundo.

Ultimadamente, sólo cuando el SO local sea irrelevante para el usuario será que la batalla por este espacio será ganada. Aunque en estado embrionario, pero ya **existen algunas tecnologías**⁵ yendo precisamente en la dirección de convertir al Web en una plataforma GUI viable; y algunas de estas tecnologías están soportadas en un backend de LAMP. Quizás las posibilidades del Web como plataforma aun estén inmaduras, pero algunos **CIOs comienzan a darse cuenta**⁶ de que las ventajas de este modelo tanto para administradores como para usuarios pueden ser múltiples y altamente deseables para ambos. Más aun, tienen un alto potencial de ser más simples y eficientes en costos, lo cual puede atraer la atención de otras áreas dentro de las corporaciones y lograr con ello su adopción.

El triunfo de Linux en el desktop quizás nunca ocurra de la misma forma en como Windows ganó así su lugar histórico, logrando que todas las máquinas lo tengan como SO local. Pero el hecho de que las aplicaciones basadas en Software Libre pueden llegar hasta el usuario, independientemente de las preferencias de SO de este, y hacerlo con la usabilidad y características que los usuarios desean, hace posible aun un triunfo total de Linux sobre Windows en el desktop.

Referencias:

- 1) http://news.zdnet.com/2100-9590_22-6134965.html
- 2) http://www.lynucs.org/index.php?screen_id=15327223504024c8bf76b8b&p=screen
- 3) http://news.zdnet.com/2100-3513_22-5175536.html
- 4) http://old.linux-foundation.org/lab_activities/desktop_linux/document.2005-10-07.1731842112/document_view
- 5) <http://franticindustries.com/blog/2006/12/21/big-webs-roundup-10-online-operating-systems-reviewed/>
- 6) http://www.computerworld.com/action/article.do?command=viewArticleBasic&articleId=9005495&source=NLT_PM&nid=8

La imagen del Che Guevara tiene copyright por Alberto Korda. El dueño del copyright permite el uso de esta imagen para cualquier propósito que se use para propagar la memoria de Ernesto 'Che' Guevara. La imagen ha sido editada por al autor de este artículo.

Oscar Mondragón es CIO, Chairman y co fundador de Sapotek Inc., una compañía basada en Nueva York que provee servicios a través de su propio framework de desktop basado en el Web. Este puede ser usado gratuitamente a través de <http://desktoptwo.com> en inglés y <http://computadora.de> en español.

Proyecto Vive Linux

Divulgación de Software Libre en Morelos, México

GRUSLIC

gruslic@googlegroups.com



Vive linux!

La historia.

Si bien es cierto que en este momento el mundo del Software Libre está en una etapa en la que el desarrollo de herramientas con uso del mismo tenga una principal importancia, también es cierto que difundirlo nunca será un punto y aparte.

Este proyecto tiene como finalidad principal promover el uso de GNU/Linux y del Software Libre.

Dicha difusión está orientada a las instituciones de nivel medio superior y superior, por ahora, del estado de Morelos. Aunque no solamente a los alumnos se dirige este proyecto, también a los profesores y todas aquellas personas involucradas con el mundo de las Tecnologías de Información.

El surgimiento.

El interés de una de las docentes del plantel CECYT perteneciente al municipio de Yecapixtla en el estado de Morelos, por enseñar a sus alumnos una opción diferente al Software Proprietario, vía uno de los miembros del **Grupo de Usuarios de Software Libre de Cautla (GRUSLIC)** se realizó el contacto para realizar una sesión de pláticas y cumplir con el objetivo de la Profesora.

El interés mostrado por una persona que aún no estaba involucrada en el mundo del Software Libre hizo que la convocatoria para realizar una sesión de difusión en este plantel no se hizo esperar dentro del grupo.

Los miembros de GRUSLIC organizaron lo que sería después el proyecto: ViveLinux.

Como primer paso, se ha redactado un manifiesto, documento que ha servido para establecer un poco los lineamientos que puede seguir el proyecto y tener una mejor organización y administración del mismo.

Todos los miembros de GRUSLIC han participado en al menos una de las sesiones, por lo que, la actividad dentro del grupo se fomenta y hace que crezca no solamente en interesados en colaborar, sino en la transmisión de los conocimientos de cada

uno de los miembros al público interesado.

La temática.

ViveLinux tiene diversidad de temas, todos establecidos por los miembros de GRUSLIC y basándose en la primera experiencia del plantel Yecapixtla y pensando en los intereses que el público al que está dirigido pudiese tener, se consideran las siguientes áreas:

- Introducción al Software Libre.
- Filosofía de la comunidad de Software Libre.
- Desarrollo Web.
- Lenguajes de programación.
- Redes.
- Diseño Gráfico.



Cada uno de estos temas puede tomar alguna de las siguientes modalidades:

- Pláticas de introducción.
- Tutoriales.
- Talleres.
- Sesiones de Instalación, mejor conocido como InstallFest.
- Integración a proyectos de Software Libre.

No solamente se ha enfocado este proyecto a las personas que se van iniciando en el movimiento, sino también a las personas que tienen un conocimiento previo y a todos aquellos que

cuentan con un conocimiento más sólido, de esta manera los niveles a los que pueden impartirse las pláticas o talleres son:

Nivel básico: Abarca todos aquellos temas introductorios al movimiento y uso del Software Libre.

Nivel medio: El público dirigido ya debe tener un conocimiento previo, por lo que, se puede hablar libremente de una configuración o el uso de una herramienta.

Nivel avanzado: Aquí, los conocimientos deben ser más sólidos, tratando así temas como seguridad, compilación, aplicación de algún patch, etc.



Casos de éxito.

La primera visita a un CECYT se realizó en septiembre del 2006, a la fecha se han realizado 4 visitas a 4 diferentes planteles de estos mismos Centros Tecnológicos ubicados en distintos municipios de Morelos.

Además, como resultado de una visita del proyecto a uno de los planteles, se realizó tiempo después un InstallFest en uno de los planteles.

1. Vive Linux - CECYT plantel I Yecapixtla, Morelos - septiembre 2006.
2. Vive Linux - CECYT plantel I Tenexcpango, Morelos - diciembre 2006.
3. Vive Linux - CECYT plantel I Emiliano Zapata, Morelos - diciembre 2006.
4. Vive Linux - CECYT plantel I Tlaycac, Morelos - febrero 2007.
5. InstallFest - CECYT plantel I Emiliano Zapata, Morelos - febrero 2007.

¿Dónde encuentro más información?

El wiki del manifiesto está disponible y esperando tu colaboración. Este, lo puedes encontrar en: http://wiki.gruslic.org.mx/Vive_linux

Así mismo, el Grupo de Usuarios de Software Libre de Cuautla, GRUSLIC, espera tu participación y que te unas a nosotros para promover este manifiesto.

<http://www.gruslic.org.mx/>

Todos los comentarios son bien recibidos, la ayuda aún más.

¿Te interesa participar? ¡Contáctanos!

<http://groups.google.com.mx/group/gruslic>.



Conclusiones.

El proyecto ViveLinux no es un proyecto cerrado ni concentrado a un sólo territorio de la República Mexicana, tampoco está enfocado a un sólo nivel de usuarios: los iniciados.

ViveLinux contempla diferentes tipos de temáticas y modalidades, adaptable a cualquiera que aún no esté considerada; por lo que son bien recibidas todas las sugerencias.

Este proyecto nació de la inquietud de una profesora del área de Informática a nivel Bachillerato en un estado de la República Mexicana, empezando en septiembre del 2006, a la fecha con 4 versiones y un InstallFest, en pocos meses el proyecto a recorrido 4 diferentes municipios, regresando a uno dos veces.

Sembrando las bases de la difusión, esperamos regresar y comenzar una etapa a nivel intermedio para que los asistentes comiencen a cosechar lo que en un principio fueron simples pláticas introductorias.

Desarrollo de aplicaciones Web

Desarrollo con LAMP.NET (3/3)

Martin Marquez

xomalli@gmail.com



Configurando las aplicaciones web (web applications) en el servidor Apache

Si bien el servidor xsp incluido con mono si bien es muy útil para probar nuestras aplicaciones cuando llega el momento de ponerlas en producción sin duda hay que recurrir a Apache, y a continuación se muestran los pasos para dar de alta nuestras aplicaciones.

Uno de los primeros pasos es revisar la instalación del paquete mod_mono dentro de apache para eso buscamos debajo del árbol de directorios de apache el archivo mod_mono.conf en mi caso la versión instalada de apache es la siguiente versión

```
dragon@solia:/etc/apache2 - Terminal - Konsole
Sesión Editar Vista Marcadores Preferencias Ayuda
solia:/etc/apache2 # apache2ctl -v
Server version: Apache/2.2.3
Server built:   Nov 25 2006 18:25:39
solia:/etc/apache2 #
```

por lo que mi archivo mod_mono.conf se encuentra localizado en la ruta /etc/apache2 una vez localizado el archivo, se procede a modificar el archivo httpd.conf el cual es el archivo de configuración de apache2 (antes de hacer la modificación a algún archivo importante de configuración es recomendable hacer un respaldo del archivo a modificar) en este caso el archivo httpd.conf queda de la siguiente manera después de agregar nuestra web application.

```
#la ultima linea del archivo httpd.conf
Alias /computo
"/home/dragon/public_html/computo"
AddMonoApplications default
"/computo:/home/dragon/public_html/computo"
<Location /computo>
    SetHandler mono
</Location>
```

Después de agregarla se debe de reiniciar apache para que tome en cuenta los cambios.

Creando el archivo de configuración Web.config

Las aplicaciones web .NET tienen una forma segura, extensible y manejable para guardar la información de la configuración, esto es en un archivo XML se crea en la raíz o cualquier parte del árbol de directorios de la aplicación, este archivo se llama web.config y se utiliza para la configuración personalizada de la aplicación Web.

En la raíz de la carpeta de la aplicación, escribimos el siguiente código y lo guardamos con el nombre web.config.

```
<configuration>
<appSettings>
<add key="connectionString"
value="Server=127.0.0.1;Port=5432;Database=inventario;User ID=postgres;Password=chikome66"/>
</appSettings>
</configuration>
```

de la misma manera podemos agregar cualquier otro parametro que necesitemos tener en la configuración de la aplicación por ejemplo, si queremos tener el valor de una tabla predeterminada agregaremos la siguiente llave con su respectivo valor.

```
<add key="defaultTable"
value="operatingsystems"/>
```

este valor después será llamado dentro del archivo de clase con la siguiente instrucción:

```
defaultTable =
ConfigurationSettings.AppSettings["defaultTable"];
```

Visualizando la información de la base de datos con el control DataGrid

Uno de los controles más comunes usados para visualizar la información de la base de datos en las páginas ASP.NET es el control DataGrid, este control ofrece además de la posibilidad de

Creamos una página .aspx con el siguiente código:

```
<%@ Page language="c#"
CodeBehind="datagrid.aspx.cs"
Inherits="computo.classes.datagrid"%>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01
Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html><body bgcolor="white">
<form id="formal" method="POST"
runat="server">
<table align="center">
<tr>
<td align="center">Relacion de
equipos</td>
</tr>
<tr>
<td>
<asp:DataGrid id="rejilla" runat="server"
BorderColor="#CCCCCC" BorderStyle="None"
BorderWidth="1px"
BackColor="White" CellPadding="3">
<HeaderStyle Font-Bold="True"
ForeColor="White"
BackColor="#006699"></HeaderStyle>
</asp:DataGrid>
</td>
</tr>
</table>
</form>
</body></html>
```

y a continuación el código del archivo fuente `datagrid.aspx.cs` asociado a esta página:

```
using System;
using System.Data;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.HtmlControls;
using System.Configuration;
using Npgsql;

namespace computo.classes {
    public class datagrid :
        System.Web.UI.Page {
        protected
            System.Web.UI.WebControls.DataGrid rejilla;
        protected string connectionString =
            "", defaultTable = "";

        private void Page_Load(object
            sender, System.EventArgs e) {
            try {
                connectionString =
                    ConfigurationSettings.AppSettings["connections
                    tring"];

                defaultTable =
                    ConfigurationSettings.AppSettings["defaultTabl
                    e"];
            }
        }
    }
}
```

```
NpgsqlConnection inventario = new
    NpgsqlConnection(connectionString);

        NpgsqlDataAdapter
            adapter = new NpgsqlDataAdapter("SELECT
            equipment.serialnumber,brands.name,operatingsy
            stems.name,equipment.rammemory,equipment.hardd
            isk,equipment.hostname from
            operatingsystems,brands,equipment WHERE
            brands.id = equipment.brand_id AND
            operatingsystems.id =
            equipment.operatingsystem_id",inventario);

        DataSet dataset = new
            DataSet();

        adapter.Fill(dataset,defaultTable);

        rejilla.DataSource =
            dataset.Tables[defaultTable].DefaultView;

        rejilla.DataBind();

    } catch(NpgsqlException ex){

        Response.Write(ex.ToString());

    }

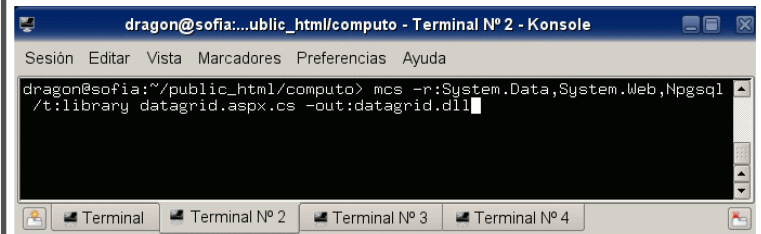
    override protected void
        OnInit(EventArgs e){

        this.Load += new
            System.EventHandler(this.Page_Load);

    }

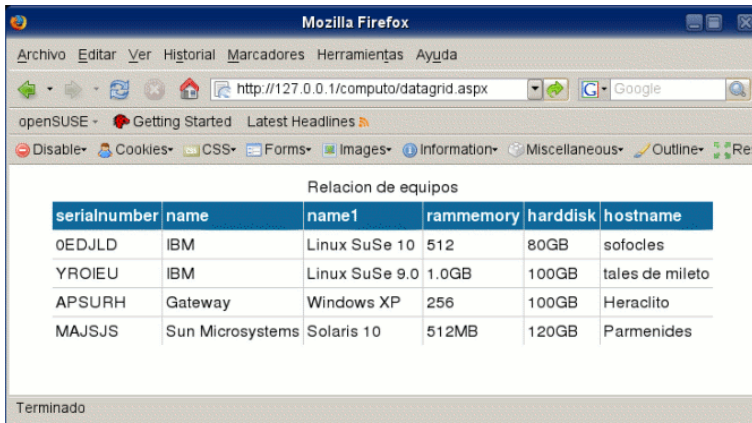
}
```

Lo compilamos de la siguiente manera:



Es importante que antes de visualizar la página en el navegador, reiniciemos el servidor Apache.

Si todo sale sin errores veremos la siguiente página dependiendo de la información en la base de datos.



Atributos configurables del control DataGrid

Los siguientes atributos son los más comunes utilizados por el control datagrid estos atributos deben de colocarse entre la etiqueta asp de apertura del control y la etiqueta de cierre.

Este atributo nos permite especificar las propiedades del encabezado, el color de la letra así como si debe estar en negrita o no.

```
<HeaderStyle Font-Bold="True"
ForeColor="White"
BackColor="#006699"></HeaderStyle>
```

El atributo siguiente especifica el pie de página

```
<FooterStyle ForeColor="Green"
BackColor="Black"></FooterStyle>
```

El atributo siguiente funciona para cambiar las propiedades de los elementos de la tabla

```
<ItemStyle ForeColor="#000000"
BackColor="#FFFFFF"></ItemStyle>
```

Ordenando la información por columnas

Una de las características más prácticas del control DataGrid es la capacidad de poder ordenar los datos con tan solo dar un click al título de cada columna, el siguiente código es una nueva página similar al ejercicio anterior pero con la capacidad de poder ordenar ascendente y descendientemente la información mostrada.

El código datagrid2.aspx

```
<%@ Page language="c#"
CodeBehind="datagrid2.aspx.cs"
Inherits="computo.classes.datagrid2"%>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01
Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html><body bgcolor="white">
<form id="formal" method="POST"
runat="server">
<table align="center">
```

```
<tr>
<TD align="center">Relacion de
equipos</TD>
</tr>
<tr>
<td>
<asp:DataGrid id="rejilla" runat="server"
BorderColor="#CCCCCC" BorderStyle="None"
BorderWidth="1px"
BackColor="White" CellPadding="3"
AllowPaging="False" AllowSorting="True"
OnSortCommand="rejilla_SortCommand">
<HeaderStyle Font-Bold="True"
ForeColor="White"
BackColor="#006699"></HeaderStyle>
</asp:DataGrid>
</td>
</tr>
</table>
</form>
</body></html>
```

su correspondiente código de clase.

```
using System;
using System.Data;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.HtmlControls;
using Npgsql;

namespace computo.classes {
    public class datagrid2 :
System.Web.UI.Page{
        protected
System.Web.UI.WebControls.DataGrid rejilla;
        protected NpgsqlConnection
inventario = null;
        protected NpgsqlDataAdapter
adapter = null;
        protected NpgsqlCommand cmd =
null;
        protected int numberDiv = 0;

        private void Page_Load(object
sender, System.EventArgs e){
            try{
                cmd =
inventario.CreateCommand();
                DataSet dataset
= new DataSet();
                cmd.CommandText
= "SELECT id,name,created_on,updated_on from
operatingsystems";
                adapter.SelectCommand = cmd;
                adapter.Fill(dataset,"operatingsystems");
```

```

rejilla.DataSource =
dataset.Tables["operatingsystems"].DefaultView
;

rejilla.DataBind();

inventario.Close();

} catch(NpgsqlException ex){

Response.Write(ex.ToString());

}

```

```

        override protected void
OnInit(EventArgs e){
            inventario = new
NpgsqlConnection("Server=127.0.0.1;Port=5432;D
atabase=inventario;User
ID=postgres;Password=chikome66");
            adapter = new
NpgsqlDataAdapter("SELECT
id,name,created_on,updated_on from
operatingsystems",inventario);
            this.Load += new
System.EventHandler(this.Page_Load);
        }

```

```

        protected void
rejilla_SortCommand(object source,
DataGridSortCommandEventArgs e)
{

```

```

BindRejilla(e.SortExpression);
}

```

```

        public void BindRejilla(string
orden){
            inventario = new
NpgsqlConnection("Server=127.0.0.1;Port=5432;D
atabase=inventario;User
ID=postgres;Password=chikome66");
            adapter = new
NpgsqlDataAdapter("SELECT
id,name,created_on,updated_on from
operatingsystems",inventario);
            cmd.CommandText = "SELECT
id,name,created_on,updated_on from
operatingsystems";
            adapter.SelectCommand = cmd;
            DataSet ds = new DataSet();

```

```

adapter.Fill(ds,"operatingsystems");
            DataView dv =
ds.Tables["operatingsystems"].DefaultView;

            dv.Sort = orden;
            rejilla.DataSource = dv;
            rejilla.DataBind();
            inventario.Close();
}

```

Tendremos ahora las cabeceras de las columnas habilitadas en el control DataGrid para poder ordenarlas

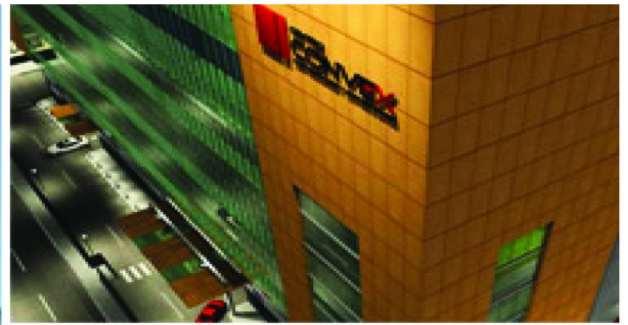
id	name	created_on	updated_on
1	Linux SuSe 10	12/03/2007 12:39:53	12/03/2007 12:39:53
2	Windows XP	12/03/2007 12:39:53	12/03/2007 12:39:53
3	Windows 2000 Profesional	13/03/2007 11:28:11	13/03/2007 11:28:11
4	Windows 2000 Server	13/03/2007 11:28:19	13/03/2007 11:28:19
5	Windows NT 4.0 Server	13/03/2007 11:28:33	13/03/2007 11:28:33
6	Windows NT 4.0 Workstation	13/03/2007 11:28:44	13/03/2007 11:28:44
7	Red Hat Linux 8.0	13/03/2007 11:28:58	13/03/2007 11:28:58
8	Red Hat Linux 9.0	13/03/2007 11:29:02	13/03/2007 11:29:02
9	Linux SuSe 9.0	13/03/2007 11:29:12	13/03/2007 11:29:12
10	Linux SuSe 9.1	13/03/2007 11:29:15	13/03/2007 11:29:15
11	Linux SuSe 9.2	13/03/2007 11:29:17	13/03/2007 11:29:17
12	Linux SuSe 9.3	13/03/2007 11:29:20	13/03/2007 11:29:20
13	Linux SuSe 10.1	13/03/2007 11:29:24	13/03/2007 11:29:24
14	Linux SuSe 10.2	13/03/2007 11:29:26	13/03/2007 11:29:26
15	Mac OS X	13/03/2007 11:29:40	13/03/2007 11:29:40
16	Solaris 10	13/03/2007 11:32:45	13/03/2007 11:32:45

ordenadas por nombre en orden alfabetico.

id	name	created_on	updated_on
1	Linux SuSe 10	03/12/2007 12:39:53	03/12/2007 12:39:53
13	Linux SuSe 10.1	03/13/2007 11:29:24	03/13/2007 11:29:24
14	Linux SuSe 10.2	03/13/2007 11:29:26	03/13/2007 11:29:26
9	Linux SuSe 9.0	03/13/2007 11:29:12	03/13/2007 11:29:12
10	Linux SuSe 9.1	03/13/2007 11:29:15	03/13/2007 11:29:15
11	Linux SuSe 9.2	03/13/2007 11:29:17	03/13/2007 11:29:17
12	Linux SuSe 9.3	03/13/2007 11:29:20	03/13/2007 11:29:20
15	Mac OS X	03/13/2007 11:29:40	03/13/2007 11:29:40
7	Red Hat Linux 8.0	03/13/2007 11:28:58	03/13/2007 11:28:58
8	Red Hat Linux 9.0	03/13/2007 11:29:02	03/13/2007 11:29:02
16	Solaris 10	03/13/2007 11:32:45	03/13/2007 11:32:45
3	Windows 2000 Profesional	03/13/2007 11:28:11	03/13/2007 11:28:11
4	Windows 2000 Server	03/13/2007 11:28:19	03/13/2007 11:28:19
5	Windows NT 4.0 Server	03/13/2007 11:28:33	03/13/2007 11:28:33
6	Windows NT 4.0 Workstation	03/13/2007 11:28:44	03/13/2007 11:28:44
2	Windows XP	03/12/2007 12:39:53	03/12/2007 12:39:53

Conclusión

Son muchas las cosas que pueden hacerse con los controles de .NET, esto ahorra muchas horas de trabajo, C# es un lenguaje totalmente orientado a objetos lo cual permite ser extendido pensando en la productividad. Espero les haya sido de utilidad esta introducción al desarrollo web usando mono.



Monterrey

15, 16 Y 17 Junio

www.xpolinux.com.mx



ARKNUS
SOLUCIONES INFORMATICAS

e-mail
dgranados@xpolinux.com.mx

Teléfono
11572167



¿Necesitas publicidad?

RevistaSL es tu mejor herramienta para hacer llegar tus productos al mundo entero.

Tan solo RevistaSL Num. 3 "Debian, hijos, primos y arrimados" supero las 20, 000 descargas desde países como México, Colombia, España, Argentina, Chile y muchos más...

¿Qué estás esperando?... escribe a publicidad@revista-sl.org o en México llama al (55) 5739 6245 y haz crecer tu negocio.



Al adquirir publicidad ayudas a mantener este proyecto.

BULMAGES

Software de gestión empresarial

Alvaro de Miguel

alvaro.demiguel@gmail.com



La necesidad de un software empresarial libre orientado a la pequeña y mediana empresa es una cuestión que no alberga prácticamente dudas dado el modelo de negocio que sustenta el Software Libre, un modelo basado en el servicio, en práctica contraposición del modelo tradicional de producto manufacturado.

Pero a raíz de una consulta en un correo electrónico que recibí no hace mucho, me hizo reflexionar por qué creo yo y los que apoyamos este proyecto, que BulmaGés es una muy buena opción como paquete de software empresarial entre todos los que se pueden encontrar cuando uno se plantea escoger un software empresarial libre a implantar en una empresa.

No es posible comprender el alcance de BulmaGés sin tener en cuenta el entorno humano que lo hace posible. Este entorno se ha ido definiendo como comunidad a medida que el proyecto ha ido creciendo tomando una identidad visible como Asociación con la fundación de **IGLUES (Iniciativa de Gestión Libre Universal para Empresas y Sociedades)** a mediados del 2004.

IGLUES es una asociación compuesta por técnicos y usuarios motivados por la necesidad de un software de gestión empresarial libre que responda, de una parte a los técnicos que necesitan unas herramientas para dar servicio a sus clientes, y de otra a las pequeñas empresas que necesitan de un software que se adapte a sus necesidades con un coste aceptable. Aunque este planteamiento parezca muy mercantilista, el sentimiento de los que formamos la asociación es que realizamos un esfuerzo en pro del bien común.

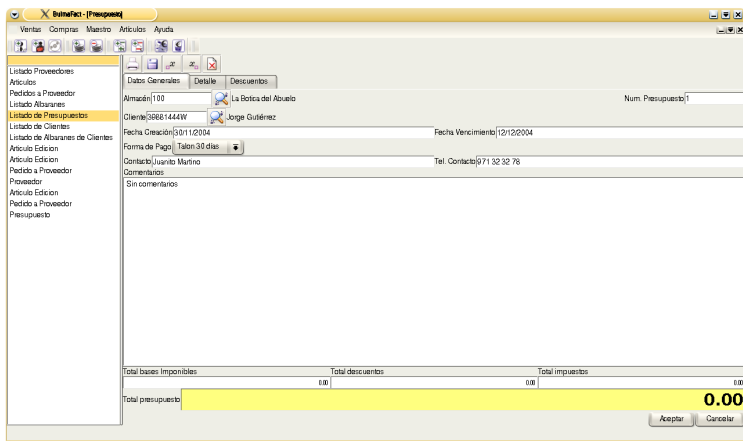
La Asociación IGLUES ha superado la frontera entre técnicos y usuarios compartiendo un proyecto común y ha roto las fronteras territoriales al tener miembros distribuidos a lo largo y ancho del territorio español.

Dicho esto, podemos empezar a ver porqué BulmaGés es una buena opción tanto para empresas informáticas de alcance local e informáticos independientes, como para la Pequeña y Mediana Empresa, necesitada cada día más de un software que de respuesta a sus necesidades específicas.

P	Data	Ordre i Concepte	Contrapartida	Debe	Haber	Saldo	C. Cost	Cancel
1	14/01/2005	37 Moned.	5730001	266.80	0.00	266.80		
2	14/01/2005	39 Pesos	5730001	747.48	0.00	1014.28		
3	14/01/2005	40 FRA: 60884	6000000	0.00	266.80	747.48		
4	14/01/2005	41 FRA: 60885	6000000	0.00	747.48	0.00		
5	15/01/2005	55 Pago Fianza	5730001	262.55	0.00	287.53		
6	15/01/2005	58 FRA: 61100	6000000	0.00	262.53	0.00		
7	25/01/2005	66 FRA: 61311	6000000	0.00	43.22	-43.22		
8	26/01/2005	68 FRA: 61429	6000000	0.00	65.21	-112.43		
9	28/01/2005	78 FRA: 61390	6000000	0.00	3.86	-116.29		
10	08/02/2005	93 FRA: 62182	6000000	0.00	2989.39	-3105.68		
11	10/02/2005	95 FRA: 62288	6000000	0.00	85.11	-3580.79		
12	11/02/2005	100 FRA: 62382	6000000	0.00	977.47	-4658.26		
13	11/02/2005	101 FRA: 62468	6000000	0.00	376.04	-5234.30		
Totals				1296.91	6543.23	-9238.32		

Por qué elegir BulmaGés:

1. Es completamente Software Libre. Todo el desarrollo del proyecto está liberado bajo la licencia GPL y todo lo que ello implica: Se distribuye con su código fuente, que se puede modificar y redistribuir.
2. El desarrollo no es dictado por intereses comerciales de una empresa particular sino por las necesidades de la Comunidad que defiende y apoya el proyecto, reunida en la Asociación IGLUES.
3. El desarrollo modular y su extensibilidad mediante conectores (plugins) dotan a la aplicación de la capacidad de responder a las necesidades de las empresas más dispares con un coste de desarrollo relativamente bajo.
4. Estandarización. Los flujos de datos de entrada y salida de la aplicación utilizan estándares abiertos (XML, HTML, SQL...) lo que además le da una gran facilidad para integrarse en cualquier entorno existente.
5. Innovación. Se realiza un gran esfuerzo en mantener la aplicación en las últimas versiones del software y librerías de las que depende. Prueba de ello es que su desarrollo se realiza sobre la distribución Debian Sid, que la base de datos siempre se actualiza a las últimas versiones de PostgreSQL, y que las bibliotecas gráficas son las últimas liberadas de Qt (actualmente QT4).



6. Continuidad. La aplicación lleva más de 5 años de desarrollo y continuamente se introducen mejoras y se solucionan los errores que se van reportando, a la vez que se añaden nuevas funcionalidades y se plantean nuevos desarrollos asociados al proyecto. Todo ello se puede ver en las listas de correo, el repositorio de subversión (<http://svn.berlios.de>) y en la propia página web (<http://www.iglu.es.org>), donde se pueden observar periódicamente novedades a todos los niveles (desarrollo, empaquetación y documentación entre otros).

7. Comunidad. Quizás el punto más fuerte y el que pasa más inadvertido. Es de los pocos proyectos de software empresarial que ha conseguido reunir a desarrolladores, usuarios y empresas trabajando conjuntamente para que el proyecto se lleve a cabo día tras día.

8. Soporte Institucional. Al ser considerado un proyecto de bien común es susceptible de ser apoyado (y de hecho lo es) por instituciones gubernamentales y entidades de obra social.

9. Es un proyecto abierto. La asociación IGLUES está abierta y anima a todo aquel que desde cualquier perspectiva quiera apoyar el proyecto. El día a día de la asociación y del proyecto se pueden seguir mediante la web, el wiki, la lista de correo y el repositorio.

10. Soporte. A diferentes niveles se da soporte a través de la lista de correo, los informes de errores (<http://www.iglu.es.org/bugzilla>), y poniendo en contacto a empresas e informáticos asociados con usuarios para realizar las adaptaciones a medida que no son el fin de la asociación.

Podríamos continuar dando motivos más técnicos o de otra índole pero van más allá del propósito de este artículo.

Como conclusión, tan solo decir que la vida del software depende de la gente que lo usa y lo desarrolla y que sin una comunidad que lo mantenga, lo utilice y lo realmente, al final no es más que líneas de código que un día quedarán obsoletas.

Más información:

Asociación IGLUES y proyecto BulmaGés:
 Sitio Web: <http://www.iglu.es.org>
 Wiki: <http://www.iglu.es.org/wiki>
 Bugzilla: <http://www.iglu.es.org/bugzilla>
 Lista de Correo:
<http://listes.bulma.net/mailman/listinfo/bulmage>
 Repositorio del Proyecto:
<http://developer.berlios.de/projects/bulmage>
 Que es software Libre:
<http://fsfeurope.org/documents/freesoftware.es.html>
 Licencia GPL:
<http://www.gnu.org/licenses/licenses.es.html>
 Debian: <http://www.debian.org>
 Debian Sid:
<http://www.debian.org/releases/unstable/>
 PostgreSQL: <http://www.postgresql.org/>
 Trolltech y QT4:
<http://www.trolltech.com/products/qt>

FacturaLUX

Software de gestión para la PyME



Carlos Augusto Lozano

carlos.augusto@revista-sl.org

Una de las necesidades principales de una PyME, es contar con soluciones que les permitan administrarse y gestionarse, a este tipo de soluciones se les ha nombrado Enterprise Resource Planning (ERP). Uno de los software ERP mas conocidos, debido a sus bondades en FacturaLUX.

FacturaLUX tiene como objetivo ofrecer una herramienta capaz de desarrollar rápidas soluciones en materia de administración, gestión comercial y finanzas.

FacturaLUX es software impulsado principalmente por una empresa llamada InfoSial y protegido bajo GPL, que puede ser descargado para plataformas Linux, Windows y MacOS X desde su sitio web (<http://www.facturalux.org>)

Instalación

Para instalar FacturaLUX (en GNU/Linux) se debe de descargar la última versión desde su sitio web; a momento de escribir este artículo, la última versión disponible es la 2.1.

Después de descomprimir el archivo, encontraremos un archivo binario, al cual debemos de darle permisos de ejecución mediante el comando:

```
#chmod 777 facturalux-lite-2.1.x86.Linux.bin.run
```

Una vez que el binario tiene permisos de ejecución, podemos iniciar el instalador mediante la línea:

```
#./facturalux-lite-2.1.x86.Linux.bin.run
```

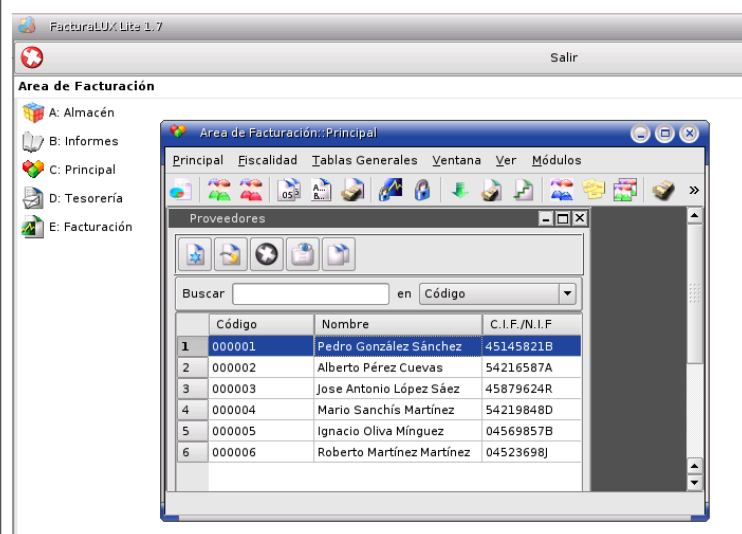
Lo primero que nos mostrará es la licencia GPL, una vez que la aceptemos podemos pasar al proceso de instalación, el cual es muy sencillo.

Aparte del archivo binario de instalación, al descomprimir nos encontraremos con un directorio, este directorio corresponde a los módulos. Por defecto encontraremos los módulos de contabilidad y facturación. En caso de descargar otros módulos es recomendable descomprimirlos en el mismo directorio.

Primeros pasos

Para iniciar FacturaLUX, desde el entorno gráfico debemos de ejecutar el comando `flite`, una vez ejecutado FacturaLUX nos presentara una primer pantalla para realizar la conexión a la base de datos.

Una vez que el sistema haya cargado, debemos de ir a la sección de Administración, donde daremos de alta los módulos que tengamos instalados. Una vez que tengamos los modulos que vienen junto con este versión tendremos todo lo necesario para comenzar a trabajar con el sistema de gestión de clientes, proveedores, impuestos, etc.



Mas información en:

www.facturalux.org
www.infosail.com

flisol 2007

Festival Latinoamericano de Instalación
de Software Libre



28 de Abril
Entrada Libre



Trae tu equipo para que se le
instale Software Legal
Software Libre y GNU/Linux

Universidad Autónoma
Metropolitana
Unidad Iztapalapa

Edificio Anexo-1
"Diego Buitrón Hernández"
Av. San Rafael Arce s/n
Col. Vicentina C.P. 09340
Iztapalapa
UNIDAD IZTAPALAPA

Escuela Superior
de Computación
(ESCOM)

Av. Juan de Dios Bátiz s/n
Unidad Profesional Adolfo López
Mateos
Col. Lindavista C.P. 07738

Facultad de Ingeniería

Circuito Interior, Ciudad Universitaria
C.P. 04510, Coyoacán, México D.F.
Edificio Principal, Ala A, Cuarto Piso
Laboratorio de Especialidades 01
(F01)

CECyT 9
"Juan de Dios
Bátiz Paredes"

Mar Mediterráneo #227, Col. Popotla,
México D.F. C.P. 11400
Entre Mar de Java y Callejón Zanja
Atrás del metro Popotla

CONFERENCIAS

TALLERES DE INSTALACIÓN

Universidad Tecnológica
de Tecámac

Km 37.3 Carretera Federal
México-Pachuca
Predio Tierra Hermosa, C.P. 55740
Tecámac, Estado de México

Universidad Autónoma
del Edo. de Méx.
UAEM
Facultad de Ingeniería

Facultad de Ingeniería
Cerro de Coatepec s/n,
Ciudad Universitaria
C.P. 50100,
Toluca, Edo. de Méx.

Instituto Tecnológico
de Toluca

Av. Instituto Tecnológico s/n,
Ex-Rancho La Virgen
Matamoros, Edo de Mex.

SISTECOM

Calle Cadetes del 47
No. 5112, casa 1,
Héroes 4ª sección,
Ecatepec Estado de México

Vive libre, se libre, usa Software Libre.

ICEnet
TEAM

LIPSO
Laboratorio de Investigación y Desarrollo
de Software Libre

SG
Software Group

COMPUGRAF
www.compugraf.com

COPYLEFT

Tríbit
Tríbit Agilores, no
es libre, es libre, es libre

GNU
GPL
v3



Software Libre Para TI
www.kolivas.com



PELOTON
www.peloton.org



Gul Neza

<http://www.flisolmexico.info> <http://www.installfest.info>

Oficina Libre

Ofimática, control y diversión

Julio M. Acuña Carrillo

julio.acuna@revista-sl.org



Si bien GNU/Linux ha conquistado una porción considerable en el mercado de los servidores, todavía no logra dar el salto definitivo al escritorio y en particular a la oficina. Algunos de los motivos por los cuales la adopción de los sistemas libres ha sido lenta se deben al desconocimiento por parte de los usuarios de aplicaciones libres que remplazan a las que ellos están acostumbrados a utilizar. En los primeros años de vida, GNU/Linux no contaba con software de oficina y si querías utilizar algo como un procesador de palabras o una hoja de cálculo tenías que usar StarOffice que era y sigue siendo software propietario, con el tiempo Sun Microsystems puso a disposición de la comunidad el código fuente y así nació OpenOffice.org que es sin duda un emblema de las suites de oficina libres. Sin embargo existen otras aplicaciones de oficina libres de las que haremos un pequeño recuento a continuación.

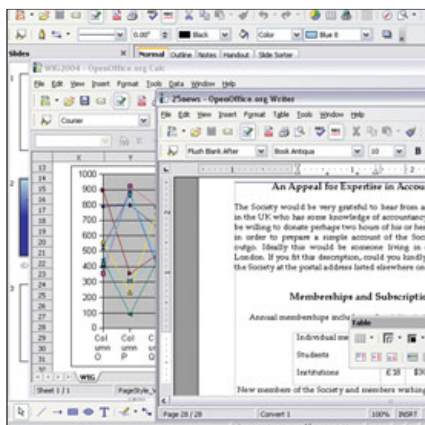
OpenOffice.org

<http://www.openoffice.org>

En 1999 Sun Microsystems adquiere StarOffice, una suite ofimática desarrollada por StarDivision, para reducir gastos y competir con la suite de oficina de Microsoft. Mas tarde ese mismo año, Sun ofrece la versión 5.2 de StarOffice sin costo para uso personal lo que populariza su uso en usuarios GNU/Linux; casi un año mas tarde se libera el código fuente y nace el proyecto OpenOffice.org.

Los componentes de OpenOffice.org son Writer, Calc, Impress, Base, Draw, Math

OpenOffice.org además de correr sobre GNU/Linux corre sobre Windows, Solaris, Mac OS X y FreeBSD.

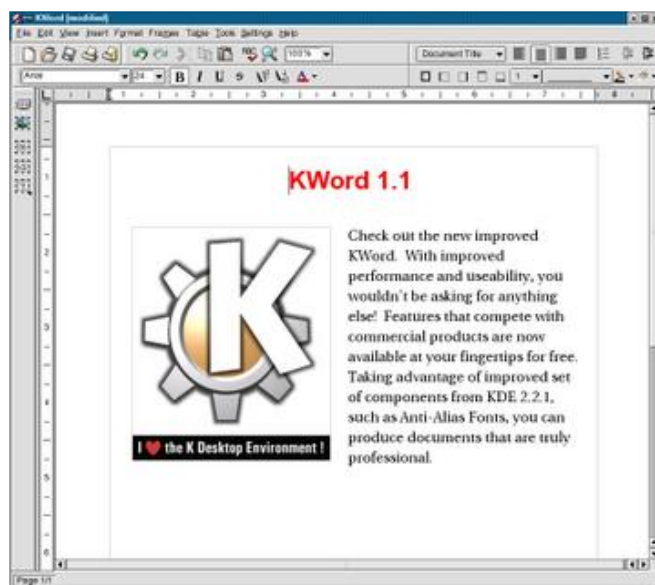


KOffice

<http://www.koffice.org>

Los usuarios del ambiente de escritorio KDE se sentirán muy cómodos con esta suite; integrada completamente con el escritorio, KOffice es una de las suites mas completas que existen en sistemas libres. Inicialmente el desarrollo de esta suite iba muy lento, lo que dio lugar a que los usuarios empezaran a utilizar otras aplicaciones. En la actualidad KOffice cuenta con mas componentes que cualquier otra ofimática y ellos son: KWord, KSpread, KPresenter, Kexi, Kivio, Karbon 14, Krita, KPlato, KChart, KFormula y Kugar.

KOffice corre sobre cualquier plataforma tipo UNIX que soporte KDE.



Siag Office

<http://siag.nu>

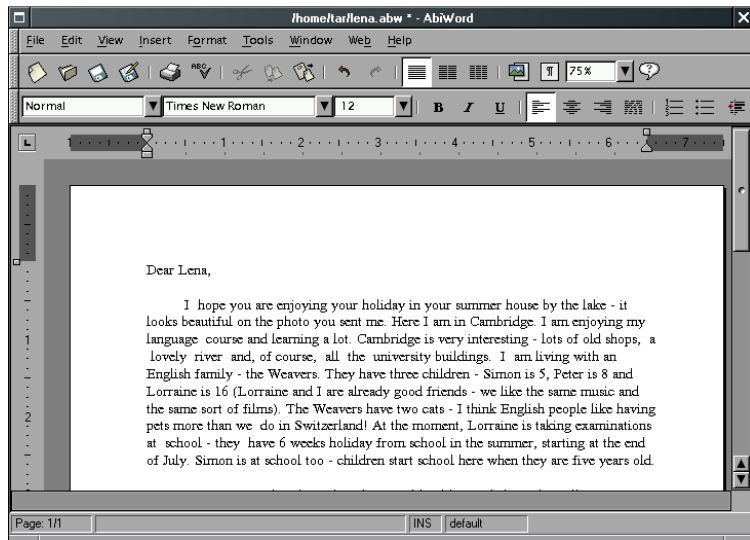
Siag (Scheme In A Grid) inicialmente fue un proyecto personal de Ulric Eriksson para crear una hoja de cálculo libre que corriera sobre las X, actualmente esta es una suite que consta de los componentes principales Siag, Pathetic Writer y Egon Animator, además de Xedplus, Xfiler y Gvu. Por su tamaño es ideal para equipos antiguos o con pocos recursos.

Siag Office corre sobre cualquier sistema operativo con sistema de ventanas X (X Window System).

Abiword

<http://www.abisource.com>

Abiword es un procesador de palabras ligero y funcional que soporta gran cantidad de formatos de texto. Una de sus principales características es el uso de plugins destacando AbiCommand con el cual podemos editar un archivo de texto desde la línea de comandos en un servidor remoto. Abiword corre en GNU/Linux, *BSD, Solaris, AIX, HP/UX, OSF1, Tru64, Mac OS X, Windows, QNX, BeOS y SkyOS.



Gnumeric

<http://www.gnome.org/projects/gnumeric>

Hoja de cálculo libre creada por Miguel de Icaza, tiene una gran cantidad de funciones, incluso mas que hojas de cálculo propietarias, soporta gran variedad de formatos y tiene una muy buena precisión en los resultados de los cálculos lo que la hace ideal para uso científico. Gnumeric está disponible para GNU/Linux y Windows.

GNOME Office

<http://www.gnome.org/gnome-office>

Es el esfuerzo de GNOME para integrar una suite ofimática dentro de su ambiente de escritorio, actualmente está compuesto por Abiword, Gnumeric y GNOME-DB que es una aplicación para manejar datos persistentes como lo son, pero no únicamente, las bases de datos.

GnuCash

<http://www.gnucash.org>

Para mantener al día las finanzas personales o de una pequeña empresa podemos utilizar GnuCash, que nos sirve para manejar nuestra contabilidad, portafolio de inversiones, entre otras cosas que tienen que ver con el manejo de dinero. Tiene

soporte para manejar monedas y lenguajes internacionales. GnuCash puede ser instalado en GNU/Linux, *BSD, Mac OS X y Windows.

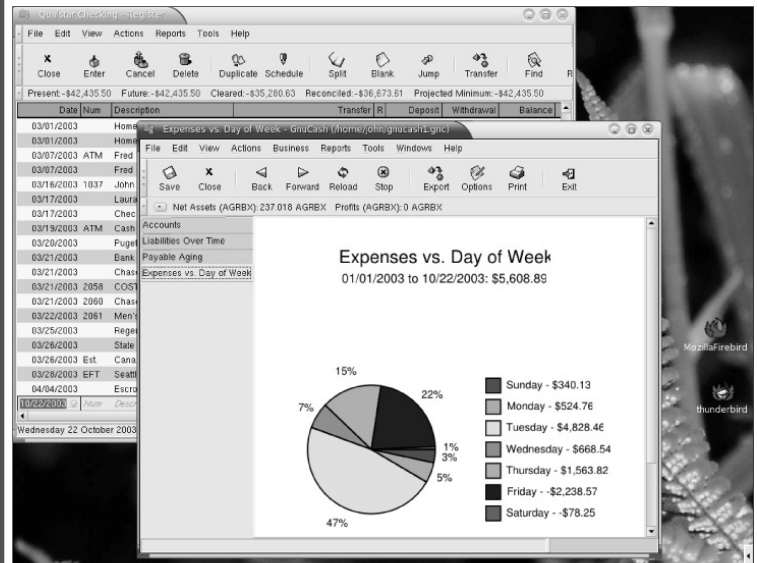


FIGURE 1 GNUCASH IS A PERSONAL FINANCE PROGRAM THAT CAN HANDLE MOST SOLE PROPRIETORSHIP FINANCIAL NEEDS

PySol

<http://www.pysol.org>

Ninguna oficina se escapa de los ratos de ocio que gastan los que laboran allí y para los que, aún teniendo una gran variedad de juegos (y libros!), tienen una extraña obsesión por los juegos de cartas existe PySol.

Como podemos ver existe una solución para cada gusto y para cada necesidad, sin embargo una característica que comparten las ofimáticas libres es el uso de estándares abiertos, por lo que no importando en qué suite elaboremos un documento, otro usuario puede abrir dicho documento con cualquier otra suite. Esto es de gran importancia, sobre todo en algunos gobiernos que requieren que los formatos de los documentos utilicen estándares abiertos y en las empresas que quieren bajar los costos de licenciamiento de sus aplicaciones y que requieran compartir información entre diferentes grupos o divisiones de la empresa. Es por esto que se creó el OpenDocument Format, impulsado por OASIS, un consorcio internacional sin fines de lucro que busca la adopción de estándares e-business.

OpenDocument es un formato XML para documentos que garantiza que dichos documentos puedan ser abiertos y editados en cualquier suite ofimática sin tener que depender del proveedor de la aplicación, si dicho proveedor deja de dar soporte a la aplicación los documentos no se perderán en el olvido.

BitBee

An IRC to other chat networks gateway

Andres Bernardo Vargas

zodman@gmail.com



BitBee es un servidor de IRC (ircd) que sirve de puerta de enlace (gateway) para conectarse a distintos protocolos de chat (msn, yahoo, icq, jabber, etc).

El proyecto surge de los amantes de la consola, que no tenían más remedio que abrir alguna aplicación gráfica (gaim) para poder comunicarse entre los distintos protocolos. Pero como buenos amantes de consola no podían dejar el IRC. Y así nació el proyecto BitBee.

Y ¿cómo funciona ?

Si instalamos el servidor IRC bitBee, y nos conectamos a éste con nuestro cliente IRC favorito (irssi, IrcII, BitchX), entrarás a un canal, donde solo existirá el bot del servidor (root), con el cual interactuarás para conectarte a la cuenta del protocolo que desees.

Cuando agregues las cuentas de mensajería (msn, jabber, icq, yahoo), la lista de contactos se irá desplegando como si te encontraras en un canal de IRC. Y para comunicarte con tus contactos simplemente manda un mensaje (query). BitBee soporta grupos de conversaciones.

Echarlo a Andar...

Primero bajamos el ircd

```
#apt-get install bitBee
```

Corremos en modo demonio

```
#bitBee -D
```

Nos conectamos al server

```
/server localhost
```

```
17:14 -!- zodman [zodman@localhost.localdomain] has joined #bitBee
17:14 -!- ServerMode/#bitBee [+t] by localhost.localdomain
17:14 [Users #bitBee]
17:14 [root] [zodman]
17:14 -!- Irssi: #bitBee: Total of 2 nicks (2 ops, 0 halfops, 0 voices, 0 normal)
17:14 -!- Topic for #bitBee: Welcome to the control channel. Type help for help
information.
17:14 <root> Welcome to the BitBee gateway!
17:14 <root>
17:14 <root> If you've never used BitBee before, please do read the help
information using the help command. Lots of FAQ's are answered there.
17:14 -!- Irssi: Join to #bitBee was synced in 0 secs
[17:15] [zodman] [2:localhost/#bitBee(+st)] [Act: 5]
#bitBee
```

Entramos al canal &#bitBee automáticamente, que es el canal de control, donde estás tú y el bot (root)

Teclamos

help

```
17:28 <zodman> help
17:28 <root> These are the available help subjects:
17:28 <root>
17:28 <root> quickstart - A short introduction into BitBee
17:28 <root> commands - All available commands and settings
17:28 <root> away - About setting away states
17:28 <root> smileys - A summary of some non-standard smileys you might find and
fail to understand
17:28 <root> groupchats - How to work with groupchats on BitBee
17:28 <root>
17:28 <root> You can read more about them with help <subject>
17:28 <root>
17:28 <root> BitBee is written by Wilmer van der Gaast together with Jelmer
Vernooij, Maurits Dijkstra and others. Bugs can be reported at
http://bugs.bitBee.org/.
17:28 <root>
17:28 <root> For other things than bug reports, you can join #BitBee on OFTC
(irc.oftc.net) (OFTC, *not* FreeNode!) and flame us right in the
face. :-)
```

Y saldrá un pequeño manual de uso de los pocos comandos que tienes en el bitBee.

Agregamos cuentas de mensajería msn y gtalk (son los que yo uso).

MSN

```
account add msn <user> <pass>
```

gtalk

```
account add jabber <user@gmail.com> <pass>
talk.google.com:5223:ssl
```

Si no funciona cambia al puerto 5222.

Y habilitamos las cuentas:

```
account on 0
account on 1
```

A continuación se muestran las capturas de pantalla de ambos servicios.

Y también les mostrare capturas en donde se envían mensajes hacia usuarios de otros clientes.

Virtual Box

Una nueva era en virtualización ha nacido

Julio César Sosa

julio.cesar@revista-sl.org



Pues bien, en esta entrega les presentamos lo que algunos creemos ser una de las soluciones estrella de este amplio, prometedor y apasionante abanico llamado Software Libre, "VirtualBox".

Tal vez algunos de nosotros hayamos o estemos utilizando alguna solución de software para emular o virtualizar, entre dichas soluciones encontramos el aclamado VMware, Xen, Qemu, Bochs por mencionar algunos ejemplos (Virtualizar y Emular son, computacionalmente hablando, términos y/o procesos totalmente distintos, aunque sin embargo nos pudieran brindar resultados similares).

VirtualBox, recientemente lanzado el pasado enero del año en curso por la compañía alemana Innotek. Es un virtualizador de propósito general para la arquitectura x86 orientado a servidores y PCs de escritorio, estando para la descarga gratuita tanto en binarios como en código fuente, los binarios o archivos ya compilados están disponibles principalmente para dos plataformas hosts, GNU/Linux y Windows, existiendo para el primero múltiples paquetes de instalación que incluyen los archivos precompilados para su fácil implementación en alguna de las distribuciones disponibles de las cuales podemos mencionar a Ubuntu 6.10 ("Edgy Eft"), Ubuntu 6.06 LTS ("Dapper Drake"), Debian 3.1 ("Sarge"), Debian 4.0 ("Etch"), openSUSE 10.2 updated!, Mandriva Linux 2007.1, Red Hat Enterprise Linux 4, Univention Corporate Server 1.3-2, All distributions, para el último punto "All distributions" tiene asignado un archivo *.bin a modo de instalador genérico, el cual puede ser ejecutado en cualquier distribución moderna, ahora bien esos son algunos de los sistemas operativos en los que podemos instalar VirtualBox, faltan los sistemas operativos huésped los cuales están soportados para virtualizar.

Las versiones del Kernel Linux 2.4 y 2.6, de cualquier distribución son susceptibles de poder ser virtualizadas, hecho que comprobaremos con un ejemplo mas adelante, al igual que FreeBSD (trabaja parcialmente), MS Windows en sus versiones Vista, 2000, XP, 2003, NT y 98, además de DOS y OS/2 (trabaja parcialmente)

En ocasiones nos vemos en la necesidad de usar o probar un determinado sistema operativo, no obstante por practicidad nos gustaría poderlo implementar sobre nuestro Sistema Operativo preferido como anfitrión y acceder a el S.O huésped de una manera sencilla como si de abrir un navegador de Internet estuviéramos hablando, pudiendo acceder e interactuar entre uno y otro con la mayor simplicidad posible. o(n_~)o

¡A instalar se ha dicho!

Para nuestra ejemplo virtualizaremos Damn Small Linux sobre Ubuntu 6.10, usando la versión 1.3.8 de VirtualBox.

Para poder nosotros instalar VirtualBox en nuestro sistema existen varias formas, primero podemos descargar cualquiera de los paquetes de instalación disponibles dependiendo de nuestro S.O.

Para nuestro ejemplo sería el paquete "VirtualBox_1.3.8_Ubuntu_edgy_i386.deb"

Una vez que hayamos descargado nuestro paquete, basta con abrir una terminal e instalar.

```
[Aplicaciones>Accesorios>Terminal]
```

1. Procedemos a instalar haciendo uso de la herramienta dpkg la cual es nuestro administrador de paquetes .deb

```
[sudo dpkg -i nombre_paquete.deb]
```

En nuestro caso:

```
$sudo dpkg -i  
VirtualBox_1.3.8_Ubuntu_edgy_i386.deb
```

2. También en la terminal pero ahora aprovechando los repositorios de aplicaciones de Ubuntu utilizando la Herramienta apt o aptitude según sea el gusto.

```
[sudo aptitude install nombre_paquete]  
[sudo apt-get install nombre_paquete]
```

En nuestro caso:

```
$sudo aptitude install virtualbox  
$sudo apt-get install virtualbox
```

3. ¿Mas fácil?, bueno para nuestra dicha Ubuntu cuenta con un instalador gráfico de paquetes .deb llamado **gdebi** el cual también nos indicará así son necesarias determinadas dependencias, de ser así y si dichas dependencias se encuentran disponibles en los repositorios, las descargará automáticamente.

Hacemos doble clic sobre nuestro archivo .deb nos aparecerá una ventana de información sobre el paquete, nombre, descripción, dependencias etc. ubicamos el botón **Instalar el paquete** y listo.

4. O=(-.-Q) otra opción sería compilar el código fuente, pero sinceramente podríamos hacer otra mini guía de cómo hacerlo, por otra parte lo que nos interesa es hacer uso de las gracias de VirtualBox siendo así y viendo las rápidas y sencillas formas de implementarlo no considero pertinente ahondar en esto. Pero de ser necesario suponiendo que necesitemos una implementación con un enfoque mas a la medida o mas meticuloso en donde tal vez quisiéramos intentar instalarlo en algún S.O. que no aparece en su lista de soporte (o simplemente eres de esos loquitos que compilan la primer línea de código que ven): <http://www.virtualbox.org/wiki/Linux%20build%20instructions>

Nota: VirtualBox es Software Libre protegido bajo la GNU GPL, sin embargo los paquetes precompilados disponibles a la descarga gratuita desde el sitio oficial de Innotek están sujetos a otro tipo de licencias no libres, en donde queda indicado que sólo se permite el uso personal y evasivo de VirtualBox prohibiendo el uso comercial o con mayor grado de especialización requiriendo para esto la compra de una licencia a dicha compañía. Ojo, que diga que permite el uso de manera personal y con motivos de evaluación no quiere decir que los paquetes disponibles sean un Demo, son totalmente estables en versión final y completa.

Todo lo anterior no aplica al código fuente también disponible en el sitio de Innotek y tampoco a los binarios derivados de la compilación del mismo ya que virtualbox está disponible en dos versiones.

The full VirtualBox package: de la cual estamos haciendo uso en esta ocasión.

The VirtualBox Open Source Edition (OSE): ésta es la versión GPL de VirtualBox que es tan funcional como la anterior excepto por algunas características orientadas al mercado empresarial según comentan los mismos creadores.

Una vez terminada la instalación en la terminal nos aparecerá un mensaje similar a este:

```
[Creating group 'vboxusers'. VM users must be member of that group!]
```

Necesitamos agregar algún usuario al grupo vboxusers, (grupo creado automáticamente por el paquete de instalación) dicho usuario o usuarios que agreguemos al grupo "vboxuser" serán los únicos que podrán hacer uso de VirtualBox.

Para realizar esto en la terminal escribimos lo siguiente:

```
[adduser usuario grupo]
```

En donde "usuario" es el nombre usuario de la sesión que están usando en su sistema o el de algún otro usuario a quien le decidan permitir usar el nuevo virtualizador.

En mi caso quedó de la siguiente forma.

```
$adduser xevox vboxusers
```

Voilà, tenemos VirtualBox totalmente instalado. (\n\n)

Aquí algunos comandos útiles no necesarios en esta ocasión pero que no está de más mencionarlos por si acaso se llegara a prescindir de ellos por algún motivo de administración.

Crea un grupo:

```
[groupadd nombre_grupo]
```

Cambia o asigna un determinado fichero o directorio a un grupo:

```
[chgrp grupo archivo.txt]
```

```
[chgrp -R grupo directorio] ejemplo: =>  
[chgrp -R mocosos /home]
```

Cambia o asigna a un determinado usuario la propiedad de un fichero o directorio

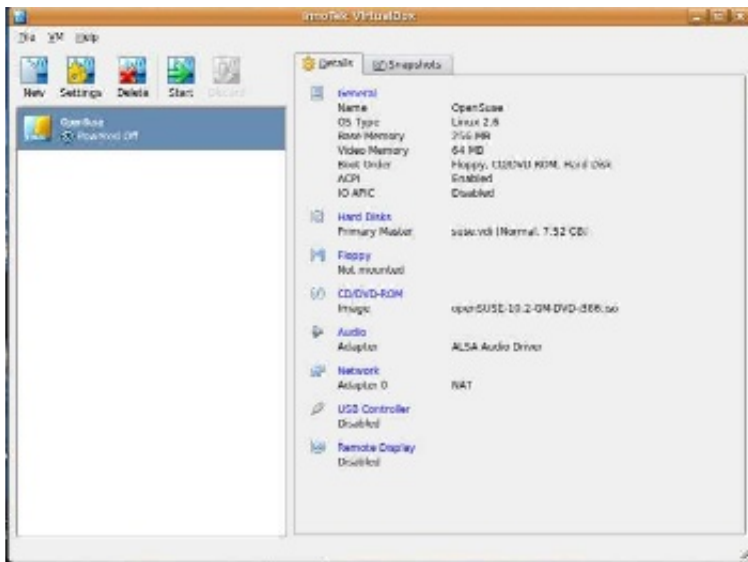
```
[chown usuario archivo.txt]  
[chown -R usuario directorio] ejemplo: =>  
[chown -R xevox /home/xevox]
```

Bien, demasiada teoría, pasemos a un ejemplo mas práctico.

Virtualizar Damn Small Linux

Ejecutamos nuestro VirtualBox:

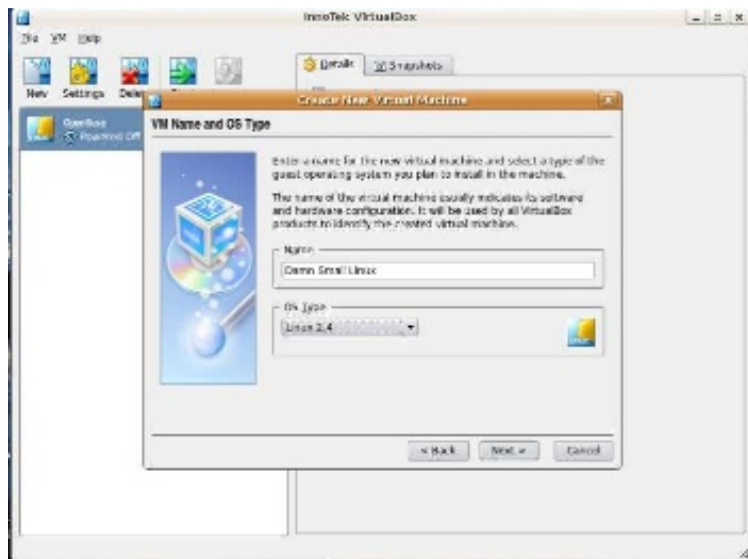
```
[Aplicaciones>Herramientas del sistema>InnoTek VirtualBox]
```



Nombre y sistema

Hacemos clic en el botón **New**, nos aparecerá una ventana de diálogo, le damos **Next >** entonces nos pedirá el nombre para nuestra **VM** o Virtual Machine (en inglés para que suene lIn) así como el tipo de sistema operativo que vamos a virtualizar.

Como podrán observar seleccione Linux 2.4 ya que Damn Small Linux funciona bajo dicha versión del kernel.



Memoria

Ahora nos pide le indiquemos la cantidad de memoria a utilizar para nuestra Máquina Virtual, de facto indica 128MB de RAM pero esto invariablemente depende de la cantidad de RAM de la cual disponga nuestro sistema anfitrión.

Disco duro

Bien, ¿Máquina Virtual?... Disco Duro virtual >:}

En esta sección nos pide seleccionemos un disco duro virtual para alojar la imagen que contendrá nuestra MV, en este caso no disponemos de

ninguno por lo que tendremos que crearlo. para esto hacemos clic en "new" y nos pedirá le indiquemos el tipo de imagen que deseamos utilizar, ya sea una imagen de expansión dinámica o una imagen de tamaño predefinido, para nuestro ejemplo seleccionamos **Dynamically expanding image** (imagen de expansión dinámica) para que vaya aumentando según nuestras necesidades al momento de hacer uso del espacio en disco duro de nuestro sistema virtualizado.



Sólo resta poner le nombre al archivo de nuestra imagen y seleccionar un tamaño inicial para la misma.



Una vez finalizado lo anterior el asistente nos regresa a la sección de selección de disco duro virtual, con nuestro nuevo disco duro seleccionado por supuesto, continuamos con los pasos lógicos clic en **Next >** **Finish** y listo ya tenemos nuestra VM para nuestro Damn Small Linux.

Hey, hey aún no terminamos >=:D

¿Y el Sistema Operativo? ya le movimos aquí, ahí, acullá, pero nos estamos olvidando de lo mas importante, el S.O. que vamos a virtualizar, para esto necesitamos una imagen de dicho S.O.

Para este ejemplo usé la imagen **current.iso** de Damn Small Linux, la cual es una distro Live-CD, para el que no la conozca ésta es una pequeña distribución GNU/Linux de tan sólo 50MB, de otra forma sería la imagen del cd o dvd de instalación del S.O. en cuestión.

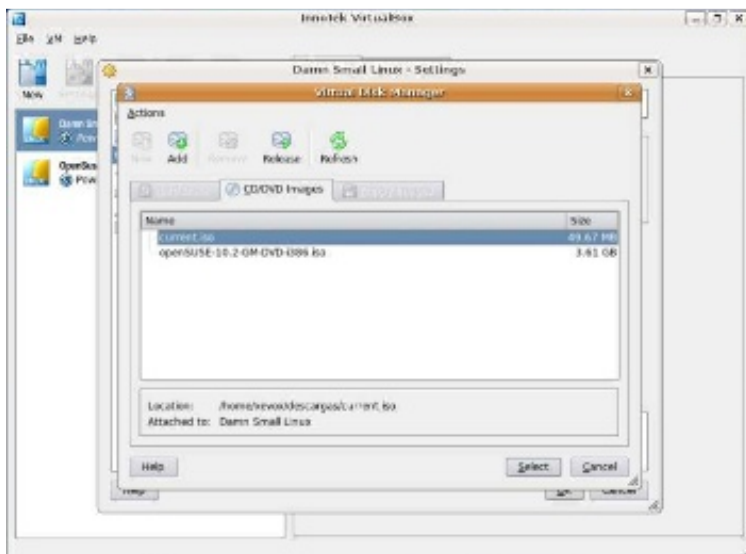
Seleccionamos nuestra máquina virtual **Damn Small Linux** y hacemos clic en el botón **Settings** en el menú lateral izquierdo seleccionamos **CD/DVD-ROM**, marcamos el cuadrado de selección **Mount CD/DVD Drive** para activar las opciones disponibles.

Entre las que se encuentra:

Host CD/DVD Drive: Ésta nos sirve para seleccionar algún dispositivo de CD o DVD con el cual queremos trabajar para acceder al CD/DVD de instalación.

ISO Image File: Para poder hacer uso de una imagen de CD/DVD de nuestro S.O. (como ha de ser en nuestro caso).

Desde luego marcamos esta última opción y presionamos el botón **select**, nos aparece la ventana del **Virtual Disk Manager** en donde podemos ir agregando las imágenes para nuestras distintas máquinas virtuales. Hacemos click en **Add**, ubicamos nuestra imagen *.iso correspondiente > seleccionamos nuestro *.iso; ya una vez que aparezca en la lista > click en **select** >

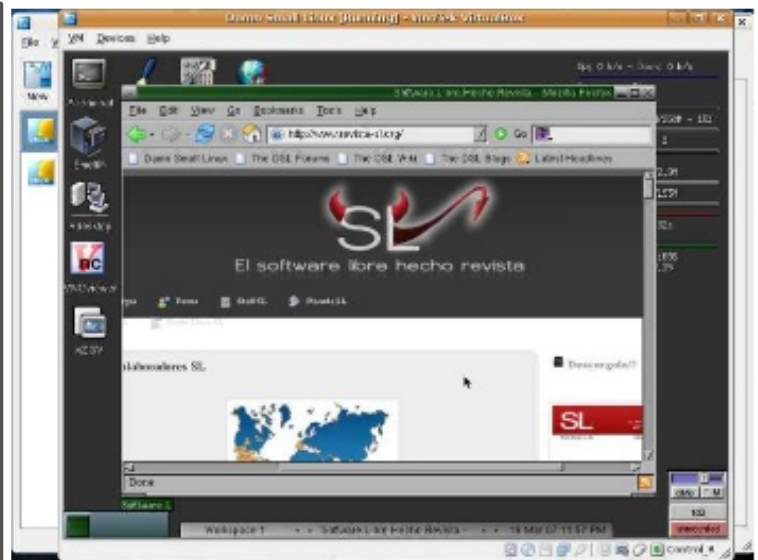


ESTA LISTO (\n_n/)

Respiremos un poco (ja, ja, como si hubiera sido la gran cosa...)

Ahora toca el turno para activar y poner en funcionamiento nuestro S.O. virtualizado

Selecciona la Máquina Virtual y has clic en **START**



Como Tip final para poder abandonar la ventana de nuestro S.O huésped presionen la tecla [Control] del lado izquierdo del teclado.

No sé si se dieron cuenta pero la interfaz y la estructuración de las opciones es prácticamente idéntica a la de VMware

Espero hayan disfrutado de esta reseña-guía sobre VirtualBox, lamento haber sido un tanto obvio en algunas explicaciones, pero si algunos amantes del Software Libre en concreto GNU/Linux, deseamos que se consolide cada día mas en la PC de escritorio, hay que dejar de pensar en complicado, además que fue pensado para que aquel que lo lea sólo tenga dos opciones, “lo entienda” o “compre un revolver y se dé un balazo”.

¡Felices virtualizaciones!

Mas información:

<http://www.virtualbox.org/>
<http://es.wikipedia.org/wiki/Chgrp>
<http://es.wikipedia.org/wiki/Chown>
<http://es.wikipedia.org/wiki/Chmod>

8 beneficios

Del Software Libre en los negocios

Sandino Araico Sánchez

sandino@sandino.net



1. Auditabilidad.

Todo el Software Libre es (por definición) software de código abierto. Al ser software de código abierto el Software Libre es auditable línea por línea. Esta característica resulta muy importante en organizaciones donde la seguridad de sus sistemas de manejo de la información es crítica y no pueden permitir que el software que utilizan tenga puertas traseras que permitan a algún intruso cualquier tipo de intervención. La auditoría de código línea por línea puede incluso descubrir errores o debilidades en los algoritmos que sean explotables por intrusos y corregirlos a tiempo.

La auditabilidad también es muy importante para organizaciones con muy altos controles de calidad en donde todos los procesos de calidad son validados para asegurar que en las mismas circunstancias controladas siempre entreguen los mismos resultados. En este caso la auditoría del código de cada uno de los sistemas que intervienen en estos procesos de calidad sirve como último recurso de verificación dentro de los procesos de validación de los sistemas de calidad, en donde más allá de los resultados de las pruebas de laboratorio efectuadas sería posible encontrar otro tipo de problemas potenciales como bombas de tiempo y errores aritméticos que pudieran (después de algún tiempo en producción) provocar variaciones en los resultados esperados o algún problema mayor.

La auditoría línea por línea del código de un sistema es costosa y generalmente se cobra por hora. Como resultado de la auditoría de código se obtiene una lista de recomendaciones y correcciones necesarias cuya aplicación tiene un costo adicional en horas de programador, este costo suele ser es muy variable. Se debe evaluar el costo máximo estimado de una auditoría en contra de los beneficios esperados y esta decisión debe ser respaldada por los altos mandos de la organización quienes a su vez deben estar conscientes de la variabilidad de los costos.

2. Portabilidad.

El hecho de que el Software Libre sea software de código abierto no necesariamente significa que sea

portable, sin embargo, los propios procesos evolutivos de una gran parte de los proyectos de Software Libre tienden hacia la portabilidad.

El lenguaje ayuda mucho; la mayoría de los lenguajes han sido diseñados para ser muy portables, pero en el software libre se destacan C, PHP, Perl, Python y Java.

Hay una diferencia muy importante entre la portabilidad a nivel de código binario que nos ofrecen las máquinas virtuales de Java y Mono contra la portabilidad de un lenguaje que puede ser compilado para generar código nativo ejecutable en diferentes arquitecturas (incluyendo las máquinas virtuales de Java o Mono). En un punto intermedio entre la portabilidad a nivel de código binario y la portabilidad a nivel de código fuente se encuentran los lenguajes interpretados como PHP, Perl y Python que en sus versiones más modernas cuentan con sus propios compiladores que crean un código binario intermedio y con unidades de ejecución en donde ese código binario intermedio es ejecutado.

En todos los casos de portabilidad es importante verificar las plataformas donde han sido portados los compiladores, intérpretes o máquinas virtuales correspondientes a los lenguajes en que está programado el código fuente de nuestra aplicación ya que el costo del desarrollo de portabilidad de estos componentes a otras plataformas puede ser muy elevado y podría llegar a ser incosteable.

Otro punto importante que hay que considerar en la portabilidad es la dependencia de alguna particularidad del sistema operativo. Las aplicaciones que hacen muchas operaciones de entrada y salida y las aplicaciones que tienen una interfaz gráfica son muy propensas a caer en este tipo de problemas aunque cada vez es más común el uso de librerías portables como GTK, SDL, QT... Nuevamente, el costo del desarrollo de portabilidad de cualquier aplicación a otra plataforma o a otro sistema operativo puede ser muy elevado por lo que es aconsejable realizar en laboratorio las pruebas de portabilidad necesarias antes de la adopción y despliegue en producción de cualquier aplicación de Software Libre.

3. Personalización.

Una de las libertades del Software Libre consiste en modificar el código fuente de la aplicación (y la aplicación por consecuencia) para adoptarlo a nuestras propias necesidades. Es precisamente esta libertad la que ha hecho al Software Libre tan popular y la que ha logrado que la adopción del Software Libre en la industria se masifique.

La otra cara de la personalización del Software Libre es su alto costo y este costo no necesariamente se relaciona con el desarrollo de personalización o de módulos personalizados sino con el costo del mantenimiento de la versión personalizada de la aplicación sobre todo cuando el proyecto principal es muy dinámico y libera versiones nuevas muy frecuentemente.

Hay unos pocos casos (muy maduros) en donde la aplicación soporta el uso de plugins o está preparada para soportarlos, en este caso el costo de la personalización se mantiene bajo mientras ésta se lleve a cabo utilizando plugins y mientras éstos sean compatibles con las versiones disponibles del proyecto principal. En cuanto el proyecto principal libera una versión incompatible se vuelve necesario pagar por un nuevo desarrollo de personalización o al menos la adaptación del desarrollo de personalización anterior al nuevo API incompatible.

Una estrategia que siguen muchas organizaciones se llama "Extend and release"; consiste en liberar la mayor parte posible del código personalizado y al mismo tiempo intentar que esas modificaciones sean integradas al proyecto principal. Inicialmente el costo de esta estrategia es más elevado que el de la clásica estrategia "Embrace and extend" en donde no se libera nada, pero con el paso del tiempo el costo del mantenimiento disminuye (al haber más organizaciones interesadas en las mismas modificaciones) y disminuye más cuando esas modificaciones son tomadas en cuenta para los releases mayores y son integradas como parte del proyecto principal. Cuanto mayor sea la popularidad de estas modificaciones los costos de mantenimiento tenderán a cero.

Es importante hacer una evaluación del costo de mantenimiento contra el posible interés que otras organizaciones puedan tener sobre nuestras modificaciones personalizadas, para con estos elementos poder tomar una decisión sobre el tipo de estrategia a adoptar: "Embrace and extend" o "Extend and release".

4. Rapidez de despliegue.

Al eliminar los trámites de licenciamiento, instalación y activación de componentes y aplicaciones es posible tener un sistema listo para su despliegue y puesta en operación en un solo día. los sistemas operativos libres (especialmente Linux) se han especializado en recopilar la mayor

cantidad de librerías, aplicaciones, frameworks y otras herramientas de desarrollo dentro de sus distribuciones de tal manera que se tengan disponibles al momento de la instalación y despliegue de un sistema la mayor parte de los componentes que vamos a necesitar por lo que el tiempo necesario para la puesta en operación es mínimo. Esto es a su vez una gran debilidad del modelo basado en componentes de software propietario cuyos múltiples proveedores tienen que ser contactados uno por uno y sus esquemas de licenciamiento son muy diversos y seguir todos los procedimientos de adquisición consume mucho tiempo y retrasa la puesta en operación del sistema.

Evidentemente se da el caso en que no todos los componentes que se necesitan para el despliegue de un sistema se encuentran disponibles en Software Libre (o en la distribución elegida). En este caso puede resultar más rápido adquirir componentes propietarios para cubrir esos huecos y ya sea programar una ligera capa de pegamento o hacer ligeras modificaciones a los demás componentes libres para hacerlos interoperar con los componentes propietarios. El costo de este desarrollo generalmente es bajo, pero el costo de adquisición de componentes especializados puede ser muy elevado.

Muchas veces se da el caso de que los términos de licenciamiento de los módulos propietarios son incompatibles con las licencias de los módulos libres, algunas otras veces el costo de algún componente propietario es muchas veces mayor al costo del desarrollo de pegamento entre componentes libres que juntos proporcionen la funcionalidad necesaria. Los costos del desarrollo del pegamento o incluso de módulos completos que hagan falta en un sistema deben tomarse en cuenta junto con el tiempo disponible para la puesta en operación sabiendo que cualquier retraso en el inicio de ésta repercute en pérdidas económicas.

Existen compañías integradoras que se dedican precisamente a recopilar todos los módulos disponibles en software libre para solamente programar el pegamento y tener el sistema operando en tiempo record.

Definitivamente hay un beneficio directo para la compañía integradora que (aunque tiene que pagar por la ingeniería y el desarrollo del pegamento) al utilizar componentes libres elimina costos de licenciamiento de componentes propietarios y al aplicar modificaciones personalizadas sobre los componentes libres puede bajar los costos de desarrollo del pegamento. Evidentemente los tiempos de desarrollo disminuyen.

Las ventajas para una organización que va a contratar a una integradora que usa Software Libre son indirectos pero la experiencia es un factor primordial que influye sobre todo en la velocidad de la integración y el despliegue.

El costo de las pruebas y correcciones necesarias para la puesta en operación del sistema también debe ser tomado en cuenta. En el caso de los componentes propietarios las correcciones necesarias pueden tomar semanas o hasta meses mientras que en los componentes libres estas correcciones se pueden aplicar de inmediato. El costo que se paga por las correcciones inmediatas a los componentes libres suele ser menor al costo por retraso de la puesta en operación del sistema.

5. Independencia de proveedor.

Al estar utilizando sistemas basados en aplicaciones y componentes populares es posible deshacerse del proveedor actual para contratar el soporte, desarrollo o mantenimiento del sistema con un proveedor nuevo. En el caso del software propietario esas posibilidades siguen siendo muy pocas y en algunos casos nos encontramos con la presencia de un distribuidor exclusivo al que vamos a llegar siempre independientemente del camino que tomemos.

El Software Libre suele ir de la mano con el uso de estándares abiertos y para el caso de las aplicaciones y componentes populares que se encuentran en distribuciones populares suelen existir varias compañías (e individuos) que se dedican a proveer soporte, desarrollo o mantenimiento. Resulta relativamente fácil cambiar de un proveedor a otro y el nivel técnico suele ser aceptable.

Incluso los esfuerzos de estandarización y portabilidad de los últimos años han logrado que se pueda cambiar la distribución e incluso el sistema operativo sobre el que corre un sistema con mínimas interrupciones en la operación y esto a su vez abre aún más el abanico de opciones para elegir entre distintos proveedores.

Saltan a la vista un par de desventajas siendo la primera la falta de uniformidad en la capacidad técnica de los proveedores. Los escenarios más elementales son cubiertos por las múltiples certificaciones en donde aquel que no es capaz de cumplir con lo que exige una certificación se va quedando fuera del mercado. Los escenarios de alta especialización son más parecidos a los del software propietario en donde hay pocos proveedores de donde se puede elegir a los más renombrados, a los más baratos o a los más conocidos, quedando siempre espacio para experimentar con algún nuevo proveedor que parezca muy prometedor....

La segunda desventaja ocurre en el caso de aplicaciones o componentes menos populares o de mucha especialización donde son pocas las personas que han trabajado con ellos y los costos pueden aumentar. El mismo aumento de costos ocurre con el software propietario, pero al Software Libre (al ser software de código abierto) nos ofrece la posibilidad de que un nuevo proveedor pueda

llegar sin conocimiento previo a leer las partes más relevantes del código (si no es que su totalidad) para enterarse del funcionamiento del sistema y tomar el control a partir de ese momento (takeover). Existen incluso compañías que se dedican al "takeover express" en donde de un día para otro pueden tomar el control completo del sistema desplazando completamente a la compañía anterior.

El costo del takeover es muy elevado pero en algunos casos es menor al de mantener a un proveedor que está aprovechando su posición monopólica para elevar sus costos.

El caso más común de takeover se da cuando el proveedor anterior desaparece y el sistema necesita seguir en operación. El costo del takeover de un sistema basado en componentes libres es mucho menor al del takeover de un sistema basado en componentes propietarios aunque se han dado casos de takeover en donde se aprovecha el cambio de proveedor para reemplazar algunos componentes propietarios por libres.

6. Mantenibilidad.

En el software propietario existen compañías muy sólidas que mantienen algún componente y es muy remota la posibilidad de que desaparezcan. Lo mismo sucede con proyectos de Software Libre muy grandes donde participan muchos desarrolladores de todas partes del mundo. En estos casos la mantenibilidad del sistema está garantizada.

El problema comienza con los componentes libres pequeños o con los componentes propietarios producidos por compañías pequeñas.

La desaparición del proveedor anterior es uno de los problemas más críticos de falta de mantenimiento y el takeover es la solución inmediata pero puede ser una de las más costosas.

En el caso del software propietario puede resultar desastrosa la desaparición del proveedor de un componente ya que éste queda inmediatamente fuera de mantenimiento y se vuelve necesaria una fuerte inversión para poder reemplazarlo. El Software Libre ofrece a priori el código fuente del componente abandonado por lo que la estrategia de takeover no necesariamente significa su reemplazo sino que puede significar su adopción con un costo mucho más bajo que el costo del reemplazo.

Incluso se abre la posibilidad de que sea la propia organización que está usando el módulo quien lo adopte por medio de sus áreas de sistemas para seguirle dando mantenimiento y emitir actualizaciones de funcionalidad y de seguridad....

Tiene sentido (tras haber tomado el mantenimiento en manos propias) liberar las actualizaciones de

funcionalidad y de seguridad cuando existen otras organizaciones (e individuos) que estén usando el mismo componente ya que inmediatamente se reactivan los mecanismos de reporte de errores y envío de mejoras que pueden ser inmediatamente aprovechadas por la organización que se está encargando del mantenimiento en este momento.

El costo del mantenimiento de un componente puede ser elevado. Su justificación es la necesidad de ese componente como parte de un sistema y un costo mayor que significaría el completo reemplazo de ese módulo.

7. Costo.

Evidentemente el costo total de propiedad se compone de la suma de todos los costos anteriores y algunos otros, pero sigue habiendo una gran diferencia de costos entre comenzar desde cero un proyecto de millones de líneas de código contra construir nuestros propios sistemas sobre millones de líneas de código que han escrito otras personas y contribuido a la humanidad como un bien público.

El costo del uso de componentes propietarios es un punto intermedio ya que proporcionan esos millones de líneas de código a cambio de dinero y del cumplimiento de condiciones de licenciamiento propietarias. El ahorro al usar componentes libres sigue siendo significativo.

Es importante evaluar además de los ahorros en licenciamiento los ahorros indirectos derivados de la personalización, la portabilidad, la independencia de proveedor, la mantenibilidad y la rapidez de despliegue.

8. Tecnología de punta.

Un porcentaje importante de los proyectos de Software Libre disponibles en Internet han salido de universidades o centros de investigación y contienen innovaciones tecnológicas de vanguardia. Otro porcentaje menor está desarrollado por individuos independientes que también están proponiendo tecnologías innovadoras.

Es cierto que existen compañías dedicadas al software propietario que se han dedicado a invertir en proyectos innovadores salidos de las universidades y es también cierto que hay compañías dedicadas al software propietario que están financiando su propia investigación para sacar al mercado productos innovadores.

La diferencia entre la innovación que provee el Software Libre y la innovación que provee el software propietario radica en la percepción del mercado: Las compañías dedicadas al software propietario tratan de aplicar cambios disruptivos con la intención de capturar al mayor número de clientes y posteriormente enfocan todos sus esfuerzos en retenerlos mientras que los proyectos de Software Libre tratan de aplicar esos cambios disruptivos al bien común y su éxito radica en la proliferación masiva de esta tecnología en donde su mayor adopción va a garantizar su mejora continua.

Encontramos más tecnología de punta propietaria en aplicaciones de alta especialización mientras que en aplicaciones de uso general o del bien común (como Internet) encontramos más tecnología de punta en el Software Libre.

La decisión en este caso depende completamente de la misión y de la visión que tenga la organización y del grado de especialización que esta organización quiera alcanzar para alcanzar su visión y cumplir con su misión.

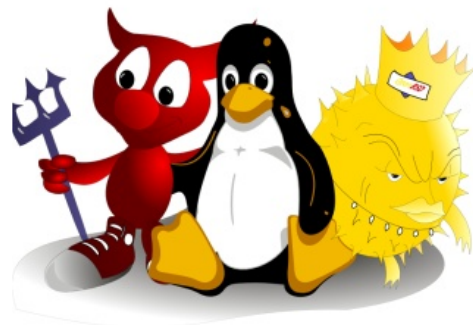


En la fotografía, Sandino Araico platicando sobre Malware nivel 3. BarCamp México v1.0

.FIC 2007

Software Libre en la Universidad de Cádiz

Luis Alberto Fernández — luisalberto.fernandez@uca.es



Durante los pasados días 7, 8 y 9 de marzo se celebró en la Facultad de Ciencias Sociales y de la Comunicación del Campus de Jerez el primer Congreso Científico de Software Libre (FLOSSIC 2007 <http://softwarelibre.uca.es/fic>), organizado por la Oficina de Software Libre de la Universidad de Cádiz y el grupo de investigación Mejora del Proceso Software y Métodos Formales y el Departamento de Lenguajes y Sistemas Informáticos con la colaboración de la Escuela de Negocios de Jerez. Este congreso ha nacido con la vocación de promover la difusión de los avances científicos referentes al uso del software libre, y se organizará anualmente en distintas universidades.

El objetivo de este congreso es ser un marco de encuentro para las principales iniciativas relacionadas con los FLOSS, incidiendo especialmente en aquellas relacionadas con la Universidad: educación, tecnología e investigación.

El congreso fue un éxito de asistencia, con más de 30 ponentes de varios países y más de 100 asistentes, incluyendo gran cantidad de alumnos de estudios técnicos de informática, que asistieron a comunicaciones de temas tan diversos como el e-learning, accesibilidad, traducción automática, desarrollo de software, modelado 3D profesional, supercomputación o inteligencia artificial.



Entre las ponencias destacó "Impact of Free/Libre or Open Source Software (FLOSS) on the

European ICT sector", presentada por Rüdiger Glott, de la Universidad UNU-Merit. En ella se presentaron las conclusiones de un informe realizado recientemente para la Unión Europea en el que se muestran las ventajas económicas y competitivas que aporta el uso y desarrollo de software libre en las empresas TIC de la Unión Europea, así como los efectos negativos que provocaría la adopción de patentes de software o formatos de almacenamiento de información cerrados.

Todas las ponencias quedaron reflejadas en el libro de actas del congreso, publicado (como no podía ser de otra forma) con una licencia libre que permite su copia y distribución gratuita. Este libro se repartió entre todos los asistentes dentro de un CD recopilatorio de documentación libre que incluye más de 200 libros, manuales y cursos sobre sistemas libre y que se puede solicitar gratuitamente en la Oficina de Software Libre de la Universidad de Cádiz o descargar desde <http://flossic.loba.es>

La organización del congreso quiso agradecer la ayuda prestada por sus patrocinadores ORO: Consejería de Innovación, Ciencia y Empresa de la Junta de Andalucía, Sadiel y HP, los patrocinadores Activa Sistemas y Yaco Sistemas, y las entidades colaboradoras: Facultad de Ciencias Sociales y de la Comunicación, Consejo Social y los Vicerrectorado de Alumnos, de Extensión Universitaria y de Investigación, Desarrollo Tecnológico e Innovación de la Universidad de Cádiz así como la empresa Loba Soluciones Informáticas y todos los miembros del comité de organización.

RevistaSL agradece al FLOSS International Conference por darnos la oportunidad de colaborar como medio asociado en este magno evento. Felicidades.

CONSOL 2007

Convergencia tecnológica del siglo XXI

Odin Mojica Ramírez

odinmojica@gmail.com



El Congreso Nacional de Software Libre (CONSOL) tuvo sin duda un año sobresaliente, tanto en audiencia, ponencias, publicidad, así como la creación de un nuevo track, el de Robótica. Este año la sede fue por segunda ocasión en la Universidad Nacional Autónoma de México (UNAM) en la Facultad de Ingeniería y se realizó del 13 al 16 de febrero, regresando a su fecha tradicional, en comparación con el año anterior, que fue llevado en el mes de agosto.

El congreso sin duda, ha llegado a ser uno de los principales puntos de reunión de la comunidad del software libre en México, ya que a lo largo de sus 6 ediciones se ha consolidado como un punto de referencia indiscutible dentro de los diferentes actores que convergen en el software libre mexicano, contando con la colaboración tanto de gente del extranjero como de ponentes nacionales quienes aportaron sus conocimientos y experiencias, haciendo de este Congreso un mar de ideas y ríos de conocimientos.

La forma de trabajo y desarrollo del software libre, así como la generación de conocimientos, se lleva a cabo en su gran mayoría en forma remota, hablando en términos computacionales, las reuniones presenciales como el CONSOL, aportan diversos beneficios, entre los cuales destacan el intercambio de conocimientos y de experiencias. Gracias a estas reuniones, también es posible la creación de nuevos proyectos y el aumento de colaboraciones en los proyectos ya existentes, nutriendo cada vez más al software libre. Todo esto con base en los principios fundamentales de la filosofía del software libre: Compartir, modificar, mejorar y distribuir conocimiento a través del software.

De los objetivos principales del CONSOL, además de dar a conocer esta opción tecnológica y de integrar a la comunidad que se encuentra involucrada, y darla a conocer a todas aquellas personas que no lo están, es hacer hincapié en la filosofía del software libre, pues ha sido la base de este movimiento con más de 20 años de vida y que partiendo de ella se ha podido llegar al punto en donde actualmente se encuentra. La filosofía del software libre es tan importante como el software mismo que se genera y por consecuencia no se puede dejar pasar por alto.



El CONSOL tiene una dinámica muy especial con respecto a otros eventos, su temática y desarrollo busca acercarse a todas las áreas que abarca el mundo del software libre. teniendo como ejes:

- Generación de conocimiento.
- Difusión del conocimiento.
- Acercamiento de los actores tanto nacionales como internacionales.
- Difusión de los trabajos de la comunidad de SL hacia las instituciones gubernamentales, educativas e industrias, a nivel nacional.

Las charlas que se llevaron a cabo en los distintos tracks temáticos abordados, fueron impartidas de manera simultánea en los diferentes auditorios, logrando un congreso muy dinámico en donde el asistente pudo seleccionar los temas de su mayor interés, ya fuera por cuestiones profesionales, personales o por simple interés cultural.

Novedades del congreso

El congreso pretende ser cada vez más novedoso e irse adaptando a los avances y novedades de la tecnología, así como del mundo del software libre, prueba de ello fue la presentación del nuevo track de Robótica, un tema de mucha importancia y en el cual México ha hecho grandes aportaciones, participando en varios eventos y concursos a nivel mundial, en los que universidades y centros tecnológicos nacionales han salido triunfadores,

colocándose en los primeros sitios, es por eso que la importancia de este tema no podía dejarse fuera de este CONSOL. Se han mostrado nuevos proyectos durante este congreso, así como iniciativas, ideas y la creación de grupos de usuarios. La difusión de este evento en otros congresos fue favorable para recuperar el número de asistentes que años anteriores se han tenido. Una de las grandes sorpresas este año, fue la gran cantidad de visitantes de escuelas, institutos del interior del país, así como de las dependencias de gobiernos municipales, estatales y federales.

Los retos del CONSOL

Uno de los retos del CONSOL es el seguir acercando tanto a la comunidad del software libre, con la gente que no pertenece a ella, con la finalidad de que continúe creciendo, y con ello, el usos de sistemas libres y abiertos lleguen a mayor cantidad de lugares y usuarios, dado que no podemos negar su eficiente funcionalidad y operatividad, es la mejor opción en lo que respecta al mundo del software.

Otro reto igual de importante, es realizar una mejor organización en los Congresos subsecuentes, con una adecuada logística y una mayor difusión en los medios masivos de comunicación, lo que dará como resultado mejores Congresos, y mayores retos.



Las Conferencias Magistrales.

Se tuvieron conferencias magistrales de Alvaro López (Sun Microsystems) sobre su proyecto Cherokee, a Javier Solís de Infotec hablando de WebBuilder, James D. Whitfield de Harvard University dando sus experiencias en la programación científica con Python, Beatriz Busaniche de Fundación Vía Libre que trató el tema de Gestión Digital de Restricciones, Enrique Chaparro de la misma fundación donde hablo de Reflexiones de un espíritu curioso sobre el futuro del software libre, Marcus Börger desarrollador de Standards de las librerías de PH hablando de de PHP 5.2 a PHP 6.0, Leonardo Yamasaki Maza de la Universidad Autónoma de Chiapas donde trató

el tema de Robótica académica y software libre, y a Héctor Colina que hablo sobre la seguridad del Kernel linux.

Pueden encontrar algunos audios de las charlas en <http://www.frecuenciacerero.com.mx/con45/> los cuales nos apoyaron durante el evento con estas grabaciones.

Datos estadísticos importantes

En el año 2005 se hizo por primera vez streaming por Internet2 (I2), sin embargo no se tuvo la audiencia e impacto esperados, toda vez que el Internet2 se centra únicamente a Instituciones Académicas y de investigación, en comparación con el Internet que es público. En el presente año, se realizó streaming por medio de Internet, logrando resultados más favorables, pues se tuvieron accesos de países como Argentina, Brasil, Colombia, Ecuador, Honduras, Venezuela, Estados Unidos, así como del interior del país, teniendo un total de 380 accesos.

Las universidades e institutos que nos acompañaron este año, según nuestros registros, fueron: Universidad Autónoma del Estado de México, Instituto Tecnológico de San Luis Potosí, Universidad Veracruzana, Universidad Autónoma de Colima, UAM Iztapalapa, Universidad Madero Campus Papaloapan, Universidad Michoacana de San Nicolás de Hidalgo, Universidad Autónoma de Chiapas, Universidad de Ciencias y Artes de Chiapas.

Universidad de Guadalajara, Universidad Mexiquense, Universidad Politécnica de Querétaro, Universidad Tecnológica de Puebla, Universidad Tecnológica de Izúcar de Matamoros, Universidad del Valle de México (3 campus), Centro de Investigación y Docencia Económica, Instituto Tecnológico Superior de Libres y del Centro de Investigación en Computación IPN.

Como experiencia.

Para parte de los Organizadores, nos queda otra grata experiencia, desde la organización y el haber podido cumplir con los objetivos principales, el compartir con la comunidad el conocimiento y proyectos, y que estos a su vez fueran difundidos. No me queda más que agradecer a nombre de todos los integrantes del comité organizador, al LIDSOL y a todos los que directa o indirectamente hicieron posible todo esto. Esto nos motiva a trabajar con más ganas para el siguiente CONSOL 2008 esperamos contar con su presencia en la siguiente versión. GRACIAS A TODOS. Esperamos sus comentarios a la lista de comité at consol.com



NUEVO INICIO DE CURSOS

Cursos Intensivos
Inicio 19 de Mayo

Costo Promocional
\$3,000.00
pesos mexicanos

CURSOS

Duración 60 Horas
CD de Sistema Operativo
CD de Material Didáctico

Apoyo y Asesoría en Línea
Garantía de Servicio
Constancia de Participación

**1 Implementación de Clientes
y Servidores con GNU/Linux**

**2 Programación para Web
y Bases de Datos**

Interlegit.com.mx

DESARROLLO Y SOLUCIONES WEB

contacto@interlegit.com.mx

ventas@interlegit.com.mx

<http://www.interlegit.com.mx>

Ing. Braulio Martinez No 192-D, Col. Guadalupe Insurgentes, Del. Gustavo A. Madero C.P. 07870
Tel: (55) 57-39-62-45 A dos calles del Metro Potrero, esquina con Insurgentes.



BAZARLIBRE.COM

Software Libre
Hosting
Souvenirs



El "urkonn" entrevista a Leo Utskot

Un noruego achilangado

Julio Acuña Carrillo

julio.acuna@revista-sl.org



RSL. ¿Nos podrias platicar un poco de tí?

LU.- Mi nombre es Leo Utskot, tengo 30 años y soy de Noruega. Llevo ocho años viviendo en México. Hace 3 años arranqué una empresa de desarrollo web llamada Copyleft con base al modelo de negocio de Copyleft Software en Oslo, Noruega. Soy un geek orgulloso y he sido amante de software libre desde hace unos diez años.

RSL. ¿Cómo se integra el software libre en una empresa como Copyleft?

LU. Utilizamos software libre en nuestros servidores y desarrollamos principalmente bajo la plataforma LAMP (Linux, Apache, MySQL y PHP). Adicionalmente utilizamos software como MediaWiki y ActiveCollab para organizarnos internamnete.

RSL. Copyleft ha producido algunas aplicaciones que ha lanzado bajo la GPL, ¿qué nos puedes platicar de estos proyectos?

LU. Aquí en México desarrollamos FreeFall, un sistema web de subastas inversas. Hay información adicional en <http://freefall.copyleft.com.mx/>

En general todo nuestro código viene con la licencia GPL v2, pero por falta de tiempo no hemos publicado mucho. Si tuviera un poco de tiempo libre, tenemos varios proyectos interesantes incluyendo un CMS modular, un ERP light y un CRM especializado para la industria inmobiliaria.

Copyleft Noruega tiene varios proyectos incluyendo un servidor de email completo llamado Shemail, un sistema de ticket tracker llamado USSR y muchos proyectos que venden allí.

RSL. ¿Cuál es el beneficio que obtienen las empresas al usar software libre?

LU. Hay 3 beneficios centrales.

Primero es el beneficio más obvio de ahorro de dinero en licencias. Nosotros y nuestros clientes no pagamos un peso en licencias y mantenimiento

para tener productos de la misma calidad que por ejemplo IIS y MS SQL. Por otro lado el uso de software libre nos conecta con una comunidad gigante de desarrolladores con una gran variedad de conocimiento. A pesar que la documentación se encuentra muy dispersa y de diferentes calidades, uno nunca se siente perdido con software libre. Una búsqueda rápida en Google siempre te regresa un howto o un comentario en algún foro sobre tu problema. Con software propietario dependes de una sola empresa y sus distribuidores. Si ese grupo pequeño no ha escuchado de tu problema, probablemente no tendrá solución.

Y por último el uso de software libre es la única opción ética. Suena un poco radical pero creo que es nuestra obligación luchar por la libertad de conocimiento que cada día más queda en manos de grandes empresas multinacionales. La economía global hoy en día está en una fase de transformación y en el futuro será en gran parte basado en información en vez de recursos naturales. Si permitimos que el software, junto con el resto del conocimiento humano quede en manos de las empresas bajo el pretexto de que es su "propiedad intelectual" tendremos un futuro donde no existan innovaciones y avances sin la aprobación de algún plan de negocios. Eso será el fin de pequeños descubrimientos hechos por simple curiosidad como por ejemplo antibióticos, la imprenta y el avión.

RSL. ¿Los clientes están conscientes de lo que es el software libre?

LU. Generalmente tenemos que hacer una pequeña labor de educación para explicar la diferencia entre libre y gratis y para mostrar que no estamos tratando de robar nada. Cada vez más hay clientes que piden específicamente el uso de software libre.

RSL. En el ambiente empresarial siempre se habla de opensource y no de software libre, ¿crees que el uso de uno u otro termino cause confusión cuando se trata de convencer al jefe de hacer uso de una tecnología en específico?, ¿cuál sería tu recomendación?

LU. Creo que el término de Open Source es muy peligroso. Fue creado por personas que no entienden los conceptos e ideales centrales del movimiento de software libre. No se trata de sólo el código, y no es suficiente sólo mostrar un poco de código para estar en lo correcto.

Software libre se trata de libertad y la creación de una comunidad global unida en contra de la idea de que alguien pueda ser dueño del conocimiento e ideas. Eso incluye abrir el código, pero también implica dejar que el código pueda ser modificado como sea y ser utilizado para cualquier fin. Puede ser que suene radical, pero lo mismo dijeron de la gente en favor de igualdad entre los sexos y la gente en contra de esclavitud.



RSL. En los últimos años los modelos de negocios han cambiado, ¿qué papel juega el software libre en esto?

LU. No estoy tan de acuerdo que los modelos de negocios han cambiado. Siempre se trata de crear un producto que el consumidor quiere y venderlo a un precio que el consumidor puede pagar. El modelo capitalista no ha cambiado desde sus principios.

Lo que si ha cambiado son los gustos del consumidor. El consumidor, sea una persona en la calle o sea una empresa, cada vez más gastan menos dinero en comprar objetos físicos para sus necesidades básicas. Antes era una gran ocasión cuando un hombre se compraba nuevos zapatos. Gracias a la revolución industrial y la globalización puedo comprarme un par de zapatos en el mercado cada mes. Lo mismo sucede en las empresas. Los nuevos productos tienen que ver con información y conocimientos. Sean archivos digitales para tu iPod o un nuevo ERP para la empresa son productos que cuestan casi nada reproducir pero tienen un valor muy alto en el mercado.

El software libre puede jugar un papel muy interesante para dar una ventaja competitiva a la gente y las empresas que lo ocupan porque puede reducir el costo de estos nuevos productos.

RSL. Las empresas tienen que adaptarse a los cambios, y esto es especialmente cierto si hablamos de tecnología, ¿cómo ves las estrategias de grandes empresas como Novell o Sun Microsystems?

LU. Novell, honestamente, es un dinosaurio de los 90s y no creo que tenga el perfil para ser la empresa que finalmente saca Linux al desktop. Era obvio que no tenían futuro con su tecnología de Netware y para intentar revitalizar su línea de productos hicieron la compra de Ximian y Suse. El problema es que hay un gran choque cultural dentro de Novell entre el área de administración/financiera y sus nuevas adquisiciones. Vi un par de videos donde el CEO pasado trató de presentar el nuevo producto de Desktop en un Keynote sin saber siquiera como pronunciar Linux, luego llegó Nat Friedman para rescatar la presentación. Ya hicieron unos cambios importantes de alto nivel para involucrar más gente joven y cambiar un poco de su cultura empresarial, pero al mismo tiempo hacen errores de relaciones públicas como su trato con Microsoft. No importa qué dice el acuerdo, el simple hecho que una empresa de software libre haga tratos con el enemigo en Redmond indica que aún no entienden de qué se trata.

Parece que Sun está haciendo un esfuerzo real para apoyar al software libre y su decisión de lanzar OpenSolaris bajo la GPL es un paso importante para ganar credibilidad. Sólo el futuro nos puede decir cuáles han sido sus motivaciones.

RSL. ¿Qué opinas acerca de la alianza Novell-Microsoft?

LU. Espero que Novell despida a su área de relaciones públicas. Fue un error muy grave e independientemente de los detalles técnicos hizo mucho daño a la imagen de Novell ante la comunidad de software libre.

Por otro lado, no creo que sea muy importante.

RSL. ¿Algo más que quieras agregar?

LU. Veo que hay mucho talento en software libre en México, sobre todo en las universidades. Quiero extender la oferta a los estudiantes con ideas de arrancar negocios de platicar con Copyleft para ver si podemos apoyarlos de alguna forma. Últimamente hemos platicado de funcionar como incubadora para jóvenes que tienen una buena idea para un negocio, pero con necesidades de oficina, ayuda con lo administrativo y sobre todo apoyo moral. Pueden contactarme por medio de leo@copyleft.com.mx

Calendario SL

la agenda de los geeks...

Victor Hugo Cordova

buzon@revista-sl.org



Festival Latinoamericano de Software Libre 2007

<http://www.installfest.net>

El FLISOL (Festival Latinoamericano de Instalación de Software Libre) es el evento de difusión de Software Libre más grande en Latinoamérica. Se realiza desde el año 2005 y su principal objetivo es promover el uso del software libre, dando a conocer su filosofía, sus alcances, avances y desarrollo al público en general.



Festival Latinoamericano de Instalación de Software Libre



Para tal fin, las diversas comunidades locales de software libre (en cada país, en cada ciudad/localidad), organizan simultáneamente eventos en los que se instala de manera gratuita y totalmente legal, software libre en las computadoras que llevan los asistentes. Además, en forma paralela, se ofrecen charlas, ponencias y talleres, sobre temáticas locales, nacionales y latinoamericanas en torno al Software Libre, en toda su gama de expresiones: artística, académica, empresarial y social.

El FLISOL 2007 se efectuará el día sábado 28 de abril.

RevistaSL desea mantenerse cerca de la comunidad de Software Libre. Si deseas dar aviso de algún evento que organices, conoscias o deseas publicidad no dudes en escribir a buzon@revista-sl.org

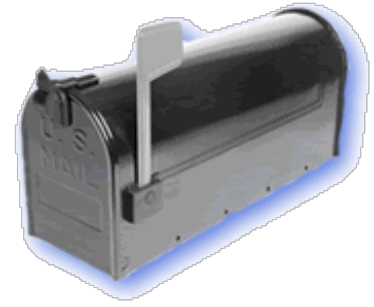
Navarro & Co. dect

Sorprende, da miedo, no lo queremos enfrentar... pero al final, nos atrapa. por Humberto Morales Sánchez (HUMD)



Buzón SL

Nuestros lectores opinan...



Victor Hugo Cordova

buzon@revista-sl.org

Hola, buenos días,

Os hago llegar la nota de prensa de OpenVz con el anuncio de la disponibilidad de su software de virtualización para el nuevo kernel de Red Hat Enterprise Linux 5 (RHEL 5). Si desean probar el software pueden hacerlo en el siguiente enlace: <http://openvz.org/download/kernel/rhel5>

Esta nueva mejora se engloba indudablemente en el deseo de servir de forma cada vez más completa a la comunidad de usuarios de software libre.

Para cualquier pregunta o información adicional, no dudéis en contactarme.

Gracias por vuestro interés y apoyo. Un saludo

Gosia Pajkowska

INTRO IBÉRICA CONSULTORES

gosia@introiberica.com

RevistaSL desea saber lo que opinan sus lectores. Escribenos a buzon@revista-sl.org y cuéntenos tus inquietudes, comentarios, críticas, tips, anuncios y todo lo que se te ocurra.

News & Contact

¡La tecnología telepática a la vuelta de la esquina!! por Humberto Morales Sánchez (HUMD)





RevistaSL

El software libre hecho revista

Colaboradores SL

RevistaSL agradece a:

Andres Bernardo Vargas

Alvaro de Miguel

Exal de Jesus García

Grupo de Usuarios de Software Libre de Cuautla

Leo Utskot

Luis Alberto Fernández

Martin Marquez

Odín Mojica Ramirez

Ozram Intriago

Sandino Araico Sánchez

~# Bytez SL



El proyecto "Revista@ SL, la revista mexicana del Software Libre" nació el 17 de Febrero del año 2005, fruto de la colaboración de la Comunidad Linux Chetumal y la Fundación de Software Libre A.C. en México.

Sin embargo este proyecto prácticamente ha desaparecido, las decenas de lectores que seguían la publicación se convirtieron en miles, los lejanos y extraños colaboradores se convirtieron en grandes y cercanos amigos y el adjetivo "mexicana" quedó chico al crecimiento del proyecto.

Esto no fue casualidad, es el resultado de un trabajo arduo por parte de las personas que bimestre a bimestre trabajan en la elaboración de la presente publicación.

A todos ellos es a quien este número está dedicado, a todos aquellos que integran el equipo editorial, editores, diseñadores, correctores, webmasters; a todos aquellos que colaboran escribiendo artículos; a todos aquellos que nos regalan un espacio para distribuir la revista; a los patrocinadores que han confiado en nosotros y a los lectores que nos felicitan y otras tantas veces nos advierten de nuestros errores.

Muchas gracias a todos los que han hecho posible dos años de "RevistaSL, el Software Libre hecho revista".

Buzón SL

buzon@revista-sl.org

Colaboraciones

colaboraciones@revista-sl.org

Publicidad

publicidad@revista-sl.org

Carlos Augusto Lozano Vargas