



2 Juntos Crecemos

Revista de Software Libre ATIX

2008

Reconocimiento-Compartir bajo la misma licencia

Usted es libre de:



copiar, distribuir y comunicar públicamente la obra



hacer obras derivadas

Bajo las condiciones siguientes:



Reconocimiento. Debe reconocer los créditos de la obra de la manera especificada por el autor o el licenciador (pero no de una manera que sugiera que tiene su apoyo o apoyan el uso que hace de su obra).



Compartir bajo la misma licencia. Si altera o transforma esta obra, o genera una obra derivada, sólo puede distribuir la obra generada bajo una licencia idéntica a ésta.

- 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
- Nada en esta licencia menoscaba o restringe los derechos morales del autor.

Colaboradores

Dirección y Coordinación General

Esteban Saavedra López (jesaavedra@opentelematics.org)

Diseño y Maquetación

Jenny Saavedra López (jennysaavedra@hotmail.com)

Esteban Saavedra López (jesaavedra@opentelematics.org)

Colaboración

Williams Chorolque Choque (williamsis@gmail.com)

Herramientas

La edición de esta revista fue realizada de forma íntegra haciendo uso de Software Libre





**Palabra quechua,
con un sentimiento profundo
y con gran significado filosófico**

El que lo sabe

El que lo intenta

El que lo puede

El que lo logra

Dar el primer paso siempre es importante, pero también estamos seguros que igual o más importantes aún serán los pasos siguientes, por que éstos reflejarán una continuidad en el compromiso adquirido con toda nuestra comunidad, en la tarea de difundir el conocimiento, las tecnologías y el Software Libre.

Contentos y muy satisfechos al ver que nuestro primer pequeño aporte fue bien recibido por todas las personas que dedicaron parte de su tiempo a leer las páginas de nuestra revista.

Que bueno fue recibir durante éste corto tiempo palabras de felicitación y sobre todo de apoyo a este noble proyecto desde dentro y fuera de nuestras fronteras, pero más importante fue ver la curiosidad y la motivación que despertamos en jóvenes y personas de áreas no técnicas, interesados en aprender e inmiscuirse en este maravilloso mundo del Software Libre.

En éste número tenemos dos aspectos a ponderar, el primero la entrevista a **Alexandre Oliva** de la **Fundación de Software Libre América Latina**, a quién ofrecemos nuestra gratitud por concedernos una entrevista; en la que muestra detalles y características del Software Libre de una forma totalmente explicativa y motivadora; y la segunda referida a que nuestra colaboradora en el diseño y maquetación (Jenny) ya no es la única dama presente en la revista, ya que el sector femenino sienta presencia en este segundo número con 2 artículos, esperemos que sirva de motivación para muchas más colaboradoras.

Juntos crecemos, realmente un titulo de lo más apropiado para este número, por que refleja todo el apoyo que la gente nos brinda y espero sigan haciéndolo, ya que juntos haremos que no sólo nuestro conocimiento crezca, sino también lograremos ese gran objetivo que perseguimos todos: **“el de Juntos Crecer como personas, colaborando y ayudando a los demás”**.

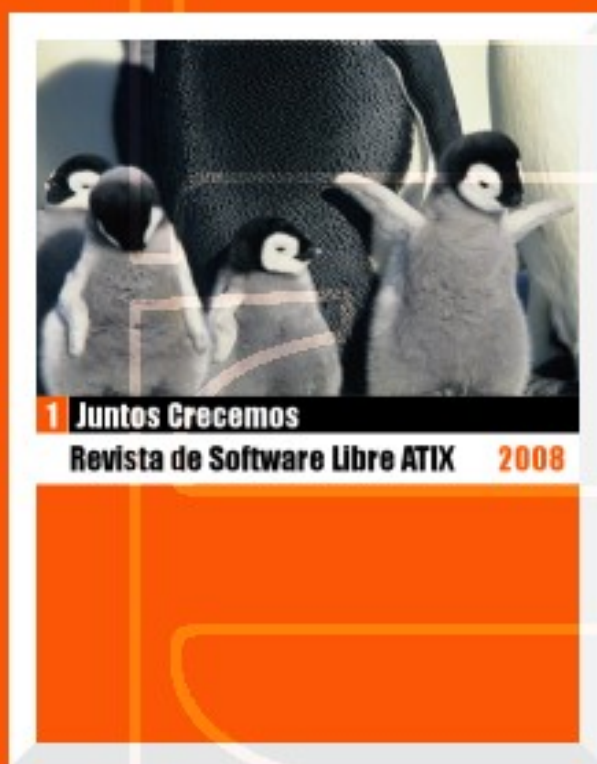
Con estas palabras les doy la bienvenida a este segundo número, esperando que sea de vuestro agrado y nos permita ratificar la confianza que vosotros depositaron en nuestra Revista.

Esteban Saavedra López
Director y Coordinador General

Contenido

Liberado el 20 de junio de 2008

- 7 Instalación paso a paso de Tuquito 2.0
- 16 Vim guía de supervivencia (2da parte)
- 20 Ogre3D un alternativa para el desarrollo gráfico
- 24 VSFTPd una forma de permitir y mantener el control de la transferencia de archivos
- 29 Openssh la magia de la administración remota
- 35 Bazaar la herramienta para el control de versiones de forma distribuida
- 43 La entrevista: Alexandre Oliva - FSFLA
- 53 La noticia: Nuevo portal de la comunidad de Software Libre Bolivia
- 57 Tips: Numeración de PostgreSQL
- 58 Comics
- 59 Conociendo lo nuestro - Turismo y Libertad
- 62 Arte Libre
- 63 Información de contacto



Instalación paso a paso de Tuquito 2.0

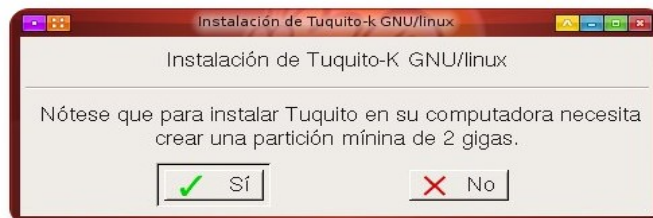
Ésta segunda entrega describe la instalación paso a paso de la distribución **Tuquito**.

Lo primero que tenemos que hacer es arrancar nuestro PC con el CD de **Tuquito** (LiveCD), seguramente tendremos ésta pantalla de inicio



Con doble clic en la opción **Instalar Tuquito** iniciamos el proceso de instalación.

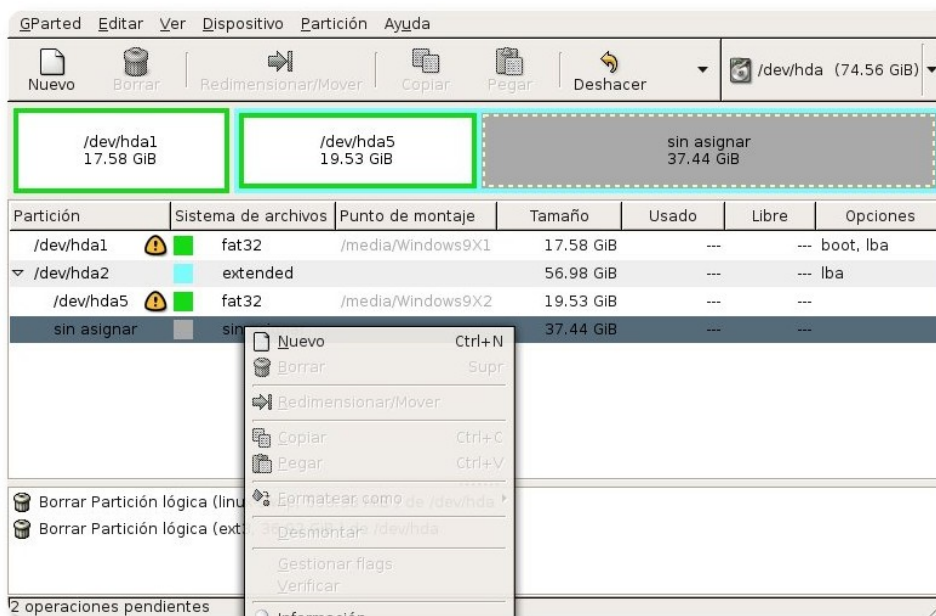
Nótese que necesitamos una partición de nuestro disco duro mínimo de 2 gigas para instalar de forma correcta ésta distro, si cumplimos con este requerimiento, hacemos un clic en Si



Gparted es una herramienta gráfica estilo fdisk para winbugs muy intuitiva, con un clic en Ok iniciaremos el proceso de particionado.

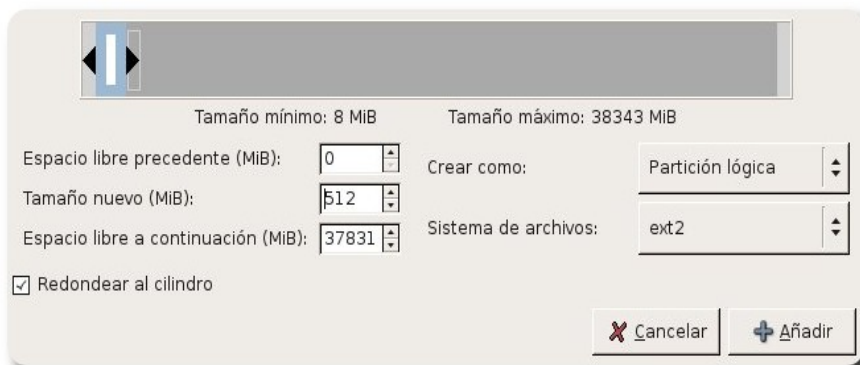


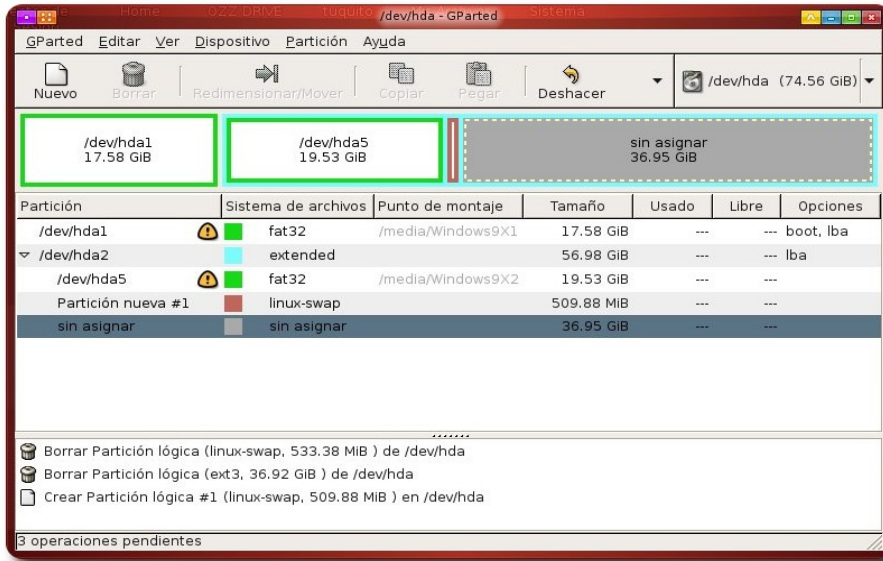
Si observamos en la gráfica, tenemos un espacio sin asignar (color gris). Con clic derecho sobre esta partición, se desplegará un menú contextual, en el cuál seleccionamos la opción Nuevo



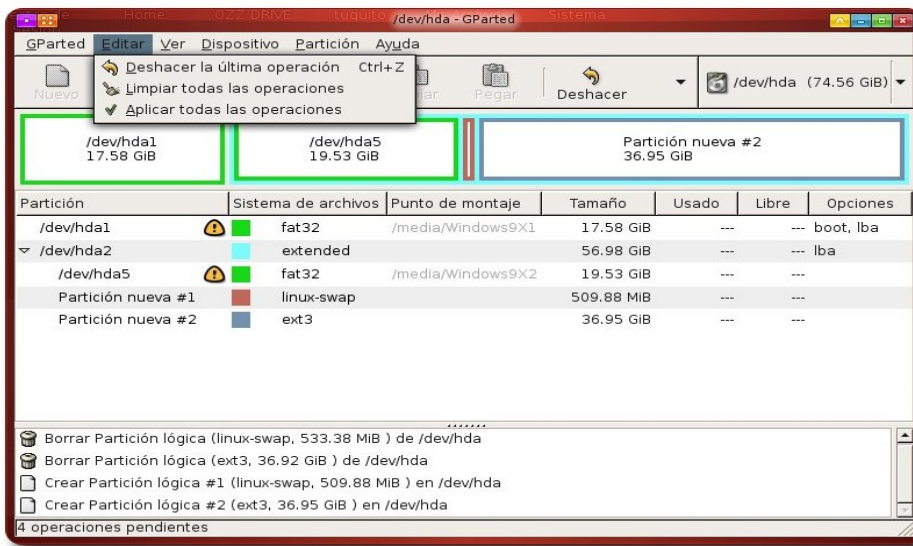
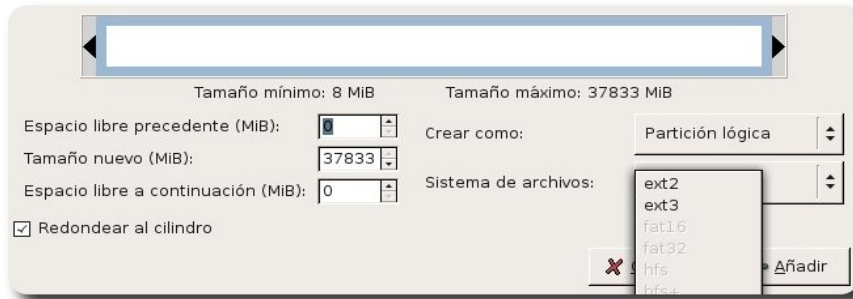
Podemos asignarle 2 particiones, una SWAP(memoria SWAP o de intercambio) que es la que vamos a hacer en este paso; ésta partición es muy al estilo memoria virtual en winbugs, le ayudará a nuestra memoria RAM cuando esté sobrecargada con muchos procesos.

Ahora en la opción sistema de archivos escogemos linux-swap, asignamos el tamaño adecuado (en muchos foros recomiendan asignarle un tamaño igual al doble de nuestra memoria RAM, por ejemplo en mi caso la memoria RAM es 256Mb por eso le asigno 512 de swap)

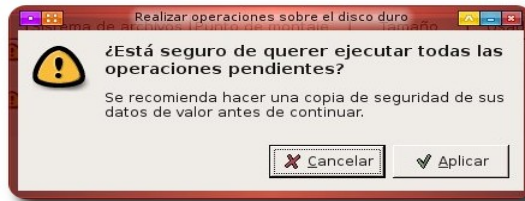




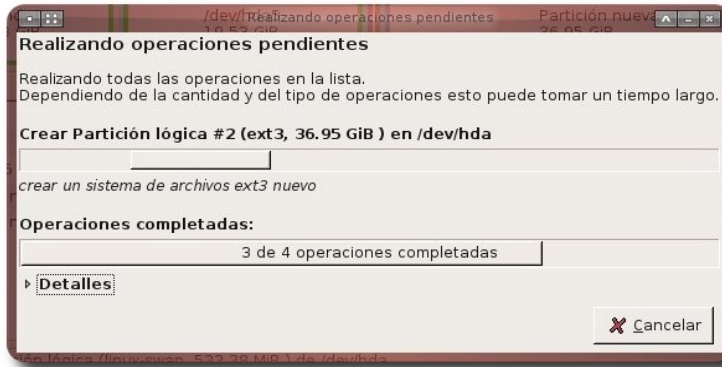
Con el mismo procedimiento damos un formato a la partición destinada al sistema base, que cuyo sistema de archivos es EXT3, ésto sería muy similar al de FAT32 o NTFS de winbugs. Todo listo para instalar **Tuquito**, vamos a la opción de Editar en el menú de Gparted y colocamos Aplicar todas las operaciones



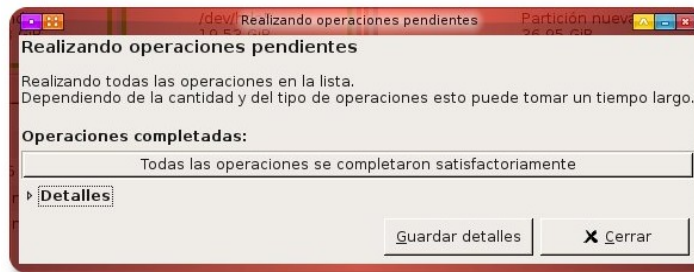
La partición que hayamos asignado como EXT3 se formateará y por supuesto se perderá todo lo que existía en ella, tomando nuestras previsiones le damos Aplicar.



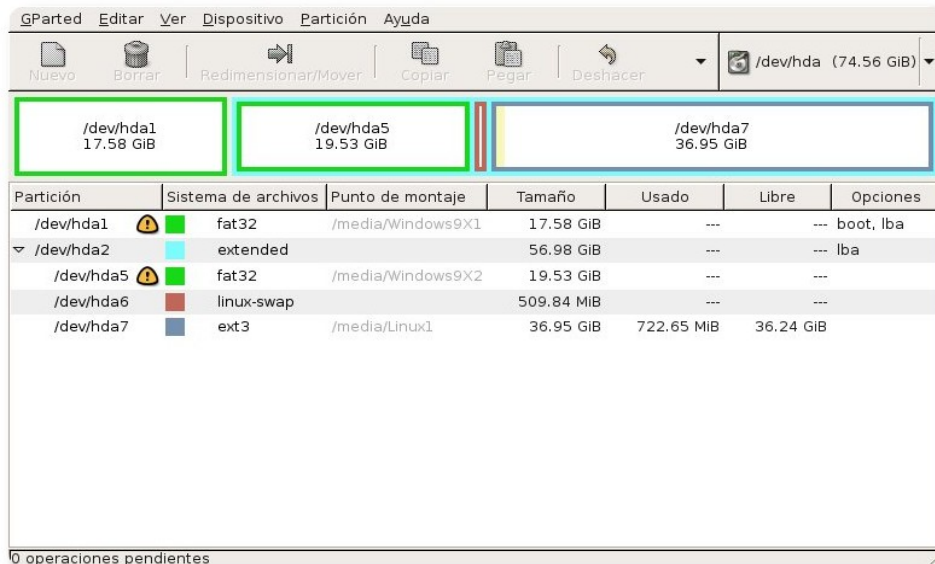
Tendremos que esperar unos momentos para que se lleven a cabo las acciones anteriormente configuradas



Ésta es la ventana que asegura que todo se realizó bien

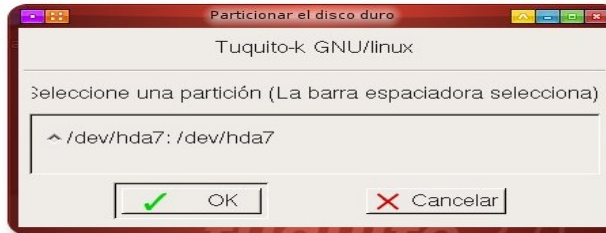


En el Gparted observamos que todo se realizó como lo esperado.

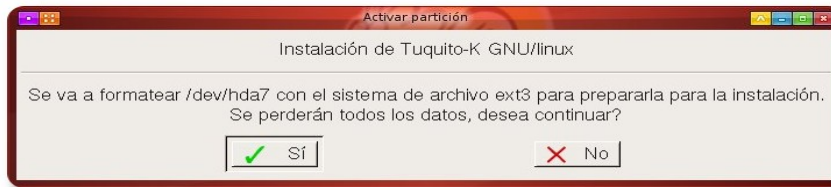


Tenemos que tomar nota de 2 cosas: la primera los puntos de montaje de nuestras particiones del disco duro. Observamos hda1, hda2,...,hda7 esto llegaría a ser como las unidades C,D,...,E en winbugs

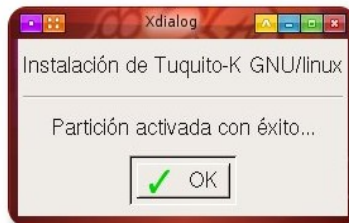
Al cerrar Gparted nos saldrá una ventana en la que nos pide seleccionar la partición dónde estará instalada **Tuquito**, y tenemos que indicarle la formateada con EXT3, en nuestro caso hda7



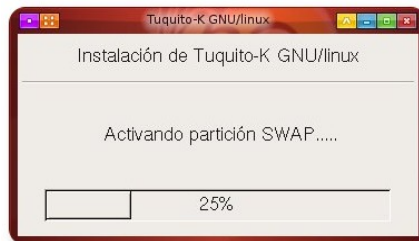
Vean que indica que la partición hda7 (para nuestro caso), en tu PC puede que sea hda#, todo dependiendo de cuantas particiones tenga tu disco duro



Una vez que la operación se realizó con éxito, nos informará mediante un mensaje como éste



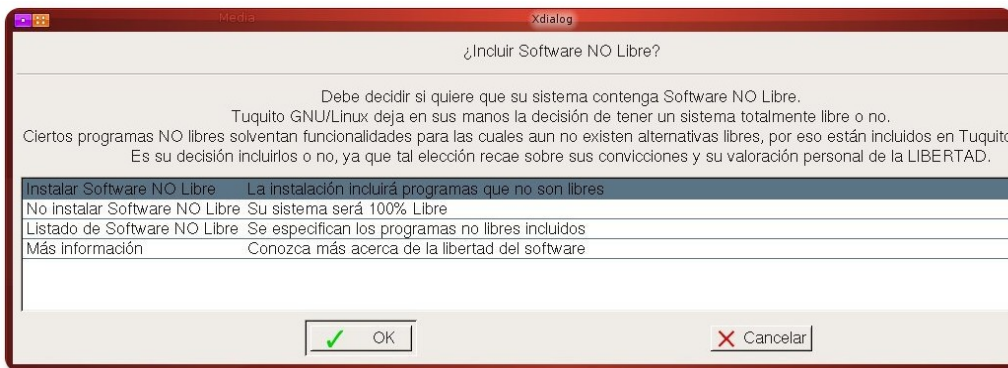
Tan sólo hay que esperar a que termine la instalación de **Tuquito**



Como se ve en la imagen, simultáneamente se esté instalando, podemos seguir utilizando el LiveCd a su plenitud, no abusen mucho, pues tardará más en instalarse



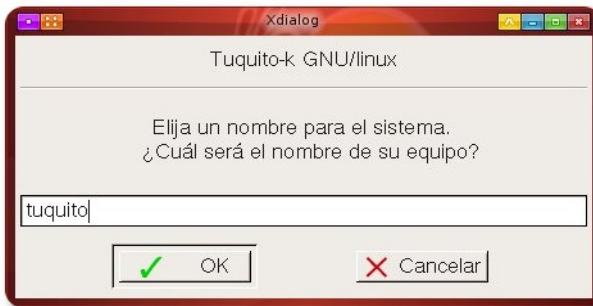
Algo muy importante **Tuquito** nos pregunta si queremos instalar Software privativo (no libre) y que de alguna forma se paga para su uso, por ejemplo: Codecs para mp3, mpg, etc. De cada uno dependerá la elección.



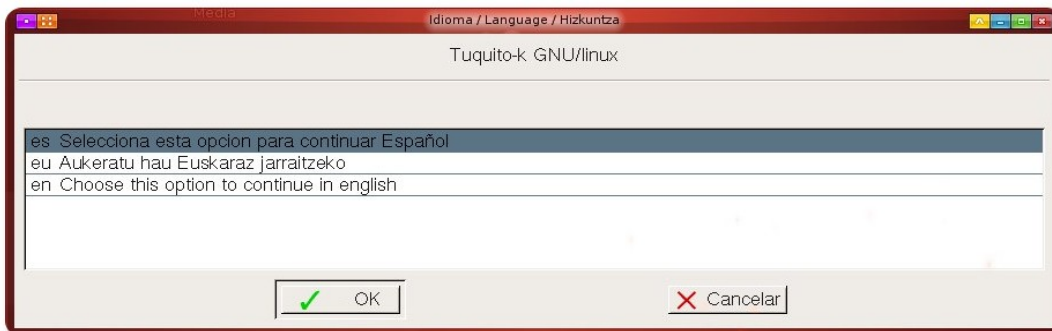
Posteriormente nos pregunta si queremos tener efectos de escritorio 3D obviamente le decimos que Sí , también si disponemos de una tarjeta de vídeo Nvidia



En este paso nos pide introducir un nombre de usuario para autenticarnos en el sistema, ingresen uno y presionen Ok



Procedemos a seleccionar el idioma, en nuestro caso el Español.



El Grub son las opciones que tenemos al momento de arrancar la Pc, si tenemos instalado winbugs, no hay problema al momento de encender la PC nos dará para escoger con cuál sistema operativo iniciar.



Al fin ya terminó nuestra instalación con éxito total sólo queda pulsar Ok

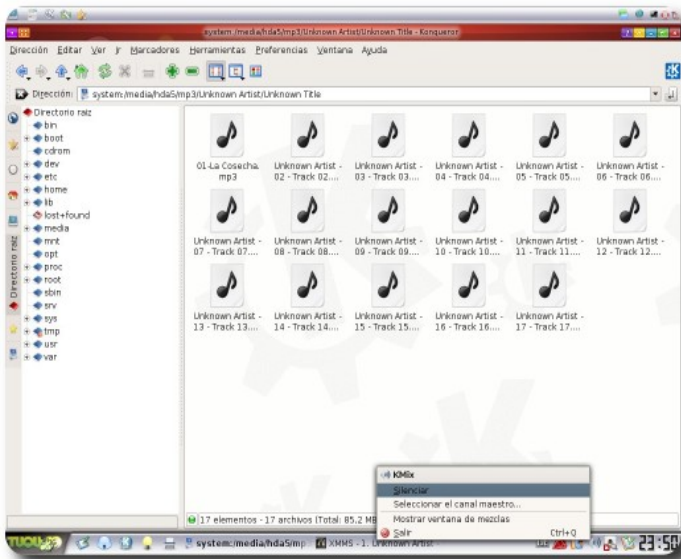


Una vez instalado el sistema podemos seguir usando el LiveCd de **Tuquito**., para probar nuestra instalación tenemos que reiniciar la PC.

Que esperas, de una vez, **reinicia tu PC!!!!!!!!!!!!!!**



La primera pantalla después de reiniciar nuestra PC, y algo muy importante, por defecto el sonido está desactivado lo único que tenemos que hacer es activarla con un clic derecho con el ratón en el icono de volúmen del panel inferior, luego abrir su consola y subir el volúmen de las entradas que tengamos



Ahora sólo nos queda disfrutar de las bondades del Software Libre de la mano de **Tuquito 2.0**.



Cualquier duda o comentario hacerla llegar a la página de la comunidad ATIX.
Gracias y hasta una nueva oportunidad

Referencias

[1] www.tuquito.com.ar

Autor



Oscar Alfredo Choque Monzón

oscar.ch.monzon@gmail.com

Estudiante de Ingeniería de Sistemas

Vim guía de Supervivencia (2da parte)

Esta segunda entrega muestra más características de éste magnífico editor de textos



En el anterior número de nuestra revista vimos una pequeña introducción a **Vim** y el uso de los comandos básicos pero muchos seguramente piensan que hasta ahí llega **Vim** y muchos pensarán que ¿Cómo es posible que ni siquiera tenga coloreado la sintaxis?. Bueno pues se equivocan ya que **Vim** es una herramienta altamente personalizable, de tal manera que podemos convertirla en una súper herramienta y de eso es precisamente de lo que hablaremos en ésta ocasión veremos algunas de sus características que harán un poco mejor y más fácil nuestra vida a la hora de editar archivos de texto.

Cambiando el color de fondo

La primera vez que abrimos cualquier consola ésta tiene un esquema (El conjunto de colores del fondo y texto) predeterminado, por lo general fondo claro y letras oscuras y **Vim** tiene algo similar pero resulta que cuando invertimos el esquema de la consola **Vim** no cambia y es que **Vim** viene predeterminado para trabajar con la configuración por defecto del sistema así que podemos ejecutar lo siguiente en modo línea de comando:

- ✓ **:set background=dark** Lo que obtenemos con éste comando es que **Vim** asumirá que el fondo es negro o algún color oscuro por lo tanto ahora

tiene que presentar el texto con letras claras.

- ✓ el opuesto de éste comando es **:set background=light** que devuelve a **Vim** a su configuración original.

Coloreando el Texto

En las capturas de texto del anterior número vieron documentos coloreados y seguramente si es la primera vez que usaron **Vim** en sus equipos habrán observado que en el documento que hayan abierto no sucedió esto es debido a que tenemos que habilitar esta opción, pero como hacemos esto? Simplemente tienen que colocar lo siguiente

- ✓ **:syntax on** ésta opción le indica a **Vim** que tiene que habilitar el resaltado de sintaxis que dependerá mucho del tipo de documento que estemos editando y del color de fondo que tengamos configurado para deshabilitar ésto sólo tenemos que colocar **:syntax off** y volvemos al modo anterior; pero una vez usado **syntax on** es poco probable querer dejarlo.

Organizando nuestro Código

Muy bonito dirán muchos pero veo que tengo que seguir usando el tabulador o espacios para poder ordenar mi código y en mi súper editor X ya tengo eso, bueno en **Vim** tampoco es necesario colocar los espacios para la indentación o sangrado automático de código a mano, en éste caso también podemos modificar éste comportamiento mediante:

- ✓ **:set autoindent** y automáticamente tendremos la indentación de nuestro documento ésto dependiendo del

formato de dicho documento.

- ✓ **:set cindent** ésta es otra forma de obtener sangrado automático en nuestro documento pero se basa en los bloques lógicos de la programación en C/C++

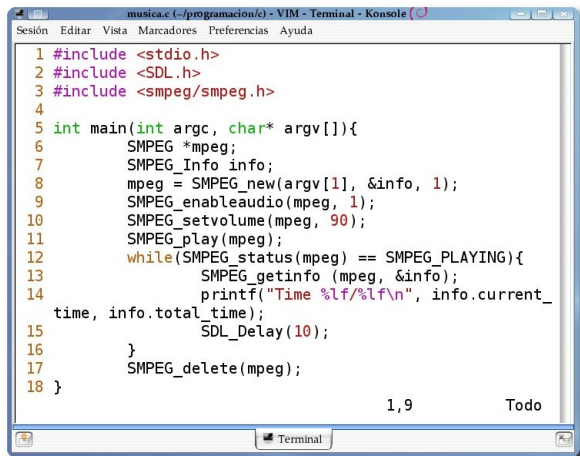


Figura 1 Documento con background=light

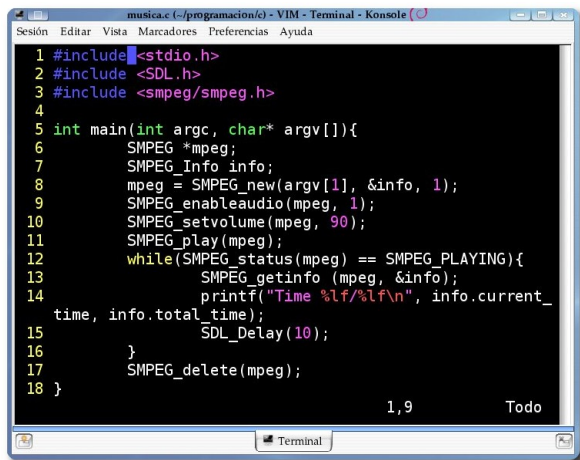


Figura 2 Documento con background=dark

Números de línea

Si bien en la parte inferior de la ventana podemos observar el número de línea y columna actual podemos mejorar nuestra orientación sobre en qué línea estamos haciendo uso del comando **:set number** y así automáticamente tendremos una mejor referencia del número de líneas de nuestro documento.

Hasta el momento hemos personalizado mucho nuestro documento, pero lastimosamente todo esto se perderá en cuanto cerremos **Vim** y tendremos que volver a realizarlo la próxima vez que carguemos **Vim**, salvo que lo guardemos en un archivo de configuración y **Vim** si tiene un archivo de configuración que lo podemos encontrar en **/etc/vim/** el archivo se llama **vimrc** todo lo que coloquemos en éste archivo automáticamente será cargado la próxima vez que usemos **Vim**. Otra opción es colocar un archivo similar en nuestro directorio personal algo como **~/.vimrc** y en éste archivo colocar las configuraciones correspondientes, como en la figura 3

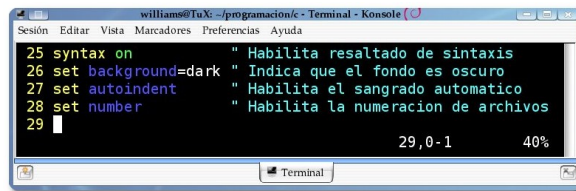


Figura 3 líneas agregadas al archivo /etc/vim/vimrc

Trabajando con varias ventanas

Trabajar con varios archivos es algo común y pensar que tenemos que cargar **Vim** cada vez que modificamos un archivo sería un poco molesto (sin contar con el número de terminales a abrir) pero afortunadamente podemos usar varias ventanas de manera simultánea haciendo uso del comando **:split** o **:sp** el cual dividirá nuestra ventana de manera horizontal si queremos dividirla de manera vertical **:vsplit** o **:vsp** de ésta manera automáticamente tendremos el mismo documento abierto en la nueva ventana; podemos realizar éste proceso varias veces de tal manera que podamos tener abierto todos los documentos que necesitemos modificar o consultar al mismo tiempo.

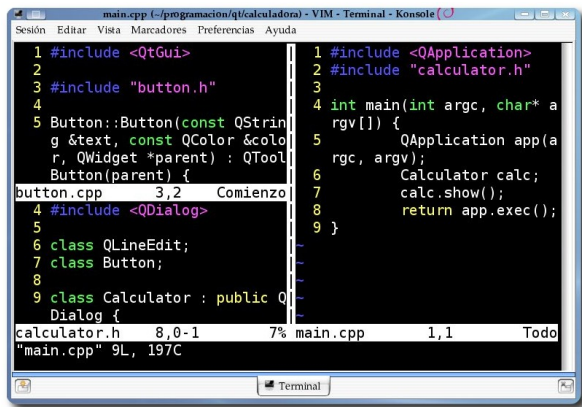


Figura 4 Vim con varias ventanas

Ahora veamos algunas combinaciones de teclas útiles:

- ✓ **Ctrl+ww** Moverse a la ventana siguiente.
- ✓ **Ctrl+wq** Abandona la ventana actual también es válido :q
- ✓ **Ctrl+w + h o j o k o l** aunque no vimos éstas teclas en el anterior artículo las veremos ahora ya que las mismas son de mucha ayuda cuando queremos movernos por un documento.
- ✓ **h** Ir a la izquierda
- ✓ **j** Ir abajo
- ✓ **k** Ir arriba
- ✓ **l** Ir a la derecha

Todo esto en modo comando, si sólo presionamos esas teclas nos movemos dentro del documento, pero si lo hacemos en combinación de **Ctrl+W** nos moveremos a través de las ventanas, también pueden hacer uso de las teclas de dirección.

Para cargar, modificar en general interactuar con el archivo actual debemos seguir usando los comandos vistos en el anterior número.

Usando pestañas

Si bien el anterior método de tener varias ventanas es útil, para aquellos que tenemos monitores pequeños se puede convertir en un problema ya que no podremos visualizar de manera adecuada los documentos que estemos editando cuando tenemos más de 2 documentos al mismo tiempo. Como solución a esto tenemos el uso de pestañas para ello sólo tenemos que usar los siguientes comandos:

- ✓ **:tabedit** o **:tabe** Abre una nueva pestaña vacía.
- ✓ **:tabedit <nombre de archivo>** o **:tabe <nombre de archivo>** Abre una nueva pestaña con el archivo indicado.
- ✓ **:tabclose** Cerrar la pestaña actual también es válido **:q** o cualquiera de las formas de cerrar vistos en el anterior número.

En la parte superior podremos observar una serie de pestañas con los nombres de los archivos abiertos, para movernos podemos usar

- ✓ **:tabprevious** o **:tabp** Ir a la pestaña anterior.
- ✓ **:tabnext** o **:tabn** Ir a la pestaña siguiente.
- ✓ **:tabfirst** **:tabfir** Ir a la primera pestaña
- ✓ **:tablast** o **:tabl** Ir a la última pestaña

Cada vez que se modifique una pestaña veremos que se agrega el símbolo “+” al inicio del nombre en la pestaña correspondiente.

Puede resultar un poco molesto tener que escribir cada vez éstos comandos así que podemos hacer los siguiente en modo línea de comando.

- ✓ **:map <F6> <Esc>:tabp<CR>**
- ✓ **:map <F7> <Esc>:tabn<CR>**

¿Qué es lo que estamos haciendo? Pues sólo estamos asignando éstos comandos a las teclas F6 y F7 así que ahora lo único que tenemos que hacer es presionar Esc y la tecla F6 si queremos ir a la anterior pestaña o F7 si queremos ir a la siguiente pestaña. Y si quieren configurar éste comportamiento por defecto sólo tienen que agregar éstas líneas a su archivo de configuración en **/etc/vim/vimrc** o en **~/.vimrc** y no será necesario realizarlo cada vez.

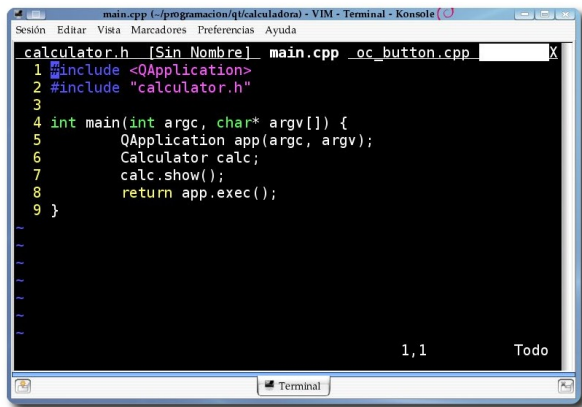


Figura 5 Haciendo uso de pestañas en Vim

Algo que me gusto mucho en C/C++ es que cuando colocan el cursor sobre el nombre de una librería o biblioteca que estemos usando y presionamos **Ctrl+w** seguido de **gf** ésta librería se abre automáticamente en una nueva pestaña, hagan la prueba con `stdio.h` y podrán observar el contenido de dicha biblioteca.

Me gustaría continuar mostrando más de las bondades de éste maravilloso editor pero mejor lo dejamos para el siguiente número donde hablaremos sobre **Vim** y la

programación en C/C++ y veremos que **Vim** tiene todas las herramientas necesarias para programar en éste lenguaje y en cualquier otro, pero nos abocaremos a éste por ser uno de los más usados en las universidades. Hasta el siguiente número y no olviden enviarnos sus comentarios y participar en la revista.

Referencias

- [1] <http://www.vim.org>
- [2] <http://es.wikipedia.org/wiki/Vim>

Autor



Williams Israel Chorolque Choque
 Estudiante de Ingeniería Informática
 email: williamsis@gmail.com

Ogre3D Una alternativa para el desarrollo gráfico

El desarrollo de juegos y animaciones gráficas se ha convertido hoy en día en una de las aficiones, de la mayoría de los desarrolladores, quienes deben considerar las diversas variantes en cuanto a librerías y herramientas disponibles que se tiene, Ogre3D es una de estas librerías que dispone de una serie de ventajas.



Introducción

Ogre3D (Object-oriented Graphics Rendering Engine) como dice su nombre, es una librería gráfica (también llamado motor gráfico), que puede ser incorporado en una variedad de aplicaciones que necesiten gráficos de alta calidad, para cualquier proyecto que desee mucha flexibilidad y facilidad de configuración en sus módulos de graficación, para desarrolladores que deseen la facilidad de uso que les da el paradigma de la programación orientada a objetos en la cual está desarrollada, finalmente para cualquier persona que desee utilizar una herramienta de desarrollo que les da libertad sobre su trabajo y demás ventajas que ofrece el software libre.

Ogre3D es utilizado principalmente en videojuegos (no todos son libres), lo que hace creer que dicha librería es un motor de videojuegos en 3D, cosa que no es cierta.

Ogre3D es una librería bastante especializada únicamente en gráficos: para crear un programa de simulación o un videojuego, es necesario tener muchas más herramientas: motores de física y colisión de objetos, librerías de entrada de datos(joysticks), librerías de red y también de sonido.

Por tanto, **Ogre3D** sigue la filosofía de desarrollo del software libre, de hacer una pequeña cosa, pero hacerla bien: los gráficos.

A la vez existe la libertad de elegir librerías de terceros que no estén relacionadas necesariamente con **Ogre3D** para combinarlos en los proyectos finales.

Muchos proyectos utilizan esta librería ampliamente. **Ogre3D** esta diseñado para una variedad de escenarios y altos requerimientos de sistemas comerciales. Entre algunos ejemplos podemos encontrar títulos como ZeroGear, Kong y Building world.



La mayoría de las librerías están diseñadas para ser combinadas con otras herramientas y librerías. Los autores de **Ogre3D** ponen énfasis en la naturaleza especializada de la librería. Usar **Ogre3D** por sí solo, es como usar un motor de un automóvil sin nada más. La mayoría de programas y herramientas con los cuales se puede combinar **Ogre3D** son modeladores 3D, aquellos que definen la

forma de los escenarios, personajes y cosas, además de definir como se mueven.



Screenshot de Building World

Es mucho más fácil usar una herramienta de diseño en 3D que programar algo por si mismo, puesto que un objeto simple tiene miles, o millones de polígonos y una compleja definición de materiales que deberían ser puestos en archivos que **Ogre3D** lee desde dentro de un programa. Una opción bastante popular es usar Blender, un programa libre de modelado en 3D, combinado con un exportador de modelos que **Ogre3D** puede fácilmente leer.



Ogre3D fue escrito originalmente en C++, sin embargo, usuarios de otros lenguajes pueden usar capas de abstracción intermedias para su uso sin problemas, como **Ogre3D4j** que es una interfaz para Java, o también la interfaz python-**Ogre3D**, su equivalente para python, finalmente axiom

una versión escrita en C# para tecnología .NET y Mono.

Técnica de renderizado "cell shading" en **Ogre3D**, usado para simular dibujos en 2D



La Licencia

La librería **Ogre3D**, está licenciada bajo LGPL (Lesser GPL), que a primera impresión puede parecer que no es del todo software libre. De hecho si lo es, LGPL fue escrito por el mismo Richard Stallman. LGPL difiere principalmente de GPL, por el hecho que la primera permite a cualquier software propietario o no usar la librería libre. Este es según Stallman, una cuestión estratégica, que sólo el desarrollador del programa o librería debe decidir. Cuando existen tanto alternativas libres y propietarias de la misma confiabilidad y calidad, un desarrollador de software propietario que quiera hacer uso de una determinada librería, no podrá usar aquellas que estén bajo licencia GPL, puesto que su software también debe ser licenciado libre, así que elegirá la alternativa no libre, en ese caso, la librería debería permitir estratégicamente ser usado por software propietario también, recuerden que mientras más programas usen software y por tanto, estándares libres, se estará haciendo mucho bien al mundo del software en general, y por tanto dando ventaja al mundo del software libre.

En muchos casos existe que, el software libre ofrece muchas más ventajas que ninguna librería de software propietario puede igualar, en este caso, usar GPL como la licencia de esta librería que obliga al desarrollador a liberar su programa bajo GPL también, o a

seguir siendo software propietario, pero sin las ventajas antes mencionadas. La idea es tratar de llevar el software libre a la mayor cantidad de usuarios posible. Como mencioné anteriormente es la estrategia.

Ogre3D también viene con una licencia propia y alternativa(en caso de que el usuario sienta que no esta de acuerdo con LGPL), que es OUL(**Ogre3D** Unrestricted Licence). Dos razones principales de la existencia de esta licencia:

Modificar el código fuente de **Ogre3D**, sin hacer públicas dichas modificaciones

Hacer un link estático de librería de **Ogre3D** con alguna aplicación, sin liberar el código fuente.

Ogre3D no necesariamente es usado en sistemas que comparten la filosofía del software libre.



Oportunidades de desarrollo

Un desarrollador que domina y desarrolla aplicaciones, u otras librerías puede cobrar por los servicios de implementar x, y ó z funcionalidad, de hecho, me pareció interesante encontrar una zona de "reclutamiento" dentro de los fotos de la comunidad **Ogre3D**, en el cual se busca gente talentosa para proyectos basados en esta librería que van desde pequeños trabajos (escritura de módulos), hasta búsqueda de líderes de proyecto con bastante experiencia en este tipo de sistemas. Obviamente existen trabajos remunerados, tanto como voluntarios. Una prueba de que desarrolladores de software libre pueden vivir de lo que les gusta hacer.



Dependencias

Ogre3D depende de librerías de bajo nivel 3D: OpenGL y DirectX. Es decir que **Ogre3D** es una capa de software de abstracción de más alto nivel que esas dos librerías, funciona utilizándolas por dentro, es como usar C++ sabiendo que este lenguaje al final se traduce a lenguaje binario de máquina. También es posible pero no indispensable usar de la librería de alto nivel de renderizado de Nvidia: Cg toolkit(shading language), para darle más poder de desarrollo a **Ogre3D**.

Videoconsolas

La idea de utilizar **Ogre3D** en las videoconsolas comerciales de última generación (Wii, PlayStation 3, o Xbox360) es algo que se viene a la mente después de ver las capacidades de este motor gráfico, de hecho una antigua versión no oficial de **Ogre3D** para Xbox ya existía para los desarrolladores independientes del tipo "homebrew" (homebrew = hecho en casa). Es posible portar **Ogre3D** a cualquier videoconsola, ya que muchas de sus características están basadas en PCs existentes. Sin embargo, aún no existe una versión oficial de **Ogre3D** para ninguna videoconsola de última generación, por las dificultades de financiamiento y logísticas. El hecho de que **Ogre3D** sea oficialmente soportado, por alguna videoconsola necesita inversión económica, para licencias del fabricante, para esto es necesario apoyo de grandes compañías que planean usar **Ogre3D** en sus futuros proyectos dirigidos al mundo de los videojuegos. Por el momento

Ogre3D seguirá siendo una alternativa para PCs, aunque existe bastante entusiasmo de la comunidad para portar **Ogre3D** consolas de última generación. El mundo de los juegos es bastante competitivo, y a veces bastante cruel. Sería excelente contar en el futuro con videoconsolas orientadas al software libre, tal como ocurrió con la antigua GP32.

Ogre3D no es la única alternativa de este nivel para crear videojuegos, también podemos mencionar XNA de Microsoft, orientado a plataformas Windows y XBOX360, e Irrlicht, que es una alternativa también libre, con características similares a **Ogre3D**.

En resumen podemos decir que **Ogre3D** da muchas libertades al usuario desarrollador, incluyendo la libertad de hacer su nueva aplicación no libre.

También el usuario final goza de muchas ventajas, puesto que puede disfrutar muchas más aplicaciones de simulación en tiempo real de alta calidad que usen software libre y estándares también libres por dentro, si es que dicha aplicación no es ya libre en su totalidad.

Referencias

- [1] <http://www.Ogre3D.org/>
- [2] <http://Ogre3D4j.sourceforge.net/>
- [3] <http://python-Ogre3D.org/>
- [4] <http://axiomengine.sourceforge.net>
- [5] <http://www.gnu.org/licenses/why-not-lgpl.html>
- [6] <http://irrlicht.sourceforge.net/>
- [7] <http://es.wikipedia.org/wiki/GP32>

Autor



Arnold Marcelo Guzmán

Desarrollador

spacerockganimedes@gmail.com

VSFTPD: una forma de permitir y mantener control de la transferencia de archivos

VSFTPD (**Very Secure FTP Daemon**) es un software utilizado para implementar el servicio de transferencia de archivos a través del protocolo FTP. Caracterizado principalmente por su seguridad y sencillez en su configuración. Actualmente se presume que VSFTPD es quizá el servidor FTP más seguro del mundo.

Introducción

FTP (**File Transfer Protocol**) o Protocolo de Transferencia de Archivos, es uno de los protocolos estándar más utilizados en Internet para la transferencia de grandes bloques de datos a través de redes que soporten TCP/IP.

Este servicio utiliza los puertos 20 y 21, exclusivamente sobre TCP. El puerto 20 es utilizado para el flujo de datos entre cliente y servidor. El puerto 21 es utilizando para el envío de órdenes del cliente hacia el servidor.

Generalmente todos los sistemas operativos y plataformas incluyen soporte para FTP,, permitiendo acceder a este servicio desde cualquier ordenador de la red.

Objetivos

- ✓ **Transferencia** de archivos por medio de una red de ordenadores .
- ✓ Brindar la **mayor seguridad** tanto en el acceso como en la transferencias .

Características

- ✓ La **seguridad** que brinda en el servicio .
- ✓ La **sencillez** en su configuración.

Que precisamos

Para hacer uso de este servidor precisamos instalarlo, en caso que no esté ya (la mayoría de las distribuciones ya incluye por defecto).

Su instalación se la puede hacer mediante el gestor de paquetes propio de la distribución:

- ✓ synaptic en Debian o Ubuntu
- ✓ yumex en CentOS o Fedora.

Conexión anónima

Inicialmente por defecto **VSFTPD** permite solamente el acceso anónimo, restringiendo las operaciones solamente a lectura de los ficheros.

Configuraciones principales

El archivo donde reside la configuración se encuentra generalmente en `/etc/vsftpd.conf`, archivo que deberá ser editado y modificado según las necesidades.

Acceso de usuarios registrados

Si deseamos que nuestros usuarios registrados en el sistema puedan acceder al servicio con sus respectivas cuentas (usuario

y contraseña) y realizar operaciones de lectura y escritura, debemos habilitar los parámetros `local_enable=YES` y `write_enable=YES` respectivamente.

Mensajes de bienvenida

Muchas veces queremos mostrar un mensaje de bienvenida que identifique nuestro servidor o servicio, a los usuarios que accedan a éste, podemos hacerlo mediante el parámetro: `ftpd_banner=Bienvenido al servidor de Desarrollo del proyecto Opentelematics`.

Tiempos de gracia

Si deseamos ser más estrictos en el tema de establecer tiempos máximos de conexión, transferencia, tiempos máximos durante la conexión, debemos hacer uso de los parámetros: `idle_session_timeout=120`, `data_connection_timeout=300`, `accept_timeout=60` y `connect_timeout=60`.

Tasas de transferencia

En el caso de disponer de una buena cantidad de usuarios y una conexión lenta, muchas veces es necesario establecer o limitar la tasa de transferencia, ésta tasa de transferencia puede ser asignada a las conexiones anónimas como a las conexiones de usuarios registrados, por medio de los parámetros: `anon_max_rate=5120` y `local_max_rate=5120` respectivamente.

Permisos de archivos por defecto

Algunas veces precisamos que los archivos al ser subidos al servidor tengan ciertos permisos de forma automática, estos pueden ser asignados mediante el parámetro: `local_umask`, por ejemplo `local_umask=022` indicará que los permisos de los archivos

serán 644.

En algún caso, pero no es recomendable se puede habilitar para que los usuarios anónimos puedan subir archivos y crear directorios, esto puede ser habilitado mediante los parámetros: `anon_upload_enable=YES` y `anon_mkdir_write_enable=YES` respectivamente.

Enjaular a los usuarios

Por temas de seguridad, no es conveniente que los usuarios conectados a nuestro servidor puedan desplazarse fuera de su directorio `HOME$`, por lo cual debemos enjaularlos y restringir su movimiento solo en su directorio `HOME$` y subdirectorios, esto se logra mediante el parámetro: `chroot_local_user=YES`, debemos mencionar que existen formas de afinar más aún este control, permitiendo solo a ciertos usuarios estar enjaulados o no, todo esto es posible mediante el parámetro `chroot_list_enable=YES` y el fichero `chroot_list_file=/etc/vsftpd/vsftpd.chroot_list` que contiene la lista de usuarios excluidos de la función `chroot`.

Restricciones de acceso

Una forma de restringir el acceso y dotarle de seguridad es limitar el número máximo de conexiones permitidas en un determinado instante y el limitar el número máximo de conexiones que se puede realizar desde una determinada dirección IP, esto se logra mediante los parámetros: `max_clients=5` y `max_per_ip=5` respectivamente.

Registro de Logs

El registro de logs es muy importante dentro de la gestión de cualquier servicio, ya que nos permite verificar y monitorear la actividad que presenta un servicio, en el caso de VSFTPD debemos activar el parámetro `xferlog_file=/var/log/vsftpd.log`

Conexión y operaciones desde consola

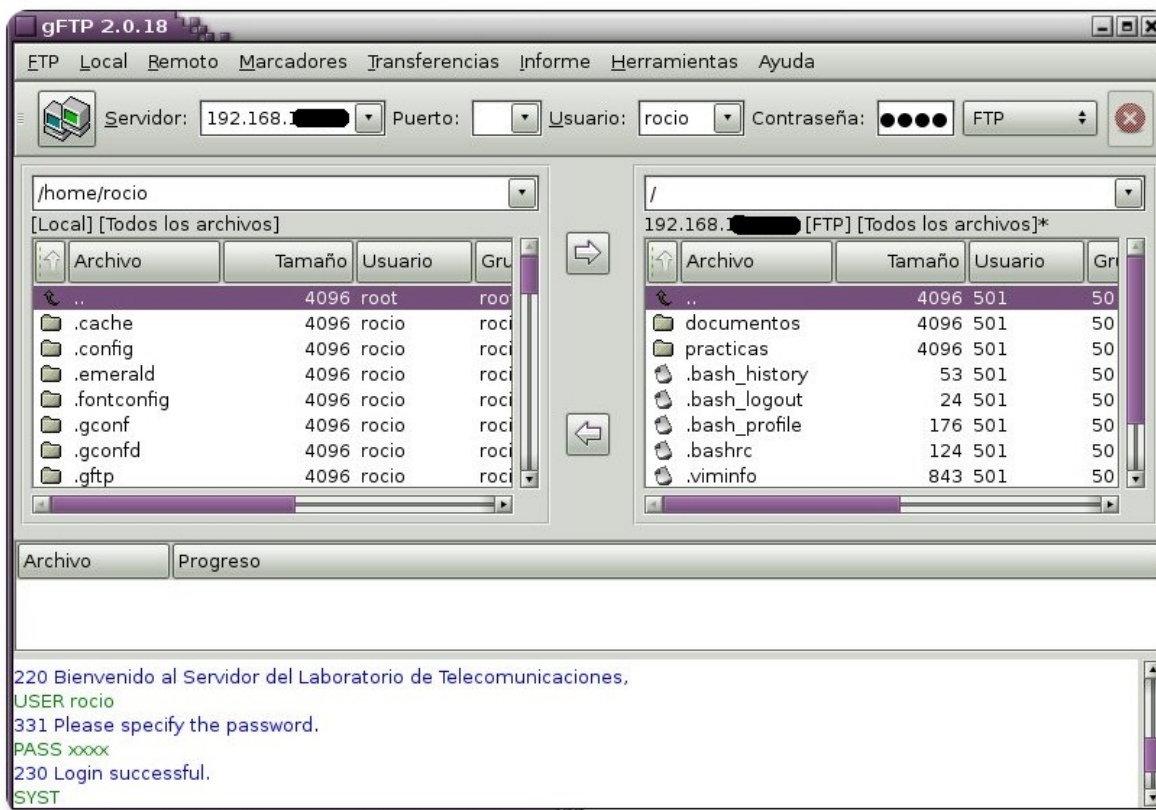
Generalmente los administradores prefieren hacer uso de la línea de comandos para realizar sus distintas tareas administrativas, el servicio de ftp no es la excepción, la conexión desde consola donde se pueden realizar comandos como: get, put, etc que permiten efectuar la transferencia de archivos desde o hacia un servidor, presenta la siguiente apariencia:

```

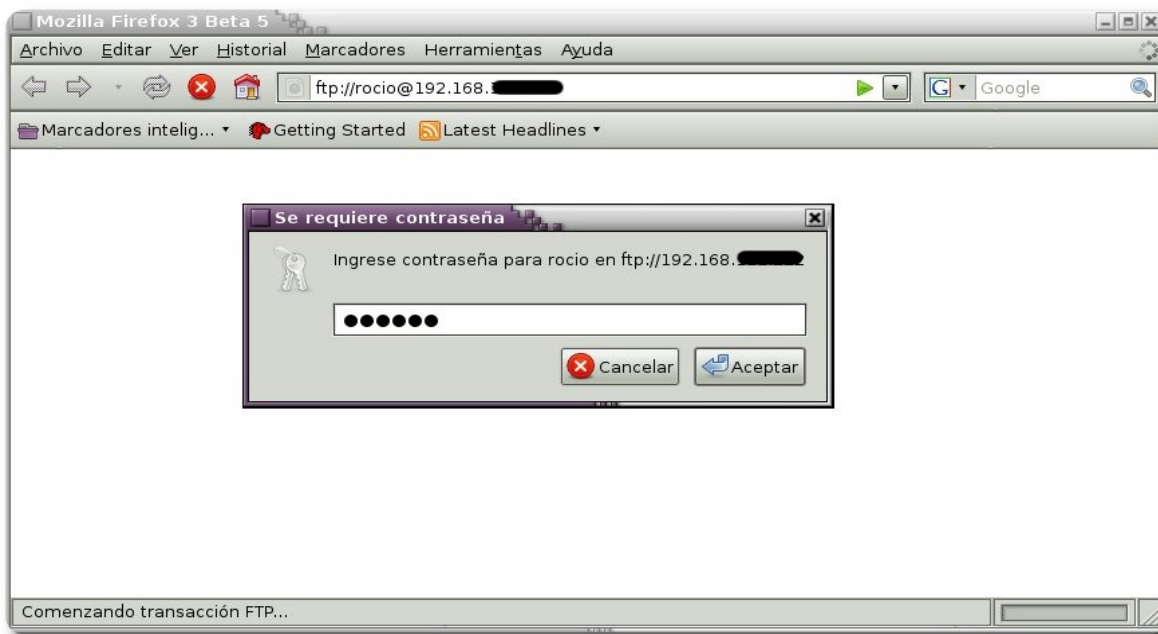
rocio@laboratorio: ~
Archivo Editar Ver Terminal Solapas Ayuda
rocio@laboratorio:~$ ftp 192.168.
Connected to 192.168.
220 Bienvenido al Servidor del Laboratorio de Telecomunicaciones,
Name (192.168.:rocio):
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
    
```

Conexión y operaciones desde programas clientes

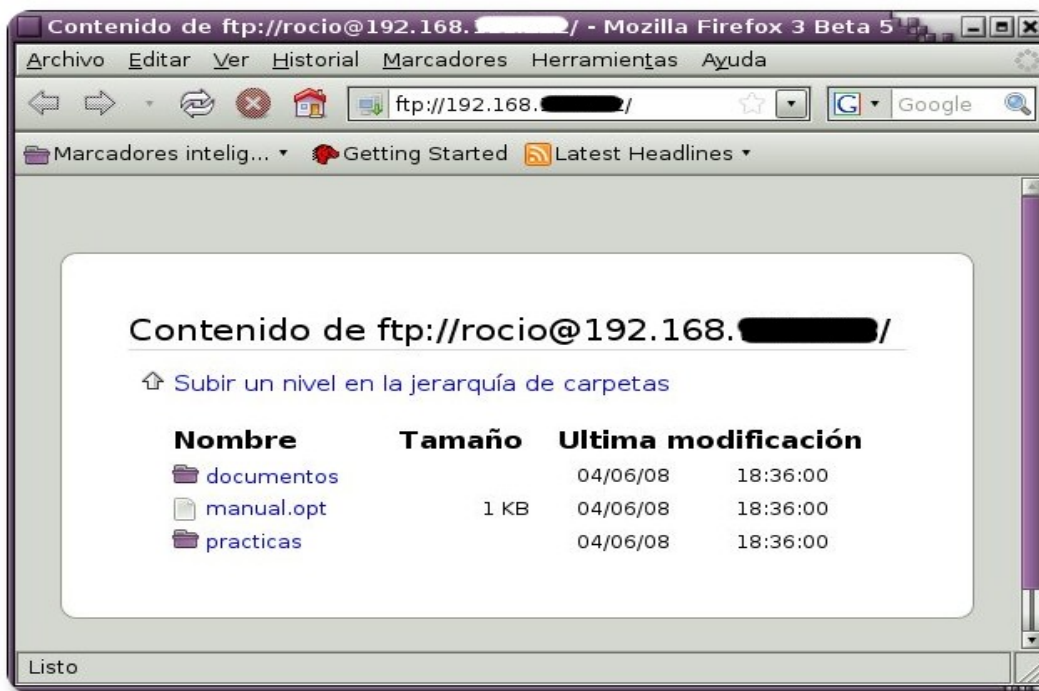
Actualmente disponemos de una gran variedad de programas cliente para el servicio de ftp, que nos permiten acceder a este servicio de forma sencilla e intuitiva mediante un front-end, por ejemplo gftp



También es posible conectarse mediante la mayoría de los browsers.



Donde una vez autenticados, podremos acceder al contenido de directorio perteneciente al usuario.



Script de automatización

Dentro de la administración de sistemas, frecuentemente es preciso contar con scripts que permitan automatizar la transferencia de archivos dadas ciertas condiciones o bajo ciertos intervalos de tiempo, para esto podemos hacer uso de las conexiones al servicio de ftp y las instrucciones de transferencia desde un script como muestra el ejemplo:

```
fichero conecta  
open 192.168.x,y  
user login password  
prompt  
bin  
put archivo.ext  
bye
```

y para invocarlo: `ftp -n < conecta`

Usuarios virtuales

VSFTPD tiene la posibilidad de brindar acceso a usuarios virtuales (usuarios que no son usuarios del sistema) que pueden ser almacenados en un base de datos por ejemplo en mysql, esto es posible haciendo uso de las librerías que permiten esta

interacción (libpam-mysql).

Agradecimiento

Un agradecimiento muy especial al director de ésta revista, Ing. Esteban Saavedra, quién frecuentemente va promoviendo actividades referidas al Software Libre y la investigación, y quién me dió la oportunidad y guió en la realización de este artículo, que si bien es básico pero representa mi primera incursión en la tarea de colaborar con la comunidad de Software Libre.

Referencias

[1] <http://vsftpd.beasts.org/>

Autor



Rocio Figueroa

Estudiante de Ingeniería en

Telecomunicaciones

rocios.figueroa@gmail.com

Openssh la magia de la administración remota

El brindar la posibilidad de administrar un equipo de forma remota se ha convertido una importante necesidad, ya que por medio de ésta, es posible realizar una serie de tareas propias de la administración y gestión de sistemas.



Introducción

La tarea de administrar un sistema, implica poder accederlo tanto de forma local (insitu) y de forma remota, esta última debe contar con la mayor seguridad posible para evitar que la transferencia y accesos sean comprometidos.

Openssh se ha convertido en la mejor opción para realizar la administración y gestión remota.

Por que Openssh

- ✓ **OpenSSH** es una implementación de protocolo SSH de OpenBSD.
- ✓ **Openssh** reemplaza rlogin y rsh, para proporcionar comunicaciones encriptadas seguras entre dos hosts en una red

Características

- ✓ Permite la administración remota
- ✓ Permite el intercambio de ficheros de forma segura y cifrada.
- ✓ Posibilita funciones como administración gráfica, túneles . . .

Que necesitamos para utilizarlo

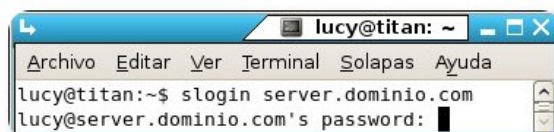
Todas las distribuciones actualmente utilizadas ya incluyen a **Openssh**, en caso que no sea así, podemos proceder a instalarlo mediante el gestor de paquetes predefinido en cada distribución.

Manejo

Un fenómeno común, es que la mayoría de los usuarios generalmente solo se limitan a hacer uso de cierto servicio en su forma más básica, sin considerar que existe muchas opciones y virtudes que pueden ser aprovechadas, en este artículo comenzaremos describiendo el uso básico de **Openssh**, y que gradualmente iremos descubriendo varias de las características desconocidas por la mayoría de los usuarios.

Establecer conexión

Las formas más comunes para iniciar una sesión remota son detalladas en las siguientes figuras:



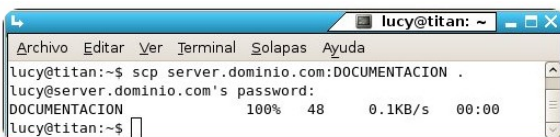


Ejecución comando remotos



Copia de archivos

Otra de las tareas que frecuentemente se realiza es la copia y transferencia de archivos, para esto podemos hacer uso de comandos como scp y sftp que vienen acompañadas de ciertos parámetros utilizados bajo ciertas circunstancias.



Copia recursiva



sftp

Permite la transferencia de archivos de forma segura.



Configuración de seguridad del servidor

Siempre es recomendable cambiar las opciones por defecto que trae consigo cualquier servicio y adecuarlas a nuestras necesidades de funcionamiento o de seguridad. **OpenSSH** dispone de una serie de parámetros de configuración que pueden ser habilitados o deshabilitados dentro de su fichero de configuración (`/etc/ssh/sshd_config`).

Acceso por cierta interfaz

Muchas veces nos encontramos con un servidor que posee más de una interfaz de red, por temas de seguridad es necesario limitar el acceso solo por medio de una interfaz determinada generalmente la (interfaz conectada a la red segura), para esto hacemos uso del parámetro **ListenAddress** `192.168.x.y` que permite especificar que interfaz será la que acepte o escuche las conexiones.

Tiempos de Gracia y número de intentos

Una forma de restringir el tiempo máximo que dispone un usuario para loguearse es hacer uso del parámetro **LoginGraceTime** `10`, también podemos acompañarlo con el parámetro **MaxAuthTries** `2` que determina el número máximo de intentos que dispone un usuario para autenticarse.

Restringir el acceso

En un entorno de producción inmerso en una red pública, es recomendable mantener el control de que usuarios pueden acceder a nuestro servidor y desde donde, estas restricciones se las realiza adecuando los parámetros: **PermitRootLogin** `no`, **AllowUsers** `kathleen, stephanie`, **AllowUsers** `scarlet, lucy@192.168.x.z`; en los cuales no permitimos el acceso como root, permitimos el acceso a los usuarios kathleen, stephanie, scarlet desde cualquier localización y permitimos sólo el acceso al

usuario lucy desde la IP 192.168.x.z.

Restringir número de usuarios

Openssh tiene la posibilidad de restringir el número máximo de usuarios conectados en un determinado instante, todo esto gracias al parámetro **MaxStartups 5** (máximo 5 usuarios concurrentes)

Acceso mediante llaves

Por lo general, accedemos a un servidor remoto por SSH usando la autenticación tradicional (usuario/contraseña de forma interactiva), ya que ésta es la opción por defecto. Esta opción deja el servidor vulnerable a ataques de diccionario o de fuerza bruta, que eventualmente rompen la barrera de la contraseña. Una alternativa a este inconveniente es la autenticación por medio de certificados que permite aumentar la seguridad de nuestra red.

Este proceso incluye básicamente lo siguientes pasos

1. Generación de la llave

```
lucy@titan: ~/.ssh$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/lucy/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/lucy/.ssh/id_rsa.
Your public key has been saved in /home/lucy/.ssh/id_rsa.pub.
The key fingerprint is:
de:a6:5d:e2:1c:99:89:49:a2:4a:a6:e3:ab:76:bb:f0 lucy@titan
lucy@titan: ~/.ssh$
```

ssh-keygen acepta muchos parámetros, como:

- ✓ el tipo de llave, -t dsa
- ✓ el tamaño, -b 2048
- ✓ el fichero donde se almacenará, -f fichero.key
- ✓ un comentario, -C "llave de seguridad"

2. Copiar la llave pública al host remoto

```
lucy@titan: ~/.ssh$ scp -p id_rsa.pub lucy@server.dominio.com:
lucy@server.dominio.com's password:
id_rsa.pub 100% 392 0.4KB/s 00:00
lucy@titan: ~/.ssh$
```

Otra variante para copiar e incluir directamente la llave generada es: **ssh-copy-id** **-i** **~/ssh/llave_oficina.key.pub** **lucy@server.dominio.com**

3. Incluir la llave dentro del conjunto de llaves autorizadas

El archivo **~/ssh/authorized_keys** del home de cada usuario en el servidor remoto almacena las llaves públicas que son utilizadas para el acceso, por tal razón debemos vaciar el contenido de nuestra llave en este archivo.

```
lucy@server: ~$ ssh server.dominio.com
lucy@server.dominio.com's password:
Last login: Thu Jun 5 15:39:02 2008 from 192.168.
[lucy@server ~]$ mkdir -p ~/.ssh
[lucy@server ~]$ chmod 700 ~/.ssh/
[lucy@server ~]$ cat id_rsa.pub >> ~/.ssh/authorized_keys
[lucy@server ~]$ chmod 600 ~/.ssh/authorized_keys
[lucy@server ~]$
```

4. Acceder al host remoto mediante la llave pública

```
lucy@titan: ~$ ssh server.dominio.com
Enter passphrase for key '/home/lucy/.ssh/id_rsa':
```

Uso de passphrase

Comentamos que el uso tradicional de autenticación puede ser vulnerado por técnicas de hacking; mientras que al hacer uso de una clave (publica/privada), es prácticamente imposible ser hackeados por este método, ya que no sólo el hacker tendrá que saber la contraseña sino que tendrá que tener una llave pública válida en el servidor para que pueda ser autenticado. (Claro suponiendo que el servidor nunca haya sido

comprometido y este completamente actualizado y con la mejor seguridad posible).

Encriptar las llaves

En alguna ocasión nos veremos en la necesidad de querer cambiar la frase (passphrase) con la que una llave privada fue encriptada, o en el caso de que la llave privada no estuviese encriptada, querer encriptarla. Para conseguir este objetivo podemos invocar al programa `ssh-keygen` con el parámetro `-p`

Definir un nuevo passphrase

```
lucy@titan: ~/.ssh$ ssh-keygen -p
Enter file in which the key is (/home/lucy/.ssh/id_rsa):
Key has comment '/home/lucy/.ssh/id_rsa'
Enter new passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved with the new passphrase.
lucy@titan: ~/.ssh$
```

Cambiar el passphrase

```
lucy@titan: ~/.ssh$ ssh-keygen -p
Enter file in which the key is (/home/lucy/.ssh/id_rsa):
Enter old passphrase:
Key has comment '/home/lucy/.ssh/id_rsa'
Enter new passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved with the new passphrase.
lucy@titan: ~/.ssh$
```

Memorizar passphrase

Una de las ventajas de emplear llaves públicas frente al uso de contraseñas de usuario es que no tenemos que recordar nada más que una única frase, la frase con la que hemos cifrado nuestra llave privada. Eso es un avance, pero **OpenSSH** dispone de una herramienta que nos puede evitar el trámite de tener que andar escribiendo dicha frase cada vez que establezcamos una nueva conexión. Esta herramienta se llama **ssh-agent** y tiene la capacidad de recordar las llaves privadas que tenemos.

Al ejecutar el agente `ssh` (`ssh-agent`), este crea un socket UNIX y establece la variable de entorno `SSH_AUTH_SOCK` con el nombre del socket. Por razones de seguridad los permisos del socket son ajustados para que tan sólo el usuario actual pueda acceder al socket. Además, el agente también crea la

variable de entorno `SSH_AGENT_PID` y establece su valor con su PID (identificador de programa).

Cuando el cliente de SSH necesita autenticar a un usuario, lo primero que hace es mirar si existe la variable de entorno `SSH_AUTH_SOCK`, de ser así, la usa para establecer una conexión con el agente, el agente no le pasa la llave privada al cliente de SSH, sino que es el propio agente el que se encarga de realizar la autenticación, de forma que la llave privada nunca sea expuesta a los clientes.

Iniciando el agente

Actualmente hay varias formas de cargar el agente, una de ellas es poniendo simplemente **ssh-agent** en la línea de comandos:

```
lucy@titan:~$ ssh-agent
SSH_AUTH_SOCK=/tmp/ssh-hthP110637/agent.10637; export SSH_AUTH_SOCK;
SSH_AGENT_PID=10638; export SSH_AGENT_PID;
echo Agent pid 10638;
lucy@titan:~$
```

Como podrá observarse, sólo basta indicar la frase contraseña una sola vez y por lo que dure la sesión no necesitas indicarla de nuevo, permitiéndote trabajar con tu servidor remoto de manera muy segura.

Añadir llaves al agente

El agente nada más al iniciarse no contiene ninguna llave, para agregarlas se emplea la herramienta `ssh-add`, si se ejecuta sin argumentos intenta añadir los archivos `~/.ssh/id_rsa`, `~/.ssh/id_dsa` y `~/.ssh/identity`:

```
lucy@titan:~$ ssh-add
Enter passphrase for /home/lucy/.ssh/id_rsa:
Identity added: /home/lucy/.ssh/id_rsa (/home/lucy/.ssh/id_rsa)
lucy@titan:~$
```

Si lo que queremos es añadir otra llave distinta a las tres que busca por defecto, lo único que tenemos que hacer es pasar el archivo con la llave como parámetro de `ssh-add`. Por ejemplo, `ssh-add llave_oficina.key`

Listar las llaves que hay en el agente

```
lucy@titan:~$ ssh-add -l
2048 de:a6:5d:e2:1c:99:89:49:a2:4a:a6:e3:ab:76:bb:f0 /home/lucy/.ssh/id_rsa (RSA)
lucy@titan:~$
```

Cada línea representa una llave. La primera columna muestra el tamaño en bits de la llave, la segunda se corresponde con el fingerprinting de la llave, la tercera columna es el archivo que contiene la llave y la última columna, indica entre paréntesis el tipo de llave que es, RSA o DSA.

También se puede emplear la opción `-L` que mostrará las llaves públicas, lo cual puede ser útil para añadirlas al archivo `authorized_keys` de una máquina remota.

```
lucy@titan:~$ ssh-add -L
ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAQEA6ovICjJfTGzrpyhVfwuyUSTvNBnpIq
Hgh3+46yyZWgnMbzAI+Qn7T+8aU0E1+PSFSj hEG7020gFWHH46wKnumFbh+4rK8My+jq
QhvL1mE3hg/Wel5Wp0YfSprk0etizmnBRNEgxfSoYnSvvoBW1nzV3bw02tNEuM5HtJIm
s0B/ahakDp7tLbzHDXSGLcxdeI2wFIBp1gMGzW5d258rKStLxYqtRvCdM27dEKHKUIy
FNbuNLFPVwCFXmouU3H4RkuqyippqgsZV4tS2oE85aAdcFOHkUg+HvR3z9n811Tcqs0T
xk+RKQB7GEmyu8k+0umoXr6P7NkzFeEK9/edMQ== /home/lucy/.ssh/id_rsa
lucy@titan:~$
```

Eliminar llaves del agente

```
lucy@titan:~$ ssh-add -d
Identity removed: /home/lucy/.ssh/id_rsa (/home/lucy/.ssh/id_rsa.pub)
lucy@titan:~$
```

También podemos eliminar todas las llaves que tenga el agente empleando la opción `-D`.

```
lucy@titan:~$ ssh-add -D
All identities removed.
lucy@titan:~$
```

Bloquear el agente

Cuando no necesitamos usar el agente, pero no queremos que se olvide de nuestras llaves, como en el caso de dejar el equipo mientras salimos, podemos optar por bloquear el agente, para ello hacemos uso de `ssh-add -x`, acompañado de una contraseña de bloqueo.

```
lucy@titan:~$ ssh-add -x
Enter lock password:
Again:
Agent locked.
lucy@titan:~$
```

Y cuando necesitamos volver a usar el agente, podemos desbloquearlo ejecutando la herramienta `ssh-add -x`, acompañada de la contraseña de bloqueo

```
lucy@titan:~$ ssh-add -x
Enter lock password:
Agent unlocked.
lucy@titan:~$
```

Tiempo de vida para llaves

Podemos hacer que el agente recuerde las llaves durante un tiempo determinado. Por defecto, el agente no olvida las llaves mientras dure la sesión o no se fuerce a olvidarlas.

```
lucy@titan:~$ ssh-add -t 30
Enter passphrase for /home/lucy/.ssh/id_rsa:
Identity added: /home/lucy/.ssh/id_rsa (/home/lucy/.ssh/id_rsa)
Lifetime set to 30 seconds
lucy@titan:~$
```

Acceso mediante varias llaves

Muchas veces puede darse el caso que disponemos de varias llaves para autenticarnos, los motivos pueden ser diversos dependiendo de algunas restricciones o la posibilidad de ejecutar una o varias configuraciones en función de la llave.

Para conectarnos mediante una determinada llave podemos hacer uso de:

```
lucy@server:~$ ssh -i llave_oficina.key server.dominio.com
Enter passphrase for key 'llave_oficina.key':
Last login: Thu Jun 5 16:55:30 2008 from 192.168.
lucy@server ~]$
```

Restricción de acceso en el uso de llaves

Como vimos anteriormente, **OpenSSH** tiene la posibilidad de restringir el acceso a ciertos usuarios y desde ciertas localizaciones, pero

éstas sólo son válidas si hacemos uso de una autenticación tradicional (usuario/contraseña); en el caso de autenticarnos solo mediante el uso de llaves, también podemos proveer ciertas restricciones, las cuales deben ser incluidas directamente en el archivo `~/.ssh/authorized_keys` del servidor remoto.

Algunos aspectos a considerar de la estructura de este archivo, es que cada llave se encuentra en una sola línea; las restricciones de acceso deben ser incluidas al inicio de la línea que identifica la llave correspondiente; las opciones se separan entre sí utilizando comas y de la clave usando espacios, por ejemplo:

```
from="*.universidad.edu.bo,  
192.168.20.2" ssh-dss AAAAB3Nz...
```

Referencias

[1] <http://www.openssh.org>

Autor



Lucy Ayarde Romero

Consultor de IT

lucyayarde@gmail.com

Bazaar la herramienta para el control de versiones de forma distribuida

Un sistema de control de versiones es el encargado de la gestión de los diversos cambios que se realizan sobre los elementos dentro el desarrollo de un producto software o en las configuraciones del mismo.



Bazaar

Introducción

Los sistemas de control de versiones , en la actualidad se han convertido en un elemento indispensable en el proceso de desarrollo de un producto software; de igual forma se convierten en un aliado importante de la ingeniería de software en la tarea de conseguir como resultado un producto de software de calidad.

Los sistemas de control de versiones desde sus inicios fueron utilizados principalmente en la industria informática, pero hoy en día estos sistemas son utilizados tanto por industrias como por desarrolladores individuales, considerando que estos últimos le dieron más dinamismo a su desarrollo y evolución.

Los sistemas de control de versiones se constituyen en un buen complemento a los sistemas de backups.

Características de un sistema de control de versiones

- ✓ **Gestionar el almacenamiento** de cada uno de los elementos del proyecto.

- ✓ Llevar un **historial de los cambios** en cada elemento del proyecto y anotar el autor de los cambios. Cada uno de los cambios se denomina revisión.
- ✓ Posibilidad de **añadir, borrar, mover o editar** los elementos.
- ✓ **Capacidad de gestionar ramas** de desarrollo paralelas a la principal.
- ✓ **Gestión de conflictos**, en el caso de que más de un usuario cambie un elemento del proyecto.
- ✓ **Generación de informes** de estado, donde se muestren las diferencias entre distintas revisiones.

Clasificación

- ✓ **Centralizados:** existe un repositorio centralizado de todo el código, del cual es responsable un único usuario (o conjunto de ellos). Se facilitan las tareas administrativas a cambio de reducir la potencia y flexibilidad, pues todas las decisiones fuertes (como crear una nueva rama) necesitan la aprobación del responsable.
- ✓ **Distribuidos:** se aumenta la capacidad de decisión distribuida, característica que dará mayor flexibilidad en el uso, pero en algunos casos puede dificultar la sincronización.

Funcionamiento

Todos los sistemas de control de versiones se basan en disponer de un repositorio, que es el conjunto de información gestionada por el sistema. Este repositorio contiene el historial de versiones de todos los elementos gestionados.

Cada uno de los usuarios puede crearse una copia local duplicando el contenido del repositorio para permitir su uso. Es posible duplicar la última versión o cualquier versión almacenada en el historial. Este proceso se suele conocer como check out.

Al momento de realizar modificaciones en la copia local existen dos alternativas:

- ✓ **Exclusivos:** para poder realizar un cambio es necesario marcar en el repositorio el elemento que se desea modificar y el sistema se encargará de impedir que otro usuario pueda modificar dicho elemento.
- ✓ **Colaborativos:** en el que cada usuario se descarga la copia, la modifica y el sistema automáticamente mezcla las diversas modificaciones. El principal problema es la posible aparición de conflictos que deban ser solucionados manualmente o las posibles inconsistencias que surjan al modificar el mismo fichero por varias personas no coordinadas. Esta semántica no es apropiada para ficheros binarios.

Tras realizar la modificación es necesario actualizar el repositorio con los cambios realizados. Habitualmente este proceso se denomina commit, check in.

Conceptos a considerar al momento de utilizar un control de versiones

- ✓ **Repositorio:** Es el lugar en el que se almacenan los datos actualizados e históricos, a menudo un servidor (proceso centralizado), el PC de los desarrolladores (proceso distribuido)

- ✓ **Módulo:** Conjunto de directorios y/o archivos dentro del repositorio que pertenecen a un proyecto común.
- ✓ **Rótulo:** Identificador asignado a un módulo o fichero, asignado en un momento determinado, orientado a una búsqueda posterior.
- ✓ **Revisión:** Una revisión es una versión determinada de un archivo.
- ✓ **Línea base:** Una revisión aprobada de un fichero, a partir del cual se pueden realizar cambios subsiguientes.
- ✓ **Injertar rama o branch:** Un módulo puede ser branched o bifurcado en un momento dado, de forma que, en adelante, dos copias de esos ficheros puedan ser desarrolladas a diferentes velocidades o de diferentes formas, de modo independiente.
- ✓ **Check-out:** Permite crear una copia de trabajo local desde el repositorio. Se puede especificar una revisión específica, por defecto se suele obtener la última.
- ✓ **Commit o check-in:** Los cambios realizados localmente son escritos o integrados sobre el repositorio.
- ✓ **Conflicto:** Un conflicto ocurre cuando el sistema es incapaz de fusionar los cambios, algunas veces precisa intervención manual.
- ✓ **Cambio:** Un cambio representa una modificación específica a un fichero bajo control de versiones.
- ✓ **Lista de cambios:** Son listas que identifican una serie de cambios a realizarse en un solo commit. Útil al momento de revisar un cambio a partir de un identificador.
- ✓ **Exportación:** Similar a un check-out, crea un árbol de directorios limpio sin los metadatos de control de versiones presentes en la copia de trabajo. Se utiliza a menudo de forma previa a la publicación de los contenidos.
- ✓ **Importación:** Una importación es la acción de copia un árbol de

directorios local (que no es en ese momento una copia de trabajo) en el repositorio por primera vez.

- ✓ **Integración o fusión:** Una integración o fusión une dos conjuntos de cambios sobre un fichero o un conjunto de ficheros en una revisión unificada de dicho fichero o ficheros.
- ✓ **Actualización:** Integra los cambios que han sido hechos en el repositorio (por ejemplo por otras personas) en la copia de trabajo local.
- ✓ **Copia de trabajo:** La copia obtenida desde el repositorio en el área de trabajo local, se la realiza en un momento del tiempo o revisión específica
- ✓ **Congelar:** Representa permitir los últimos cambios (commits) para solucionar las fallas a resolver en una entrega (release) y suspender cualquier otro cambio antes de una liberación, con el fin de obtener una versión consistente.

Qué es Bazaar?

Bazaar es un sistema de control de versiones distribuido, de origen open source.



Por qué Bazaar?

- ✓ **Adaptable:** Orientado a la mayoría de los trabajos de desarrollo de software (local y/o remoto).

- ✓ **Amigable:** Orientado principalmente a personas que recién empiezan a hacer uso de este tipo de herramientas, enfocado principalmente a la usabilidad y eficiencia.
- ✓ **Elegante:** Soporta el renombrado de archivos y directorios
- ✓ **Rápido:** Permite obtener rápidamente sus ventajas y características sin afectar al proceso de desarrollo.
- ✓ **Eficiente:** Su almacenaje utiliza un formato altamente eficiente y limpio.
- ✓ **Ligero:** No precisa tener un servidor dedicado
- ✓ **Extensible:** Posee una arquitectura basada en plugins, aspecto que permite dotarle de mayores características específicas y acorde a cada necesidad.
- ✓ **Embebido:** Característica que le permite estar presente en una serie de aplicaciones y servicios libres y/o comerciales.
- ✓ **Seguro:** Al margen de las características que posee por si mismo, cuenta con el patrocinio de Canonical tanto para su desarrollo como para su soporte
- ✓ **Libre:** Disponible bajo licencia GPL.

Beneficios de un sistema de control de versiones distribuido

Trabajo personal

- ✓ Posibilidad de revertir una revisión previa
- ✓ Facilitar la revisión de nuestro propio código (antes del commit)
- ✓ Sincronización de múltiples máquinas

Desarrollo colaborativo

- ✓ Un desarrollador puede trabajar sin interferir con el trabajo de otros desarrolladores
- ✓ El proceso de fusión es automatizado en su generalidad.
- ✓ Instalación

Que precisamos

Bazaar es un software que puede ser instalado en varias plataformas mediante sus respectivos gestores de paquetes:

- ✓ Debian, Ubuntu (apt-get)
- ✓ RHEL, CentOS, Fedora (yum)
- ✓ Windows, posee un instalador
- ✓ Mac OS X (MacPorts).

Identificación Inicial

Antes de comenzar a trabajar es recomendable identificarse, haciendo conocer nuestro nombre y email:

```

jessaavedra@zeus: ~
jessaavedra@zeus:~$ bazaar whoami "Esteban Saavedra <jessaavedra@opentelematics.org>"
jessaavedra@zeus:~$
    
```

Verificación de Identificación

Esto permitirá modificar el archivo de configuración, modificación que puede ser verificada de la siguiente forma:

```

jessaavedra@zeus: ~
jessaavedra@zeus:~$ bazaar whoami
Esteban Saavedra <jessaavedra@opentelematics.org>
jessaavedra@zeus:~$
    
```

Inicializar el control de versiones

Inicializamos el control de versiones de nuestro directorio

Poniendo a prueba a Bazaar

Para lograr un mejor entendimiento, de todas las características y funcionalidades que dispone **Bazaar**, basaremos nuestro ejemplo considerando el siguiente directorio y contenido:

```

jessaavedra@zeus: ~
jessaavedra@zeus:~$ tree proyecto
proyecto
|-- cuadrado.py
|-- factorial.py
-- practicas
    -- serie.py

1 directory, 3 files
jessaavedra@zeus:~$
    
```

```

jessaavedra@zeus: ~/proyecto
jessaavedra@zeus:~/proyecto$ bazaar init
jessaavedra@zeus:~/proyecto$
    
```

Esto ha hecho que **Bazaar** cree un branch donde almacenará los ficheros y el histórico de revisión de los mismos. (crea un directorio oculto con una serie de subdirectorios y archivos que le permiten mantener el control y registro de archivos y tareas realizadas, el contenido es mostrado en la figura siguiente)

```

jesaavedra@zeus: ~/proyecto
Archivo Editar Ver Terminal Solapas Ayuda
jesaavedra@zeus:~/proyecto$ tree .bzz/
.bzz/
|-- README
|-- branch
|   |-- branch.conf
|   |-- format
|   |-- last-revision
|   |-- lock
|   |-- tags
|-- branch-format
|-- branch-lock
|-- checkout
|   |-- conflicts
|   |-- dirstate
|   |-- format
|   |-- lock
|-- repository
|   |-- format
|   |-- indices
|   |-- lock
|   |-- obsolete_packs
|   |-- pack-names
|   |-- packs
|   |-- upload

```

11 directories, 11 files
jesaavedra@zeus:~/proyecto\$

Poniendo ficheros bajo el control de versiones

Ahora es momento de ir añadiendo los ficheros y directorios de los cuales queremos mantener un control de versiones.

```

jesaavedra@zeus: ~/proyecto
Archivo Editar Ver Terminal Solapas Ayuda
jesaavedra@zeus:~/proyecto$ bzz add
added cuadrado.py
added factorial.py
added practicas
added practicas/serie.py
jesaavedra@zeus:~/proyecto$

```

Registrando los cambios

Bazaar nos permite realizar este registro tanto a nivel global (todo el árbol de directorios) como a nivel específico (solo ficheros determinados)

```

jesaavedra@zeus: ~/proyecto
Archivo Editar Ver Terminal Solapas Ayuda
jesaavedra@zeus:~/proyecto$ bzz commit -m "Importacion Inicial"
Committing to: /home/jesaavedra/proyecto/
added cuadrado.py
added factorial.py
added practicas
added practicas/serie.py
Committed revision 1.
jesaavedra@zeus:~/proyecto$

```

Deshacer cambios

Esta característica puede darse en tres casos distintos:

Desversionando un archivo

```

jesaavedra@zeus: ~/proyecto
Archivo Editar Ver Terminal Solapas Ayuda
jesaavedra@zeus:~/proyecto$ bzz remove cubo.py
deleted cubo.py
jesaavedra@zeus:~/proyecto$

```

Deshacer el último commit

```

jesaavedra@zeus: ~/proyecto
Archivo Editar Ver Terminal Solapas Ayuda
jesaavedra@zeus:~/proyecto$ bzz uncommit
5 Esteban Saavedra 2008-06-04
se añadio cubo.py

The above revision(s) will be removed.
Are you sure [y/N]? y
jesaavedra@zeus:~/proyecto$

```

Deshacer todos los cambios desde el último commit:

El deshacer los cambios pueden ser globales o de forma específica de algún archivo.

```

jesaavedra@zeus: ~/proyecto
Archivo Editar Ver Terminal Solapas Ayuda
jesaavedra@zeus:~/proyecto$ bzz revert
+N cubo.py
jesaavedra@zeus:~/proyecto$

```

Realizando cambios en nuestros archivos

Ahora procederemos a realizar algunos cambios en alguno de nuestros archivos (cuadrado.py) y verificar las diferencias con la versión registrada.


```

jesaavedra@zeus: ~/proyecto
Archivo Editar Ver Terminal Solapas Ayuda
jesaavedra@zeus:~/proyecto$ bzipdiff
=== modified file 'cuadrado.py'
--- cuadrado.py 2008-06-03 20:24:03 +0000
+++ cuadrado.py 2008-06-03 20:26:03 +0000
@@ -1,5 +1,5 @@
#!/usr/bin/python
def cuadrado(x):
    return x*x
- print cuadrado(5)
+ print cuadrado(4)
jesaavedra@zeus:~/proyecto$
    
```

Visualizando el estado de archivos

```

jesaavedra@zeus: ~/proyecto
Archivo Editar Ver Terminal Solapas Ayuda
jesaavedra@zeus:~/proyecto$ bzipstatus
modified:
cuadrado.py
jesaavedra@zeus:~/proyecto$
    
```

Una vez que los cambios se hayan realizado podemos proceder a registrar el cambio en nuestro branch mediante un commit, es recomendable incluir un mensaje descriptivo en cada commit.

Visualizando el historial de revisiones

```

jesaavedra@zeus: ~/proyecto
Archivo Editar Ver Terminal Solapas Ayuda
jesaavedra@zeus:~/proyecto$ bzipcommit cuadrado.py -m "modificado el valor"
Committing to: /home/jesaavedra/proyecto/
modified cuadrado.py
Committed revision 2.
jesaavedra@zeus:~/proyecto$
    
```

```

jesaavedra@zeus: ~/proyecto
Archivo Editar Ver Terminal Solapas Ayuda
jesaavedra@zeus:~/proyecto$ bziplog
-----
revno: 2
committer: Esteban Saavedra <jesaavedra@opentelematics.org>
branch nick: proyecto
timestamp: Tue 2008-06-03 16:24:03 -0400
message:
  modificado el valor
-----
revno: 1
committer: Esteban Saavedra <jesaavedra@opentelematics.org>
branch nick: proyecto
timestamp: Tue 2008-06-03 16:18:40 -0400
message:
  Importacion Inicial
jesaavedra@zeus:~/proyecto$
    
```

Publicando el branch con sftp

Precisamos tener instalado y configurado un servidor SFTP, que servirá como plataforma de publicación. Asumiremos que nuestro branch será publicado en <http://proyecto.dominio.com/repositorio>:

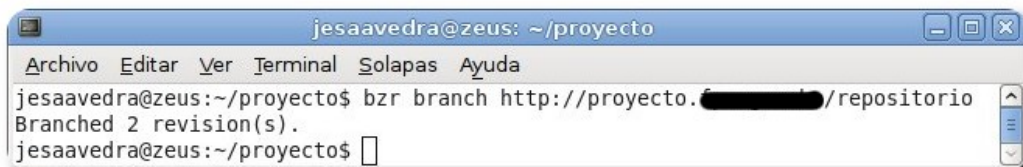
```

jesaavedra@zeus: ~/proyecto
Archivo Editar Ver Terminal Solapas Ayuda
jesaavedra@zeus:~/proyecto$ bzippush sftp://proyecto@proyecto.██████████/www/proyecto/repositorio
proyecto@proyecto.██████████'s password:
Created new branch.
jesaavedra@zeus:~/proyecto$
    
```

Esto hace que **Bazaar** cree el directorio repositorio y dentro de él coloque el branch de nuestro proyecto.

Creando una nueva rama (branch)

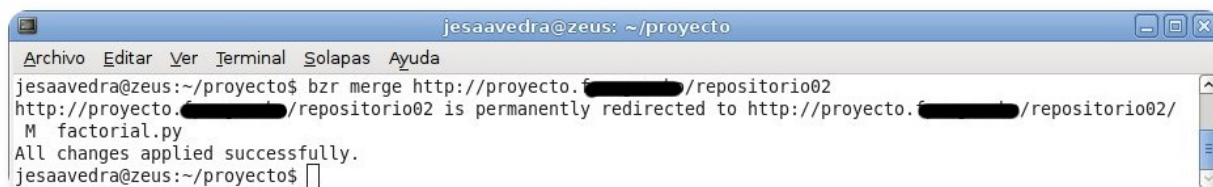
Muchas veces es necesario trabajar sobre ramas publicadas por otros desarrolladores; para esto podemos traer una copia de la publicación que hicimos anteriormente y crear una nueva rama dentro nuestro proyecto.



```
jesaavedra@zeus: ~/proyecto
Archivo Editar Ver Terminal Solapas Ayuda
jesaavedra@zeus:~/proyecto$ bazaar branch http://proyecto.██████████/repositorio
Branched 2 revision(s).
jesaavedra@zeus:~/proyecto$
```

Combinando ramas

Dentro del desarrollo podemos contar con varias ramas, cada una con ciertas modificaciones; en caso que quisiéramos combinar nuestra copia de trabajo con una rama en especial, podemos hacer un merge de la siguiente forma.



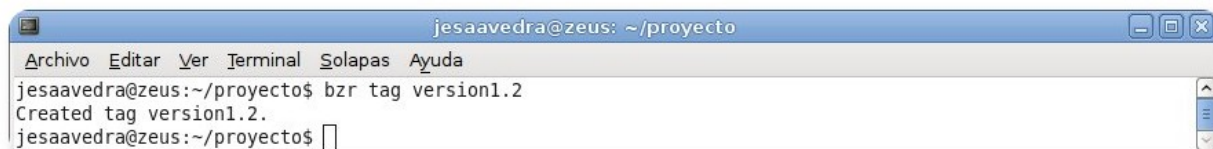
```
jesaavedra@zeus: ~/proyecto
Archivo Editar Ver Terminal Solapas Ayuda
jesaavedra@zeus:~/proyecto$ bazaar merge http://proyecto.██████████/repositorio02
http://proyecto.██████████/repositorio02 is permanently redirected to http://proyecto.██████████/repositorio02/
M factorial.py
All changes applied successfully.
jesaavedra@zeus:~/proyecto$
```

En caso de desear registrar estos cambios en nuestro repositorio local deberemos hacer un commit.

Distribuir nuestros proyectos

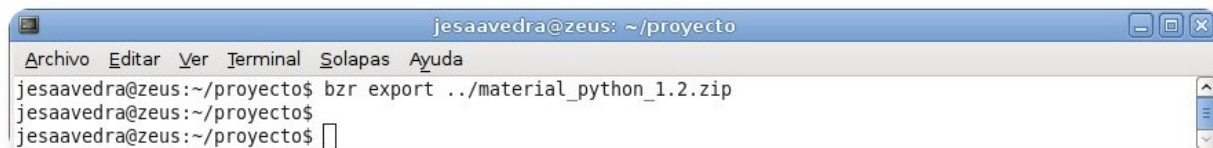
Antes de empezar a distribuir los archivos de nuestro desarrollo, podemos etiquetarlos para mantenerlo en el registro y posteriormente empaquetarlos.

Etiquetado



```
jesaavedra@zeus: ~/proyecto
Archivo Editar Ver Terminal Solapas Ayuda
jesaavedra@zeus:~/proyecto$ bazaar tag version1.2
Created tag version1.2.
jesaavedra@zeus:~/proyecto$
```

Empaquetado

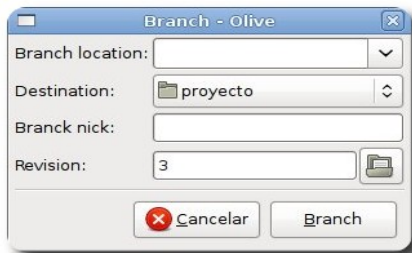


```
jesaavedra@zeus: ~/proyecto
Archivo Editar Ver Terminal Solapas Ayuda
jesaavedra@zeus:~/proyecto$ bazaar export ../material_python_1.2.zip
jesaavedra@zeus:~/proyecto$
jesaavedra@zeus:~/proyecto$
```

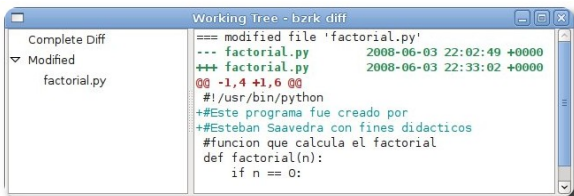
GUIs de ayuda

Bazaar dispone de una serie de GUIs que permiten realizar varias de las tareas de forma gráfica.

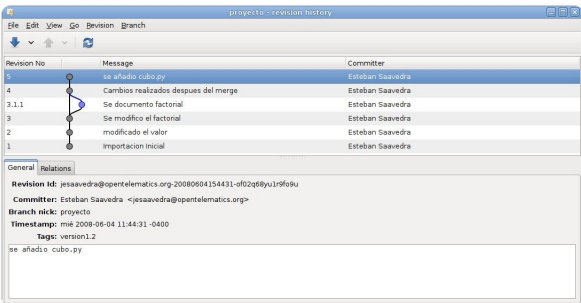
Gbranch



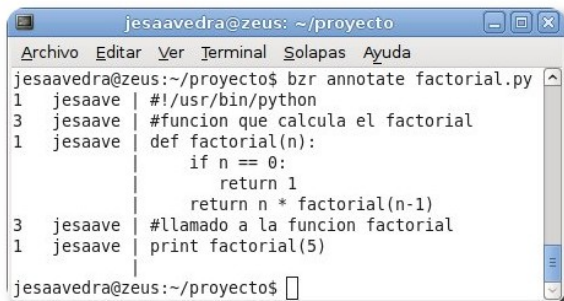
Gdiff



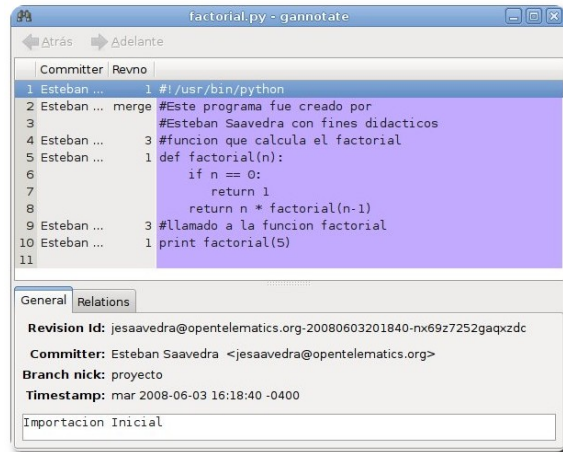
Visualize



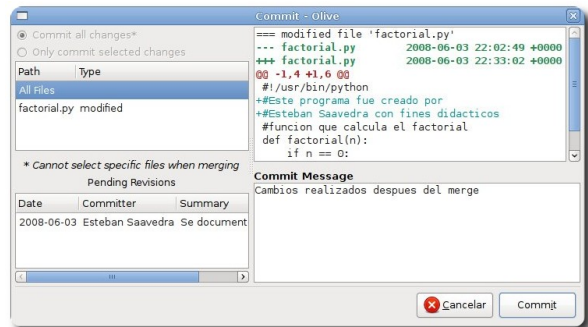
Anotate



Gannotate



Gcommit



Referencias

- [1] <http://bazaar-vcs.org/>
- [2] <http://bazaar-vcs.org/Documentation>

Autor



Esteban Saavedra López

Líder de la Comunidad ATIX (Oruro – Bolivia)
 Activista de Software Libre en Bolivia
jesaavedra@opentelematics.org
<http://jesaavedra.opentelematics.org>

La Entrevista



Alexandre Oliva

Fundación Software Libre América Latina



1. ¿Cuáles fueron las principales motivaciones para conformar la FSFLA?

a. Formar parte de la red de **FSFes**, una red de "tolerancia a fallas" de organizaciones independientes y autónomas, que colaboran, cada una en su región, en la misma misión: **defender las libertades y los derechos de usuarios y desarrolladores de software, para que, si un día ocurre algún problema con alguna de las FSFes, que la misión siga cumpliéndose.**

b. Actuar en esta región, bajo los principios, ideales y filosofía del Software Libre y la misión de las **FSFes**, en los idiomas locales, con actividades independientes y en cooperación con actividades globales de otros miembros de la red.

2. ¿Cómo está constituida la estructura de miembros de la FSFLA?

Los miembros de **FSFLA** forman su consejo, y son juntos responsables por los actos y decisiones de la organización. **FSFLA** es una democracia directa: las decisiones, incluso las de delegar poder limitado de decisión a otros, son tomadas en el consejo, normalmente por consenso.

El consejo aún cuenta con la participación de observadores invitados, tanto de otras organizaciones hermanas, como de activistas de **Software Libre en América Latina**, que se están involucrando con **FSFLA**.

3. ¿Cuál el criterio para aceptar o invitar nuevos miembros en la FSFLA?

No hay una fórmula que garantice volverse un miembro de **FSFLA**, es decir, no hay condiciones suficientes, pero hay condiciones necesarias, como un compromiso claro con la filosofía del Software Libre, interés y disponibilidad para trabajar para avanzar la misión de **FSFLA**, y aprobación del consejo.

Muchos de éstos criterios son subjetivos, y mensurables solamente con interacción en un largo tiempo. Quién quiera involucrarse con **FSFLA** puede hacerlo contactando directamente a los miembros, pero la mejor manera de hacerlo es participar en las actividades y equipos de **FSFLA**, o proponer nuevas actividades. Nuestra página de ¡Ayuda! en www.fsfla.org presenta nuestros proyectos, campañas, equipos de trabajo y formas de contacto. Si participas, los miembros pueden percibir que tienes interés y disponibilidad, y si compartes la filosofía.

Antes de invitar alguien a volverse miembro, normalmente invitamos a que la persona se vuelva observador, así que la persona puede empezar a participar en las decisiones y discusiones estratégicas internas y conocer cómo funcionamos internamente antes de decidir si quiere asumir el compromiso de volverse co-responsable por la organización, mientras los consejeros pueden conocer mejor aún al observador y tener más información para decidir en los criterios subjetivos.

Por otro lado, hay cuestiones que pueden dificultar que alguien se vuelva miembro. Por ejemplo, **FSFLA** busca mantenerse libre de influencias externas muy fuertes, entonces buscamos no tener una fracción significativa de los miembros bajo las leyes de un mismo país, o trabajando en una misma compañía, o involucrados en una misma organización.

Así que podemos operar, para avanzar nuestra misión sin mucha preocupación si alguna acción va a generar rechazo o amenazas de un empleador de muchos miembros, si van a existir conflictos de interés con otras organizaciones que podrían afectar significativamente nuestras

decisiones, o si miembros podrían ser responsabilizados por protestos de **FSFLA** en contra de leyes injustas en sus países.

4. ¿Cuál el grado de participación y aceptación de la FSFLA dentro las actividades de Software Libre e interacción con las comunidades en América Latina?

Después de un par de años "cerrada" en las pampas (Argentina, Uruguay y sur de Brasil), **FSFLA** buscó una nueva estructura más abierta a la participación de toda la América Latina, y hoy tenemos consejeros en Brasil, Chile, Colombia, Venezuela y México, y observadores latinoamericanos de Argentina, Bolivia, Brasil, Chile, Colombia, Costa Rica, Ecuador, El Salvador, México, Perú, Uruguay y Venezuela.

A pesar de esto, aún hay mucho que avanzar en establecer contactos positivos con comunidades de Software Libre y las que agregan (o confunden :-)) Software Libre y otros movimientos en América Latina.

De la misma manera que otras **FSFes**, la posición estricta de rechazo al Software no-Libre dificulta algunos contactos en comunidades que no conocen o no aceptan las diferencias entre las filosofías del Software Libre y otras variantes más limitadas de ella, y con las que efectivamente promueven esas otras filosofías y rechazan la nuestra, aunque sigan llamándola de Software Libre. Esto es un tema delicado, pero nuestra misión requiere que sigamos buscando superar estas dificultades, para lograr promover la libertad de usuarios y desarrolladores de software.

Asimismo, nos parece que **FSFLA** es bienvenida por la gente que cree en la motivación social, ética y moral del movimiento de Software Libre de la que formamos parte en casi toda América Latina.

Lamentablemente, hay excepciones, tanto gente que se ha opuesto a la creación de **FSFLA**, por principio o por su composición original, como gente con quienes **FSFLA** no ha logrado mantener relaciones positivas.

Creo que oposición inicial, dificultades para superar desconfianzas y para formar equipos distribuidos funcionales son parte del inicio de cualquier organización virtual. Sigo animado para ayudar **FSFLA** a superar estas dificultades. Por supuesto, ayuda, participación y apoyo de comunidades de Software Libre son muy favorables y bienvenidas, no sólo porque éstas relaciones son muy importantes para **FSFLA** para avanzar en su misión, sino también porque cuando algunos más alejados o desconfiados perciban lo que buscamos, vamos a poder acercarnos y trabajar mejor juntos por la libertad digital.

5. ¿Cuál consideras, el mejor y peor acierto que haya tenido la FSFLA?

Creo que nuestro mayor error fue empezar la organización con una mayoría de gente fuertemente involucrada en una misma organización.

Con esto se volvió muy difícil separar las identidades de las organizaciones, sus diferentes objetivos, prioridades, estrategias, alianzas y relaciones, e incluso elegir cómo dividir el tiempo y recursos personales dedicados a cada una de ellas. Concentración y superposición excesiva de intereses, poderes e identidades, veo hoy, fueron un grave error para **FSFLA**, y aún sufrimos las consecuencias.

Pienso que nuestro mayor logro es la campaña en contra de los Softwares de Impuestos, empezada en Brasil para rechazar la obligación, impuesta a ciudadanos por oficinas de gobierno, de requerir el uso de Software no-Libre para preparar declaraciones de impuestos. Hemos trabajado mucho este tema en Brasil desde 2006, cuando **FSFLA** prácticamente se volvió FSF Brasil mientras buscábamos abrirla a toda América Latina con la iniciativa (GNU)², y logramos algunos pasos importantes involucrando cuestiones legales y técnicas, como por ejemplo volver libre el software no-Libre publicado por el gobierno para preparar declaraciones de impuesto de renta en 2007 y 2008. Pero la oficina del

gobierno sigue resistiendo, entonces hay que seguir peleando, y hay mucho que avanzar aún.

6. ¿La FSFLA mantiene alguna relación con algunas organizaciones? ¿De qué tipo es ésta relación?

Somos miembros de la red mundial de **FSFes**, por lo que invitamos como observadores de nuestro consejo a los presidentes de la FSF original, de FSF Europa y de FSF India, así como también de la Iniciativa Software Libre en Japón. Nuestra relación es de cooperar en actividades, divulgación y campañas mundiales, compartir información, discutir estrategias y planes. Cada una de las organizaciones, incluso **FSFLA**, mantiene su autonomía, pero los observadores frecuentemente ofrecen sus opiniones, que han sido de gran ayuda en la formación, dirección y reconstrucción de **FSFLA**.

7. ¿La filosofía del Software Libre se difunde como la FSFLA desea o lo espera, o existe algún obstáculo principal?

Siempre podría ser mejor, ¿no? :-) Pero podría ser mucho mejor.

Hay muchos obstáculos. Los más obvios son la resistencia humana natural al cambio, especialmente a lo que no se conoce, y el poder de los monopolios de seducir, dominar, influenciar, y corromper.

Otros menos obvios son el Síndrome de Estocolmo Digital, en que la víctima del Software no-Libre defiende aquél que la aprisiona, y la difusión de conceptos errados o incompletos acerca de qué es Software Libre (el respeto a las 4 libertades, y no sólo GPL, sólo copyleft, sólo GNU, o sólo GNU/Linux), por qué existe (un movimiento social, ético, moral y político que defiende derechos humanos esenciales) y por qué es importante que toda la gente rechace el Software no-Libre (es un problema social, que daña al usuario y a toda la sociedad).

No nos ayuda mucho la popularidad de otro movimiento, creado para promover el Software Libre en sí (los programas, no la filosofía) por méritos técnicos y económicos, pero también la idea de que el Software no-Libre, que no respeta al usuario ni permite una sociedad solidaria, es aceptable, y elegir Software Libre o no-Libre es sólo una cuestión de libertad de elección, por lo que éste otro movimiento rechaza las cuestiones éticas, morales, de responsabilidad social, y respeto al prójimo.

Lamentablemente mucha gente confunde los dos movimientos, y recibe el mensaje más limitado y conveniente, lo que nos da más trabajo para incentivar a la gente, a realmente buscar la libertad, y explicarles que cuando sacrifican su libertad aceptando y pronto volviéndose dependientes de Software no-Libre, se hieren a sí mismos y a todos, un poco de forma inmediata y mucho más al largo plazo, puesto que cada pequeña dependencia va a ser explorada por los monopolistas para introducir otras dependencias.

Tampoco ayuda que la gente que promueve éste otro movimiento, elija llamar Linux la combinación del sistema operativo GNU, creado para liberar a la gente, con el núcleo (kernel) Linux, creado para poner el sistema operativo GNU, al cual aún hacía falta un núcleo, a funcionar en un i386. Con esta confusión de nombres, hacen parecer que el trabajo más grande o más importante lo hicieron los que pusieron sólo las últimas piedras en la construcción del sistema operativo, y se niega a nosotros no sólo el crédito por nuestro trabajo, sino también lo más importante para nosotros: que la gente conozca por qué el sistema operativo que prueban o usan fue creado. Si más gente llamara a la combinación GNU+Linux o GNU/Linux, más gente buscaría saber qué es el GNU, y entonces conocería la historia y la filosofía del movimiento Software Libre, y no sólo la del otro movimiento, y sabría que el GNU es más importante "y" más socialmente responsable que los pingüinos.

8. Sabemos que el dinero no lo es todo, pero muchas veces sirve de mucho. Si la FSFLA tuviese un financiador o un patrocinador, que financie sus actividades ¿qué tipo de proyectos encararía?

No lo hemos discutido, porque por ahora me parece un escenario muy hipotético.

Pienso que algunos de los primeros pasos serían establecer brazos jurídicos de **FSFLA** en quizá dos o tres países, y poder mantener una estructura más formal, útil para algunas acciones. Establecer nuestro propio servidor en la Internet, para liberar la máquina virtual que nos ofrece nuestra hermana FSFE.

Luego tengamos infraestructura básica independiente; si aún hay dinero, quizá podríamos emplear gente para tareas administrativas y burocráticas, y después activistas del Software Libre para campañas, desarrolladores de Software Libre para desarrollo, pago de viajes para participación en eventos, quizá incluso emplear a activistas del Software Libre interesados en dedicar más tiempo a **FSFLA** y a la causa y menos a pelear por el pago de las cuentas.

Bueno, y ya que estamos hablando de sueños distantes, ¿por qué no pensar en comprar compañías que promueven la cultura de aceptar depender de Software no-Libre y vivir con sus restricciones, empezando por Microsoft y Google, y ponerlas a respetar las libertades de la gente?

9. ¿Qué tiempo estimas que hace falta aún para que el Software Libre predomine sobre el Software propietario?

Veo un progreso muy lento, no sé si voy a vivir para ver el logro final, o si tiene sentido hablar sólo en predominio.

Lo que veo es que crece el uso de Software Libre, pero también crecen las dependencias que quitan libertades impuestas por hardware, las aplicaciones no-Libres que

ejecutan en sistemas operativos Libres y son promovidas muchas veces como si fueran soluciones para los problemas de dependencia.

Otra gran amenaza son las aplicaciones basadas en la Internet, bajo control exclusivo de pocas grandes compañías, que quitan del usuario casi todo el control sobre sus computaciones "y" sus datos, sin importar tanto si el software ejecuta integralmente en los servidores o si parte de él es enviada para ejecución en el navegador del usuario. El problema va más allá de ofertar el código fuente correspondiente bajo licencias de Software Libre, involucrando también el uso (o no) de Estándares Abiertos Libres, garantías de acceso a la información personal y pública, y la posibilidad de duplicar, cambiar y interoperar la infraestructura (la porción que ejecuta en el servidor) como le guste al usuario. Pienso que ya tarda algo como un **GNWWW** manifiesto, o una campaña por la Web Free.0, o algo así, pero lamentablemente desde que empezamos a discutirlo, al final de 2007, no fue posible avanzar mucho en esto.

Además, la misión no estará completa cuando logremos que el Software Libre predomine, sino cuando toda la gente pueda vivir sin aceptar que se le quiten sus libertades y su control acerca de sus propias computaciones, y toda la gente sepa por qué no lo debería aceptar si tuviera la oportunidad. Esto es un esfuerzo permanente de educación social, moral y cívica. **La libertad no se recibe, se le conquista, y para mantenerla el precio es la eterna vigilancia.**

10. ¿Por qué la FSF o la FSFLA no ha propiciado una distribución propia? ¿Consideras que esto ayudaría a comprender de mejor forma los objetivos y la filosofía que persigue?

No veo que crear y mantener una distribución más, sea un buen uso de nuestros escasos recursos. De hecho, aunque tuviéramos

muchos más recursos, no veo ventajas en hacer lo que otros ya hacen bien, incluso en nuestra región.

Me parece que más eficiente que fragmentar, "*competir*" con distribuciones que ya tienen compromiso con nuestra filosofía de libertad es divulgar, apoyar y sumar esfuerzos con ellas. UTUTO y Musix GNU+Linux son desarrolladas de manera independiente en América Latina. gNewSense y BLAG tienen desarrolladores activos en nuestra región. Incentivar desarrolladores de la región a apoyar éstas y otras con el mismo compromiso, como dyne:bolic, GNUstep, GNU Darwin y JNode, me parece mejor que duplicar o fragmentar estos esfuerzos.

Otro esfuerzo importante en que hemos trabajado como parte de nuestra campaña **¡Sé Libre!**, de rescate de los valores fundamentales de la filosofía del Software Libre, de busca por la libertad y rechazo al software que la quita, es el proyecto **linux-libre**, en que quitamos de los paquetes publicados por Linus Torvalds todo el Software no-Libre que él ha aceptado añadir, y que la mayoría de las distribuciones GNU/Linux distribuye, muchas sin ni saber de esto y de los riesgos jurídicos y problemas éticos que esto genera.

<http://www.fsfla.org/mailman/listinfo/linux-libre>

linux-libre ya lo usan gNewSense (Ubuntu sin Software no-Libre), BLAG (Fedora sin Software no-Libre) y dyne:bolic, y es muy probable que las otras distribuciones GNU/Linux lo adopten y se vuelvan en distribuciones **GNU/Linux-libre** :-)

Nos involucramos con **linux-libre** cuando buscábamos ofrecer sus binarios para distribuciones semi-Libres, y que podrían ser usadas sin quitar libertades del usuario tan sencillamente como reemplazando su kernel por **linux-libre**, y evitando la instalación de algunos paquetes.

Así que ya tenemos paquetes binarios de **linux-libre** para freed-ora 7, 8, 9 y rawhide, y buscamos voluntarios para ayudar con freed-ebian, uhurubuntu (uhuru significa libertad en el idioma africano swahili) y otras.

11. En estos tiempos donde las distribuciones abundan, ¿qué características consideras que debiesen ser tomadas en cuenta, de tal forma que coadyuven a la formación y objetivos que persigue la FSFLA?

No distribuir Software no-Libre. No recomendar, sugerir o incentivar al usuario la instalación de Software no-Libre para "mejorar" su sistema. No hacer parecer que Software no-Libre es moralmente aceptable, deseable, o una solución para un problema. El Software no-Libre es parte del problema, cuando no es él mismo el problema.

No sacrificar las libertades esenciales del usuario por un poco de conveniencia. Ya decía **Benjamin Franklin que quienes sacrifican libertades esenciales por conveniencia temporaria no merecen ni libertad ni conveniencia.** Si un usuario cree que necesita usar un Software no-Libre, la víctima debe buscarlo junto al proveedor de este Software no-Libre, no en la comunidad de Software Libre. Y si esto no parece conveniente, está bien: se vuelve más visible el problema del Software no-Libre, para que incluso el usuario no técnico lo entienda más sencillamente.

12. Las continuas migraciones que se van dando en universidades e instituciones en el mundo entero, ¿serán el reflejo de la aceptación del Software Libre como tal, o sólo el reflejo de usar un software sin licencias o de forma gratuita?

Sin duda una combinación de estos dos factores, aunque me gustaría que fuera por que más y más gente aprende a valorar su libertad.

13. ¿Qué impacto consideras que ha tenido el Software Libre en el desarrollo de productos software en general en América Latina?

Aún veo (y lamento) nuestra región más como consumidora pasiva o mera empaquetadora/retocadora (personalización). Por supuesto, usando Software Libre, existe la "posibilidad" de cambiar la situación y volverse en co-productora.

De hecho, hay esfuerzos de desarrollo individuales, gubernamentales y de la iniciativa privada, pero la participación de nuestra región en el desarrollo de proyectos estratégicos de Software Libre aún me parece muy pequeña.

No puedo negar que me quedo un poco entristecido cuando veo una persona decir con orgullo que es "**usuaria**" de GNU/Linux. ¡Muy bien!, el software lo escribimos para volverla Libre. Por supuesto, usarlo y promoverlo nos ayuda, pero ¿esto es todo lo que puede hacer para ayudarnos a hacerlo más y mejor?

No es decir que exista obligación legal o moral de devolver a la comunidad beneficio equivalente al que ha recibido de ella; esto es imposible. La mayoría incluso de los desarrolladores recibe más de la comunidad que podría imaginar poder contribuir. Pero la regla de oro, el mismo principio moral fundamental que justifica y demanda las 4 libertades de la Definición del Software Libre, un principio respetado en todas las sociedades estables, indica que **algún esfuerzo para contribuir a las comunidades que le ofrecen el Software Libre que usa sería no sólo un regalo deseable y bienvenido, sino también más justo.**

14. ¿Cómo consideras que la comunidad en América Latina podría colaborar con la FSFLA?

Participar y apoyar nuestras campañas y equipos de trabajo es la manera más obvia. Invitaciones para presentar charlas y divulgar la filosofía son bienvenidas. Donaciones de tiempo, esfuerzo o dinero sin duda son colaboraciones que nos gustan.

Pero me parecen colaboraciones más importantes la búsqueda por la propia libertad, la protesta cuando intentan quitarla, la divulgación de la filosofía, el incentivo a que otros busquen y demanden respeto a sus libertades, y el desarrollo de Software Libre que haga tareas útiles para las cuales aún no exista Software Libre, para que más gente pueda elegir la libertad.

Cuanto más gente esté determinada a rechazar productos que no respeten sus libertades, más pequeño será el mercado para estos productos. Así que los proveedores no perderán la libertad de ofrecer estos productos, sino sencillamente van a elegir respetar la libertad de sus clientes para alcanzar un mercado mayor. Así que rechazar productos que no respeten sus libertades se vuelve cada vez más fácil.

Siempre que se acepta un Software no-Libre, o que requiera Software no-Libre, se impide el avance de este ciclo virtuoso, y también se da fuerza al ciclo vicioso de la dependencia. No se va a lograr alcanzar la libertad aceptando cada vez más Software no-Libre en sistemas operacionales semi-Libres. Lo que se lograría es volver los sistemas operacionales semi-Libres más populares pero menos Libres, y llevar más gente a no comprender qué es Software Libre, y a creer que el Software no-Libre es aceptable y incluso indispensable. Pero no es, y no debe ser. Si los usuarios lo aceptan, ni los proveedores de hardware ni los de los sistemas operativos semi-Libres tendrían incentivo para buscar algo mejor, y entonces siguen ofreciéndole, y los usuarios siguen siendo entrenados para aceptarlo.

Promover la filosofía del Software Libre y el Software verdaderamente Libre, la libertad para los usuarios y el rechazo al Software no-

Libre, es la mejor colaboración que se puede ofrecer no sólo a la FSFLA, sino también a sí mismo y a la sociedad.

15. ¿Qué estrategia consideras que se debiese utilizar para conseguir más adeptos al Software Libre?

No me queda claro si hablas del software en sí o del movimiento.

La popularidad del Software Libre en sí no es algo que me parezca un fin importante en sí mismo. Por ejemplo, volver el Software Libre más popular no me parece muy ventajoso si esto se logra promoviendo el sacrificio de las libertades en otros niveles del sistema.

La estrategia para lograr más adeptos para la filosofía, creo que es la educación: enseñar a más y más gente la importancia social y individual de demandar las libertades, y oponer y rechazar las amenazas a ellas. Un efecto colateral de este logro es que el Software Libre en sí también gana más adeptos, motivados por las razones que más dificultan volver a dar pasos en la dirección errada.

Otro efecto positivo es el desarrollo de más Software Libre para los que descubran que necesitan Software Libre que aún no existe y se mantengan fieles a la decisión de no aceptar más Software no-Libre.

16. ¿Qué recomendaría la FSFLA a los gobiernos y universidades de América Latina con respecto al uso y desarrollo de Software?

Gobiernos democráticos y responsables requieren soberanía, transparencia y eficiencia en el uso de los recursos públicos. No es posible ser soberano ni eficiente si estás bajo control de un monopolio, y todo Software no-Libre es un monopolio. No es posible ser transparente si no puedes saber qué hace el software que usas, ni mostrarlo a los ciudadanos.

Gobiernos democráticos y responsables requieren el uso exclusivo de Software Libre.

Escuelas, universidades y otras instituciones que tengan como misión difundir conocimiento, incentivar la búsqueda por conocimiento y promover la ética y el bienestar social no pueden adoptar y promover el uso de software que prohíbe la difusión del conocimiento codificado en sí mismo, limita el conocimiento que se puede difundir con él, impone restricciones no-éticas a sus usuarios, les da un mal ejemplo de conducta social, y vuelve la institución y sus alumnos rehenes de los intereses monopolistas de los proveedores del software y además agentes que pagan para promover estos productos.

Educación con ética requiere el uso exclusivo de Software Libre.

17. ¿Qué opinas sobre la relación entre el software Libre y la investigación?

La ciencia está basada en compartir y sumar conocimiento, y reproducir experimentos. Software no-Libre lo prohíbe.

La investigación en informática no es viable sin permiso para compartir y usar las descubiertas, o para obtener el software utilizado en un experimento.

Además, investigar informática con Software no-Libre vuelve la investigación artificialmente más difícil.

Si fuera posible aprender y verificar las leyes de la física estudiando el "código fuente" del universo, se perdería menos tiempo investigando suposiciones equivocadas y teorías incorrectas, y se podría concentrar en estudiar los efectos útiles de estas leyes.

Si fuera posible aprender y verificar el funcionamiento de un organismo vivo estudiando no sólo el código objeto genético, sino también su código fuente, con documentación, comentarios, estructuras bien organizadas, etc., se podría direccionar mejor los esfuerzos para evitar o curar enfermedades.

Estos códigos fuentes no existen. Pero el código fuente del Software no-Libre que lamentablemente se usa en investigación de informática sí existe, pero los investigadores

están artificialmente privados de acceder a él, o limitados en lo que pueden hacer con los resultados.

Investigación no sólo en informática se beneficia y requiere Software Libre, como cualquier usuario que se vuelve libre de amenazas, dependencias, monopolios, y restricciones. La ciencia requiere libertad para investigar todas las posibilidades viables y publicarlas, no restricciones impuestas a través del software o de demandas incompatibles con la ciencia necesarias para obtenerlo.

18. ¿Qué planes y actividades a futuro tiene previstas la FSFLA?

Destaco una vez más la campaña ¡Sé Libre!, la actividad más importante y permanente que hacemos: educación y divulgación de la filosofía en nuestro sitio, en publicaciones de informática y para el público en general, y desarrollar y divulgar software esencial para que más usuarios puedan elegir la libertad, como **linux-libre**, distribuciones **GNU/Linux-libre**, otras variantes 100% Libres del sistema operativo GNU y otros sistemas operativos 100% Libres.

La campaña en contra de los Softwares de Impuestos debe seguir avanzando, quizá incluso para otros países en América Latina. Me parece que sería útil poner más esfuerzos en los trabajos en Estándares Abiertos Libres y en contra de Gestión Digital de Restricciones (DRM).

Voluntarios para ayudar en éstas y otras de nuestras iniciativas son bienvenidos.

19. Un mensaje para la comunidad de Software Libre en Bolivia.

¡Un saludo a todos los gnudistas de Bolivia! Espero que éste sea el primero de muchos contactos favorables para la libertad de los usuarios y desarrolladores de software en **Bolivia**, que esta discusión aclare y refuerce los conceptos y la importancia de la filosofía del Software Libre para todos, y vuelva más gente interesada en pelear por la libertad, junto con **FSFLA** y en esfuerzos

independientes.

¡Sé Libre!, para intentar, para saber, para poder y para lograr.

20. Tus palabras finales y mensaje para la Revista ATIX.

Desde cuando supe el significado de la palabra **ATIX**, me encantó la idea de una comunidad de Software Libre llamada así. La revista es un gran logro no sólo para la comunidad de **Oruro**, sino para todas las comunidades de Software Libre hispanohablantes. Mis congratulaciones por la iniciativa, y por lo que vi en la primera edición, en la temática y la calidad visual.

Les deseo mucho suceso en la promoción de la filosofía del Software Libre y sus muchas ventajas para la sociedad, en su entorno y en toda la comunidad, y agradezco por la oportunidad de participar en este trabajo que tanto avanza nuestra misión.

¡Muchas gracias!, y **¡Sé Libre!** para escribirme en lxoliva@fsfla.org :-)



Enlaces

- [1] <http://www.fsfla.org>
- [2] <http://www.lsd.ic.unicamp.br/~oliva/>

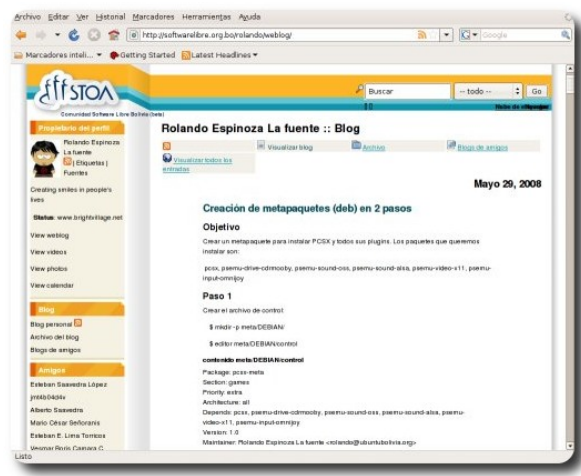
A T I X

N o t i c i a s

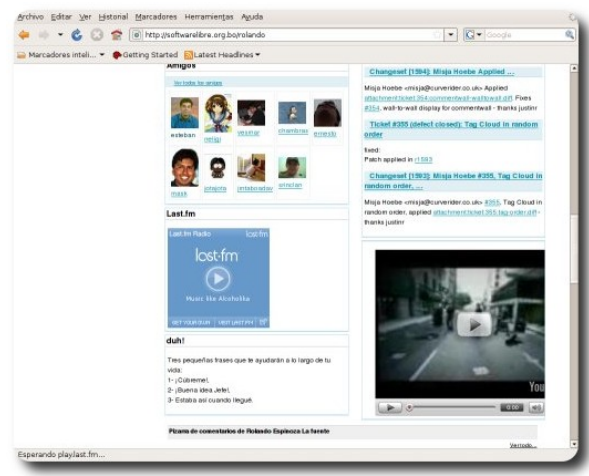


Al fin se ha puesto online la nueva versión del portal de la comunidad de **Software Libre Bolivia**. SLOB es la comunidad que a nivel nacional intenta concentrar a toda la gente para coordinar proyectos, conocimientos entre nosotros, organizar eventos, participar de los foros, etc.

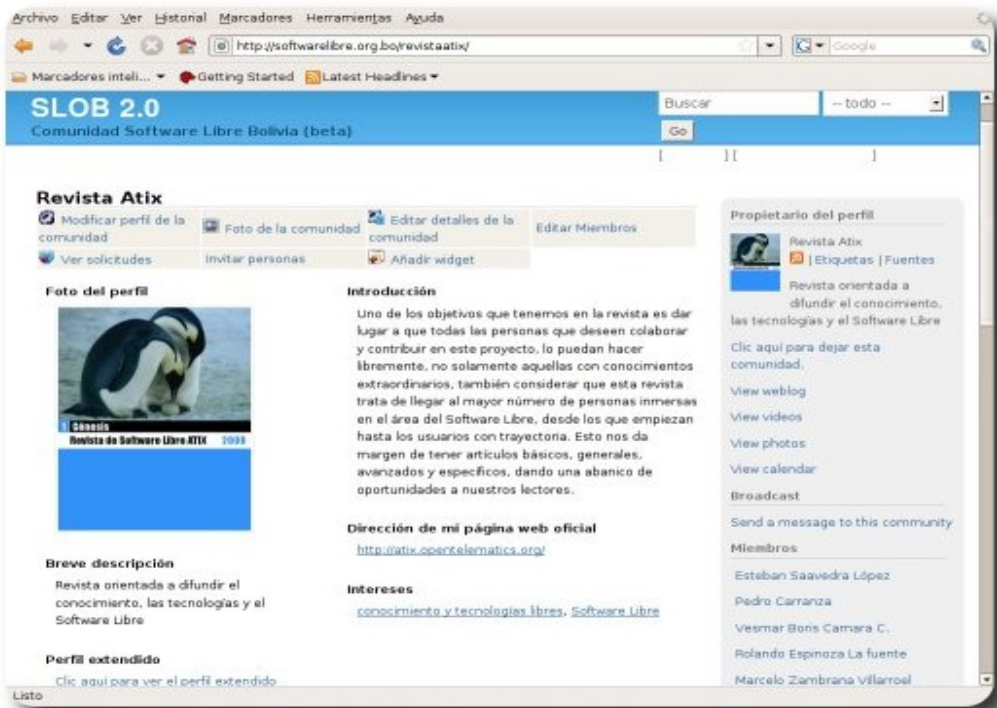
Lo que pretende el nuevo portal es ir un paso más allá de las listas de correo, de facilitar un espacio más ameno y libre para interactuar informalmente. Podrán compartir artículos, blogs, fotos, videos, formar comunidad, hacer amigos, publicar su perfil personal o profesional, formar su red de contactos, etc. **¡no hay límites!**



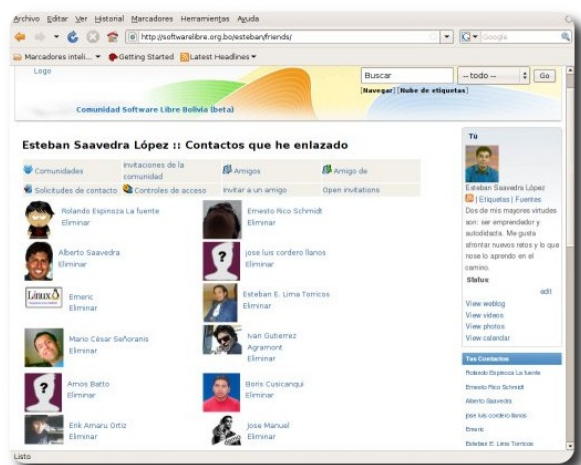
Blog de usuarios, permite que cada usuario registrado disponga de un blog, con la facilidad de enlazarlo con su blog personal si lo tiene en otro sitio Web.



Videos, fotos y mucho más, esta nueva plataforma nos permite incrustar videos. subir fotos, imágenes y poner a disposición de la comunidad y de los visitantes una serie de recursos que pueden ser privados o públicos.



Otra de las facilidades que provee esta nueva plataforma es la de poder crear comunidades o grupos de intereses comunes, donde los miembros pueden ingresar o retirarse en el momento que ellos desean, facilitando de esta forma el intercambio de información, comentarios en base a criterios e intereses comunes.



Amigos o contactos, una característica propia de las redes sociales y de plataformas similares es la de poder realizar invitaciones a miembros suscritos a manera de contactos personales o amigos.



Comunidades o grupos de interés, donde un usuario registrado puede pertenecer a las comunidades que el desee o sobre las cuales tenga algún interés en particular.

EL QUE LO PUEDE EL QUE LO LOGRA

EL QUE LO INTENTA

EL QUE LO SABE

ATIX

Guía de numeración de versión de PostgreSQL

La numeración de versiones de **PostgreSQL** tiene ciertas particularidades, que las describiremos tomando como ejemplo la versión de **PostgreSQL** 8.2.7 aunque la última versión liberada es la 8.3.1.

El número 8 es el "*mayor número de versión*", pero también puede ser conocido como el "*número de versión comercial*". Cada ciertos años y basados en algunos hitos, sobresale el primer dígito. Para la versión 7 (en 1999) se dió estabilidad; para la 8 se dió un mejor soporte a Windows junto con las mejoras acumuladas. Cuando se realice la 9, claro que nadie sabe cuando; probablemente sea cuando empezamos a golpear a bases de datos propietarias en todas las áreas de trabajo.

El 2 es el lugar del segundo dígito y también una importante liberación. Este Indica la liberación anual de **PostgreSQL**. Si usted combina los dos primeros dígitos ... en este caso "8.2" obtendrá la versión mayor, o liberación mayor o simplemente la "versión" de **PostgreSQL**. Los usuarios que se actualizan entre versiones mayores pueden esperar un proceso de actualización implícito y mucho testeó, esta es la razón por la cual soportamos versiones anteriores tanto tiempo. El uso de los dos primeros dígitos confunde a muchos desarrolladores de software comercial, que suponen que sólo el primer dígito es significativo. Si usáramos solo el primer dígito, aunque estemos cerca de la próxima liberación **PostgreSQL** 9 ya iríamos por la versión 21 por ejemplo.

El 7 es la última posición, es la "*versión menor*" que es un sinónimo de la "*cantidad de parches liberados*". Por lo tanto la versión 8.2.7, incluye 7 parches acumulativos a la versión principal "8.2". Sea cual sea su versión principal, es muy importante que usted se actualice a la última versión menor.

El proyecto **PostgreSQL** en este punto sólo realiza correcciones de errores con los parches liberados, y no incorpora características mejoradas, por lo que si usted no se encuentra en la última versión menor, entonces usted se está exponiendo a las potenciales amenazas conocidas en seguridad y pérdida de datos.

Actualizarse a la última versión menor no requiere un "initdb" o "dump / reload"[3], y la mayoría de las liberaciones pueden realizarse dentro de 2 minutos de inactividad en el servidor (prácticamente el tiempo que le lleva copiar los binarios y reiniciar el sistema). De vez en cuando, a veces, una mejor seguridad requiere algunos cambios en la API que son documentados en las notas de la versión. Si usted se está saltando una media docena de versiones menores, asegúrese de revisar las anteriores notas de advertencia de versiones liberadas.

Autor

Jared López Leño

jared.lopez@gmail.com

<http://jaredlopez.net>

Nota: El autor realizó la traducción [1] y adaptación del artículo original.

[1]

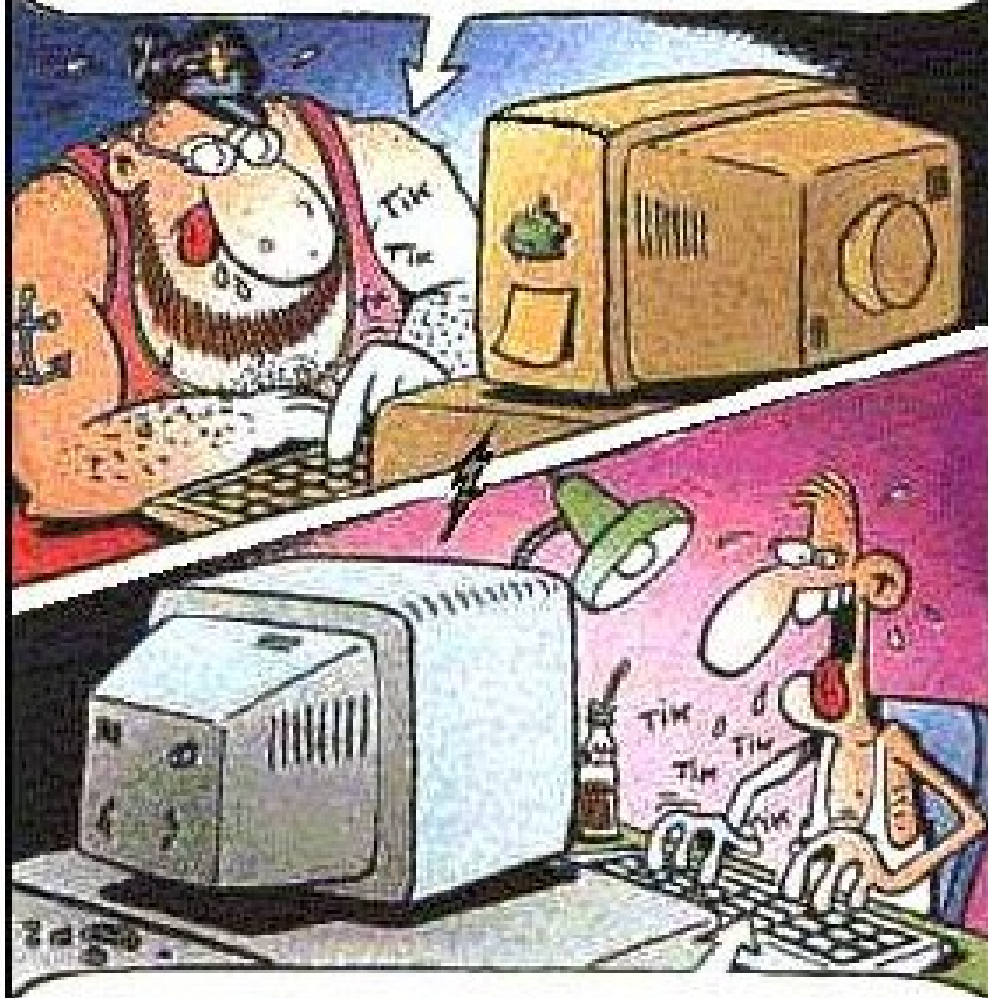
<http://blogs.ittoolbox.com/database/soup/archives/guide-to-postgresql-version-numbers-19177>

[2]

<http://www.postgresql.org/support/versioning>

[3] initdb: Comando para inicializar el cluster de base de datos.

Tengo pelo largo, rubio, pechos grandes y unos labios jugosos...



Oh Rita! Mi cuerpo musculoso tiembla de deseo por ti...



Conociendo lo Nuestro

Oruro



Anualmente durante las celebraciones de semana santa, estudiantes y profesores de la escuela de bellas artes de la Ciudad de Oruro se dan cita en **Las Dunas de San Pedro**, para dar rienda suelta a su creatividad y formar impresionantes obras de arte sacro en base a arena y agua, con motivos de la celebración religiosa.



Oruro, siempre se caracterizó por ser una ciudad minera; al ingreso de ésta ciudad se encuentra un enorme casco de minero, que simboliza esta actividad.



La enorme planicie del **Salar de Coipasa** se convierte en otro atractivo de la región conjuntamente a su fauna y vegetación



Mural en la avenida del folklore: Oruro capital del folklore de Bolivia, caracterizada por su fastuoso carnaval, actualmente ostenta el título de “**Obra maestra y patrimonio cultural e intangible de la Humanidad**”. Carnaval que anualmente sorprende a propios y extraños por su majestuosidad cultural y artística.



El **Santuario de la Virgen del Socavón**, donde peregrinos y devotos acuden a renovar su fé.



La Virgen de la Candelaria, llamada la “**Mamita del Socavón**”, patrona y protectora de la Ciudad de Oruro, en cuyo honor y devoción se realiza el carnaval.

Libres para pensar, libres para decidir, libres para crear

 **Arte** **Libre** 

Te ofrecemos este espacio para mostrar tu Creatividad



Envíanos tus diseños y creaciones para publicarlos

Contacto

Para solicitar cualquier información, puedes contactar a:

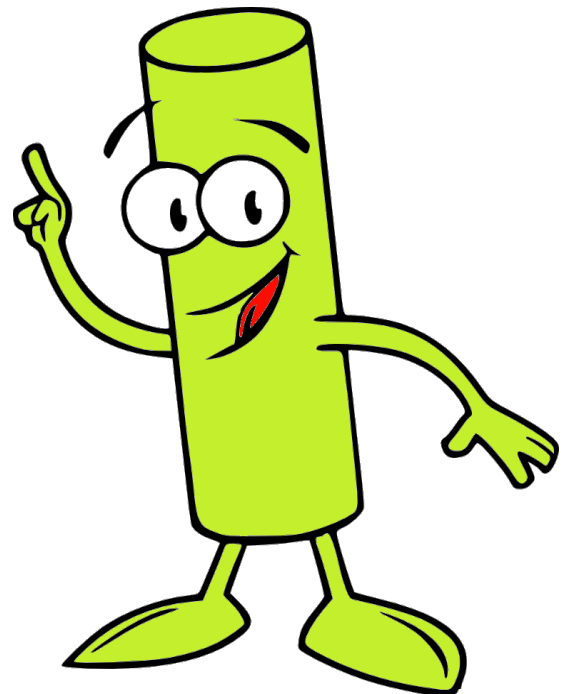
- ✓ Esteban Saavedra López (jesaavedra@opentelematics.org)
- ✓ Williams Chorolque Choque (williamsis@gmail.com)

Publicación

Te invitamos a ser parte de la **Revista ATIX**. La forma de participar puede ser enviándonos:

- ✓ Artículos referidos a áreas como:
 - ✓ Instalación y personalización de Aplicaciones
 - ✓ Scripting
 - ✓ Diseño gráfico
 - ✓ Programación y desarrollo de aplicaciones
 - ✓ Administración de servidores
 - ✓ Seguridad
 - ✓ y cualquier tema enmarcado dentro del uso de Software Libre
- ✓ Trucos y recetas.
- ✓ Noticias.
- ✓ Comics.
- ✓ Links de interés.

Usa siempre
Software Libre





Marcamos Huella



<http://atix.opentelematics.org>