ISSN 1909-3138

ESSENTIA LIBRE

Essentia Libre · No 4 · Noviembre - Diciembre 2006

Pyragua: Un editor colombiano para Python

Índice

ESSENTIA LIBRE

Creada por ACLibre
www.aclibre.org

Director:
Jeffrey Steve Borbón Sanabria
E-Mail: jeffto@aclibre.org

Diseño y Maquetación:
Erika Tatiana Luque Melo
E-Mail: ruri@aclibre.org

Editor Invitado:
Gustavo González Girón
"Xtingray"

Comite de edición:
Hernán Quishpe Guagrilla

Columnistas:
Lorena Giraldo
Robinson Andrés Palacios
David Mora Rodríguez

Entrevistas
Robert Dodier
Aaron Seigo
Jhon Alexis Guerra

Articulistas:
Daniel Rodríguez Cárdenas
Jairo Enrique Serrano Castañeda

Portada
Pyragua, logo diseñado por:
Jhon Alexis Guerra
Juan Pablo Valois

Herramientas empleadas:
Maquetación: Scribus-ng
Edición de Imágenes: The Gimp
Edición vectorial: Inkscape
Sistema: Debian GNU/Linux
(LoStHeAvEn / Kodou / Kaze) & Kyoto

Sugerencias y artículos:
aclibre@gmail.com
Noviembre - Diciembre 2006
Bogotá - Colombia

ParqueSoft y el Software Libre

Luego de un recorrido de más de 5 años y de su reconocimiento a través de diversos medios de comunicación, la Fundación ParqueSoft ha logrado posicionarse a nivel nacional como el cluster de tecnología más grande de Colombia.

Aunque sus inicios se dieron en la ciudad de Cali, los pilares ideológicos que inspiraron el concepto original se encargaron de evolucionar el modelo de una sede única a lo que actualmente se conoce como la Red de Parques Tecnológicos. En un principio se apellidó "del Suroccidente Colombiano", pero que hoy riega semillas por todo el país como un fenómeno revolucionario, brindando la oportunidad a muchos jóvenes empresarios de desarrollar sus ideas de negocio relacionadas con tecnología y especialmente con Software.

Acerca de ParqueSoft podría escribirse un libro completo, pero dado que se trata de una columna editorial, voy a centrarme en un tema puntual: la relación entre el cluster de tecnología más grande de Colombia y el Software Libre.

Empecemos por las buenas noticias.

En la actualidad varios miembros reconocidos de la comunidad de Software Libre están desarrollando proyectos de emprendimiento al interior de ParqueSoft, lo cual es bastante positivo para el movimiento, pues de alguna manera, están sirviendo como replicadores de las ideas de libertad y de las propuestas de software que solemos difundir desde nuestros grupos de usuarios locales, sólo que en un ambiente 100% empresarial.

Algunas de las empresas de ParqueSoft que están íntimamente ligadas al movimiento de Software Libre son Avatar en Popayán, Qhatu en Pasto y Soluciones Kazak en Cali, sólo por mencionar algunas -porque se que hay muchas más- lo que resulta bastante positivo, teniendo en cuenta la fuerte influencia de las estrategias de Microsoft al interior de la Fundación.

De otro lado, también quiero resaltar tres proyectos de software que nacieron al interior de ParqueSoft bajo licencia GPL. El ya bastante mencionado KToon (Un entorno para animación en 2D), Blindux (Una distribución GNU/Linux para invidentes, aun en proceso de maduración) y eMaku (Una solución ERP que espera ver la luz en enero del próximo año).

Con respecto a estos proyectos, me interesa compartir el modelo a partir del cual han sido construídos, pues me parece que rompe con el esquema tradicional (desde la academia o de los LUGS) en que han surgido otros desarrollos colombianos de Software Libre.

Es interesante ver como existen proyectos de código abierto y de libre distribución que nacen desde la iniciativa empresarial, en el caso de ParqueSoft, estos desarrollos son financiados a partir de proyectos de investigación presentados al Sena y a Colciencias.

En lo personal, lo que más me agrada de este tipo de estrategias es que la dinámica de desarrollo está obligada a ser bastante formal y continua, debido a los compromisos que adquieren los emprendedores al recibir incentivos económicos por parte de instituciones del Estado. Esto se traduce en la contratación de programadores, en la creación de cronogramas de entrega y de toda una logística que garantiza que los proyectos puedan llegar a su etapa madura en un tiempo especifico. Lo que me emociona del asunto, es que estamos hablando de Software Libre y no de productos comerciales (de código cerrado) !

Aunque parezca mentira, estos desarrollos llevan varios años en proceso de maduración y sin embargo, aun requieren de más tiempo para alcanzar su versión

Editorial

final, debido a que no se trata de programas sencillos, sino de ambiciosas aplicaciones. Lo que me lleva a concluir que si estos proyectos no son generados desde los espacios que ofrece ParqueSoft, seria bastante complicado que existieran desde nuestro país. Afortunadamente existen y espero que en algunos meses o tal vez años (si es necesario) hagan parte de las aplicaciones más populares de Software Libre de la escena mundial.

Pasemos ahora a temas menos emocionantes y hablemos un poco de las dinámicas que se manejan en un entorno empresarial como el de ParqueSoft.
Partamos afirmando que la comunidad de Software Libre en su mayoría (y no digo que toda) proviene de ambientes académicos, en donde podemos contar tanto a docentes como estudiantes.

Dichos ambientes se caracterizan por una fuerte aceptación de la filosofía del Software Libre y de un profundo respeto por sus ideales, aunque sin necesariamente llegar a posiciones extremistas.

Ahora, si miramos al interior de ParqueSoft, encontramos a una comunidad centrada principalmente en la idea de hacer negocios y de generar riqueza, desde una postura 100% capitalista, y quiero ser claro, no hablo de la Fundación como tal, hablo de la comunidad de los emprendedores. Lo que no significa que esté mal, es simplemente una característica de ese medio. Se que puede sonar frío y calculador para algunos, pero las empresas son para hacer dinero.

En este punto, es fácil imaginar cuan diferente y posiblemente antagónica puede ser la postura de ParqueSoft, como cluster empresarial frente a la comunidad de Software Libre. No quiero decir con esto que exista un enfrentamiento directo entre los dos grupos o algo parecido, pero los hechos demuestran que ParqueSoft, en su gran mayoría de emprendimientos, es apático al tema del Software Libre y muestra de ello es que sigue apuntando a un modelo tradicional de negocios en el que el factor clave para generar economía es la venta de licencias de productos de código cerrado.

Sobre este tema en particular, no me interesa proyectar mi visión acerca de "como deberían ser las cosas", simplemente quiero compartir con ustedes, mi percepción sobre el entorno actual en el que me muevo día a día.

Aunque en la gran mayoría de emprendimientos de la Fundación, suelen encontrarse uno o varios equipos instalados con una distribución GNU/Linux, o un servicio (ej: Squid) o al menos una aplicación de Software Libre (ej: OpenOffice), la verdad es que los empresarios utilizan estos recursos simplemente por el hecho de ser "gratuitos", despreciando todas las implicaciones tanto técnicas como sociales que puede implicar el uso de Software Libre. Es más, muchos de los emprendedores al interior de ParqueSoft ignoran qué significa "GPL" y por ende, desconocen las libertades que ofrece el software cubierto por dicha licencia. Lo que en lo personal, me resulta bastante triste.

En ese sentido, varios miembros de la comunidad hemos tratado de librar una batalla contra la desinformación y la ignorancia. Sin embargo, el reto es más complicado de lo que parece, sobre todo en un ambiente tan frío y hostil en donde lo que importa exclusivamente es "lo que me sirve y no me cuesta" y "lo que puedo vender". En un escenario así, intentar cambiar la cultura de las personas es bastante complicado (aunque no imposible), pero de nuevo, quiero insistir en que la naturaleza de ParqueSoft no es exclusiva de dicha Fundación, sino de cualquier espacio que pretenda ser un centro de incubación de ideas de negocio.

Desde mi perspectiva, sería ideal que desde la comunidad de usuarios de Software Libre se elaborara un plan de difusión enfocado a impactar a la Red de Parques Tecnológicos de Colombia, pues resulta un espacio supremamente estratégico para que el Software Libre y las empresas con base tecnológica se encuentren para desarrollar proyectos conjuntos, generando beneficios tanto para la comunidad de usuarios como para los emprendimientos.

Si bien ya se han realizado varios eventos relacionados con el tema en las diferentes sedes, creo que debemos continuar con la tarea desde una visión más colectiva y con la conciencia de que el Parque Tecnológico del Software es un recurso valioso no sólo para la comunidad de Software Libre, sino para el país. ■

Gustavo González Girón
Email: xtingray@kazak.ws

Llevando la belleza al escritorio GNU/Linux

Aaron Seigo
Desarrollador KDE

Entrevista desarrollada en el marco del IV Foro Mundial de Conocimiento Libre desarrollado del 17 al 21 de Octubre del 2006 en Maturín (Estado de Monagas), Venezuela.

Essentia Libre: Hola Aaron, bienvenido a la revista Essentia Libre. Dada tu experiencia con el trabajo en torno al Proyecto KDE, nos gustaría conocer un poco del mismo y cómo ha trabajado a lo largo de estos diez años. Pero antes de eso, ¿podrías hacer una introducción de KDE para quienes lo desconocen?

Aaron Seigo: Si, claro. Bueno, fundamentalmente KDE es un entorno de escritorio, que nació hace 10 años con la necesidad de crear un entorno amigable para los usuarios que iniciaban su proceso de aprendizaje y trabajo en torno al sistema operativo GNU/Linux. Este cometido poco a poco se ha ido logrando y podemos decir con orgullo que aquella lista de cosas por hacer en ese entonces, ya está cerca de ser absolutamente completada, a razón de que se ha llegado a un punto en el cual el escritorio KDE cuenta con una cantidad impresionante de aplicaciones y funcionalidades que han ido mejorando gracias al modelo de desarrollo de software libre.

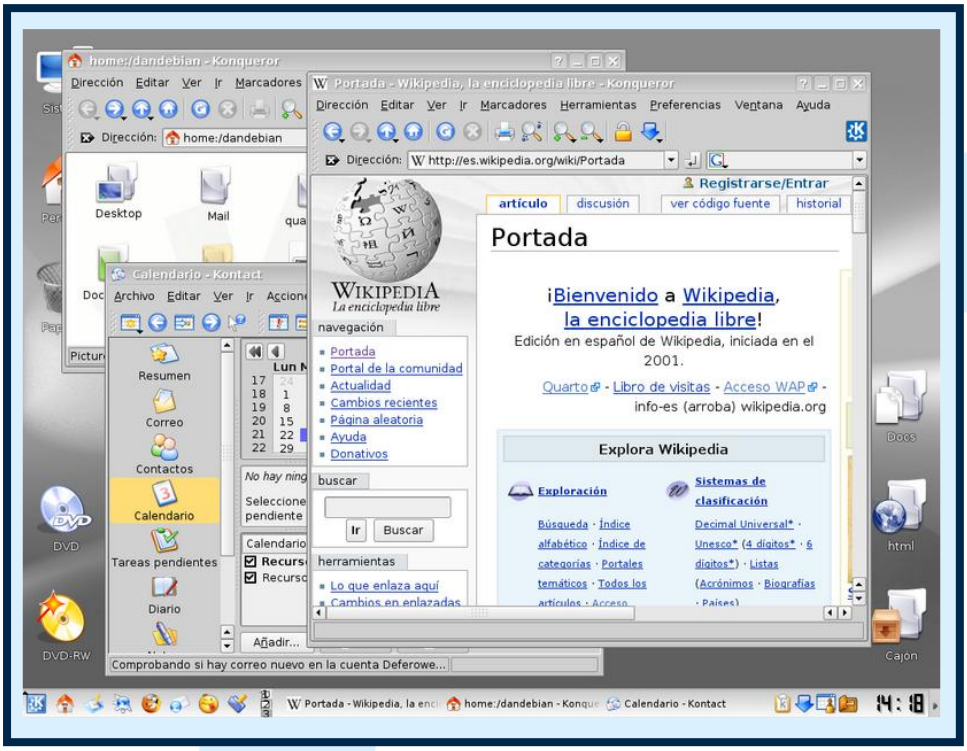
EL: Es interesante saber que el "TODO" o "Cosas por hacer" se encuentra en tal estado, eso implica un crecimiento progresivo a través del tiempo, lo que nos lleva a la pregunta ¿Cuál es el estado actual del Proyecto?

AS: Bueno, hace una semana se ha hecho el lanzamiento de KDE 3.5.0. En cuanto a lo que en su interior encontramos es una gran utilización de las funcionalidades que proveen las librerías QT. Adicionalmente el trabajo se ha enfocado a crear una gran interacción entre las diversas aplicaciones y el mejoramiento de las actuales. Podemos hablar de casos concretos como lo son KOffice, que en su nueva versión ya incluye soporte de archivos estándares tales como lo son los ODT, OTP y otros, esto gracias al trabajo en conjunto con personas parte del proyecto OpenOffice.org.

Otro caso importante es la posibilidad de encontrar múltiples aplicaciones que a simple vista serían las mismas, caso Juk y Amarok, pero que vistas de cerca se diferencian en los fines específicos de cada una de ellas. La primera de estas es un reproductor de audio simple, el otro es todo un sistema para administración de listas de reproducción al mismo tiempo que es un reproductor de audio.

En otras palabras en el estado actual del escritorio y todo el sistema que abarca se puede ver claramente que no nos hemos dedicado a retomar aplicaciones y adaptarlas al escritorio, todo lo contrario, hemos innovado y mejorado mucho, gracias al trabajo de centenares de desarrolladores al lo largo del mundo, al igual que comunidades de traducción, creadores de documentación y por supuesto usuarios.

Kde 10 años



Algo importante para mencionar es que, anteriormente hablaba acerca de que no hemos tomado otros proyectos y los hemos adaptado a KDE. Consideramos que si una aplicación por si misma ya es muy completa o funcional no existe la necesidad de hacerle un porte a la arquitectura KDE. Un caso concreto de esto es Inkscape, el cual por si mismo ya es muy completo y no requiere que lo integremos a nuestro modelo.

EL: ¿Y qué expectativas hay para el futuro?

AS: Nuestro futuro inmediato es la implementación de las librerías QT4, y llegar así al tan esperado KDE4.0, el cual tomará un rumbo nuevo en los que a entornos de escritorio se refiere.

El hablar de este nuevo proceso consiste en abarcar todas la nuevas funcionalidades que implementan las librerías QT4, lo que permitirá a los desarrolladores dedicar más tiempo a la lógica y funcionamiento de la aplicación y no a su integración o trabajo al interior de KDE. El hablar de la tecnología Plasma, en la cual participan desde desarrolladores hasta usuarios, harán de la parte gráfica del escritorio algo verdaderamente espectacular. Adicionalmente con las mejoras en gestión en memoria e interacción entre las diversas aplicaciones, el escritorio será tan fácil y completo como es hoy en día el sistema gráfico del sistema operativo MacOS.

Y como primicia, con la futura liberación de las librerías QT4 bajo licencia GPL, para el sistema operativo Microsoft Windows, será posible el porte y desarrollo de nuevas aplicaciones nativas en dicho sistema operativo. Esto es una medida que nos permitirá crecer en cuanto a la cantidad de desarrolladores que participarían en KDE, ya que desde hace un buen tiempo hemos recibido excusas por parte de buena parte de desarrolladores a razón que no desean cambiar al sistema operativo GNU/Linux. A su vez esto permitirá que aparezca nuevo software libre y así permitir más adelante que el proceso de cambio para las usuarios de plataformas privativas a plataformas libres sea simple y sin tropiezos.

En otras palabras, en el momento que sea lanzado la versión 4.0 de KDE, será posible para un usuario escoger entre las tres plataformas (Microsoft Windows, GNU/Linux y MacOS) y notar que en cuanto a uso, vistosidad y sobre todo rendimiento se pueden pasar a la plataforma GNU/Linux ya que van a encontrar lo mismo que podrían encontrar en los otros sistemas pero con el valor agregado de ser software libre y contar con una comunidad gigantesca alrededor del proyecto.

EL: Por lo que comentas se nota que valdrá la pena la espera. Cambiando un poco de tema has mencionado algo acerca del trabajo de la comunidad, ahora surge una duda, ¿Cómo ha sido la participación de la comunidad Latinoamericana en el proceso de desarrollo, traducción y documentación del proyecto KDE?

AS: Es a veces sorprendente como la comunidad latinoamericana aporta al proyecto. Es cierto que no hay demasiados desarrolladores. De hecho ampliaría el criterio a hispanoamérica, ya que contamos con unos buenos desarrolladores españoles y brasileños. Adicional a esto, la participación en proyectos de internacionalización ha crecido en los últimos tiempos no sólo en cantidad sino también en calidad. En mi opinión personal es de gran importancia el invitar más activamente a no sólo esta gran comunidad americana sino a la de muchos países donde por ejemplo la lengua inglesa no es la principal. Como experiencia especial en el anterior aKademy llevado a cabo en Dublin (Irlanda), hace poco tiempo se incorporó una interesante comunidad asiática que ha permitido trascender fronteras y marcar pautas en el desarrollo comunitario. Esto siempre ha destacado al proyecto. En pocas palabras si logramos una gran participación de comunidades como la hispanoamericana que siempre se ha destacado por su excelente y constante trabajo, el proyecto llegará más lejos.

EL: Es importante la participación de nuevas comunidades ya que permiten el crecimiento y la retroalimentación en los proyectos, pero un esquema que poco a poco se ha venido implementando es la enseñanza de estas tecnologías desde edades tempranas, a nivel educativo, ¿Cómo es el compromiso de KDE con la educación?

AS: El entorno KDE cuenta con una muy completa suite de aplicaciones educativas, a esta suite se le conoce como KDEEdu. Está conformada por alrededor de 10 aplicaciones que abarcan desde el aprendizaje de matemáticas hasta la astronomía e idiomas. Voy a comentar una experiencia, estoy al tanto de una institución educativa que tiene conexión directa con un centro astronómico, y en clases de astronomía los estudiantes pueden hacer fotografías usando el telescopio del observatorio y por medio de la aplicación KStars.

Por otra parte, en un viaje que realicé a Brasil, pude observar el caso de una gran cantidad de instituciones educativas cuyas distribuciones de GNU/Linux cuentan con la aplicación llamada Kiosk, la cual les permite la administración de aulas de informática desde un servidor, lo que permitía una labor por parte de profesores y administradores muy interesantes y una mejor interacción con los estudiantes.

Algo que me gusta recalcar en cuanto a esta suite educativa es que debido a su excelente desempeño, funcionamiento y facilidades para todos los usuarios, han sido agregadas en Edubuntu a pesar de que este cuenta únicamente con el entorno de escritorio Gnome. A partir de esto se puede hablar de que ya no sólo encontramos en el mundo del software libre aplicaciones tan completas como Gcompris y TuxPaint sino ya hay más, gracias al trabajo comunitario de muchos en torno a KDE.

EL: Estas aplicaciones claramente le permitirán a muchas instituciones educativas adaptar el entorno de escritorio KDE como su escritorio favorito. Aaron, ha sido un placer compartir estas experiencias con los lectores de nuestra revista Essentia Libre, ¿deseas extender algún mensaje para todos ellos?

AS: Por supuesto, deseo invitarlos a participar activamente del desarrollo, internacionalización, documentación, pruebas del proyecto KDE y así lograr hacer del software libre un grupo de tecnologías más completas, más poderosas y sobre todo, teniendo presente el sentido social que abarca este movimiento. Y no duden en usar el entorno KDE.

Entornos de escritorio 3D

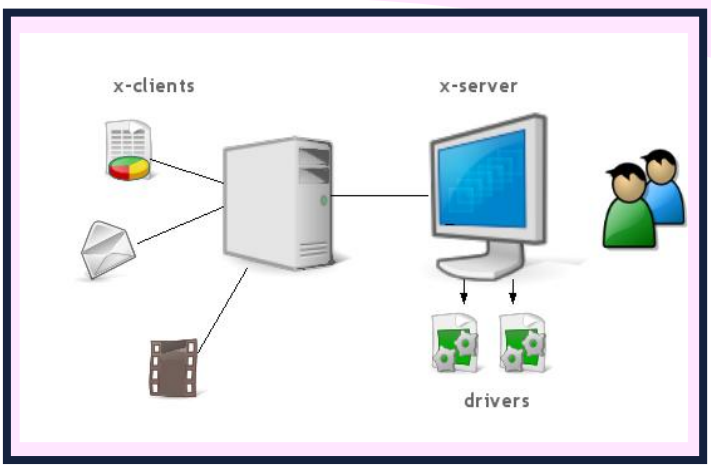
Por: Daniel Rodríguez Cárdenas
Grupo Linux Universidad Distrital
IEEE Computer
danielrcardenas@gmail.com
http://danielrcdeb.blogspot.com/

Un entorno de escritorio es un conjunto de software para ofrecer al usuario de un ordenador, un ambiente más cómodo sobre el cual trabajar y sacar el mayor provecho de las aplicaciones en un menor tiempo y de manera mas intuitiva.

El software es una interfaz gráfica o GUI que ofrece iconos, barras de herramientas y habilidades como arrastrar y soltar (drag & drop).

X-Server

El sistema gráfico de Linux es X-Window. Fue creado a mediados de los años 1980. Este sistema asume que las aplicaciones actúan como un cliente (x-client) de un servidor X-Window (x-server). El x-server se encarga de lograr la interacción entre el usuario y la aplicación. El rol del servidor es desplegar la imagen en pantalla y recibir los eventos de teclado, ratón y otros dispositivos. Hay distintas implementaciones de x-servers, los hay para GNU/Linux, MacOSX y también en Windows. @



Mientras mejor sea el x-server, mejor es la representación de la aplicación. Para ir mejorando un servidor x-server se van proponiendo extensiones (x extensions) y con el tiempo estas extensiones se van implementando en los distintos x-server y drivers. Ejemplos de x-server son XFree86 y Xorg.

Normalmente, tanto los x-client y el x-server se ejecutan en un mismo computador. La separación entre x-client y x-server también permite que una aplicación que se ejecuta en un computador, pueda utilizarse desde otro computador en forma remota con un x-server corriendo en el computador local al usuario en forma independiente del sistema operativo.

¿Cómo se dibujan las ventanas?

En la actualidad los sistemas tradicionales dibujan las ventanas como un rectángulo en donde se ubican botones y demás componentes que servirán para la utilización de la aplicación. Si se colocan dos ventanas una sobre otra, la ventana que se encuentra detrás de la otra o parte de ella que no es visible, no se dibujará. Sin embargo, cuando la ventana que cubre a la otra se mueve, la nueva superficie visible ha de ser dibujada por cada aplicación. Esto es conocido como composición.

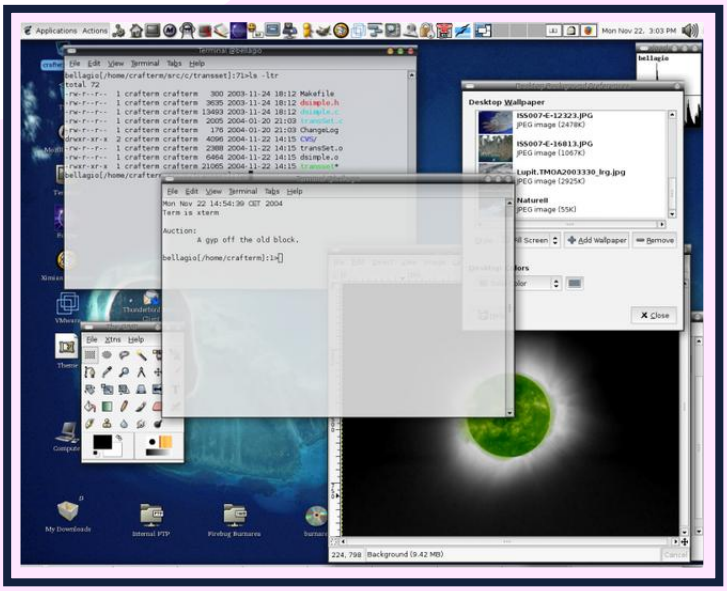
Lo anterior es experimentado cuando movemos rápidamente una ventana y las demás no alcanzan a redibujarse o cuando una aplicación deja de responder, entonces veremos simplemente un rectángulo vacio.

La desventaja es la necesidad de constantemente redibujar la ventana aunque no sea necesario y que no se pueden mostrar transparencias a una ventana. Para crear este efecto se les opaca 100%, para dar un aspecto de por ejemplo una ventana con bordes redondos. Pero si pasamos una ventana detrás de esta transparencia o moviéndola veremos que es rectangular. Además de esto, la CPU es la encargada de hacer los procesos de posicionamiento y movimiento.

Composite Manager – Window Manager

Para solucionar los problemas anteriores, se ha planteado dibujar las ventanas fuera de la pantalla (off-screen). Esto quiere decir que cada aplicación se redibujará sólo cuando algo en ella cambie y su contenido siempre estará disponible, aunque no sea visible y se guarda en la memoria del vídeo mientras esté en ejecución (Frame-Buffer del tamaño de la pantalla). Una aplicación especializada se encargará de mostrar las ventanas en el escritorio que se ve en la pantalla. Esta aplicación se le conoce como **Composite Manager** y es capaz de renderizar las ventanas y de esta manera cambiarles la forma, el tamaño, rotarlas, agregarles sombras, crear miniaturas activas y demás efectos. Un **Window Manager** se encargará de posicionarla y redimensionarla en el escritorio, visualmente son el título y los bordes de la ventana como por ejemplo Metacity, Kwin y Window Maker.

Un Composite Manager puede usar OpenGL para renderizar y tomar la ventana como si fuera una textura o imagen vectorial y colocarla sobre un polígono que es la forma que obtendrá la misma. De esta manera podemos no sólo tener interfaces rectangulares sino de cualquier forma y aprovechar el rendimiento de las tarjetas de vídeo que pueden ejercer la función de forma rápida y se libera a la CPU para que pueda trabajar sobre los procesos de las aplicaciones y no de la parte gráfica.



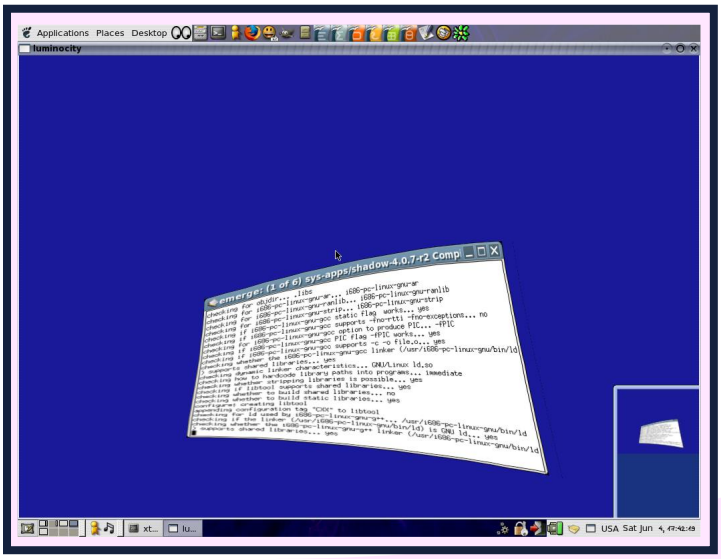
Además de otras características se puede resaltar que gracias al buffer, en ventanas con scroll como navegadores web, no se tienen que repintar en su totalidad cada vez que lo movamos sino que sólo la nueva porción de ventana se desplaza directamente del contenido del buffer a la memoria del vídeo.

XcompMgr fue el primer Composite Manager que agregó sombras y transparencias a las ventanas, pero su principal problema fueron los drivers Xorg ya que sólo las tarjetas Nvidia lo puede acelerar.

Red Hat comenzó luego a modificar Metacity que es el Window Manager de GNOME, para que incluyera funciones de Composite Manager y pasó a llamarse Luminocity.

Entre tanto, Sun Microsystems comenzaba el desarrollo del proyeto Looking Glass liberado bajo licencia GPL en 2004 y busca dar una nueva funcionalidad a los escritorios, brindar al usuario un

Entornos de escritorio 3D



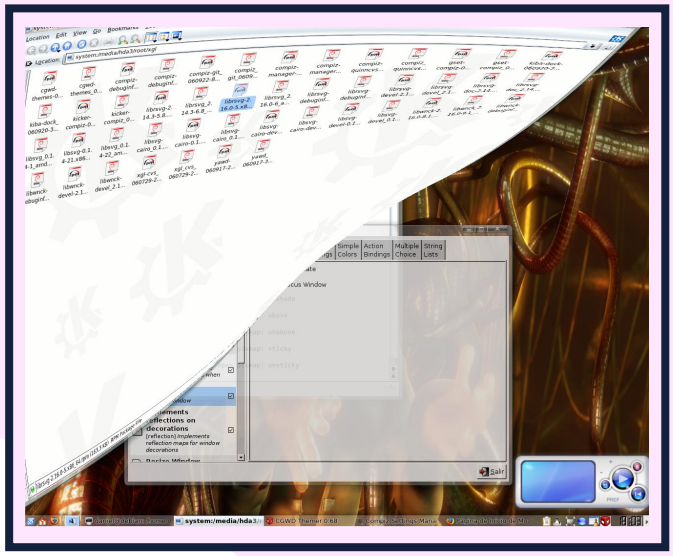
mejor manejo de la información y de las aplicaciones haciendo uso del 3D. Poderla ver en mayor volumen y acceder a ella de forma más rápida@, ya que "somos capaces de utilizar de mejor forma el espacio de nuestra pantalla", dijo Juan Carlos Soto, Sun engineering manager para el Proyecto Looking Glass. Entre sus características está el de poder voltear cada ventana y de verlas en perspectiva, así como navegar en el escritorio en cualquier dirección.

El equipo de **Apple** no se queda atrás y hace algún tiempo cambiaron su modelo de composición pudiendo incluir en su versión Mac OS X v10.2 Tigger a Quartz Extreme un gestor de ventanas acelerado por hardware por medio de OpenGL. Esto ofrece características tan conocidas con el Expose o el poder ver las ventanas pequeñas con el intercambio entre ellas (alt-tab).

Lo mismo ocurre en Microsoft con su nueva versión de Windows Vista, el cual cuenta con **Windows Aero** que brindará efectos similares a los de Mac y vista en perspectiva de ventanas (Flip 3D), pero que afecta su rendimiento. En cuanto a sus requerimientos no pasan de ser un procesador Dual Core, además de incluirse esta tecnología únicamente en algunas ediciones.

XGL

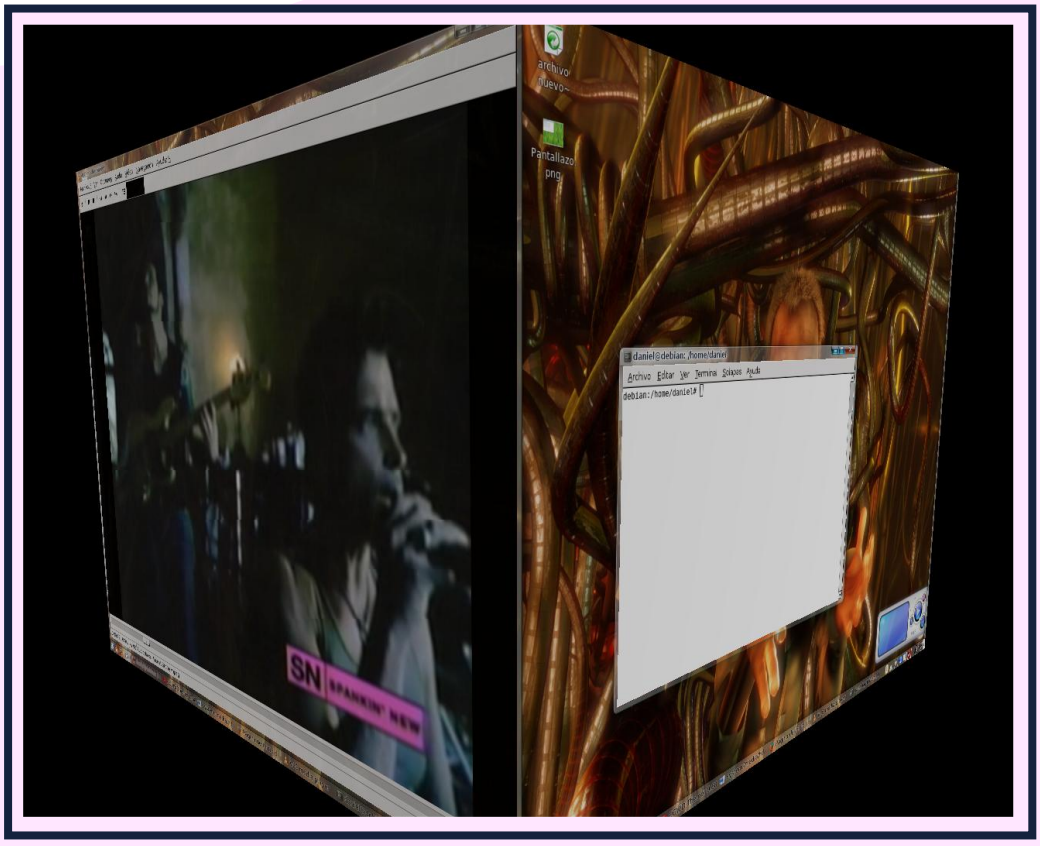
Xgl publicado recientemente por Dave Reveman de Novell, surge como una variante en el cual se quiere ofrecer más soporte y mayor provecho de las tarjetas de vídeo actuales y por lo tanto se comenzó a trabajar en un x-server nuevo, olvidándose del ya existente Xorg. Una tarea muy extensa pero que obtuvo solución. Cuando este realiza sus operaciones render bajo el driver de OpenGL, lo que hace es conectarse a otro servidor con soporte OpenGL a través de la exención GLX. De esta manera si cargamos Xorg mas Xgl obtenemos un x-server capaz de de acelerar nuestro escritorio y poder montar un Composite Manager que nos cree los efectos deseados.



Uno de ellos se llama Compiz, que transitó por varias versiones y un desarrollo rápido, empezando por una lista de correo de la comunidad a un grupo de desarrolladores que lo liberaron, creando mejoras y agregando características casi diarias. Por tal razón,

siempre se mantuvo como versión inestable, construida por un núcleo y varios diversos que agregan efectos como movimientos difusos, el cubo formado por los cuatro escritorios característicos en GNU/Linux, Expose de Mac, sombras , etc. Compiz es un gestionador de ventanas para Xgl y contrario a lo que se piense, no se necesita de una gran máquina para poderla correr, obviamente es necesario una tarjeta de aceleración gráfica (aquí una lista de tarjetas soportadas). La última es la versión **Compiz-Quinnstorm** incluida en los repositorios de distribuciones como OpenSuse 10.1 y Ubuntu Dapper 6.06.

Muchos de los usuarios tuvieron(mos) problemas con el Window Decorator de Compiz pues usaba Metacity de GNOME o Kwin de KDE y al ejecutarlo desaparecían los bordes. Entonces se desarrolló **Cgwd** (Custom Generic Window Decorator), un decorador que también le da efectos a los bordes - como iluminar los botones de la barra de título cuando pasa el puntero encima o se oprime- al igual que transparencias y temas de escritorio.



En el pasado SLUD 5 presenté una breve demostración de Compiz sobre Debian Testing para lo cual se convirtieron paquetes .rpm de Suse a paquetes .deb de Debían. Se configuró la aceleración gráfica y un pequeño script que lo ejecutará. En estos momentos Compiz ya se encuentra entre los repositorios de Debian Sid (inestable) aquí y aquí; esta ya incluye los scripts en Python para ejecutarlo. Sin embargo se debe configurar el archivo xorg.conf y gdm.conf para la aceleración.

Ahora Beryl es un reemplazo, que surge como un derivado de Compiz. Debido a algunos desacuerdos dentro del grupo de desarrolladores, este último es mejorado e impulsado por otro equipo de programadores que promete darle continuidad al proceso. En sus repositorios se encuentra un mayor soporte frente a las diferentes distribuciones de Linux como también FreeBSD (wiki).

Y finalmente, Aiglx es un X-server creado por el grupo de desarrolladores de Fedora como una opción más de XGL, que comenzó haciendo pequeñas modificaciones a Xorg, por lo tanto, tiene un mayor soporte de hardware. Aunque no se construyó desde cero, tiene muy buen rendimiento en combinación con Beryl e igualmente tiene aceleración por medio de OpenGL. ■

Pyragua
Un editor Colombiano para Python

Jhon Alexis Guerra
Desarrollador Pyragua

Entrevista a Jhon Alexis Guerra, Realizada en el marco de la V Semana Linux de la Universidad Distrital (SLUD5).

Docente Universidad Tecnológica de Pereira
Email: aguerra@parquesoftpereira.com

Essentia Libre: ¿Qué es Pyragua?

Jhon Guerra: El Pyragua es proyecto que busca construir un editor Pythonico para Python, que sea un entorno integrado que supla la necesidad que hay en la actualidad en la comunidad y un entorno portable, liviano, que tenga diseñador de interfaces, documentación y sencillo, respetando así la filosofía Pythonica y llegar a ser el mejor editor de Python que exista.

EL: Hasta el momento ¿qué editores de Python existen?

JG: Existen muchos editores para Python, la mayoría tienen cosas buenas pero también tienen deficiencias. El primero de ellos es IDLE que viene por defecto con Python, pero no es amigable y causa de ello es que está hecho en librerías gráficas Tkinter. En él no hay un diseñador de interfaces, tampoco cuenta con la función de autocompletado, aunque su punto a favor es que busca en la documentación. A pesar de que cuenta con un buen diseño ya está un poco obsoleto.

Editores como Emacs y Vi, pues se puede decir que están hechos exclusivamente para hackers ya que su utilización no es trivial. A pesar que en el caso de Emacs puedes decir con orgullo que usas los cinco dedos para identar una línea de código o todo un programa, pero no es amigable para un usuario nuevo. El otro caso es que no cuentan con el soporte en Windows que uno esperaría y para el caso de un desarrollador en Python que viene de usar Java con un entorno NetBeans -que es el más competitivo en el cual pues es sólo hacer en un menú: Siguiente, Siguiente- pues no es tan fácil.

Hay otro llamado SPE el cual he utilizado hasta hace poco tiempo, pero su mayor pecado es que su desarrollador acostumbra a hacer las cosas bonitas y no tan funcionales, lo que deriva en una gran cantidad de bugs y errores, debido a que es un proyecto en el cual se integra demasiadas funcionalidades de otros editores, pero no cuenta con un buen proceso de depuración.

Existe otro llamado BoaConstructor, que es el único que conozco libre que cuenta con un entorno integrado de edición que incluye diseñador de interfaces basado en la librería wxPython (la cual escogimos en nuestro proyecto Pyragua), pero su punto en contra es que es muy inestable y que a pesar de que soporta el manejo de sizers -lo cual es imprescindible en la construcción de interfaces y especialmente en wxPython- pero posee por defecto una opción que le indica que ubique los componentes de la GUI de una manera que no es muy ideal para desarrollos multiplataforma.

Está Eclipse. Su pecado radica en que se encuentra hecho en Java lo que lo hace muy pesado, además la instalación del plugin Pydev no es muy sencilla y requiere de una muy buena conexión de internet para este proceso.

Y hay muchos más como Eric3 que es muy famoso ya que el trabajo es sobre librerías QT, PYEP que es un intento para hacer un editor Pythonico, pero no se por qué no gusto, y ahora con el lanzamiento mundial de Pyragua, inclusive nos refirieron a un Wiki de desarrolladores de editores de entornos de desarrollo para Python y ahí hay un listado que permite observar cómo hacer para no repetir la historia, lo cual es una experiencia muy interesante.

Así pues después de un análisis nos dimos cuenta que no había un editor como el que pensábamos y una de las razones por la que muchos usuarios Java no se pasan a Python es por la falta de un entorno de desarrollo bueno. De esta manera sabiendo que esta falla existe y que se estaba conformando un grupo en la Universidad Tecnológico de Pereira (UTP), pues decidimos arrancar ese desarrollo.

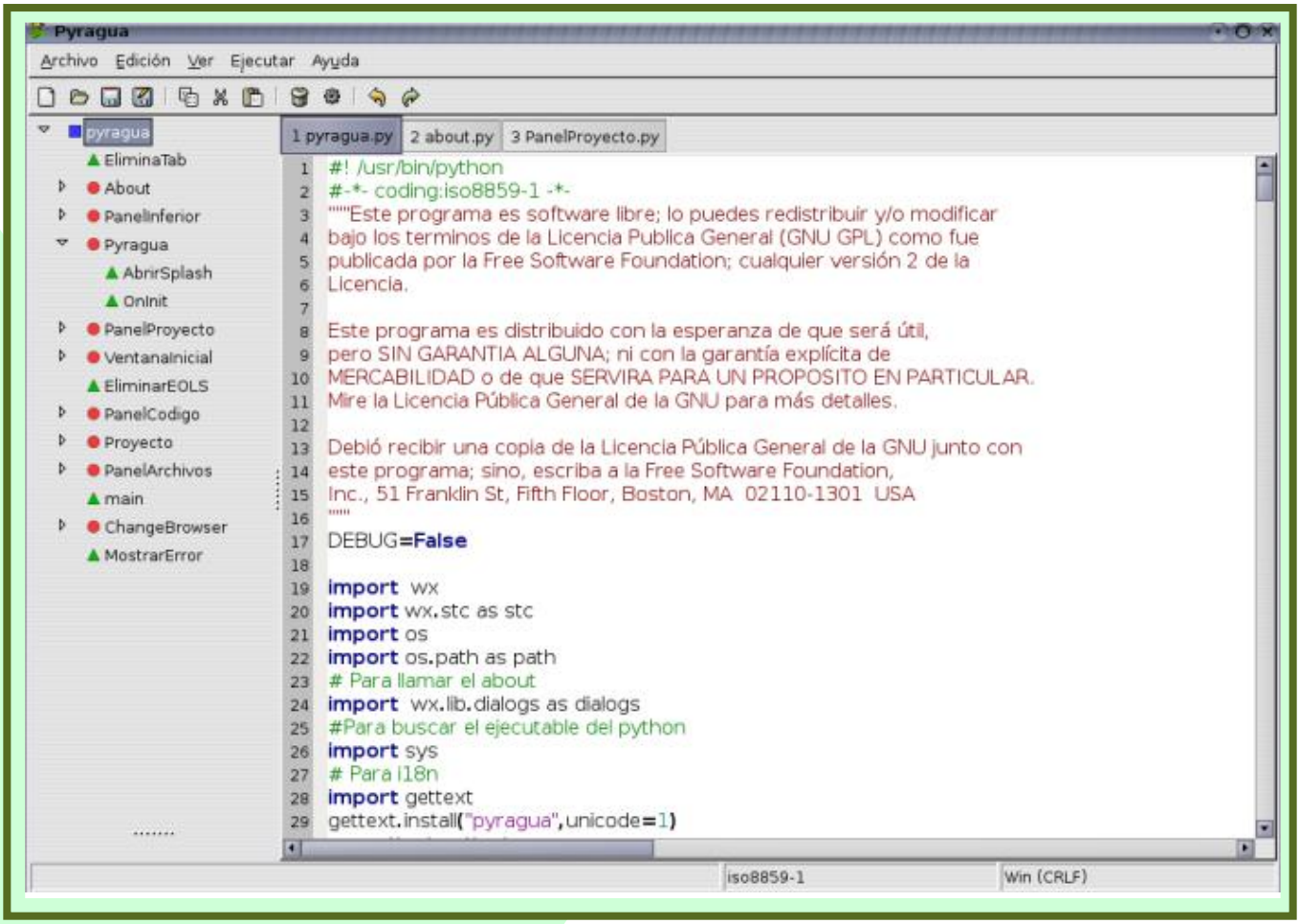
EL: ¿Qué hace especial a Pyragua?

JG: La primera razón es que es colombiano, a nivel técnico lo que lo hace especial es que podemos garantizar la continuidad del proyecto, ya que no es un desarrollo hecho por una única persona. No es sólo mi desarrollo o el de Juan Pablo Valois o Fabian Sabogal o Victor Urrea o algún otro colaborador del proyecto, lo que no es individual sino es la participación de todo un grupo.

Pyragua es un proyecto apoyado por una institución que es el grupo Pyrox, que a su vez esta apoyado por Pulpa y pues este es apoyado por la UTP. Muestra de esto es el servidor Quimbaya y que lo que deseamos garantizar es que, pese al paso de personas el proyecto continúe. Mi posición de profesor en la UTP me brinda ciertos privilegios al respecto, para garantizar eso y no que sea desde la posición de estudiante cuando nadie le hace caso. Por eso podemos asegurar que no dejaremos morir el proyecto.

Además queremos atender a las sugerencias de la comunidad. Un ejemplo: si alguien de la comunidad dice “A mi me gusta más Emacs”, pues el trabajo se puede centrar en reunir todas esas características para así crear el mejor editor de Python que exista. Aparte de eso ya tenemos, en el primer lanzamiento de Pyragua que lo pueden encontrar en la web del proyecto, un editor de texto básico con autocompletado básico, con syntax highlighting. Esto fue tomado de scintilla donde ya estaba hecho. Ya posee un ejecutar básico y hemos empezado a implementar algo único que hemos denominado “Pyraguazos”. Cuando se empieza a trabajar en Python una de las cosas más molestas es que todos los métodos de una clase, el primer atributo es self. Con Pyragua al escribir el def para definir la clase y al presionar la barra espaciadora, él abre paréntesis, coloca el self, coloca una coma, cierra paréntesis y coloca los dos puntos, de esta manera se hace que el lenguaje Python que ya es sencillo de escribir, sea más fácil. Esto aplica para las clases y más funciones de Python. Queremos implementar más “Pyraguazos”, así que si alguien tiene más ideas, son bienvenidas y sabemos que hay muchas cosas de ese estilo que se pueden hacer en Python.

Por otro parte, ya hablando como desarrollador y no como vocero del proyecto, mi idea es desarrollar el mejor autocompletado que existe para implementarlo en Pyragua. El mejor que yo conozco es uno que viene con una aplicación llamada PyCrash que viene con wxPython. Es un shell de Python. Dado que es un shell puede saber de qué tipo es cada objeto cuando se crea una función, por el nivel de polimorfismo. Los atributos de la función pueden ser de cualquier tipo, puede ser una lista, un entero o puede ser cualquier cosa. Mi objetivo es hacer el mejor sistema de autocompletado, a tal punto que adivine tus gustos para poner las variables. Por ejemplo si yo siempre uso variables que son listas, y si yo no le indico nada diferente pues serán listas; si me equivoco pues corrijo



eso y pues el sistema irá aprendiendo, ya sea por el uso o el contexto. Aún no se exactamente cómo hacer eso, dado que Python tiene muchas herramientas que permiten la introspección, que permiten saber qué métodos pertenecen a cada objeto, siempre y cuando sepas de qué tipo es ese objeto.

Viendo esto nos damos cuenta que existen muchas herramientas. En estos momentos el autocompletado que es similar al que tiene el SPE -que es muy bueno-, aunque nos faltan algunas cosas. Aparte de eso vamos a crear un diseñador de GUI que sea intuitivo. Uno de los problemas de diseñar utilizando wx es que hay que hacer mucho código que es repetitivo, cosa que un diseñador de interfaces podría hacer por uno y aparte tener que lidiar con los sizers como con herramientas como el wxGlade. Debido a esto queremos crear un diseñador de interfaces que infiera la posición de los sizers de tal manera que una persona que no sepa qué es un sizer pueda trabajar con ellos y le sea trasparente, manteniendo la potencia de permitir que una persona que lo sepa pueda trabajar con ellos. Esto estará disponible en próximas versiones. Adicional a eso buscamos lograr una integración muy buena con la documentación, una de las cosas feas en Python es tener que trabajar por un lado y tener que buscar la documentación en otro.

Y finalmente reunir las mejores características de los mejores editores que hay en el mercado sean privativos, sean libres para Python, y especialmente para Python y no desviar nuestra atención en crear un editor multilenguaje que ya hay muchos, por ejemplo Scite es muy bueno, pero si se hace eso se pierden esfuerzos en otras cosas que por el momento no interesan.

EL: Nos gustaría conocer qué lineamientos o directrices marcan el proceso de desarrollo de Pyragua.

JG: Pyragua es un experimento local de probar lo que es el software libre. Básicamente el lineamiento

principal de Pyragua es que sea software libre y que la metodología de desarrollo sea la misma del modelo de software libre. Para eso nos estamos apoyando en el módulo Gforce del servidor Quimbaya, el cual es la versión libre de Sourceforge y pues viene incluido en Debian. Así poder crear una comunidad virtual de desarrolladores porque los desarrolladores son todos, por el momento, de estudiantes y miembros UTP, pero las puertas están abiertas para todos los que deseen colaborar.

Hasta el momento hemos podido vivir todas las ventajas y desventajas que encuentra uno en un desarrollo libre. Empezando, hemos lanzado la primera versión con aproximadamente un mes o mes y medio de desarrollo, que mostrándole eso a los desarrolladores tradicionales de la Universidad es algo completamente ilógico. Esto debido a que es un desarrollo caótico. Entonces, para nosotros es eso, un laboratorio de exploración para saber qué es como tal software libre. Dado que queremos que sea un proyecto de calidad, yo me preocupo mucho por vigilar eso y en el equipo de trabajo tenemos claro que debemos pasar por unos períodos de prueba, lo suficientemente razonables, para no perder credibilidad en el proyecto, como ha ocurrido ya con otros editores. Queremos ser muy receptivos a la comunidad, ya que en ella tienen quejas, sugerencias, requerimientos y tratar de atenderlas bien, a pesar de que aun no se ha presentado mucho. El apoyo de la comunidad ha venido es con las descargas que superaron lo que esperábamos inicialmente. Todos pensábamos que sólo contaríamos con el apoyo de nuestras mamás, pero a la fecha (Octubre 3 de 2006) llevamos 210 descargas. Las 200 se alcanzaron con 20 días de publicación. En los primeros 3 días se llevaban 40 aproximadamente. Y pues, esos son los lineamientos.

EL: Hablando de Pyragua, ¿por qué ese nombre?

JG: El nombre Pyragua, nace de una discusión que tuvimos en el grupo acerca de ponerle el nombre a un editor de Python. Así que como es tradición tener la X en los productos derivados de Unix, para los productos derivados de Python, la tradición es que incluya “py”. Entonces con los muchachos del proyecto probamos algunas cosas como “Pyrex”, “Pyrox” y sobre todo les gustaba cosas “gringas” que significaban fuego y cosas así, pero a mi me han gustado los nombres más tradicionales. Un ejemplo es Pulpa, se refiere a la pulpa del café y a su vez representa nuestra área y a su vez cuenta con la sigla ULP (Usuarios Linux de Pereira). En todo caso la idea con Pyragua es que cualquier colombiano que no conozca que es la piragua no es colombiano y pues le colocamos la “y”. Esto es una forma de dar a conocer nuestro país, especialmente en esa área de desarrollo de software libre en la que tal vez estamos un poco quedados con relación con otros países, y no sólo su uso. Entonces es una manera de dar a conocer al país, indirectamente.

EL: Coméntanos un poco acerca del grupo de desarrollo de Pyragua

JG: Nosotros somos el grupo Pyrox, ese nombre lo colocaron los mismo estudiantes a pesar de mi desacuerdo ya que suena muy extranjero, pero pues el nombre del editor Pyragua es de mi autoría y ese si es muy colombiano. Como grupo somos una especie de hijo de PULPA, el cual es el Grupo de Usuarios GNU/Linux de Pereira y semillero de investigación de la UTP. En la web del proyecto Pyrox que se encuentra hospedada en el servidor Quimbaya podrán encontrar toda la información acerca del grupo y de nuestro proyecto estrella que es Pyragua. La invitación especial es a todos los desarrolladores en Python de la comunidad Colombiana a que empecemos a centralizar esfuerzos para trabajar juntos. No para absorberlos en el grupo ni mucho menos, sino para hacer que la comunidad prospere y así crear una gran comunidad de Python Colombiana.

EL: Retomando la idea de Pyragua, ¿este proyecto ya está avalado por Python?

JG: Ese es uno de nuestros objetivos, queremos que se convierta en el editor estándar de Python y así sacar

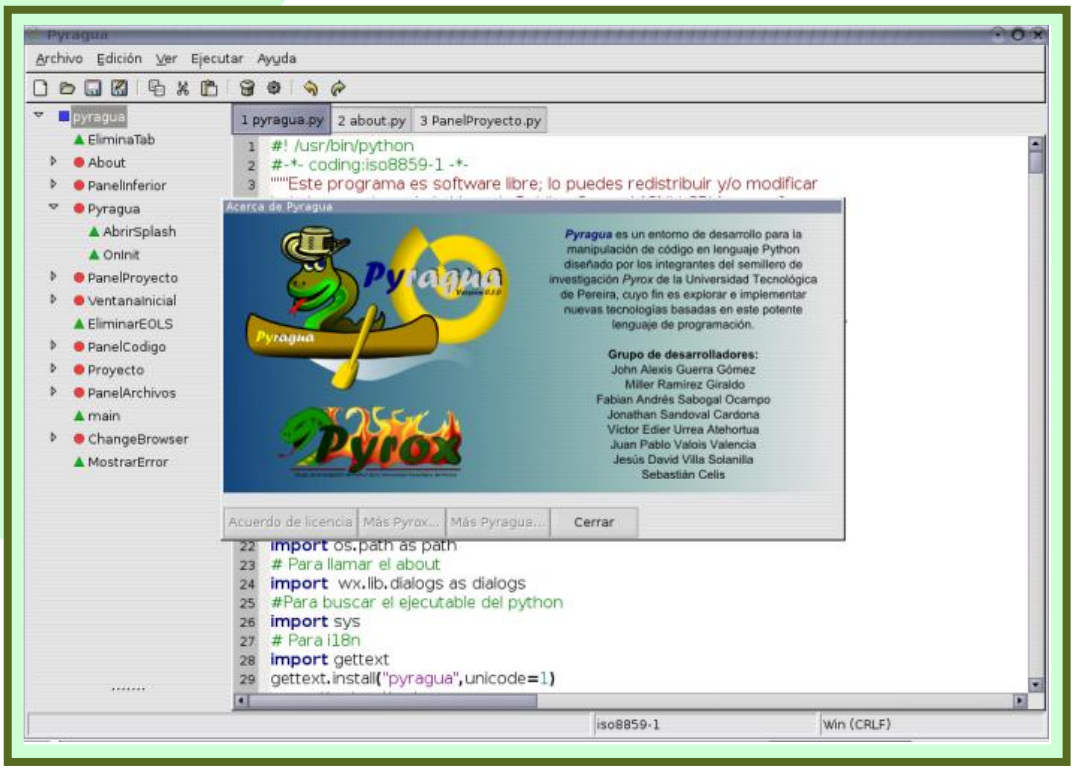
al editor IDLE, para eso debemos avanzar más. Por ejemplo, cuando hicimos la presentación nos escribieron de España dándonos apoyo y sugerencias: cómo agregar nuevas funcionalidades y características. De todas formas, el primer lanzamiento se hizo para mostrar que el proyecto si está funcionando, para presentarlo a la comunidad, para que la gente empiece a apoyarnos (si lo desea) o empiece a pedirnos cosas. Cuando el proyecto alcance un nivel de madurez, donde podamos decir, Pyragua es mejor que IDLE en todas sus características, iremos a las lista oficial de Python o lograr convencer a Guido van Rossum para que lo incluya, incluso creo que vendría de la mano con la inclusión de wx como librería estándar general dentro de Python. En el caso de que no fuese aceptado, creo que empezaríamos a prestar como una especie de versión paralela para Python, donde se encuentre una sin Pyragua y la otra con Pyragua y wx. Esto para que no sea necesario instalar tres o cuatro paquetes, algo similar a como lo hace ActiveState pero más libre.

EL: Pero, ¿Pyragua sólo está disponible actualmente para plataforma GNU/Linux?

JG: No, una de las ideas principales es que sea multiplataforma. Las herramientas que usamos son completamente multiplataforma. Sin embargo en estos momentos sólo contamos con un empaquetado para Windows y un .zip para plataforma general. Tenemos un miembro del equipo que está encargado de generar unos paquetes .deb (Distribución Debian) y queremos empezar a generar otro tipo de binarios para otras plataformas. La idea es que sea lo más portable que se pueda y esa ventaja nos la brinda las herramientas que estamos usando.

EL: Y finalmente, en la etapa de desarrollo ¿cuál crees que será el proceso más complejo y complicado?

JG: El diseñador de interfaces será el punto más difícil porque hasta este momento nos hemos apoyado mucho en la experiencia de otros desarrolladores de interfaces, especialmente en ciertas librerías de Python que son muy buenas y es cuestión de ponerlas a trabajar en conjunto. Pero para el diseñador de interfaces yo no conozco nada bueno. Personalmente no conozco nada bueno para wxPython, lo único que conozco es el wxGlade pero tiene muchos bugs; de Boa ya exprese mis diferencias con ellos y trabajar con los archivos XML no me llama mucho la atención ya que no es muy intuitivo, aunque lo estamos evaluando. Por eso creo va a ser lo más duro.



Drupal; casos de éxito, comunidad, diseño y estándares.

Jairo Enrique Serrano Castañeda
Universidad Tecnológica de Bolívar
jairo.serrano@gmail.com
http://www.jsnat.com

Al momento de idear una forma de presentar información en Internet, en lo primero que se piensa es en una página web: diseñarla, copiar y pegar textos desde un editor de documentos a un editor de páginas, subirlas a un ftp ubicado en algún servidor y luego comprobar si realmente se puede acceder a ellas, supervisando como quedo el trabajo final; en algunos casos son cientos de páginas las que se necesitan diseñar, editar cada una de ellas, formatear textos, colores y publicar, rutinariamente cada una de ellas;

¿Se alcanza a imaginar el grado de trabajo que tiene que mantener un portal web de un periódico? cientos de paginas todos los días, esto puede ser un trabajo extenuante. Llega un momento donde es tal la cantidad de documentos que hay que portar a la web que se convierte en un trabajo inmanejable y estresante.

En este punto entra en juego el concepto de CMS (**Content Management System**) o Sistema de Gestión de Contenidos, el cual es un sistema.

¿Qué es un CMS?

Consiste en una interfaz que controla una o varias bases de datos donde se aloja el contenido del sitio. El sistema permite manejar de manera independiente el contenido por una parte y el diseño por otra. Así, es posible manejar el contenido y darle en cualquier momento un diseño distinto al sitio sin tener que darle formato al contenido de nuevo, además de permitir la fácil y controlada publicación en el sitio a varios editores. (Tomado de la Wikipedia: CMS)

¿Quién lo usa?

Cualquier persona que desee darle dinamismo a su portal personal o empresarial.
Aplicaciones prácticas

Tener una bitácora ó blog personal con diseño y estructura profesional, el portal de la empresa, un álbum de fotos, un foro fácil manejable, una web de noticias categorizadas o la necesidad de una página para incluir información muy variada, son fácilmente realizadas por un CMS sin esfuerzos y mucha funcionalidad pre-fabricada.
Drupal

Existen cientos de gestores de contenido, pero al momento de seleccionar el que mejor se acople a las necesidades se debe tener en cuenta el buen diseño estructural y gráfico que se puede lograr con la herramienta. Drupal cumple con estas expectativas y el hecho de que es software libre le adiciona un componente altamente atractivo a cualquiera de sus usuarios

¿Qué es Drupal?

Es un sistema de administración de contenido para sitios Web. Permite

publicar artículos, imágenes, u otros archivos y servicios añadidos como foros, encuestas, votaciones, blogs y administración de usuarios y permisos. Drupal es un sistema dinámico: en lugar de almacenar sus contenidos en archivos estáticos en el sistema de ficheros del servidor de forma fija, el contenido textual de las páginas y otras configuraciones son almacenados en una base de datos y se editan utilizando un entorno Web incluido en el producto. (Tomado de la Wikipedia: Drupal)

Características generales

Desde el primer contacto con Drupal se dispone de un amplio catalogo de utilidades:

- Ayuda en línea disponible en cualquier punto de la interfaz.

- Es código abierto, por ende se pueden crear distribuciones de Drupal con una funcionalidad especifica que no se encuentra en la versión original.

- La comunidad desarrolladora es excelente y colaboradora.

- Existe empresas de soporte, desarrollo y diseño exclusivamente en Drupal.

- Gracias a la misma comunidad y a las características modulares del proyecto se puede crear funcionalidad extra en forma de módulos redistribuibles y publicados en la mayoría de los casos en el catalogo del portal principal.

- El portal generado con Drupal se puede configurar extensivamente desde una interfaz intuitiva y fácil de manejar.

- Las URL de los contenidos resultantes pueden ser configurables a gusto del creador del portal.

- Drupal es completamente multiplataforma, esto se logra al estar completamente desarrollado usando software libre como PHP, Apache y MySQL. Puede ejecutarse en un servidor en cualquier sistema operativo donde existan estas herramientas.

Gestión de usuarios

- Autenticación de usuarios. Los usuarios pueden registrarse o ser registrados por el administrador, para luego con un usuario y clave acceder a más servicios disponibles en el portal. Existe otro método de autenticación interesante. Un usuario registrado en un sitio Drupal puede ingresar directamente en un segundo sitio con las mismas credenciales si el administrador del segundo sitio así los dispone, esto refuerza las características innatas de creador de comunidades de la cuales Drupal dispone.

- Se puede crear perfiles o roles de usuarios basados en permisos asignados granularmente a cada uno de ellos. Por ejemplo, pueden existir grupos de usuarios que pueden publicar contenidos directamente, otros pueden necesitar aprobación de un tercer usuario que desempeña el rol de administrador.

Gestión de contenido

- Existen diferentes tipos de contenidos, pero se maneja un concepto de unidad llamado nodo, así sea un articulo, un comentario, un libro, una página de este, una duda en los foros o una imagen. Todo se puede enlazar y acceder muy fácilmente.

- Gracias a la característica anterior se hace uso de otro concepto llamado “Enlaces permanentes”, no importa la URL configurada por el creador para acceder al contenido, siempre hay un acceso directo estandarizado con base al nodo que representa.

- Cada página de contenidos generada puede ser relacionada directamente por RSS.

- A cada nodo se puede relacionar comentarios creados por los usuarios. El potencial es enorme, una encuesta

puede ser comentada con la misma facilidad que una imagen o un articulo publicado.

- Existen el concepto de taxonomía o categorías. Todos los nodos pueden ser clasificables o agrupar clasificaciones.

- Pueden crearse "Libros colaborativos". Cada usuario puede escribir una página y luego alguno de ellos las enlaza, esto se puede usar para generar cualquier tipo de documentación.

- Existe un control de versiones entre ediciones de cualquier nodo. Una utilidad realmente agradable.

- Los usuarios disponen de un blog desde que ingresan al sistema.

- La presentación visual del portal se basa en hojas de estilo CSS y plantillas para la presentación de contenidos; Cambiar de formato al millar de páginas en el portal se hace con 2 botones.

- Para hacer uso de la anterior funcionalidad hay que tener en cuenta que soporta los estándares más importantes en la presentación de contenidos para la Web: XHTML y CSS. Cualquier portal desarrollado con Drupal puede ser validado 100% con las herramientas de la W3C, siempre y cuando el desarrollador siguiera las mismas pautas de estilo y diseño.

Existen otras características específicas de Drupal en cuanto a la administración del portal. Todas estas se pueden consultar directamente en http://www.drupal.org o en Drupal Hispano http://www.drupal.org.es.

Casos de éxito

En el año 2004 en la Universidad Tecnológica de Bolívar se inicia la experimentación con Drupal como portal lanzador (con esto se entiende que Drupal sólo es usado para mostrar la información y noticias del portal, no da funcionalidad extra a las aulas) del Sistema de Aprendizaje Virtual Interactivo SAVIO el cual presta sus servicios a todo el pregrado, especializaciones y a cursos de formación permanente, los docentes y alumnos.

A finales del año 2005 toda la infraestructura de portales web de la universidad estaba soportada por esta potente herramienta. El portal principal (http://www.unitecnologica.edu.co) se construyó en muy poco tiempo, demorándose más la planeación de los contenidos y secciones que la implementación y organización de los mismos.

Claro, toda la migración se realizó analizando las otras posibilidades. Por nuestras manos pasaron Mambo (Joomla), Xoops, Nuke, B2Evolution. Wordpress, pero ninguno de ellos colmaron nuestras expectativas. Se necesitaba una herramienta que generara código html limpio, con un marcado estándar y sintácticamente correcto; los primeros en fallar fueron Nuke y Mambo. Teníamos entonces xoops, B2 y Worldpress. Estos 2 últimos siendo descalificados por ser especialmente diseñados únicamente para llevar blogs. Quedaron Xoops y Drupal, decidiéndonos por este último, gracias a una extensa comunidad que brinda el soporte, por un gran catalogo de módulos adicionales y por ser en ese entonces una herramienta de gestión de contenidos con un futuro muy prometedor. Las comunidades y empresas que giran a su alrededor se incrementan día a día, como el caso de Drupal Hispano, del cual hablaremos más adelante.

Actualmente se soportan los siguientes portales:

Principal de la Universidad.
Noticias e información de SAVIO
Convocatorias, eventos y noticias de la "Dirección de educación y desarrollo docente".
Noticias de la Universidad en "El Cartero"

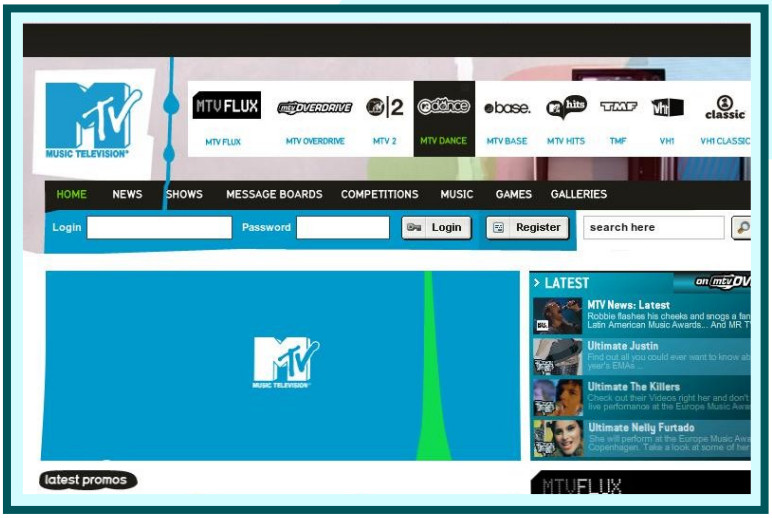
Y se esta trabajando en una implementación del portal informativo de la biblioteca, donde se publicarán las novedades bibliográficas y noticias de esta.

Ahora no solamente se usa como herramienta que ayuda a la academia. Drupal es usado en portales de alto tráfico y no tiene limitantes en cuanto a su presentación visual. Es completamente configurable, gracias a las características nombradas anteriormente. Basta con construir un buen tema gráfico y la presentación cambia completamente.

Se puede tomar como ejemplo los portales en Internet de:

- The Onion (http://www.theonion.com): Un sitio de noticias de alto tráfico.

- MTV Inglaterra, Música, Videos, Noticias, encuestas, juegos y comunidad. (http://www.mtv.co.uk)



- Terminus1525 (http://www.terminus1525.ca): una comunidad canadiense de artistas.

Estos son simples ejemplos que nos ayudan a darnos cuenta que sólo basta conseguir buenas herramientas, mucha creatividad y ganas de hacer las cosas bien.

Comunidades Drupal

La comunidad principal es la original (http://www.drupal.org), pero eso no impide que diversos grupos de personas apuntando hacia un mismo fin se pongan de acuerdo y funden un portal dedicado a brindar soporte sin ningún tipo de interés.

El 11 de abril del 2005 oficialmente se dio inicio a la comunidad Drupal Hispano (http://www.drupal.org.es), partiendo de la traducción de los manuales al español de la cual disponía gracias al proyecto de montaje de los portales a la universidad. Así mismo fueron sumándose esfuerzos, más colaboradores generaron el tema visual y la estructura que se mantiene hasta el momento.

Actualmente se está realizando un estudio de cambios a la comunidad: brindarles más herramientas a los usuarios, interacción entre ellos y formas de participar más fácilmente, no basarse únicamente en foros. 2866 usuarios registrados son los que respaldan el éxito de Drupal Hispano. Claro, no podemos esperar que todos actualmente estén activos, pero lo que sí sabemos es que todos en algún momento solventaron una necesidad o colaboraron con alguien a solventarla. ■

David Mora Rodríguez,
Ingeniero Electrónico,
david.mightyd@gmail.com
http://mighty-linux.blogspot.com/

El diablo, la manzana y un desconocido.

"Ser consciente de la propia ignorancia es un gran paso hacia el saber."
Benjamin Disraeli.

A medida que sucede la gran experiencia de la vida, notamos que las frases más excitantes no son un "lo he logrado" sinó un inquietante "¿qué tal si…?". Vencer la ignorancia y comprender el cómo y el porqué de las cosas es un ejercicio que todos hemos practicado alguna vez.

Mientras charlaba con Jeffrey Borbón, el director de la revista, acordamos hacer un artículo para mostrar los sistemas MACINTOSH y BSD. El principal problema era que yo no conocía bien estos sistemas, había trabajado anteriormente con versiones de FreeBSD pero nunca me había sentado al frente de una power pc. Me tocó pues hacer la tarea y experimentar con estos sistemas operativos; la experiencia fue algo compleja y de mucho aprendizaje pero bastante entretenida :), pero no nos adelantemos y empecemos por el principio… La familia BSD: MAC OSX, FreeBSD, OpenBSD, NetBSD.

Nombre del sistema operativo: **BSD** (FreeBSD, OpenBSD, NetBSD)
Costo aproximado de instalación: $0
Tipo de licencia: **Licencia BSD**
Requerimientos mínimos del sistema: DIFERENTES ARQUITECTURAS. (De acuerdo a la distribución BSD que desee usarse)
Beneficios extras: distribuciones muy estables
Contras: desarrollo pausado

BSD (Berkeley Software Distribution) fue desarrollado por la Universidad de Berkely en California, USA.

AT&T y los Laboratorios Bell habían autorizado a varias universidades para que utilizaran el código fuente de Unix. Años más tarde estas concesiones otorgadas por AT&T fueron retiradas y la Universidad de Berkeley tomó la decisión de continuar con el proyecto, lo que se llamaría BSD (BERKLEY SOFTWARE DISTRIBUTION).

Cuando hablamos de software libre, siempre pensamos en la FSF (Free Software Foundation), su proyecto GNU y su licencia GPL. La licencia BSD tiene por mucho todas las ideologías del software libre e inclusive es aún más permisiva que la GPL. Básicamente la diferencia que existe entre las dos licencias es que la licencia BSD permite que su código sea utilizado por sistemas propietarios.

Es importante aclarar ciertas diferencias de BSD con GNU/LINUX; BSD es un OS que se basa en 4.4BSD-Lite. Su código es, en efecto, un legado del primer UNIX. Sin embargo, GNU/LINUX es un clon de UNIX, es decir, un sistema tipo UNIX.

Estilo libre y propietario, con aumento.

El diablo, la manzana y un desconocido.

BSD es un OS bastante desconocido en la comunidad libre debido a los inconvenientes que experimentaba con las licencias del viejo UNIX. Gracias a esto, su desarrollo de hardware ha sido mucho más lento que otros sistemas libres de código abierto.

Una característica de los sistemas BSD es que estos son desarrollados y mantenidos por un único grupo de personas. La mayoría de las versiones son estables y no tienen la ideología "libéralo rápido" de Linus Torvalds.

Existen tres versiones de BSD que son bastante importantes e interesantes. Estos son:

OpenBSD: Esta distribución de BSD esta totalmente enfocada en seguridad. Sostiene las ideas del código abierto y revisiones rigurosas del mismo. Esta distribución puede jactarse de ser uno de los sistemas operativos -si no el más- seguros del mundo.

NetBSD: Antes de empezar a hablar de la familia BSD ilustré la información general de estos sistemas. En ella dice que BSD corre en cualquier ordenador existente, pues esta distribución nos da la razón con su flamante lema "Of course it runs" (por supuesto que ejecuta). NetBSD es portada a numerosos sistemas con toda clase de arquitecturas, desde PDAS y teléfonos celulares hasta supercomputadoras de la NASA en su exploración espacial.

FreeBSD: Aunque en sus principios FreeBSD estaba enfocada a los procesadores 386, hoy ha sido portado a numerosas arquitecturas. Su ideología es ofrecer alto rendimiento y facilidad de uso al usuario final. No es difícil entender por qué es la distribución con mayor número de usuarios. Inclusive es en esta distribución en la que se basa el BSD "Darwin" de MAC OS X.

UN POCO DE FreeBSD Y GNU/LINUX

El proceso de instalación de FreeBSD es un poco complejo y es aconsejable que sea realizado por personas con algo de experiencia o que tengan la intención y paciencia para leer, aprender y encontrar soluciones. El proceso está basado en scripts y permite configurar y administrar los discos de una manera muy completa, configuración de "slices", entre otros.

Les recomiendo este vínculo para obtener más información:
http://www.freebsd.org/doc/es/books/handbook/install-start.html

Como pueden darse cuenta la mayoría de la información se encuentra en el sitio oficial de FreeBSD y esto es algo muy interesante.

FreeBSD tiene una buena compatibilidad con binarios de GNU/Linux y por tanto es posible correr la mayoría de los programas de GNU.

El rendimiento de FreeBSD en comparación con el de GNU/Linux no es muy diferente (al menos después de la aparición del kernel 2.6). La diferencia radica en la estabilidad garantizada de FreeBSD con hardware soportado de acuerdo a las listas de compatibilidad. Las facultades de este sistema operativo son más importantes si pensamos en servidores, donde las facilidades para compilar código y la estabilidad del sistema es un punto crítico. Para un usuario de GNU/Linux no sería muy complicado operar este sistema, los mismos comandos con una que otra opción diferente.

En mi concepto las diferencias entre la comunidad GNU/Linux y los FreeBSD radican en que el trabajo esté centralizado o descentralizado. Algunos piensan que el trabajo centralizado es una ventaja para los sistemas, otros pensamos que el modelo de desarrollo tipo bazar (como lo definió Eric S. Raymond) es el más adecuado para desarrollar nuevas tecnologías. Algo es muy claro, este sistema operativo está lejos de ser un sistema secundario y muchas personas lo consideran su primera opción a la hora de configurar servidores de alto rendimiento.

Nombre del sistema operativo: **MAC OS X TIGER**
Costo aproximado de instalación: € 129
Tipo de licencia: **Licencia propietaria MAC OS** X
Requerimientos mínimos del sistema:
- Procesador PowerPC G4, G5 y ahora procesadores Intel.
- FireWire incorporado
- Se requiere una unidad de DVD para la instalación
- RAM física de al menos 256 MB

El diablo, la manzana y un desconocido.

Beneficios extras: Soporte MACINTOSH para sus equipos
Contras: Es difícil conseguir los dispositivos en Colombia y su costo es elevado.

Esta importante compañía del mundo informático, nace en 1975 cuando sus fundadores Steve Wozniak y Steve Jobs venden respectivamente una calculadora HP-65 y un WV Ómnibus para financiar su proyecto "Apple". Con varias dificultades atribuidas a la inexperiencia de sus fundadores, Apple consigue el símbolo de la manzana multicolor y una importante inyección de capital por parte de un tercero.

La "manzana" Apple nos ofrece varios modos de instalación para iniciar MAC OS X "tiger". Si tenemos una versión mas viejita del sistema, simplemente insertamos el DVD y corremos el instalador (en caso de no tener DVD es necesario pagar unos dólares extra por los discos de instalación en CD), este reiniciará el ordenador para comenzar el proceso de instalación. Otra posibilidad es empezar desde el DVD presionando la tecla Apple-C en el inicio. Cualquiera de las dos vías nos llevará al mismo punto. Es posible hacer una actualización, copiar en otro sector del disco o hacer un formateo total. El proceso de formateo es bastante rápido, y la instalación igual, aunque depende del tipo de procesador y la máquina que estemos utilizando. A mi me tomó aproximadamente 45 minutos para que el sistema reiniciara una sola vez antes de entrar al sistema operativo.

Mi impresión general cuando inicié el "tiger" por primera vez fue "wow, esto si que se parece al gnomo (GNOME) de GNU". Después me enteraría de las razones: MAC OS X es en realidad un "hijo" más de Unix. En realidad este sistema operativo tiene sus raíces en FreeBSD, o mejor en 4.4 BSD, por lo tanto muchas de sus características son en realidad extraídas de las librerías del software libre (como el sistema X11).

El sistema es bastante estable y rápido. La compatibilidad con el hardware es excelente. El hardware MACINTOSH se ha caracterizado por tener un excelente desempeño y un buen diseño en el manejo de memorias y buses. Las arquitecturas de sus procesadores son basadas en un conjunto de instrucciones más pequeño (RISC) y además, sus implementos en los procesadores son excelentes. Hoy es posible encontrar máquinas ensambladas con procesadores fabricados por Intel y los poderosos POWERPC G4 y G5.

La cuestión que desfavorece al sistema son las triquiñuelas que deben hacerse para correr aplicaciones GNU (por lo menos es posible si uno se sienta a trabajar con el equipo, y leer aquí y allá) y las licencias que acompañan dicho OS. La mayoría de programas para MAC son propietarios y esto se refleja en lo cerrado que es este sistema operativo para ser modificable (al menos respecto al sistema). Este sistema no trae una suite ofimática, por ejemplo, y sin la posibilidad de instalar aplicaciones libres de fácil acceso uno se siente algo limitado para encontrar soluciones económicas que complementen el uso de la máquina.

Me llamó la atención el programa "DASHBOARDS". Este tiene la facultad de invocar miniplicaciones hacia la pantalla y esconderlas nuevamente. La idea es genial, aunque aún tiene sus fallas, en especial en la aplicación del clima que no tiene una buena base de datos de las diferentes zonas horarias (como la mayoría de estas aplicaciones). Esta aplicación guarda muchas similitudes con el equivalente GNU "SUPERKARAMBA" o "KARAMBA".

El SPOTLIGHT es una herramienta de búsquedas muy rápidas de todo tipo de archivos, programas, etc... Esta herramienta ha causado gran sensación dentro de los usuarios MAC. Aun no existen equivalentes completos de este programa en el software libre,

existen aproximaciones muy buenas como "KATAPULT" (que es una versión mas parecida al QUICKSILVER) y otras herramientas de búsqueda del shell de Linux/Unix. A modo de noticia, quisiera comentar que una herramienta más parecida está por aparecer en KDE 4, y ya se está desarrollando.

MAC OS X es un buen sistema, bastante estable, con un porcentaje alto de inmunidad a virus y otros ataques, es altamente integrado y viene con varias aplicaciones de fábrica bastantes útiles; pero, sigue teniendo los inconvenientes de un OS propietario, tanto para el sistema como para los programas que lo acompañan.

EL DESCONOCIDO HURD

De acuerdo a Wikipedia GNU/Hurd es un conjunto de herramientas tipo servidor que simulan un kernel Unix. Esta desarrollándose desde 1990 con las ideologías de la FSF (Free Software Foundation).

GNU/Hurd esta diseñado para correr sobre el microkernel MACH (el mismo que utiliza MAC). La gran diferencia es que Hurd cambia la ideología de un único servidor administrando el reloj del sistema, las memorias, los buses, etc. e intenta implementar múltiples servidores para controlar el acceso a todos estos recursos. El objetivo de este proyecto es mejorar las condiciones de otros núcleos para hacer un sistema más escalable, extensible y estable. La característica de utilizar múltiples servidores para manejar cada recurso del sistema le da en teoría, una enorme ventaja para la modularización del desarrollo y la depuración de componentes. No es necesario reiniciar nunca la maquina y pueden cargarse los componentes del núcleo sin afectar el trabajo de otros usuarios en la misma máquina.

Existen muy pocas implementaciones de este kernel. El popular Debian tiene una distribución basada en GNU/Hurd; sin embargo, todavía falta mucho para que veamos verdadera acción por parte de este núcleo.

El diablo, la manzana y un desconocido.

Quisiera aprovechar este punto para hacer un llamado filosófico. Esto lo hago a sabiendas de que algunos de nosotros no somos muy receptivos a "los cuentos" de la filosofía; sin embargo, debemos recapacitar sobre algunas concepciones que tenemos del negocio del software y la tecnología.

Además de los beneficios claros en seguridad y desarrollo que nos presentan las tecnologías abiertas, el punto clave no es otro que la libertad. Cuando tomamos la decisión de utilizar sistemas propietarios estamos alquilando el uso de nuestro sistema a un tercero que administrará y regulará lo que yo puedo hacer con el.

Con las alternativas de sistemas abiertos tenemos un control mayor sobre nuestros sistemas, podemos ejecutar tareas con mayor libertad, realizar cambios o contratar servicios de personas que nos ofrecen la mejor alternativa, no por que son los únicos en el mercado sino por que pueden y son los mejores.

Algunas veces tendremos que esperar a que se desarrolle algún concepto que no existe en el software, otras será pertinente contactar al desarrollador y financiarlo económicamente para que incluya alguna prestación que aún no tiene instalado el programa; sin embargo, este aspecto es secundario cuando evaluamos el valor que tiene un programa hecho a medida y la concepción de invertir en investigación y desarrollo para nuestras empresas, en lugar de adquirir licencias repetitivas para cada una de nuestras máquinas.

Esto es lo más importante del software libre: la libertad para compartir información. Como lo diría IBM en una de sus propagandas acerca del negocio de GNU/Linux: "Perderse en el grupo, por el bien del grupo". En una época donde la información es el recurso mas preciado, uno debería asegurarse de tener la alternativa de acceder a esa información y apoyar las alternativas donde se comparta este objetivo.

No queda más que despedirme, espero hayan encontrado útiles estos artículos que cubrían los sistemas operativos más comunes (al menos una porción de ellos). En la próxima oportunidad abordaremos las alternativas a nivel de aplicación con estilo libre y propietario.

EL PROYECTO MAXIMA

Robert Dodier
Desarrolador Maxima

Entrevista a Robert Dodier, Realizada en el marco de la V Semana Linux de la Universidad Distrital (SLUD5).
Desarrollador del proyecto MAXIMA
robert.dodier@gmail.com

Essentia Libre: Agradecemos tu colaboración para desarrollar esta entrevista en torno a tu trabajo en el proyecto MAXIMA. Como es bien conocido a nivel académico MAXIMA es un motor de álgebra computacional muy completo que ha sido desarrollado desde hace unos años bajo el modelo del software libre. Ya con esto presente, surge la pregunta, ¿Cuál es el estado actual del desarrollo de MAXIMA?

Robert Dodier: En estos momento MAXIMA se encuentra en un estado de desarrollo a razón de que se están corrigiendo errores, ampliación de las interfaces, especialmente las de Internet y gráficas. Debido a esto MAXIMA no se encuentra en un estado final. Es posible su uso, pero a futuro será mucho mejor.

EL: A lo largo de las charlas impartidas en el marco de SLUD5 comentaste a la audiencia sobre el momento en que retomaste el proyecto a la muerte de su creador, ¿cuáles han sido tus motivaciones para continuar trabajando en el proyecto MAXIMA?

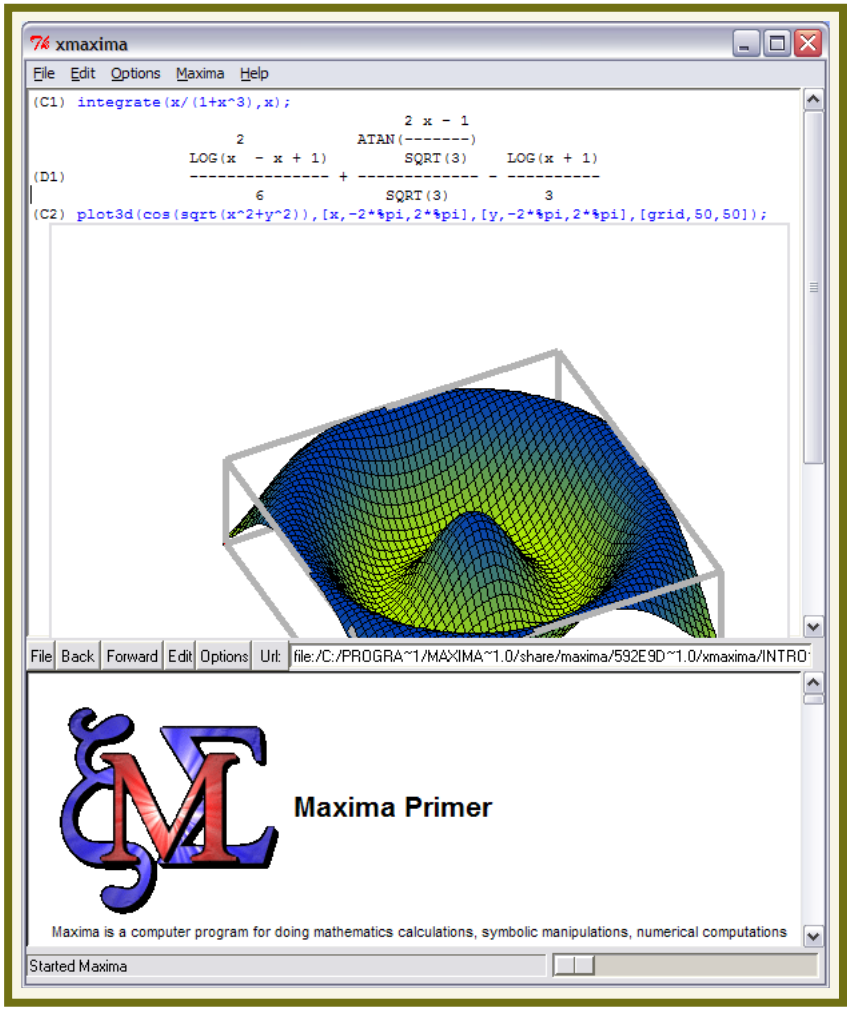
RD: El valor de MAXIMA en mi opinión lo encuentro en que es un sistema que expresa la manera de hacer matemáticas que uso de manera informal, especialmente la representación de cada cosa como una expresión y la actitud "LAISSEZ FAIRE", son los conceptos más básicos de MAXIMA y también son muy semejantes a la matemática y se acerca a la manera en la que suelo hacer matemáticas.

EL: ¿Cuál ha sido el papel de la comunidad en torno al desarrollo de MAXIMA? no es desconocido que has contado con participación de otros colaboradores en el proyecto lo que ha permitido, por ejemplo, la implementación del módulo de estadística por parte de un desarrollador español.

RD: MAXIMA es un proyecto grande ya que se trata de muchos sujetos matemáticos y actualmente hay personas que poseen intereses muy diversos. Gracias a esto es muy posible que muchas personas se interesen en determinados aspectos y así puedan trabajar en él. No es necesario, para estar en el equipo de desarrollo, contar con los mismos intereses. Por ejemplo, ahora mismo un grupo de desarrollo de MAXIMA, necesariamente, no debe saber mucho de otro grupo que también trabaja en el proyecto. Esta es una característica del equipo actual, y creo que vamos a ver que más personas se integran a los esfuerzos de desarrollo, gracias a la diversidad que permite MAXIMA.

EL: En tu opinión, ¿cómo ha sido el papel de la comunidad hispanohablante en el proceso de desarrollo, documentación, internacionalización y difusión del proyecto MAXIMA?

RD: Lo que he podido observar es que MAXIMA, posiblemente llegue a ser el sistema más popular en latinoamérica y en otros países hispanohablantes para hacer matemáticas generales. Por eso es importante que tengamos equipos de desarrollos hispanohablantes. Hasta el momento se ha podido visualizar que se ha iniciado la participación con la traducción de documentos, posteriormente amplían sus esfuerzos e inician trabajo con sujetos matemáticos generales. Un caso concreto es Mario Rodríguez en España, quien desarrollo el actual módulo de estadística. Debido a esto creo que hemos podido ver, poco a poco, el inicio del trabajo del equipo hispanohablante.



EL: ¿Y qué nos puedes contar del futuro de MAXIMA?

RD: Los desarrollos más inmediatos serán la interfaz de cuaderno y el desarrollo de algunos otros paquetes adicionales, pero a lo largo de un futuro inmediato y lejano, vamos a seguir corrigiendo errores y así poder continuar nuestra tarea en torno al proyecto MAXIMA.

EL: Nos inquieta un poco tu participación con otros proyectos de software libre, hasta el momento ¿has participado en algún otro diferente al proyecto MAXIMA, ya sea en la parte de desarrollo, la parte de documentación o asesorando algún otro proyecto?

RD: He usado software libre desde hace 10 años. He tenido un gran interés en esto, pero antes de que me integrara con el desarrollo de MAXIMA, no he participado en proyectos de software libre, con la única excepción de mi proyecto personal RISO que es un software para trabajo con redes bayesianas heterogéneas. MAXIMA es el primer proyecto grande en el cual he trabajado.

EL: ¿Te ha llamado la atención trabajar con algún otro proyecto diferente a MAXIMA?

RD: Un proyecto en el cual he pensado, es un sistema para análisis de decisiones. Este proyecto puede ser un paquete adicional para MAXIMA, que permitirá la realización de inferencias probabilísticas y también los combina con funciones de utilidad. Esto sería un nuevo funcionamiento con nuevas capacidades para MAXIMA, ya que sería un paquete o proyecto por si mismo. Es más o menos integrar un proyecto de análisis de decisiones con MAXIMA.

EL: Finalmente, ¿Qué imagen te llevas de la comunidad de software libre colombiana?

RD: Es grande el interés de las personas por el software libre. He notado en las personas que he podido conocer que son muy capaces y podrían perfectamente colaborar con el desarrollo de MAXIMA, como con las interfaces y la creación de nuevas funciones para integrar con MAXIMA. ■

Por Lorena Giraldo G.
Ingeniera de Sistemas y Telecomunicaciones
Ciudad Autónoma de Buenos Aires – Argentina

Argentina y el Software Libre

Hablar de Software Libre es hablar de sociedad, de libertad, de comunidad, de libre expresión…

¿Qué tiene que decir Argentina frente a todo esto?

Pues mucho, en la actualidad muchas personas que acceden a Internet en Argentina, sin darse cuenta están utilizando aplicaciones libres tales como: una suite ofimática, un navegador de Internet, un cliente de chat, todos estos basados en Software Libre lo que sin duda revela un gran avance.

"Hasta el año pasado el Software Libre ocupaba un 22% del mercado Argentino, según datos de la Consultora IDC. Esta misma consultora realizó una nueva investigación donde los resultados indican que hasta 2009 la implementación de aplicaciones de sistemas abiertos aumentará en un 26% hasta ocupar dos tercios del mercado." @

Tanto el Sector privado comercial como el público están adoptando Software Libre, esto es debido a que muchas empresas se están dando cuenta que el uso de aplicaciones libres les da una ventaja no sólo a nivel monetario sino a nivel de seguridad, flexibilidad y sobre todo legalidad.

"Un estudio realizado por la consultora *Trends Consulting* entre 120 grandes y medianas empresas dedicadas a la industria, servicios de comunicación, finanzas y comercio, indica que el 60% de los encuestados ya utiliza Software Libre en aplicaciones de misión crítica de su negocio. La principal razón declarada para implementar este tipo de programas es la reducción de costos (77%), en primer lugar, y la flexibilidad de adaptación que permite el sistema, en segundo término." @

¿Qué dice el Estado -Léase Administración Pública- en Argentina?

La Administración Pública ha estado presente en varios de los eventos que se han realizado con respecto a la tecnología y el Software Libre, dando como precedente el hecho que "En el marco del Primer Congreso Nacional de Software Libre realizado en el año 2004, uno de los integrantes del comité de Gestión Informática de la Administración Pública informó que el 90% del software que utilizaban era ilegal." @

El estado, ante todo esto se ha dado cuenta que no es sólo una cuestión financiera, sino de beneficio tecnológico, independencia frente a software privativo y sobre todo de seguridad nacional.

A tiempo presente, ya son varias las Administraciones Públicas que decidieron migrar a Software Libre, tales como: la provincia de La Pampa, Neuquén, Buenos Aires, Córdoba, Jujuy, Santiago del Estero, Tierra del Fuego, Tucumán, Mendoza y Santa Fe, con la experiencia del Programa Munix* en la ciudad de Rosario.

En el mismo año de 2004 en una provincia llamada Santa Fe, a unos cuantos kilómetros al norte de Capital Federal, se desarrolló la ley No. 12.360 que dice:
"Se dispone que los Poderes Ejecutivo, Legislativo y Judicial, los organismos descentralizados y las empresas donde el Estado Provincial posea mayoría accionaria emplearán en sus sistemas y equipamientos de informática preferentemente Software Libre, quedando en manos de la Autoridad de Aplicación (Ministerio de Hacienda y Finanzas) la determinación

de los casos en que podrá utilizarse software propietario. Esta ley dispone que el ahorro que signifique la utilización de Software Libre se destinará, a través de la Dirección Provincial de Informática, a la capacitación del personal provincial en la utilización de los nuevos programas. ” @

¿Y los proyectos y/o desarrollos?

A raíz de la implantación del Software Libre en la Administración Pública con el Proyecto “Munix”, se empezaron a desarrollar algunos otros proyectos de todo tipo, es el caso de **Ututo** y uno de sus hijos **UTUTO XS GNU/Linux**, una distribución de Software Libre que brinda facilidades para radios comunitarias, transmisiones de televisión y varios proyectos más para movimientos sociales, desarrollado por un grupo de 70 personas pertenecientes a las organizaciones Solar www.solar.org.ar e Hipatía.

Otro proyecto es el **OpenHIS** que significa Entorno Abierto de Información para Salud **(Open Health Information System Environment)** y esta diseñado con **HOME (Hospital Object Model Environment)**, un nuevo modelo de objetos para el análisis y creación de sistemas hospitalarios. Este sistema está siendo desarrollado por el grupo BioLinux http://www.biolinux.fac.org.ar. Este grupo ya lleva desarrollado otro tipo de proyectos tales como: Salux, OpenCare y Care2x-Ar.

Plumíferos el cual es un largometraje de animación en 3D desarrollado con Software Libre, más específicamente con Blender.

Musix es un Sistema Operativo 100% libre destinado a músicos, técnicos sonidistas, DJs, y usuarios en general: una enorme colección de programas **libres** que pueden usarse como alternativa a Windows.

Proyecto **Dogo** tiene como objetivo generar un software que permite reemplazar herramientas de seguridad, que además de ser importadas, exigen contar con hardware más potente. El propósito es extender la protección de las redes mediante la simplificación del sistema y el resultado será un software de código abierto dirigido a brindar seguridad en redes y disponible para uso abierto e investigación en el ámbito universitario y privado.

Argentina y el Software Libre

Finalmente, el Software Libre tanto en Argentina como en el mundo cada día está captando más atención de la gente y esto se debe a la amplia difusión que se le está dando, tanto parte de las comunidades de usuarios como de aquellas personas que contribuyen con su desarrollo. En Buenos Aires, Capital Federal existe varios Lugs que ayudan con dicha disufión, tal es el caso de Cafelug, Solar, LugFI , entre otros.

Los eventos más importantes que se celebran en Argentina son: Las Jornadas Regionales de Software Libre que cada año se realizan en una ciudad diferente, el FLISOL; el Cafeconf, un evento liderado por el grupo de usuarios Cafelug, el cual se realiza en Capital Federal y donde participan los lugs de todo Argentina; las CTT que son unas charlas técnicas y están abiertas para cualquier persona que quiera asistir.

Cafelug - Cafeconf 2005

Por último y ya para terminar, existen muchísimos más proyectos desarrollados y en vía de desarrollo, pero para adentrarme en estos temas quise como primera medida contar qué ha pasado en Argentina referente al Software Libre, qué se tiene ahora y lo que nos depara el futuro. ■

Así que, en nuestra próxima edición entraremos más a fondo en cada uno de los proyectos... “Be Open Be Free”.

Fotografía: Sebastián Panigazzi

Miembro Oficial del Cafelug (Capital Federal Lug) www.cafelug.org.ar

http://www.barrahomebarrasepa.com.ar/sepa/htdocs/spgm/

* Munix: consiste en la implementación de Software Libre en todas las áreas municipales, con el objetivo de reducir costos en licencias, regularizar el uso del software licenciado, unificar características del equipamiento y software del municipio y centralizar su administración garantizando su uso adecuado.

Por: Robinson Andrés Palacios
GEHMA

Pyslide para principiantes.

Seguramente son pocas las personas que conocen **Pyslide** y aún menos aquellas que logran dominarlo por completo. Es por esto que en esta ocasión dedicaremos este artículo a mostrar los elementos básicos que esta potente herramienta nos ofrece.

Pyslide es una aplicación desarrollada en Python que nos permite realizar presentaciones de diapositivas con una alta calidad visual, basándose en archivos XML y CSS. A pesar de que en ocasiones puede parecer un poco complejo y requiere de algunos conocimientos técnicos importantes, Pyslide es una excelente alternativa para aquellos que gustan de la total personalización de sus presentaciones y , claro está, para los amantes de Python.

Antes de empezar debemos asegurarnos de tener instalado el paquete Pyslide en nuestro computador, lamentablemente para algunos, este paquete se encuentra disponible especialmente para el sistema operativo GNU/Linux y su distribución Debian.

Para instalarlo de manera sencilla a nuestro computador lo ideal es tener instalado Debian, tener conectividad a internet y configurar las fuentes del APT con un servidor Debian de nuestra elección.

Lo primero: archivo XML y CSS

Debemos empezar por tener claro cómo y para qué usamos los archivos XML y CSS de nuestra presentación. Básicamente utilizamos el archivo XML para formar la estructura de nuestras diapositivas, mientras que el archivo CSS es usado para darle las propiedades de estilo (color, tamaño, aparición, ...) a todos los objetos de un tipo especifico, incorporados en la diapositiva. A medida que vayamos avanzando nos podremos dar cuenta exactamente de cómo funcionan estos archivos.

Estructura básica

La estructura de nuestra presentación se define en el archivo XML. Es por esto que debemos crear un archivo en texto plano (usando Emacs, Gedit, Kwrite,...) y guardarlo con la extensión .xml. Por ejemplo, para nuestro caso vamos a crear y guardar nuestro archivo con el nombre *presentacion.xml*. En este archivo es donde vamos a empezar nuestro trabajo.

Para este caso, las etiquetas son palabras o instrucciones colocadas entre los signos < y > , usadas para definir las características y propiedades de un tipo de elemento dado, y que en ocasiones se encuentra delimitado por dos etiquetas, una de apertura y otra de cierre señalada por el signo / después del signo <.

Se empieza por colocar la etiqueta <presentation> en la primera línea de nuestro archivo, cerrándola con la etiqueta </presentation>, así:

```
<presentation>

</presentation>
```

En medio de estas etiquetas podemos empezar colocar una primera, de varias diapositivas, usando la etiqueta <page> y dentro de nuestra diapositiva colocaremos un primer grupo con la etiqueta <group>; los grupos son un conjunto de elementos que van a aparecer en cada diapositiva después de realizado un click o digitado la tecla ENTER. Pueden haber varios grupos

Pyslide para principiantes.

en una misma diapositiva. Debemos tener muy en cuenta que los grupos van dentro de las diapositivas y estas a su vez están en medio de nuestra presentación y es por esto que es es sano sangrar las instrucciones con el fin facilitar el trabajo y la lectura de nuestros archivos, así:

```
<presentation> <!--Inicio Presentación. -->

   <page> <!-- Primer diapositiva. -->

      <group> <!-- Primer grupo de la
      primer diapositiva. -->

               <!-- Elementos contenidos.
                (Texto, imágenes, ...). -->

      </group> <!-- Fin primer grupo. -->

      <group> <!-- Segundo grupo. -->

               <!-- ... -->

      </group> <!-- Fin segundo grupo. -->

    </page> <!--Fin primera diapositiva.-->

    <page> <!-- Segunda diapositiva. -->

      <group> <!-- Primer grupo. -->

      </group> <!-- Segundo grupo. -->

    </page> <!--Fin segunda diapositiva.-->

</presentation> <!-- Fin presentación. -->
```

Comentarios:
Para poner comentarios en nuestros archivos XML debemos usar la instrucción <!-- antes del texto comentariado y finalizarlo con -->. Estos comentarios no son interpretados por nuestra aplicación.

En las lineas anteriores hemos creado dos diapositivas, la primera contiene dos grupos, es decir, que tendremos un contenido al iniciarla y el segundo al hacer click, la segunda diapositiva sólo tiene un grupo.

Es importante agregar la etiqueta de estilo a nuestra presentación. En esta etiqueta vamos a relacionar el archivo XML que contiene la estructura, junto con el archivo CSS que contiene el estilo, así:

```
<presentation> <!-- Inicio Presentación. -->

  <style source="estilo.css"/>
  <!-- Estilo para la presentación. -->
  ...
  ...

</presentation> <!-- Fin presentación. -->
```

donde estilo.css es el archivo creado en el cual colocaremos las características de los elementos a incorporar.

Teniendo en cuenta la estructura anterior, podemos empezar a agregar a cada grupo los componentes necesarios para la presentación.

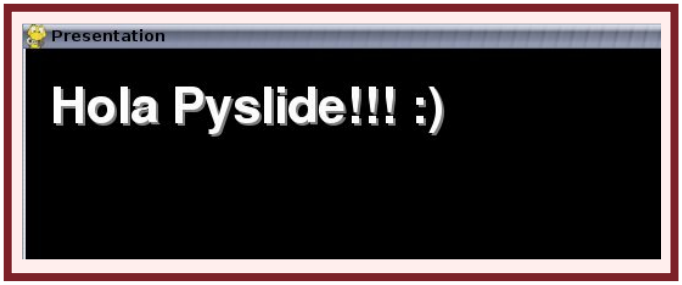
Para ejecutar pyslide abrimos un interprete de comandos de nuestro sistema y ejecutamos la instrucción: `pyslide presentacion.xml` donde presentacion.xml es el nombre de nuestro archivo XML, teniendo en cuenta que nos encontremos en el mismo directorio donde se encuentran los archivos XML y CSS.

Texto

Para agregar texto a nuestra diapositiva debemos usar la etiqueta <text> en el archivo XML en el grupo indicado, de la siguiente forma:

```
<presentation>

   <style source="estilo.css"/>

   <page>

      <group>

         <text> Hola Pyslide!!! :) </text>
            <!-- Texto -->

      </group>

   </page>

</presentation>
```

Pyslide para principiantes.



Posición:
La pantalla usada en pyslide tiene un tamaño de 1000 unidades en sentido horizontal (x) y 750 en sentido vertical (y), la posición empieza desde la posición 0,0 en la esquina superior izquierda y finaliza en la esquina inferior derecha.

En primer lugar debemos especificar el elemento al que vamos a establecerle las características. En este caso es el objeto text y a continuación entre corchetes establecemos los atributos a modificar: el tamaño (font-size), color (color) y posición (xy), como a continuación se muestra:

```
text{

   font-size: 100;
   color: darkblue;
   xy: 100,50;

}
```

Color:
Los colores en Pyslide se pueden establecer por medio del nombre del color o del código RGB del mismo.
Para ver los colores disponibles debemos usar la instrucción:

```
pyslide --color-names
```

Debemos tener muy en cuenta la necesidad de terminar, sólo en este archivo, con un punto y coma (;) cada instrucción.

Las propiedades especificadas en el archivo CSS afectarán todos los objetos de ese tipo en nuestra presentación. Si queremos darle características especificas a sólo un texto, podemos usar la etiqueta <text> de la forma siguiente:

```
<text font-size="100" color="darkblue"
xy="100,50"> Texto independiente </text>
```

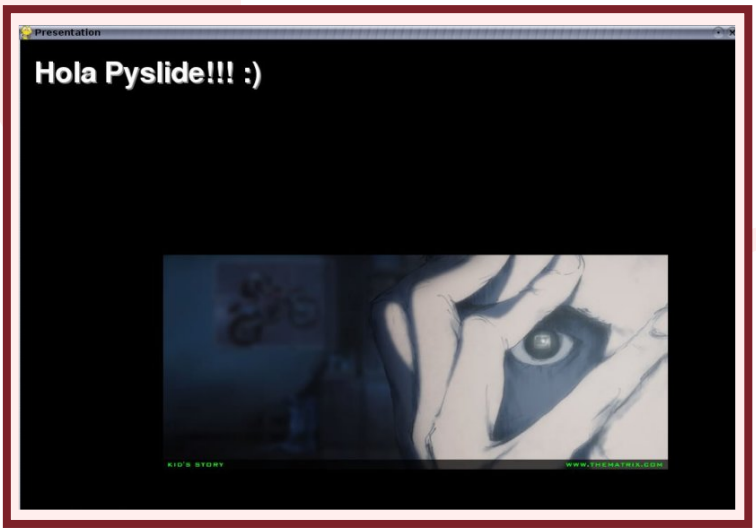
Con claridad en los procedimientos para los cambios de propiedades del texto va a resultar mucho más sencillo trabajar con los demás elementos.

Imagen

Para introducir una imagen usamos la etiqueta <image> en la cual podemos especificar el archivo a introducir y la posición de dicha imagen, así:

```
<presentation>

   <style source="estilo.css"/>

   <page>

      <group>

         <text> Hola Pyslide!!! :) </text>
            <!-- Texto -->
         <image src="imagen.png"
            xy="200,300"/> <!--Imagen en
            posición 200,300-->

      </group>

   </page>

</presentation>
```

Donde imagen.png es el archivo de imagen a introducir y 200,300 es la posición de la imagen en la diapositiva. Es muy importante tener el archivo de la imagen en el mismo directorio donde se encuentran los archivos XML y CSS de la presentación.



Pyslide para principiantes.

Listas

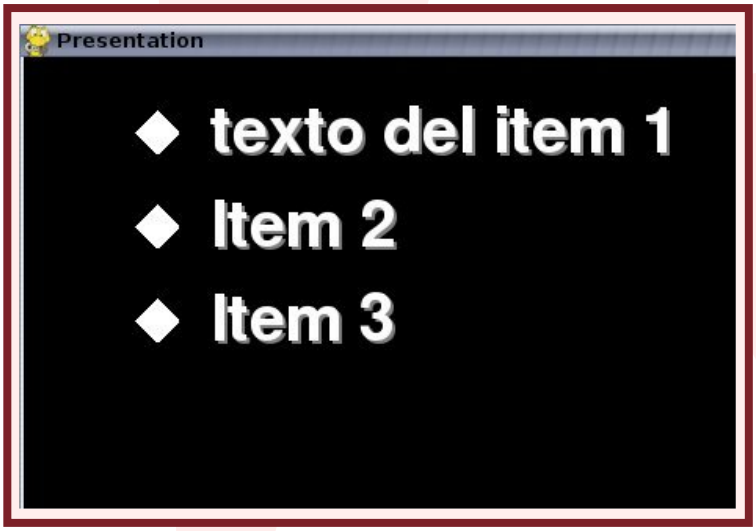
Los items que componen las listas están definidos por la etiqueta <list> así:

```
<presentation>
   ...
      <list> texto del item 1 </list>
      <list> Item 2 </list>
      <list> Item 3 </list>
  ...
</presentation>
```

Por defecto las listas vienen con una viñeta en forma de rombo y un color de texto blanco. Para cambiar estos valores podemos colocar el archivo estilo.css los siguientes valores:

```
list{

   font-size: 100;
   color: yellow;
   list-type: square;

}
```

Las viñetas, junto con el rombo, pueden ser cuadrados (square) y círculos (circle).



Fondos

Para poner un fondo a una diapositiva podemos usar la etiqueta page de las siguientes formas:

```
<page bg="imagen.jpg">  <!-- Imagen de
fondo expandida. -->

<page bgcolor="white">  <!-- Fondo color
blanco. -->

<page bggrad="white-black">  <!-- Fondo
color degradado de blanco a negro. -->
```

Opciones de la presentación:

Para cambiar el tamaño a la presentación:

```
<presentation size="800x600">
```

Agregar un título en la cabecera de la presentacion:

```
<presentation caption="Presentacion">
```

Para adicionar un fondo a toda la presentación:

```
<presentation bg="imagen.png">
<presentation bgcolor="white">
<presentation bggrad="red-blue">
```

Animaciones

Además de cambiar características a los objetos podemos realizar animaciones de aparición a cada uno. Para ello primero debemos definir la función en nuestro archivo CSS. En nuestro caso vamos a crear una función de aparición en la cual el objeto aparezca de la nada, así:

Pyslide para principiantes.

```
.aparecer{

   open-type: full;
   open-time: 20;

}
```

en donde la instrucción open-type establece la forma de aparecer, ya sea completa (full), en sentido vertical (vertical) o sentido horizontal (horizontal), y la instrucción open-time establece cuánto tiempo debe tardar en aparecer, el cual esta dado en deci-segundos. A continuación asignamos al objeto que vamos a animar, la función creada por medio de la instrucción effect-alpha:aparecer; colocándola donde cambiamos las características del objeto. Por ejemplo, para el caso del texto tendríamos:

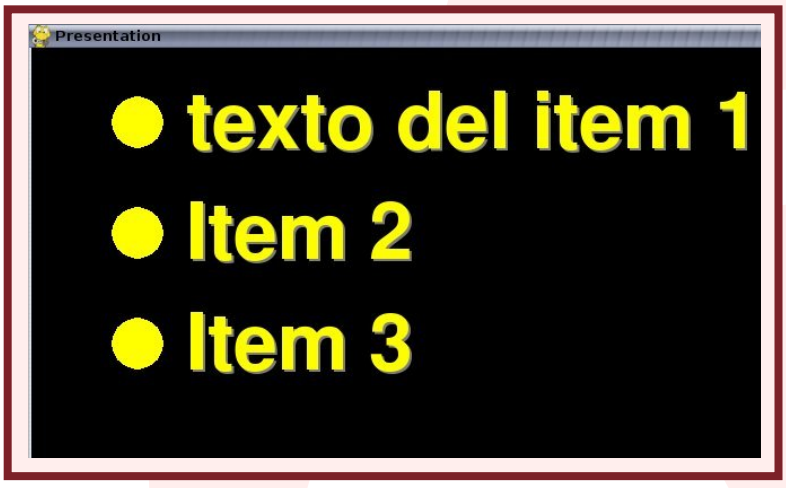
```
text{

font-size: 100;
color: darkblue;
xy: 100,50;
effect-alpha:aparecer;

}
```

Teclas importantes:

Q para cerrar presentacion.
F para pantalla completa.

Con estos conceptos básicos podemos empezar a realizar diapositivas con una buena calidad. Sin embargo explorando un poco más a fondo podemos potencializar nuestras presentaciones a niveles sorprendentes.

En nuestro próximo número ampliaremos un poco más acerca de esta herramienta, adicionando nuevas funciones de animación, viendo formas de definir objetos propios, recibiendo algunos tips para mejorar nuestras presentaciones y viendo los archivos XML y CSS de una presentación terminada de alta calidad.



Tiras Cómicas

Raulito el friki

http://recurrente.afraid.org

Informe especial
IV Foro Mundial de Conocimiento Libre
17 al 21 de Octubre 2006

Jeffrey Steve Borbón Sanabria
Director Essentia Libre
Miembro proyecto ACLibre
jeffto@aclibre.org

Gracias a los esfuerzos de comunidades de Software Libre venezolanas, especialmente SoLVe, se desarrolló en días pasados el IV Foro Mundial de Conocimiento Libre en la ciudad de Maturín, ubicada en el estado de Mónagas al oriente de la República Bolivariana de Venezuela. El marco del evento no encerraba sólo temas puramente técnicos y relacionados con Software Libre, sino que abarcaba un gran tema como lo es el conocimiento Libre , el cual aplica para tecnologías libres, contenidos abiertos y gestión del conocimiento en torno al trabajo comunitario.

En esta cuarta ocasión el evento contó con el apoyo de entidades del Estado como PDVSA, Conatel, el Ministerio de Cultura, el Ministerio de Ciencia y Tecnología, entre otros, dándole toda la vistosidad y seriedad del caso. A lo largo de toda la semana se presentaron gran cantidad de charlas, conferencias y talleres con la activa participación de la comunidad.

La inauguración del Foro

El día martes 17 de octubre en las horas de la noche y situados en la Universidad Bolivariana de Venezuela (UBV), se dio el inicio al evento con la participación de los representantes de los Ministerios de Ciencia y Tecnología, Ministerio de Cultura, patrocinadores, entidades relacionadas con las tecnologías libres y por supuesto las comunidades de Software Libre implicadas en la realización del foro. Es importante resaltar la emotividad de los discursos y palabras presentadas por los diversos representantes, cuyo enfoque principal fue el desarrollo tecnológico endógeno, la soberanía tecnológica y el sentido de comunidad que evoca la filosofía libre y como lo indicaba el eje central del evento, el conocimiento libre.

Posterior a esto, se presentaron varios actos culturales de danzas representativas de la cultura del estado de Mónagas, y finalmente con un bello acto de juegos pirotécnicos se dio oficialmente apertura al IV Foro Mundial de Conocimiento Libre.

Un gran evento con asistencia masiva

Complejo Cultural de Mónagas

En la mañana del miércoles 18, en el Complejo Cultural Mónagas, cuya inmensa estructura permitió en 6 salas diferentes realizar en manera simultánea alrededor de 70 conferencias, 5 paneles y la asamblea final llevada a cabo por SoLVe el último día del evento, arrancó en pleno el IV Foro Mundial de Conocimiento Libre.

Conferencias

La cuota colombiana fue el proyecto ACLibre con la conferencia "Un nuevo modelo educativo basado en tecnologías libres" abriendo el evento, donde se contó con buena participación de público, una constante que se presentó a lo largo de todo el evento para la mayor parte de las actividades.

Por otra parte, en las instalaciones de la Universidad Bolivariana de Venezuela, se realizaron los talleres que involucraron áreas como el desarrollo de software (Talleres de Perl, Librerías QT – KDE y Zope), diseño gráfico, ofimática, geomática y análisis de imágenes, todos estos impartidos usando software libre. Para muchas personas que asistieron a estos talleres fue muy particular la experiencia dado que algunos fueron impartidos por personas con gran trayectoria en el mundo del software libre como fue el caso del primer taller de Perl impartido por Randal Schwartz, quien es uno de los desarrolladores principales del proyecto o con el caso del taller de librerías QT y KDE impartido

por Aaron Seigo quien hace parte del grupo de desarrollo del entorno de escritorio KDE.

Conocimiento libre, sociedad y tecnología

Algo que destacó el evento fue la interacción de paneles enfocados a la parte social y comunitaria del conocimiento libre, como lo fueron las charlas

de Aarti Sethi (Proyecto Sarai) y Arun Madhavan (Free Software Fundation India) que cada uno de un modo diferente logró mostrar cómo la cultura libre se ha impuesto poco a poco en la India, todo gracias al trabajo comunitario y el deseo de preservar la cultura.

Pero, el anterior no fue el único caso donde no fue lo técnico y tecnológico el punto central, otras conferencias como por ejemplo: "Pensamiento nuevo en materia de derechos culturales e intelectuales" presentado por Lilliam Alvarez de Cuba o la charla presentada por Jordi Mas, acerca acerca del modelo económico que involucra el desarrollo y uso del software libre.

Adicionalmente la conferencia presentada por Bernardo González, acerca del proyecto Open Puppets logro cautivar a los asistentes con ingeniosos peluches representativos de mascotas del software libre y psicodélicos diseños, demostrando cómo se puede acceder a los nuevos usuarios, así dejando claro que en el mundo del conocimiento libre hay espacio para la diversión y la moda.

Una charla que llamó mucho la atención fue la presentada por Edgar Guzmán (Edgarin), proveniente de Guatemala, quien a sus cortos 15 años presentó con propiedad y destreza el sistema LTSP (Linux Server Terminal Project) en el cual viene trabajando desde hace unos cuantos años. Otro caso similar fue Alejandro Garrido (Mogaal), quien a sus 17 años lidera un LUG denominado CHASLUG y quien en su cuenta personal ya tiene dos paquetes desarrollados en lenguaje Perl en el proyecto Debian; él por su parte desarrolló un taller enfocado al desarrollo de aplicaciones usando Perl. Esto dejo claro que no existen requisitos o condiciones para hacer parte de la comunidad del conocimiento libre.

Como es bien conocido por muchos, el proyecto Debian cuenta con muchos fanáticos, pues para ellos se presentaron varias charlas enfocadas al uso, desarrollo, mantenimiento y proceso de este gran proyecto y finalizó con un Mini-DebConf dirigido con excelente participación por Ana Delgado, quien ferviente seguidora y colaboradora del proyecto Debian dejó en alto el importante papel de la mujer en el software libre.

En otras palabras se presentaron muchas opciones para escoger, a decir verdad, por momentos el tomar la decisión de la conferencia a la cual asistir era tarea no muy sencilla, a muchos les ocurrió el día jueves 19 cuando en la tarde a la misma hora se presentaron dos charlas de conferencistas internacionales (Randal Schwartz y Jeff Zucker), pero finalmente, eso demostró la diversidad que podemos encontrar en el conocimiento libre.

Mezcla Cultural

A lo largo de la semana se presentaron diversos momentos en los cuales fue posible compartir momentos de intercambio cultural y de ideas. No era extraño observar en una misma mesa nacionalidades diversas, pero todos relacionados bajo ideas como el software libre y la cultura libre.

El intercambio de experiencias fue enriquecedor para todos los asistentes que están iniciando en el mundo del software libre, por ejemplo, ya que personalidades que llevan 10 años o más trabajando en estas áreas aportaban ideas y hacían de estos, momentos enriquecedores. Visto más de cerca fueron estos los instantes en los cuales se generaban nuevas comunidades; un caso muy puntual fue la generación de la comunidad desarrolladores en español creada al finalizar el taller de las librerías QT y KDE, esta nueva comunidad apadrinada por el mismo Aaron Seigo (Desarrollador Kde) o los mismos Perl Monguers de Venezuela.

Las noches fueron los momentos en los cuales cada grupo, sin importar el hotel donde se hospedasen o el idioma en que se hablase, se discutían proyectos, se hacían planes para la creación de nuevos trabajos o simplemente al son de alguna bebida se tenía una buena charla acerca de la trascendencia que ha tomado el software libre en Venezuela. Esto fue algo que a más de un extranjero sorprendió, debido que gracias a la implementación del decreto 3390 el sueño de contar con el apoyo del Estado a las tecnologías libres y por supuesto al software libre, ya es una realidad.

La cueva de los Guacharos

El día final del evento (21 de Octubre), con la concurrencia de muy buena parte de los asistentes al foro, se realizó un tour turístico a una cueva ubicada a dos horas de Maturín en cercanías a un pueblo llamado Morichal. El lugar es llamado “La cueva de los Guacharos”.

En este interesante lugar fue posible observar de cerca unas extrañas aves nocturnas llamadas Guacharos, los cuales tienen sus nidos en el interior de la cueva y salen en las noches a comer fruta y que comparten su hábitat con murciélagos, cangrejos y algunos peces ciegos.

La experiencia de caminar en la penumbra contando sólo con una lámpara de gas y un muy preparado guía turístico, fue algo nuevo para muchos de los asistentes a la pequeña travesía. Posterior a la permanencia, a casi un kilómetro de distancia en el interior de la cueva y el retorno al exterior, se asistió a un restaurante donde al terminar se pusieron a prueba las capacidades musicales de varios de los invitados y asistentes al participar en un Karaoke donde se logro un momento de complicidad y amistad. Finalmente en las horas de la tarde, todos de regreso a sus respectivos hoteles o lugares de procedencia ya que muchos regresaron ese mismo día.

A futuro

La opinión a nivel latinoamericano acerca de la comunidad colombiana de software libre y conocimiento libre es muy positiva. En boca de una buena cantidad de asistentes quedó claro que los logros de la comunidad son muy significativos e invitaron a seguir en ese proceso. Se marcó la importancia de crear un ente administrativo y legal del software libre para la comunidad colombiana tal como desde hace unos años es Solar en Argentina o SoLVe y GLOVE en Venezuela.

Finalmente, como conclusión personal, es prioridad la consecución del apoyo estatal o de las diferentes gobernaciones e incluso del sector privado para poder llegar a realizar eventos de tal escala en Colombia, por que aquí también hay mucho para mostrar. ■

Fotografías:
Edgar Guzmán (Guatemala)
Luciano Bello (Argentina)
Jeffrey S. Borbón S. (Colombia)