CULTURA LIBRE

Número 2 ::

CONTENIDO

- Escribiendo documentos con latex
- Cronicas del comienzo del fin
- La revolucion del desarrollo web
- QT: C++ framework
- Todos los P2P en uno solo
- Enseñanza de GNU/Linux en Cuba
- Un vistazo a fedora
- Linux & Open source Mono
- Usando dialog y shell script

http://www.apglug.org.pe

Noticias

El viernes 29 a las 12 del día la FSF anuncia el lanzamiento oficial de la GPL v3, mas allá de la creación de una licencia mejorada, el proceso de sus borradores ha ayudado a resaltar detalles importantes para la comunidad de usuarios del software libre.
Con el lanzamiento de la GPL v3 veremos la extensión de nuevas defensas con respecto al software libre, estas defensas continuaran la larga historia de luchas contra los intentos de convertir al software libre en propietario.
http://donate.fsf.org/

Compiz Fusion: La fusion entre Bery y Compiz.
Beryl el fork de Compiz se han unido finalmente,

ambos proyectos que han dado una nueva imagen y funcionalidades 3D a nuestros escritorios se llamara "Compiz Fusion".
El nombre es el producto de la encuesta que se llevó a cabo en los foros de discusión del proyecto conjunto. Y muy pronto se eligira un logo que los identifique.
Ya es posible bajar los avances obtenidos de esta fusion.
http://gitweb.opencompositing.org

¿Internet fomenta la amistad?
El Proyecto Internet Catalunya(PIC), impulsado por la Universitad Oberta de Catalunya, ratifico que las nuevas tecnologias activan la sociabilidad, asi mismo internet es considerado como el elemento clave para la autonomía personal, política, económica y profesional.
Imaginan un mundo sin televisores, telefonos celulares, ordenadores ... peor aun, que no tengamos Internet.

¿Pingüinos Gigantes en Perú?

Una noticia publicada en la revista científica "Proceeding of the National Academy of Science", reveló que dos especies de pingüinos gigantes vivieron en las costas de Perú.
El gigantesco Icadyptes, habitó hace 36 millones de años las costas del sur de Perú, y median más de un metro y medio de estatura. En tanto, el Perudyptes, vivió hace 42 millones de años y tenía una altura cercana de 76 centímetros.
Este hallazgo contradice las hipótesis sobre la evolución de estos animales en el mundo. Además, los nuevos fósiles datan de uno de los períodos más cálidos de la historia.

Índice

Editorial 2
Escribiendo documentos con LaTeX 3
Crónicas del comienzo del fin 7
La Revolución del desarrollo Web 9
Qt: C++ Framework 11
Todos los P2P en uno solo 14
Enseñanza de Gnu/Linux en Cuba 16
Un vistazo a Fedora 7 18
Linux & Open Source MONO 20
Usando Dialog y Shell Script 22

Año 1 - Número 2 - Junio 2007

Editor
Luis Revilla Amezquita

Colaboradores
Alonso Cárdenas Marquez
Ayax Fernández Rosado
Cesar Vargas Deza
Felix Arismendi Quispichuco
José Balmaseda Novoa
Julio César Zevallos
Leonel Iván Saafigueroa
Paola Castillo Vizarres
Rony Yabar Aizcorbe
Steve Ataucuri Cruz

Diseño y diagramación
Ayax Fernández Rosado

Carátula
Abraham Montaño Mena

Dirección Web
revista.aqpglug.org.pe

Lista de correo:
revista@listas.aqpglug.org.pe

Editorial

Ya estamos en el segundo número de la revista, nos alegra de gran manera a todos los que conformamos este estupendo equipo, en este número tratamos algunos temas referente a lenguajes de programación muy difundidos y conocidos en el mundo del software libre; es asi, que tratamos de Rubi on Rails, Mono y la libreria dialog, para elaborar simpaticas ventanas en nuestra consolare.

Otros temas que tratamos y de gran importancia como la aplicacion de las tecnologías libres en la educación y un articulo de opinión muy interesante, sobre las patentes.

Nuestra meta es también que las herramientas que tenemos disponibles y que lo usamos en nuestro que hacer diario se ha utilizado también en sistemas propietarios, porque ahí esta presente el software libre y tenemos muchos ejemplos de su uso apache, joomla, Qt, latex, etc.

Mis palabras finales estan destinadas a agradecer a todas las personas que han colaborado: aportando artículos, tutoriales, artes gráficas, y demás tareas. Nuestro sueño se va realizando, ya que contamos con la colaboración de personas de Argentina, Cuba y de todas las latitudes de nuestro querido Perú; esperamos que en un futuro, más gente se decida a apoyar este esfuerzo y sigamos adelante; en mejorar y hacer crecer la revista que nació como una idea y hoy es una realidad.

El editor

Luis Revilla A.

ESCRIBIENDO DOCUMENTOS CON LaTeX

Escrito por: César Vargas

Latex es una implementación de Tex, originalmente creado para sistemas UNIX, pero que también está portado a sistemas como Windows, MacOS, DOS, Amiga y otros, se encuentra en todos las distribuciones de GNU/Linux, la uersión que trataremos será la pdfeTex 3.141592-1.21a-2.2 (Web2C 7.5.4)[1] una de las más usadas en la actualidad.

Muchas veces nos hemos visto decepcionados al momento de imprimir un trabajo, ya sea para la universidad o para el empleo y nos damos con la sorpresa de que todo lo que habíamos hecho no sale como queremos; los márgenes cambiaron como por arte de magia, el tipo de letra no es como el que se muestra en pantalla, los gráficos se imprimen sobre el texto o simplemente las tablas no aparecen, nos ponemos a pensar en la forma en que podemos conseguir que nuestro trabajo sea de una calidad profesional... pues, la solución es simple, hagamos uso de LaTeX.

Se preguntarán ¿Qué es LaTeX?

Pues les diré que LaTeX no es un procesador de textos (como podría pensarse), es un paquete que se encarga de preparar automáticamente un documento, darle formato y prepararlo para obtener una apariencia estándar y de alta calidad. LaTeX nació en el mundo UNIX, es por tal razón que se lleva tan bien en los sistemas GNU/Linux.

Para la preparación de un documento en LaTeX sólo necesitamos de un editor de texto, (si es posible que reconozca la sintaxis de LaTeX) el que prefieran, (particularmente uso el editor de texto Cooledit[2] (Fig 1), se puede utilizar VIM, Emacs, XEmacs, Kate, etc.), la ventaja de un editor así es que nos colorea los tags de LaTeX y nos permite analizarlo y verificar que no haya errores antes de "compilar" el documento.

Una de las ventajas principales es el tamaño de un archivo de LaTeX, varias veces más pequeño que el documento generado por un procesador de texto convencional como OpenOffice, StarOffice y ni que hablar de Microsoft Office.

Otra característica es su sintaxis, la cual trataremos a continuación, es muy sencilla, todas las instrucciones comienzan con un backslash "\", seguido de la instrucción; en LaTeX hay dos tipos de parámetros, los obligatorios, que se colocan dentro de llaves "{" y "}", y los opcionales, los cuales van dentro de corchetes "[" y "]".

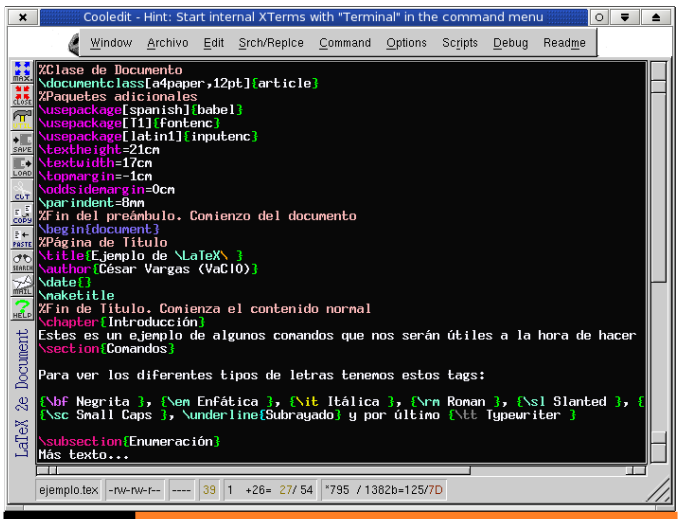


Figura 1 Archivo de LaTeX en editor Cooledit

El documento de LaTeX consta en sí mismo de dos partes: Preámbulo y Cuerpo.

Preámbulo

En el preámbulo se coloca la clase de documento que se realizará, los paquetes que se utilizarán, además de algunos parámetros adicionales para darle forma al documento, como por ejemplo los márgenes, tamaño de letra, etc.

Cuerpo

La segunda parte es el Cuerpo del documento, es decir, el contenido del documento que estamos redactando, aquí indicaremos algunos datos adicionales como el título, el autor, la fecha, el formato de letra, ya sea negrita, cursiva, itálica, etc; tamaño: tiny, script, normal, large, etc; también podemos indicarle a LaTeX que introduciremos una fórmula matemática, una tabla o un gráfico.

Para darnos una idea de como funciona haremos un ejemplo para familiarizarnos con este paquete y su sintaxis.

Ejemplo 1

```
---------- Empieza aquí ----------
%Cortar el texto y guardarlo en un archivo con la extensión .tex
%Clase de Documento
\documentclass[a4paper,12pt]{article}
%Paquetes adicionales
\usepackage[spanish]{babel}
\usepackage[T1]{fontenc}
\usepackage[latin1]{inputenc}
\usepackage[dvips]{graphicx}
\textheight=21cm
\textwidth=17cm
\topmargin=-1cm
\oddsidemargin=0cm
\parindent=8mm
%Fin del preámbulo. Comienzo del documento
\begin{document}
%Página de Título
\title{Ejemplo de \LaTeX\ }
\author{*Su nombre*}
\date{}
\maketitle
%Fin de Título. Comienza el contenido normal
Este es un ejemplo de algunos comandos que nos serán útiles a
la hora de hacer nuestro trabajo con \LaTex.
\section{Comandos}
Para ver los diferentes tipos de letras tenemos los siguientes
tags:
{\bf Negrita}, {\em Enfática}, {\it Itálica}, {\rm Roman}, {\sl
Slanted}, {\sf Sans Serif}, {\sc Small Caps},
\underline{Subrayado} y por último {\tt Typewriter}.
\subsection{Enumeración}
En esta sección veremos la forma de centrar, tamaño de letra
y enumerar algunos ítems...

Otros {\large parámetros } para darle más {\bf forma } al
documento


\begin{center}
Este texto irá centrado
\end{center}
\begin{enumerate}
\item{\bf Partes de} {\em \LaTeX}
\item{\bf Contenido}
\item[] Preámbulo
\item[] Cuerpo
\end{enumerate}

Para {\Huge terminar} {\bf Uds.} pueden agregarle el texto que
deseen y {\Large hacer} sus ejemplos en este excelente
paquete.

\begin{center}
\fbox{{\Huge \LaTeX}}
\end{center}

\begin{center}
\includegraphics{tuxt.eps}
\end{center}

\end{document}

---------- Termina aquí ----------
```

Una vez copiado el texto, lo guardaremos en un archivo de nombre ejemplo.tex para poder usarlo luego. Como se habrán dado cuenta en el ejemplo anterior se utilizan algunos comandos básicos para la elaboración de este documento.

Para ver el resultado de este archivo, si nos encontramos en una shell, simplemente tecleamos latex ejemplo.tex (asegurémonos de estar en la misma carpeta donde guardamos el archivo de ejemplo) como en el figura 2.

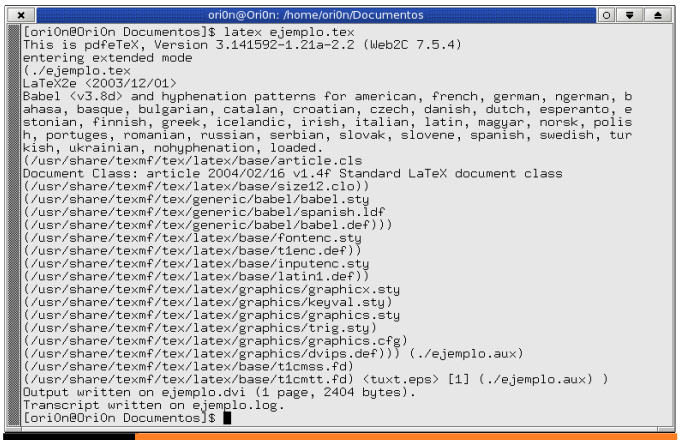
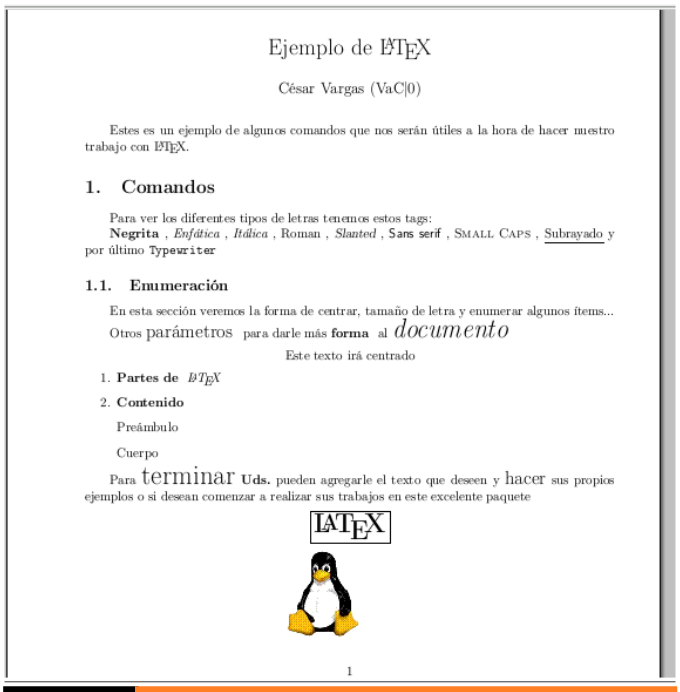


Figura 2 "Compilando" archivo LaTeX en Konsole

Esta es la salida (si no hay problemas o errores) de un documento "compilado" con LaTeX, el cual genera un archivo de extensión .dvi figura 3; el cual puede ser visualizado con cualquier visor de documentos (KDVI, Konqueror, nautilus, etc.), el resultado es impresionante. Pero ¿de dónde salió el formato?, ¿quién puso los títulos?, pues la respuesta es una sola LaTeX!!!.


Ejemplo de LaTeX

César Vargas (VaC|0)

Estes es un ejemplo de algunos comandos que nos serán útiles a la hora de hacer nuestro trabajo con LaTeX.

1. Comandos

Para ver los diferentes tipos de letras tenemos estos tags:
Negrita , *Enfática* , *Itálica* , Roman , *Slanted* , **Sans serif** , Small Caps , Subrayado y por último **Typewriter**

1.1. Enumeración

En esta sección veremos la forma de centrar, tamaño de letra y enumerar algunos ítems...
Otros parámetros para darle más **forma** al *documento*

Este texto irá centrado

1. **Partes de** LaTeX
2. **Contenido**

Preámbulo

Cuerpo

Para terminar **Uds.** pueden agregarle el texto que deseen y hacer sus propios ejemplos o si desean comenzar a realizar sus trabajos en este excelente paquete

LaTeX

1

Figura 3 Ejemplo.dvi

Hay editores como el de la figura 4, que cuenta con un script que se encarga de "compilar" el archivo .tex, este script nos mostrará la salida en una ventana emergente y si hay errores nos indicará en que línea se encuentra (Figura 5), haciéndonos la tarea un poco más rápida y sencilla.

En caso de querer hacer un documento en formato PDF directamente, simplemente escribimos en el prompt de la consola: "pdflatex ejemplo.tex"

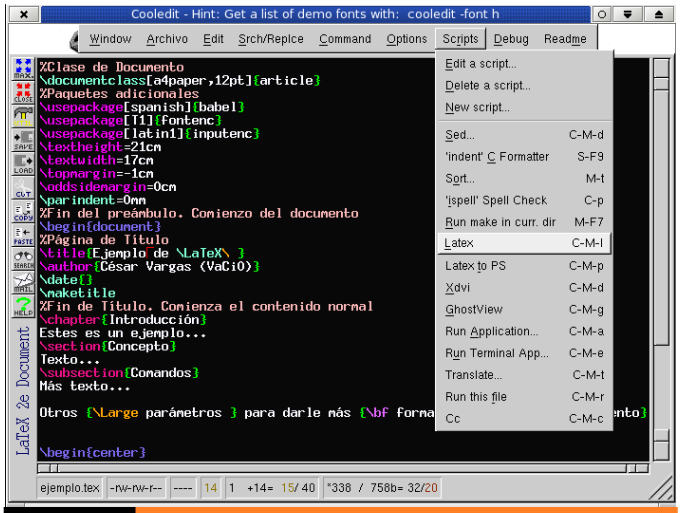


Figura 4 Script para "Compilación" de un archivo LaTeX

Comandos usados en el ejemplo

Como se habrán dado cuenta las instrucciones en el preámbulo indican que el documento es de la clase "article" (artículo), (en LaTeX hay 3 clases más usadas book, article y report, también existen la clase letter y slide, ambas requieren de un estudio especial y más profundo).

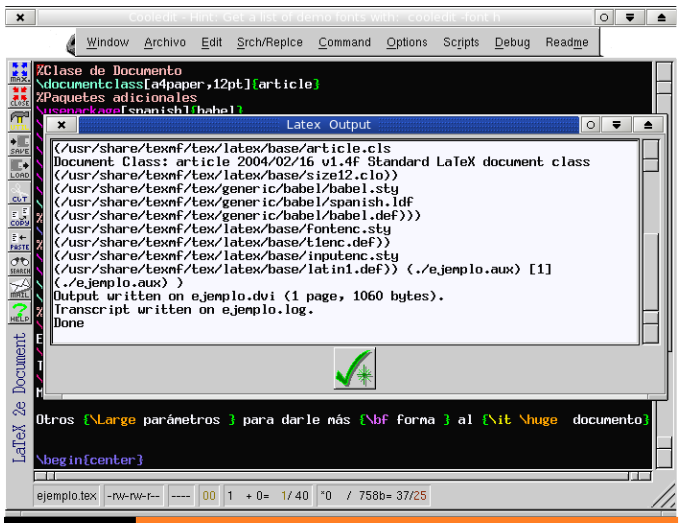


Figura 5 Salida de la "compilación" en el editor Cooledit

%Clase de documento: Los comentarios comienzan con el símbolo "%".

\documentclass: define la clase de documento.
[a4paper]: tamaño de papel en el que se imprimirá el documento.
[12pt] tamaño de la letra que se usará en el documento (por defecto 10pt).
\usepackage[spanish]{babel}: Usa la librería babel en español.
\usepackage[T1]{fontenc}: División correcta de palabras.
\usepackage[latin1]{inputenc}: Soporte completo para el dioma español.
\usepackage[dvips]{graphicx}: Soporte para la inclusión y manejo de gráficos.
\textheight=21cm: ancho del texto en la página.
\textwidth=17cm: largo del texto en la página.
\topmargin=-1cm: margen superior (por defecto 3 cm.).
\oddsidemargin=1cm: margen izquierdo (por omsión es 4.5 cm.).
\parindent=8mm: medida de la sangría (tabulación).

Hasta aquí la definición del preámbulo de nuestro documento.

Ahora veamos los tags usados en el cuerpo del documento:
\begin{document}...\end{document}: todo lo que vaya dentro de estos tags será lo que se imprima.
\title{...}: Título de nuestro trabajo.
\author{...}: Nombre de quien está desarrollando el documento.
\date{}: Este tag colocado de esta forma(con las llaves vacías) no imprimirá la fecha del documento. (Eliminen el tag si desean que aparezca la fecha)
\maketitle: ordena a LaTeX que imprima el título que le hemos asignado.
\section{..}: Inicia una sección de nuestro documento.
\subsection{..}: Inicia un subsección del documento.
(Formatos de texto en la tabla 1).
{\large ... } Asigna al texto entre las llaves un tamaño más grande del normal, es uno de los varios tamaños que podemos elegir para darle forma a nuestro documento.
(Más comandos para tamaños de letra en la tabla 2).
\begin{center} ... \end{center} todo lo que este encerrado entre estos tags, irá centrado en el documento.
\fbox{...} todo lo que se encierre en estos tags estará encerrado en un cuadro o caja.

Otros comandos

Para el manejo de gráficos es recomendable el uso del formato «.eps» (Encapsuled PostScript) o «.ps» (Post Script) con los cuales podremos obtener una mejor calidad al momento de imprimirlos. También se puede usar los formatos «.wmf» «.bmp», pero la calidad es muy baja.

En LaTeX, como en un lenguaje de programación, hay caracteres reservados (tabla 3), en caso de querer imprimirlos, se muestra el comando necesario para tal fin.

La inclusión de fórmulas matemáticas es uno de los puntos fuertes de LaTeX, los comandos son igual de sencillos, pero serán tratados en otra oportunidad.

Por el momento hemos hecho uso de algunos comandos que nos ayudarán mucho para poder realizar nuestros trabajos. Pero no crean que son todos los comandos o tags que existen, son muchos más, los iremos viendo poco a poco.

Tablas Anexas

Tabla 1: Tipos de letra

Comando
{\rm Roman }
{\em Enfático }
{\bf Negrita }
{\it Itálica }
{\sl Slanted }
{\sf Sans Serif }
{\sc Small Caps }
{\tt Typewriter }
{\underline Subrayado }

Tabla 2: Tamaños de fuente

Comando
{\tiny Tiny}
{\scriptsize Script}
{\footnotesize Foot}
{\small Small}
{\normalsize Normal}
{\large large}
{\Large Large}
{\huge huge}
{\Huge Huge}

Tabla 3: Sustitución de caracteres

Caracter	Comando	Significado
\	$\backslash	Usado para comenzar un comando
{,}	$\{$, $\}$	Usado para abrir y cerrar bloques de código
$	\$	Usado para abrir y cerrar el modo matemático
&	\&	Usado como tabulador(tablas)
_,^	_ , \^{}	Usado en exponentes y subíndices
#	\#	Usado en macros (parámetros)
~	\~{}	Usado para evitar cortes de renglón
%	\%	Usado para los comentarios

Enlances de Interés

(1) http://www.latex-project.org/ Descarga de la última versión de TeX
(2) http://www.ibiblio.org/pub/Linux/apps/editors/X/cooledit/!INDEX.short.html (Editor CoolEdit)

Para terminar esta entrega, sólo queda decirles que laTeX es uno de los paquetes que más nos puede ayudar a la hora de formatear nuestros documentos, la simplicidad de sus comandos y la potencia de su procesamiento son una ventaja sobre los editores o procesadores convencionales, en Internet hay cientos de páginas con tutoriales y trucos para hacer de Latex uno de nuestros preferidos.

Sobre el Autor

Cesar Augusto Vargas Deza (a.k.a. VaC|0): usuario de sistemas GNU/Linux desde finales del año 2000, actualmente pertenece a las comunidades de Software libre de Arequipa (AqpGlug y Debian Arequipa).

Si quieres hacerle algún comentario, escribe a: cvargas@aqpglug.org.pe

Microsoft CRÓNICAS DEL COMIENZO DEL FIN

Escrito por: *Leonel Iván Saafigueroa*

Todo el mundo conoce a Microsoft, una empresa que desde sus inicios tuvo un objetivo: "Tener algo que la gente necesite y que nadie mas que ellos puedan dárselo"; quien quiera disfrutar de una buena película escuchara estas palabras en boca del actor que personifico a Bill Gate en "Piratas de Silicon Valley". Esta película cuenta como Bill Gate (fundador de la empresa Microsoft) les hizo la vida imposible a Steve Job cofundador de Apple Computers.

Bill Gate trabajo para apple con el solo propósito de copiar el entorno gráfico de la MacInToush, pero vamos... la gente de Apple antes se habia tomado las mismas ideas de Xerox; es por eso que estamos ante una realidad: "cuando el futuro se acerca es imposible frenarlo"; pero Microsoft se las ingenio muy bien y patento muchísimas cosas como por ejemplo el doble click del mouse, esto es como que Ford diga que es el inventor del volante y pretenda que las otras empresas de automotores les paguen por haber copiado el volante de ellos.

¿Por que les cuento todo esto?, pues verán... somos los privilegiados de presenciar un show mediático en donde Microsoft demuestra tenerle mucho miedo a Gnu-Linux. El "jefe de licencias" de Microsoft en una entrevista a la revista Fortune dijo que el sistema operativo Linux así como muchas otras populares aplicaciones de software libre tienen una gran calidad, pero que se debe principalemente a que violan sus patentes; siendo mas especifico la interfaz gráfica violaría unas 65 patentes, OpenOffice 45 patentes, el kernel de linux 42, los programas de correo electrónico 15 patentes y el resto de las aplicaciones que forman parte de Linux, otras 68 licencias.

Estas declaraciones hicieron que el mismísimo Linus Torvalds saliera enardecido pero siempre con esa cara de papa bondadoso refutando a Microsoft:
"Es mucho más probable que Microsoft viole patentes que Linux. Si el código fuente de Windows fuera sometido a la misma revisión crítica que ha experimentado el código de Linux, Microsoft comprobaría que está violando las patentes de otras empresas."

Vaya vaya... esto quiere decir que ¿Estamos en guerra?!.

El vocero de OpenOffice también respondió tranquilo:
"No entiendo qué ha podido mover a Microsoft a arriesgarse tanto [...] Es un acto extraordinario y desesperado. "

No tardo en llegar las opiniones de Sun Microsystem:
"Deberían ser ustedes sabios para escuchar a los clientes que están amenazando con demandar. Pueden abandonarles, sobre todo si ustedes mismos les proporcionan motivación. [...] Innoven, no pleiteen."

Y si... el Windows Vista tiene un sistema de escritorio 3D que la verdad se parece mucho al inventado muchos meses antes para sistemas Unix y derivados... ¿eso también lo habrá patentado Microsoft? :P

Hace un tiempo nos enteramos que la gente de novell llego a un acuerdo económico con Microsoft para no ser demandado por usar Linux, pero ellos ahora dicen que es solo un acuerdo de colaboración lean; la cita de Novell:

"Discrepamos de las recientes afirmaciones realizadas por Microsoft sobre el tema de Linux y patentes. Nuestro acuerdo con Microsoft no es de modo alguno un reconocimiento de que Linux infrinja propiedad intelectual alguna de Microsoft.", aquí es cuando ellos se lavan las manos, y es que es así, microsoft quiere cobrarnos aun cuando no usamos su sistema operativo.

Para poner fin a todo este revuelo la gente de Microsoft salio a calmar las aguas diciendo:
"No estamos litigando. Si hubiéramos querido lo habríamos hecho hace años. Lo que queremos son más acuerdos con los distribuidores de Linux similares al que hicimos con Novell. Creamos un puente entre dos mundos que anteriormente estaban desconectados.", si... claro... un puente para sacarnos dinero, seamos sinceros no los necesitamos, y todo esto creo que viene por que muchas empresas fabricantes de notebooks están empezando a considerar vender sus equipos con Gnu-Linux preinstalado.

Microsoft primero nos ignoro, luego se burlo de nosotros y ahora nos atacan...
En una pagina de noticias un ingenioso lector dejo un comentario: "Pobres patentes de microsoft... han sido violadas por un tal linux....".

Sobre el Autor

Leonel Iván Saafigueroa: Es analista de Sistemas, docente, radioaficionado (LU5ENP), consultor en informática y conductor del programa de radio libre hispano - Red-Handed Radio (www.red-handed-radio.com.ar).

Si quieres hacerle algún comentario, escribe a: leonel@saafigueroa.com.ar

LA REVOLUCIÓN DEL DESARROLLO WEB

Escrito por: Ronny Yabar Aizcorbe

Desde la aparición de Internet, hemos sido testigos de diversos sitios web, variedad de aplicaciones y tecnologías que evolucionan constantemente.

Actualmente se está viviendo una revolución en el mundo Web denominada: Web 2.0. Podríamos referirnos a Web 2.0 como el desarrollo de aplicaciones modernas, super dinámicas, orientadas a la interacción y creación de redes sociales que fomenten la participación de los usuarios. La Web 2.0 hace uso de técnologías y técnicas como AJAX, XML, SEO, CSS avanzado, Sindicación de contenidos, patrones de diseño, estándares Web, folksonomía, usabilidad, enfoque en la simplicidad etc. Muchas de estas aplicaciones han sido y están siendo desarrolladas con la plataforma Web Ruby on Rails, y al parecer, elegir este entorno de desarrollo, es una gran decisión.

Si hay alguien que está haciendo ruido, captando la atención de varios programadores, usuarios, empresas de software y ganando mucho prestigio en la Web, es sin duda Ruby on Rails. Y seguramente nos preguntamos: ¿Qué es Ruby?, ¿Qué es Rails?, ¿Qué es Ruby on Rails?, ¿Porqué está haciendo tanta ruido y ganando mucha popularidad?, ¿Porqué sus aplicaciones son robustas, eficientes, y rápidas, creadas con un código sencillo? ¿Porqué algunos expertos como Martin Fowler, Bruce Perens lo están considerando como el lenguaje de programación Web del futuro? ¿Porqué dicen que incrementa la productividad, creatividad e innovación en los programadores?.

¿Qué es Ruby?

Es un lenguaje de programación dinámico totalmente orientado a objetos, multiplataforma y Software Libre, creado en 1995 por Yukihiro Matsumoto, en Japón. Dentro de sus fortalezas, el lenguaje Ruby permite una gran productividad del programador gracias a un enfoque hacia la simplicidad, menos código, menos errores, mayor facilidad de mantenimiento, sin necesidad de compilación. El resultado: el tiempo se dedica a construir aplicaciones potentes, de una manera eficiente, rápida y sencilla.

Ha combinado muchas de las características positivas de Perl, PHP, Java, C, Smalltak, Lisp. haciendo un lenguaje potentísimo para el desarrollo de aplicaciones.

Es considerado un lenguaje muy intuitivo casi a un nivel de lenguaje humano, que de una forma se asemeje al lenguaje natural, de esta forma hace que la experiencia de programación sea más divertida.

¿Qué es Rails?

Rails es un framework, es decir, un conjunto de programas, librerías que ayudan a desarrollar y unir los diferentes componentes de un proyecto de software. Rails es Software Libre, multiplataforma, distribuido bajo la licencia del MIT.

Fue desarrollado integramente con el lenguaje Ruby por el danés David Heinemeier Hansson y liberado al público por primera vez en Julio de 2004.

Soporte de Rails

Plataformas:

- GNU/Linux
- Unix
- FreeBSD
- Mac OS X
- Windows.

Bases de Datos:

- PostGreSQL
- MySQL
- Oracle
- SQL Server
- SQLite
- IBM DB2

Servidores Web:

- Webrick (Servidor integrado con Rails)
- Apache
- Lighttpd.

De la combinación del lenguaje Ruby más el framework Rails, surge la plataforma para desarrollo Web.

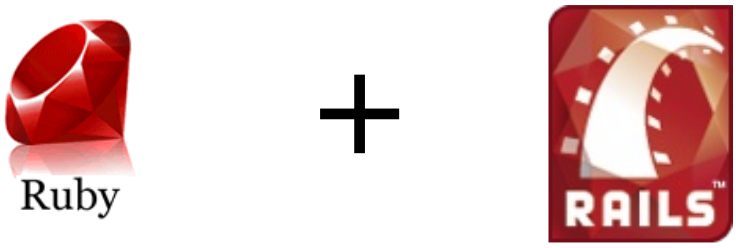


Figura 1 Ruby on rails, también conocida como RoR

La filosofía de Ruby on Rails, se basa en dos principios fundamentales: "No te repitas" y "Convención sobre configuración".

No te repitas se refiere a que las definiciones deberían hacerse una sola vez. pues los componentes están integrados de manera que no hace falta establecer puentes entre ellos, Cada pieza de conocimiento en un sisema, deberá ser expresada en un sólo lugar. Este principio se basa en escribir menos lineas de código para implementar la aplicación. Si el código es pequeño quiere decir que el desarrollo es más rápido y con menos errores, lo que hará que el código sea fácil de entender, mantener y mejorar.

Convención sobre configuración significa que el programador sólo necesita definir aquella configuración que no es convencional. En lugar de archivos de configuración, utilizamos una serie de convenciones simples que permiten averiguarlo todo.

Arquitectura: Patrón MVC

Ror facilita el diseño y desarrollo de aplicaciones web separando automáticamente en 3 capas todos los componentes de la aplicación (Modelo, Vista y Controlador).

- Modelo es todo acceso a base de datos, y las funciones que contienen la "lógica de negocio".

- La vista se encarga de mostrar la información al usuario final: HTML, XML.

- El controlador une la vista con el modelo, contiene toda la lógica de programación. Almacena las funciones que toman los valores de un formulario, delega consultas de base de datos al modelo y produce valores que invocarán a la vista adecuada.

En conclusión con Ror nos centramos en lo que verdaderamente importa: la funcionalidad de nuestra aplicación y podemos crear aplicaciones más complejas y de funcionamiento más "suave" con muchísimo menos esfuerzo. Algunas ejemplos de grandes aplicaciones hechas con RoR son: Basecamp, La Coctelera, Odeo, 43things, Shopify, Fluxiom, Typo, Bigcartel, WriteBoard, SoapBX, Campfire, BackPack, entre otras.

INSTALACIÓN DE RUBY ON RAILS EN GNU/Linux

Descargamos la última versión de Ruby desde:
http://www.ruby-lang.org/en/downloads/
(Actualmente la 1.8.6 es la versión estable).

Descromprimimos el archivo:
```
$ tar xvzf ruby-1.8.6.tar.gz
```

Ingresamos al directorio de ruby y compilamos:
```
$ cd ruby-1.8.6
$ ./ configure && make && make install
```

Para instalar en Debian y sus derivados.
Actualizamos nuestro listado de paquetes disponibles.
```
$ sudo apt-get update
```

Instalamos el lenguaje ruby, el archivo con la documentación de ruby y el modulo de ruby para apache:
```
$ sudo apt-get install ruby rdoc
libapache-mod-ruby
```

Luego descargamos el gestor de paquetes Rubygems desde
http://rubyforge.org/frs/?group_id=126
(Actualmente la última versión es la 0.9.3)

Lo descomprimimos con:
```
$ tar xvfz rubygems-0.9.3
```

Ingresamos al directorio de rubygems y lo instamos con:
```
$ cd rubygems-0.9.3
$ ruby setup.rb
```

Ahora pasamos a instalar Rails con el comando gem (nos instalará la última versión):
```
$ gem install rails --remote --include-
dependencies
```

Este parámetro --remote --include-dependencies significa que debemos de instalar algunas dependencias con las que debe contar rails. En total son 6 paquetes:

```
rake
activesupport
activerecord
actionpack
actionmailer
actionwebservice
```

Listo, ya tenemos instalado Ruby, Rails y podemos empezar a desarrollar.

Ahora crearemos la estructura de nuestra primera aplicación rails. Ejecutamos el comando rails con el nombre de nuestra aplicación.
```
$ rails nuestra_aplicacion
```

EL comando rails creara un directorio con el nombre de nuestro proyecto y dentro de él armará una estructura, una serie de subdirectorios con los cuales iniciaremos nuestro trabajo con ruby on rails.

```
create
create app/controllers
create app/helpers
create app/models
create app/views/layouts
create config/environments
create components
create db
create doc
create lib
create lib/tasks
create log
create public/images
create public/javascripts
create public/stylesheets
create script/performance
create script/process
create test/fixtures
create test/functional
```

Ahora para ver nuestro proyecto en marcha nos ubicamos dentro del directorio nuestra_aplicacion y ejecutamos:

```
$ cd nuestra_aplicacion
$ ruby script/server
```

En esa consola obtendremos una salida como esta:

```
=> Booting WEBrick...
=> Rails application started on http://0.0.0.0:3000
=> Ctrl-C to shutdown server; call with --help for options
[2006-01-23 16:45:54] INFO WEBrick 1.3.1
[2006-01-23 16:45:54] INFO ruby 1.8.4 (2005-12-24) [i386-linux]
[2006-01-23 16:45:54] INFO WEBrick::HTTPServer#start: pid=7627
port=3000
```

Nuestro servidor atenderá el puerto 3000. Si usamos nuestro navegador y apuntamos a la dirección http://localhost:3000 nos saldrá un mensaje de bienvenida (Ver figura 2)

NOTA:

- Si ejecutamos con ctrl + c, en el terminal o consola donde se está ejecutando nuestra aplicación detendremos el servidor Webrick y por ende la aplicación.

- Para facilitar el desarrollo podemos instalar plugins para distintos editores como Gedit, Jedit, Komodo Edit, la librería FastCGI para Ruby y un IDE como RadRails, aunque existen otros.

- Podemos probar ruby desde nuestra consola, instalando el intérprete irb.

ALGUNAS ESTADÍSTICAS DE ROR:

Estadísticas que publica en la edición de Febrero de 2006 la revista de IEEE Computer Society.

En enero de 2006 eran 230 000 el número de copias descargadas de Ruby on Rails.

En la actualidad el 5% de los programadores lo usan de forma regular. Su principal fortaleza es la rapidez para el desarrollo de aplicaciones web: "Un programador de Rails puede hacer el mismo trabajo que un equipo de Java porque las cosas son simples, rápidas, pero muy potentes" (Curt Hibbs).

"En un pequeño desarrollo para una startup el código fue la cuarta parte, y el tiempo de configuración el 10% del que hubiéramos empleado con Java" (Bruce Tate).

Segú Tiobe.com, Consultora dedicada al seguimiento de la calidad de software y la popularidad de los lenguajes de programación. De enero de 2006 a enero de 2007, Ruby escala 10 posiciones. Del puesto número 21, ahora se encuentra en el puesto 10.

El prestigioso sitio para de diseño y programación para la web SitePoint publicó los resultados de una reciente encuesta a 5000 desarrolladores logrando "la más completa encuesta dedicada a webdevelopers".

Además de los números que era previsible encontrar, como qué el 68% de ellos programan en PHP (con el 33% haciéndolo solamente en ese lenguaje), también hay otros datos resaltantes: 24% de los desarrolladores (o sea, aproximadamente 1200) migrarán a programar en Ruby on Rails en los próximos 12 meses.

ALGUNAS OPINIONES SOBRE ROR:

En todos lados. Desde la programación de computadoras Palm hasta la programación de supercomputadoras, desde programación embedida hasta bioinformáica, la programación deberá SIEMPRE de ser divertida. Ruby cumple con esta mención.(David Heinemeier Hansson).

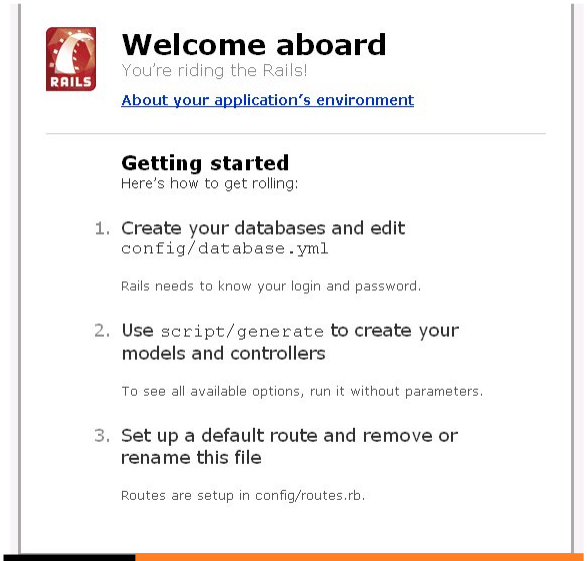


Figura 2 Mensaje de bienvenida de WEBrick

Ruby on Rails es increíble. Usarlo es como estar viendo una película de kung-fu, donde una docena de frameworks malvados se preparan para golpear al nuevo y pequeño recién llegado, pero al final terminan derrotados por una gran variedad de imaginativas formas". (Nathan Torkington de O'Reilly).

"Cada vez que hago la pregunta a un programador si es significativamente más productivo con Ruby que con Java/C#/PHP, recibo de respuesta un contundente sí. Esto es suficiente para mí, para empezar diciendo que tú deberías probar Ruby, lo cual porsupuesto, hará que te hagas la pregunta a ti mismo y respondas si lo crees conveniente". (Martin Fowler).

Que Ruby sea riesgoso, es una opinión común, y por buena razón. Las nuevos lenguajes son intrínsecamente riesgosos. Pero Ruby on Rails está más cercano a ser el principal, ese riesgo disminuirá, porque tendrás acceso a un sistema con un crecimiento cada vez mayor de programadores, de componentes (llamados gems o los plug-ins), de libros, y de nuevos modelos de negocio.(Bruce Tate).

ENLACES DE INTERÉS:

www.rubyonrails.org
www.rubyonrails.org.es
www.ruby-lang.org/es
www.ruby.org.es
www.rubylandia.com
www.sobrerailes.com
www.menearailes.com
www.planetarails.com
www.rubycorner.com/blogs/lang/es
es.wikipedia.org/wiki/ruby_on_rails

Sobre el Autor

Rony Yabar Aizcorbe: Egresado del Instituto del Sur de la especialidad de Informática. Activista de la comunidad de Software Libre, Co-fundador de la comunidad MenteLibre.

Si quieres hacerle algún comentario, escribe a: ronny@mentelibre.org

QT: C++ framework

Escrito por: *Julio Cesar Zevallos*

Desde el momento en que la interfaz gráfica que apareció en el mundo de la computación, se ha visto grandes avances en la forma en que se muestra información al usuario y de cómo este manipula el software, también han surgido una variedad de librerías y frameworks para la elaboración y diseño de interfaces gráficas, una de ellas QT que es la que veremos en este artículo.

¿Qué es QT?

QT es un entorno de trabajo (framework) para el desarrollo de aplicaciones multiplataforma de manera rápida y flexible. Introduce una nueva manera de trabajo a base de SIGNALS y SLOTS los cuales son fundamentales para cualquier aplicación QT.

QT nace del desarrollo de dos programadores que crearon una GUI toolkit para c++ por los años 90. Inicialmente crean una empresa llamada Quasar Technologies, después pasaría a llamarse Troll Tech y finamente Trolltech. La "Q" viene de la forma estilizada que tenía la Q en la aplicación emacs del ordenador de uno de los programadores y la "T" viene de la palabra Toolkit.

QT actualmente es utilizado en una amplia variedad de compañías y también software libre en el mundo, entre los más resaltantes: KDE (K Desktop Environment), Adobe Photoshop elements, Google Earth y un largo etcétera.

Existen actualmente 2 ramas de QT.
QT. Para el desarrollo de aplicaciones de escritorio de todo tipo.
QTopia. Para el desarrollo de aplicaciones para dispositivos móviles con Linux instalado así como greenphones que son teléfonos móviles creados por la empresa Trolltech.

Además de esto también está en desarrollo QT Jambi que en vez de usar C++ para programar, usa Java.

¿Por qué QT?

Digamos que eres un programador que crea una aplicación para Windows, en un momento dado quizás quieras explorar nuevos mercados (GNU/Linux por ejemplo), pues para hacerlo sólo tendrás que recompilar el código fuente y voila!, la aplicación está lista para correr, no tendrás que preocuparte de cómo se ve la aplicación en otra plataforma por que corre en el sistema gráfico del nuevo sistema operativo de forma nativa, no importa a qué plataforma migres, la aplicación lucirá y funcionará de la misma forma en todas.

Aparte de esto tienes la opción de crear software propietario o software libre si lo deseas. También cuenta con una excelente documentación en línea y soporte por parte de una gran compañía y de una amplia comunidad de usuarios alrededor del mundo.

Y... ¿Cómo lo hace?, ¿Cuál es el negocio?

Hemos dicho que QT es multiplataforma, esto QT lo logra implementando una API para cada plataforma, se hace uso de código nativo y es por eso que, dependiendo de la plataforma (OS) se verá diferente. Una manera de entender esto se ve en el siguiente gráfico.

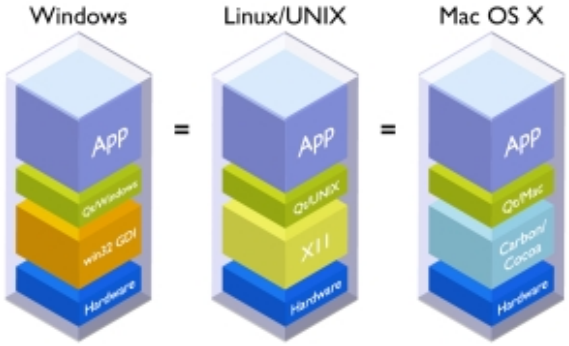


Figura 1 Modelos de las Api por cada plataforma

QT actúa directamente sobre la API gráfica, sin emulaciones ni máquinas virtuales obteniendo así una performance bastante respetable.

La empresa Trolltech se alimenta directamente de la comunidad Open Source para realizar nuevas versiones de la librería, es por eso que también pone bajo disposición GPL el software que ellos hacen, claro que con ciertas restricciones.

Características principales

Base de datos. Tiene soporte para las Bases de datos más conocidas: Oracle, MySQL, Postgresql, SQLite, IBM DB2, InterBase y cualquier DB ODBC (por ejemplo SQL Server).

Multiplataforma. Soporte para los sistemas operativos: Todas las ediciones de MS Windows desde 98 hasta Vista™ (incluyendo NT), MacOS X, GNU/Linux, Solaris, FreeBSD, HP-UX, IRIX, AIX y muchas otras variantes UNIX. También con soporte para 64 bits.

Programación.

Soporte para programación multihilo (multi-threading) para ambientes multiprocesador, Uso de modelo de programación Modelo-vista, programación de aplicaciones consola para aplicaciones de alta performance.

Usando QT

Bien, en mi caso Usaré QT 4.3 RC1 para Windows, no hay mucho misterio en la instalación debido a que viene empaquetado en un conveniente instalador. Para instalar en otras plataformas revise la documentación pertinente. En GNU/Linux si usted usa KDE tenga de por seguro que tiene una edición de QT instalada en su distribución, si es que no es así será necesario bajar e instalar desde su página web o repositorio correspondiente, si tiene instalado KDE y desea programar necesitará los paquetes de desarrollador de KDE, comúnmente llamados KDE SDK.

Es necesario recordar que necesitaremos tener instalado los archivos de ejecución de MinGW para que funcionen los programas.

Pues bien, al finalizar la instalación nos encontramos con tres aplicaciones fundamentales:

QT Designer.

Poderoso constructor de interfaces de usuario, nos permite crear nuestras ventanas con la ya conocida forma de "arrastrar y soltar".

QT Linguist.

Aplicación que nos permitirá hacer más fácil la implementación multi-idioma de nuestros programas.

QT Assistant.

Permite navegar a través de la documentación de aplicaciones creadas con QT, soporta RTF y HTML.

Aparte de esto también encontramos una amplia variedad de ejemplos y aplicaciones de demostración para que veamos el potencial de esta arquitectura, todas las aplicaciones con su código fuente y documentación, un buen punto en dónde empezar para conocer mejor esta arquitectura.

Sólo es necesario saber C++ orientado a objetos para empezar a trabajar con esta plataforma.

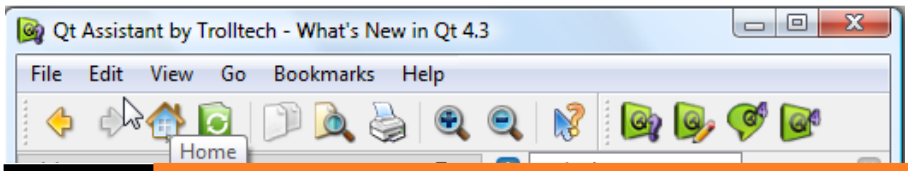


Figura 2 QT Assistant en Windows Vista

Figura 3 QT Assistant en Mac OS X

Integración con MS Visual Studio
(Edición comercial solamente)

Para aquellos programadores a quienes les guste esta IDE la transición les será mucho más fácil. Aparte de que QT assistant es oficialmente "certified for Windows Vista".

Gráficos.

Puede renderizar y generar gráficos SVG (ScalableVector Graphics), permite usar OpenGL en toda su amplitud para generar controles de usuario y gráficos dinámicos.

Redes.

Qt ofrece soporte para HTTP, HTTPS y FTP, soporte a bajo nivel de los protocolos TCP y UDP para crear aplicaciones cliente/servidor, también tiene soporte para SSL (Secure Socket Layer).

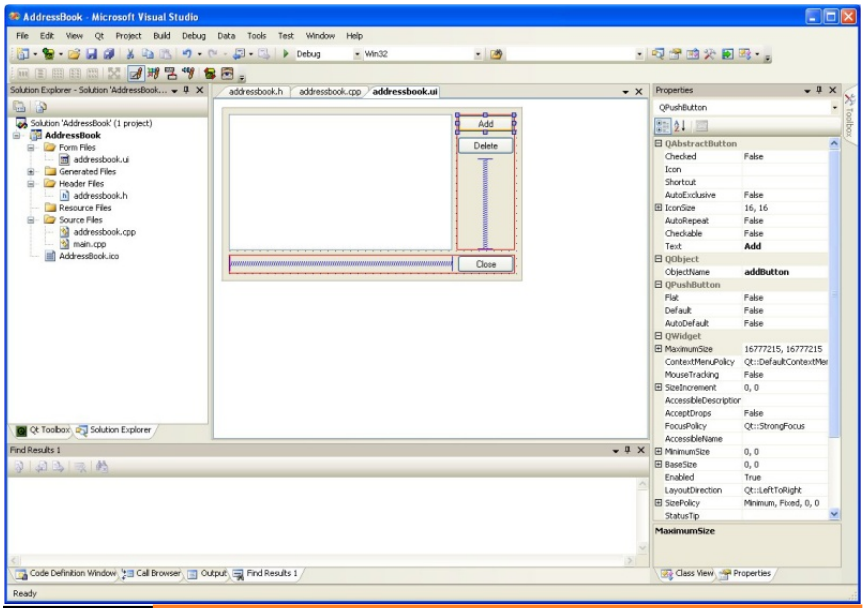


Figura 4 Integración con MS Visual Studio

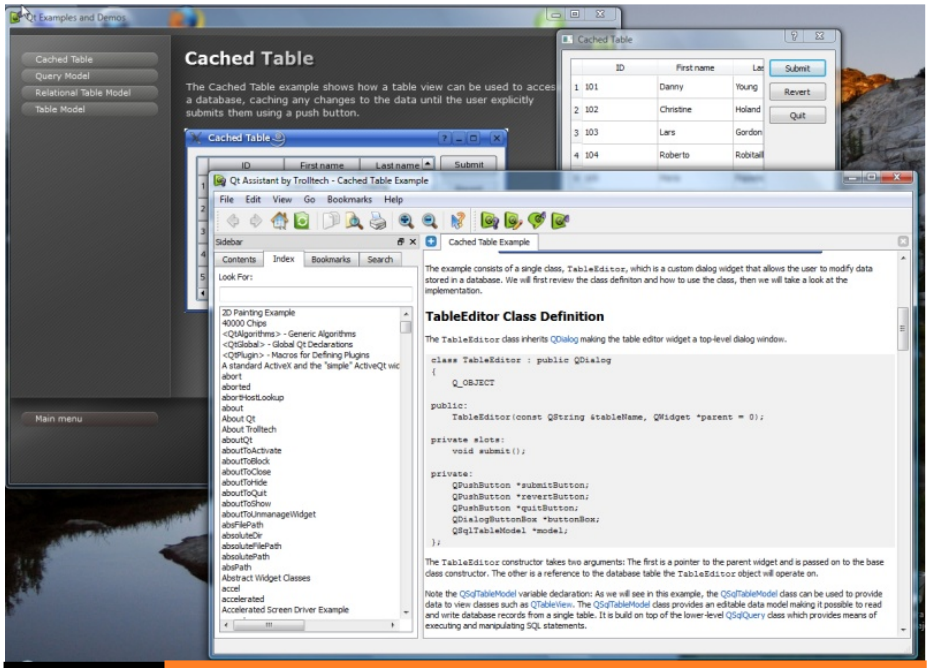


Figura 5 Documentación detallada

En el QT designer tenemos la opción de pre visualizar la forma en que cómo se verán los programas que desarrollemos con diferentes temas puestos por defecto. A parte de esto tiene una multitud de controles para insertar a los formularios.

Licencia dual

QT usa una licencia dual, es decir, tiene dos licencias para cada tipo de uso que se le dé. Hay dos posibilidades:

Una licencia comercial, usada cuando se quiere desarrollar software propietario, es decir sin liberar el código fuente, se tiene que comprar una licencia de desarrollador a Trolltech. Hay diferentes licencias comerciales, cada una con sus limitaciones particulares.

Una licencia Open Source (GPL), que se usa para crear software liberando el código fuente respetando los términos de la licencia GNU GPL.

Instalación y configuración de MinGW (MS Windows™ solamente)

Para que las aplicaciones QT (en la edición Open Source) corran perfectamente necesitaremos el archivo MinGW runtime que se descarga de la página http://www.mingw.org, no obstante, para que podamos compilar nuestro código fuente necesitaremos descargar los siguientes archivos de la página antes mencionada, trabajaremos con los archivos *.tar.gz que contienen archivos binarios:

Gcc-core: Archivos principales de GNU C Compiler.
Gcc-g++: Soporte para el lenguaje C++ en GCC.
Binutils: Herramientas para la construcción de archivos binarios.
Mingw32-make: Implementación del programa "make" para MS Windows™.
W32api: Archivo cabecera de implementación de la API de MS Windows™.

Lo único que hay que hacer extraer todos los archivos a una carpeta común, supongamos C:\mingw, luego hay que agregar a la variable de entorno PATH los valores: "C:\mingw\bin;C:\mingw\libexec\gcc\mingw32\3.4.2" tomando en cuenta que en mi caso tengo instalada la versión 3.4.2 de gcc.

Listo, ya tenemos instalado y configurado nuestro compilador C++ libre, la cual, usaremos para trabajar con QT o, de otra manera, con cualquier código C++ estándar.

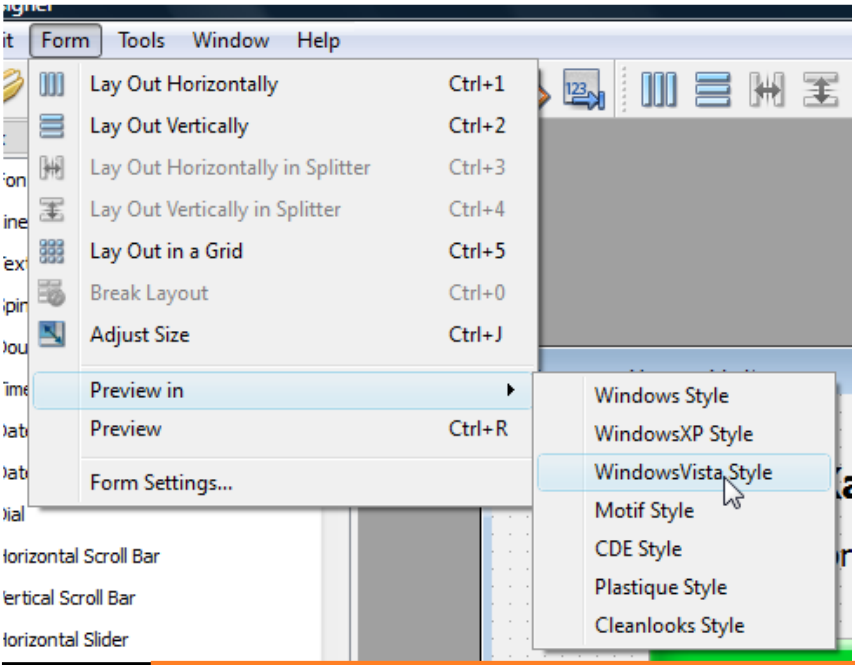


Figura 6 Previsualizacion en los diferentes escritorios

Sobre el Autor

Julio César Zevallos: (JulCes) es Técnico en computación e informática, usuario GNU/linux desde 2002, miembro de varias comunidades Open Source, actualmente estudiante Ingeniería de Sistemas, desarrollador y miembro activo del AQPGLUG.

Si quieres hacerle algún comentario, escribe a: jzevallos@aqpglug.org.pe

MLDonkey
TODOS LOS P2P EN UNO SOLO

Escrito por: *Leonel Iván Saafigueroa*

Desarrollado inicialmente por Fabrice Le Fessant del INRIA, MLDonkey es el único programa P2P multired capaz de conectarse a varias redes simultáneamente, escrito en el lenguaje de programación Ocaml, se distribuye públicamente bajo licencia GPL.

Breve explicación

MLDonkey en sus inicios fue el único cliente no oficial para acceder a la red eDonkey, pero luego fue incorporando módulos para acceder a otras redes como: BitTorrent, FastTrack (KaZaA), Gnutella, Kademlia, Overnet, Gnutella2 y FileTP.

Lo interesante de MLDonkey es que funciona de manera "invisible" sin presentar una interfaz gráfica al ejecutarlo; es lo que ellos denominan core o núcleo, siendo necesario acceder por telnet, una interfaz gráfica externa (GUI), o a través de http con nuestro navegador favorito de internet. Esta modalidad nos permite controlarlo a distancia desde otra computadora, el core nos genera además toda una estadística de las descargas.

Como obtenerlo

MLDonkey es multiplataforma pudiéndose ejecutar en la mayoría de las distribuciones Gnu/Linux, Unix, Free/OpenBSD y sí... también en los productos de la empresa de Redmond que no me gusta mencionar.

El desarrollo de MLDonkey es bastante activo y aunque se pueden descargar de internet binarios listos para usar, yo recomiendo compilar siempre la última versión CVS que trae siempre muchas mejoras.

Necesitamos instalar antes en el sistema en donde se irá a compilar los paquetes autoconf, ocaml y zlib1g-dev, luego:

1. Descargamos el código fuente desde Savannah:

```
cvs -z3 -d:pserver:anonymous@cvs.savannah.nongnu.org:/sources/mldonkey co mldonkey
```

2. Ejecutamos la configuración siempre dentro del directorio mldonkey.

```
cd mldonkey
./configure
```

3. Comenzamos a compilar y luego vamos a buscar algún refrigerio a la nevera que seguro tardará un poco.

```
make
```

4. Finalmente lo instalamos:

```
make install
```

Funcionamiento

Bien como explicamos antes MLDonkey funciona en dos partes, tenemos por un lado el core o núcleo que lo ejecutamos con mldonkey:

```
debian:/mldonkey# mldonkey
2007/01/11 14:36:32 [cO] Starting MLDonkey 2.8.2.CVS ...
2007/01/11 14:36:32 [cO] Language ES, locale ISO-8859-15, ulimit for open files 1024
2007/01/11 14:36:32 [cO] MLDonkey is working in /root/.mldonkey

2007/01/11 14:36:32 [cO] creating new MLDonkey base directory in /root/.mldonkey

2007/01/11 14:36:32 [cO] loaded language resource file
2007/01/11 14:36:32 [DNS] Resolving [debian] ...
2007/01/11 14:36:32 [DNS] Resolving [www.mldonkey.org] ...
2007/01/11 14:36:48 [cO] Logging in /root/.mldonkey/mlnet.log
2007/01/11 14:36:48 [dMain] Core started
```

Si todo salio bien tendremos que ver el mensaje "Core started" como última linea, esto nos indica que el núcleo ya está trabajando.

El programa nos crea un directorio llamado ".mldonkey" dentro del usuario que estemos usando (en este caso es root), todos los ficheros que se descarguen aparecerán allí dentro del directorio "incoming".

Para acceder al núcleo podemos a través de telnet introduciendo localhost o la dirección IP de nuestra máquina junto al puerto 4000:

```
debian:~# telnet localhost 4000
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
Welcome to MLDonkey 2.8.2.CVS
Welcome on mldonkey command-line

Use ? for help

MLdonkey command-line:
>
```

También podemos acceder a través de nuestro navegador favorito de internet introduciendo localhost o la dirección IP de nuestra máquina junto al puerto 4080:
http://192.168.1.1:4080/
Veremos la interfaz predeterminada que es de color verde, con élla podremos manejar todas las redes, hacer búsquedas y gestionar nuestras descargas.

GUIs Independientes

Aunque MLDonkey se puede usar con telnet, http, también es posible usarlo por medio de otros GUIs independientes, algunos de éllos son:

Sancho

Sancho es un gui multiplataforma programado en java, su apariencia es muy similar al programa edonkey pero es algo lento y a veces puede presentar algunos cuelgues, esto no influye en las descargas, pues de ello se encarga el núcleo del MLDonkey.

Platero

Platero es una interfaz gráfica programada para KDE, incluye muchas novedades como una perfecta integración al usa la misma configuración que el resto de los componentes KDE, además se integra perfectamente con el navegador Konqueror. Es rápido y eficiente, no consume mucha memoria, está programado in C++ usando las librerias Qt y KDE.

KMLdonkey

También es bastante flexible, mantiene el estilo de KDE y quizás sea el más adecuado para usuarios de OSX.

Conclusiones

El MLDonkey es cierto que soporta muchas redes, es cómodo y flexible, trabaja a la perfección al mismo tiempo en las redes donkey y BitTorrent, pero aquellos que quieran descargar sólo de una red le recomiendo usar MLDonkey para Elinks y cualquier otro cliente independiente para BitTorrent.
Tengan paciencia, MLDonkey es lento pero muy eficiente y seguro, y por sobre todas las cosas les aconsejo usar la interfaz web que es las más estable.

En Internet:

El sitio del proyecto MLDonkey:
http://mldonkey.sourceforge.net/
El sitio del Gui Sancho:
http://sancho-gui.sourceforge.net/
El sitio del Gui Platero:
http://www.nongnu.org/platero/
El sitio del Gui KMLdonkey:
http://kmldonkey.org/

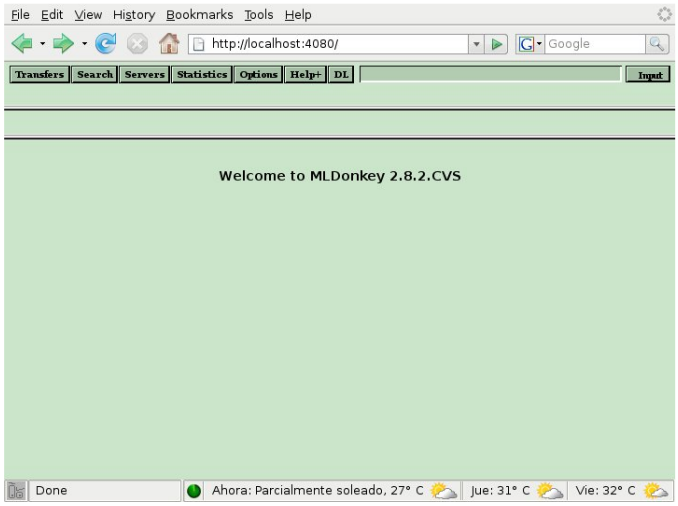


Figura 1 Interfaz predeterminada del MLDonkey en web

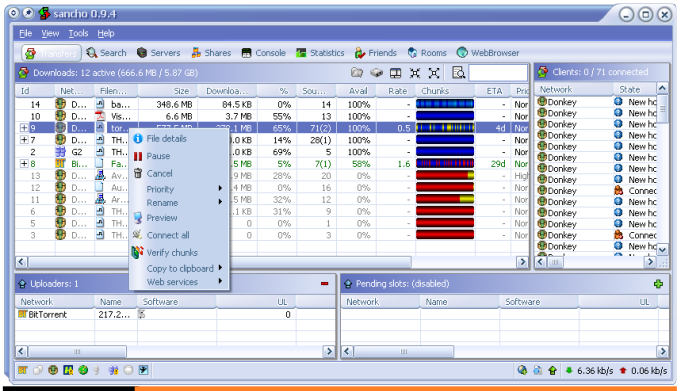


Figura 2 Interfaz gráfica Sancho

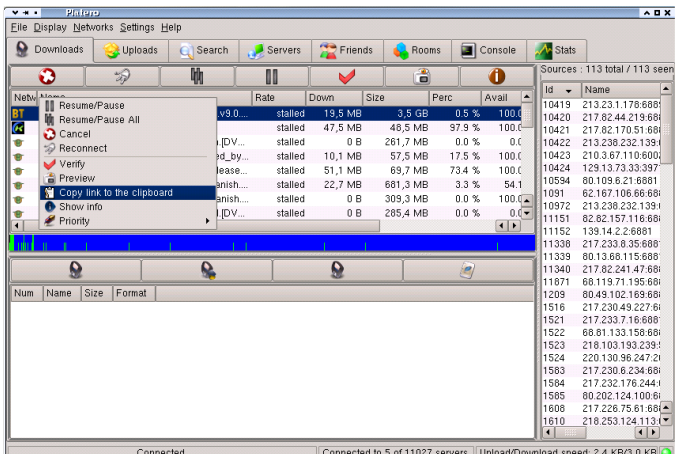


Figura 3 Interfaz gráfica Platero

Sobre el Autor

Leonel Iván Saafigueroa: Es analista de Sistemas, docente, radioaficionado (LU5ENP), consultor en informática y conductor del programa de radio libre hispano - Red-Handed Radio (www.red-handed-radio.com.ar).

Si quieres hacerle algún comentario, escribe a: leonel@saafigueroa.com.ar

ENSEÑANZA DE GNU/LINUX EN CUBA

Escrito por: *José Candelario Balmaseda Novoa*

Propuesta de programas para la organización curricular de la enseñanza-aprendizaje del sistema operativo gnu/linux en los jóvenes club de computación y electrónica.

Con el desarrollo y el uso masivo de las TICs en Cuba ha comenzado una nueva etapa en la educación; en ella la computación se constituye en uno de sus recursos tecnológicos de mayor trascendencia. En este sentido los Jóvenes Club de Computación y Electrónica desempeñan un papel esencial, ya que por sus instalaciones pueden y deben pasar todos los estratos de la población.

Este movimiento cuenta con una basta red de instituciones a todo lo largo y ancho de nuestro país, que tiene como misión "Proporcionar una cultura informática a la comunidad con prioridad hacia niños y jóvenes, jugando un papel activo, creativo y de formación de valores en el proceso de informatización de la sociedad cubana".

El curso de mayor demanda, entre los que se ofertan en los Jóvenes Club, es el de Operador de Microcomputadora, que inicia a los estudiantes en el Sistema Operativo Microsoft Windows. Sin embargo, en estos momentos es una necesidad y política de nuestro estado "… la migración progresiva de las computadoras instaladas en los organismos de la Administración Central del Estado hacia el software libre, sobre la base del sistema operativo Linux, eliminando así la presencia casi exclusiva del Windows en las máquinas". Se requiere aclarar, en relación con esta cita, que el nombre correcto es GNU/Linux, ya que Linux se refiere, en lo fundamental, al Kernel o núcleo del sistema.

¿Por qué migrar hacia el software libre?

Entre los Software, los más importantes son los relacionados con los sistemas operativos. De ellos, durante años se ha venido enseñando sólo uno: el Microsoft Windows, que es el que hoy día, en nuestro país y otras partes del mundo, está instalado en la inmensa mayoría de las computadoras personales. Sin embargo, es un Sistema Operativo que es del tipo Software propietario, es decir, que para usarlo debemos comprarlo y regirnos por la licencia copyright que ampara este tipo de programa.

Según las consultas bibliográficas realizadas por el autor, el movimiento de software libre presenta aspectos positivos tales como, el no ser monopólico; el oponerse a las patentes; el cuestionar la propiedad privada del conocimiento; y lo más importante, trascender el modo de producción en el que fue engendrado, cuya contradicción principal se daba entre la producción social y la apropiación privada de sus resultados. En él, tanto la producción como la apropiación del software es social.

Las bondades antes descritas, para nuestro bloqueado país, son esenciales. Continuamente se nos niegan patentes y tecnologías, e incluso publicaciones, participación en congresos y ventas de productos soportados en tecnología Microsoft Windows.

A diferencia de esta férrea posición, Cuba es paladín de la educación y transmisión gratuita de conocimientos. Afortunadamente existe una opción: usar un software del tipo libre, que podamos adaptarlo a cualquier necesidad y distribuirlo por todo el país sin ningún problema legal.

Estudiantes de informática del Joven Club

Según Roger Peña, en una entrevista concedida a La ventana, Portal Informativo de La Casa de las Américas en el artículo Entrevistas: Enviado el viernes, 7 de Mayo del 2004 "… La mayoría de la gente cree que Windows es más fácil de usar que Linux, que este es sólo para entendidos, pero no es exactamente así. Lo que sucede es que existe muy poco conocimiento de Linux y por tanto falta personal humano que entrene a quienes quieren adentrarse en este mundo. Esto se aprecia en los diferentes niveles de enseñanza de nuestro país, por cuanto, como barreras subjetivas se subutiliza en los pocos centros que lo tienen instalado actualmente y se subestima por la mayoría de profesionales que se inician en su conocimiento.

En GNU/Linux hay una potencialidad que debe ser explorada y explotada. Como sistema operativo es totalmente funcional y la distribución del mismo, que se usa en tres de los cinco Jóvenes Club de Computación y Electrónica del municipio Puerto Padre, la Tinux, basaba en una resmasterización de OpenSuSE, es compatible con las aplicaciones del Microsoft Office.

La tesis en opción al Título Académico de Master en Nuevas Tecnologías para la Educación "PROPUESTA DE PROGRAMAS PARA LA ORGANIZACIÓN CURRICULAR DE LA ENSEÑANZA-APRENDIZAJE DEL SISTEMA OPERATIVO GNU/LINUX EN LOS JÓVENES CLUB DE COMPUTACIÓN Y ELECTRÓNICA" sugiere una vía para la generalización de la enseñanza de este sistema operativo.

Primeramente se partió de hacer una nueva distro (remasterización) que fuera amigable para el usuario, el Licenciado Alberto Méndez Pérez, quien es el padre, nos aclara que partió de la necesidad de una distribución fácil para su uso. Adaptable para la migración de Windows a Linux. Amigable al usuario. Que incluya los paquetes básicos de Oficina. Pocos CD de instalación para su generalización.

¿Por qué OpenSuSE?

Reúne todas las condiciones anteriormente mencionadas. Se basa en Software libre, no comercial. Su centro de administración, Yast, hace más fácil su instalación y administración del sistema. Característica que lo diferencia de las demás distribuciones y lo hace menos hostil al administrador del sistema y al usuario en sí.

Esa propuesta consta de tres programas, uno de operador de microcomputadora sobre GNU/Linux, otro de usuario avanzado de Cálculo, el tabulador electrónico u hoja de cálculo, similar y que no le cede en nada, al Microsoft Excel y un programa para iniciar al estudiante en la programación en C++.

¿Por qué estos tres programas?

Pensemos que mañana despertáramos con la noticia de que no podemos usar Microsoft Windows, ¿Cómo preparar en breve tiempo a los usuarios que se inicien en GNU/Linux? Un curso de operador de microcomputadora desde cero, para personas que nunca han oído hablar de informática es necesario, en el se incluyen elementos de sistema operativo, de ofimática, (procesador de texto, hojas de cálculo, presentaciones y base de datos) correo electrónico y redes.

El curso para usuarios avanzados de hoja de cálculo es fundamental, hasta que se construyan sobre software libre los principales ERP (Enterprise Resource Planning o Planificación de Recursos Empresariales) las empresas deben resolver inmediatamente numerosos problemas que pueden solucionarse de esta forma.

La inmensa mayoría del código fuente que acompaña al software libre es hecha en C++, de ahí que un curso que inicie a los estudiantes en este lenguaje de programación sería vital para su comprensión y desarrollo.

Estos cursos están confeccionados basándose en 64 horas clases, con 32 encuentros (dos semanales) de dos horas de duración. Se comenzaron a aplicar, el de operador de microcomputadoras, en octubre del 2005 y hasta la fecha (contando el que se está desarrollando actualmente) se han hecho cuatro, el de cálculo se aplicó de octubre del 2006 a enero del 2007 y el de programación den C++ se aplicado dos veces, de octubre del 2006 a enero del 2007 y de marzo a Junio del presente año.

La aceptación por parte de los estudiantes ha sido muy buena y ha existido un buen aprovechamiento de las clases impartidas.

Creemos firmemente que debemos estudiar cada día más para demostrarles a los incrédulos que existe otra filosofía que debemos aplicar, debemos buscar cada día nuevos caminos para demostrarle a los fanáticos del Microsoft Windows, esos que no piensan, esos que no tienen paz, esos que repiten frases de otros que Un mundo mejor es posible.

Como bien dijera el venezolano José Luis Regalado.
"Donde hay ignorancia hay fanatismo, donde hay fanatismo no hay tolerancia, donde no hay tolerancia no hay paz"

Conozcamos a GNU/Linux en particular y al Software Libre en general para eliminar la ignorancia, para dejar se ser fanático y alcanzar la paz.

Estudiantes de informática del Joven Club

Sobre el Autor

Lic. José Candelario Balmaseda Novoa
Joven Club de Computación y Electrónica Puerto Padre III

Si quieres hacerle algún comentario, escribele a:
cande02034@ltu.jovenclub.cu

Un Vistazo a FEDORA 7

Escrito por: *Felix Manuel Arismendi Quispichuco*

El 31 de mayo fue lanzada la uersión 7 de Fedora, ciertamente muy esperada debido a las prorrogas previas a su lanzamiento.

Introducción

Esta versión, a diferencia de las previas, como principal novedad viene en el conocido dvd, pero además está disponible en livecd, ya sea con gnome o kde, y al estilo de ubuntu, sólo ocupa un cd.

No está demás señalar que como principal diferencia al instalar de dvd se tiene la opción de hacer upgrade sobre la versión inmediata anterior, en tanto que con los livecds se puede optar por instalar luego de cargar el sistema.

Descargué el livecd de gnome vía torrent, a no negar la descarga fue rápida, luego de quemar el cd y de una probada procedí a la instalación, el proceso muy similar al de la versión anterior, aunque claro desde el livecd sin muchas opciones para elegir que software instalar, mal podría esperarse contar con openoffice en el cd, este contiene gnumeric y abiword como herramientas de oficina.

Instalación

El proceso de instalación en si en una pentium IV con 512 de ram y controladora de video intel i810, resultó libre de problemas, todo el equipo fue reconocido. Eso sí, durante la prueba del livecd, compiz funcionó a la primera, sin embargo luego de instalar y reiniciar la pc, al habilitar los efectos gráficos sencillamente resultó en un mensaje de error y no habilitó los mencionados efecto.

Eso se corrigió luego de instalar la red y descargar el driver experimental de intel, luego de lo cual habilitados los efectos gráficos, compiz funcionó, entonces procedí a instalar yumex y con el instalé una significativa cantidad de software, entre ellos el ya afamado beryl, que como se observa, realmente muy bien, anduvo a la primera sin tropiezos, nada más que algunos toques a la configuración, echarlo a andar y ya.

No deja de ser extraño para los acostumbrados a Fedora y en general a las distro con varios cds, el asunto de pasar buen rato instalando vía internet.

Por cierto muchos también hemos experimentado con ubuntu, particularmente con Feisty, pero para los habituados a Fedora esto resulta totalmente raro.

Si observan bien yumex ha sufrido algunas modificaciones, sin embargo se trata de cambios menores, Fedora 7 como se denomina ahora, pues se ha eliminado el termino core, en correspondencia a la unificación de los repositorios a cargo o en cierta forma en dependencia de Redhat, con los extras a que son las contribuciones comunitarias.

Figura 2 Screenshot de Fedora + Beryl

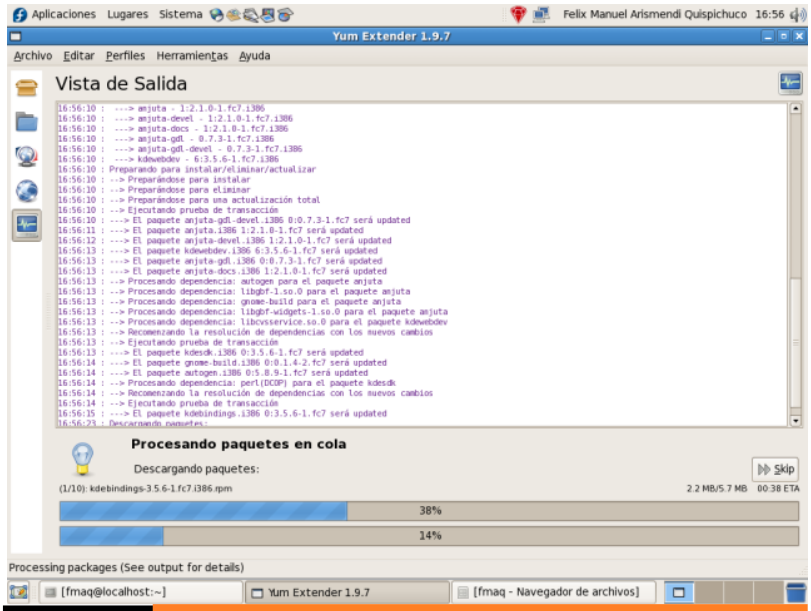


Figura 2 Herramienta para la instalacion de paquetes - YUM

Fedora 7 incluye la versión 2.18 de gnome o la 3.5.6 de KDE, Firefox 2, liberation fonts, una diferencia importante es el hecho que ahora las unidades de disco son denominadas /dev/sda1 o según corresponda, esto en razón del uso de la interfaz libata que comparte codigo del driver sata. Si bien se ha reescrito el soporte para firewire, circulan algunas observaciones con referencia a problemas con periféricos que se conectan por esa vía.

En lo referente a virtualización, se incluye KVM y qemu ademas de Zen.

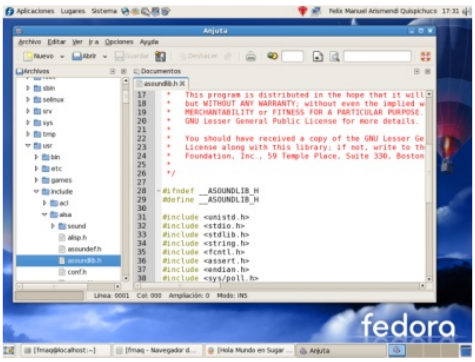
El rendimiento de pirut, pup y yum, conocidas herramientas de actualización de Fedora han sido mejoradas, brindando un mejor tiempo de respuesta. Un detalle realmente interesante resultan ser las herramientas koji, revisor y pungi. Koji es una herramienta de entorno texto que a partir de datos ingresados por el usuario, crea una imagen iso de cd/dvd o livedc, mientras que revisor es un entorno gráfico que trabaja con koji, pungi por su parte, provee librerías para encontrar resolver dependencias y agrupar paquetes así como correr rutinas de anaconda sobre los paquetes previamente agrupados y crear isos en base a ellos, en menos palabras herramientas que permiten personalizar cd's, dvd's o livecd's.

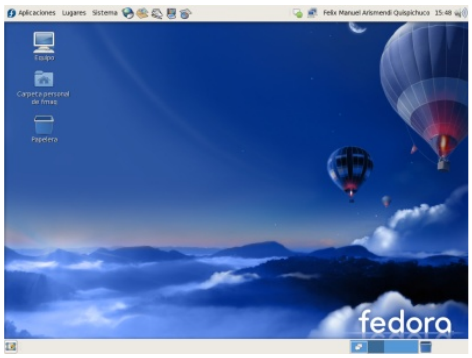
Incluyo este comentario para hacer más evidente lo relativo a reconocimiento de hardware por parte de Fedora 7. "Debo confesar que me encontré con la grata sorpresa de que en esta ocasión Fedora 7 me reconoció absolutamente todo mi hardware sin necesidad de descargar e instalar algún controlador externo: tanto mis tarjetas de red (la alámbrica y la inalámbrica que es una Intel® PRO/Wireless 3945 802.11 a/b/g y que generalmente es un dolor de cabeza pero en esta ocasión ya tiene los controladores incorporados, así como algunos otros más) como la tarjeta de vídeo (aunque no tengo aún aceleración por hardware gracias a que ATI no ha liberado las especificaciones puedo instalar su propio controlador para tenerla, al menos ése es un consuelo aunque no lo he hecho aún) y en general el resto de los dispositivos. No tuve que hacer nada más que clic, clic, clic y tuve todo funcionando a la perfección."[1]

En conclusión

Como observación personal, que estando activados los efectos gráficos (compiz o beryl), xine abortaba antes de iniciar en tanto que gxine si funcionaba, lo mismo gmplayer tenía problemas, los cuales desaparecieron al desactivar los efectos gráficos, xine antes de abortar (iniciando desde consola), mencionaba algo referente a xvideo aun habilitando xvideo en xorg.conf no solucionó el problema.

Otro detalle a señalar, es que las imágenes de los livecd's (Gnome o KDE), sólo son para la arquitectura i686, por lo que para las i586 se requiere el dvd. Otros detalles, entre ellos, sshd viene desactivado por defecto, network manager habilitado por defecto, esto en referencia a la instalación desde livecd.





[1] www.imoqland.com/Article778.phtml

Otros Links Interesantes

http://www.howtoforge.com/the_perfect_desktop_fedora7
http://fedoraproject.org/wiki/Docs/Fedora7VirtQuickStart

Sobre el Autor

Felix Manuel Arismendi Quispichuco
Comunidades en las que participo: Fedora-es/Ubuntu-es/Ubuntu-lat

Si quieres hacerle algún comentario, escribe a: felix-a@speedy.com.pe

LINUX & Y OPEN SOURCE MONO

Escrito por: *Steve Ataucuri Cruz*

Mono es una implementación Open Source del framework .NET, del ECMA Common Language Infraestructure (CLI) que ofrece caracteristicas escabilidad, portabilidad y sobre todo interoperatibilidad entre múltiples lenguajes.Creando una poderoso framework de desarrollo robusto para correr aplicaciones en Linux,Solaris, Mac OS X, Windows y la familia de BSDs. Liderada por Miguel de Icaza y patrocinada por Novell con una extensa comunidad de desarrolladores.

Una pequeña introducción

Mono nace en el ano 2001 como una solución frente a los problemas que se presentaban a la hora construir aplicaciones para Linux, en ese entonces desarrollar aplicaciones era muy tedioso y un desarrollador debería dominar muchos lenguajes de programación para resolver distintos problemas, pero la odisea no terminaba allí aun se presentaban mas problemas. Ningún lenguaje de programación hablaba con ninguno de una forma transparente, se llego a construir pocos esfuerzos pero que a la larga era como patchs que no soportarían una arquitectura escalable.

Distintos grupos de Api's (perl, python, tcl/tk etc) ofrecian soluciones pero no había un plataforma común a todas para que puedan comunicarse entre si.

Cuando Microsoft Corporation envió dos especificaciones formales a la ECMA(European Computer Manufactures Association)de C# y la plataforma .NET fueron aprobadas y estas mismas especificaciones fueron enviadas a la ISO(International Standards Organization(ISO).

Miguel Icaza toma las especificaciones para implementar varias tecnologías en una sola llamada Mono y ofrecer una migración transparente de aplicaciones en .Net a Linux y construir nuevas aplicaciones en esta.

Características

Multiplataforma, capaz de abstraer la capa de software en un paso intermedio llamado IL (Intermediate Language) y poder ejecutarse en distintas plataformas.
Gestión automática de memoria todo esta automatizado
Soporte de servicios web RPC (Remote Procedure Call), soporte SOAP
Compatibilidad con XML
Aplicaciones GUI multiplaforma GTK# y Windows Forms
Una extensa librería compatible con .NET (System, Http, Criptografía, I/O etc
Especificaciones Basada en el estándar ECMA/ISO, aquí se divide en dos partes :

ECMA-334: el cual define la sintaxis y semántica del lenguaje de programación C#
ECMA-335: una serie de detalles de la plataforma .NET llamada CLR (Common Language Runtime)o comúnmente conocido como CLI(Common Language Infraestucture)

El ECMA-334 sirvió para construir las características potentes del compilador de Mono. El ECMA-335 se divide en varios tópicos para su implementación :

Parte I Arquitectura Describe varios caracteristicas del CLI como Common Type System, Common Language Specification.
Parte II Metadata Características de los formatos en .Net
Parte III CLI Describe la sintaxis y semántica de la programación del CLI
Parte IV Librerías Describe la compatibilidad con librerías nativas de .NET

Arquitecturas Soportadas

La tecnología Mono tiene optimizaciones del just-in-time (JIT) runtime, a las siguientes arquitecturas:

CPUs	Sistema Operativo
x86	Linux, BSD's, Solaris, OS X, Windows
x86-64(AMD64,EM64T)	Linux
IA64(itanium2 64bit)	Linux
ARM familia	Linux
Alpha	Linux
SPARC, SPARCv9	Solaris, Linux
s390, s390x(31, 64bit)	Linux
PawerPC	OS X

Herramientas SDK – Herramientas de desarrollo

Para el desarrollo de mono existe una variedad de herramientas de desarrollo las mas usadas dentro del proyecto tenemos : monodevelop (IDE que soporta las característica de Visual Studio),emacs, Eclipse en modo C#.

Monodevelop: Gnome IDE diseñado primordialemen en C#, que da soporte a lenguajes .NET:

- Basado en SharpDevelop.
- Integrado con Stetic(sucesor de glade)
- Integrado con Monodoc.
- Completacion de codigo.
- Soporte add-in (NUnit, MonoQuery, Acceso a Base de datos).

Aparte del soporte de Gtk# (binding para toda la APIs GNOME), tambien soporta Windows.Forms(GUI microsoft) y Cocoa#(construir aplicaciones nativas para OS/X)

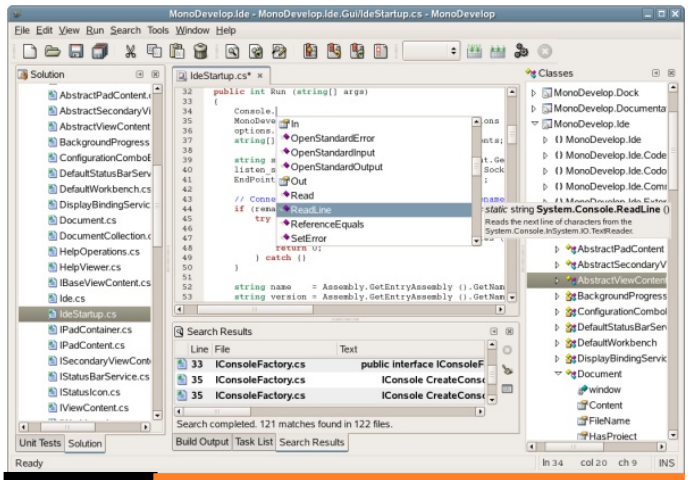


Figura 1 Screenshot de Monodevelop

Como funciona Internamente

El runtime ofrece una serie de características como la compilación just-in-time (JIT)y ahead-of-time(AOT), la idea una vez creado un programa se traduce a IL(Intermediate Language)en donde la ingeniería del runtime genera código para un CPU en particular de dos maneras:

- Compilación Ahead-of-time: Basado en ELF, a idea es permitir generar ficheros precompilados este es código nativo de la arquitectura que reduce el tiempo de inicio de las aplicaciones.
- Compilación Just-in-Time: Compila el código IL a código nativo y lo ejecuta

Además ofrece una serie de servicios como : garbage collection, thread management, I/O , construcción de sistemas remotos con RPC y una amplia integracion con librerías del sistema.

Lenguages soportados

Mono soporta una variedad de lenguajes internamente se tiene un compilador para el lenguaje C#, Basic etc. Externamente se utiliza compiladores de terceros como el de python(IronPython), java, php,Boo, Nemerle, Ruby, Object pascal (RemObjects). Una característica del runtime es soportar a cualquier lenguaje que genere IL; entonces esto se hace portable entre plataformas.

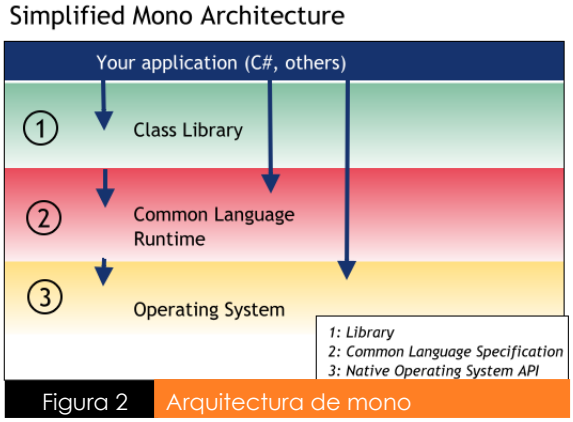


Figura 2 Arquitectura de mono

Compiladores que trabajan con mono:

Compilador C# 1.0, C# 2.0 y trabajando para el compilador C# 3.0
Compilador Visual Basic.NET, (escrito en VB.NET)
Compilador Java IKVM (librerias para compatibilidad de Java, y el uso de GNU Classpath.
Boo inspirado en Python.
IronPython open source de microsoft.
Phalanger, compilador de php recientemente liberado como open source

Compiladores bajo desarrollo:
PHP.NET (Google Summer code 2005)
Ruby.NET

También ofrece una extensa compatibilidad con Apis como ADO.NET, ASP.NET, Windows Forms.

Licencias

Distintas partes de Mono están protegías con tres licencias :
El compilador, esta liberado bajo los términos de la GNU General Public License (GPL)
El runtime, bajo GNU Library GPL 2.0(LGPL 2.0)
Librerías de clases, liberada bajo la licencia MIT X11

Lo que permite a terceras partes crear software propietario sin ningún problema de patentes.

La comunidad

La existencia de mono se debe a la comunidad que aportan desde distintas partes:contribuciones individuales, compañias y organizaciones que usan mono, mediante google summer code, y el grupo de desarrollo mono, que hacen posible su existencia. Llevada a cabo con 18 desarrolladores(Novell), 8 desarrolladores(MainSoft), +400 colaboradores externos.

Enlaces de Interés

http://www.mono-project.com
http://www.ecma-international.org
http://www.monodevelop.com
http://www.php-compiler.net

Sobre el Autor

Steve Ataucuri Cruz: Es técnico en informática, desarrollador en mono desde el 2004, participa en varias comunidades incluidas el AQPGLUG.

Si quieres hacerle algún comentario, escribe a: sataucuri@aqpglug.org.pe

USANDO DIALOG Y SHELL SCRIPT EN FreeBSD

Escrito por: Alonso Cárdenas Marquez

Este tutorial pretende ser una guía para iniciarnos en el uso de dialog, un programa que permite mostrar cajas de dialogo utilizando shell scripts, como son yes/no, message, input, menu, text, info, checklist, program, ftree, tree boxes.

Introducción

Que mejor manera de aprender el funcionamiento de dialog en combinación con shell scripts, que utilizando un ejemplo, el ejemplo que desarrollaremos será un pequeño frontend para el uso de burncd, un utilitario para la grabación de CD/DVD en FreeBSD.

Usando DIALOG

En esta sección explicaré como gnerar las diferentes cajas de dialogo usando dialog que serán especificados en el ejemplo práctico, como son la caja de dialogo yesno, menu y msgbox.

Sin duda una información más detallada sobre el uso pueden ser encontrados en el man de dialog o ejecutando el comando dialog en línea de comando.

Para generar una caja yesno utilizaremos la siguiente sentencia:

```
$dialog --title "Título de ventana" --yesno "Mensaje que contendra la caja" altura ancho
```

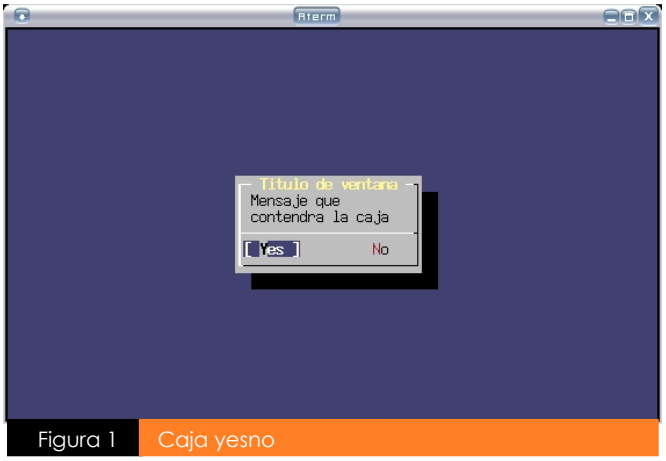


Figura 1 Caja yesno

Para generar una caja de msgbox utilizaremos la siguiente:

```
$dialog --title "Título de ventana" --msgbox "Mensaje de la caja" altura ancho
```

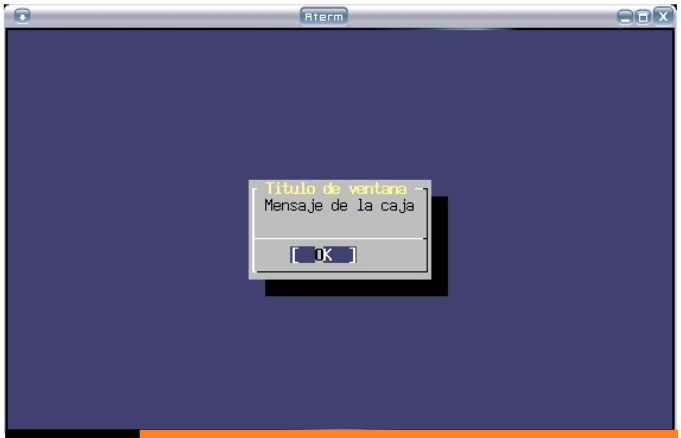


Figura 2 Caja de msgbox

Para generar una caja de menu utilizaremos la siguiente (ver figura 3):

```
dialog --title "Título de ventana" --menu "Mensaje de la caja"
altura ancho altura_menu \
        opcion1 "Descripción 1" \
        opcion2 "Descripción 2" \
        opcion3 "Descripción 3"
```

Algunos parámetros adicionales cómo --title que podemos utilzar son --hline que mostrará una línea en la parte inferior de la caja de dialogo, --hfile abrirá un archivo al presionar la tecla de F1 generalmente usado como ayuda en línea, --clear limpiará la ventana al salir de la ejecución de la caja de dialogo, entre otras.

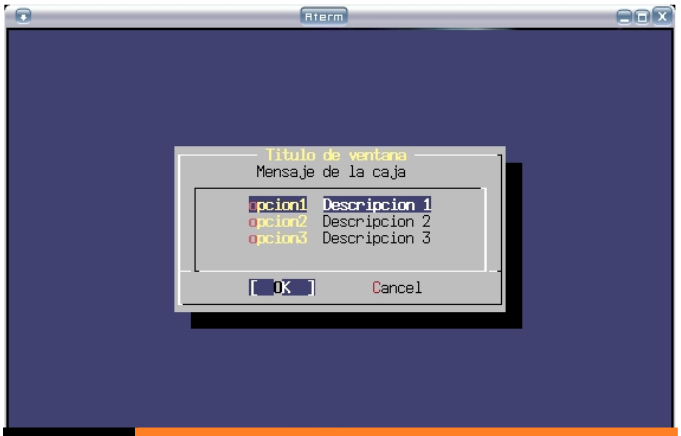


Figura 3 Caja de menus

Por ejemplo generamos una caja menu con estos parámetros

```
dialog --title "Título de ventana" --hline "Ventana 1" --hfile "/var/log/auth.log" --clear --menu "Mensaje de la caja" altura ancho altura_menu \
      opcion1 "Descripción 1" \
      opcion2 "Descripción 2" \
      opcion3 "Descripción 3"
```

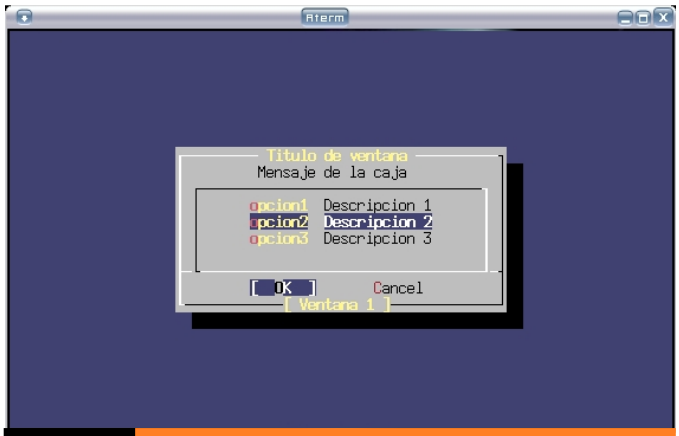


Figura 4 Caja de menus avanzada

La implementación del programa dialogo está un poco imcompleto en FreeBSD comparado con el de GNU/Linux que contiene más parámetros para generar más cajas de diálogo, pero son fáciles de implementar teniendo en cuenta la presencia de la librería libdialog.

¿QUÉ MÁS NECESITAMOS?

Para lograr el correcto funcionamiento de nuestro programa ejemplo que mostraremos más adelante necesitaremos tener acceso algùn dispositivo de CD/DVD que nos permita grabar, para esto necesitamos un usuario que pertenesca al grupo operator y tenga permisos de escritura en estos dispositivos.

Para referencia podemos darle un vistazo al archivo /etc/group como también al archivo /etc/devfs.conf, los permisos en /etc/devfs.conf para el acceso a nuestro dispositivo deberían ser algo así:

```
own   acd0   root:operator
perm  acd0   0660
```

Y reiniciamos el servicio devfs

```
# sh /etc/rc.d/devfs restart
```

Esto siempre y cuando tengamos sólo un dispositivo de grabación si tuvieramos varios sólo agregamos más lineas como las anteriores cambiando el parámetro acd0 por acd1, acd2, acd3 etc segùn el nùmero de dispositivos de CD/DVD que tengamos.

NUESTRO PROGRAMA DE GRABACIÓN

Una vez llegado a este punto crearemos un archivo llamado dburncd o lo ubicaremos si queremos en el directorio /usr/local/bin para poder ser ejecutado por cualquier usuario o si queremos ejecutarlo sólo por nuestro usuario, creamos en nuestro home el directorio ~/bin y copiamos ahí nuestro script con el siguiente contenido:

```
#!/bin/sh
# Definimos algunas variables

BURNCD=/usr/sbin/burncd
TMPDIR=/tmp
ISOFILE=$@
NROPARAM=$#


# Función de inicio

do_init()
{
dialog --title 'Usando dialog y shell script en FreeBSD' --yesno 'Este programa facilitará la grabación de un cd/dvd, ayudandonos mediante un entorno gráfico a que el proceso de grabación de cd/dvd utilizando burncd sea más fácil. ¿Desea continuar?' 10 100
   if [ $? != 0 ]; then
            exit
   else
            devices_detection
   fi
}


# Función para detectar y generar el menù de dispositivos de CD/DVD

devices_detection()
{

# Generamos el menú de acuerdo a los dispositivos encontrados

unset tmp
numDev=`(dmesg | grep acd | cut -d ":" -f 1 | sort -u | wc -l)`

for i in `(dmesg | grep -m $numDev acd | cut -d ":" -f 1)`
   do
            infDev=`(dmesg | grep -m 1 $i | cut -d ":" -f 2 |
            cut -d ">" -f 1 | sed -e 's,^ ,,g' -e 's, ,-,g')`
            tmp=$tmp"$i $infDev "
done

dialog --title "Grabación de CD/DVD" --menu "Dispositivos disponibles" 10 70 3 $tmp 2>$TMPDIR/dburncd-burnopt

   line_jump
   devices_option
}
```

```
# Función para detectar y generar el menù de dispositivos de
CD/DVD

devices_detection()
{
# Generamos el menù de acuerdo a los dispositivos
encontrados
   unset tmp
   numDev=`(dmesg | grep acd | cut -d ":" -f 1 |
   sort -u | wc -l)`

   for i in `(dmesg | grep -m $numDev acd | cut -d ":" -f 1)`
   do
            infDev=`(dmesg | grep -m 1 $i | cut -d ":" -f 2 |
            cut -d ">" -f 1 | sed -e 's,^ ,,g' -e 's, ,-,g')`
            tmp=$tmp"$i $infDev "
   done

   dialog --title "Grabación de CD/DVD" --menu "Dispositivos
   disponibles" 10 70 3 $tmp 2>$TMPDIR/dburncd-burnopt

   line_jump
   devices_option
}

# Generamos el menú para especificar que deseamos hacer
#con el dispositivo de grabación
devices_option()
{
   dialog --title "Grabación de CD/DVD" --menu "¿Qué desea
hacer?" 12 50 5 \
            grabar "Grabar un CD/DVD" \
            borrar "Borrar un CD-RW/DVD-RW" \
            blanquear "Borrado rápido de un CD-RW/DVD-RW" \
            salir "Salir del programa" 2>$TMPDIR/dburncd-opt

   optSel=`(cat $TMPDIR/dburncd-opt)`

   case $optSel in
   grabar)
            file_checking;;
   borrar)
            erasing_opt;;
   blanquear)
            blanking_opt;;
   salir)
            exit;;
   esac
}

# Verifica si se ha especificado algùn archivo para grabar y la
# existencia del mismo
file_checking()
{
   if [ $NROPARAM != 0 ]; then
            if [ -f $ISOFILE ]; then
                     burning_speed
            else
                     dialog --title "Error en la grabación de
CD/DVD" --msgbox " Debes especificar un archivo ISO valido
para ser grabado utilizando la línea de comando " 8 42
                     exit
            fi
   else
            dialog --title "Error en la grabación de CD/DVD" --
msgbox " Debes especificar un archivo ISO valido para ser
grabado utilizando la línea de comando " 8 42
            exit
   fi
}

# Función que genera el menù para seleccionar la velocidad
# de grabación

burning_speed()
{
   dialog --title "Grabación de CD/DVD" --menu "Velocidad de
grabación" 12 50 5 \
            4x "Grabar a 4x" \
            8x "Grabar a 8x" \
            16x "Grabar a 16x" \
            32x "Grabar a 32x" \
            Max "Grabar max. velocidad" 2>>$TMPDIR/dburncd-
            burnopt
   line_jump
   burning_mode
}

# Función que genera el menù de modo de grabación

burning_mode()
{
   dialog --title "Grabación de CD/DVD" --menu "Modo de
   grabación" 10 50 4 \
      data "Grabar un cd de datos" \
      audio "Grabar un cd de audio" \
      multi "Grabar un cd multisesión" 2>>$TMPDIR/dburncd-
      burnopt
   line_jump

   c=0
   for i in `(cat $TMPDIR/dburncd-burnopt)`
   do
            c=`(echo $c+1 | bc -l)`
            case $c in
            1)
                     devSel=$i;;
            2)
                     if [ $i != "Max" ]; then
                              i=`echo $i | sed -e 's,x,,g'`
                     else
                              i=max
                     fi
                     devVel=$i;;
            3)
                     if [ $i = "multi" ]; then
                              i="-m data"
                     fi
                     burTip=$i;;
            esac
   done

   tmp="-f /dev/$devSel -s $devVel -e $burTip $ISOFILE fixate"
   $BURNCD $tmp

   if [ $? -eq 0 ]; then
   dialog --title "Grabado de un CD-RW/DVD-RW" --msgbox
   "Grabado del CD-RW/DVD-RW finalizado con éxito" 5 50

            else

   dialog --title "Error en el grabado de un CD-RW/DVD-RW"
   --msgbox "Grabado del CD-RW/DVD-RW no ha finalizado
   con éxito, verifique que la unidad contenga un CD/DVD
   valido" 8 50

            fi

            devices_detection
}
```

```
# Funcion para borrar totalmente un CD-RW/DVD-RW
erasing_opt()
{
   devSel=`(cat $TMPDIR/dburncd-burnopt)`
   $BURNCD -f /dev/$devSel -e erase

   if [ $? -eq 0 ]; then
            dialog --title "Borrado de un CD-RW/DVD-RW"
            --msgbox "Borrado del CD-RW/DVD-RW finalizado
            con éxito" 5 50
   else
            dialog --title "Error en el borrado de un CD-RW/
            DVD-RW" --msgbox "Borrado del CD-RW/DVD-RW
            no ha finalizado con éxito, verifique que la unidad
             contenga un CD/DVD valido" 8 50
            fi

            devices_detection
}

# Función para blanquear un CD-RW/DVD-RW
blanking_opt()
{
   devSel=`(cat $TMPDIR/dburncd-burnopt)`
   $BURNCD -f /dev/$devSel -e blank

   if [ $? -eq 0 ]; then
            dialog --title "Borrado de un CD-RW/DVD-RW"
            --msgbox "Borrado del CD-RW/DVD-RW finalizado
            con éxito" 5 50
   else
            dialog --title "Error en el borrado de un CD-RW/
            DVD-RW" --msgbox "Borrado del CD-RW/DVD-RW
            no ha finalizado  con éxito, verifique que la unidad
            contenga un CD/DVD valido" 8 50
    fi

    devices_detection
}

# Función para agregar un salto de linea
line_jump()
{
   echo "" >> $TMPDIR/dburncd-burnopt
}

# Iniciamos la ejecucion del script
do_init
```

Grabamos el archivo, le aplicamos permisos de ejecución.

```
# chmod 754 dburncd
```

Y ejecutamos nuestro programa, si todo fue realizado correctamente hasta aquí, observaremos la siguiente pantalla de bienvenida. (ver figura 5)

```
# dburncd [ARCHIVO]
```

Si no especificamos un nombre de un archivo iso valido, sólo se podrá realizar el borrado de discos regrabables.

NOTAS FINALES

Sin duda el programa podría ser mejorado, complementado etc, pero esto pretende ser una guía para aquellos que deseen iniciarse en el uso de dialog y shell script, hacia donde quieren llegar o que aplicaciones tan complejas se desarrollar dependerá mucho de la investigación y práctica de cada uno.

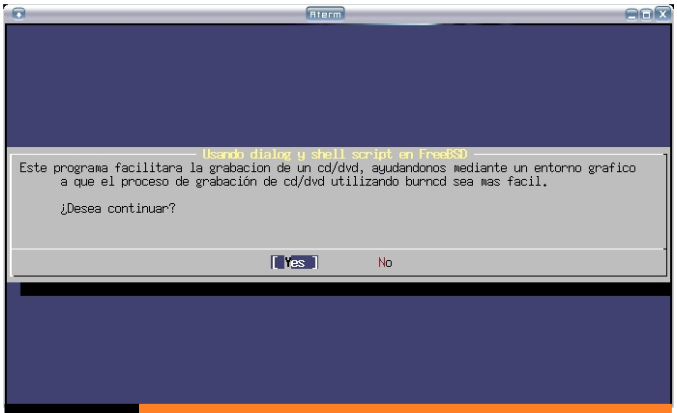


Figura 5 Ejecución del script dburncd

Sobre el Autor

José Alonso Cádenas: Se inicio en el mundo del software libre desde el año 2000, actualmente es desarrollador oficial (committer) del proyecto FreeBSD.

Si quieres hacerle algún comentario, escribe a: acm@freebsd.org

CONSOL

Trujillo – Perú

Próximamente...

Octubre – 2007