Número 4
Septiembre / 2009

Revista de Software Libre en Viñales

LINVIX

Opera 10.0

Konsolas Virtuales
Tilda y Yakuake
Programas que siempre nos deben acompañar

SVN Control de versiones
Parte II

Django
Framework de desarrollo web

linvix.wordpress.com

Editorial

Ya está aquí la cuarta edición de Linvix, aunque un poco retrasada por problemas técnicos que nos han llevado a realizar el número dos veces, queremos agradecer a Javier, -diseñador y maquetador- por el doble esfuerzo que ha realizado para que esta revista llegue a ustedes lo antes posible.

Este mes Linvix está bien marcada con el color rojo, y es que tenemos muchos motivos para estar alegres, estamos estrenando un nuevo logotipo, hemos incluido nuevas secciones, -Internet y Programación-, lanzamos un concurso y se nos han unido colegas del programa de los Joven Club y otras partes del mundo colaborando con artículos para que Linvix continúe con calidad que ha mantenido hasta hoy.

Ahora es el momento de disfrutar este número 4 de Linvix, que viene cargada de nuevos artículos, como el de Opera, donde se da una revisión a las novedades de este nueva versión del Navegador. También iniciamos un Curso de Django, fantástico framework de desarrollo web que cada día llama más la atención de los desarrolladores, además de esto lanzamos un concurso de escritorios, invitando a todos los lectores que nos envién su escritorio, todo esto y mucho más lo verán en está nueva publicación.

Quiero una ves más agradecer a todas las personas que han colaborando en la realización de la revista, sin lo cual no sería posible llegar a ustedes.

El equipo de Linvix.

Coordinador
Roylan Suárez Reyes
roylan04012@pri.jovenclub.cu

Redacción
Fernando Arencibia Pita
fernando04014@pri.jovenclub.cu

Diseñador
Javier Suárez Rodriguez
javier04017@pri.jovenclub.cu

Corección
Milaidys Rodríguez Martín
milaidys04025@pri.jovenclub.cu

Corección
Danelia Gonzáles Martínez
danelia04016@pri.jovenclub.cu

Contenido

Una vez concluida la etapa de vacaciones, comienza el nuevo curso escolar 2009/2010 y con ello la satisfacción de haberle prestado a nuestra comunidad y fundamentalmente a los niños gratas y diversas actividades durante este período, dentro las que se destacan: competencia de juegos, navegación nacional, actualización de antivirus, etc.

Ahora todos los Joven Club del país se preparan para el proceso de matrícula que se efectuará el próximo día 15, donde se ofertarán los cursos a la población sin importar grado escolar, edad, sexo o color de la piel. El objetivo fundamental de estos cursos es informatizar a la sociedad y preparar a todas aquellas personas que lo necesiten en dependencia de sus necesidades laborales o científicas.

Para este período instructivo tenemos previsto ofrecer curso de Programación de Páginas Web en PHP y Django, también daremos cursos de Fotoshop y Gimp, Arquitectura de Máquina, Operador de Micro sobre Linux (OpenOffice.org Write, OpenOffice.org Calc, OpenOffice.org Impress) entre otros, siempre haciendo énfasis en los cursos de Software Libre.

Estamos en estos momentos inmersos en un proceso de cambios y de transformaciones, donde los Joven Club juegan un papel protagónico en la informatización de la sociedad, donde la calidad de los servicios que se brindan debe ser cada vez mejor y más diversa.

Fernando Arencibia Pita
fernando04014@pri.jovenclub.cu

CONCURSO

Mi Escritorio

Bases del Concurso

- Nombre, Apellidos, y Correo Electrónico del Concursante.
- Resolución mínima de la imagen 1024 x 768
- Tamaño máximo de la imagen 700 KB
- Enviar a linvix@gmail.com
- El plazo de envió vence el día 18 de Noviembre de 2009.

PARTICIPA ...

Konsolas Virtuales

Programas que siempre nos deben acompañar

Un sistema GNU/Linux siempre ha sido sinónimo de consola o terminal, como prefieras llamarlo. Aunque la cosa ha cambiado mucho y los escritorios son más fáciles de usar, aún se hace necesaria en ocasiones el uso de la consola. En otras muchas ocasiones simplemente es más rápido usar una consola que navegar por un menú con todas sus opciones hasta que lleguemos a ella, cargue la ventana y hagamos lo que nos sea necesario. Con esta entrada les quiero hacer llegar un par de consolas muy especiales que probablemente ya la tengas instalada o la tendrás.

Muchas veces tenemos varias consolas abiertas y dispersas por todos los escritorios, en ocasiones es un auténtico caos. Si somos un poco ordenados quizá con suerte sólo tendremos una consola abierta, pero con varias pestañas. Dentro de lo que cabe no está mal, pero es probable que las opciones que les voy a presentar les gusten más.

Pues esta es básicamente la idea que les quiero presentar. Si no has probado nunca ese tipo de consola, les aseguro que se están perdiendo mucho. Tenemos principalmente dos, Tilda y YaKuake.

Tilda es la opción para GNOME. Es realmente muy, muy configurable, desde dimensiones, posición, falsa transparencia, fondo personalizado, cambio de fuente, permite pestañas, temas de color, animación al abrirse y cerrarse, configurar scrollba, en fin, varias opciones que la hacen muy dinámica y fácil de usar.

Instalando Tilda:

```
$ apt-get install tilda
```

Asegúrate de habilitar la casilla “Habilitar doble memoria temporal” en la pestaña “General” de las preferencias pues es probable que sin ella no se vea bien el texto o lo que estés escribiendo.

Para acceder a sus Preferencias con un simple click derecho/Preferencias, ya tenemos una ventana para cambiar todas las opciones de la aplicación.

Lo que debes saber:

· Mostrar/ocultar: F1 por defecto, se puede cambiar en las preferencias
· Copiar/Pegar: Ctrl+Shift+C / Ctrl+Shift+V
· Nueva pestaña: Ctrl+Shift+T
· Cerrar pestaña: Escribir dentro “exit”
· Es recomendable que desactives la animación si usas Beryl pues la mezcla no queda muy bien

Lo malo:

· No se pueden cambiar los atajos de teclado, sólo el de mostrar/ocultar tilda, aunque los que viene por defecto personalmente los veo bastante bien.

Problemas solventados:
Si tienes Beryl, muchas veces no recibe bien el foco cuando pulsas la tecla de tilda.
Lo ideal es que al pulsarla la consola tome el foco automáticamente y así es en Metacity, pero no con Beryl.
Tal y como nos comenta Juani (gracias) en un comentario de esta entrada, el problema del foco de tilda es fácilmente solventable cambiando una opción en Beryl. Abrimos el gestor de ajustes de Beryl y en las opciones generales buscamos la opción "Nivel de prevención del robo de foco". Le ponemos como valor "Ninguno" y adiós problema.
Ejemplo de como se ve Tilda en mi escritorio.

Nota: Pueden ver el marco blanco que delimita la consola de Tilda.

Yakuake

El nombre viene de su predecesor, kuake (y este del famoso juego de id, Quake), que si no me equivoco fue el primer terminal de este tipo para GNU/Linux. Su nombre viene de Yet Another Kuake (otro kuake más). Esta es la mejor alternativa para KDE. Las prestaciones son prácticamente las mismas que las citadas en tilda.

Instalando Yakuake:

```
$ apt-get install yakuake
```

Lo que debes saber:

· Mostrar/ocultar: F12 por defecto
· Pegar: Shift+Insert
· Nueva pestaña: Ctrl+Shift+N
· Cerrar pestaña: No definido por defecto (en su defecto teclear dentro 'exit')
· Renombrar pestaña/sesión: Alt+Control+S

· Pestaña anterior/siguiente: Shift+Izquierda / Shift+Derecha
· Todas las teclas anteriores son configurables
· Es recomendable que desactives la animación si usas Beryl pues la mezcla no queda muy bien
· Se puede usar la configuración de konsole
· El interfaz de configuración es más intuitivo para algunas cosas que el de tilda
Podemos acceder a las preferencias dando clic en

Lo malo:

· No puedes definir la consola en un lugar exacto de tu escritorio, siempre partirá de la parte superior. Sólo puedes definir sus dimensiones, pero no el punto de origen. Hay mucha gente que prefiere tenerla en un lado, en la parte inferior o simplemente pegada al

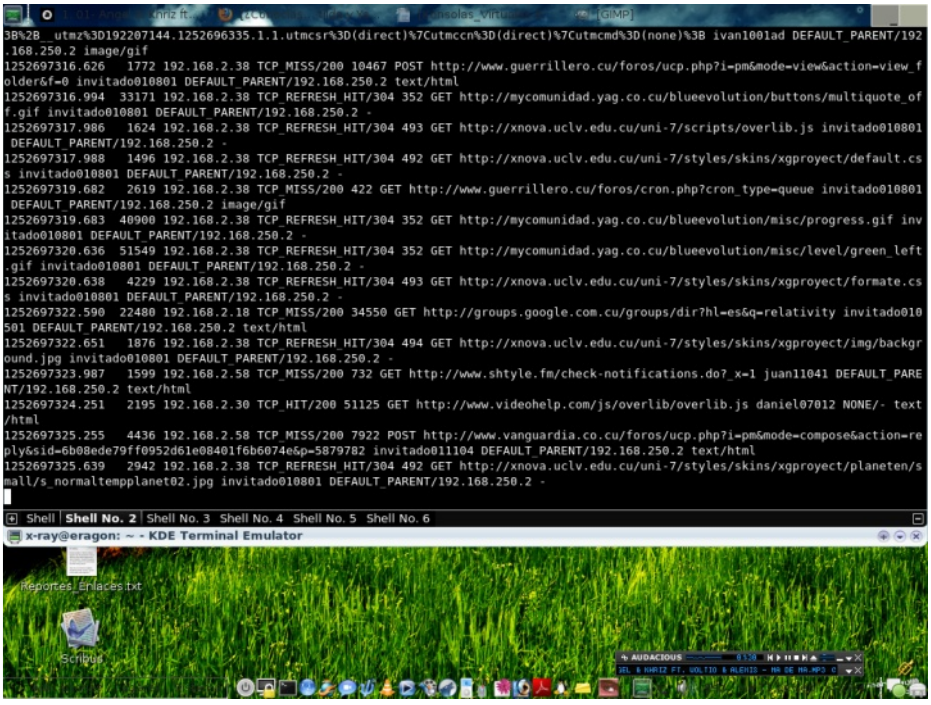
Boton para
Acceder a las
Preferencias

escritorio en un lugar determinado. También que no nos permite la transparencia de la consola, ya que no estamos usando KDE, es una de las pocas cosas que yakuake no nos permite.

¿Cómo hacer que se inicien al principio?

Sistemas > Preferencias > Sesiones > Programas de inicio > Nuevo
· Nombre: tilda / Comando: tilda
· Nombre: YaKuake / Comando: yakuake

Ejemplo de como se ve Yakuake en mi escritorio.



Existen muchas otras aplicaciones emuladoras de consolas, por el momento estas 2.

Salu2.

Artículo de referencia:
http://tuxpepino.wordpress.com/2007/04/30/¿conocias-tilda-y-yakuake/
http://granjeromoelinuxero.wordpress.com/2009/06/19/yakuake/
http://es.wikipedia.org/wiki/Yakuake

Raydel Hernández Martínez
raydel11082@pri.jovenclub.cu

WEBS DE REFERENCIA

www.linuxparatodos.net

www.espaciolinux.com

www.linuxhispano.net

Kompozer

Editor de páginas webs

En la evolución de las nuevas tecnologías internet ha jugado un papel trascendental. La autopista de la información como también se le suele llamar a este cúmulo de páginas web con múltiples contenidos informativos. En el mundo moderno muchas personas tienen acceso a este amplio mundo. La internet se ha convertido rápidamente y en muy corto tiempo en un espacio para el negocio y el conocimiento, no olvidemos que este último está generando gran cantidad de capital. Por lo que poder publicar una página o un sitio web completo, nos dará la posibilidad de interactuar con otras personas del mundo y poder divulgar quienes somos y lo que hacemos. La creación de paginas web ya ha dejado ser un trabajo solo para entendidos de la materia, aunque para llevar a cabo un trabajo de este tipo en el ámbito profesional se requieren conocimientos de lenguajes de programación para la web como por ejemplo HTML(Lenguaje de marca de hypertexto), u otros de scripts como PHP(Hypertext Preprocessor) o JavaScript por solo citar estos dos, usted mismo puede comenzar desde ya ha desarrollar sus propios contenidos web y más tarde convertirlo en un sitio que un tiempo después podrá publicar si este cuanta con los requerimientos necesarios y el contenido resulta interesante para los internautas de la red de redes.

Para la creación de páginas web existen ciertos programas que facilitan la tarea a usuarios con pocos conocimientos o que se enfrentan a esta labor por primera ves. En esta ocasión hablaremos de Kompozer un potente editor de páginas web para GNU/Linux.

Kompozer es una versión no oficial del conocido NVU proyecto desarrollado por Linspire, ahora rescrito por Fabien Cazenave y bajo Mozilla Composer Kompozer implementa muchas mejoras y corrige los errores de programación(bugs) que presentaba su predecesor. Su diseño facilita el trabajo incluso a los usuarios más inexpertos. Entre sus novedades más notables podemos encontrar:

- Administrador de sitios FTP: Con la ayuda de una barra lateral usted podrá navegar cualquier sitio previamente definido en las opciones de navegación.
- Nueva forma del selector de colores: Está más ligado al usuario final, pues es muy similar al que están acostumbrados
- Trabajo con pestañas: Los usuarios de Mozilla-Firefox ya saben de esta utilidad. Usted podrá organizar sus trabajos en pestañas diferentes, para más tarde ser integrado en uno solo, por ejemplo un sitio web.
- Personalización de las barras de tareas: Mediante esta posibilidad usted hará su trabajo mucho más cómodo, pues podrá disponer en cada de tareas de las herramientas que realmente necesita y prescindir de las demás.

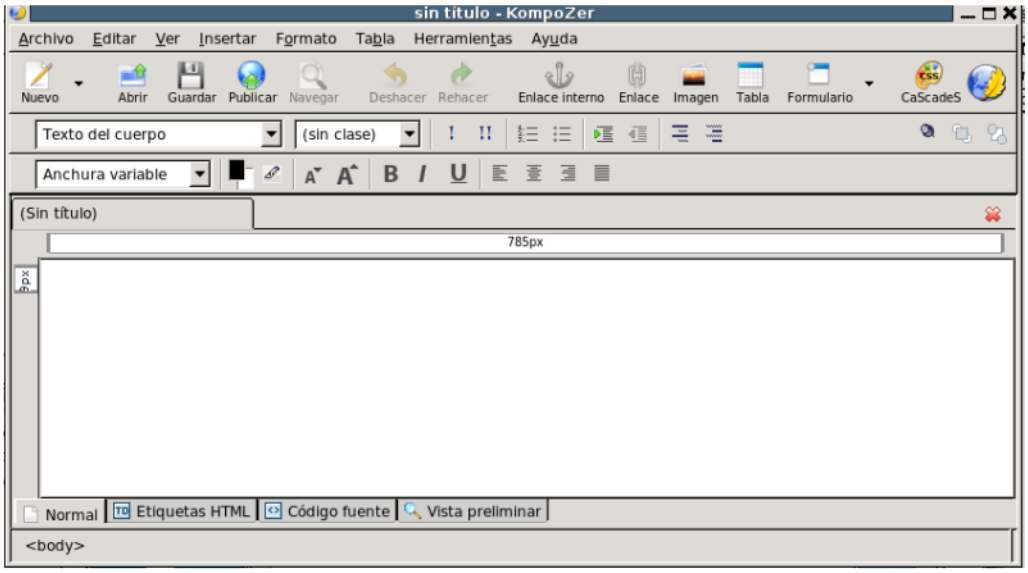
Usted pude obtener Kompozer en la siguiente URL:

versiós para Linux (8.2Mb):
http://sourceforge.net/project/showfiles.php?group_id=170132&package_id=194013&release_id=523233

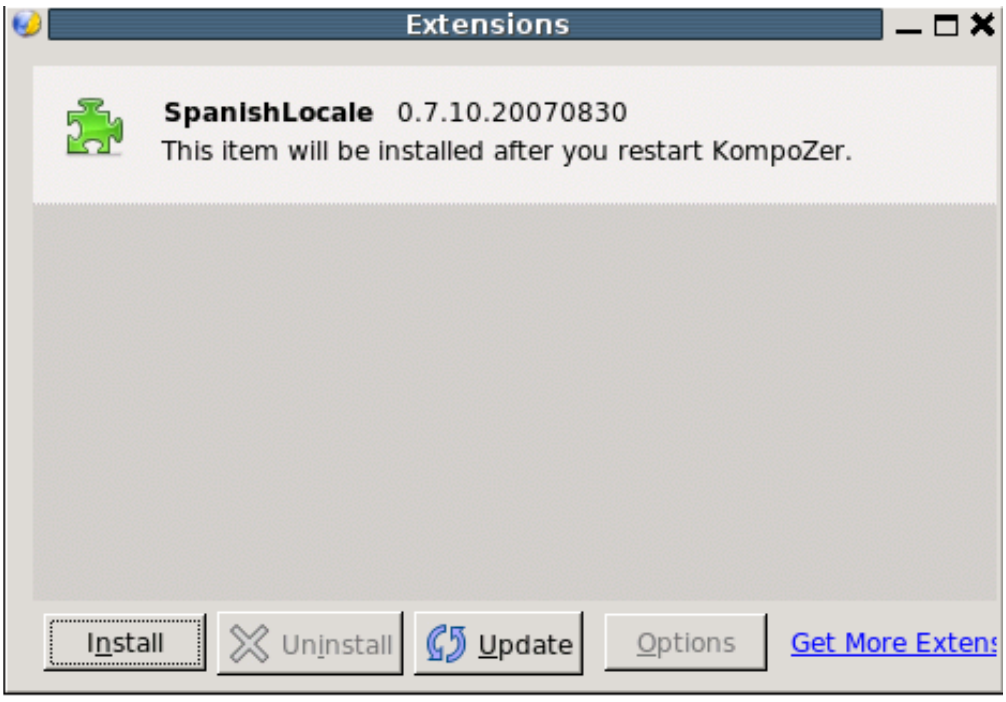
Plugins de idiomas (347.6):http://kompozer.sourceforge.net/lang/kz/kz-075-langpack-esES.xpi

Nota: Si usted es usuario de otro sistema operativo como por ejemplo Windows Kompozer también está disponible.
El entorno de trabajo de Kompozer.

Al iniciar Komposer usted verá una interfaz como la siguiente:



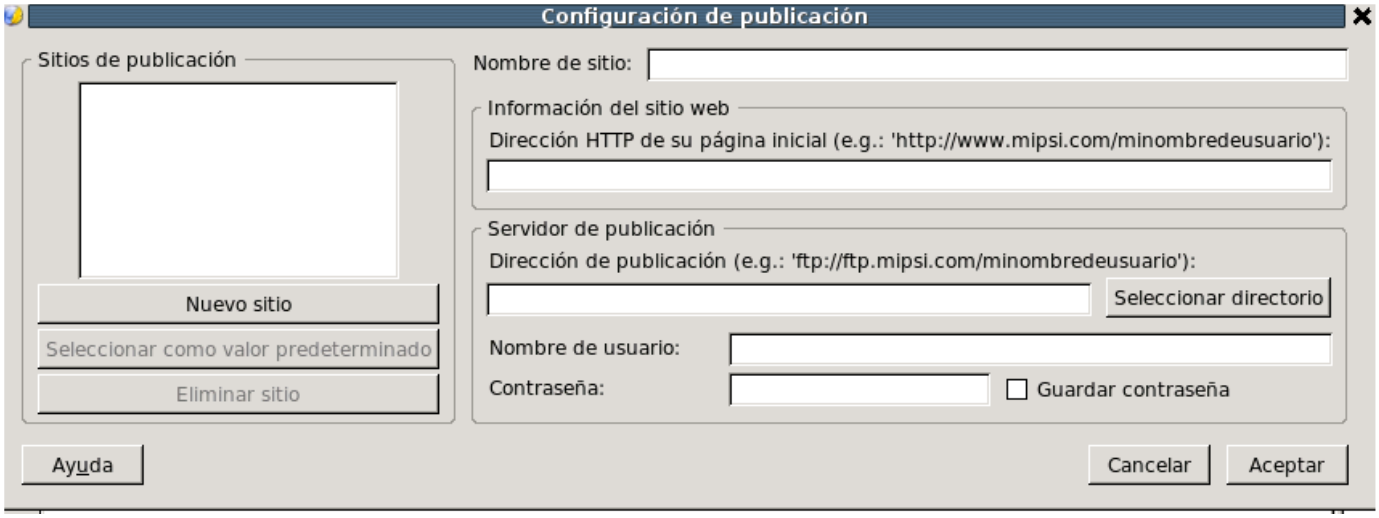
De seguro notará que el suyo una ves instalado a quedado en ingles, para traducirlo simplemente cuando halla descargado el plugins de idioma que menciono más arriba seleccione el menú tool y luego extensions. En la ventana que aparece damos clic en install y apuntamos hacia el directorio donde hemos descargado el plugins de idioma. Si todo fue correcto Kompozer mostrará la siguiente ventana:



indicando que el plugins ha quedado instalado y que tenemos que reiniciar Kompozer para que los cambios tengan efecto. Una ves reiniciada la aplicación ya estará en nuestro idioma nativo.

Con Kompozer tendremos la posibilidad de crear tanto páginas web individuales como sitios web completos y de gran calidad. Veamos un ejemplo del segundo caso.

Para crear un sitio web lo primero sería llegarnos al menú Editar y escomemos Configuración de publicación. La siguiente ventana se hará visible:



En este apartado podremos definir todo lo necesario para la creación de nuestro sitio web.

Haciendo clic en nuevo sitio se habilitarán las opciones de la derecha:

Nombre de sitio: Establecemos el nombre que llevará nuestro sitio web. Por ejemplo Manual de Kompozer, mi_sitio_web, etc.

Información del sitio web: URL de la página de inicio de nuestro sitio web. En nuestro caso podemos escoger un directorio local quedando de la siguiente manera: /home/mi_directorio_personal/mi_pimera_web/index.html

Servidor de publicación: Pulsando sobre el botón Seleccionar directorio podremos elegir el directorio donde alojaremos nuestro sitio web. Es conveniente la creación previa de este directorio para así tener organizados los elementos que compondrán nuestra web.

Nombre de Usuario y contraseña: Si para el directorio antes indicado se ha establecido nombre y contraseña de acceso coloque estos correspondientes valores en los campos indicados.

Aceptamos los cambio y listo.

Pasando a crear nuestra primera página web.

Para el que conozca otros entornos de desarrollo web la filosofía de trabajo no difiere mucho. Existen varias formas de trabajo entre ellas las fundamentales son en vista diseño y en vista código, la primera de ellas recomendadas para usuarios inexpertos como ya se dijo, así que asumamos esta modalidad.

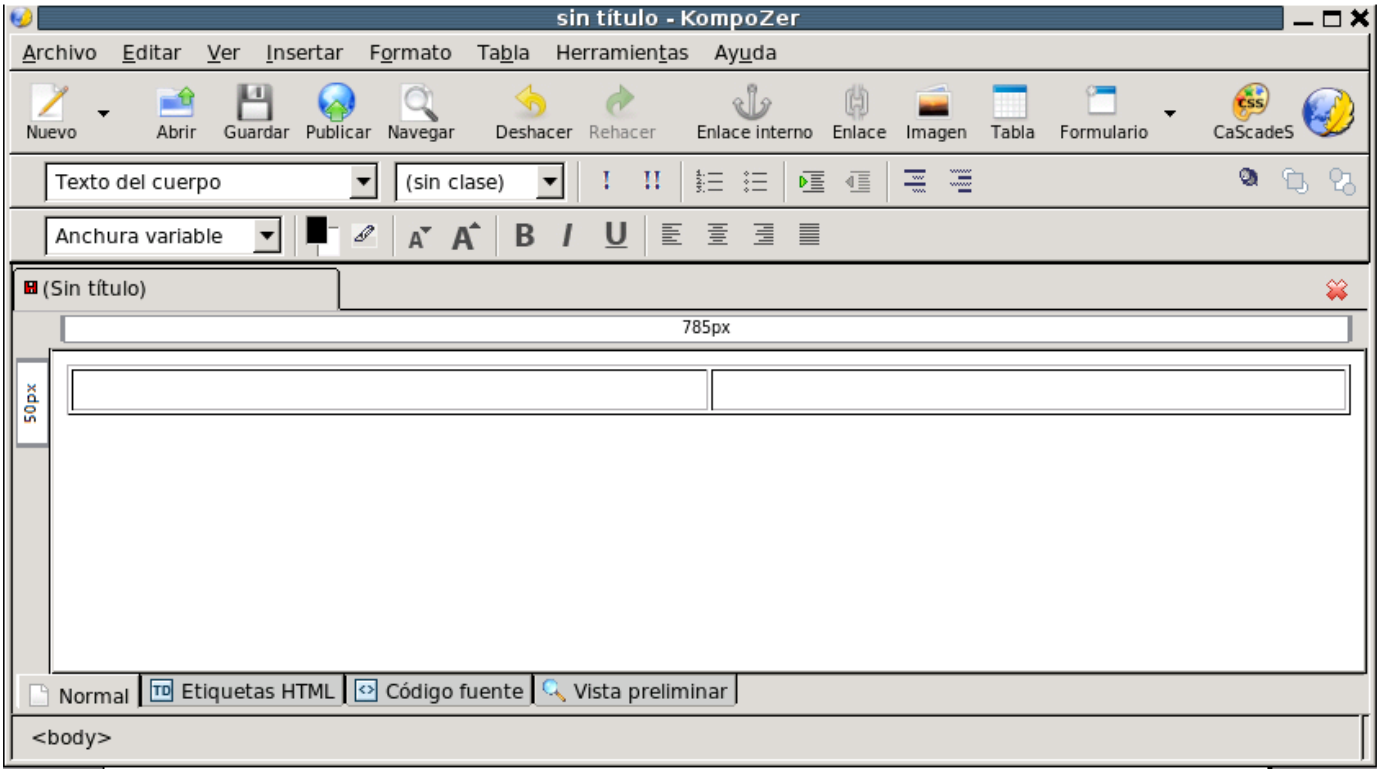
Como es conocido por todos las páginas web incorporan varios elementos como son:

- Imágenes
- Texto
- Flash
- Audio

Todos son posible incluirlos en un proyecto con Kompozer, veamos como poner por ejemplo Imágenes y texto.

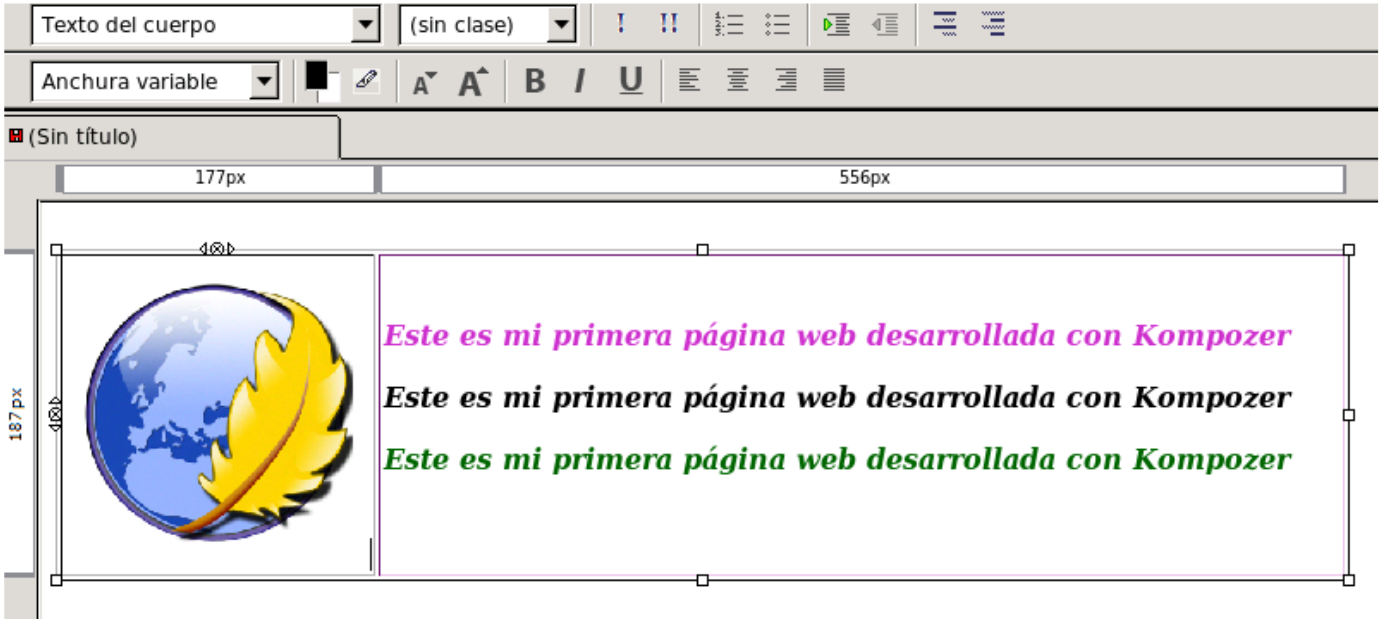
Para organizar más el trabajo y fundamentalmente la estética de nuestra web lo más cómodo es crear tablas que contendrán en cada una de sus celdas estos elemento. Supongamos que queremos insertar el propio logo de Kompose y junto a este un texto como por ejemplo "Este es mi primera página web desarrollada con Kompozer" OK manos a la obra.

En la barra de menú o en la de herramientas seleccione tabla, establezca los valores cantidad filas y columnas. En nuestro caso sera 1X2 es decir una fila y dos columnas. En la primera columna pondremos el logo de Kompozer y en la otra nuestro texto. Hecho esto tendremos algo así:



Para continuar copiamos al directorio que hemos creado anteriormente el logo de Kompozer y luego nos situamos en la primera celda de nuestra tabla. En el menú insertar seleccionamos imagen y apuntamos al directorio y dentro de esta escogemos el logo de Kompozer.

En la próxima celda escribimos el texto y listo. Tendremos algo más o menos así:



Como puede observar intencionalmente he repetido el texto con tres colores diferentes, usted también pude dar formato al texto usando la barra de formato que aparece en la parte superior del área de trabajo.

Si previsualizamos nuestro modesto trabajo en un navegador web veríamos el el logo, la tabla y el texto tal y como lo hemos desarrollado. Para eliminar la tabla simplemente hacemos clic derecho sobre ella y seleccionamos propiedades de la celda, escogemos la pestaña tabla y el valor borde lo establecemos en 0 y listo el borde de la celda se sustituye por una linea fina de color rojo que solo será visible en tiempo de diseño. Guardamos los cambios y presionando F5 visualizamos nuestra página en el navegador, pero ahora sin el borde de la tabla.

Esto es solo un pequeño comienzo en el mundo de Kompozer. Existe un manual muy completo de esta aplicación donde se dan instrucciones y ejemplos precisos de como sacar el máximo de esta potente herramienta de diseño web.

Hasta la próxima entrega.

Vladimir Valero.
vladimir08032@pri.jovenclub.cu

Opera 10.0

Navegador Web y Suit de Internet

Opera es un navegador web y suite de Internet creado por la empresa noruega Opera Software. La aplicación es gratuita desde su versión 8.50, habiendo sido previamente shareware o adware y antes de su versión 5.0, únicamente de pago.

Es reconocido por su gran velocidad, seguridad, soporte de estándares (especialmente CSS), tamaño reducido, internacionalidad y constante innovación. Implementó ya desde sus primeras versiones la navegación por pestañas, el Speed Dial, los movimientos del ratón en la navegación, personalización por sitio, y la vista en miniatura por pestaña

Está disponible para Windows, Mac OS X, GNU/Linux, OS/2, Solaris y FreeBSD. Además, hay dos versiones móviles: Opera Mini (móviles sencillos) y Opera Mobile (versiones específicas y de pago para teléfonos inteligentes y computadores de bolsillo). Por último, también está presente en las videoconsolas Nintendo DS y Wii. Se ha anunciado igualmente que el navegador estará disponible para televisores y reproductores DVD.

Su cuota de mercado global se sitúa en torno al 2,1 %

Según las estadísticas la cuota de mercado de Opera lo sitúa, con una gran diferencia, por detrás de Internet Explorer y Firefox. Sin embargo es importante mencionar que gran parte de los usuarios de Opera camuflan su navegador como otro ya que algunas páginas aún siguen sin identificar correctamente a este navegador.

Después de dos RC que se lanzaron recientemente, la versión final estable de Opera 10.0 ya está disponible. Algunas de las novedades en este lanzamiento son:

- Opera Turbo.- Un compresor que permite una navegación más rápida, pensada sobre todo para conexiones de poco ancho de banda o compartidas.
- Nuevo diseño y pestañas visuales.- Las pestañas muestran una previsualización de las páginas, pero sobre todo, si se arrastran se pueden obtener miniaturas de las mismas. Además se puede redimensionar cada pestaña.
- Integración Web.- Ahora Opera permite enlazar un cliente de correo (por ejemplo Thunderbird) para que gestione los enlaces de correo. También se puede determinar que lector de Feeds utilizar para las suscripciones RSS/Atom.
- Marcado rápido.- El Speed Dial o Marcado rápido puede configurarse para adjuntar de 2 a 25 sitios que pueden ser accedidos de manera rápida cuando se abra una pestaña nueva.
- Campo de búsqueda.- Se puede redimensionar el campo de búsqueda para adaptarlo a la cantidad de palabras que se va a ingresar, de modo que pueda ser más cómodo.
- 40% más rápido.- El nuevo motor Opera presto 2.2 hace al navegador 40% más rápido que su antecesor.

Opera se ha conocido como uno de los navegadores más rápidos, innovando constántemente con características incluidas, sin necesidad de extensiones. Todavía queda por incluir su nuevo motor JavaScript que dicen será el más rápido del mercado, el cual competirá con los motores de Firefox y Google Chrome que para versiones futuras serán mejorados considerablemente.

Instalación

Para la mayoría de las distribuciones Linux hay paquetes binarios para su fácil instalación o en su defecto en forma ejecutable, sin necesidad de instalar. Opera esta disponible desde:
http://www.opera.com/browser/download/
donde debemos seleccionar el sistema operativo para el cual queremos la instalación.

En nuestro caso el sistema operativo utilizado fue Ubuntu 9.04. Una ves descargado damos clic derecho sobre el paquete .deb y elegimos la opción Abrir con Instalador de paquetes Gdebi en el cuadro de diálogo que aparecerá hacemos clic en el botón Instalar el Paquete y en unos pocos segundos tendremos nuestro nuevo navegador listo.

Una vez instalado lo podemos abrir a través del menú Aplicaciones – Internet – Opera el cual tendrá un aspecto como la siguiente imagen:

Interfaz

Opera posee una interfaz muy parecida a los demás navegadores, Barra de Menú, Pestañas de navegación, Barra de dirección, etc... se ve muy limpio y pulido. Los dos botones añadidos a la barra de navegación hacen que el usuario pueda ir hacia el final o el principio de todas las páginas cargadas bajo el dominio que está visitando. Imagen2.

Pestañas

Una de las cosas que más se destaca en esta nueva versión son las pestañas con vista previa, esta función la están adoptando muchos otros software y Opera no ha sido la excepción. Cuando tenemos varias pestañas abiertas solo con pasar el Mouse por encima obtendremos una vista previa del contenido que posee la página, también es posible arrastrar hacia abajo la barra de navegación y veremos una vista previa de todas las páginas que tenemos abiertas en ese momento. Lo que sin dudas apreciarán es el botón Deshacer Pestañas Cerradas ubicado a la derecha de la barra de navegación permitiendo recuperar alguna pestaña cerrada accidentalmente. Imagen3.

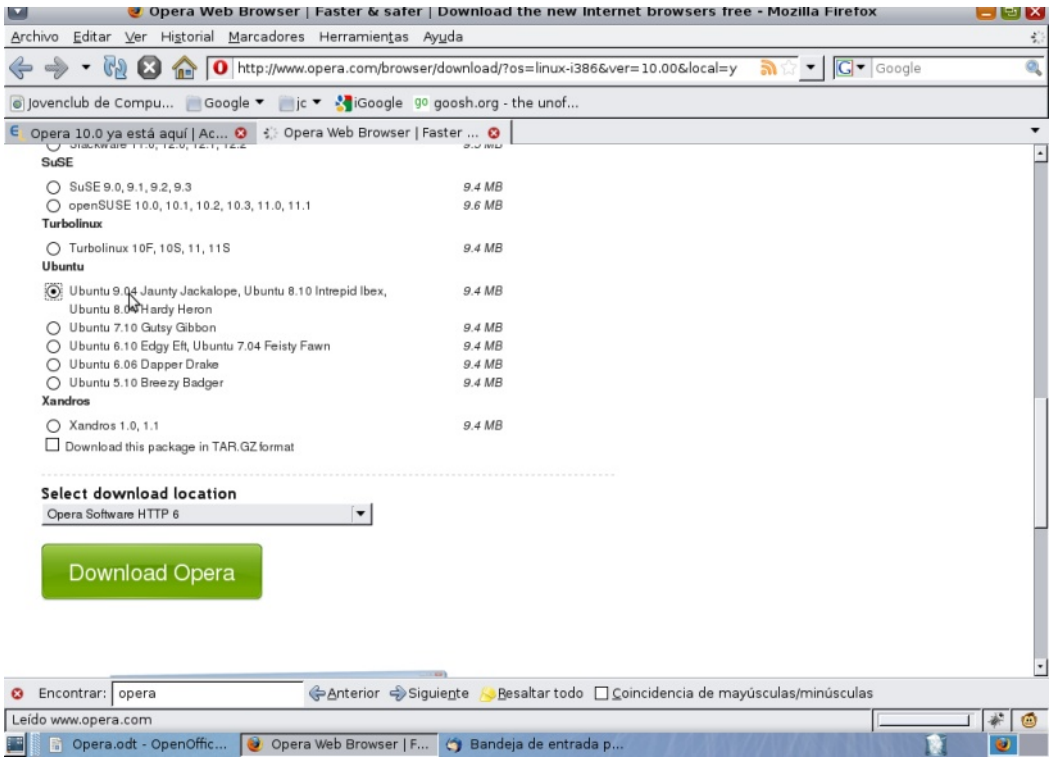
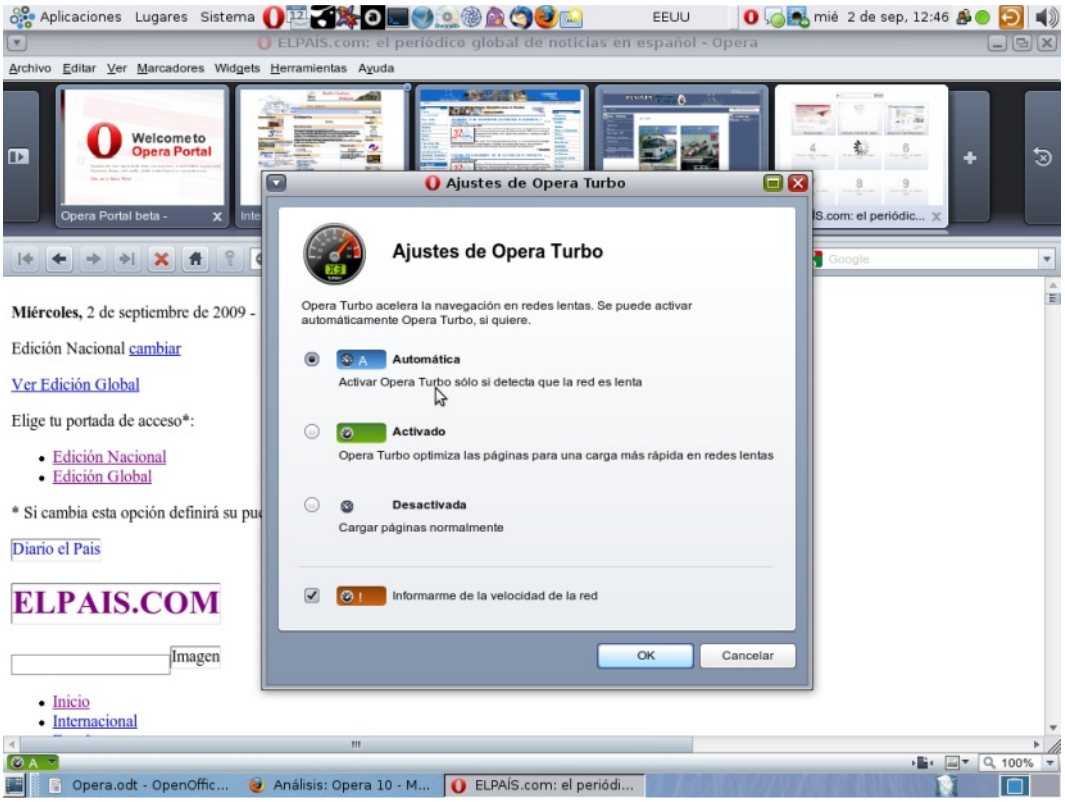


Imagen 2

Imagen 3

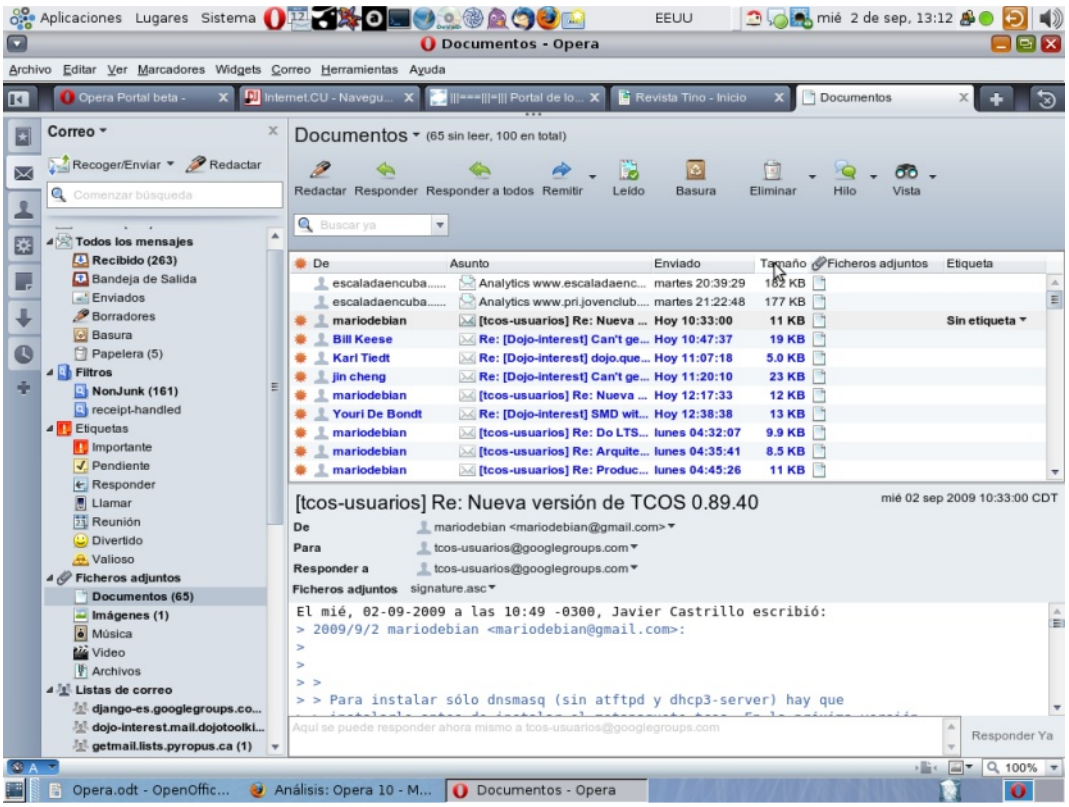
Opera Turbo

Opera Turbo viene para contrarrestar los efectos de las conexiones lentas, al comprimir el contenido de las páginas a través de sus propios servidores, Opera puede lograr un rendimiento de navegación superior para aquellos usuarios que cuentan con una velocidad de navegación limitada. Los que posean una conexión de banda ancha normal no deben activar esto porque obtendrán todo lo contrario. El modo turbo se puede configurar a través del icono que aparece a la izquierda de la barra de estado. Como se puede observar en la figura posee tres estados, Automático, Activado y Desactivado.



Correo

Opera permite configurar un cliente de correo a través del menú Herramientas – Cuenta de Correo y Chat. Con solo unas pocas opciones tenemos configurado un cliente de correo con una cantidad de opciones increíbles, permite visualizar los correos por contenidos, o sea, se pueden visualizar todos los correos, las listas de correo a las que estamos subscritos, los correos con documentos adjuntos, etc..., es muy cómodo responder a un correo ya que en la parte inferior del mensaje viene un pequeño cuadro de texto para responder al remitente de forma rápida, en fin, son muchas las opciones que posee.



Speed Dial

Speed Dial (Accesos Rápidos) permite ir a sitios Web preferidos cuando se abre una nueva pestaña. Se hace clic en un área vacía para añadir una nueva entrada, se borra haciendo clic en la equis(x) y se arrastran los iconos para ordenarlos. Es posible cambiar el fondo de los Speed Dial así como el número a mostrar a través de las opciones que hay en la parte inferior de los mismos. Los Accesos Rápidos han tenido una gran aceptación entre los usuarios que usan Opera.

Widgets

En la página de Widgets de Opera hay más de mil doscientos de ellos, y creo que este es un punto débil sobre todo para los desarrolladores Web, he recorrido todos los widgets de Desarrollo Web y no he encontrado ninguno que se parezca a Firebug o ayude al debug de sitios con javascript y ajax. Imagen 7.

Estandares

Opera cumple a la perfección con los estándares y así lo demuestra el triturados de navegadores Acid3. **Imagen 8**

Conclusiones.

Su interfaz es muy agradable y las opciones de configuración son más que suficiente pero para quienes intenten pisar el acelerador no obtendrán buenos resultados sobre todo si competimos contra otros navegadores como FireFox. Opera 10.0 es un navegador sólido frente a sus anteriores versiones. Aunque aún es muy temprano hay que esperar a que Opera saque el nuevo motor Java Script que según ellos será el más rápido del mercado. Estas son algunas de las novedades más significativas de Opera 10.0

Fuentes:

www.opera.com

www.wikipedia.org
http://www.neoteo.com/analisis-opera-10.neo

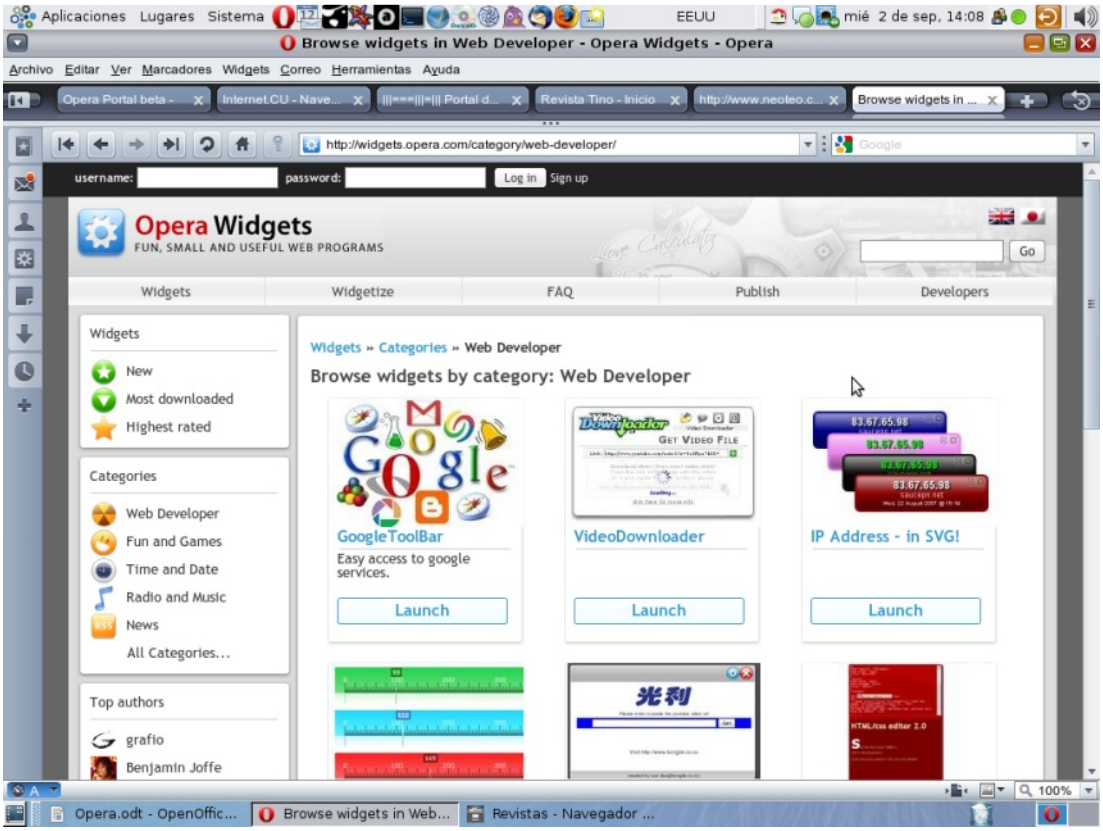


Imagen 7

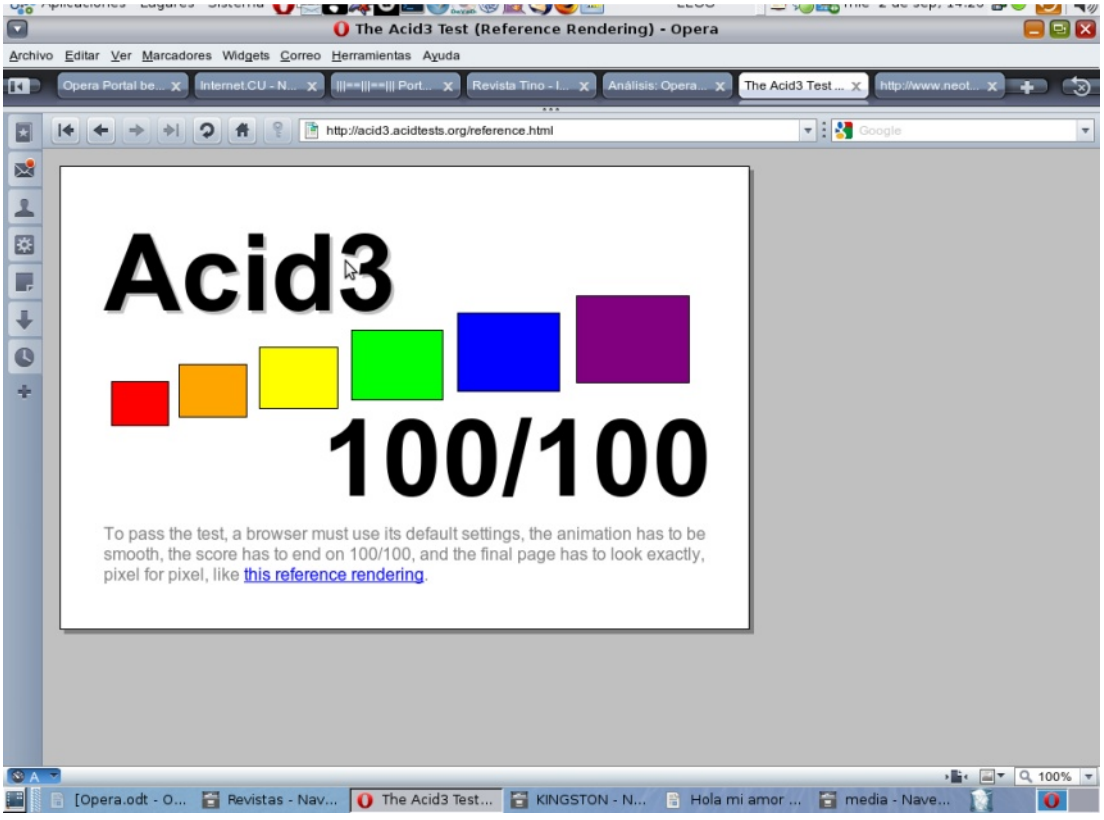


Imagen 8

Roylan Suáres Reyes
roylan04012@pri.jovenclub.cu

SVN Control de Versiones

Parte II

En el número anterior habíamos visto como configurar un servidor Subversion para el control de versiones asi como las ventajas que esto trae para el desarrollo en equipo. En este artículo continuamos con la segunda parte donde explicaremos como usar un cliente para descargar y actulizar las versiones de un proyecto en cual trabajamos.

Subversion es un sistema de control de versiones libre y de código fuente abierto. Es decir, Subversion maneja ficheros y directorios a través del tiempo. Hay un árbol de ficheros en un repositorio central. El repositorio es como un servidor de ficheros ordinario, excepto porque recuerda todos los cambios hechos a sus ficheros y directorios. Ésto le permite recuperar versiones antiguas de sus datos, o examinar el historial de cambios de los mismos. En este aspecto, mucha gente piensa en los sistemas de versiones como en una especie de “máquina del tiempo ”

Para el desarrollo de este articulo vamos a suponer que el servidor svn se encuentra en el la direccion IP 192.168.0.61 y el proyecto de trabajo se llama Prueba, asi la url sería:

http://192.168.0.61/svn/prueba/

¡Ayuda!

Antes de seguir leyendo, aquí está el comando más importante que usted necesitará cuando esté usando Subversion: svn help. El cliente de línea de comandos Subversion está auto documentado en cualquier momento, un svn help <subcomando> rápido describirá la sintaxis, las opciones y el comportamiento del subcomando.

Descarga inicial

La mayor parte del tiempo, usted empezará a usar un repositorio de Subversion haciendo un checkout de su proyecto . Descargar un repositorio crea una copia de éste en en su máquina local. Esta copia contiene el HEAD (última revisión) del repositorio de Subversion que usted especifica en la línea de comandos:

```
$  svn checkout http://192.168.0.61/svn/prueba/trunk/ prueba
A    prueba/manage.py
A    prueba/.project.screem
A    prueba/mogote.webprj
A    prueba/galeria
A    prueba/galeria/views.py
A    prueba/galeria/__init__.py
A    prueba/galeria/models.py
A    prueba/__init__.py
A    prueba/internacional
A    prueba/internacional/views.py
A    prueba/internacional/__init__.py
A    prueba/internacional/models.py
…..
A    prueba/templates/internacionales/inter_detalle.html
Revisión obtenida: 1
```

En el ejemplo de arriba hemos descargado el directorio trunk hacia el directorio prueba pero tambien es posible descargar cualquier directorio más profundo especificandolo en la url del repositorio.

Usted ya puede empezar a realizar cambios a los ficheros y directorios en su copia de trabajo local. Su copia local es justo como cualquier otra colección de ficheros y directorios en su sistema. Usted puede editarlos y cambiarlos, moverlos, usted puede incluso borrar la copia local entera y olvidarse de ella.

Comando Básicos de Trabajo

Subversión tiene numerosos comandos y opciones que se necesitaría un libro para exponerlo en su totalidad, nos centraremos en los comandos más básicos para el desarrollo de este articulo.

Actualizar su copia de trabajo local
Cuando se trabaja en un proyecto con un equipo, usted querrá actualizar su copia de trabajo local para recibir cualquier cambio hecho desde su última actualización por otros desarrolladores en el proyecto. Use svn update para sincronizar su copia de trabajo local con la del repositorio.

```
roylan@linvix:$ svn update
U  contador.py
U  sendmail.py
En la revisión 47.
```

En este caso, alguien envió modificaciones a contador.py y sendmail.py desde la última vez que usted actualizó, y Subversion ha actualizado su copia de trabajo local para incluir estos cambios.
Vamos a examinar la salida de svn update un poco más. Cuando el servidor envía cambios a su copia de trabajo local, un código de letras es mostrado al lado de cada elemento para hacerle saber qué acciones realizó Subversion para actualizar su copia de trabajo local:

U contador.py
El fichero contador.py fue actualizado.

A contador.py
El fichero o directorio contador.py fue Añadido a su copia de trabajo local.

D contador.py
El fichero o directorio contador.py fue borrado de su copia de trabajo local.

R contador.py
El fichero o directorio contador.py fue Reemplazado en su copia de trabajo local; esto es, contador.py fue borrado, y un nuevo objeto con el mismo nombre fue añadido. Aunque pueden tener el mismo nombre, el repositorio los considera objetos distintos con historiales distintos.

G contador.py
El fichero contador.py recibió nuevos cambios del repositorio, pero su copia local del fichero tenía las modificaciones que ud. ya había hecho. O los cambios no se intersectaron , o los cambios eran exactamente iguales que sus modificaciones locales, así que Subversion ha fusionado satisfactoriamente los cambios del repositorio en el fichero sin ningún problema.

C contador.py
El fichero contador.py recibió cambios Conflicting del servidor. Los cambios del servidor directamente se superpusieron sobre sus propios cambios en el fichero.

Hacer cambios en su copia de trabajo local

Ahora puede conseguir trabajar y hacer cambios en su copia de trabajo local. Generalmente es más conveniente decidir un cambio particular a hacer, por ejemplo añadir un nuevo fichero, escribir una nueva característica, corregir un fallo, etc...

Modifique y cree ficheros/directorios en su copia local, veremos como es posible actualizar el repositorio con los nuevos cambios realizados en la copia de trabajo local.

Añadir ficheros y directorios al repositorio

```
$ svn add plantillas
A         plantillas
```

Programa añadir plantillas al repositorio. Cuando haga su próximo envío, plantillas se convertirá en hijo de su directorio padre. Fíjese que si plantillas es un directorio, todo por debajo de plantillas será programado para la adición . Si solo quiere añadir el propio plantillas, pase la opción --non-recursive (-N) .

Borrar ficheros y directorios del repositorio

```
$ svn delete sendmail
```

Programa borrar sendmail del repositorio. Si sendmail es un fichero, se borrará inmediatamente de su copia de trabajo local. Si sendmail es un directorio, este no es borrado, pero Subversion lo programa para borrarlo. Cuando envíe sus cambios, sendmail será borrado de su copia de trabajo y del repositorio.

Crear copias

```
$ svn copy sendmail.py enviar.py
```

Crea un nuevo objeto enviar como duplicado de sendmail. enviar es automáticamente programado para la adición.

Examine sus cambios

Una vez que haya terminado de hacer cambios, necesita enviarlos al repositorio, pero antes de hacerlo, generalmente es buena idea echar un vistazo a lo que ha cambiado exactamente.

Puede descubrir que ha cambiado inadvertidamente un fichero, y esto le da la posibilidad de invertir esos cambios antes de enviarlos.

```
$ svn status
?      templates
?      devweb.webprj
M       locale/en/LC_MESSAGES/django.po
M       locale/en/LC_MESSAGES/django.mo
A       contador.py
M       plantillas/invest/index.html
L       settings.py
~       noticias/views.py
A  +    enviar.py
?       media_dev/investigacion/2009/repo.jpg
```

Los códigos impresos en la primera columna son:

A file_or_dir
El fichero o directorio file_or_dir ha sido programado para la adición en el repositorio.

C file
El fichero file está en un estado de conflicto. Esto es, los cambios recibidos del servidor durante una actualización se solapan con cambios locales que usted tiene en su copia de trabajo. Debe resolver este conflicto antes de enviar sus cambios al repositorio.

D file_or_dir
El fichero o directorio file_or_dir ha sido programado para la supresión del repositorio.

M file
El contenido del fichero file ha sido modificado.

X dir
El directorio dir está sin versionar, pero está relacionado con una definición externa de Subversion.

? file_or_dir
El fichero o directorio file_or_dir no está bajo control de versiones.

! file_or_dir
El fichero o directorio file_or_dir está bajo el control de versiones pero falta o está de alguna manera incompleto.
~ file_or_dir
El fichero o directorio file_or_dir está en el repositorio como un tipo de objeto, pero actualmente está en su copia de trabajo como otro tipo. Por ejemplo, Subversion pudo tener un fichero en el repositorio, pero usted borró el fichero y creó un directorio en su lugar, sin usar los comandos svn delete o svn add.

I file_or_dir
Subversion está “ignorando” el fichero o directorio file_or_dir, probablemente porque usted se lo dijo.

La segunda columna dice el estado de las propiedades de un fichero o un directorio. Si aparece una M en la segunda columna, entonces las propiedades han sido modificadas, si no un espacio en blanco será impreso.

La tercera columna solo mostrará un espacio en blanco o una L la cual significa que Subversion ha bloqueado el objeto en el área de trabajo .svn. Usted verá una L si ejecuta svn status en un directorio donde un svn commit esté en progreso.

La cuarta columna solo mostrará un espacio blanco o un + el cual significa que el fichero o directorio está programado para ser añadido o modificado con historial adicional adjunto.

La quinta columna solo mostrará un espacio en blanco o una S. Esto significa que el fichero o directorio ha sido movido de la ruta del resto de la copia de trabajo.

$ svn diff

Otra manera de examinar sus cambios es con el comando svn diff. Puede descubrir exactamente cómo ha modificado cosas ejecutando svn diff sin argumentos, el cual imprime los cambios de los ficheros en formato unificado del diff:

```
Index: locale/en/LC_MESSAGES/django.po
===================================================================
==
--- locale/en/LC_MESSAGES/django.po      (revisión: 48)
+++ locale/en/LC_MESSAGES/django.po    (copia de trabajo)
@@ -8,7 +8,7 @@
msgstr ""
"Project-Id-Version: PACKAGE VERSION\n"
"Report-Msgid-Bugs-To: \n"
-"POT-Creation-Date: 2009-07-01 17:13-0400\n"
+"POT-Creation-Date: 2009-07-02 14:39-0500\n"
"PO-Revision-Date: YEAR-MO-DA HO:MI+ZONE\n"
"Last-Translator: FULL NAME <EMAIL@ADDRESS>\n"
"Language-Team: LANGUAGE <LL@li.org>\n"
@@ -142,40 +142,55 @@
….
….
```

El comando svn diff produce esta salida comparando sus ficheros de copia de trabajo contra la copia almacenada en el área .svn. Los ficheros programados para la adición se visualizan como texto-añadido, y los ficheros programados para la eliminación son visualizados como texto eliminado.

svn revert

Ahora suponga que usted ve la salida del diff anterior, y se da cuenta que sus cambios a django.po son un error; quizás accidentalmente tecleó ese texto en el fichero equivocado en su editor. Esta es una oportunidad perfecta para usar svn revert

```
$ svn revert django.po
Revertido 'django.po'
```

Subversion invierte el fichero a un estado pre-modificado reescribiéndolo con la copia almacenada en el área .svn. Pero también observar que svn revert puede deshacer cualquier operación programada por ejemplo, usted puede decidir que no quiere añadir un nuevo fichero después de todo:

```
$ svn status enviar.py
?      enviar.py
```

```
$ svn add enviar.py
A         enviar.py
```

```
$ svn revert enviar.py
Revertido 'enviar.py'
```

```
$ svn status enviar.py
?      enviar.py
```

Resolver conflictos

Ya hemos visto cómo svn status -u puede predecir conflictos. Suponga que ejecuta svn update y ocurren algunas cosas interesantes

```
$ svn update
U  INSTALL
G  README
C  enviar.py
Updated to revision 5
```

Los códigos U y G no son causa para la inquietud; esos ficheros absorbieron limpiamente los cambios del repositorio. Los ficheros marcados con una U no contienen cambios locales pero fueron actUalizados con cambios del repositorio. La G representa merGed, lo que significa que el fichero tenía para comenzar cambios locales, pero los cambios que venían del repositorio no se solaparon de ninguna manera.

Pero la C representa conflicto. Esto significa que los cambios del servidor se solapan con los suyos propios, y ahora tiene que elegir manualmente entre ellos.

Siempre que ocurra un conflicto, ocurren tres cosas para ayudarle a notar y resolver ese conflicto:

- Subversion imprime una C durante la actualización, y recuerda que el fichero está en un estado de conflicto.
- Subversion coloca marcas de conflicto—secuencias especiales de texto que delimitan los “lados” del conflicto—en el fichero para demostrar visualmente las áreas solapadas.
- Para cada fichero en conflicto, Subversion coloca tres ficheros extra en su copia de trabajo local:

filename.mine
Este es su fichero como existió en su copia de trabajo antes de que usted actualizara su copia de trabajo esto es, sin marcas de conflicto. Este fichero tiene su últimos cambios y nada más.

filename.rOLDREV
Este es el fichero que era la revisión BASE antes de que usted actualizará su copia de trabajo. Esto es, el fichero que usted descargó antes de que hiciera su última edición.

filename.rNEWREV
Este es el fichero que su cliente de Subversion recibió del servidor justo cuando usted actualizó su copia de trabajo. Este fichero corresponde con la revisión HEAD del repositorio.
Aquí OLDREV es el número de revisión del fichero en su directorio .svn y NEWREV es el número de revisión del HEAD del repositorio.

Por ejemplo, Javier hace cambios al fichero sendmail.py en el repositorio. Yoel acaba de cambiar el fichero en su copia de trabajo y lo ha enviado. Javier actualiza su copia de trabajo antes de enviarlo y recibe un conflicto:

```
$ svn update
C  sendmail.py
Updated to revision 2.
```

```
$ ls -1
sendmail.py
sendmail.py.mine
sendmail.py.r1
sendmail.py.r2
```

En este punto, Subversion no le permitirá enviar el fichero sendmail.py hasta que los tres ficheros temporales sean borrados.

```
$ svn commit --message "Update sendmail.py"
svn: Commit failed (details follow):
svn: Aborting commit: '/home/javier/projects/vi/svn-work/sendmail.py' remains in conflict
```

Si obtiene un conflicto, necesita hacer una de tres cosas:

- Fusionar el texto en conflicto “a mano” (examinando y editando las marcas de conflicto dentro del fichero).
- Copiar uno de los ficheros temporales sobre su fichero de trabajo.
- Ejecutar svn revert <filename> para eliminar todos sus cambios locales.

Una vez que usted haya resuelto el conflicto, necesita dejar que Subversion lo sepa ejecutando svn resolved. Esto borrará los tres ficheros temporales y Subversion no considerará por más tiempo que el fichero está en estado de conflicto.

```
$ svn resolved sendmail.py
Resolved conflicted state of 'sendmail.py'
```

Enviar sus cambios

¡Finalmente! Su edición está terminada, ha fusionado todos los cambios del servidor, y está listo para enviar sus cambios al repositorio.
El comando svn commit envía todos sus cambios al repositorio. Cuando usted envía un cambio, necesita proveer un mensaje de registro, describiendo su cambio. Su mensaje de registro será adjuntado a la nueva revisión que ha creado. Si su mensaje de registro es breve, puede querer proveerlo en la línea de comando usando la opción --message (o -m):

```
$ svn commit --message "Actualizando cambios del modulos sendmail"
Sending        sendmail.py
Transmitting file data .
Committed revision 3.
```

Examinando el historial
Como hemos mencionado anteriormente, el repositorio es como una máquina del tiempo. Este mantiene un expediente de cada cambio enviado, y le permite explorar este historial examinando versiones anteriores de ficheros y directorios así como los metadatos que los acompañan. Con un único comando de Subversion, puede descargar el repositorio exactamente como era en cualquier fecha o número de revisión en el pasado. Sin embargo, a veces solo desea mirar al pasado en vez de ir al pasado.

Hay varios comandos que pueden proporcionarle datos históricos del repositorio:

svn log
Le muestra amplia información: mensajes de registro unidos a las revisiones, y que ruta de fichero cambió en cada revisión.

svn diff
Le muestra los detalles específicos de cómo cambió un fichero en un cierto plazo.

svn cat
Este se utiliza para recuperar cualquier fichero tal como existió en un un número de revisión particular y lo muestra en su pantalla.

svn list
Muestra los ficheros en un directorio para cualquier revisión dada.

Bueno hasta aquí este artículo, aunque está bastante extenso intenté incluir la manor cantidad de texto y que no se quedará nada util. Svn necesita todo un libro, espero que sea de su utilidad.

Bibliografia Consultada
http://svnbook.red-bean.com
http://www.wikipedia.es

Roylan Suárez Rodríguez
roylan04012@pri.jovenclub.cu

Firewall iptables

Y Herramientas de Seguridad

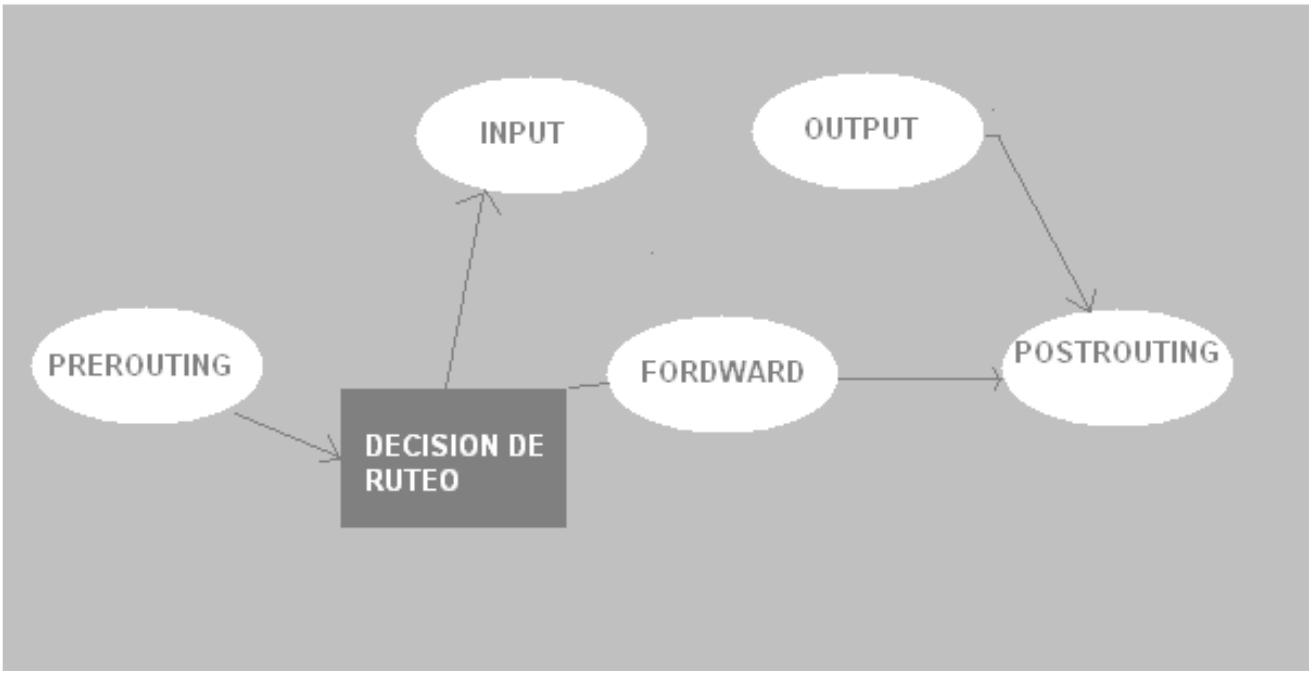
Bueno voy a explicar brevemente la instalación y configuración del firewall iptables de Linux, pero voy a hacer hincapié en el concepto y la seguridad que éste y otros comandos nos brindan para prevenir ataques y sólo habilitemos en nuestro servidor los servicios necesarios.

La instalación se realiza simplemente con un:

```
sudo apt-get install iptables
```

El concepto de iptables es que un paquete para por ciertas CADENAS. Una cadena es solo una lista de reglas que se le van a aplicar al paquete.

Estas cadenas son gráficamente:



1 - PREROUTING: Es la primera contra la que choca un paquete antes de entrar a nuestro sistema. Desde un punto de vista de NAT, este es el punto ideal para realizar un DNAT, con la cual se cambia la dirección ip destino del paquete.

2 – FORDWARD: Esta se utiliza en caso de reenvío de paquetes, donde el destinatario no es el servidor donde estamos parados si no algún otro.

3 – INPUT: Aquí se aplican las reglas para el servidor local. Como por ejemplo habilitar el puerto 80 a alguna red en particular o algún otro servicio como el de ssh para una ip en particular, en este punto vamos a decidir qué y a quién le habilitamos/deshabilitamos los servicios que brinda el servidor anfitrión.

4 – OUTPUT: Aquí habilitamos /deshabilitamos los puertos y/o protocolos de los paquetes que salen de nuestro servidor, como por ejemplo no queremos que nuestro servidor se conecte por ssh o Telnet a algún otro servidor.
5 – POSROUTING: En esta cadena se pueden alterar la dirección ip fuente para los fines de Source Nat (SNAT).

Bueno vamos tirar un poco de comandos para salir del teórico que siempre es muy aburrido.

Antes que nada se define una política por defecto para todos los paquetes, por ejemplo podemos decidir bloquear todo y después aplicamos las políticas para ir aceptando los paquetes o a la inversa aceptamos todo y vamos bloqueando lo que no nos interesa:

Ejemplo:

```
iptables – P INPUT ACCEPT        # POR DEFECTO ACEPTAMOS TODO
iptables – P FORDWARD DROP  # POR DEFECTO NO PERMITIDOS REENVIOS
iptables – P OUTPUT ACCEPT     # POR DEFECTO PERMITIMOS LA SALIDA
```

Esto lo definimos de acuerdo a nuestras necesidades, pero debemos conocer bien nuestro servidor, porque si establecemos que vamos a bloquear todo y vamos aceptando hay que tener en cuenta que los servidores descargan actualizaciones y si sin darnos cuenta bloqueamos la salida, siempre tendremos nuestro servidor disponible a vulnerabilidades.

Ejemplos Sencillos para entender el concepto:

- Queremos habilitar el puerto tcp 80:

```
iptables –A INPUT –p tcp  - - dport 80  - j ACCEPT
```

- Queremos que una red en particular (192.168.1.0) no pueda usar nuestro servidor bajo ninguna circunstancia:

```
iptables –A INPUT –s 192.168.1.0/24  –j DROP
```

Si somos un servidor intermedio y no el anfitrión y hacemos reenvío hacia otro servidor:

```
iptables –A FORDWARD –d 10.120.0.0/24 –j ACCEPT
```

Esta cadena deja pasar todos los paquetes que van por FORDWARD con destino a la red 10.120.0.0.

Esto es lo básico de iptables, el tráfico se puede ver con herramientas como iptraf que nos mostrarán el movimiento de los paquetes.

Otros conceptos a tener en cuenta es que se pueden cargar objetivos adicionales, por ejemplo se puede cargar el objetivo REJECT el cual dejará caer el paquete y le enviará de regreso un error de icmp al remitente.

Para bloquear un ping:

iptables – A INPUT –m icmp –icmp-type echo-request

Para obtener ayuda para el módulo icmp hacemos:

iptables –m icmp –h , y nos mostrará una lista completa de los tipos de paquetes icmp soportados:

Todas estas reglas se deben poner en un script y dar permiso de ejecución para que se levanten cuando nuestro sistema se inicia.

Esto es un vistazo general, es muy amplio el tema de iptables pero quería hacer hincapié en conceptos generales para entender el concepto.

Para saber actualmente que puertos tenemos abiertos ejecutamos el comando:

```
nmap –sT localhost


PORT     STATE SERVICE
22/tcp   open  ssh
25/tcp   open  smtp
80/tcp   open  http
111/tcp  open  rpcbind
139/tcp  open  netbios-ssn
443/tcp  open  https
445/tcp  open  microsoft-ds
631/tcp  open  ipp
3306/tcp open  mysql
```

Acá observamos los puertos que tengo habilitados, ahora tendremos que garantizar que si son servicios que son necesarios brindar o no, pero antes de desinstalarlos por las dudas primero lo bloqueamos con los comandos mostrados anteriormente.

Una vez definido nuestras reglas podemos observar el tráfico como dije anteriormente con iptraf, pero hay una herramienta muy útil para ello que se llama tcpdump.

tcpdump nos permitirá capturar el tráfico mediante comandos que le especifiquemos, por ejemplo: si quiero ver el tráfico sólo de los puertos 80 y 53 hacemos esto:

```
tcpdump –n port 80 or port 53
```

Capturar el tráfico cuya IP origen sea 192.168.1.1

```
tcpdump src host 192.168.1.1
```

Capturar todo el tráfico con destino al puerto 25

```
tcpdump dst port 25
```

Capturar todo el tráfico excepto el web

tcpdump tcp and not port 80
Supongamos que bloqueamos sin querer el puerto del mail (25) y nos hartan a llamados que no pueden enviar mails, con tcpdump podemos observar filtrar el tráfico al puerto 25 y veremos que se queda en nuestro servidor, entonces nos daremos cuenta y lo habilitaremos nuevamente.

```
root@pc:~# tcpdump -n port 25

tcpdump: verbose output suppressed, use -v or -vv for full protocol decode

listening on eth0, link-type EN10MB (Ethernet), capture size 96 bytes
```

Vemos que no muestra nada, bloqueamos:

iptables -A INPUT -p tcp --dport 25 -j DROP

enviamos un mail:

```
echo "test" | mail -s " test" mail@dominio.com
```

y veremos que el tcpdump no muestra ningún movimiento sobre este puerto, ahora vamos a habilitarlo:

iptables -F # solo para prueba borro todas las políticas

enviamos un mail:

```
echo "test" | mail -s " test" mail@dominio.com
```

y en tcpdump:

```
16:07:27.156851 IP 192.168.1.66.25985 > 210.22.33.55.25: S
1160722584:1160722584(0) win 5840 <mss 1460,sackOK,timestamp 60514227
0,nop,wscale 6>

16:07:27.158272 IP 210.22.33.55.25 > 192.168.1.66.25985: S
4049060577:4049060577(0) ack 1160722585 win 5792 <mss 1460,sackOK,timestamp
485010362 60514227,nop,wscale 0>

16:07:27.158337 IP 192.168.1.66.25985 > 210.22.33.55.25: . ack 1 win 92
<nop,nop,timestamp 60514227 485010362>

16:07:27.205393 IP 210.22.33.55.25 > 192.168.1.66.25985: P 1:40(39) ack 1 win 5792
<nop,nop,timestamp 485010367 60514227>
```

vemos que hay movimiento.

Di un vistazo general de lo que hay que tener un cuenta para la seguridad, empezamos por entender el concepto de iptables , luego con nmap vimos que puertos tenemos habilitados y si son necesarios o no , luego con tcpdump aprendimos a capturar tráfico. Tantos los conceptos de iptables, nmap y tcpdump son muy extensos para explicarlos a todos, pero esta es una guía básica para poder empezar.

Ing. Roque Moyano
Administrador de Servidores Linux y base de datos Oracle
Argentina
roque.ing@gmail.com

Django

framework de desarrollo web

Django es un framework de desarrollo web de código abierto escrito en Python, que te permite hacer aplicaciones web de una manera muy rápida cumpliendo con el patrón de diseño de arquitectura del software Modelo Vista Controlador (MVC) aunque de manera peculiar. Está muy orientado a programar gestores de contenidos CMS personalizados.

Inicialmente fue creado para gestionar páginas de noticias de la World Company de Lawrence, Kansas y fue liberada bajo licencia BSD en el 2005. Hoy en día andamos por la versión 1.1.

Entre las múltiples ventajas que presenta Django, podemos destacar las siguientes:

- Un mapeador objeto-relacional: Puedes definir tus modelos de datos integramente en Python. Estos modelos son mapeados a la BBDD obteniendo inmediatamente un acceso dinámico a la BBDD a través de su API. Aun así, puedes seguir escribiendo sentencias SQL si es necesario.
- Una API de base de datos robusta.
- Un sistema extensible de plantillas basado en etiquetas, con herencia de plantillas.
- Una de las máximas de Django es reutilizar y aprovechar código ya escrito.
- Diseño elegante de URL. Puedes diseñar las URL de la manera más flexible.
- Aplicaciones "enchufables" que pueden instalarse en cualquier página gestionada con Django. Cualquier aplicación python tiene cavida dentro del framework Django.
- Un sistema incorporado de "vistas genéricas" que ahorra tener que escribir la lógica de ciertas tareas comunes.
- Un sistema "middleware" para desarrollar características adicionales; por ejemplo, la distribución principal de Django incluye componentes middleware que proporcionan cacheo, compresión de la salida, normalización de URLs, protección CSRF y soporte de sesiones.
- Soporte de internacionalización, incluyendo traducciones incorporadas de la interfaz de administración.
- Documentación incorporada accesible a través de la aplicación administrativa (incluyendo documentación generada automáticamente de los modelos y las librerías de plantillas añadidas por las aplicaciones).

Como hemos indicado, Django se ciñe a la filosofía de arquitectura MVC como otros muchos frameworks (Struts, Ruby on rails, ...) de una manera peculiar:

- Vista: la capa de presentación se basa en plantillas HTML. Django presenta un template engine y un template loader muy potente que permite presentar al usuario diversas páginas HTML usando una base como plantilla. Esto es posible

porque en cada una de las plantillas se pueden introducir determinadas etiquetas Django que el template loader se encargará de interpretar.

- Controlador: Es lo que en Django se llama views. Puede llevar a la confusión, aunque Django lo llame views, éstas son las que actuan como controlador. Escritas en puro código Python cada view atenderá una petición HTML según el mapeo de URL del que ya se hablará más adelante.

- Modelo: Una de las partes más potentes de Django, su modelo de datos. Cada uno de los modelos creados se mapean en diferentes tablas en la BBDD. Esto permite aislar la BBDD del código y olvidarte de los diferentes select y updates a veces tan tediosos.

Por ello decimos que presenta alguna peculiaridad dentro de la arquitectura MVC.

Nuestro primer proyecto Django

A continuación se pretende describir como empezar con un proyecto Django. Se irán citando paso a paso todo lo necesario para arrancar tu primer entorno Django.

Requisitos:

- Instalar python
- Descargar la última versión de Django

No es un requisito pero facilitará mucho más las cosas que vuestro sistema operativo sea GNU/Linux. Lo recomendamos plenamente. Dada nuestra filosofía nuestra guía se construirá sobre un sistema GNU/Linux basado en debian.

Instalación de python.
El código de Django está 100% escrito en puro Python así lo primero que necesitaremos es instalar python. Python se encuentra en los paquetes de cualquier sistema operativo basado en GNU/Linux.
Por ejemplo en debian:
como root:

```
root@McM:~$ apt-get install python
```

Actualmente versión 2.6. Para comprobar que la instalación de python se realizó con éxito, basta entrar en la consola de python:

```
root@McM:~$ python
Python 2.6.2 (release26-maint, Apr 19 2009, 01:56:41)
[GCC 4.3.3] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

Vemos que la versión instalada es la 2.6.2

Insatalación del framework Django
En la página oficial de Django, podreis encontrar la última versión estable:

```
root@McM:~$ wget http://www.djangoproject.com/download/1.1/tarball/
root@McM:~$ tar xzvf Django-\*.tar.gz
```

Realmente no es necesario mucho más, ni siquiera instalarlo en el sistema. Simplemente se puede añadir a la variable de sistema PYTHONPATH para que incluya así las librerias django:

```
PYTHONPATH=/path_a_la_aplicacion_django/:/path_al_codigo_fuente_django/
```

y también incluir:
DJANGO_SETTINGS_MODULE=<nombre_de_mi_proyecto>.settings

Donde path_al_codigo_fuente_django es el path de instalación, donde hemos descomprimido Django y <nombre_de_mi_proyecto>.settings es donde se encuentra el settings.py con todos los parámetros de confguración del framework django.

```
root@McM:/tmp/Django-1.1$ ls
AUTHORS  docs     extras  LICENSE     PKG-INFO  scripts   setup.py
django   examples INSTALL  MANIFEST.in README    setup.cfg tests
```

Si se desea, también se puede instalar para que estas variables aparezcan en el sistema:

```
root@McM:/tmp/Django-1.1$ python setup.py install
```

Después de esto, podreis ver como django-admin.py se ha instalado en /usr/bin/:

```
root@McM:/tmp/Django-1.1# ls /usr/bin/django-admin.py
/usr/bin/django-admin.py
```

Esto no es estrictamente necesario ya que bastaría en un principio con meter en el sistema las variables antes mencionadas:

DJANGO_SETTINGS_MODULE
PYTHONPATH

Así por ejemplo si queremos usar las librerías de Django por consola:

```
root@McM:/tmp/Django-1.1$ echo $PYTHONPATH

root@McM:/tmp/Django-1.1$ export PYTHONPATH=/tmp/Django-1.1
root@McM:/tmp/Django-1.1$ python
Python 2.6.2 (release26-maint, Apr 19 2009, 01:56:41)
[GCC 4.3.3] on linux2
Type "help", "copyright", "credits" or "license" for more information.
```

```
>>> import django
>>> django.VERSION
(1, 1, 0, 'final', 0)
```

Esto es el mismo efecto que instalarlo en el sistema. Recordad que son simples librerías de python por lo que usando la variable de entorno PYTHONPATH debería ser suficiente.

Estructura de un proyecto Django

Si hemos instalado el framework Django, tendremos en el path django-admin.py (basta poner en consola django-admin.py) pero como hemos indicado no es estrictamente necesario hacer la instalacion:

```
mcm@McM:/tmp$ export PYTHONPATH=/tmp/Django-1.1
```

Creamos un proyecto nuevo llamado prueba:

```
mcm@McM:/tmp$ Django-1.1/django/bin/django-admin.py startproject prueba
```

Esto habrá creado un directorio nuevo en nuestro sistema de archivos llamado prueba que contiene los siguientes ficheros:

```
mcm@McM:/tmp$ ls -lrta prueba/
total 28
-rw-r--r--  1 mcm  mcm    542 2009-09-12 16:49 urls.py
-rw-r--r--  1 mcm  mcm   2773 2009-09-12 16:49 settings.py
-rwxr-xr-x  1 mcm  mcm    546 2009-09-12 16:49 manage.py
-rw-r--r--  1 mcm  mcm      0 2009-09-12 16:49 __init__.py
drwxrwxrwt 19 root root 12288 2009-09-12 16:49 ..
drwxr-xr-x  2 mcm  mcm   4096 2009-09-12 16:49 .
```

- __init__.py: Es un archivo vacio. Indica a Python que ese directorio debe ser considerado com oun paquete Python (una aplicación)
- manage.py: Es una utilidad con la cual mediante linea de comandos podremos interactuar con el proyecto Django de diferentes maneras.
- settings.py: Aquí se configura el proyecto Django, BBDD, aplicaciones, etc ...
- urls.py: contiene las URLs que son mapeadas y posteriormente despachadas mediante vistas genericas o personalizadas. Es como una tabla de contenidos, si la URL no se encuentra en este archivo no puede ser servida.

manage.py

Llegados a este punto ya podemos arrancar nuestra simple aplicación. Para ello utilizaremos el manage.py antes mencionado. Por defecto el framework Django arranca en localhost puerto 8000

```
mcm@McM:/tmp/prueba$ python manage.py runserver
Validating models...
0 errors found

Django version 1.1, using settings 'prueba.settings'
Development server is running at http://127.0.0.1:8000/
Quit the server with CONTROL-C.


mcm@McM:~/tmp$ netstat -an | grep 8000
tcp        0      0 127.0.0.1:8000          0.0.0.0:*               LISTEN
```

Con el navegador podremos acceder a nuestra primera aplicación Django: http://localhost:8000/. En el navegador se mostrará la página inicial indicando que Django está funcionando:

"It worked!
Congratulations on your first Django-powered page."

Podemos arrancar el proyecto Django en cualquier otra interfaz y cualquier otro puerto de la siguiente manera:

python manage.py runserver 8080 --> puerto 8080

python manage.py runserver 0.0.0.0:8000 --> todas las interfaces, puerto 8000

settings.py
==========
Como hemos indicado en este archivo se configurará el proyecto Django. Llegados a este punto, es necesario tener la BBDD instalada y configurada. Para ellos seguir el punto 2 del artículo anterior. Será necesario también instalar los bindings oportunos.

Visto nuestra gran orientación al software libre, nosotros eligiremos MySQL como BBDD. Como se indicó en el manual anterior, instalaremos:

```
root@McM:~$ apt-get install mysql-server
root@McM:~$ apt-get install python-mysqldb
```

Crear una nueva BBDD y un usuario nuevo:

```
mcm@McM:~$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 321
Server version: 5.0.67-0ubuntu6-log (Ubuntu)

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.
```

```
mysql> create database linvix;
Query OK, 1 row affected (0.00 sec)

mysql> create user linvix_user identified by 'passwd';
Query OK, 0 rows affected (0.00 sec)

mysql> GRANT ALL ON linvix.* TO 'linvix_user'@'localhost' IDENTIFIED BY 'passwd';
Query OK, 0 rows affected (0.00 sec)

mysql> flush privileges;
Query OK, 0 rows affected (0.00 sec)
```

Ahora ya podemos configurar el settings.py y poder sincronizar nuestra base de datos con la aplicación:

```
# Django settings for prueba project.

DEBUG = True
TEMPLATE_DEBUG = DEBUG

ADMINS = (
# ('mcm', 'miguelcm@gmail.com'),
)

MANAGERS = ADMINS

DATABASE_ENGINE = 'mysql'    # 'postgresql_psycopg2', 'postgresql', 'mysql', 'sqlite3' or 'oracle'.
DATABASE_NAME = 'linvix'             # Or path to database file if using sqlite3.
DATABASE_USER = 'linvix_user'          # Not used with sqlite3.
DATABASE_PASSWORD = 'passwd'     # Not used with sqlite3.
DATABASE_HOST = ''                 # Set to empty string for localhost. Not used with sqlite3.
DATABASE_PORT = ''                 # Set to empty string for default. Not used with sqlite3.
```

A la hora de sincronizar, Django crea sus tablas por defecto que son las de Sesión, usuarios, grupo, permisos, message y site. También te pide que introduzcas un usuario administrador de la aplicación. Recordad que Django ofrece una consola muy completa de administración directa de todos los modelos de datos y usuarios, entre ellos el administrador (esto se va almacenando en la tabla auth_user)

```
mcm@McM:/tmp/prueba$ python manage.py syncdb
from sets import ImmutableSet
Creating table auth_permission
```

```
Creating table auth_group
Creating table auth_user
Creating table auth_message
Creating table django_content_type
Creating table django_session
Creating table django_site

You just installed Django's auth system, which means you don't have any superusers defined.
Would you like to create one now? (yes/no): yes
Username (Leave blank to use 'mcm'): mcm
E-mail address: mcm@noway.es
Password:
Password (again):
Superuser created successfully.
Installing index for auth.Permission model
Installing index for auth.Message model
```

Lo mejor en cualquier projecto Django es que uses la tabla de usuarios y el modelo User que ya facilita Django. Esto hará todo más automático. Podemos mirar los campos de auth_user:

```
mysql> desc auth_user;
+--------------+--------------+------+-----+---------+----------------+
| Field        | Type         | Null | Key | Default | Extra          |
+--------------+--------------+------+-----+---------+----------------+
| id           | int(11)      | NO   | PRI | NULL    | auto_increment |
| username     | varchar(30)  | NO   | UNI | NULL    |                |
| first_name   | varchar(30)  | NO   |     | NULL    |                |
| last_name    | varchar(30)  | NO   |     | NULL    |                |
| email        | varchar(75)  | NO   |     | NULL    |                |
| password     | varchar(128) | NO   |     | NULL    |                |
| is_staff     | tinyint(1)   | NO   |     | NULL    |                |
| is_active    | tinyint(1)   | NO   |     | NULL    |                |
| is_superuser | tinyint(1)   | NO   |     | NULL    |                |
| last_login   | datetime     | NO   |     | NULL    |                |
| date_joined  | datetime     | NO   |     | NULL    |                |
+--------------+--------------+------+-----+---------+----------------+
11 rows in set (0.00 sec)
```

Los modelos Django siempre funcionan de la misma manera. El primer campo suele ser el id que es siempre un campo que se autoincrementa. En los mismos modelos de Python serás tu quien establezcas las relaciones entre modelos (campos many to many, etc ...)

Con la BBDD y el framework funcionando ya se puede empezar a crear modelos, vistas y plantillas.

En el archivo settings.py también podemos ver las aplicaciones instaladas (INSTALLED_APPS). Por defecto y en relación también a las tablas creadas que

acabamos de ver se crean por defecto las siguientes aplicaciones:

- django.contrib.auth -- Sistema de autenticación
- django.contrib.contenttypes -- framework para los content types.
- django.contrib.sessions -- framework para controlar las sesiones.
- django.contrib.sites -- framework para gestionar varios sites en una misma aplicación Django.

Como hemos dicho estas aplicaciones son instaladas por defecto y es conveniente su uso para gestion de usuarios, autenticación, sesión, etc ...

Creando la primera aplicación

Ya hemos creado el primer proyecto Django, pasemos ahora a crear la primera aplicación. Cada aplicación Django consiste en un paquete Python que, por consiguiente, deberá estar en el path de tu sistema. De igual manera que hemos creado el proyecto, Django ofrece una utilidad para crear la estructura de una aplicación:

(en el directorio del proyecto prueba)

```
mcm@McM:/tmp/prueba$ python manage.py startapp polls
mcm@McM:/tmp/prueba$ tree polls/
polls/
|-- __init__.py
|-- models.py
|-- tests.py
`-- views.py
```

Definiremos ahora nuestro primer modelo. La manera de actuar es la siguiente:

- Creamos modelo de datos
- Sincronizamos nuestra aplicación Django con la BBDD (syncdb)

```
mcm@McM:/tmp/prueba$ vim polls/models.py

from django.db import models

class Poll(models.Model):
question = models.CharField(max_length=200)
pub_date = models.DateTimeField('date published')

class Choice(models.Model):
poll = models.ForeignKey(Poll)
choice = models.CharField(max_length=200)
votes = models.IntegerField()
```

Se recomienda mirar la documentación más detenidamente para ver los tipos de variables que se pueden definir en un modelo (charField, IntegerField, DateTimeField, ...).

Podeis observar como hemos introducido aquí la primera relación entre modelos (recordad modelo = tabla) poll = models.ForeignKey(Poll). De esta manera relacionamos el modelo Choice con el modelo Poll. Todo modelo Choice tendra un campo Poll que corresponde con una entrada en la tabla del modelo Poll.

Esta nueva aplicación, polls, tendrá que ser añadida en el settings.py para que el framework Django la encuentre:

```
INSTALLED_APPS = (
'django.contrib.auth',
'django.contrib.contenttypes',
'django.contrib.sessions',
'django.contrib.sites',
'mysite.polls'
)
```

Otra de las utilidades de manage.py es poder ver exactamente la sentencia que Django lanza internamente al crear las tablas en la BBDD. Esto resultará muy util cuando tengais que modificar modelos y alterar tablas...

Ojo: si no añadimos la aplicación al settings.py, obtendremos el siguiente error:

```
mcm@McM:/tmp/prueba$ python manage.py sql polls
Error: App with label polls could not be found. Are you sure your INSTALLED_APPS setting is correct?
```

Si todo va bien,

```
mcm@McM:/tmp/prueba$ python manage.py sql polls

from sets import ImmutableSet
BEGIN;
CREATE TABLE `polls_poll` (
`id` integer AUTO_INCREMENT NOT NULL PRIMARY KEY,
`question` varchar(200) NOT NULL,
`pub_date` datetime NOT NULL
)
;
CREATE TABLE `polls_choice` (
`id` integer AUTO_INCREMENT NOT NULL PRIMARY KEY,
`poll_id` integer NOT NULL,
`choice` varchar(200) NOT NULL,
`votes` integer NOT NULL
)
;
ALTER TABLE `polls_choice` ADD CONSTRAINT `poll_id_refs_id_5d896c23` FOREIGN KEY (`poll_id`) REFERENCES `polls_poll` (`id`);
COMMIT;
```

Otras utilidades del manage.py son las siguientes:

- python manage.py validate -- comprueba errores en tu modelo de datos
- python manage.py sqlcustom polls -- saca por pantalla cualquier sentencia sql de tus modelos.
- python manage.py sqlclear polls -- saca por pantalla las sentencias DROP TABLE para esta aplicación teneiendo en cuenta las dependencias añadidas
- python manage.py sqlindexes polls -- saca por pantalla las sentencias CREATE INDEX para esta aplicación.
- python manage.py sqlall polls -- combinación de todos los comandos sql.

```
mcm@McM:/tmp/prueba$ python manage.py syncdb

Creating table django_admin_log
Creating table polls_poll
Creating table polls_choice
Installing index for admin.LogEntry model
Installing index for polls.Choice model
```

Ahora si miramos la BBDD, observaremos estas dos nuevas tablas creadas.

```
mysql> show tables;
+---------------------------+
| Tables_in_linvix          |
+---------------------------+
| auth_group                |
| auth_group_permissions    |
| auth_message              |
| auth_permission           |
| auth_user                 |
| auth_user_groups          |
| auth_user_user_permissions |
| django_admin_log          |
| django_content_type       |
| django_session            |
| django_site               |
| polls_choice              |
| polls_poll                |
+-------------------------
```

Ahora podemos añadir métodos para cada uno de esos modelos. Al ser métodos y no nuevos parámetros no hará falta modificar la BBDD (alter) ni sincronizar la aplicación nuevamente.

Continuará en el próximo número...

Miguel
miguelcm@gmail.com
España

Viñales

Gran Caverna Santo Tomas

LINVIX

Participa en la revista, envienos tus artículos, y opiniones ...

linvix@gmail.com

Ubuntu Karmic Koala