# TUXINFO 70

www.tuxinfo.com.ar

- Fortifica tu Wordpress con Latch
- -TUXMOVIL Lanzamiento de nuevos Nexus 2014
- -Systemd es un demonio de administración de sistema
- -Opinión ¿La mejor distro de Linux del año?
- Conociendo el interprete de comando Bash

#### **Nexus Player**



- ¿Como conociste a GNU/Linux? #1
- Aprendiendo Laravel Primera Parte

### Ubuntu 10 años!





Esta revista se publica bajo una licencia de Creative Commons CC BY-SA 3.0 tuxinfo 🙋 tuxinfo

Octubre 2014

Esta revista se publica bajo una licencia de Creative Commons CC BY-SA 3.0 distribuir, Puedes copiar, mostrar públicamente su contenido y hacer obras derivadas, siempre y cuando a) reconozcas los créditos de la obra y b) la compartas bajo la misma licencia.

Microsoft, Apple, Sun, Oracle, así como otras marcas comerciales mencionadas en esta revista son propiedad de sus respectivas empresas.

Dirección Ariel M. Corgatelli

Corrección Ariel M. Corgatelli

Diseño de tapa Ariel M. Corgatelli

Diseño Ariel M. Corgatelli

www http://www.tuxinfo.com.ar

facebook http://www.facebook.com/tuxinfo

email info@tuxinfo.com.ar

twitter @tuxinfo

#### Editorial

Un mes muy importante tenemos los usuarios de Software Libre y Linux. Ya que se cumplen los 10 años de lanzamiento de Ubuntu/Linux. La cual marco sin lugar a dudas un antes y un después en esta distribución. Principalmente por toda la campaña de marketing que llevo adelante su creador desde su lanzamiento y se extendió por unos años (enviando sin costo alguno Cds de Ubuntu a todo el mundo). Además tenemos nuevos dispositivos Nexus en el mercado mundial: un Nexus 6 como smartphone y un Nexus 9 como tableta. Ambos dejaron de ser esos equipos a buen precio con características altamente interesantes; para ser equipos destinados a la gama premium. Obviamente esta decisión de parte de Google tuvo varias controversias en los medios especializados.

Y como era de esperarse entre nosotros hay una nueva versión de Android; la misma se llama Lollipop, siendo la que corresponde a su numero 5, o Android L tal cual se había denominado en sus comienzos.

Este fue un mes muy interesante, con mucho movimiento y muy buenas noticias. De nuestra parte hemos escrito varias notas que seguramente serán de su interés.

No quiero dejar pasar la oportunidad para invitar a quien desee colaborar con artículos en la revista; ya que estamos necesitando cubrir algunos huecos, y nos vendría muy bien sumar colaboradores. Los interesados pueden escribir a nuestra cuenta info@tuxinfo.com.ar. Muchas

## EDITORIAL

gracias de ante mano.

Y para cerrar, un breve resumen de las notas que encontraran en el número 70 de Tuxinfo: Systemd enemigo; Informe Chakra Linux; Aprendiendo Laravel – Primera Parte; ¿Que es Xanadu GNU/Linux?; ¿Como conociste a GNU/Linux? #1 y varias más.

#### Fan page:

https://www.facebook.com/tuxinfo User Twitter: @tuxinfo Mail de contacto: info@tuxinfo.com.ar

¡Sigan pasando la voz! Hay otro nuevo número de TuxInfo para descargar de forma gratuita.

Ariel M. Corgatelli @arielmcorg



IMPORTANTE - Tuxinfo es una revista en la cual participan colaboradores de varios países; por lo tanto puede que noten determinados modismos a la hora de realizar la lectura de algún articulo. Y con el fin de acelerar el proceso de publicación, nos vemos obligados a no realizar la unificación completa del contenido. Espero sepan entender este cambio.

2

## #70



orge Cacho



Linda Martinez



Alfieri Morilla



Hernan Saltiel



Fabio Baccaglion



Naudio Villarroel Urquiola



Natanae Andres Garrido



Jeidi Brito

- 3 Editorial
- 4 Noticias del mes
- 9 Fortifica tu Wordpress con Latch
- 12 Software Freedom Day Barquisimeto
- 13 Software Freedom Day El Tigrito
- 14 TUXMOVIL Lanzamiento de nuevos Nexus 2014
- 18 Systemd es un demonio de administración de sistema
- 19 Opinión ¿La mejor distro de Linux del año?
- 20 Conociendo el interprete de comando Bash
- 23 ¿Como conociste a GNU/Linux? #1
- 25 ¿Que es Xanadu GNU/Linux?
- 27 Aprendiendo Laravel Primera Parte
- 37 Informe de Chakra, distribución LinuxRootear



## ÍNDICE



Fuente: Linware http://www.linware.com.ar

#### #Ubuntu – 10 años de libertad

Sinceramente no se me ocurrió mejor titulo que este; ya que de alguna manera tengo que reconocer que para mi principalmente Ubuntu refleja eso mismo; 10 años de libertad en materia informática.

Pero vayamos en principio a la noticia; y la misma habla que un 20 de octubre de 2004, el empresario sudafricano llamado Mark Shuttleworth; lanzaba desde su empresa Canonical la distribución Linux 4.10 Warty Warthog. Desde ese mismo momento quienes estábamos trabajando con el software libre, y que además teníamos un pensamiento muy abierto, veíamos una gran oportunidad para hacer que Linux entre en millones de hogares de todo el mundo.



Revista Linux Users -07 – Primer medio Argentino en papel que escribió sobre Ubuntu

#### Y porque digo esto?

Muy simple, ya que de la misma manera que Mark lanzaba la distribución, iniciaba una excelente campaña en donde con tan solo completar un formulario desde la web, solicitando tantas copias de Ubuntu, las mismas eran enviadas de forma completamente gratuita a nuestra casa. Y así fue realmente que Ubuntu comenzaba a cobrar forma.

## Noticias del mes

Los envíos constaban de dos formatos por arquitectura, ya sea que optáramos por 32 o 64 bits. El sobre tenia un liveCD y un CD instalable. El Live nos permitía utilizar desde la lectora de nuestra PC toda la distribución sin realizar ningún tipo de instalación, y que cuando reiniciábamos el equipo todo estaba como antes; y por el otro lado el cd install se podía justamente instalar en el equipo. Gracias a ello muchas personas comenzaron a probar Linux y seguramente muchas nunca mas abandonaron la plataforma.

#### Ubuntu un antes y un después en Linux

Si bien Ubuntu tuvo muchas personas que se enmarcaban encontrar de la distribución; muchas con amplio pensamiento vieron la oportunidad de hacer un buen marketing del software libre. Claro que Ubuntu tenia ciertas cosas que a muchos usuarios de la comunidad no les gustaba, mucha desconfianza en su fundador, la forma en que Ubuntu realizaba las configuraciones, la simplicidad y un largo etc.

Lo cierto es que hoy después de 10 años vemos que aunque haya muchas criticas es sin duda la distribución que mas ayudo a la difusión de Linux.

#### Mi experiencia con Linux

Hoy después de utilizar hace mas de 12 años de forma constante Linux en varios sabores; mirando hacia tras ve un largo camino recorrido de la mano de la libertad. Mis comienzos con RedHat, Conectiva, Mandrake, Conectiva, OpenSuse, y por ultimo Ubuntu hasta la actualidad; son sin dudas una buena señal de que el sistema operativo avanza de forma constante. Sin importar banderas, países, distribuciones o el motivo que fuera.

En toda mi vida profesional tuve el placer de trabajar con grandes de la informática; escribir en Users, luego en Linux Users; brindar cursos; recorrer rutas dando charlas; brindar seminarios; llevar adelante Tuxinfo; y este mismo año escribir mi primer libro basado en Ubuntu Linux. Cosas que sin duda me llenan mucho.



Cerrando - Tengo que re afirmar que Ubuntu, fue un antes y un después en lo que a Linux se refiere. Sin el apoyo de esta distribución las cosas no habrían llegado tan lejos.

Desde nuestro lado, Feliz 10 años!!! y por muchas décadas más!.

#### Linux esta en más lugares de los que crees

Según un estudio de Net Applications, Linux esta presente en el 1 o 2% de los equipos de escritorio. Pero claro que las cosas no son tal cual parecen, de hecho si miramos al rededor de nuestros equipos de uso diario podríamos ver que Linux esta mas presente de lo que pensamos.

Y el motivo es simple; al poseer un sistema completamente embebido y moldeable, amparado en el software libre, o bien Open Source. Se hace muy simple la implementación en cualquier lugar incluso en televisores, smartphones, routers, y muchos mas lugares.

Para ello los remitimos al sitio fuente Genbeta; quienes publicaron un informe con casos de ejemplo. Donde podrán ver estamos de alguna manera de forma directa o indirecta usando Linux en algún dispositivo de forma diaria.

Nuestra opinión es que tampoco es tan importante sumar fuertes cifras en lo que respecta a escritorios con Linux; ya que hoy día todos los usuarios trabajamos desde servicios en la nube. Que a su vez estan corriendo en gran medida sobre software libre. Con lo cual estamos usando tecnología Linux de forma diaria.

Los invitamos a que investiguen mas, en las fuentes de los gadgets que utilizan, ya que se llevaran muchas sorpresas.

#### Nuevo kernel Linux publicado, versión 3.17

Así lo anunciaba su creador Linus Torvalds, desde la lista. El cual además de forma inmediata estaba listo para la descarga y utilización por cualquier distribución GNU/Linux a lo largo de todo el mundo. En el mismo no vamos a encontrar grandes cambios o novedades; quizás una de las más importantes es el soporte para el hardware ARM, mejor soporte a controladores de NVIDIA y AMD. Tal cual mismísimo Torvalds, explica en su post, no hay grandes cambios y además explica que muy pronto puede que este la próxima versión 3.18; haciendo mención por ultimo en la mejora de controladores para i915, nouveau, Ethernet, SCSI, sonido, además de algunas correcciones en lo referido a redes.

#### Fuente: Lista Torvalds

http://lkml.iu.edu/hypermail/linux/kernel/1410.0/0 2818.html

#### Dos nuevas versiones Enterprise Linux

En esta oportunidad les contamos que hay dos nuevas versiones de distribuciones Linux Enterprise. Por un lado CentOS 5.11; y por el otro Oracle Linux 5.11.

Ambas versiones pueden ser utilizadas para cometidos corporativos; en la primera opción contaremos con una gran comunidad desde CentOS, y el apoyo reciente desde RedHat hacia esta distro.

oracle-linux

Y por el otro lado si necesitamos una distribución que sea 100×100 compatible con la base de datos Oracle, nada mejor que su misma distro, Oracle Linux 5.11.

Las dos opciones cuentan con el soporte a las arquitecturas convencionales Intel de 64Bits, ambas versiones integran las mas nuevos paquetes instalados, obviamente solucionado el bug de bash y los escritorios gráficos GNOME.

Más información de CentOS 5.11 http://lists.centos.org/pipermail/centosannounce/2014-September/020601.html Manual de CentOS 5.11 http://wiki.centos.org/Manuals/ReleaseNotes/Cent OS5.11 Descargas de CentOS 5.11 http://isoredirect.centos.org/centos/5/isos/

Más información de Oracle Linux 5.11 https://blogs.oracle.com/linux/entry/oracle\_linux\_ 5\_11\_is Manual de Oracle Linux 5.11 https://wikis.oracle.com/display/oraclelinux/Down loading+Oracle+Linux Descargas de Oracle Linux 5.11 http://ftp5.gwdg.de/pub/linux/oracle/EL5/U11/x86 \_64/Enterprise-R5-U11-Server-x86\_64-dvd.iso

#### Zimbra fue nombrado como uno de los finalistas para Next Generation Award experiencia del cliente

En una nota publicada en el blog oficial de Zimbra; se hace mención a el concurso en cual la empresa esta compitiendo; no desde el punto de vista propio; sino desde la visión del cliente.

http://blog.zimbra.com/blog/archives/2014/09/zim bra-customers-named-finalists-next-generationcustomer-experience-award.html

Y para ello su CEO Patrick Brandt, expreso: "nos sentimos orgullosos de forma excepcional cuando nuestros clientes tienen éxito, y sentimos el privilegio de contribuir a sus esfuerzos. Este reconocimiento de Dell y Rackspace por Constellation Investigación, dice mucho acerca de cómo las marcas líderes son aquellos que actúan como agentes de cambio para el beneficio de sus clientes ".

Las empresas Dell y Rackspace personifican a una "marca líder" con sus comunidades externas. Por su parte Dell cuenta con la solución Toad World, para sus soluciones de bases de datos Toad, y Rackspace está impulsando la educación de jóvenes con un programa llamado Open Cloud Academy. Dos grandes proyectos los cuales fueron logradas con sus comunidades.

Para quienes lo deseen, la votación está abierta por una semana más. Y es por eso que Zimbra invita a visitar los proyectos y a partir de ahí votar.

El cierre de la nota es: "Gracias por ayudarnos a honrar a estas marcas únicas y líderes".

## #Zimbra migra sus foros de comunidad a un nuevo lugar

Esta semana la gente de Zimbra, anuncio de forma oficial en su blog la novedad sobre el moviendo de su foro para la comunidad. Desde el comunicado podemos leer lo siguiente: "nuestro foro de Zimbra vBulletin (zimbra.com/forums) fue cerrado recientemente. Nos disculpamos enormemente con nuestros miembros por este tiempo de inactividad. Sabemos que es muy importante para nuestra comunidad de desarrollo de código abierto estos foros, y que son un recurso valioso para el asesoramiento y la información".

"Para resolver el problema tan pronto como nos sea posible, nos estamos moviendo de Vbulletin a la principal comunidad de Zimbra en community.zimbra.com. El proceso nos llevará alrededor de una semana. Mientras tanto, hemos pasado todos los perfiles de usuario de foro a nuestra principal comunidad y les hemos invitado a un espacio temporal donde los usuarios pueden abrir nuevas discusiones acerca de nuestra Colaboración Zimbra y productos de código abierto. Una vez que todos los datos y los debates se migran al nuevo espacio, vamos a pasar a estos miembros y los nuevos debates en las secciones de foro apropiadas".

Excelente noticia y apuesta desde Zimbra para toda la comunidad Open Source. Estaremos atentos a cualquier nuevo avance; mientras tanto pueden obtener mas información desde el blog oficial.

#### Ultimo momento:

La próxima semana, nuestro equipo de gestión de la comunidad será responder preguntas y ayudar a orientar estos nuevos usuarios en el nuevo espacio. Lunes, 22 de septiembre, vamos a abrir los nuevos foros con todos los datos antiguos del foro y los debates en su lugar.

Fuente: Zimbra http://blog.zimbra.com/blog/archives/2014/09/zim bra-forums-migrating-new-home.html

#### **Disponible Red Hat Enterprise Linux 5.11**

Red Hat, Inc. el proveedor mundial de soluciones de código abierto, anunció la disponibilidad de Red Hat Enterprise Linux 5.11, la versión menor definitiva de la madura plataforma Red Hat Enterprise Linux 5. Red Hat Enterprise Linux 5.11 reitera el compromiso de Red Hat de ofrecer un ciclo de vida de 10 años de los productos para todas las versiones principales de Red Hat Enterprise Linux, y ofrece una plataforma segura, estable y confiable para las aplicaciones empresariales esenciales.

La nueva versión definitiva de Red Hat Enterprise Linux presenta mejoras en cuanto a seguridad y rendimiento, y una nueva versión de OpenSCAP.

Además de las mejoras en seguridad y estabilidad, Red Hat Enterprise Linux 5.11 añade actualizaciones a la gestión de suscripciones y las capacidades de depuración, y más aún:

Nuevos controladores de almacenamiento:
 actualizaciones que brindan a los clientes los
 beneficios de algunos de los últimos adaptadores
 de almacenamiento de los socios de hardware de
 Red Hat.

– Mejoras en las capacidades de depuración de Red Hat Access Support: Red Hat Access Support facilita que los clientes gestionen, diagnostiquen y se pongan en contacto directamente con Red Hat a través de una consola dentro de Red Hat Enterprise Linux 5. - Mejoras para los huéspedes de Red Hat Enterprise Linux que operen en VMWare ESXi. Más información de los servicios de gestión de suscripciones que hace más fácil la gestión de suscripciones tanto a través de la interfaz gráfica del usuario (GUI) como de la línea de comando. – Actualización de OpenSCAP a la versión 1.0.8, certificada por el Protocolo de Automatización de Contenido de Seguridad (SCAP) 1.2 del Instituto Nacional de Estándares y Tecnología (NIST), en la categoría Analizador de Configuración Autenticado con la opción Vulnerabilidades y Exposiciones Comunes (CVE). - Esta nueva certificación añade a Red Hat a una pequeña lista de proveedores calificados para el complejo estándar SCAP, que convierte a OpenSCAP y a Red Hat en factores importantes en el ámbito de la seguridad. Con esta nueva certificación, openscap 1.0.8 puede ejecutar contenido SCAP escrito correctamente según el estándar 1.2 proveniente de cualquier fuente [confirmar que se pueda ejecutar el contenido desde todas las fuentes actuales], tal como el contenido del – Lenguaje Abierto de Vulnerabilidad y Evaluación (OVAL) del Equipo de Respuesta de Seguridad de Red Hat, las Guías de Implementación Técnica de

Seguridad (STIG) de DISA, el repositorio OVAL de la Base de Datos Nacional de Vulnerabilidad (NVD), entre otros.

#### Disponibilidad

Red Hat Enterprise Linux 5.11 ya está disponible en el Portal del Cliente de Red Hat para todos los clientes y socios que tengan una suscripción a Red Hat Enterprise Linux activa. Red Hat Enterprise Linux 5 recibirá soporte activo hasta el 31 de marzo de 2017, fecha en la que está previsto retirarlo del mercado.

Jim Totton, vicepresidente y gerente general, Unidad de Negocio Plataformas, Red Hat "El compromiso de Red Hat con un ciclo de vida de productos prolongado y estable es un beneficio clave para aquellos clientes empresariales que dependen de Red Hat Enterprise Linux para sus aplicaciones esenciales. Si bien Red Hat Enterprise Linux 5.11 es una versión menor definitiva de la plataforma Red Hat Enterprise Linux 5, las mejoras que presenta en términos de seguridad y confiabilidad están pensadas para mantener la viabilidad de la plataforma en los años venideros."



Somos una empresa líder en soluciones OpenSource y contamos con más de 5 años de experiencia instalando servidores de colaboración Zimbra.





#### EN CUALQUIER LUGAR, EN CUALQUIER MÁQUINA

www.linware.com.ar / zimbra@linware.com.ar +54(011)60090219 / +54 (351)5891012 / +56 (2)5952714



## Fortifica tu Wordpress con Latch

POR JORGE CACHO HERNÁNDEZ

Si hay algo que preocupa cada vez más en el mundo informático es la seguridad. Cada vez son más los ataques que sufrimos a diario y que afectan tanto a nuestra identidad digital (cuentas de Facebook, Twitter, Tuenti, ..) como a nuestras comunicaciones (Gmail, ....), nuestros datos sensibles (datos bancarios, números de tarjetas de crédito), o a la seguridad de nuestros blogs o webs (Wordpress, Joomla, Druplal, ...)

Todo aquel que administre una página web o un blog sabe perfectamente lo que son los ataques y los intentos de entrar de diversas formas a nuestros sistemas, sobre todo cuando usamos algunos de los CMS (gestores de contenido) más populares como pueden ser Wordpress o Joomla.

En el artículo de hoy veremos cómo podemos aumentar nuestra seguridad informática usando **Latch** y concretamente explicaremos como implementarlo en nuestras webs basadas en **Wordpress**.

#### ¿Qué es Latch?

Latch, como indican es su propia página web [1], "es un servicio que permite agregar un nivel extra de seguridad a tus servicios online. Para ayudar a evitar un uso no autorizado, Latch proporciona al usuario la capacidad de desconectar sus cuentas con un solo movimiento cuando no las esté utilizando"

Para explicarlo de forma más gráfica Latch es nuestro "pestillo digital". De la misma forma que en nuestras casas podemos cerrar el pestillo de una puerta cuando no queremos que nadie entre aunque tenga la llave, Latch nos permitirá cerrar el acceso online a nuestras cuentas, webs, ... impidiendo que nadie entre a ellas aunque tengan (bien porque lo conozcan o bien porque lo obtengan por fuerza bruta) nuestro usuario y contraseña.

Latch además nos ofrecerá de forma opcional añadir un **segundo factor de autenticación** de forma que para hacer uso de un servicio online además de saber el usuario, la password y estar habilitado el "pestillo" de Latch también tendremos que introducir la OTP (One Time Password) que recibiremos en nuestro móvil.

#### La seguridad en nuestras manos

El "pestillo" que implementa Latch se materializa en una **app para móvil** que será nuestro centro de control desde el que podremos habilitar nuestros accesos online, deshabilitarlos o programar su encendido y apagado de forma horaria (por ejemplo podremos de forma automática apagar todos nuestros accesos online cuando nos vamos a dormir o cuando nos vamos de vacaciones y sabemos que no vamos a hacer uso de ellos)

Latch no sólo nos permitirá desde la app habilitar o deshabilitar servicios online sino que también nos enviará una alerta cada vez que se produzca un intento de acceso a una cuenta que tengamos bloqueada.

Actualmente Latch se ofrece como app para las plataformas Android, iOS, Windows Phone y Firefox OS.

#### ¿Dónde puedo usar Latch?

Los primeros sitios web que ya han implmenteado Latch son: Telefónica, Movistar, Acens, Tuenti, Grupo Cortefiel, Cajamar, la Universidad de La Rioja y la Universidad de Salamanca aunque es previsible que muchas más web las que lo vayan integrando en los próximos meses.

Pero Latch también nos ofrece plugins y una SDK para que nosotros, en cualquier de nuestros blogs, páginas web o aplicaciones web también podamos implementarlo. De esta forma ya hay disponibles plugins para: Windows, Wordpress, Joomla, Redmine, Prestashop, Drupal, OpenVPN, OpenSSH, RoundCube, SquirrelMail, Ubuntu, phpMyAdmin, ... y SDK para los siguientes lenguajes de programación: php, Java, Ruby, Python, C y PowerShell.

#### Manos a la obra

Tal y como hemos comentado al inicio de este artículo vamos a mostrar cómo aumentar la seguridad de nuestro sitio Wordpress gracias a Latch:

#### Paso 1: dar de alta nuestra aplicación en Latch

La primero que tendremos que hacer es ir a la web de Latch y darnos de alta como desarrolladores **[2]**.

Una vez creado nuestro nuevo usuario de Latch automáticamente seremos redirigidos al área de desarrolladores **[3]** donde podremos dar de alta nuestro Wordpress pulsando el botón "Añadir una nueva aplicación".

Tan solo tendremos que dar un nombre para nuestra aplicación y ya se nos mostrarán todos los datos que necesitamos para la configuración de nuestro Wordpress: *"ID de aplicación"* y *"Secreto"*. En esta pantalla como vemos también podremos activar el segundo factor de autenticación (por defecto estará deshabilitado)

#### Paso 2: Instalar el plugin de Latch

Lo siguiente que tendremos que hacer es ir a nuestro Wordpress e instalar el plugin de Latch.

Este paso es tan sencillo como entrar en nuestro panel de control de Wordpress, ir al apartado de Plugins y buscar el **plugin llamado "latch"**. Tras instalarlo y activarlo ya estamos en condiciones de configurarlo.

Para realizar la configuración iremos a la opción *"Ajustes de Latch"* dentro de *"Ajustes"* y copiaremos los datos que antes nos han dado en el área de desarrolladores: el *"ID de aplicación"* y el *"Secreto"*. La URL de la API que también nos pide el plugin es opcional, pero si la queremos escribir, su valor debe ser https://latch.elevenpaths.com

Con estos sencillos pasos ya tenemos casi listo nuestro Wordpress para implementar Latch, pero aún nos queda una pequeña tareas más.

#### Paso 3: Instalar la app móvil e iniciar el pareado

El tercer paso será instalar en nuestro móvil (Android, iPhone, Windows Phone o Firefox) la app de Latch e iniciar al pareado del teléfono con el usuario de Wordpress para el que queremos aumentar la seguridad.

La instalación de la app se hace desde los store oficiales de cada aplicación aunque en la web de Latch [4] tendremos todos los accesos directos por si los necesitamos.

Tras instalar la aplicación tendremos que entrar en ella con el mismo usuario y contraseña que hemos creado como desarrolladores en la web de Latch y ya estaremos en condiciones de iniciar el pareado. Para ello pulsaremos en la app el botón de "Añadir nuevo servicio".

Tras pulsar el botón "Generar nuevo código" nos aparecerá en pantalla un código alfanumérico que tendremos que asociarlo a nuestro usuario de Wordpress (por motivos de seguridad tendremos tan sólo 60 segundos para hacer el pareado antes de que el código caduque)

**NOTA:** En caso de error en este paso asegurate de que tienes instalado el módulo php-curl en tu servidor web, ya

#### que es un requisito para funcione Latch [5]

A partir de este momento desde nuestra app movil ya podremos bloquear manualmente el acceso a la administración de nuestro Wordpress siempre que sepamos que no vamos a usarla, además de poder programar de forma horaria el bloqueo y desbloqueo de este acceso.

Además de poder bloquear y desbloquear accesos individuales también podremos hacer de forma masiva para todos los servicios que tengamos integrados con Latch.

Si hacemos la prueba de bloquear el acceso e intentamos entrar en nuestro Wordpress veremos como nos llega una notificación a nuestro móvil indicando este acceso no autorizado que podremos aceptar o denegar. Time-Password (OTP).

Si hacemos ésto cada vez vez que intentemos entrar en nuestro Wordpress además de pedirnos el usuario y contraseña nos pedirá también un password de un sólo uso que recibiremos en nuestro móvil.

Como habéis podido ver Latch incrementa de forma exponencial la seguridad de nuestro sitio web, en nuestro caso una instalación de Wordpress, gracias al uso del "pestillo digital" y de la One-Time-Password.

#### Enlaces de interés

- [1] https://latch.elevenpaths.com
- [2] https://latch.elevenpaths.com/www/
- [3] https://latch.elevenpaths.com/www/developers/editapplication
- [4] https://latch.elevenpaths.com/www/download.html
- [5] https://github.com/ElevenPaths/latch-plugin-wordpress



Jorge Cacho Hernández

http://about.me/jorge.cacho.h



#RADIOGEEK Podcast diario de Tecnología www.radiogeek.ivoox.com

#### Paso opcional: Habilitar la One-Time-Password

Tal y como hemos visto al final del paso 1 es posible, entro del área de desarrolladores de Latch, habilitar la One-



Cada año, en el mes de septiembre, las comunidades de Software Libre del mundo entero festejan el "Día de la Libertad de Software" o " Software Freedom Day". En este 2014 el evento se realizo en dos diferente ciudades de Venezuela, en la cuidad de Barquisimeto, Estado Lara http://goo.gl/CvahO9 y Tigrito, Estado Anozategui http://goo.gl/Qfxc6L

#### Software Freedom Day Barquisimeto

En conversación que con Samuel Rojas, el nos cuenta que el Software Freedom Day se realizo el Viernes 19-09-2014 en las instalaciones de INCES LARA, Carrera 19 con calle 23 , de la cuidad de Barquisimeto (se realizo ese día ya que por motivos laborales de la sede no se podía el Sábado)

A pesar del cambio de fecha, que lógicamente incidió en la asistencia un total de aproximadamente 45 personas de 110 inscritas, se contó con una serie de charlas y ponencia tales como :

 Conociendo el Software Libre, una introducción a la filosofía del Software Libre dictada por Tamara Lucena
 Ley de Infogobierno, Conociendo dicha ley, alcances y metas , Samuel Rojas
 Como Dockerizar las aplicaciones en una distribución por Eduardo Echeverría

.- Como usar Pendrive como discos de Rescate por Jose Sales.

.- Radios en internet, Conociendo como montar una radio por Augusto Dávila

.- Introducción para programar en Python, por Samuel Rojas.

 Manejador de proyectos con Github, como usar los milestones para manejar un proyecto, por Alexander Salas

El equipo organizador del Software Freedom Day en Barquisimeto los intregaron: Jose Sales, Samuel Rojas del Frente Revolucionario de Tecnologías Libres http://www.frtl.org.ve, Jorge Escalona en representación de INCES Lara.

Algunas fotos del SFD Barquisimeto lo puede ver en el siguiente link: http://srosuna.blogspot.com/

#### Software Freedom Day El Tigrito

El 20 de septiembre del año 2014 se llevo a cabo en las instalaciones del Complejo Cultural Simón Bolívar de la ciudad de El Tigrito – Anzoátegui, la celebración del Software Freedom Day (SFD). Un evento multitudinario y que de talla mundial que se llevo a cabo en varias ciudades de Venezuela.

Al igual que muchos eventos de esta clase, el SFD de el tigrito contó con un equipo de trabajo integrado por los miembros oficiales de la Comunidad Ccubik que superviso constantemente toda la jornada con la finalidad de llevar a cabo un evento de calidad. Igualmente un grupo de ponentes de excelente categoría hicieron gala de sus conocimientos habilidades de comunicación para promover la cultura del software libre a todos los asistentes del evento.

El día para el equipo organizador inicio a las 8am donde en su mayoría llego a la locación del evento para organizar el espacio y los preparativos para recibir a nuestro público un equipo que trabajo de la manera más organizada y eficiente posible, comenzamos a recibir asistentes a las 8:30am registrando su asistencia.

El evento toma protagonismo en la voz de nuestra presentadora la cual da a conocer a nuestro publico cual será el cronograma de la actividad durante el día, un resumen de los organizadores Comunidad Ccubik presentando a los miembros, los cuales fueron recibidos con un aplauso por parte del público y así se da la apertura al ciclo de ponencias del día.

**Software Libre ¿Con qué come eso?;** tema el cual hizo una relación entre el conocer de cocinar y los principios básicos lo cuál lo volvió algo interesante e innovador para el público.

**Canaimitas OP;** tema que dio a conocer las potencialidades reales del proyecto canaima, proyecto educativo de SL impulsado en el país, demostrando así que la mayoría de las personas tiene una visión errática de lo que representa este proyecto en el futuro educativo de nuestra sociedad.

**Introducción a Gimp y Conociendo a Blender;** temas que tuvieron cómo semejanza una pequeña demostración introductoria sobre el potencial que podrías encontrar en tales herramientas de diseño y animación tan populares dentro del SL.

Innovación Libre; tema que demostró el concepto de innovación y como esta intrínsecamente incrustado en la misma filosofía del software libre, que desde su génesis ha sido innovador y ha regalado al mundo una tecnología de calidad.

Desarrollo de Vídeo Juegos; Donde se expuso maravillosamente los principios básicos de los videojuegos y las herramientas, libres por supuesto, con las que se pueden contar al momento de decidirse a realizar un videojuego. Además de dar varios tips para mejorar la realidad actual de mundo de los vídeo juegos.

Así entre rifas y concursos promovidas por los ponentes al terminar su ponencias, saludos recibidos por parte de comunidades patrocinante como *Ubuntu-Ve;* todo el evento se realizo en un ambiente de cordialidad y paz. Compartir el conocimiento y que este sea libre, es una expresión se refleja en la misión de nuestra comunidad.



Naudy Villarroel Urquiola @naudyu





### TUXMOVIL -Lanzamiento de nuevos Nexus 2014

Desde que la gente de Google decidió lanzar su primer versión de sistema operativo Android; es que lanzo su línea insignia Nexus. Y el primero en ver la luz fue el a principios del año 2010, con su Nexus One. El mismo estaba fabricado por HTC, bajo licencia y supervisión de Google. Con lo cual este equipo tuvo la mayor cantidad de actualizaciones libres y puras hasta que el hardware le dio oportunidad.



Por supuesto, desde ese lanzamiento de Android, diferentes fabricantes lanzaron sus modelos para tratar de insertarse en el camino. Y claro, las cosas para los fabricantes no eran tan simples ya que se necesitaba de Google para poder avanzar de versión en versión; y ello llevaba toda una ingeniería complicada en la empresa fabricante. Ojo, no hablo de los equipos Nexus. Con lo cual las actualizaciones a los usuarios estaban limitadas en principio a la liberación de la nueva Rom de Google, después a la voluntad del fabricante, seguido del operador con el que el cliente trabaja. Es decir muchos pasos para que el usuario final tenga la última versión que por lo general solución errores de software, problemas con el hardware y mucho más.

Durante años los fabricantes se escudaron con que había equipos de media gama el cual cada año se debían cambiar; personalmente estuve delante de varios directivos de varias empresas; quienes siempre me decían lo mismo: "Ariel, el valor es bajo y es un equipo de entrada, te duró un año, cámbialo" (no me pidan que revele los nombre de las empresas por favor).

Y con esto al usuario le complicó bastante la vida, ya que en muchas oportunidades pasaba que no tenían más actualizaciones de determinadas aplicaciones y así los usuarios se veían obligados a cambiar el equipo.

#### Mi opinión al respecto del soporte

Los equipos deberían de mantener mínimamente dos años de soporte, como para que el usuario amortice la compra, y si cada año debemos cambiarlo no creo que sea rentable. Además, con la economía de toda América Latina, siendo esta no muy buena, inclusive con los problemas que tenemos los usuarios argentinos con la "fabricación" Nacional y los problemas de exportación. Debemos sin lugar a dudas hacer rendir al equipo lo máximo posible. Y de esa manera no sirve cambiar el equipo una vez al año; obviamente ¿quién no quisiera hacerlo? pero el presupuesto no creo que sea para todos.

Entonces cuál es la ventaja de los equipos Nexus La principal tal cual comentaba al inicio de la nota es que permite actualizar mínimamente por el termino de 2 años; y de esta manera sin pasar por el fabricante, tampoco del operador; los equipos se actualizan solos sin importar tampoco en que lugar del mundo estemos.

Por lo tanto la mejor opción si querés usar un equipo Android sería optar por las lineas Nexus; ya sea en smartphone o tabletas. Siendo además equipos completamente puros, donde no hay software de terceros que lo hacen lentos o consumistas en batería.

#### Qué hay de nuevo en los Nexus lanzados

Google realizo el lanzamiento de su nueva propuesta para smartphone y tabletas. Además sumó el lanzamiento de Android 5 L, cuyo nombre es Lollipop; y por último modernizó Android TV.

Antes de avanzar con las novedades me gustaría plasmar mi opinión en relación a el lanzamiento que se realizo este miércoles 15 de octubre. En principio si bien había muchas filtraciones; las cuales se cumplieron al pie de la letra; el problema creo que pasa en los valores. Donde las cosas no son como al comienzo de la gama Nexus; ya que la misma siempre tuvo la premisa de contar con hardware aceptable y potente a un valor competitivo. Hoy se demostró la rotura de esta última frase; ya que los valores se dispararon incluso por arriba de equipos de alta gama. Algo que Nexus nunca tuvo en su esencia. Haciendo que un Nexus 6, de 6 pulgadas (muy grande para el mercado que apunta) tenga un valor muy elevado: incluso la tableta Nexus 9. Obviamente estamos hablando de equipos premium, y no de los que nos estaba acostumbrado a presentar Google.

Pero bueno, acá estamos, y por supuesto con mucha expectativa a probarlos, cuando llegue al mercado argentino. Y paso seguido vamos con un resumen de los lanzamientos realizados.

## La tableta Nexus 9 (8.9 pulgadas) de la mano de HTC

Como les habíamos adelantado el fabricante que esta vez tomó Google para su tableta fue HTC; quien desarrolló un excelente producto.



Sus características técnicas:

Pantalla 8,9 pulgadas IPS QXGA (2048×1536) Procesador 64-bit NVIDIA Tegra K1 a 2.3 GHz RAM 2GB Sistema operativo Android 5.0 Lollipop Almacenamiento 32 ó 64GB Cámara trasera 8MP OIS Cámara frontal 3MP Altavoces Frontales Batería 6700mAh Dimensiones 153.68mm x 228.25mm x 7.95mm Peso 425g

#### Valor y disponibilidad

La tableta estará disponible a partir del próximo 17 de octubre en preventa, con lo cual la disponibilidad del mismo a los consumidores será a partir del 3 de noviembre. Sobre su valor sabemos que será de 399 dólares para la versión de 16GB y de 499 dólares para la versión de 32GB y LTE.

## El nuevo SmartPhone Nexus 6 de la mano de Motorola

Uno de los Smartphone más esperados cada año por los usuarios afines al sistema operativo Android. El motivo es muy simple, ya que cuenta con valor razonable y características destacadas; con su sistema operativo completamente libre de cualquier capa del fabricante. Y que además puede ser actualizado de forma constante gracias a las actualizaciones vía OTA desde Android directo, sin pasar por ningún proveedor de servicio.



Sus características técnicas: Pantalla AMOLED de 5,96 pulgadas Resolución QHD de 1440×2560 píxeles (493 ppi) Microprocesador Qualcomm Snapdragon 805 CPU 4 núcleos a 2,7 GHz GPU Adreno 420 Almacenamiento 32 ó 64 GB Sistema operativo Android 5.0 Lollipop Cámaras Trasera 13 megapíxeles con flash de anillo LED Cámara delantera 2 megapíxeles Conectividad LTE, Wi-Fi, NFC, Bluetooth, GPS Batería 3.220 mAh con carga rápida

Una de las características más importantes y que los usuarios van a destacar es su sistema Turbo Charger, el cual promete, con una carga de tan sólo 15 minutos, una autonomía de 7 horas.

#### Valor y disponibilidad

El mismo estará disponible para reserva en Google Play a partir de finales de este mismo mes. En tiendas, hará su aparición a partir del 12 de noviembre. El valor estará dado desde 649 dólares para la versión de 32 GB. Un valor un tanto elevado si tenemos en cuenta la línea que llevaba hasta ahora. Por supuesto estamos hablando de un terminal premium por sus características técnicas





#### Google anuncio el Nexus Player de la mano de ASUS

Siendo este un dispositivo redondo y totalmente portable el cual puede ser mantenido en una sola mano. Y como se imaginan esta diseñado para poder emitir contenido multimedia hacia una TV con puerto HDMI. Y a no olvidar que será el primer equipo en correr Android TV.

tuxmóvil 20

Las aplicaciones con las cual nos encontraremos son: Neflix, TuneIn, HULU Plus, YouTube, TED, Crackle, VEVO, Pandora, Songza, Pluto.TV, Google Play Music, Google Play Games, iHeart Radio, entre otros.

#### Sus características técnicas:

Control remoto Activado por voz y un gamepad CPU Intel Atom de cuatro núcleos a 1,8 GHz GPU PowerVR serie 6 Memoria 1GB de RAM Capacidad interna 8GB Salida HDMI Conectividad Wi-Fi ac MIMO y Bluetooth

#### nexus player servicios

Una interesante opción para quienes deseen hacer uso de un completo centro multimedia en el living del hogar.



www.google.com/nexus





tuxmóvil



## **Systemd** es un demonio de administración de sistema

Por Alfieri Morillo

#### Systemd enemigo???

**systemd** es un demonio de administración de sistema diseñado exclusivamente para la API del núcleo Linux. systemd fue desarrollado para reemplazar el sistema de inicio (init) heredado de los sistemas operativos System V y Berkeley Software Distribution (BSD). En el proceso de arranque en Linux, es el primer proceso que se ejecuta en el espacio de usuario, por lo tanto, también es el proceso padre de todos los procesos hijos en el espacio de usuario.

Desde un tiempo para acá ha dado que hablar el demonio Systemd, y no precisamente por sus avances si no todo lo contrario, esto ha llevado a la creación de uselessd, un nuevo demonio init basado en systemd que intenta ser el mejor y destronarlo, hacer un mejor trabajo, "innecesaria cantidad de funcionalidades que éste incorpora". Es interesante la elección del nombre que, en inglés, puede leerse como "el inservible systemd" o "usemos menos systemd".

En las primeras etapas de desarrollo, Uselessd no es más que un fork de systemd reducido a su mínima expresión. En palabras de sus desarrolladores, es "un demonio init (initd) básico, un supervisor de procesos y un sistema de dependencias transaccional, que minimiza la intrusión y el aislacionismo". Entre las funcionalidades que fueron eliminadas se encuentran: journald, libudev, udevd y ciertos tipos de unidades consideradas superfluas, a saber, dispositivos, temporizadores, swaps, mounts y automounts.

Como si esto fuera poco, ya han añadido soporte para compilarlo bajo MUSL y uClibc, como alternativas al uso de glibc. Porque ya esta siendo reemplazada por el nuevo proyecto llamado Uselessd.

También está en procesos de ser portado a FreeBSD, mientras que systemd sólo tiene

compatibilidad para Linux.

No se encuentra claro cómo van a terminar las "guerras de init" y si uselessd terminará generando un reemplazo realmente funcional; pero aquellos que deseen aprender más sobre este proyecto pueden visitar su sitio web oficial.

Acá unas de las características mas resaltantes

#### uselessd

- es ligero. Uno de nuestros objetivos con uselessd era purgar systemd de una base de código en su mayoría inútiles, a favor de un proyecto más pequeño y sencillo. uselessd no depende de libudev, y compila sin journald.
- tiene un objetivo concreto. Inicializar el sistema, supervisar los procesos, y el apagado del sistema – Y la no fluencia característica más indeseable.

- es portátil. Limpiar el desorden que se realice desde el funcionamiento interno SystemD, pero reimplementar características en glibc específica (un proceso que llamamos deGNUification) en formas mayormente libcagnósticos. uselessd compila en glibc, MUSL, uClibc e incluso libc BSD.

- es grokkable. Uno de los mayores problemas que tiene la gente con systemd, ya que es tan "monstruosamente" compleja, aparentemente sólo un equipo altamente cualificado de desarrolladores podría mantenerla. uselessd no se esfuerza para abarcar todos los aspectos principales de su sistema operativo, y creemos que esto lo convierte en un producto de limpieza "bloque de construcción del espacio de usuario."

- tiene un logotipo fresco.

Alfieri Morillo @cyberthrone\_war





## Opinión - ¿La mejor distro de Linux del año?

#### Por Fabio Baccaglioni

Los de Linux Voice lanzan la piedra y tiran la posta de cuales han sido las mejores distros este año, es una interesante mirada a cada pro y contra en varias categorías y con un ganador final.

Obviamente entre linuxeros ya sabemos lo subjetivo que es esto y más aun cómo nos movilizan los fanatismos, pero si bien sigo siendo un Debianita anticuado, uso Ubuntu a diario en mi PC de trabajo y mi notebook personal (combino con Windows 7 porque uno es gamer, claro). Aun así debo lidiar con los bemoles de una distro que es muy buena pero en algunas áreas hace un poquito de agua.

En Linux Voice consideran a Ubuntu 14.04 como la mejor para principiantes, estoy de acuerdo pero para mi es un poco cesgado juzgarla para principiantes solamente, es una distro al fin y al cabo y ni un experto ni un principiante necesitan lidiar con drivers o compilando ridiculeces, Ubuntu ofrece todo lo que un sistema operativo completo debería, aun con sus caprichos como Unity que, al menos, no apesta tanto como en versiones anteriores y con la ventaja de poder instalarle cualquier cosa, Gnome, XFCE, LXDE o KDE y que funcione (uso mucho KDE, por ejemplo).

En el atractivo visual se decantan por Elementary, para mi el mayor bleff de toda esta camada de distros "bonitas", vamos, no quiero un OSX, si fuese así me compraría una Mac, sin embargo hay una buena cantidad de usuarios de esta "pretty-distro" que a mi no me convence para nada. Utiliza Mint Cinnamon como WM y mucho GTK que nunca me gustará (lo siento, prefiero QT).

En el área de repositorios se decantan por Arch aunque es discutible ya que Debian, si habilitamos todos los repositorios posibles, sigue siendo el caballo de batalla sobre el cual se montan todas las demás distribuciones que no se basan en RedHat o SUSE, pero Arch tiene una muy buena selección y los paquetes al menos parecen más refinados que los de Ubuntu/Mint. En el departamento de documentación es Arch quien nuevamente se llevaría el premio según este relevamiento, no lo se, nunca usé sus docs, pero tienen razón que por un lado la de Debian apesta de anticuada y la de Ubuntu de principiante, pero si hay un lugar donde siempre encuentro la ayuda correcta es en los foros de Ubuntu así que ni idea. Subjetivo.

Para seguridad la más llamativa del año fue TAILS usada por Snowden, pensada para ejecutarse desde un pen drive y con todo por default para ser encriptado, por SSL y seguro.

A la hora de hablar de performance la más ridícula gana, Slacko Puppy 5.7.0, tan liviana que también es la más rápida en casi todos los benchmarks, aunque me gustaría ver una competencia más realista realizada por Phoronix y su suite de testeo que esto que analiza Linux Voice.

La elección final que realizan es Arch Linux, una distro que realmente ha crecido mucho en los últimos años, personalmente no siento razón para pasarme de Ubuntu/Kubuntu a ninguna otra porque tengo todo lo que necesito, funciona y lo hace bien, pero a veces me gustaría un trabajo más refinado para hardware no tan moderno y no una distro que gaste el 80% de los recursos en cosas irrelevantes.

Por suerte todas las distros mencionadas nos permiten configurar a nuestro antojo el funcionamiento del hardware, algo que los demás sistemas operativos (con excepción de los BSD, claro) no lo permiten.

Fabio Baccaglioni www.tecnogeek.com @fabiomb





## <u>Conociendo el interprete</u> <u>de comando Bash</u>

Por Ariel Corgatelli

Podemos definir como Bash a un interprete de comandos entre nosotros y nuestra PC.

Siempre que tengamos que realizar cualquier tipo de operación en una consola de texto acudiremos a el para realizarla.

Hay Varios interpretadores de comandos pero este es el mas utilizado.

#### Un Poco de Historia

Por aquellos tiempos en los que Dennis Ritchie y Ken Thompson de AT&T diseñaban UNIX, fue su meta crear un medio para que las personas interactuaran con el sistema de una forma mas amena y estándar.

Los sistemas de entonces venían con intérpretes de comandos, que aceptaban comandos de los usuarios y los interpretaban para que los utilizara la máquina.

Pero Ritchie y Thompson querían más, algo que ofreciera más que los intérpretes de comandos del momento. Es por eso que nació la Bourne shell (Bash), creada por S.R. Bourne. Desde la creación de la shell de Bourne se han desarrollado otras shells, tales como la C shell (csh) y la Korn shell (ksh).

Cuando la Free Software Foundation buscaba una shell sin patentes, los desarrolladores empezaron a trabajar en el lenguaje de la Bourne shell y en características de otras shells del momento.

Por lo que surgió Bash. (Bourne Again Shell)

Seguro que ya vimos en varias oportunidades mensajes de error como el siguiente bash: command not found.

Para lo cual tenemos que obtener muchas veces ayuda para el uso del mismo.

#### Comenzando a Usarlo.

Uso del Man.

El comando Man nos da la opción de tener al alcance de nuestra mano los manuales de todos y

cada uno de los comandos que podemos utilizar en Bash.

La forma de utilizarlo es muy simple.

#### linux@linux: > man bash

O bien podemos guardar en un archivo de texto el manual completo en este caso de bash con el siguiente comando.

**linux@linux: >** man bash | col -b > bash.txt

Con este comando le pedimos que cree un archivo de texto y que lo coloque en donde estamos posados ahora directamente en el */home/linux* 

Con el comando antes descriptos podemos consultar el manual o generar un archivo de texto de forma totalmente simple no solo de bash, también de otros comandos. Mas adelante explicados.

Lo primero en ejecutarse es el logueo, el cual se hace previo una verificación en el sistema. Donde nos pedirá el login y la clave, estos dos archivos lo encontramos ya definidos en: /home/usuario /etc/passwd o /etc/shadow

#### /home/usuario

Este es el creado en forma directa cuando ingresamos un usuario en el sistema. La creación del mismo se realiza de forma automática con el comando # adduser usuario # useradd usuario

#### /etc/passwd o/etc/shadow

Estos archivos de configuración, passwd y shadow son simplemente archivos donde se guarda las claves de los usuarios.

Passwd no es mas utilizado en los sistemas actuales por su baja seguridad que bastaba con

editar este archivo y veremos en forma transparente el contenido del mismo. Shadow cumple la misma función que el anterior pero la diferencia radica en su encriptación. Por lo que no veremos la clave en forma transparente al contrario veremos simbología. La forma de asignarle a un usuario su clave es simple y en todos los casos se utiliza el mismo comando.

# passwd usuario

El sistema nos pedirá que ingresemos la clave y luego que la confirmemos.

#### Sigamos.

Un intérprete interactivo de ingreso se ejecuta después de una entrada correcta al sistema, mediante /bin/login, leyendo el fichero /etc/shadow. El intérprete invocado lee normalmente durante el arranque /etc/profile y su equivalente privado, ~/.bash\_profile Una ves producido el login el interprete de comandos cd etc

bash esta operativo y listo para realizar nuestros pedidos.

Además el sistema bien podría ingresar como xterm o konsole, dentro de un entorno gráfico. Este tipo de invocación del intérprete normalmente copia el entorno padre y luego lee en el fichero ~/.bashrc del usuario las instrucciones de configuración adicionales para el arranque.

A los archivos estándar le añadimos también /etc/bashrc, que es llamado desde el ~/.bashrc del usuario para establecer los valores del sistema de inicialización de intérpretes de no ingreso. Podremos tener mas información ingresando el siguiente comando.

#### #info bash

#### /etc/profile

Ahora vamos a dar un ejemplo de un archivo básico /etc/profile básico. Este archivo comienza estableciendo ciertas funciones auxiliares y algunos parámetros básicos. Especifica varios parámetros del historial de bash y, con lo que a razones de seguridad se refiere, desactiva el almacenamiento de un fichero de historial permanente para root. Además establece un símbolo del sistema por defecto. Entonces invoca a pequeños guiones de propósito único en el directorio /etc/profile.d para proporcionar una mayor inicialización.

Por ejemplo tenemos el archivo *complete.bash*, el cual es donde se guardan la configuraciones generales de nuestro bash.

#### MUY UTIL.

Bash guarda en el directorio de cada usuario un historial de los comandos tipeados. El mismo se llama .bash\_history Siendo un archivo de texto con lo que se podrá editar su contenido y ver su contenido.

Algunos comandos muy útiles.

Varias veces nos encontramos con la necesidad de realizar una operación de permisos, modificación de archivos de sistema o cualquier operación que pueda requerir la participación directa del Root. Para ello tenemos como primera medida ser de nuestro conocimiento la clave del mismo, con lo cual estando logueados como usuarios comunes podemos cambiar y ser root con el simple comando *su*.

linux@linux >\$ su Passwd: \*\*\*\*\*\*\* linux:/home/linux #

Además podemos tener la necesidad de ejecutar algún tipo de aplicación en forma de root, para lo cual bastara con tipear:

linux@linux>\$ su -c mc

Con *su -c* y seguido de la aplicación como en este caso Midnight Commander , entraremos en el como root. (operación aplicable para cualquier comando)

#### ALGUNOS COMANDOS BASICOS

#### \$ uname -a

Con esta orden podremos ver todo lo relacionado con el Kernel del sistema, su versión de Linux,

distribución, etc.

#### \$ su usuario cualquiera

Con este comando podremos cambiar por ejemplo de pepe a root previo a tipear su contraseña de root par poder ingresar al login de root.

#### \$ ls -l

Podremos ver el directorio en el cual estamos parado en el sistema. (siempre que se logee algún usuario estará parado en /home/usuario determinado)

Solo podremos ver su contenido.

Veremos con este comando un completo informe del directorio con su respectivo dueño, permisos, fecha de creación del directorio y por supuesto nombre del directorio.

Forma de reconocer si estamos enfrente a un directorio, archivo, o documento. Es por el color. Siendo:

Azul un directorio. Blanco un archivo. Amarillo dispositivos. Celeste link o enlace.

#### \$ mkdir

Crear un directorio en su /home.

#### \$ rmdir

Borrar un directorio.

#### \$ cd

Nos da la posibilidad de volver siempre a /home/usuario sin importar en que directorio estemos ahora.

Si tipearamos cd / pasaríamos a estar directamente en la barra / (donde parten todos los directorios).

#### \$ touch

Crea archivos vacíos.

#### *\$ cp* origen y destino

Copia un archivo partiendo siempre parado en el

archivo y tipeando totalmente el destino en donde se alojara el archivo.

#### \$ mv

Este comando es utilizado igual que en anterior con la única diferencia que solo con este se podrá mover un archivo de un directorio a otro.

#### \$ rename

Sirve para renombrar un archivo.

#### \$ In

Este comando es solo utilizado para crear un acceso directo.

#### CONCLUSION.

No pretendemos con este articulo dar una clase de programación en Bash solo brindar conocimientos generales de el y explicar algunos comandos muy útiles.

Hay que tener en cuenta la utilización de los manuales man los mismos son una herramienta imprescindibles en el mundo de Linux.

Ariel M. Corgatelli @arielmcorg







Bienvenidos a esta nueva sección, donde vamos a conocer historias sobre diferentes personas y su primer cercanía a GNU/Linux o al software libre en general.

Como es el primer informe, por así decirlo, voy a arrancar yo con mi historia.

La idea de la sección es que sea mas en tono de entrevista si es posible, sino será algo del tipo historia y experiencia.

#### Arrancamos.

Mi acercamiento a GNU/Linux (vamos a decirle así para que nadie se espante), está íntimamente ligado a mi acercamiento con las computadoras y tecnología en si.

Tengo 28 años y nací en la ciudad de Mercedes, provincia de Buenos Aires, a unos 100km de la Capital Federal.

Nos tenemos que remontar al año 1996, o sea que yo tenía 10 años. Mi papá por medio del Banco Provincia, donde trabajaba, accede a un beneficio para comprar una PC en cuotas. Mi hermano mayor, en ese momento tenía 18 años (soy el cuarto de cinco hermanos) y ya trabajaba por lo que el decidió comprar la PC con mucho esfuerzo. Era una PC marca Acer, Pentium 200Mhz MMX, 64MB de RAM y 3 Gb de disco. Obviamente venía con Windows 95.

Imagínense, para un chico de 10 años tener una máquina en su casa, a pesar de poder utilizarla casi nada porque mi hermano nos tenia que brindar permiso, "era una locura!!

El tiempo fue pasando y la máquina empezó a andar lenta y a fallar. Ahí aprendimos a formatear por primera vez. Recuerdo que la instalación de

## <u>Como conociste a GNU/Linux? #1</u>

#### Por Natanael Andres Garrido

Windows tardaba como unas 4 horas, mas o menos y dejar de nuevo la PC a punto era cuestión de días. Si no me equivoco, por año la formateábamos unas 5 veces. Era tanta la emoción por querer aprender que mi hermano empezó a comprar las famosas revistas "PC Users" donde traían cds con programas freeware, shareware(se acuerdan de esa palabra?) y demos de juegos.

Llegando al final del año 1998 y no me pregunten como pero mi hermano había conseguido un CD de Windows 98. Nosotros no tenías internet en casa y el dial-up era carísimo porque la llamada se tenía que hacer a capital, las revistas y algunos libros eran la info mas fuerte que nos llegaba. Obviamente nos rompimos la cabeza para poder instalar Windows 98 y Windows 98SE (Segunda Edición) que llegó mas adelante.

En una de esas tantas publicaciones de la revista PC Users, el cd que traía no era de software en general sino que traía un sistema operativo totalmente desconocido. Era Mandrake Linux, se acuerdan del gran Mandrake? Bueno, si mal no recuerdo era su primera versión que al ser libre la gente de PC Users la habían cargado en un CD y lo regalaban. Como ya teníamos un poco de experiencia en instalar un sistema operativo desde y ya manejábamos el tema de las particiones casi a la perfección (ya teníamos una partición separada para guardar datos, si si, en ese disco de 3GB), decidimos instalar Mandrake borrando completamente a Windows de los circuitos de la PC.

Después de horas y horas de intento Mandrake estaba instalado. Habíamos tocado tanto las particiones que rompimos el árbol de directorios y perdimos todos los datos (docs de word y planillas de excel mas que nada) que teníamos resguardados.

Nos encontramos con un KDE en sus inicios, así como GIMP, OpenOffice (si mal no recuerdo) y algunos más.

Lo habíamos conseguido, pudimos instalar ese extraños sistema operativo que cuando la PC booteaba decía "Linux Mandrake 5.1" en el viejo Lilo.





Lamentablemente no duró mucho, porque nos dimos cuenta que no podíamos hacer casi nada. No teníamos soporte para la impresora, ni para el scanner (venían separados en ese tiempo :P). Apenas teníamos el audio funcionando. Nos dimos cuenta que el sistema pedía actualizarse, obviamente a través de internet para poder obtener todo lo antes mencionado y que el sistema quede OK. Pero no pudimos, tuvimos que sacar Mandrake para volver a Windows y empezar a sufrir otra vez con los formateos (cosa que en Mandrake no se garantizaba pero eso no lo sabíamos).

Pasó el tiempo y el sistema de las ventanas nos mantuvo presos varios años hasta que en el año 2006, mi hermano mayor ya casado aparece en casa con una notebook y Ubuntu 06.06 instalado.

Claro, era la época del ADSL y en casa ya teníamos conexión. Cuando mi hermano me mostró que el sistema se actualizaba y podía hacer todas las cosas igualmente que en el Windows XP que teníamos en la PC de casa, quedé asombrado.

Mi amor por el software libre y GNU/Linux había resurgido. Pero había un pequeño problema: no podía instalar un nuevo sistema operativo en la PC porque iba tener problemas en mi casa, con los demas integrantes de la familia. Una vez mas, el sueño de tener GNU/Linux se había esfumado. Pero el tiempo pasó, mas precisamente pasaron 3 años, yo me había puesto de novio y me había mudado a la ciudad de Lanús a unos 10 minutos de Capital Federal, donde resido actualmente, por cuestiones laborales.

Ya con trabajo y dinero propio pude comprarme una netbook Acer (les suena esa conocida marca?) y gracias a Facebook y mas precisamente a Emanuel "Timbis" Lopez (colaborador de la revista) tuve otra vez contacto con la comunidad y con GNU/Linux. Apenas me enteré que podía instalar Ubuntu 9.10 con Windows en otra partición, no dudé en hacerlo y desde ese día no me pude despegar de la cultura libre. Dejé de usar cracks en Windows e invertía cada vez mas tiempo en conocer otras distros y formas de trabajo con GNU/Linux que por muchos años fue el único Sistema Operativo de cabecera hasta que necesitaba algo del amigo de las ventanas pero siempre fue para cosas simples de compatibilidad.

Bueno, y esa fue y es mi historia con GNU/Linux. Espero que les haya gustado y que me comenten en Twitter o G+ que les pareció y si quieren sugerir a alguien para que cuente su historia hacemos el contacto y listo. Nos leemos en el próximo número!

Natanael Andrés Garrido Twitter: @NatanaelGarrido G+: Natanael Garrido Web: www.neositelinux.com.ar





## ¿Que es Xanadu GNU/Linux?

Por Jeidi Brito

Es un distribución rolling release (No necesita reinstalar cada vez que una nueva versión es liberada, los programas se actualizan en cuanto sale una nueva versión de ellos. De esta forma nuestra distribución siempre estará actualizada) basada en Debian SID garantizando un sistema innovador y 100% compatible con los paquetes de Debian o derivados, desarrollada para ser rápida, ligera y a la vez útil pensando en las necesidades del usuario y ofreciendo herramientas para el usuario avanzado.



#### Rápida y Ligera

Incorpora el entorno gráfico LXDE y el gestor de ventanas Openbox, lo que le permite mantener un consumo reducido cediendo el uso de los recursos a otras herramientas y programas. Haciendo que el entorno de escritorio sea rápido, ligero y eficiente.



#### Útil

Contiene herramientas para análisis forense como Wireshark, un analizador de protocolos de red, y ClamTK, aplicación que permite detectar virus en sistemas Windows. Además cuenta con aplicaciones como Gigolo el cual provee una interfaz sencilla para conectarse a carpetas compartidas en red y Filezilla como cliente FTP.

Incluye algunas herramientas para rescate de sistemas tal como Testdisk, Foremost, Magicrescue y utilidades de análisis forense, tal como Drbl, Autopsy, Sleuthkit.

#### Segura

Para el equipo de Xanadu es importante la seguridad por lo que se incluyen programas como Shorewall, Fail2ban y Psad, que pueden activarse desde el post-instalador con sus configuraciones por defecto. Así como también cuenta con sistemas de detección de intrusos como Aide y para detección de rootkits incluye Rkhunter.

#### Navegación Anónima

También incorpora i2p y TOR para la navegación anónima complementándolos con Polipo para hacer cache y Privoxy que se encarga de filtrar contenido, adicionalmente durante la ejecución de tor Dnsmasq se encarga de torificar las peticiones DNS para asegurar que su navegación se mantenga anónima. Mediante estas herramientas es fácil replicar el modelo cliente-servidor utilizado por la distribución Whonix pero con un consumo reducido de recursos.

#### Ofimática y Multimedia

En este apartado, Xanadu incluye la ultima versión de la suite Libreoffice, VLC como reproductor de vídeo y Lxmusic como reproductor de audio, sin dejar atrás a brasero como programa de grabación de discos.



Mensajería y adicionales

En este apartado se incluye la aplicación Pidgin para mensajería y como gestor de archivos tenemos a Thunar y Pcmanfm.

#### **Requisitos mínimos:**

- 512 Mb RAM (1 Gb RAM recomendado)
- CPU a 800 MHz o superior (1.0 GHz recomendado)
- Tarjeta de vídeo a 800x600, 16 Mb vídeo RAM
- BIOS con capacidad de arranque por USB o CDROM

- 5 Gb de capacidad de disco para instalar

El nombre Xanadu se ha considerado como sinónimo de opulencia y misterio, sobre todo gracias al poema de Samuel Taylor Coleridge; también fue la capital de verano del imperio Mongol de Kublai Jan y además fue el nombre seleccionado por Theodore Holm Nelson para su idea de:

"concebir un documento global y único (docuverse), que cubra todo lo escrito en el mundo, mediante una gran cantidad de ordenadores interconectados, que contenga todo el Conocimiento existente o, mejor dicho, información en forma de hipertexto."

Para descargar este excelente sistema solo deben dirigirse a https://xanadulinux.wordpress.com donde encontrara información adicional y noticias referentes a esta poderosa distribución.

Jeidi Brito @JeidiBrito







## – Aprendiendo Laravel Primera Parte

En esta serie de artículos estudiaremos este nuevo (quizá ya no tan nuevo) framework de desarrollo para PHP que está dando mucho que hablar.

En mi experiencia personal me encontré con un producto de código abierto muy bien pensado, diseñado, e implementado. Creo que es un excelente punto de partida para quien quiera aprender MVC sin morir en engorrosos documentos conceptuales con una curva de aprendizaje demasiado empinada.

Comencemos a programar en pocos minutos, y logremos buenos y mantenibles productos.

#### Instalación

Suelo comenzar mis artículos entregando una profunda clase teórica de lo que haremos en los siguientes pasos, pero en este caso, y considerando que el público quiere ver sangre, y quiere verla pronto, vamos a ir a los papeles, y dejaremos para más adelante conceptos que si los incluyera aquí lograría aburrirlos de entrada. Sólo por hoym dejo el aburrimiento para más adelante.

Ante todo, nosotros estamos acostumbrados a pensar en la instalación de un producto como un proceso unitario, en el cual bajamos un archivo .tar.gz, en el peor de los casos, o aplicamos un "apt-get", "rpm" o como es mi caso, "pacman". Para el caso de Laravel, no vamos a instalar un paquete llamado "Laravel", sino que bajaremos el denominado "Composer". Si nuestra distro no lo incorpora como parte de su mundo, lo podremos bajar de http://getcomposer.org. En caso contrario, Por Hernan Saltiel

un "sudo pacman -S php-composer", resolverá el dilema. Si somos usuarios de Ubuntu, o alguna de sus variantes, tendremos que usar el procedimiento manual.

Composer no es un gestor de paquetes nuevo, sino una herramienta para la gestión de dependencia en PHP. Eso quiere decir que si un desarrollador declara en su código dependencias tales como librerías, Composer se ocupará de bajarlas del sitio correspondiente, o avisarnos que tal o cual no están satisfechas, dejándonos a nosotros la tarea de bajarlas a mano.

Con una herramienta de este estilo, si necesitamos una herramienta de manejo de imágenes, o de web services especiales, puedo declarar tanto dicha herramienta como su versión para que luego sea descargada y ubicada en el sitio adecuado para que mi propia aplicación funcione como es debido. Ya acaba el uso de Pear, por ejemplo.

Tanto si lo bajamos a mano, como si usamos algún gestor de paquetes que posea nuestra distro, debemos recordar que es necesario contar, por lo menos, con PHP instalado y funcionando, desde su línea de comandos. Sin PHP y si nuestro gestor de paquetes no es tan inteligente, no podremos hacer demasiado en este caso.

Para el caso manual, ejecutaremos:

[root@battlelloyd-m hecsa]# curl -sS
http://getcomposer.org/installer | php

Luego de este paso podré decidir si lo deseo activo para todo el sistema operativo, o sólo para mi propio usuario. Si lo deseo para todo el sistema operativo puedo proceder a mover composer.phar (es éste el archivo resultante, un Php Archive, por eso el nombre "phar") a un lugar tal como /usr/local/bin con el nombre "composer":

[root@battlelloyd-m hecsa]# mv
composer.phar /usr/local/bin/composer

Sólo con estos pasos (en el peor de los casos) ya puedo ejecutar "composer", y ver el listado de opciones de que dispongo (nótese por el prompt de la siguiente salida de pantalla que he dejado de ser root para ser un usuario plano):

[root@battlelloyd-m hecsa]\$ composer Warning: This development build of composer is over 30 days old. It is recommended to update it by running "/usr/bin/composer self-update" to get the latest version.



Composer version a8adbfeb9fc7861deade782938222714168a22a8 2014-09-05 16:28:50

Usage: [options] command [arguments]

Options: --help -h Display this help message. --quiet -q Do not output any message. --verbose -v|vv|vvv Increase the verbosity of messages: 1 for normal output, 2 for more verbose output and 3 for debug

(...)

Ya casi tenemos Laravel instalado. Lo que tendremos que hacer ahora es explicarle a Composer que queremos bajar Laravel con todas sus dependencias. Y lo haré creando mi entorno de desarrollo en mi home directory, ya que el mismo Laravel cuenta con un servidor web muy flexible y versátil. Para ello, ejecutaremos:

[hecsa@battlelloyd-m ~]\$ composer create-project laravel/laravel miproyecto (...)

En este caso, si el mismo Composer carece de todos los elementos del sistema operativo para poder finalizar con su implementación, nos lo dirá con claros mensajes en pantalla. Uno de los errores más comunes es bien no tener implementado el paquete "php-mcrypt", bien no haberlo declarado en el archivo "/etc/php/php.ini", por ejemplo.



En el caso de no haber errores de dependencias de paquetes, y teniendo en cuenta que "laravel/laravel" es un identificador, y que "miproyecto" es el nombre que le estaré dando a la implementación, al final de la ejecución de este comando tendremos un entorno de Laravel listo para ser usado dentro del directorio "miproyecto".

Lo podremos verificar ejecutando el comando:

[hecsa@battlelloyd-m ~]\$ cd miproyecto/ [hecsa@battlelloyd-m miproyecto]\$ php artisan serve Laravel development server started on http://localhost:8000

...e ingresando desde un navegador a la página "http://localhost:8000", donde veremos una pantalla muy parecida a la siguiente:



Excelente, ya tenemos nuestro entorno de desarrollo listo. Si queremos salir del Web server que Laravel levantó, sólo tendremos que presionar Ctrl-C. Más adelante veremos y analizaremos a "artisan".

#### ¿Qué es MVC?

Para algunos desarrolladores que se han topado con estas siglas, significará "Me Voy Corriendo". Para nosotros, serán las iniciales de "Model View Controller". Muy lindo, pero ¿qué eso, y qué significa?

MVC es una forma de dividir nuestro código en diferentes responsabilidades. Cuando miramos el mundo actual del desarrollo, encontramos MVC por todas partes, ya sea en Ruby, ASP.NET, por supuesto, PHP.

La idea subyacente de esta técnica es la separación de la lógica de las capas de representación (Vistas, o Views), permitiendo que se escriba código que pueda ser dividido en las siguientes capas:

Model: Los modelos son la forma en la que se interactúa con los datos. Es una capa entre los datos mismos y la aplicación. Esos datos pueden estar en bases de datos, en archivos simples de texto, o planillas de cálculo, por sólo citar algunas fuentes.
Views: Las vistas o views son la representación visual de los datos. Las views nos permiten escribir una presentación adecuada para los datos que estamos manipulando. Sólo sabe cómo recibir datos, y representarlos, no tienen un "cerebro" que los procese.

- Controller: Son un punto de enlace entre el Model y las Views, cuya responsabilidad primaria es la de gestionar los pedidos y enviar datos desde el Model a los Views.

Entonces, encontraremos que los controllers gestionan las validaciones de los pedidos y datos de usuarios y la lógica de negocios, los Models representan los datos, y las Views generan los datos de salida de dichos pedidos.

En el caso particular de Laravel, tendremos también Routes, o rutas, pueden ser importantes cuando tenemos un sitio pequeño o queremos definir páginas estáticas de forma rápida. Las Routes nos permiten evitar tener que escribir Controllers para cada una de estas páginas estáticas, como puede ser el típico "About" que tienen todas las aplicaciones.

#### Primeras codificaciones

Para entender cómo funcionan las rutas, vamos a ingresar al directorio "app" de "miproyecto", donde encontraremos un archivo llamado "routes.php". Editemos ese archivo, y veremos que ya tenemos una ruta armada:

```
Route::get('/', function()
{
         return View::make('hello');
});
```

En este caso, le estamos diciendo a Laravel que cuando se ejecuta un GET de la raíz de nuestro sitio, mostrará lo que exista en "hello". Y ¿qué es "hello"? Bien puede ser una View llamada "hello.php", bien puede ser un archivo especial llamado "blade" llamado "hello.blade.php". Más adelante veremos que son los blades. Olvídense de los héroes vampirescos.

Ahora bien, ingresemos al directorio "views" dentro de "app", y encontraremos el archivo "hello.php", junto a otros directorios. Su contenido es el "You have arrived!" que vimos antes. Entones, ya nos imaginamos que si queremos hacer una página estática denominada "about", podremos declararla en el archivo "routes.php", creando luego una página llamada "about.php" en el subdirectorio "views". Vamos a hacerlo, editando primero el archivo "routes.php", y agregando lo siguiente:

```
Route::get('about', function()
{
```

```
return View::make('about');
```

});

Ahora, en el subdirectorio "views" crearemos un archivo llamado "about.php" con el siguiente contenido:

<?php

echo "<h1>Bienvenido al tutorial de Laravel!</h1>"

?>

Si ahora vamos al URL

http://localhost:8000/about veremos que aparece una página con el contenido recién agregado. Así de fácil es generar una ruta, y crear las páginas necesarias para alimentarla.

Ahora bien, ¿qué es lo que en realidad ocurrió cuando pedimos ese URL al servidor? Se siguió esta secuencia:

- Cuando un usuario envía un pedido de página tal como /index o /about, primero ingresa al archivo .htaccess que se encuentra en la raíz del subdirectorio "public" de la instancia que creamos de Laravel. Ese archivo redirige todos los pedidos de este estilo al archivo "index.php" que también está situado en la raíz de ese subdirectorio.

El archivo index.php carga
"bootstrap/autoload.php", quien a la vez carga los paquetes de Composer incluyendo
"vendor/autoload.php", y el classloader de Laravel.

 Entonces "index.php" carga
 "bootstrap/start.php", con lo que se crea un objeto de aplicación.

 Ahora Laravel carga el entorno y las configuraciones existentes invocando los objetos del tipo "route" que sean necesarios para determinar a qué lugar se deberá reenviar el pedido del usuario.

- Basándonos en la respuesta del router o controller, se crea un objeto y se le envía al usuario.

Como vemos, las operaciones internas pueden ser algo más complejas de lo que imaginábamos.

Bien, hemos creado nuestra primera vista, y ahora queremos que esa vista sirva como un template, cosa para la cual PHP no es el mejor compañero, que digamos. Vamos a agregar, en la vista recién creada, una referencia a una variable en particular. Y para eso, vamos a usar el concepto de blades, que antes dejamos inconcluso.

Blade es un motor de templates incorporado en Laravel tal que buscará los archivos .blade.php para referenciarlos. Y ¿para qué queremos un motor de templates? Para no usar código estático, por un lado, y para evitar ensuciar nuestras vistas definiendo, como se suele hacer en PHP, variables o valores en general dentro de un archivo.

Movamos el archivo "about.php" a "about.blade.php", ejecutemos de nuevo nuestro servidor de desarrollo, y veamos que, en un principio, nada ha cambiado. Lo bueno llega en muy breve.

Vamos a cambiar el contenido de nuestra vista "about.blade.php" de lo que tenía antes por código HTML puro, o casi puro, ya que agregaremos una referencia a una variable entre "{{ }}", como se puede ver en el siguiente listado:

```
<head>
</head>
<body>
<h1>Bienvenido al tutorial de
Laravel, {{ $nombre }}!</h1>
</body>
</html>
```

Luego, cambiemos la ruta antes generada para declarar esta variable en el archivo "routes.php":

```
Route::get('about', function()
{
     $nombre = 'HeCSa';
     return View::make('about')-
>with('nombre', $nombre);
});
```

Notemos dos cosas: por un lado, la declaración de la variable "\$nombre" como siempre lo hemos hecho en PHP, y por otro, el agregado de "->with('nombre', \$nombre", con lo cual establezco a nivel de variables que utilizaré el valor de "\$nombre" cuando sea invocado. Y así es como el mensaje cambiará a "Bienvenido al tutorial de Laravel, HeCSa!".

#### **Control y Kaos**

Respecto de los controllers, nos permitirán agrupar determinadas acciones en clases. Por lo tanto, en un proyecto MVC típico, tendremos un coltroller que será responsable de todas las acciones del usuario como ser su registro, su logueo, edición de perfil, etc. Generalmente las rutas son usadas para pequeñas acciones, o para crear páginas estáticas rápidamente, pero los controllers proveen opciones mucho más profundas para crear un grupo de métodos que pertenezcan a una determinada clase.

<!doctype html> <html> Una desventaja de las rutas es que no se puede compartir código entre rutas, por lo que si necesitáramos una funcionalidad similar en dos rutas, la misma debiera ser reescrita.

Crearemos entonces nuestro primer controller editando el archivo "routes.php" y agregando lo siguiente:

Route::get('contacto','PaginasController@con tacto');

Eso quiere decir que en algún lugar tendremos un nuevo archivo llamado "PaginasController.php", y dentro de él un método llamado "contacto". Como se podrán imaginar, crearemos el archivo "PaginasController.php" dentro del directorio app/controllers de nuestra instalación de desarrollo, agregando el siguiente contenido al mismo:

```
<?php
class PaginasController extends
BaseController {
   public function contacto()
   {
     return View::make('contacto');
   }
}</pre>
```

Nótese que como curiosidad, no se han cerrado los tags de PHP. No es un error. Pero ¿qué es lo que hemos hecho?

Hemos escrito una clase llamada "PaginasController" que extiende a la clase "BaseController" (si lo desean pueden leer el contenido del archivo "BaseController.php", que está dentro del directorio "controllers", tal como el que ahora hemos creado), que es parte de Laravel, y que posee un método llamado "contacto()" el que, a la vez, entregará sólo una vista llamada "contacto". Sí, como se lo están imaginando, el siguiente acto será crear el archivo "contacto.php" dentro del directorio de "views". Agreguémoslo con el siguiente contenido:

```
<!doctype html>
<html>
<head>
</head>
<body>
Bienvenido a la pagina de contacto!
</body>
</html>
```

Levantemos nuestro servidor de desarrollo, y apuntemos nuestro navegador a http://localhost:8000/contacto . ¡Sorpresa!



#### Aves migratorias

En Laravel las estructuras de las bases de datos se declaran para ser luego creadas en una base vacía. También para ser eliminadas, así que a tener cuidado con esto.

Lo cierto es que los desarrolladores suelen usar sistemas de control de versiones para su código, y eso les permite trabajar en grandes equipos sin tener problemas, así como retroceder entre sus versiones cada vez que lo necesitan. Los usos de GIT, Mercurial, o SVN les hacen la vida notablemente más fácil.

Pero esos sistemas de control de versiones, si bien muy buenos y flexibles, no cubren la totalidad de la realidad de un desarrollo. Existen otras áreas de cambio para las que no teníamos, antes, elementos para versionar. Ese es, por ejemplo, el campo de las bases de datos.

Esos esquemas de bases de datos se modifican continuamente, cada vez que lo hacemos con nuestra aplicación, generalmente. Y esos cambios, salvo que exista un procedimiento manual para sincronizarlos con los scripts de instalación de aplicaciones, no se mantienen en los sistemas de control de versiones.

Las "migrations" constituyen las formas que tiene Laravel de ayudar a mantener la evolución de la estructura de nuestras bases de datos que soportan nuestras aplicaciones, sin tener que borrar y recrear dichas bases de datos cada vez que se haga una modificación. Por otro lado, permite hacerlo desde el código PHP, sin necesidad de ejecutar scripts de bases de datos, ni siquiera de usar una línea de SQL. Luego, el definirla a nivel de código, y no de bases de datos, nos permite mantener una buena independencia del tipo de motor de bases que se esté utilizando.

Las "migrations" se aplicarán en Laravel usando la herramienta "artisan", tal como antes lo hemos hecho con otras funciones. Existen sólo unas pocas instrucciones que se le puede dar a "artisan" cuando se desee trabajar con las bases de datos:

 migrate:install: Es la primera ejecución que se hará en Laravel. De esta forma, comenzamos con el esquema de migraciones.

- Migrate: Cada vez que estemos actualizando una base de datos con nuevas estructuras ejecutaremos este comando, de forma tal de migrar lo que hemos agregado en el código para modificar las estructuras de las bases de datos.

migrate:rollback: Si hemos generado errores en nuestro código, podemos volver atrás con este comando. Es de notar que sólo se puede ejecutar este comando del estilo "deshacer" sobre la última migration, y que los datos correspondiente a la columna que se elimina. Por otro lado, si ya teníamos datos en la tabla que se está eliminando, modificando, y demás, los mismos se eliminarán, perdiendo todo.

migrate:reset: Tira atrás todas las migraciones que alguna vez se hayan hecho, por lo que elimina todas las tablas y datos.
migrate:refresh: Si descubriéramos que tenemos que rehacer todo, pero absolutamente todo, dado que por algún motivo nuestra base de datos quedó algo rota, podríamos usar este comando, ya que elimina la base entera, y la recrea aplicando inclusive todas las migrations que hayamos creado.

 migrate:make: Este comando crea el esqueleto de una migración para que luego la modifiquemos a nuestro gusto.

Antes de pensar en generar una migration, debemos revisar la configuración de nuestras conexiones de bases de datos. Eso lo hacemos editando el archivo "app/config/database.php" dentro del directorio "miproyecto". En nuestro caso particular, estaremos usando el motor de "MariaDB" (MySQL para quienes poseen éste en lugar de aquél) en el que tendremos una base de datos llamada "miproyecto" con un usuario dueño llamado "miproyecto" y clave "MiProyecto01", por lo que el archivo mencionado deberá contar con, al menos, las siguientes entradas:

```
return array(
```

"'''''''''''''''''''''''''''''''''''''	=> 'mysql', ons' => array(	
····	nysql' => array( 'driver'	=>
'mysql',	'host'	=>
'localhost',	Idatabasal	
'miproyecto',	ualabase	=>
'miproyecto',	'username'	=>
'MiProvecto01'.	'password'	=>
!utfo!	'charset'	=>
	'collation'	=>
'utf8_unicode_ci',	, 'prefix'	=>
'',		
),		
)		

Lógicamente, uno podría generar una nueva entrada, que en lugar de llamarse "mysql" se llame "miproyecto", colocar los mismos datos de conexión, y en la entrada de "default" colocar "miproyecto".

Recordemos que para poder ejecutar el comando de artisan correspondiente, nuestra configuración de PHP debe contener el módulo "pdo\_mysql.so", eso quiere decir que en el archivo /etc/php/php.ini debe estar la siguiente entrada descomentada:

extension=pdo\_mysql.so

Por otro lado, recordemos que si el usuario "miproyecto" no existe aún en la base de datos, así como si no posee permisos de creación de bases, la ejecución del "migrate:install" generará un error como el siguiente:

```
[PDOException]
```

SQLSTATE[HY000] [1045] Access denied for user 'miproyecto'@'localhost' (usi ng password: YES)

Para nuestro usuarios, previamente, se debe haber ejecutado la siguiente línea desde dentro de nuestro motor MariaDB (si no existe la DB, crearla con la primera línea):

```
MariaDB [(none)]> create database
miproyecto;
```

MariaDB [(none)]> grant all on miproyecto.\* to miproyecto@localhost identified by 'MiProyecto01';

Entonces, ejecutaremos el tan deseado comando de artisan:

[hecsa@sdf1 miproyecto]\$ php artisan migrate:install Migration table created successfully.

Lo interesante es que luego de este comando si ingresamos a nuestra base de datos "miproyecto" veremos que, aunque no hemos creado ninguna tabla aún, tendremos una dedicada a registrar las migraciones:

```
MariaDB [miproyecto]> show tables;
+-----+
```

	Tabl	Les_	_in_n	niproye	ecto	
	migr	rati	Lons			
1	row	in	set	(0.00	sec)	

Ahora sí, crearemos nuestra primera migration propiamente dicha, en la que crearemos una tabla en la que registraremos películas, por lo que tendrá sólo un campo de ID, uno con el nombre de la película, y otra con la fecha de lanzamiento.

Lo primero será crear el archivo de la migration en sí mismo, para lo que ejecutaremos el siguiente comando:

```
[hecsa@sdf1 miproyecto]$ php artisan
migrate:make crear_tabla_pelis --table
pelis --create
Created Migration:
2014_10_12_142938_crear_tabla_pelis
Generating optimized class loader
Compiling common classes
Compiling views
```

El nombre "crear\_tabla\_pelis" lo elegí porque es sencillo recordar qué es lo que esta migration tendrá como finalidad. Podría haber elegido un nombre como por ejemplo "pelis", pero eso no me diría demasiado sobre si tiene por fin crear y destruir esta tabla, cargar datos, descargar datos, o qué.

De acuerdo a la salida del comando anterior, les dejo como tarea el revisar el contenido del archivo generado, que es "app/database/migrations/2014\_10\_12\_1429 38\_crear\_tabla\_pelis.php", y sobre todo, dos métodos, uno que hace referencia a "up" y otro a "down". En el siguiente artículo abordaremos la creación de una aplicación completa, y nos pondremos a modificar este archivo a nuestro gusto.

#### Modelos para armar

Ya hemos hablado de Views, Controllers, y Routes. También hemos creado nuestra primera "migration". Llega el momento de abordar el tema de los Models.

Los models son la forma de interactuar con los datos cuando usamos Laravel, por lo que son una capa (layer) entre los datos y la aplicación. Estos datos pueden estar en bases de datos, o en archivos de planillas de cálculo o XML, por sólo citar algunas fuentes. Por ello, un model conserva en su interior toda la lógica de negocios de una aplicación.

Ahora bien, ¿qué es la lógica de negocios? Es una representación de los objetos del mundo real que estarán presentes en nuestra aplicación. En ella se especifica la forma en la que interactúan unos con otros para lograr un determinado fin. Son las reglas que permiten saber cómo se acceden y actualizan esos objetos.

Para comprender bien cómo funciona un model, comenzaremos viendo qué es Eloquent ORM, uno de los bloques constitutivos fundamentales de la tacnología que da origen al uso de los models. Un ORM es un "Object Relational Mapper", y en particular Eloquent nos permitirá representar a través de modelos las tablas de nuestra base de datos, o fuente de datos en general, las operaciones que en cada una de ellas podremos ejecutar a través de los métodos que hemos diseñado.

En los siguientes artículos abordaremos muy fuertemente la generación de modelos, ya que la idea será comenzar armando una pequeña aplicación que nos permita manipular una lista de películas, agregarle las funciones de alta, baja y modificación, para luego restringirla por medio de un login, y demás bellezas con las que veremos lo sencillo que es trabajar con Laravel y crear una aplicación.

Nos leemos en la próxima entrega, en la que crearemos la aplicación completa.



Hernán "HeCSa" Saltiel AOSUG Leader CaFeLUG Member Twitter: @hcsaltiel hsaltiel@gmail.com http://www.facebook.com/hcsaltiel http://www.aosug.com.ar





El mundo del escritorio Linux está aumentando su personalización y complejidad como nunca antes. Las interfaces de usuario se están volviendo más y más específicas con las distros a medida que pasa el tiempo.

Buscando y buscando nos hemos encontrado con una muy buena distribución que será muy bienvenida sobre todo por los simpatizantes del escritorio KDE.

Afortunadamente, una distro no necesita ser personalizada demasiado para resultar fácil de usar. La respuesta es Chakra Linux, el cual es un "fork" de Arch Linux. Ésta no es la clase de fork con la cual come; un fork es cuando alguien toma el software de alguien más, lo modifica a su gusto, y le da un nuevo nombre. Desde luego, todos sabemos que si alguien hiciera eso con el software de Microsoft o Adobe tendría una increíble demanda, pero aquí afuera en la tierra del software libre esto sucede todo el tiempo, y a nadie le importa

#### **UN POCO DE HISTORIA**

Chakra Project fue iniciada por los desarrolladores de KDEmod, quienes decidieron crear un live CD basado en Arch Linux. El objetivo de Chakra Project era disponer de un sistema operativo de escritorio fácil de usar pero que mantuviera todas las características, claridad, potencia y responsabilidad de Arch Linux, basándose en el principio KISS.

El 30 de agosto de 2010 el equipo Chakra Project lanzó su primera versión independiente, llamada Chakra GNU/Linux 0.2. No era más que un Arch Linux live CD pre-configurado.

Versión	Nombre en clave	Fecha de lanzamiento
Alpha1	Geen	21-12-2008 <sup>7</sup>
Alpha2	Geen	17-02-2009 <sup>8</sup>
Alpha3	Test	30-09-2009 <sup>9</sup>
Alpha4	NewAge	17-11-2009 <sup>10</sup>
Alpha5	Panora	29-03-2010 <sup>11</sup>
0.2	Jaz	30-08-2010 <sup>6</sup>
0.3	Ashoc	2011
0.4	Cyrus	2011
2011.04-r2	Aida	17-07-2011 <sup>12</sup>
2011.09	Edn	29-08-2011 <sup>13</sup>
2012.02	Archimedes	12-02-2012 <sup>14</sup>
2012.08	Claire	24-08-2012 <sup>15</sup>
2013.02	Benz	10-02-2013 <sup>16</sup>
2013.09	Fritz Van Tom 🖉	22-09-2013 <sup>17</sup>
2014.02	Curie	03-02-2014 <sup>18</sup>
2014.05	Descartes	21-05-2014 <sup>19</sup>

(Cronologia historica de las versiones)

## Informe de Chakra, distribución Linux

Por Javier Ibar

Podemos decir que Chakra es una versión bastante joven pero muy prometedora por el desarrollo que tuvo hasta la fecha. Actualmente se encuentra en la versión "Euler".

#### ALGUNAS CARACTERISTICAS DE ESTA INTERESANTE DISTRIBUCIÓN

Nos encontramos con una distro muy poderosa, rápida y amigable originalmente derivada de Arch Linux. La misma viene equipada con un moderno escritorio KDE. Contiene tanto software libre como privativo en sus repositorios, y es capaz de instalar controladores privativos para las tarjetas de vídeo soportadas sin configuración adicional.

Esta distribución es publicada bajo un modelo semi-continuo (half-rolling release), donde, sobre una base estable de paquetes que es probada y no actualizada frecuentemente, se envían al usuario actualizaciones constantes de los programas y juegos. Este sistema permite que Chakra Project sea a la vez estable, y tenga las últimas versiones de la mayoría de los programas, que generalmente están disponibles inmediatamente después de ser publicados, si es que no hay algún problema grave.

La característica fundamental de Chakra Project es su devoción absoluta al entorno de escritorio KDE. En un sistema por omisión, las librerías necesarias para ejecutar programas GNOME o GTK no son siquiera instaladas, y los programas que requieren dichas librerías deben ser instalados mediante el sistema de contenedores (bundles), que instala estas librerías separadas del resto del sistema. Estas librerías pueden, en todo caso, ser instaladas, pero la selección de programas GNOME es mínima. El sistema de bundles fue remplazado por el repositorio extra, debido a que su mantenimiento era muy complejo.

Actualmente los desarrolladores de Chakra Project trabajan en un gestor de paquetes que reemplazará a Pacman, denominado "Akabei" (en honor a uno de los fantasmas que aparecen en el famoso vídeo juego, de origen japonés). Una vez que el gestor de paquetes Akabei esté bien implementado, será lanzada la interfaz gráfica "Shaman". Según los propios desarrolladores de Chakra Project, el gestor de paquetes de Arch, Pacman, no se desarrolló para poder soportar interfaces gráficas, de ahí que se encuentren en estos momentos desarrollando su sucesor, Akabei.

Chakra en realidad comenzó como un KDEMod, una versión más modular del entorno de escritorio de KDE para Arch. KDE viene como un grupo de software bastante grande, incluyendo un montón de cosas que quizas no necesariamente se desee pero que no puede desinstalarse. KDEMod despejó un poco las cosas, liberando sólo el entorno de escritorio con aplicaciones útiles y dejando al usuario instalar lo que desee. Eventualmente, los desarrolladores de KDEMod decidieron separarse de Arch y crear su propia distro, a la cual nombraron como Chakra.

Chakra aún es muy similar a Arch, manteniendo el sistema base igual de simple, despejado, y tan configurable por el usuario como sea posible. Pero principalmente es una distro de escritorio, y en lugar de intentar dar de todo a su público ofreciendo una amplia variedad de escritorios se enfoca exclusivamente en el entorno de escritorio de KDE y sus aplicaciones nativas. El objetivo es proveer la mejor experiencia del entorno de escritorio de KDE sobre una base estable, simple y configurable. Para lograr esto Chakra ha evolucionado a un nuevo sistema de lanzamientos del tipo "semi-rolling" (Liberación media). El núcleo del sistema tiene lanzamientos regulares aproximadamente cada seis meses pero evitando las rígidas fechas que garantizarían un sistema bugeado (con errores). El escritorio y las aplicaciones son actualizadas en la medida en que hay nuevos lanzamientos disponibles. De esta manera, usted obtiene una base sólida y estable, cargada con las últimas aplicaciones.

Un punto interesante de esta distro también es que cuenta con una interesante guía completa en todos los idiomas ("La guía completa para el principiante de Chakra"). Pueden darse una vuelta y verla online:

http://chakraos.org/wiki/index.php?title=Beginner %E2%80%99s\_Guide/es

MANOS A LA OBRA CON LA INSTALACIÓN

Como primera medida nos bajaremos la ISO (Chakra 2014.09 "Euler") del siguiente link:

http://chakraos.org/home/?get

Una vez bajada la imagen podremos optar por quemar la misma en un dvd o virtualizarla con Virtualbox por ejemplo tarea que se llevo a cabo para revisionar esta distro. Aprovechamos para dejarles la web de Virtualbox por si quieren bajar y experimentar esta magnifica aplicación:

#### https://www.virtualbox.org/wiki/Downloads

Al arrancar la instalación y presionando la tecla F2 podremos definir el idioma del instalador:



#### (Selección de idioma)

Presenta un interesante instalador gráfico llamado "Tribe". La instalación de esta distro y particionado de nuestro disco la podremos hacer con esta intuitiva aplicación gráfica. Esta interfaz de instalación es uno de los fuertes que tiene esta distro.



(Instalador Tribe)

El proceso de instalación es muy fácil y no tendremos muchos problemas para completar el mismo. - Como primera medida nos pedirá definir la distribución del teclado



(Distribución del teclado)

Luego seguiremos seleccionando el uso horario



(Uso horario)

El paso siguiente sera la creación de usuarios



(Particionado del disco)

Y llegamos al ultimo paso de la instalación en donde elegiremos el tipo de instalación.

<b>Sienvenido</b> Notas sobre la vers	<ul> <li>Instalación fuera de línea. Instala e</li> <li>Instalación en línea. Descarga e in</li> </ul>	el sistema usando los paquetes que se incluyen en la ISO, opción predeterminada stala un escritorio KDE mínimo y funcional. La rapidez de la instalación depende j	por co
Revisión de las licer	🔿 I <u>n</u> alámbrica	Herramientas para conexiones inalámbricas	
<ul> <li>Teclado</li> <li>Idioma y hora</li> </ul>		Herramientas para el Repositorio Comunitario de Chakra	
Creación de usuaric Preparación del disc		Aplicaciones de trabajo con gráficos	v
<ul> <li>Tipo de instalación</li> <li>Listo para la instala</li> </ul>		Conjunto de paquetes para grabación de discos	V
Instalación del siste		داmprimes mucho?	V
onfiguración Reiniciar y disfrutar		Reproductores de música y video	V
		Herramientas y aplicaciones útiles	v

(Tipo de instalación)

Por un lado podremos elegir una instalación fuera de línea en donde se instala el sistema usando los paquetes que se incluyen en la ISO (opción predeterminada o una instalación en línea en donde se descarga e instala un escritorio KDE mínimo y funcional. En esta ultima opción la rapidez de la instalación dependerá de los paquetes seleccionados.

Nos armaremos de algo de paciencia mientras corre nuestra instalación



(Progreso de la instalación)

Una vez terminada la instalación podremos reiniciar nuestro equipo y lanzarnos a experimentar el mundo Chakra.



(Pantalla de bienvenida)

Luego de reiniciar nuestro equipo podremos tener nuestro primer contacto con Chakra Linux.

#### ¿POR QUÉ CHAKRA?

Existen varias y muy buenas razones para utilizar Chakra, no sólo para usuarios nuevos de Linux sino también para usuarios avanzados. Primero de todo, usted obtiene la estabilidad que provee el núcleo de un sistema base corregido, pero



también tiene las últimas versiones del escritorio y las aplicaciones (aunque estas son probadas una semana o más antes de ser incluidas en los repositorios principales). También hay una devoción en Chakra por la mente en el entorno de escritorio de KDE y sus aplicaciones nativas. KDE es probablemente el entorno de escritorio más fino del mundo, para cualquier plataforma, y Chakra provee la experiencia de escritorio KDE muy suave y veloz.

En pocas palabras, Chakra es más estable que Arch, más actualizado que Slackware, muy veloz, y provee un muy buen escritorio KDE. Tiene un diseño muy simple y limpio.

Javier Ibar

Lic. en Sistemas



javier1056@yahoo.com.ar

https://www.facebook.com/javier.ibar

www.tecnoroot.com.ar



