

TUXINFO

Entorno
de Escritorio
TRINITY



Diseño: Anubis4D
Modelo: María Zasali
Fotos: Matías Irigoien

Slitaz 4.0
Fedora 17
EtherApe I
SolusOS 1.1
Boys say Go!
Raspberry Pi
Toshiba Thrive
Guía GNU/Linux IX
Fuduntu Rolling Release
Command & Conquer
Tiberium alliance

49



Revista Tuxinfo



Esta revista se publica bajo una licencia de **Creative Commons CC BY-SA 3.0**. Puedes copiar, distribuir, 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, edición y coordinación

Ariel M. Corgatelli

Marketing, edición y ventas

Claudia A. Juri

Corrección

Luis Luque

Oscar Reckziegel

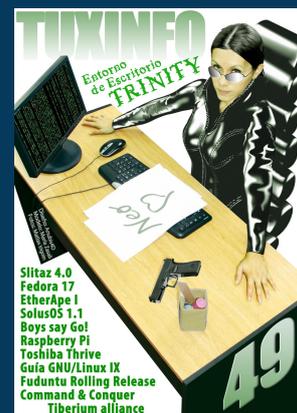
Diseño

Jorge Cacho Hernández

- 03 Editorial
- 05 Toshiba Thrive
- 07 Raspberry Pi: un ordenador por \$35
- 11 Boys say Go!
- 16 EtherApe I
- 18 Fuduntu: distribución Linux Rolling Release
- 20 Fedora 17: Instalación
- 28 Guía de GNU/Linux (IX)
- 31 SolusOS 1.1 (Eveline): una alternativa clásica
- 34 Trinity: entorno de escritorio fork de KDE 3.5
- 35 Command & Conquer Tiberium Alliances

web: <http://www.tuxinfo.com.ar>
facebook: <http://www.facebook.com/tuxinfo>
email: info@tuxinfo.com.ar
twitter: @tuxinfo

Trinity: María Zasali
Fotografía: Matías Irigoin
<http://www.esqueproducciones.blogspot.com.ar>
<http://www.youtube.com/user/irigoin7>
<http://www.youtube.com/loquillostv>
Arte de tapa: Marcos Caballero "Anubis4D"
<http://marquitux.blogspot.com.ar>



Nuevo número de TuxInfo, nuevos temas y muchas notas para compartir con amigos. En este mes ha pasado mucha agua debajo del puente, por un lado tenemos que Linus Torvalds, en una entrevista, se quejó mucho de la gente de NVIDIA. El motivo es más que claro: la poca importancia que le están dando al mundo libre, excepto por sus micros Tegra dispuestos para Android. Otro de los temas importantes, es el avance constante que está teniendo Android en todo el mundo y los futuros lanzamientos de nuevas versiones. Algo que la verdad no termino de entender, ya que la última versión estable (Ice Cream Sandwich 4.0) todavía no está disponible en la gran mayoría de terminales a lo largo de todo el mundo. Aún así, Google ya anuncia su sistema 5, (en desarrollo) y la continuación de Android 4.2. Hay veces que pienso en la gran fragmentación actual de Android y otras veces pienso que no es tanto, lo cierto es que hay muchas versiones de este sistema operativo en muchos smartphones y muchas tablets, que no estarían soportando ICS. Obviamente hay cosas positivas, como el anuncio de Motorola Mobility sobre las actualizaciones de smartphones hacia ICS y a Gingerbread.

Y como último tema relevante a nivel regional, está lo sucedido con la reunión de nuestra Presidente, Cristina Fernández de Kirchner y el segundo al mando de Microsoft mundial. Dicha reunión tuvo sin lugar a dudas un contexto político en donde la mandataria celebró un acuerdo de provisión de nuevas tecnologías para el ámbito académico y estudiantil. De forma automática la comunidad de software libre Argentina salió a repudiar dicho acuerdo, se gestó una gran revolución política y de forma automática muchos intentarán sacar rédito de dicha cuestión.

Muchas veces hablo en los programas de Radiogeek y TuxInfo, sobre la actual comunidad de software libre en nuestro país. Sin lugar a dudas se ha politizado mucho sobre ello, con la clara intención de poder concretar algún tipo de acuerdo y hasta incluso intentar "ganar alguna licitación" con el Estado Nacional. Lamentablemente, luego de un tiempo podemos entrever la relación que hay entre algunos dirigentes políticos y algunos empresarios supuestamente afines al software libre. Sinceramente me siento muy defraudado por ciertas actitudes de ciertos referentes del software libre de mi

país, quienes dejaron de serlo simplemente porque pasaron de ser militantes de dicha acción, para pasar a ser empresarios detrás de negociados políticos.

Igualmente siempre se termina por visualizar cualquier tipo de acción asociada a estas cuestiones, el caso puntual es la carta enviada por el sitio Web "Software Libre con Cristina" [1], la cual tuvo respuesta directa desde el Ministerio de Industria [2]. Y luego de un tiempo, los mismos administradores de esta web, fueron quienes se reunieron con la Ministra de Industria para llevarles una propuesta formal de los cambios necesarios y el apoyo que ellos podrían brindarles al gobierno actual y su política del software público [3].

Es decir una empresa ligada a ciertos funcionarios, armó una web y la dio a conocer de forma automática cuando se produjo la reunión con la máxima Mandataria y Microsoft. La respuesta salió de forma automática desde el mismo Ministerio de Industria, y un mes después los mismos políticos y dueños de la empresa/cooperativa que lanzó dicha carta se reúne con la Ministra. Disculpen si soy medio "perseguido" pero hay algo que no me cierra. Además en ningún lugar de la web se puede divisar quién esta detrás, y tampoco en su nota informa de la reunión, pero sí desde un comunicado oficial de Telam.

Cierro este tema y dejo abierto a cualquier comentario que ustedes quieran enviarnos; ¿no les parece rara esta situación?

Creo que este mes extendí mucho la sección, no era mi interés hablar de política en esta columna, pero en esta ocasión mucho espacio para otro tema no me ha quedado.

Cerrando y como todos los meses, los invito a que lean toda la revista y si al hacerlo es de su agrado, los invitamos a que ¡sigan pasando la voz! Hay otro nuevo número de TuxInfo para descargar de forma gratuita.

[1] <http://www.infosertec.com.ar/blog/?p=37277>

[2] <http://www.infosertec.com.ar/blog/?p=37313>

[3] <http://www.telam.com.ar/nota/29285/>



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



**Carrera Linux
Argentina**
Capacitación a Distancia

Centro de Reservas
informes@carreralinux.com.ar
+54 11 4792 8308

Programación C Expert

- Programación C Expert - Nivel 1
- Programación C Expert - Nivel 2
- Programación C Expert - Nivel 3
- Programación C Expert - Linux Programming

La carrera Programación C Expert tendrá inicio el día Viernes 24 de Agosto, la misma tiene un año de duración. La carrera abarca desde conceptos fundamentales de programación estructurada en lenguaje C, y a medida que se recorren las clases, se van incorporando nuevos temas y tópicos más avanzados.

Linux Certified Security Systems

- Linux Security - Basics
- Linux Security - Network
- Lenguaje C - Conceptos
- Lenguaje C - Linux Programming
- Lenguaje C - Hacking Code

La carrera LCSS inicial el día 1 de Septiembre y tiene una duración de dos años y medio. Esta propuesta, cuenta como siempre con la posibilidad de certificarse con UTN, convenio vigente desde el año 2002 a la fecha.

www.carreralinux.com.ar



Linux
Professional
Institute

UTN
UNIVERSIDAD TECNOLÓGICA NACIONAL

Res. 107/12





Toshiba Thrive

POR ARIEL M. CORGATELLI

La primera impresión que tuvimos fue en relación al tamaño, ya que el mismo, a diferencia de otras tabletas que hemos observado, se excedía en casi 4 centímetros. Por cierto, a medida que íbamos probándola, las cosas fueron cambiando. Las funcionalidades con las que viene equipada son magníficas. No todas las tabletas tienen, por ejemplo, un conector hembra USB en el cual se puede conectar un pendrive cualquiera y además un slot para conectar una tarjeta SD convencional.

Como si fuera poco, el puerto clásico para conectar la tableta a una PC, es mediante una conexión USB clásica (no una microUSB, punto a favor). Y por supuesto, algo que estábamos buscando, era la conexión HDMI, algo que encontramos de forma automática debajo de una tapa que la cubre.

La tableta tiene características bien clásicas: Sistema Operativo Android Honeycomb 3.1, (por ahora no actualizable) wifi, USB 2.0, microprocesador Dual-core de 1GHz, tarjeta de vídeo NVIDIA GeForce, pantalla de alta resolución bajo 1280x800 píxeles, 10.1 pulgadas, un peso de tan solo 700 grs, cámara posterior de 5mpx, cámara frontal 2mpx HD para vídeo llamadas y sus medidas 27.3 cm de largo, 1.57 cm de ancho, 17.73 cm de alto.

Hasta acá vemos las características básicas de una tableta cualquiera, pero lo interesante de este equipo son las prestaciones y las diferentes opciones que trae para interactuar con el usuario.

Como primera medida tenemos que se pueden intercambiar hasta 5 carcasas de diferentes colores, luego todo tipo de conectividad para insertar cualquier

dispositivo, salida HDMI, cuenta con la posibilidad de colocarla en un dock y así poder interactuar directamente con una TV, además posee indicadores frontales de power, de red wifi y de nivel de batería; y por si esto fuera poco, botones físicos de traba en la orientación, sumados a los clásicos de volumen.

Con lo cual el equipo se termina complementando como si fuera un "equipo portátil convencional", al mejor estilo de una netbook (obviamente con un teclado bluetooth).

Vayamos al detalle técnico de su hardware:

- Sistema Operativo: Android™ 3.1, Honeycomb, en Español
- Procesador y gráficos: Gráficos NVIDIA® GeForce®
- Procesador Mobile de 1GHz Dual-Core
- Memoria interna: Memoria interna de 1GB DDR2
- Cámara con micrófono frontal de 2MP HD
- Cámara trasera de 5MP con enfoque automático y zoom





- Resolución: 1280 x 800p de resolución (WXGA)
- Capacitive Multi-touch (supports optional capacitive stylus)
- Amplio ángulo de visión
- Toshiba Adaptive Display Technology
- Toshiba Resolution+ Video Enhancement Technology
- Teclado virtual con Swype ®
- Wi-Fi® Wireless networking (802.11b/g/n)
- Bluetooth® version 3.0 + HS

digital Captura de video de 720p

- Multitarjetas: Full-size lector de tarjetas SD
- Puertos de datos: Full sized puerto HDMI, Full sized puerto USB v2.0; Puerto Mini-USB v2.0, Puerto para conectar a docking
- Unidad de almacenamiento interna: 16GB
- Bocinas (altavoces) estereofónicas
- Conector de auriculares estéreo de 3.5mm
- Batería prismática de Ion Litio
- Dimensión: (L x A x AL): 273mm x 15.8mm x 177mm
- Peso: 0.765 kgs
- Botones Físicos: Encendido; Control de Volumen; Bloqueo de rotación de pantalla; Bloqueo de la cubierta trasera.
- 10.1 Pulgadas Diagonal
- Tipo: LED backlight

Conclusión

La tableta de Toshiba, es una excelente opción para quienes necesitan tener un dispositivo ampliamente competitivo, (referido al costo específicamente) y con la mayor cantidad de puertos de entrada que hemos visto. Y por supuesto excelente potencia y estabilidad.

Puntuación

7/10



Ariel Corgatelli
@arielmorg





Raspberry Pi

Un ordenador por \$35 (*)

POR JORGE CACHO HERNÁNDEZ

A estas alturas casi todos los lectores de Tuxinfo habrán oído hablar de Raspberry Pi, una verdadera revolución dentro del mundo del hardware libre.

Raspberry Pi [1] es un ordenador, así de sencillo, pero unas peculiaridades muy concretas. La primera de ellas es que lo podemos comprar por \$35 (sí, \$35, no me he olvidado de ningún cero). Y por si fuera poco a este hay que añadirle que tiene un tamaño casi similar al de una tarjeta de crédito (85.60mm x 53.98mm x 17mm) y tiene un consumo de tan sólo 3W.

Detrás de este ordenador se encuentra la Fundación Raspberry Pi, una fundación caritativa de Gran Bretaña sin ánimo de lucro.

Raspberry Pi nace con vocación educativa, pero sus posibles usos son tan extensos como los que podamos imaginar: ordenador de sobremesa para usos habituales (procesador de texto, hojas de cálculo, navegación web, ...), centro multimedia (reproduce video en alta definición), servidor web, servidor de nuestra nube libre ownCloud (ver Tuxinfo 48 [2]), ... entre otros muchos posibles usos.

Evidentemente no podemos pensar que Raspberry Pi va a sustituir a nuestro actual PC de escritorio u ordenador portátil, ya que las prestaciones y los recursos hardware que nos ofrece son limitados (los veremos a continuación), no obstante la relación calidad/precio que nos ofrece hacen de Raspberry un producto de primera línea.

Especificaciones

Cuando adquirimos una Raspberry Pi lo que estamos comprando es una placa con todos los conectores integrados y con las siguientes características:

CPU: Broadcom BCM2835 SoC a 700MHz

RAM: 256 MB (compartida con la GPU). No hay posibilidad de ampliación de la RAM que viene de serie.

Almacenamiento: Ranura para inserción de tarjetas SD (tarjeta no incluida). No dispone de disco duro interno. Esta tarjeta SD se usará tanto para el arranque del sistema operativo como para el almacenamiento.

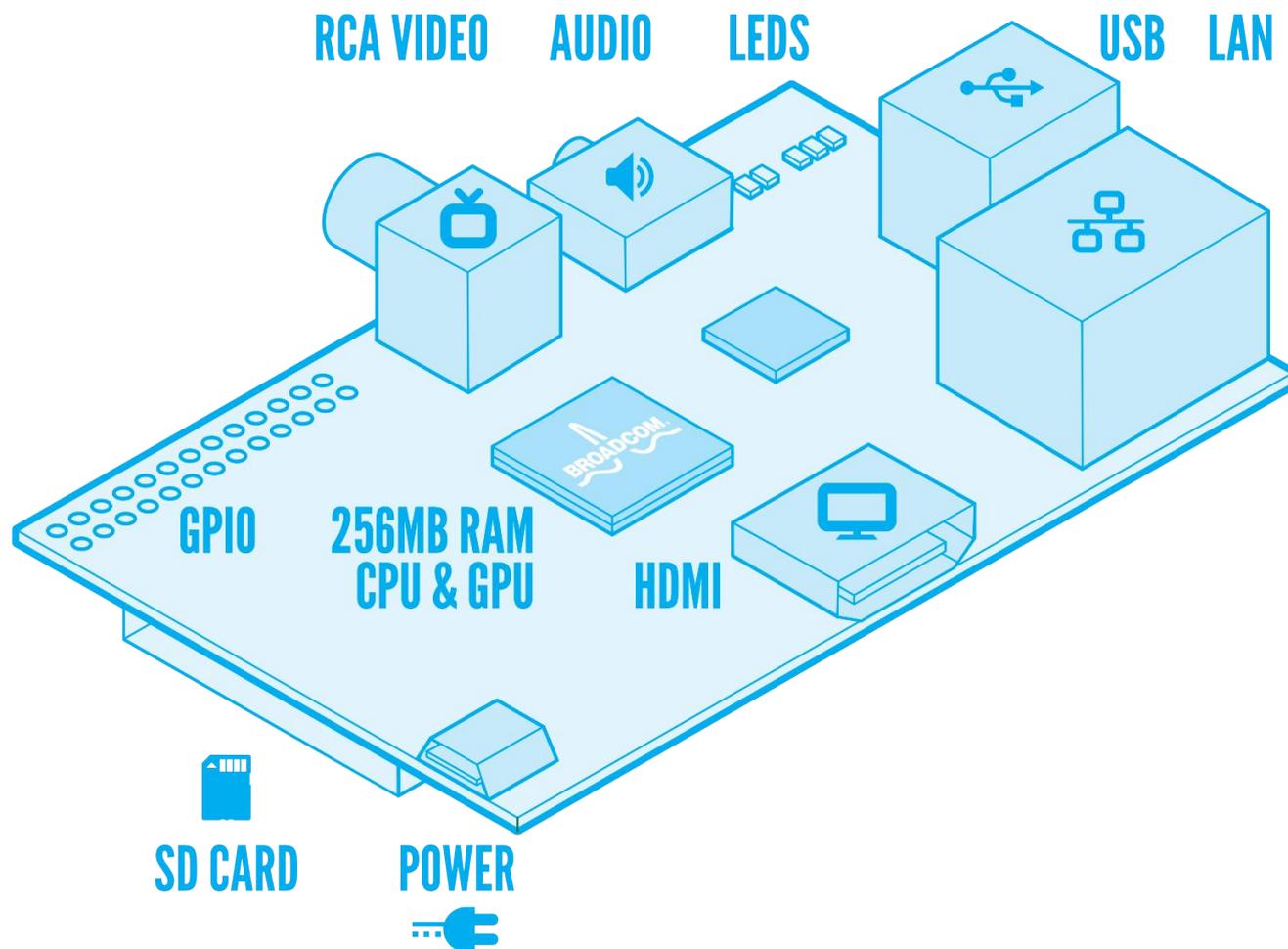
Gráficos: Dual-core VideoCore IV GPU

Salida gráfica: Salida HDMI y salida RCA

Sonido: Salida digital vía HDMI y salida analógica estéreo vía jack de 3.5 mm. Podremos usar los puertos USB disponibles para entradas de micrófonos.

Conectividad: 2 puertos USB 2.0 (sólo 1 en el caso del "Modelo A"), ranura para tarjetas SD, puerto Ethernet (sólo para el "Modelo B") y 26 pines GPIO (para conexiones físicas con otros elementos hardware, como LEDs, tarjetas Arduino, etc ...)

(*) Todos los precios están expresados en dólares USA.



Dimensiones: 85.60mm x 53.98mm x 17mm

Consumo: Variable desde 2W (en espera) hasta 3W (consumo pico)

Peso: 35 gramos

Dos modelos

Tal y como hemos visto existen dos modelos de Raspberry Pi, denominados "Modelo A" y "Modelo B". De momento sólo está a la venta el "Modelo B". En todo caso las únicas diferencias entre ellos son los puertos USB (1 en el caso del "Modelo A" y 2 en el caso del "Modelo B") y la conexión Ethernet (inexistente en el "Modelo A" y presente en el "Modelo B"). Inicialmente el "Modelo A" sólo tenía 156 MB de RAM pero ha sido rediseñado y ya cuenta con los mismos 256 MB que el "Modelo B"

Lo que no tiene

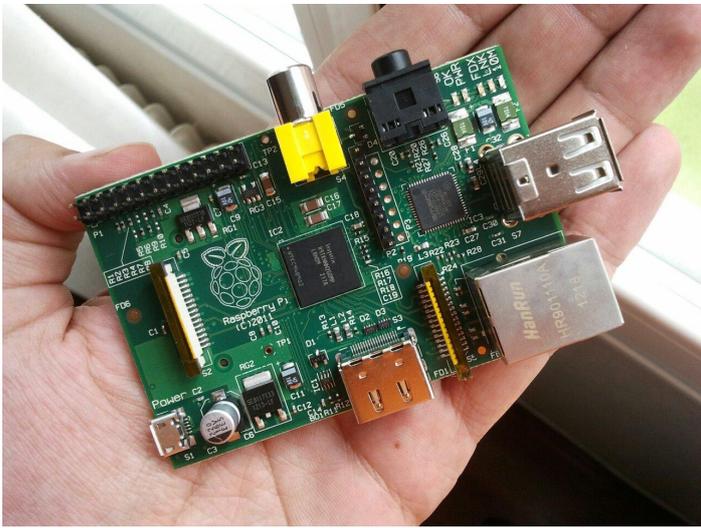
Raspberry Pi tiene, como hemos visto, un precio excepcional de \$35 pero como podemos ver no trae todos los elementos que necesitamos para trabajar con ella.

Para ello necesitaremos adquirir de forma obligatoria u opcionalmente:

Periféricos: no trae ni teclado ni ratón. Dependiendo del uso que vayamos a darle nos harán falta de forma permanente (si vamos a usarla como un PC de escritorio o centro multimedia) o tan solo para la configuración inicial (si vamos a acceder a ella vía ssh)

Tarjeta SD: Será obligatorio añadir a nuestra lista de la compra una tarjeta SD (ya que desde esta tarjeta SD se hará el arranque de la Raspberry Pi). Desde la fundación nos recomiendan comprar una buena tarjeta de clase 4 de al menos 4GB. Se han probado oficialmente tarjetas de hasta 32 GB, en todo caso en la wiki del proyecto tendremos más información sobre otro tipo de tarjetas soportadas. Tener el arranque en la tarjeta SD tiene la ventaja de que su reseteo será tan sencillo como reformatear la SD o incluso podemos tener varias SD configuradas con diferentes sistemas operativos y programas de forma que podamos alternar el uso de la Raspberry Pi con tan solo un cambio de tarjeta.

Pantalla: tal y como hemos visto Raspberry Pi viene con conectores HDMI y VGA por lo que seremos nosotros



quienes tengamos que poner el monitor. Al igual que en el anterior caso de los periféricos este monitor será permanente o temporal en función del uso que vayamos a darle.

Toma de corriente: tampoco viene de serie con la Raspberry Pi por lo que tendremos que adquirirla aparte (nos costará alrededor de \$6). Tendrá que ser una fuente de alimentación que nos dé al menos 700mA a 5V. Sus creadores nos indican que no podremos alimentar la Raspberry Pi desde el puerto USB de otro ordenador, aunque al parecer sí sería posible alimentarla con 4 pilas AA.

Cable Ethernet: Tan solo necesario para el “Modelo B” siempre y cuando necesitemos hacer uso de la red.

Wifi: Raspberry Pi ni dispone de conectividad Wifi por lo que si queremos hacer uso de esta tecnología deberemos comprar un adaptador USB-WiFi (conviene en este sentido revisar los foros del proyecto para ver cuales son fáciles de configurar y cuales no)

Caja: Las primeras unidades se están vendiendo sin caja, no obstante a lo largo de este año está previsto que las nuevas unidades se vendan con una caja (también se venderán cajas sueltas).

Botón de encendido: Raspberry Pi no tiene un botón de encendido/apagado, por lo que bastará con conectarla y desconectarla a la corriente para poder apagarla y encenderla.

Sistema operativo

Tal y como hemos visto Raspberry Pi viene sin tarjeta SD y por tanto sin sistema operativo. Seremos nosotros los

que nos descarguemos el sistema operativo que prefiramos desde el apartado de descargas [3]. De forma oficial se nos recomienda usar Debian “Squeeze”

Dónde comprar

Raspberry Pi está disponible tanto en Premier Farnell/Element 14 [4] como en RS Components [5] y Allied Electronics [6] Las tres tiendas online tienen una red de distribución global por lo que podremos recibirla en nuestro país sin problema. Estos mismos proveedores nos ofrecen también los componentes adicionales que necesitemos (tarjetas SD, tomas de corriente, cables, etc)

El “Modelo A” aún no está a la venta pero saldrá con un precio de \$25 y el “Modelo B”, ya disponible para su compra, tiene un precio de \$35. A estos precios habrá que añadir los gastos de envío así como los posibles impuestos que sean aplicables en cada país.

Debido a la gran demanda que se ha producido sólo se puede comprar temporalmente una Raspberry Pi por persona, en todo caso esta restricción será eliminada tan pronto como se normalicen los valores de oferta y demanda.

Posibles usos

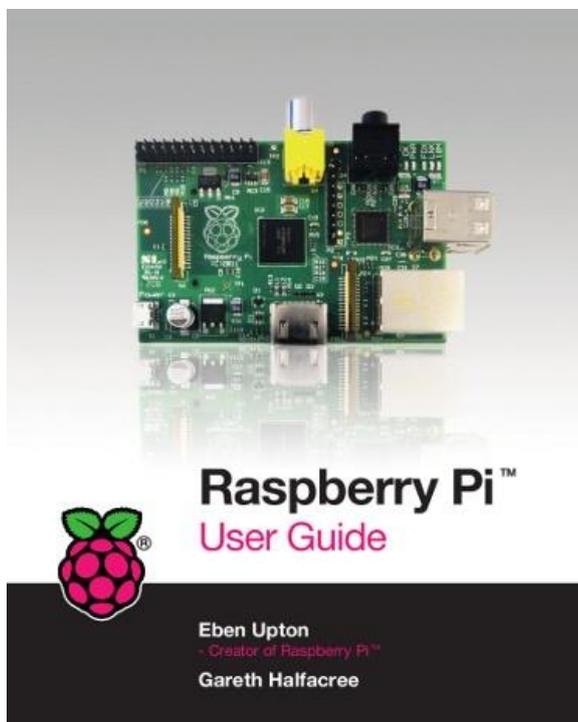
Ya hemos señalado al comienzo de este artículo de los posibles usos que podemos dar a la Raspberry Pi: uso como PC de sobremesa, centro multimedia, servidor web, servidor ownCloud, servidor eyeOS, ...

La lista es tan larga como queramos, en todo caso en el próximo número de la revista Tuxinfo estaremos publicando tutoriales de cómo configurar Raspberry Pi para diferentes usos.

Alternativas

Raspberry Pi ha sido sin lugar a dudas el caso de mayor éxito dentro de un nuevo modelo de ordenadores de muy bajo coste. En todo caso no está sola sino que existen alternativas.

Via APC: Es la apuesta de Vía Technologies con un formato muy similar al Raspberry Pi. Dispone de mayor RAM, procesador más rápido, y mayor conectividad. Se recomienda utilizar con Android 4.0 y aunque aún no está a la venta está previsto que tenga un coste de \$49



Libro acerca de Raspberry Pi que ya se encuentra en fase de preventa

FXI Cotton Candy: Es un ordenador con forma de pendrive con procesador a 1.2 Ghz, Wifi, Bluetooth, 1 GB de RAM, HDMI y USB. Podemos encontrar información más detallada en <http://www.fxitech.com>. Aún no disponible a nivel mundial pero tiene un precio cercano a los \$200

MK802: Creado específicamente para correr Android es un dispositivo muy similar al Cotton Candy de FXI pero a un precio mucho menor: \$74

BeagleBoard: otro ordenador integrado en un circuito impreso, con mayor conectividad que Raspberry Pi y con un precio cercano a los \$150. Más información en <http://beagleboard.org>

Mele A1000: por \$70 nos ofrece un procesador más rápido que el que trae Raspberry Pi así como un puerto SATA, todo ello dentro de una caja.

Como podemos ver el mercado de los ordenadores ultra-low-cost está emergiendo y cada vez son más las empresas que nos ofrecen sus productos.

Enlaces de interés

A pesar de la alta demanda registrada aún no han podido venderse muchas unidades de la Raspberry Pi (por problemas iniciales en la fabricación de las tarjetas) y por ello es aún escasa la información práctica existente en Internet.

En castellano merece la pena visitar los blogs de Fernando Manso [7] y de Juanmol [8], así como el apartado de Raspberry Pi de elinux.org [9] En inglés la mejor información la encontraremos en la propia página del proyecto [1] donde encontraremos el Foro, la Wiki, FAQs, ... en el canal de YouTube de RaspberryPiBeginners [10] (un canal no oficial pero lleno de excelentes tutoriales) o en el libro "Raspberry Pi: User Guide" [11] que está a punto de publicarse y cuyo autor es el propio creador de la Raspberry: Eben Upton.

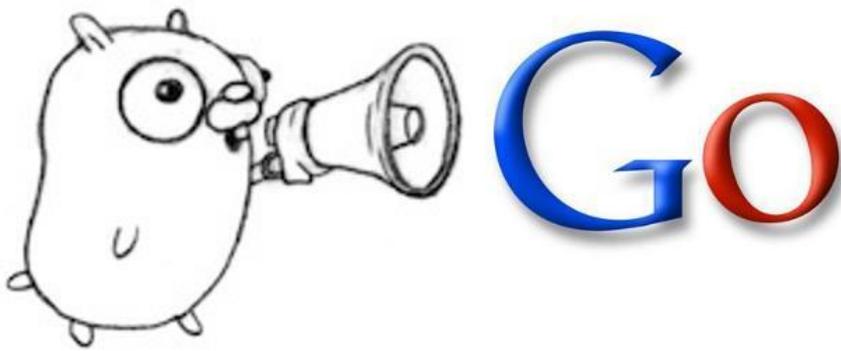
En el próximo número de la revista Tuxinfo os ofreceremos tutoriales prácticos de cómo configurar y utilizar la Raspberry Pi para diferentes usos.



Jorge Cacho Hernández
<https://about.me/jorge.cacho.h>

Enlaces de interés

- [1] <http://www.raspberrypi.org/>
- [2] <http://www.tuxinfo.com.ar/tuxinfo/?p=672>
- [3] <http://www.raspberrypi.org/downloads>
- [4] <http://www.element14.com/community/groups/raspberry-pi>
- [5] <http://uk.rs-online.com/web/generalDisplay.html?id=raspberrypi>
- [6] <http://www.alliedelec.com/RaspberryPi/>
- [7] <http://fmanso.tumblr.com/>
- [8] <http://rsppi.blogspot.com.es/>
- [9] http://elinux.org/R-Pi_Hub
- [10] <http://www.youtube.com/user/RaspberryPiBeginners>
- [11] <http://www.raspberrypi.org/archives/1498>



Boys say Go!

POR HERNÁN “HeCSa” SALTIEL

Ya desde hace aproximadamente unos cinco años que la gigante empresa Google comenzó a trabajar en un nuevo lenguaje de programación. Desde 2007, al menos, tenemos noticias del desarrollo del lenguaje Go. Y desde fines del 2011 sabemos que está suficientemente estable como para darle una oportunidad en nuestras máquinas. En este artículo vamos a meternos un poco en las generalidades del lenguaje Go 1, y quién nos dice, quizá en los próximos encuentros podamos jugar un poco más con él.

Generalidades

En un anuncio de fines del año 2009, Google anunció la existencia de las primeras versiones beta del lenguaje de programación Go, autoproclamándose como rápido, productivo, y divertido para los desarrolladores. El concepto de “diversión de desarrolladores” me dejó algo preocupado. Los otros dos tampoco pasaron desapercibidos por mi cabeza.

En ese momento Goggle describió al lenguaje Go como experimental, y un intento de combinar la velocidad de trabajo de lenguajes dinámicos como lo es Python, con los niveles de performance que se pueden encontrar en C/C++. Cuando leemos las opiniones de algunos miembros del “Go team”, encontramos que lo describen como “compilable en pocos segundos” y “veloz casi como el C/C++”.

Los que hemos tenido el gusto de trabajar con Java sabemos lo que eso significa, y cómo nuestros ojos se pueden llenar de lágrimas con el sólo hecho de pensar que la promesa de performance se cumpla de una vez por todas. Y si aparte hemos trabajado con Perl, Python ó Ruby, nos corre un frío por la espalda de pensar en términos de simplicidad de programación.

Según la literatura nativa del gigante de las nubes, el mundo ha cambiado radicalmente en términos de computación en la última década, pero no han emergido nuevos lenguajes de programación que los acompañen. Por ejemplo, la capacidad de procesamiento de las máquinas se ha multiplicado de forma exponencial, hecho no acompañado por un decremento en los tiempos necesarios para programar dichos sistemas.

Es cierto que la gente de sistemas lentamente (o no tanto) comenzó a dejar los lenguajes de programación fuertes como lo son C/C++ y Java en favor de los interpretados como son Python y Javascript.

Por lo tanto, Go nace como respuesta a la creciente frustración generada en torno de los lenguajes y entornos de programación existentes, dado que la programación se ha vuelto en extremo complicada, así como la elección del lenguaje adecuado para cada tarea. Dicha elección se debía basar en simplicidad en la programación versus rapidez en la compilación, versus ejecución eficiente, no existiendo una combinación de las tres en un mismo lenguaje.

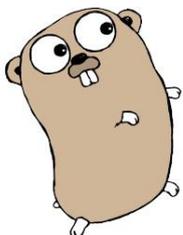
Go intenta cubrir el gap existente entre la simplicidad de programación existente en un lenguaje interpretado y dinámico con la eficiencia y velocidad de uno estático, del estilo de los compilados.

Algo de historia

Robert Griesemer, Rob Pike y Ken Thompson comenzaron a pensar en las bondades de este nuevo lenguaje de programación allá por el 21 de septiembre del 2007, cuando acá en Argentina estábamos festejando la primavera, y allá en USA estaban sintiendo los primeros fríos de otoño, y donde ya se oía el ruido de empresas

como Lehman Brothers que comenzaban a decir que tenían algunos problemitas económicos.

En unos pocos días los objetivos planteados se transformaron en un plan de desarrollo de una solución, y mejor aún, ya se sabía aproximadamente cómo debía ser. Claro está, en ese momento aún todo era una promesa, por lo que se trabajaba part-time en ese proyecto, mientras que se seguía con las tareas habituales.



En enero del 2008 Ken trabajó en un compilador sobre el cual volcar y explorar las ideas que se habían plasmado un tiempo atrás. Ese compilador ya generaba código C en su salida. Pero las cosas comenzaron a ponerse más jugosas para mediados del 2008.

En ese momento las acciones de Lehman Brothers habían bajado un 75%, AIG anunciaba que no podía seguir brindando seguridad para los fondos de inversión en USA, se veía en el horizonte una nueva crisis económica de dimensiones bíblicas, y en Google el proyecto se había vuelto de tiempo completo. Así es, todo el team comenzó a trabajar en forma intensiva en la generación del lenguaje de programación Go.

En mayo del 2008, Ian Taylor de forma independiente comenzó a trabajar en un front end de GCC para Go usando el borrador de las especificaciones existentes. Para finales de ese mismo año, se unió al proyecto Russ Cox para ayudar a mover el proyecto del estado de prototipo al de realidad.

Así, Go se volvió público el 10 de noviembre del 2009, momento desde el cual mucha gente de diferentes comunidades contribuyeron con ideas, discusiones, y por supuesto, código.

De todo este proceso, obtuvimos un lenguaje nuevo, cuya sintaxis puede tener algo de parecido a C, pero con una intensa lavada de rostro, y alguna que otra cosa que nos hará la vida miserable a la hora de la depuración, como lo es la declaración del tipo de una variable luego de su nombre. Si antes pensábamos en "int x", ahora debemos imaginar un "x int".

Ah, también obtuvimos una mascota que es, desde mi humilde punto de vista, espantosa. La idea de ese bicho inmundito, dibujado por un niño que nunca la ha visto, con

lápices gastados, y bajo los claros efectos de drogas pesadas, me deja pensando sobre la verdadera capacidad de la gente de marketing de Google. No entiendo cómo algunas cosas les salen tan bien y otras tan mal. Pero bueh, esta vez la gente de IT se llevó todas las rosas. Como debe ser, qué tanto.

Ni chicha ni limonada



Go es y no es un lenguaje orientado a objetos. Si bien tiene tipos y métodos, y permite un estilo de programación orientada a objetos, no posee una jerarquía de tipos. El concepto de "interface" de Go provee un aspecto bastante diferente del que posee Java por ejemplo, ya que se consideró, desde el momento del diseño, que ésta sería mucho más fácil de usar, y de una u otra forma, mucho más general.

Así y todo tenemos formas de embeber tipos dentro de otros tipos, logrando un comportamiento similar (nótese que digo similar, y no igual) al que tendrían las subclasses. Luego, los métodos en Go son más sencillos que en C++ o Java, ya que se pueden definir para cualquier tipo de datos, aún para los incorporados en el mismo lenguaje.

En sí, en elementos como éstos, o como los referidos al manejo de interfaces en general, funciones y métodos, o dispatch automático de método; vemos el intento de Google por generar un lenguaje de programación que simplifique muchas de las cosas que hoy en día nos molestan en otros lenguajes.

Empecemos

Sé que ya les debe estar picando el cuerpo por las ganas de probar este lenguaje de programación. Si no es así, hay médicos y otros dealers que pueden ayudarlos. Entonces vamos a ver cómo instalamos el lenguaje Go en nuestras máquinas, y comenzamos a hacer de las nuestras.

Lo primero será entender que tenemos dos tipos de compilador Go: uno llamado "gc", independiente, y otro llamado "gccgo", que apunta a ser parte del proyecto GCC. Duerman tranquilos, porque si ejecutan "yum search gccgo", o "apt-cache search gccgo", no van a encontrar nada.

Según dice Google, gc es más maduro y está mejor probado que gccgo. Por lo tanto, iremos por la primer opción.

Recordemos que para poder instalar gc nuestro sistema deberá poseer FreeBSD 7 o superior (al día de la fecha Debian GNU/kFreeBSD no está soportado), Linux 2.6.23 o superior con glibc (CentOS, RHEL, o sus derivados en versión 5 no están soportados, así como las distribuciones para ARM), MacOS X 10.6/7 (utilizando el gcc que

obtenemos con Xcode), o Windows 2000 o superior (usando mingw gcc, obviamente).

Bajaremos entonces el paquete correspondiente a nuestra distribución desde el URL <http://code.google.com/p/go/downloads/list>, lo descomprimiremos y extraemos del tar en /usr/local, y agregaremos los valores de este nuevo path a nuestras variables de entorno:

```
# cd /usr/local
# tar zxvf /home/hecsa/Downloads/go1.0.2.linux-amd64.tar.gz
...
$ vi .bash_profile
(agregar estas líneas)
export GOROOT=/usr/local/go
export PATH=$PATH:$GOROOT/bin
(salvar el archivo)
$ . ~/.bash_profile
]$ which go
/usr/local/go/bin/go
```

Listo, el prolongado proceso de instalación del lenguaje Go 1 ha llegado a su fin. Sencillo, ¿no?

La hora de la verdad

Como siempre, un programador no puede jactarse de serlo si no ha escrito su primer programa, es decir, el tan afamado "Hola, mundo!". No podemos ser menos en este caso, así que aquí vamos:

```
$ vi hola.go
(agregar este contenido)
package main
import "fmt"
func main() {
    fmt.Printf("Hola, mundo!\n")
}
(salvar el archivo)
```

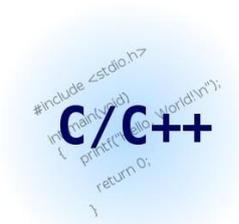
Y como es de esperar, lo ejecutamos con el comando "go run":

```
$ go run hola.go
Hola, mundo
```

Ahora bien, lo primero que me vino a la mente cuando vi este tan elaborado programa en base al cual se erigen cursos enteros de varios meses de duración, fue analizar hasta dónde lo comentado por Google era o no real, y para eso, decidí armar dos programas más, también una nueva demostración de increíbles niveles de desarrollo por mi parte humildemente.

El primero es en C nativo:

```
$ vi hola.c
(agregar estas líneas al archivo)
#include<stdio.h>
main()
{
    printf("Hola, mundo!\n");
}
(salvar el archivo)
```



Y lo compilo con el GCC, pero utilizando el comando "time" antes, veremos para qué en breve:

```
$ time gcc -o hola hola.c
real    0m0.775s
user    0m0.085s
sys     0m0.079s
```

Como uno de los puntos de comparación es el lenguaje de programación Java, no quiero dejar de comparar el nivel de performance frente a un programa altamente elaborado como el que estamos escribiendo aquí.

Entonces, escribo su equivalente en Java:

```
$ vi HolaMundo.java
(agrego estas líneas al archivo)
class HolaMundo
{
    public static void main(String args[])
    {
        System.out.println("Hola, mundo!");
    }
}
(salvo el archivo)
```



Y lo compilo con el JDK 1.7.0_05 de Oracle:

```
$ time javac HolaMundo.java
real    0m2.020s
user    0m2.826s
sys     0m0.143s
```

Como el código de Go puede también compararse con un código interpretado (en palabras de los mismos desarrolladores, se lo compara en simplicidad con Python), he decidido armar el mismo código avanzadísimo, el "Hola, mundo!", pero en Python:

```
$ vi Hola (nótese la mayúscula para diferenciarlo del programa en C)
(agregar estas líneas)

#!/usr/bin/python
print "Hola, mundo!"

(salvar el archivo)

$ chmod +x Hola
```



Ahora ha llegado la hora de la verdad. Ejecutaremos cada uno de los tres "Hola, mundo!", y veremos el tiempo que significa para un mismo sistema:

```
$ time go run hola.go
Hola, mundo!

real    0m0.521s
user    0m0.433s
sys     0m0.074s

$ time java HolaMundo
Hola, mundo!

real    0m0.234s
user    0m0.138s
sys     0m0.040s
```

```

$ time ./hola
Hola, mundo!

real    0m0.003s
user    0m0.000s
sys     0m0.003s

$ time ./Hola
Hola, mundo!

real    0m0.077s
user    0m0.055s
sys     0m0.018s

```

Como podemos ver, el programa en C es por lejos una mejor opción para este tan elaborado ejemplo. Pero no olvidemos una pequeña trampa: nosotros nunca hemos compilado el programa en Go. Por lo tanto, si queremos ser justos, al menos en un primer momento, y con un razonamiento casi irracional, tendríamos que sumar los tiempos. Entonces:

```

Go:
real    0m0.521s
user    0m0.433s
sys     0m0.074s

C:
real    0m0.003s + 0m0.775s = 0m0.778s
user    0m0.000s + 0m0.085s = 0m0.085s
sys     0m0.003s + 0m0.079s = 0m0.082s

Java:
real    0m0.234s + 0m2.020s = 0m2.254s
user    0m0.138s + 0m2.826s = 0m2.964s
sys     0m0.040s + 0m0.143s = 0m0.183s

Python:
real    0m0.077s
user    0m0.055s
sys     0m0.018s

```

¿Es posible extraer una conclusión de esto? Sí, pero muy vaga: C es por lejos más rápido que Go, y ni que hablar que Java. La relación en tiempos de ejecución es de 1 a 180, aproximadamente. A Java directamente lo dejamos fuera de este partido, juega en otra liga, mucho más lenta, por supuesto. PERO, si consideramos que Go contempla también dentro de su entorno el interpretar el código, vemos que Python aún es más rápido que Go en una relación de 1 a 7.

Obviamente, no es esto lo que hay que evaluar en un lenguaje de programación, ya que estamos de acuerdo que escribir, por ejemplo, una página web utilizando C va a llevarnos notablemente más tiempo que 0.778 segundos, así como dudo que podamos armar un driver en html. Pero bueno, son las primeras pruebas que ejecuto, y son reales.

Conclusión

En este artículo hemos entregado sólo una breve reseña de lo que es el lenguaje de programación Go 1, y hemos hecho una despiadada demostración de genialidad de

desarrollo al adentrarnos en complicadísimos algoritmos dignos de la más avanzada bibliografía (no se lo crean, por favor). En futuros números podremos ponernos a sacarle un poco más de jugo a este nuevo lenguaje, viendo para qué es bueno, y qué podemos hacer con él. ¡Hasta pronto, amigos!

Y recuerden: "Don't say no, boys say Go!"

Hernán "HeCSa" Saltiel
 AOSUG leader
 CaFeLUG Member
 Boca happy fan
 Club Amigos de Pumper Nic
 hsaltiel@gmail.com
<http://www.aosug.com.ar>



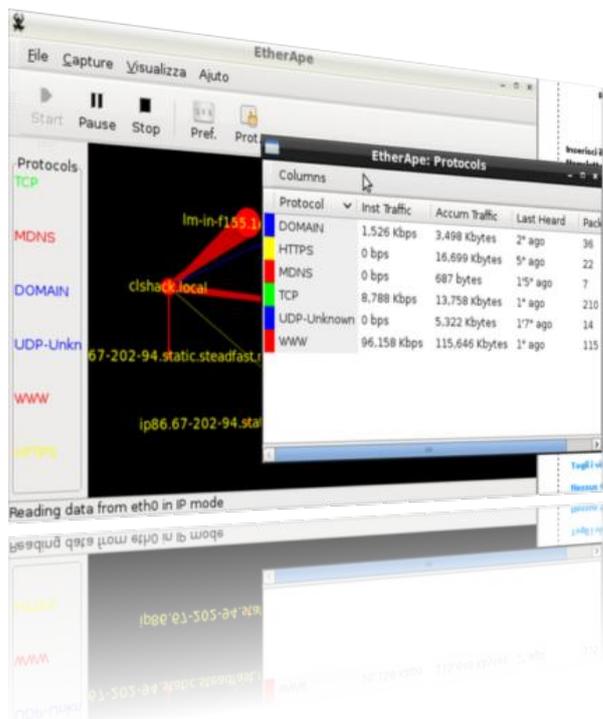
PD: El título de este artículo y su frase final se refieren a un tema musical antiquísimo de Depeche Mode que se puede escuchar aquí, en su excelente versión en vivo en Londres, año 1986:

<http://www.youtube.com/watch?v=qVAUOxveuvc>

EtherApe I

Monitoreando conexiones de Red

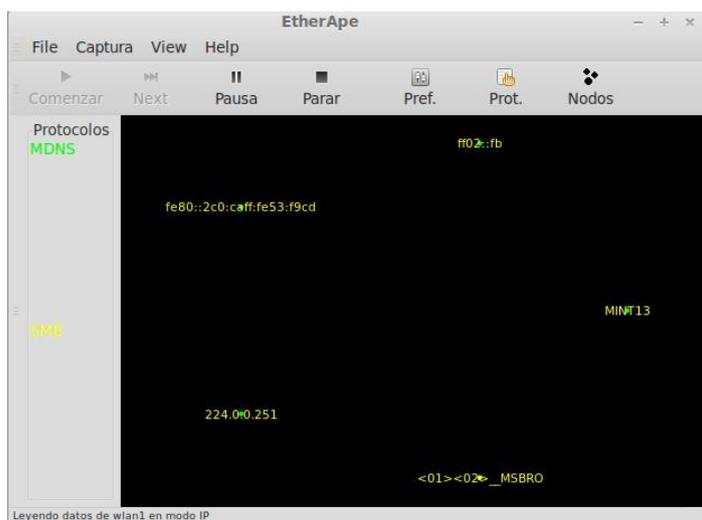
POR VICTOR HUGO



Luego de varios años de ausencia, estamos de regreso. Me voy a enfocar en la instalación y uso de programas libres.

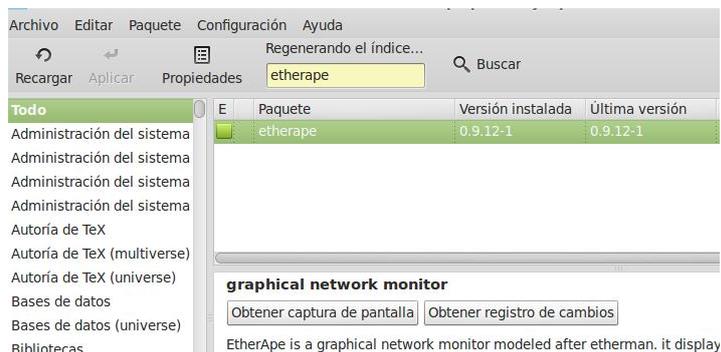
Soy usuario de la distribución Linux Mint, siendo esta una distro que va de maravillas en mi Netbook.

Para esta primera parte, vamos a mostrar las generalidades del "EtherApe", que según su desarrollador (Juan Toledo), es un Software Libre creado bajo licencia GNU que sirve para monitorear las conexiones de red de manera gráfica, en la capa IP y TCP.



Instalación:

Podemos instalarlo mediante el Gestor de Paquetes Synaptic o mediante línea de comandos con `sudo apt-get install etherape`



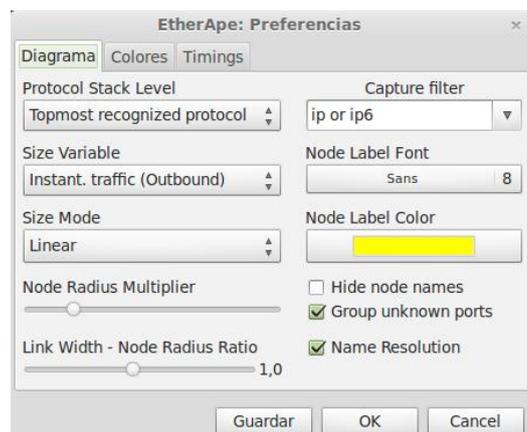
Uso:

No olviden que para ejecutarlo y usarlo debemos tener los privilegios de root. La ubicación para poder comenzar a utilizarlo, en mi caso, se encuentra en la siguiente ruta: Internet --> EtherApe (as root)

También podemos ejecutarlo mediante línea de comandos, para ello digitamos: `etherape`

Configuración

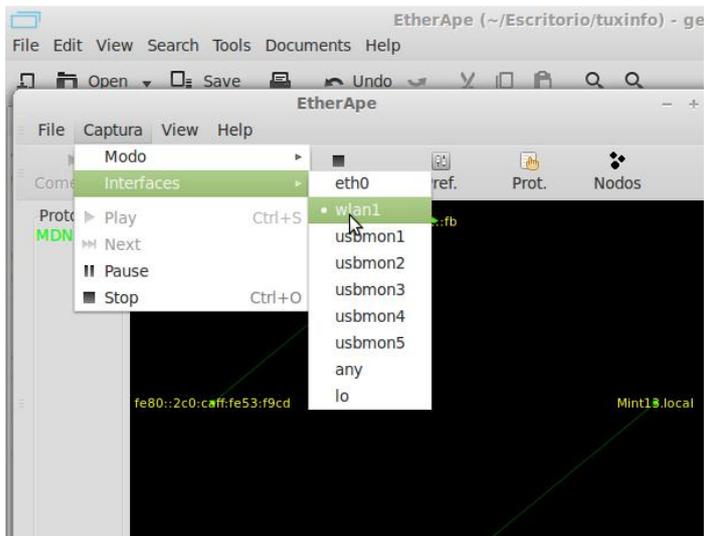
Este programa tiene una configuración bastante intuitiva de usar. En la barra de menú, vamos a File --> Preferences



Ahora vamos a configurar la interfaz que vamos a escuchar, para ello vamos a:

Captura --> Interfaces

Ahí seleccionamos la interfaz que tenemos activa y podremos ver todas las conexiones que realiza nuestro equipo.



Protocol	Port	Inst Traffic	Accum Traffic	Avg Size	Last Heard	Packets
DHCPV6-CLIENT	546	0 bps	29,60 Kbytes	150 bytes	3'57" ago	202
DOMAIN	53	0 bps	370,39 Kbytes	120 bytes	1'27" ago	3148
HOSTMON	5355	0 bps	18,15 Kbytes	70 bytes	2'50" ago	267
HTTP	80	0 bps	564,89 Kbytes	403 bytes	4" ago	1435
HTTPS	443	0 bps	9,67 Kbytes	211 bytes	1'9" ago	47
IP_UNKNOWN	-	0 bps	450 bytes	90 bytes	9'58" ago	5
MDNS	5353	0 bps	588 bytes	98 bytes	2'25" ago	6
NETBIOS-NS	137	0 bps	90,85 Kbytes	96 bytes	2'49" ago	971
SMB	-	0 bps	96,29 Kbytes	210 bytes	58" ago	470
TCP_UNKNOWN	-	0 bps	244,30 Kbytes	655 bytes	50" ago	382
UDP_UNKNOWN	-	0 bps	56,40 Kbytes	183 bytes	10" ago	315
XMPP-CLIENT	5222	0 bps	203,10 Kbytes	203 bytes	39" ago	1025

Bueno espero que se haya entendido, para la siguiente entrega, lo vamos a instalar compilando el código fuente y trabajaremos con algunos ejemplos.

Hasta la próxima.

Web: <http://etherape.sourceforge.net/>

Victor Hugo
aguilarpia33@gmail.com

Vistas

Tenemos dos tipos de vistas: Nodos y Protocolos

Name	Address	Inst Traffic	Accum Traffic	Avg Size
192.168.1.132	192.168.1.132	0 bps	7,27 Kbytes	92 bytes
224.0.0.252	224.0.0.252	0 bps	256 bytes	64 bytes
23.56.227.172	23.56.227.172	0 bps	102,54 Kbytes	544 bytes
239.255.255.250	239.255.255.250	0 bps	1,54 Kbytes	175 bytes
MYCOMPUTER-PC	MYCOMPUTER-PC <20>	(Server service)	0 bps	243 bytes
Mint13.local	192.168.1.240	0 bps	13,58 Mbytes	444 bytes
OSCAR	OSCAR <20>	(Server service)	0 bps	243 bytes
WORKGROUP	WORKGROUP <1d>	(Master Browser backup)	0 bps	24,95 Kbytes
client-200.60.190.25	200.60.190.25	0 bps	10,64 Kbytes	218 bytes

Zimbra™
Collaboration Suite
Linware
www.linware.com.ar
zimbra@linware.com.ar

En cualquier lugar, en cualquier máquina

Zimbra webmail interface showing an email inbox for Chris Smith. A message from Chris, Lars is selected with the subject 'project status - meeting'.

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

vmware®
Business Partner

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



Fuduntu

Distribución Linux Rolling Release

POR LOZANO JUAN PABLO

Fuduntu es una distribución Linux ligera y divertida basada en Fedora 14. El objetivo es proporcionar una experiencia de usuario agradable estéticamente, con un máximo de aplicaciones actualizadas que hacen de Fuduntu una distro ideal para el escritorio.

Los usuarios de portátiles y netbooks estarán encantados de saber que Fuduntu está optimizado para la computación del día a día y proporciona herramientas para ayudar a alcanzar la vida máxima de la batería. Se puede apreciar que la vida de la batería mejora en un 30% o más con respecto a otras distribuciones Linux.

Otra gran ventaja es que Fuduntu es una distribución Rolling Release. Después de la instalación, no hay que preocuparse de volver a llevar a cabo la misma nuevamente en unos meses después, para beneficiarse de la última versión de software disponible. También usa el último Kernel Linux estable en su versión 3.2.13, aumentando la cantidad de controladores (drivers) para diferente hardware, mejorando así la compatibilidad de este Sistema Operativo.

Las características de Fuduntu:

- Estéticamente agradable
- Escritorio centrado en el diseño
- Increíble rendimiento
- Duración de la batería sin igual

Algunos de los paquetes predeterminados que se encuentran en Fuduntu:

- Chromium (Navegador de Internet)
- Banshee (Reproductor de Música)

- VLC (Reproductor de Videos)
- Adobe Flash (Con licencia)
- Fluendo Codec MP3 (Con licencia)
- Infinality Freetype
- Nautilus Elementary (Gestor de archivos)
- Júpiter (Administrador de la Batería)

Como han notado, no todo el software que se incluye es precisamente LIBRE, pero hacen que funcione con más soporte y optimismo (es el caso de algunos codecs y plugins requeridos para el uso diario).

El curioso nombre

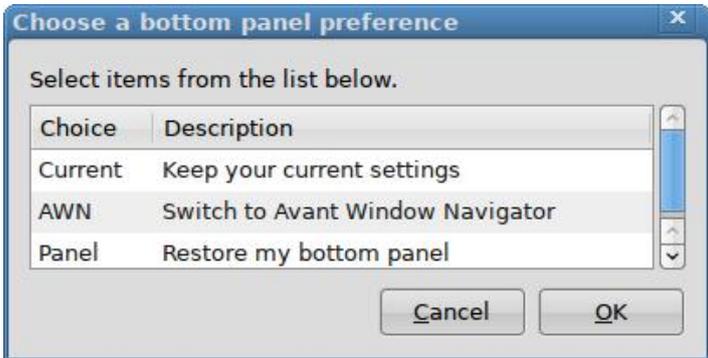
El equipo de desarrollo lo bautizó con el nombre Fuduntu con la intención de estar entre medio de dos grandes distribuciones, Fedora y Ubuntu. Si bien se basa en Fedora, ha tratado de igualar en cuanto a cantidad de software incluido en la imagen final para poder así resolver algunas incomodidades que nos surjan y hacer todo lo que necesitemos de una forma más rápida y fácil como Ubuntu. De aquí el famoso slogan de la distribución: "Punny Name, Serious Distro!" que significa "¡Curioso Nombre, Distribución Seria!".

En detalles

El entorno de escritorio predeterminado que usa Fuduntu es Gnome 2.X que si bien no es un entorno actual (sin contar que ya no recibe soporte) es uno de los más estables y rápidos que existe hoy en día.

Tal vez para muchos no es de su agrado utilizar un Dock en la parte inferior de la pantalla como al estilo Mac OS X (de Apple) y tengan ganas de usar un panel, para ello hay

a disposición algunas aplicaciones exclusivas de la distribución que permiten hacer fácilmente con 1 clic este cambio. Además contamos con otro poderoso software que nos ayuda a activar el Cubo 3D del famoso compiz y desactivarlo de manera muy fácil.



Sí, es cierto que todas estas aplicaciones no tienen un nombre propio, pero nos abastecen con gran cantidad de funcionalidades y configuraciones para poner más a gusto nuestro entorno de trabajo y así disfrutar a pleno todos los aspectos de Fuduntu.

Una breve comparación

NO vamos a criticar cada detalle de otras distribuciones para que Fuduntu sobresalga. Mi experiencia personal me ha demostrado que Fuduntu, por el hecho de usar Gnome 2.X, hace que funcione más ligero y estable que el entorno predeterminado de Ubuntu, ya que tiene más años de trayectoria que Unity. A la vez es mucho más eficaz

trabajar por su simplicidad, que el entorno predeterminado de Fedora 17 (Gnome 3.X), esto no quiere decir que tanto Unity como Gnome 3.X sean malos entornos, pero sí afirmamos que este último es mucho más liviano y ágil que los otros comparados.

La estabilidad que han logrado, así como la rapidez, solidez y buen diseño estético, hacen del entorno de Fuduntu una opción a tener en cuenta antes de elegir una distribución Linux. Por ello y por muchas otras cosas, te invitamos a que pruebes tú mismo esta distribución y te animes a algo diferente.

Si quieres obtener una copia del sistema puedes pasar por la página oficial del proyecto Fuduntu y descargar una réplica en línea del sistema.
<http://www.fuduntu.org/get.html>



Lozano Juan Pablo
lozanotux@gmail.com
twitter: @lozanotux



Síguenos
también en
Facebook



Instalación Fedora 17

POR RINO RONDAN

Antes de empezar la instalación he bajado las imágenes live-cd desde este sitio <http://get.fedoraproject.org/>

Modos de instalación de Fedora:

- Live-cd -> <http://fedoraproject.org/en/get-fedora-options>
- Dvd -> <http://fedoraproject.org/en/get-fedora-options#formats>
- Spins -> <http://spins.fedoraproject.org/>
- Mirrors -> <http://mirrors.fedoraproject.org/publiclist/>

Los diferentes modos se diferencian en que el live-cd contiene únicamente una instalación básica del sistema operativo y si quisieran tener más software disponible tendrían que usar el DVD pero la gran diferencia con el live-cd es que el DVD sólo es de instalación, no sirve para probar alguno de los spins, ya que lo menciono, los spins son una especie de "sabores", tiene varias opciones de Fedora donde cada uno se perfila para un uso diferente. Importante no olvidar que Gnome es el escritorio predeterminado, si desean utilizar otro tienen que bajarse el spin correspondiente.

Para otros fines más diversos hay una imagen de Fedora-17-x86_64-netinst.iso donde está mucho más personalizada la instalación para que tengan un fedora mínimo y lo puedan usar como servidor.

Ahora ya sabemos de donde obtener la imagen de Fedora 17, podemos probarla en nuestro equipo por medio de un live-cd o instalarlo que es lo que vamos a comenzar. Doy por entendido que se sabe cómo grabar un cd y probar un live-cd, este tipo de medio sirve para ver si funciona bien la distro que elegimos.

Recomendaciones Previas:

Una recomendación previa es hacer backup de todo lo que crean necesario porque una opción errónea a la hora de la instalación puede resultar en pérdida de datos.

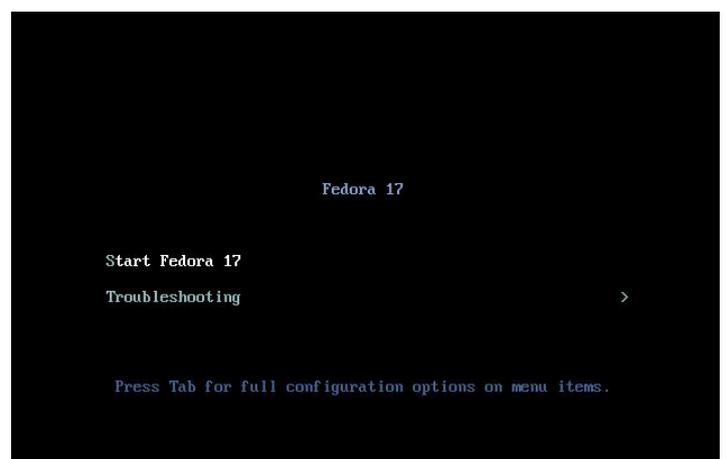
Otra recomendación previa es con alguna herramienta como Gparted Live, System Rescue CD, Ultimate Boot CD, o el mismo livecd, redimensionar el disco para poder tener una partición libre. Existen muchas herramientas para estas tareas, estas son algunas de todas las que hay.

Instalación:

Como mencionamos anteriormente, podemos grabar la imagen Live-CD descargada y probarla para luego instalarla o podemos grabar la imagen del DVD. En este caso voy a tomar una imagen de Live-CD, que se puede grabar o en un CD o en un pen drive, es mucho más rápido (ver liveusbcreator).

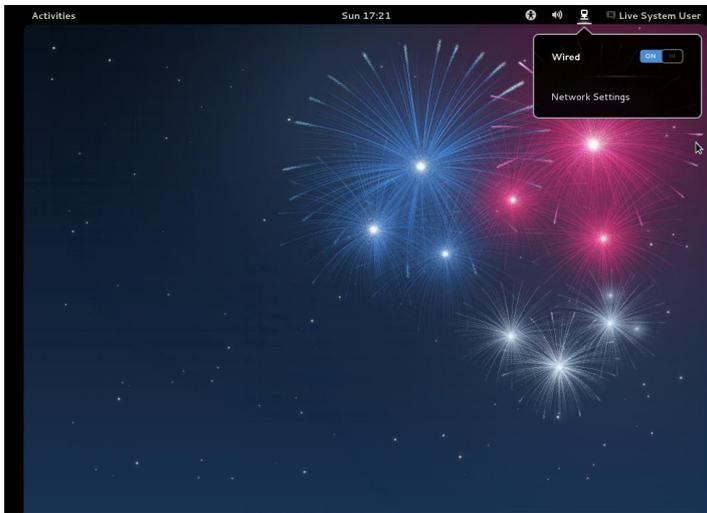
Primeros Pasos:

Vamos a ver las primeras pantallas del inicio de la instalación. Este es el menú del livecd:



Una vez que el sistema va iniciando vamos a ver una barra azul de progreso si apretamos la tecla escape vamos a ver con más detalle qué está pasando.

Ahora una vez que arrancó tenemos gnome3 y el sistema ya iniciado:



Seguramente más de uno se lleve la sorpresa de que con Gnome3 la forma de como se administran las ventanas cambió mucho y no se acostumbren, les comento esto porque me ha pasado, tan solo hay que usarlo un tiempo y se acostumbran, me ha pasado que he usado gnome2 en otra vm y no me encontraba cómodo (extrañaba gnome3), así que sólo es cuestión de acostumbrarse :).

Podrían tomarse un rato para probar con la nueva interfaz, si están totalmente decepcionados o inconformes basta elegir otro spin y realizar la instalación que no depende del tipo de gráfica, es igual para todos.

Para empezar hacen click en donde dice activities o mueven el mouse bien en esa esquina y se activa el menú.

Van a ver que tienen un buscador si escriben la palabra adecuada intenta buscar la aplicación con ese nombre, no hace falta hacer foco, al escribir ya se completa en donde corresponde.

La otra opción es hacer click en Application y tienen el menú a la izquierda y sobre la derecha los atajos a las aplicaciones más usadas.

Todo esto es super configurable, hasta se puede activar el menú antiguo para los más nostálgicos pero no vamos a entrar en esos detalles.

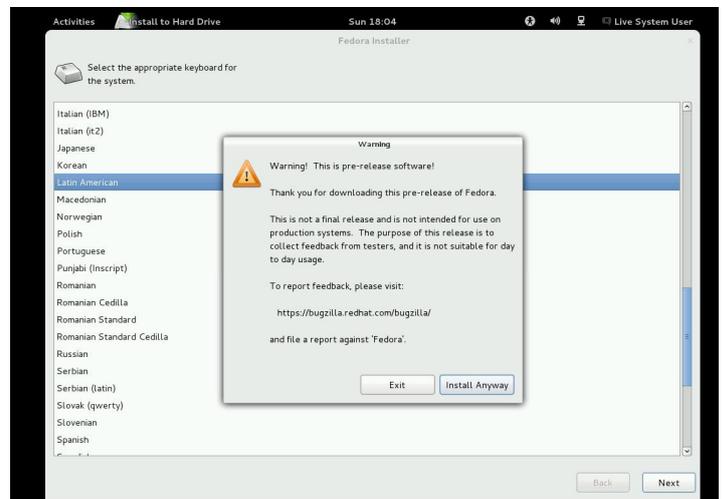


Instalar:

Ahora que ya nos tomamos el tiempo necesario para probarlo, hacemos click en activities y luego en la barra de la izquierda en el último icono (el dibujo del disco con una flecha verde que apunta hacia abajo), si utilizan otro spin basta con buscar el mismo icono.

Paso 1:

En este paso debemos seleccionar el mapa de caracteres correspondiente al idioma que deseamos manejar:



Como verán cada aplicación que abren muestra en la barra de arriba la que está activa con su icono correspondiente.

Luego del primer next (siguiente) van a ver un mensaje de advertencia que les avisa que el sistema es un pre-release o sea beta u alpha, seleccionan install Anyway. (Aclaración ya eso en la versión Estable no aparece)

Paso 2:

En esta ventana nos va a preguntar qué tipo de dispositivo de almacenamiento tenemos, seleccionan el que ajusta a sus necesidades (la primer opción). Al hacer click en next va aparecer un cartel que nos indica que está analizando los dispositivos.



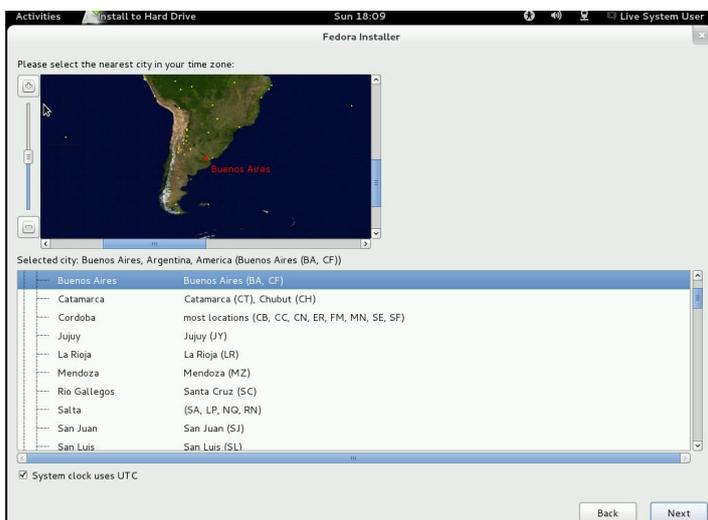
Paso 3:

En esta ventana deberíamos poner el nombre del equipo, en mi caso lo dejamos como está.



Paso 4:

Aquí deberíamos configurar el huso horario que nos corresponda, es más fácil utilizar el mapa y hacer click en la ciudad que corresponda.



Paso 5:

Este paso es uno de los más importantes porque estamos configurando la clave de lo que va a ser el usuario root (administrador con todos los privilegios) por lo cual pongan una clave segura y que no se olviden, si por el contrario están probando cosas en alguna máquina virtual la clave clásica podría ser usada :).

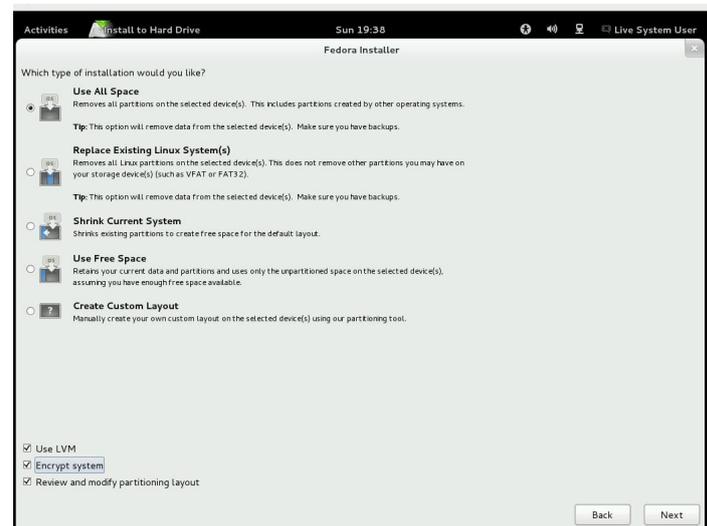
Paso 6:

Si ven el mismo cartel que está en la imagen es porque la clave que escribieron no es muy segura y tendrían que elegir la opción Use Anyway o bien poner algo más seguro :).



Paso 7:

Otro paso muy importante a la hora de la instalación que hay que tener mucho cuidado con lo que se elige porque acá por lo general se selecciona la opción incorrecta y terminan borrando todo el disco (en el caso que tengan dos o más sistemas operativos).



Analizemos cada una de las opciones:

Use All Space: la más peligrosa de toda porque utiliza la totalidad del disco para instalar Fedora sin importar qué hay en el disco, si van a elegir esta opción tengan en cuenta que volver a recuperar los datos no es una tarea sencilla ni para gente con poca paciencia.

Replace Existing Linux System(s): Esta opción va a reemplazar el sistema Linux que tengamos instalado en el dispositivo seleccionado.

Shrink Current System: Con esta opción podemos redimensionar las particiones existentes para así poder liberar espacio para crear una partición vacía que utilizaremos para instalar Fedora.

Use Free Space: Utilizará las particiones definidas que no contengan datos dejando todo lo demás sin tocarlo.

Create Custom Layout: Esta es la favorita de todos :), pero en este caso no es recomendable si no tenemos mucha práctica en instalaciones o menos aun si recién estamos iniciandonos en el mundo GNU/LINUX.

Otras Opciones:

LVM: Fundamental usar esta característica dado que facilita mucho la administración del espacio en nuestro disco.

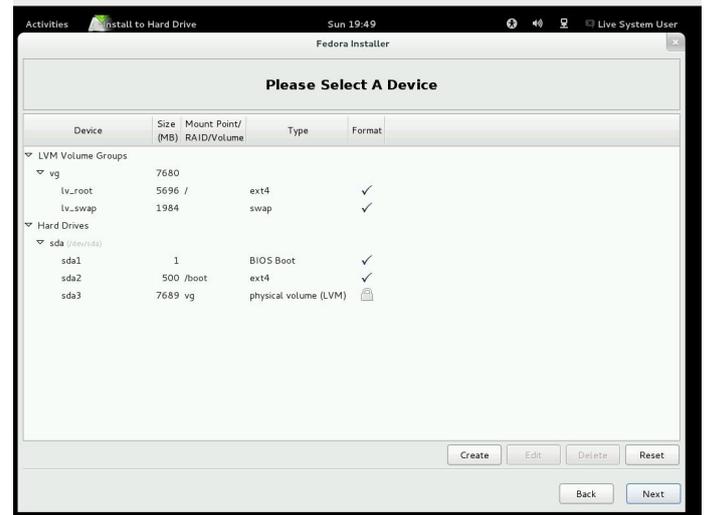
Encrypt system: Si quisieramos tener nuestros datos encriptados es muy importante, tener en cuenta que al iniciar el sistema siempre nos pedirá la clave para desencriptar los datos.

Review and modify partitioning layout: Esta opción me mostrará cómo quedaría particionado todo nuestro disco, dado que si no elegimos la última opción no tendríamos idea de qué fue lo que hizo, por eso si decidimos hacerlo todo automático y quisieramos saber cómo quedó es una buena práctica tildarla.

Recordar que Fedora ya está utilizando la tabla GPT así que no es mala idea empezar a utilizarla.

En este caso voy a seleccionar lo que se ve en pantalla y luego next, acto seguido examinaré los discos y en mi caso la ser una vm el disco está totalmente vacío y me

preguntará si quiero inicializar y borrar todo.



Como verán en el esquema de particiones utiliza LVM con una partición que está encriptada (en este caso se encripta el disco y no el fs, es más práctico sino me preguntaría la clave por cada punto de montaje encriptado), otro punto a tener en cuenta es que el boot quedó afuera del LVM por ciertas limitaciones y dolores de cabeza a la hora de que tengamos problemas con LVM y la sorpresa es que tenemos una partición BIOS Boot que se debe a que a partir de Fedora 15 ya se utiliza GPT y por omisión nos genera todo en GPT, teniendo esta partición BIOS que hace uso de este feature.

Si estamos de acuerdo con lo que creó, le damos next, sino podríamos retocar el esquema a nuestro gusto.

Paso 8:



Al seleccionar la encriptación nos va a pedir que generemos una clave para el sistema encriptado por lo cual es IMPORTANTÍSIMO no olvidar esa clave, y acordarse de no poner algo sencillo si justamente lo elegimos por seguridad.

Luego la pregunta del millón es si deseamos escribir los cambios y acá es donde efectúa TODOS los cambios en el disco.



Paso 9:

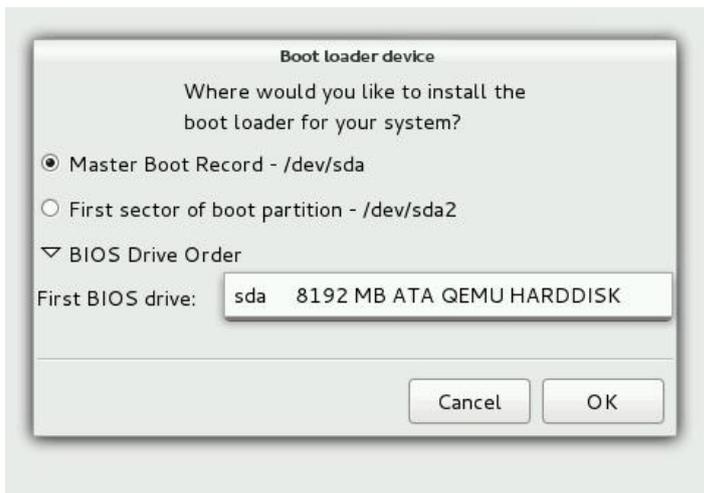


Llegamos a la parte no menos importante de seleccionar dónde se instalará el gestor de arranque, por omisión tomará el disco primario, el que se utiliza para arrancar la máquina, si tenemos varios discos en nuestro sistemas con varios sistemas operativos también, esta parte es muy importante porque una mala configuración puede hacer que o no arranque nuestro Fedora recientemente instalado o que perdamos acceso a los demás sistemas que tenemos instalados.

Resumiendo un poco cuando la máquina inicia necesita pasarle el control a un sistema operativo, si nosotros tenemos varios sistemas operativos justamente necesitamos un gestor de arranque que se encargue de poder administrar el arranque de cada uno de los sistemas que tenemos, entonces al decirle en qué dispositivo se va a instalar el código que contiene la primera etapa de inicio del gestor de arranque en este caso Grub2 deberíamos seleccionar justamente el disco que tenemos configurado en el BIOS como primer disco de arranque, por lo general es así pero se puede configurar e instalar en otro lado si uno quiere.

Luego cuando se pasa el control al gestor de Arraque (grub2), éste se carga en memoria leyendo este código que lo busca justamente en la unidad que le mencionamos en esta pantalla para luego ir a cargar la partición /boot en el disco que elegimos como /boot, como verán este código

de arranque de grub puede estar en otro disco, es por eso que si tenemos varias particiones o discos es importante ver qué es lo que queremos. También podríamos utilizar la misma partición /boot para almacenar varias versiones de Linux (ejemplo si tendríamos F17 y F18 alpha).



Si deseáramos cambiar un poco el arranque.

Paso 10:

Este paso está totalmente dedicado al copiado de la imagen del CD al disco.



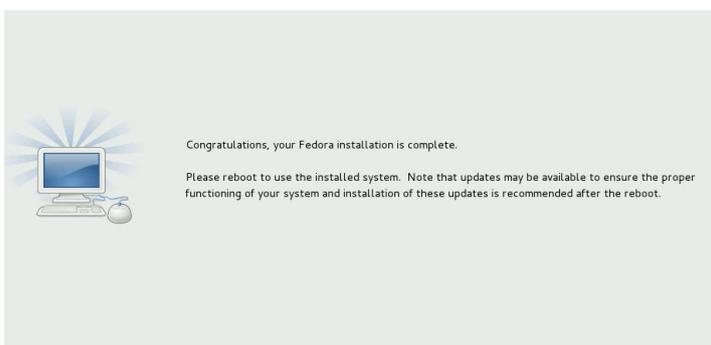
Una vez que termina el copiado de la imagen empieza a ejecutar los pasos más significativos de la instalación con respecto al sistema de archivos.



Luego comienza la instalación del gestor de arranque.



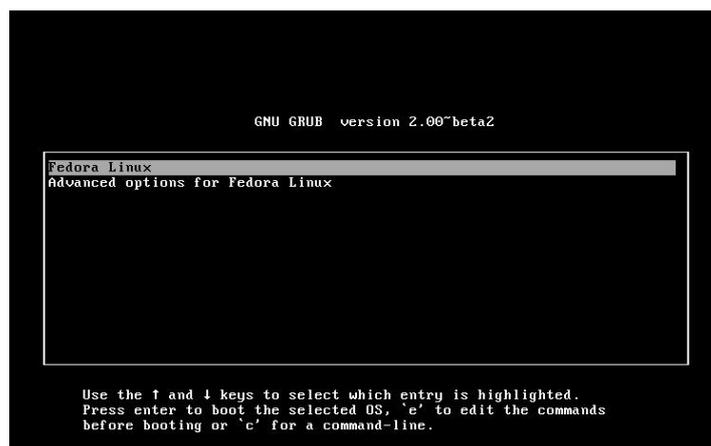
Al terminar todo vemos esta pantalla que nos anuncia que ya está el sistema listo:



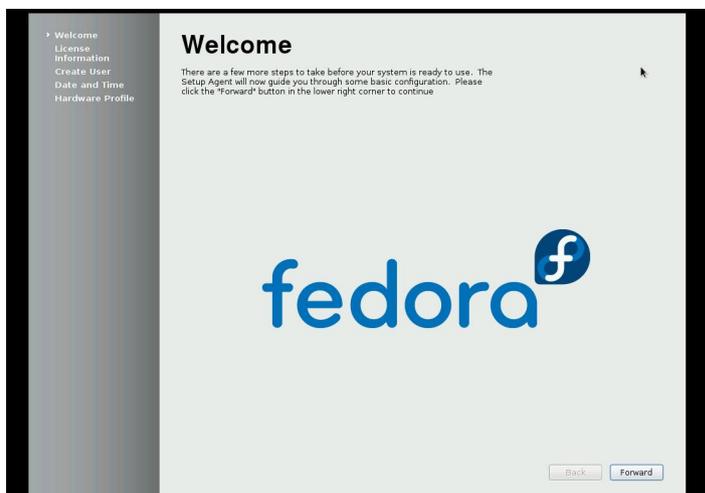
Ahora sí, el paso final es reiniciar y chequear que inicie el sistema instalado.

Paso 11:

Podemos ver que el gestor de arranque inició correctamente y nos muestra el menú del sistema a iniciar.



Luego nos pedirá la clave, y si después de la clave apretamos la tecla escape podremos ver el proceso de arranque para ver si todo está bien, una vez que termina el proceso de arranque nos cargará el login gráfico, al ser la primera vez que lo iniciamos vamos a tener que configurar ciertos aspectos que son los siguientes:



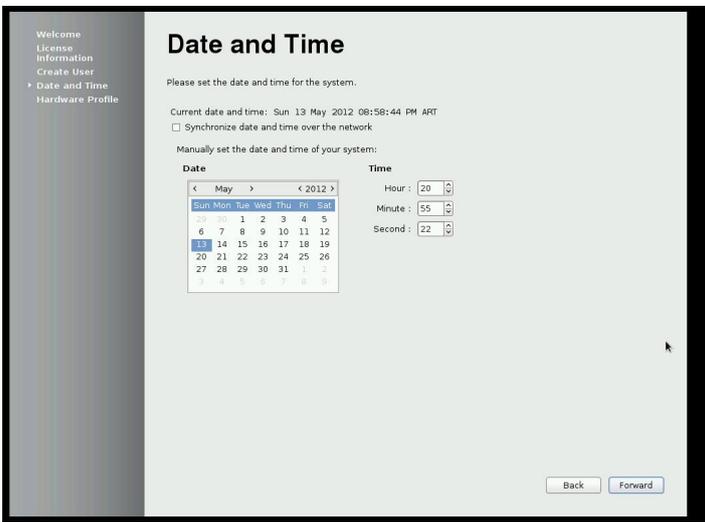
Hacemos click en Forward:



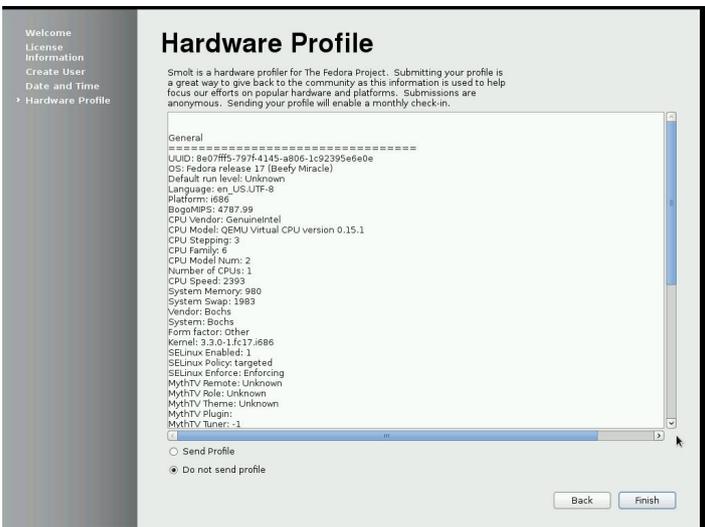
Aceptamos la licencia.



Luego deberíamos poner el usuario que va a utilizar el equipo, hay ciertos aspectos más avanzados que en primera instancia no haría falta configurar si se trata de una instalación hogareña.

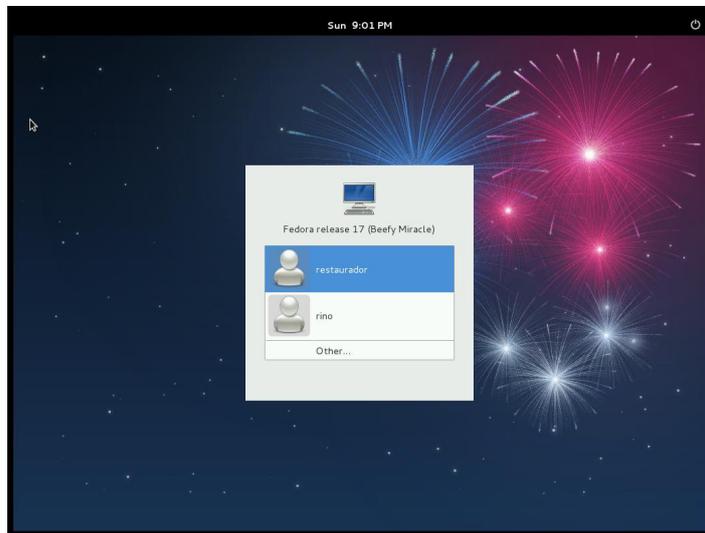


Aquí configuramos la hora como corresponde, ya sea manual o con algún servidor por la red.



Aquí, si quisiéramos subir nuestro perfil de PC al proyecto Fedora, no es necesario así que lo omitimos.

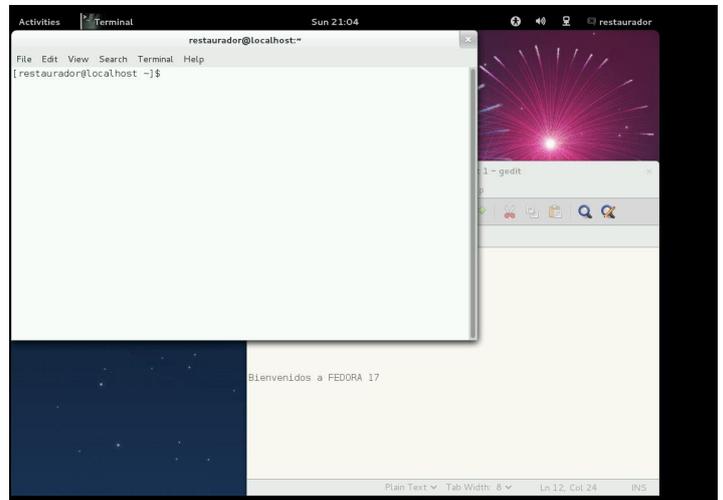
Y luego de esto ya estamos listos para usar nuestro sistema:



Elegimos el usuario que corresponda e iniciamos sesión.



Ahí vemos que utilizamos el buscador de aplicaciones escribiendo el nombre.



Ya tenemos nuestro sistema andando.

Felicitaciones :)

Pasos Post Instalación (opcional):

Una vez con todo el sistema instalado hay que ejecutar algunos pasos extras para personalizarlo más.

En la siguiente pantalla en donde hicimos click en activites, luego applications y finalizando con System Tools, nos mostrará una serie de herramientas importantes como:

Terminal: una terminal de línea de comandos, que no está mal que sepamos usar lo básico

System Settings: Un panel de control de aspectos de configuración del equipo

Add/Remove Software - Software Update: Una aplicación para instalar/actualizar programas de forma gráfica



Agregar Repositorio RPMFUSION:

En una terminal hacemos los siguientes pasos: `su` (ponen la password de root que configuraron en la instalación)

```
yum localinstall --nogpgcheck
http://download1.rpmfusion.org/free/fedora/
rpmfusion-free-release-stable.noarch.rpm
http://download1.rpmfusion.org/nonfree/fedora/
rpmfusion-nonfree-release-stable.noarch.rpm
```

Con esos dos comandos agregamos unos repositorios que nos permitirán poder tener una variedad más grande de programas para instalar, además de los que tenemos en la comunidad Fedora.

Instalación de AutoPlus:

Este aplicativo nos facilitará la tarea de instalar determinado set de programas que no vienen en el repositorio de Fedora. En la terminal donde estábamos como root:

```
yum -y --nogpgcheck install
http://dnmouse.org/autoplus-1.4-5.noarch.rpm

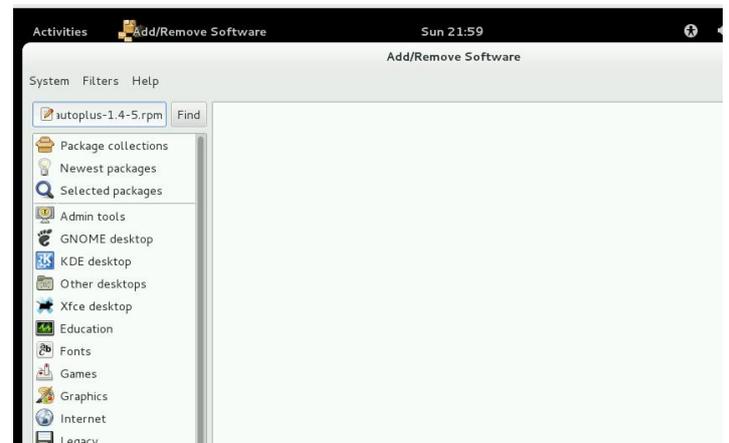
rpm --import http://dnmouse.org/RPM-GPG-KEY-dnmouse
```

Luego vamos al menú de aplicaciones y buscamos autoplus. Y podremos elegir la aplicación que necesitemos e instalarla.

Documentación:

La documentación es una parte muy importante al seleccionar una distribución y es por eso que cabe destacar que Fedora posee muchísima documentación con un nivel de detalle muy preciso, que además tenemos que mencionar con orgullo que esto se debe a la tarea de los documentadores dentro de la comunidad: <http://docs.fedoraproject.org/en-US/index.html> (seleccionar el idioma, la mayoría está en Inglés, falta más gente que traduzca, a sumarse :))

Instalación por medio del programa Add/Remove Software: Podríamos instalar con esta aplicación nuestros programas.



Conclusión:

Vimos cómo instalar Fedora 17 y poder tunear algunos aspectos post instalación, cabe aclarar que si bien usamos la versión BETA, hoy en día ya es estable y es el mismo procedimiento.

Enlace: <http://fedoraproject.org/es/>

Escrito por:

Rino Rondan

<https://fedoraproject.org/wiki/User:Villadalmine>
Embajador

Con la colaboración de:

Matias Kreder

<https://fedoraproject.org/wiki/User>Delete>
Embajador

Adrian Alves

<https://fedoraproject.org/wiki/User:Alvesadrian>
Embajador y Desarrollador

Guía de GNU/Linux (IX)



POR RAFAEL MURILLO

Llevamos ya un largo camino recorrido por esta Guía de Linux, y ha llegado el momento de entrar en un tema que personalmente me parece un poco tedioso y que además requiere de mucha práctica para dominarlo por completo.

Vamos a conocer el editor de texto más utilizado en GNU/Linux, estoy hablando de vim, que aunque parece un editor muy arcaico, sin duda nos sacará de muchos problemas en nuestra carrera como administradores de servidores Linux. Una de las ventajas de este editor de texto sobre otros, es la facilidad que tiene de abrir y editar archivos de texto grandes, muy grandes (estamos hablando incluso de gigas).

Y la pregunta ahora es, ¿Cómo abrimos el editor de textos vim? Pues justo como ya te lo debes estar imaginando, basta con escribir su nombre en la terminal para que se abra nuestro editor de texto:

```
vim
```

Al dar enter en esta instrucción, obtendremos la siguiente pantalla con algunas instrucciones de uso:

```
VIM - Vi IMproved
          version 7.2.445
    by Bram Moolenaar et al.
Modified by pkg-vim-maintainers@lists.aliases.debian.org
Vim is open source and freely distributable

  Become a registered Vim user!
type  :help register<Enter>  for information

type  :q<Enter>               to exit
type  :help<Enter> or <F1>    for on-line help
type  :help version7<Enter>  for version info

  Running in Vi compatible mode
type  :set nocp<Enter>       for Vim defaults
type  :help cp-default<Enter> for info on this
```

Para editar un archivo existente, debemos escribir el comando vim seguido del nombre del archivo (obviamente debemos estar ubicados en el mismo sitio donde tenemos nuestro archivo, o bien, poner la ruta completa hacia el archivo que vamos a editar seguido de su nombre). Por ejemplo, en la siguiente pantalla, podemos apreciar que tengo un archivo llamado listado.txt en la misma ruta donde estoy actualmente, para editarlo debo escribir **vim listado.txt**, veamos el resultado de esto:

```
total 8
-rwxr--rwx 1 root    root      29 may  5 20:40 archivo1.gz
-rw-r--r-- 1 root    root        0 may  5 20:40 archivo2
drwxr-xr-x 2 rmurillo rmurillo 4096 may  5 20:02 Desktop
-rw-r--r-- 1 root    root        0 may  5 21:51 listado.txt
~
```

Podemos observar ahora el contenido del archivo, así como un resumen del mismo en la parte inferior.

Nota: Si ejecutamos vim seguido del nombre de un archivo inexistente, vim lo creará en el directorio donde nos encontramos posicionados.

Ahora bien, nuestro editor de textos vim opera de dos modos distintos:

1. Modo comando: Como podemos imaginar, se utiliza para introducir comandos propios de vim que realizan funciones específicas. Este es el modo predeterminado de nuestro editor.

2. Modo entrada: Obviamente se utiliza para introducir texto o borrarlo.

De los puntos anteriores, debemos rescatar el primero ya que es muy importante hacer una aclaración. Cuando somos usuarios nuevos en Linux y utilizamos este editor

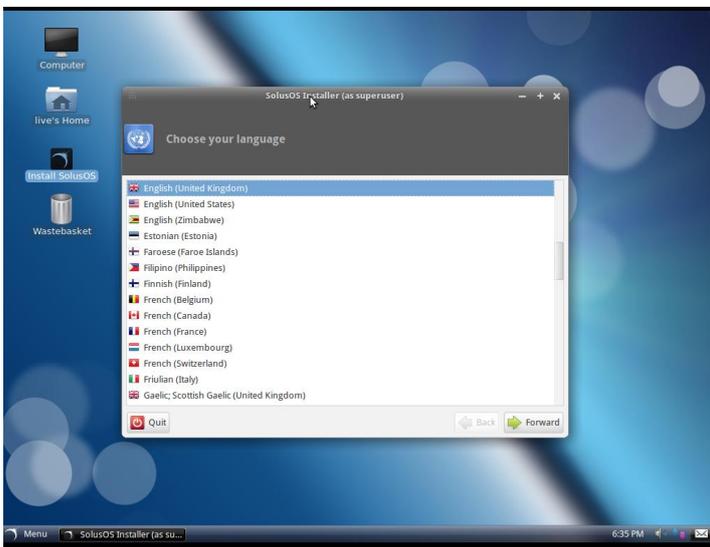
SolusOS 1.1 (Eveline)

Una alternativa clásica

POR JUAN MANUEL DANSA

En mi camino de la búsqueda de una “distro” estable, rápida, actualizada y fácil de instalar, me he encontrado con SolusOS 1.1; la cual es realizada por el creador inicial de LMDE (Linux Mint Debian Edition) Ikey Doherty, quien por motivos personales con respecto a la dirección que estaba tomando LMDE decide crear por su cuenta SolusOS. Una distribución basada en Debian Estable (Squeeze) y que contiene, como importante, repositorios personalizados más los Debian backports, los cuales poseen las últimas versiones de algunas aplicaciones importantes. Cabe aclarar que no es una Rolling Release. Esta versión tendrá soporte mientras lo tenga “Squeeze” por lo tanto nos asegura una estabilidad en el tiempo.

Instalación y primeras impresiones

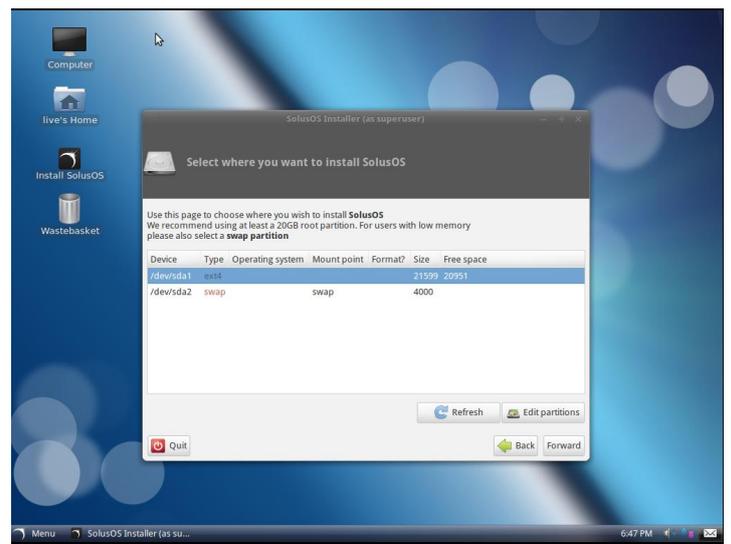


Pantalla de Instalación (selección de Idioma)

La distribución viene en formato LIVE-DVD, la cual pueden descargar del siguiente enlace: <http://solusos.com/download/> tanto en 32 como 64 bits, para esta review he utilizado la versión x64. Pero ¿i por qué el título de clásica!?, porque sencillamente luego de bootear se van ha encontrar con un extremadamente bien logrado Gnome 2.3.x.



El proceso de instalación es fácil he intuitivo pero un solo punto no me terminó de convencer, siempre pensando en el usuario inexperto en sistemas GNU/Linux, y es el particionado del sistema ya que tuve que armar el punto de montaje y la swap a mano, para lo cual se abre la tan conocida aplicación Gparted para realizar esta tarea, convengamos que no es nada conflictivo pero puede alejar a los “newbies” rápidamente.

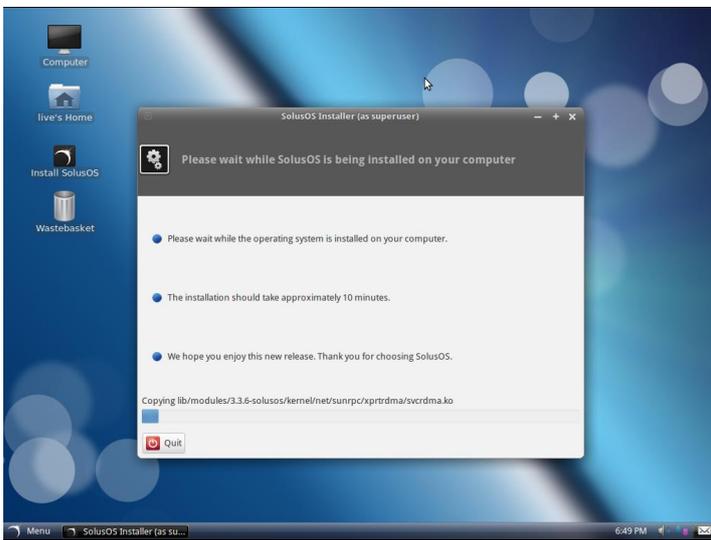


Creando Particiones Básicas

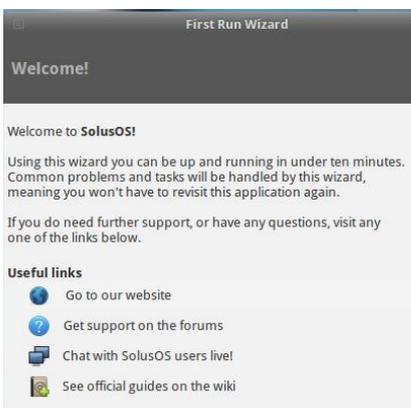
A esto le podemos sumar que, por más que hayamos elegido el idioma de “Cervantes” al principio, el de “Shakespeare” continúa durante toda la instalación hasta su reinicio (por lo menos desde modo Live).

Pasado este paso, el proceso continúa como cualquier distribución estilo Canonical u otra “distro” con instalador gráfico.

Como bien se describe tarda aproximadamente 10 minutos en el proceso, aunque puede variar dependiendo la máquina que posean.



Proceso final de Instalación



Uno de los aspectos que más me gustó, llega luego del reinicio y es el **First Run Wizard**, una aplicación que nos guiará en los primeros pasos de la puesta a punto y utilización del sistema, desde ahí se instalan automáticamente los

controladores gráficos propietarios, más una gran cantidad de códecs multimedia. Ésto redundará en una instalación más rápida y completa sin tener que andar buscando repositorios ni comerse grandes configuraciones que a veces le quitan las ganas a uno de andar instalando este tipo de sistemas.

¡¡Desde este mismo Wizard podremos configurar nuestra conexión de red y hasta el Firewall!! Dándonos un espectro de instalación más que amplio y no siempre presente.



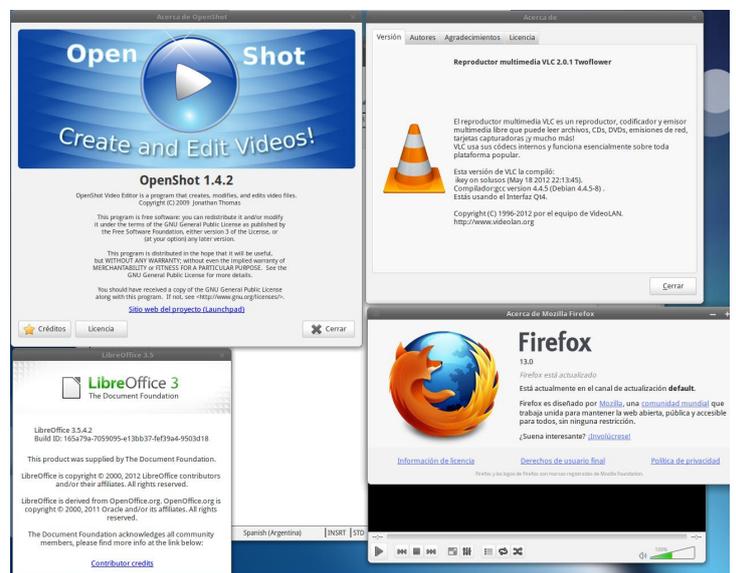
Configuración de Red y Firewall

¡¡Pasado este proceso ya tendremos listo para trabajar nuestro SolusOS 1.1!!

Dentro nos encontramos con:

- LibreOffice 3.5.4.2
- Minitube 1.7
- Flash 11.2.202.233
- PlayOnLinux 4.0.18
- Elementary icon theme.
- FirstRunWizard.
- Compositor para el Escritorio.
- VLC 2.0.1
- Firefox 13 + Thunderbird 12
- Cliente Dropbox
- Kernel 3.3.6
- Gimp 2.8
- Y mucho más...

Siempre recomiendo hacer una actualización completa del sistema una vez instalado, ya sea desde el "Gestor de Actualizaciones" provisto en este caso en la "distro", o desde la terminal con el siguiente comando: `sudo apt-get update && sudo apt-get dist-upgrade -y`.



OpenShot 1.4.2 - VLC 2.0.1 - LibreOffice 3.5.4.2 - Firefox13.0

Requisitos del sistema

- Procesador de 32-bit Edition: x86 PAE
- Procesador de 64-bit Edition: 64-bit
- 512MB RAM
- 3GB de espacio libre en disco
- Resolución de pantalla: 1024×768
- DVD-R/RW ó USB

Cabe destacar que en estos momentos mientras escribo esta "Review" se encuentra a disposición la Alpha 3 Test de SolusOS 2.0, la cual contará con Gnome 3.x basado en Debian "Wheezy" incluyendo gnome-session y gnome-panel lo cual traerá la misma experiencia que Gnome 2.3.x, manteniendo 100% de compatibilidad.

Para mayor información se pueden dirigir a la página del proyecto: <http://www.solusos.com/>

Conclusiones

La verdad que me he encontrado con una distribución más que recomendable, la cual ya forma parte junto con Xubuntu de mis favoritas, en especial para instalar a gente que recién ingresa al mundo GNU/Linux, o a los que no y quieren tener un escritorio rápido, estable y a la última en software; hay muchas otras opciones sin lugar a dudas y de seguro mejores, pero el mundo "TUX" nos da la posibilidad de elegir y eso es lo que mantiene viva a esta gran comunidad... ¡¡la del Software Libre!!

Juan Manuel Dansa (Amonal)

amonal88@gmail.com

twitter: @Amonal_

g+: Amonal Novell



(*) Únete a "Radio Geek", nuestro podcast semanal de actualidad tecnológica
<http://radiogeek.ivoox.com>



Trinity

Entorno de escritorio
Fork de KDE 3.5

POR LOZANO JUAN PABLO

Este entorno de escritorio, es un FORK de KDE 3.5 basado en Qt4. Tiene como objetivo mantener vivo el estilo del escritorio descontinuado KDE3.5, y también pulir algunos detalles estéticos que estaban presentes en KDE 3.5.10. En el camino, nuevas características útiles se han añadido para mantener el entorno hasta al día de hoy. Con ese fin, se han realizado importantes mejoras en áreas como el control de la pantalla, conectividad de red, autenticación de usuarios ¡y mucho más!

Este proyecto NO es una continuación oficial de KDE 3.5 y tampoco será la creación de nuevas versiones de la serie de KDE3. Se trata de un entorno independiente utilizado por una comunidad de desarrolladores.

Nuevo o sensiblemente mejorado

Como se puede apreciar en la siguiente imagen, además de mantener el aspecto estético del entorno de escritorio, también conserva una gran cantidad de software que se incluía junto a KDE 3.5 portado ahora a QT4.



Mejoras en la interfaz

- Añadido un nuevo módulo "monitor y pantalla" en el centro de control para el sistema single/multi monitor y configuración de la pantalla.

- Mejorado el motor de temas GTK Qt, pestañas, casillas de verificación, fondos de menú y mucho más; ahora se muestran correctamente en el estilo de Trinity.
- Una clave opcional segura se ha implementado para proteger aún más los cuadros de diálogo de inicio de sesión y bloqueo de escritorio.
- Un nuevo tema de widget, Asteroid, se ha añadido.
- Construido en el compositor X11, ahora ciertas aplicaciones de Trinity como Amarok, detectan y utilizan la transparencia ARGB siempre y cuando esté disponible.
- Un cliente para TDE específico de notificación para DBUS ha sido añadido para una mayor integración con las aplicaciones comunes, tales como Firefox y NetworkManager.
- Corregido el plugin de Flash en Konqueror.
- Mejorado el QuickLaunch con Kicker applet.

Nuevos programas

Los siguientes nuevos programas se han añadido a la Suite Trinity: kbookreader, kdbusnotification, kmymoney, kstreamripper

¡Mucho más!

Los desarrolladores del proyecto Trinity han puesto en práctica todo su potencial para poder llevar este estupendo entorno de escritorio a diferentes distribuciones Linux como: Debian, Ubuntu, RedHat, Fedora, Slackware, etc. Y cada vez son más los beneficios que se suman a este entorno conforme va creciendo.

Trinity también ofrece imágenes pre-armadas con el entorno para que lo pruebes en modo Live y así poder experimentar sin riesgo el sinfín de posibilidades que brinda. Si estás interesado en probar desde un sistema Live-CD este escritorio, te invitamos a que pases por el siguiente link: <http://apt.pearsoncomputing.net/cdimages/> y si deseas instalar el entorno de escritorio sobre tu distribución Linux actual, explorar cuales distribuciones soporta y las instrucciones a seguir para poder llevarlo a cabo, puedes visitar su web oficial: <http://www.trinitydesktop.org/>

Lozano Juan Pablo
lozanutux@gmail.com
[twitter: @lozanutux](https://twitter.com/lozanutux)



Command & Conquer Tiberium Alliances

Los primeros pasos de Electronic Arts (EA) en GNU/Linux

Más allá de ser una empresa privada, es importante la necesidad de dar a conocer una de las dos propuestas que EA pone a nuestra disposición desde el Centro de Software de Canonical. Esto a futuro podría crear una revolución en el sector de los videojuegos en GNU/Linux, ya que el mismo siempre fue el “Talón de Aquiles” de nuestro querido “TUX”. Si buscamos atentamente, nos encontraremos con dos juegos de esta galardonada firma, Command & Conquer Tiberium Alliances y Lord of Ultima; en esta entrega me decantaré por el primero ya que soy un seguidor de la saga desde sus comienzos.



Antes que nada, hay que aclarar que este es un juego 100% on-line, es decir que no se instala ningún componente en la máquina; desde el Centro de Software de Canonical se nos instalará un acceso directo con su icono en el menú juegos, pero no es necesario esto ya que ingresando a <http://alliances.commandandconquer.com> conseguiremos el mismo objetivo. Antes de comenzar a jugar hay que crear una cuenta de usuario, si es que no la poseemos. Esta edición sigue fielmente la historia de la saga, la lucha por la supremacía y el preciado “Tiberium”: mineral radioactivo en forma de cristal proveniente de otra galaxia.

Es básicamente un juego de estrategia, donde podemos elegir entre dos bandos, GDI (Global Defense Initiative) o Brotherhood of the NOD, tendremos que recolectar recursos, crear alianzas, ampliar nuestro territorio, adquirir y actualizar nuevas armas, y un sinfín de opciones. Una de



las ventajas, es que se puede seguir el juego desde cualquier parte y en cualquier momento ya que por más que no estemos logueados, el mismo seguirá produciendo constantemente y hasta podremos ser atacados.



Dentro de las opciones del juego cabe destacar la posibilidad de habilitar el modo WebGL, el cual nos proporciona, si poseemos una placa gráfica aceleradora, un rendimiento y visual increíble; se encuentra en estado experimental pero funciona perfecto y aumenta notablemente el rendimiento, igualmente al ingresar al juego se nos pregunta si queremos activarlo o arrancar sin él. Recomiendo tener el navegador actualizado a la última versión; aunque les diría que utilicen Google Chrome ya que en Firefox no funciona como debería, haciendo que el juego se torne pesado.

En conclusión estamos ante un juego muy dinámico acompañado de contenido social aceptable, con una línea de aprendizaje normal, ya que dedicándole unas horas o menos se comprende como jugar, hacerse de amigos en alianzas y combatir por la supremacía del territorio.

“The Rivers will flow with the blood of those who oppose us” Kane. Leader Brotherhood of the NOD

Juan Manuel Dansa (Amonal)
amonal88@gmail.com
twitter: @Amonal_
g+: Amonal Novell

TUX **INFO**
WWW.TUXINFO.COM.AR