

Guía APT

(y otras formas de instalar aplicaciones en GNU/Linux)

Una copia gratuita de este documento puede obtenerse en la sgte. Dirección:

<http://stoneheads.wordpress.com/2009/09/29/guia-apt/>

o contactando directamente con el autor del mismo en el correo indicado.

Autor: Carlos Felipe Alvarez González

carlos_felipe@asturlinux.org

Juan Antonio Silva Villar

juan@asturlinux.org

Fecha: 29 de Septiembre de 2009

Título: guia-apt

Versión: 1.0

Copyright © 2009 Carlos Felipe Alvarez González

© 2004 Juan Antonio Silva Villar,

Este documento está protegido bajo los términos de la GNU FDL (Free Documentation License). Ha sido escrito con la esperanza de que sea de utilidad a la comunidad pero no viene con garantía alguna, úselo bajo su propio riesgo.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, with no Front-Cover Texts, and with no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

Se concede permiso para copiar, distribuir y/o modificar este documento bajo los términos de la Licencia de Documentación Libre de GNU, Versión 1.2 o cualquier otra versión posterior publicada por la Free Software Foundation; sin Secciones Invariantes, sin Textos de Cubierta Delantera ni Textos de Cubierta Trasera. Una copia de la licencia está incluida en la sección titulada "GNU Free Documentation License". (Puede encontrarse una traducción no oficial de esta licencia al castellano en <http://curso-sobre.berlios.de/gfdles/gfdles.html>)

Esta documentación se publica bajo la Cláusula Mariposa. Una copia de la cláusula está incluida en la sección titulada "Cláusula Mariposa/Butterfly Clause". Para más información, visite <http://www.clausulamariposa.es>

This document is released under the Butterfly Clause Agreement. A copy of the clause is included in the section entitled "Cláusula Mariposa/Butterfly Clause". For more information visit: <http://www.butterflyclause.org>

Condiciones de uso

Esta obra puede copiarse, distribuirse, comunicarse públicamente, modificarse y hacer obras derivadas de ella bajo ciertas condiciones. Aquí se enumeran las principales, pero ante cualquier duda, **para ver una explicación exhaustiva** y precisa de las mismas, **consulte el texto de la licencia.** Tenga en cuenta que algunas de estas condiciones pueden no aplicarse si se obtiene el permiso del titular de los derechos de autor.

- **Reconocimiento:** Debe reconocer los créditos de la obra. Si se limita a copiarla debe conservar la misma portada y las notas de copyright. Si hace modificaciones u obras derivadas, debe cambiar el título, mencionar como autor a quien haya hecho las modificaciones u obras derivadas y reconocer explícitamente la obra original de la que deriva, indicando el título, autor, versión, fecha y el URI (dirección de internet), si existe, del original, en la sección titulada Historia (y en caso de que no la haya, crearla). En dicha sección deberán aparecer también al menos el título, los autores, la fecha y el editor (si existe) de la versión modificada. No sugiera que el autor original le da su apoyo o apoya el uso que hace de su obra.
- **Comercial.** Si lo desea, puede utilizar esta obra para fines comerciales.
- **Compartir bajo la misma licencia.** Tanto si copia como si altera o transforma esta obra, o genera una obra derivada, sólo puede distribuir la obra generada o copiada bajo una licencia idéntica. La licencia GNU FDL, las cláusulas y limitaciones de garantía y las notas que indican que dicha licencia y cláusulas se aplican al documento deben reproducirse en todas las copias.
- **Cláusula Mariposa.** El licenciador excluye cualquier uso de la información contenida en este documento en las Administraciones Públicas del Estado adquirente en los supuestos de que el Estado proceda a efectuar un acto de agresión armada contra un tercer Estado o proceda a efectuar actos constitutivos de genocidio, crímenes de lesa humanidad o crímenes de guerra.

Por supuesto, **se agradece cualquier comentario, crítica, corrección, aportación, modificación**, etc... que se haga con ánimo **constructivo**.

Índice de contenido

- Introducción.....4
- Prerrequisitos.....4
- Casos de uso (o para qué me sirve esto).....4
- * Formas de instalación *.....5
- Instalación por repositorios: Uso de apt, aptitude y synaptic.....6
 - * El APT. Uso básico. *.....6
 - Los repositorios (fuentes del apt). Instrucciones de apt.....6*
 - Configurar y requeconfigure paquetes.....9*
 - El APT. Uso avanzado.....11
 - Los repositorios en detalle.....11*
 - Búsqueda avanzada de paquetes.....14*
 - Experimentando con gaseosa.....15*
 - Mezclando tipos de distribución.....15*
 - Instalando una versión concreta de un paquete.....16*
 - Creando un repositorio local.....16*
 - Tener un sistema con los mismos paquetes que otro.....17*
 - APT seguro.....18*
 - Soluciones a algunos problemas.....19*
 - * APTITUDE *.....20
 - * SYNAPTIC *.....21
- * Instalación por binarios. *.....22
 - Comprobación sumas MD5.....23
- Backports.....24
- Compilación.....25
- Historia:.....27
- Anexo I. Licencia del documento: GNU Free Documentation License.....28
- Anexo II. Cláusula Mariposa/Butterfly Clause32

Introducción

En este documento **se explica de manera sencilla la instalación de aplicaciones en un entorno GNU/Linux**. Particularmente, **se centra en el manejo del APT, aunque también toca los temas de instalación mediante binarios y de compilación a partir del código fuente**. El APT es un sistema de manejo de paquetes introducido por la distribución **Debian**, aunque hoy en día ha sido adoptado/incluido por otras muchas distribuciones, entre ellas, por supuesto, las derivadas de Debian, algunas muy extendidas, como Ubuntu.

En la actualidad, la instalación de aplicaciones es bastante fácil y transparente al usuario en prácticamente todas las distribuciones: un menú gráfico donde aparecen las aplicaciones disponibles y las ya instaladas; el usuario simplemente selecciona las que quiere instalar/desinstalar y listo. **Sin embargo, lo que hay por debajo es el sistema predeterminado de manejo de paquetes de la distribución** (apt, en muchos casos).

Entonces conocer el sistema de manejo de paquetes nos dará no sólo un mayor control y una visión más profunda de cómo funciona nuestro sistema, sino también nos ayudará a solucionar los posibles problemas que puedan surgir y a poder exprimir al máximo las capacidades de nuestro sistema, haciéndolo aún más flexible.

Inicialmente fue escrito tomando como base la distribución Debian, pero es perfectamente aplicable a cualquier otra que use el sistema de paquetería apt. La instalación de binarios y la compilación son comunes a cualquier sistema GNU/Linux.

Que lo disfrutéis.

Prerrequisitos

- **Suponemos al lector unos conocimientos básicos de manejo de GNU/Linux.**
- **El documento está pensado tanto para usuarios que se inician en el mundo GNU/Linux y quieren ir un poco más allá como para usuarios experimentados** que quieran profundizar en el manejo del sistema de paquetes. Para que sea usable por ambos grupos, se ha dividido de una forma elemental: **los capítulos y secciones que sean esenciales para un usuario inicial tienen un asterisco antes y después de su nombre, * tal que así ***. Verdaderamente, a un principiante en el tema le bastaría con leer sólo estos temas y más adelante si quiere profundizar, el resto. Si un capítulo, sección, etc... está marcado de esta forma, esto también se aplica a todos sus apartados, subsecciones...
- **Nomenclatura:** Las instrucciones a teclear en la línea de comandos tienen un tipo de letra distinto (estilo consola) y van en negrita. Normalmente hay que escribir literalmente lo que aparece, excepto lo que va entre <> que indica que esa parte (caracteres < y > incluidos) se sustituye por otra cosa.

Casos de uso (o para qué me sirve esto)

- Aprender a instalar y desinstalar aplicaciones en GNU/Linux.
- Conocer en profundidad una parte tan importante de nuestro sistema como es la gestión de paquetes.
- Mantener nuestro sistema actualizado.
- Solucionar posibles errores en la instalación, actualización y configuración de aplicaciones.
- Controlar a fondo qué tipo de sistema queremos tener instalado, pudiendo tener aplicaciones de distintos repositorios e incluso de otros tipos de distribución.
- Obtener aplicaciones no disponibles en los repositorios oficiales de la distribución.
- Obtener las versiones más actuales de aplicaciones, utilidades y drivers sin comprometer la estabilidad del sistema ni afectar al resto de aplicaciones.

* Formas de instalación *

En GNU/Linux tenemos cuatro maneras principales de instalar aplicaciones. Más adelante trataremos cada una de ellas con más detalle:

- **Por repositorios:** Esta manera suele ser específica de cada distribución (o sea, que aunque el sistema es el mismo, la forma de hacerlo cambia de una distribución a otra). **Es la manera predeterminada y recomendada**, salvo que la aplicación concreta que queramos (o una versión más actual) no esté disponible, en cuyo caso podemos probar alguno de los otros métodos. De momento basta con saber que **se trata de bajar automáticamente las aplicaciones de unos servidores de Internet dados** (los predeterminados de la distribución o los que nosotros especifiquemos); y no, **no hay que pagar por ello ni registrarse** ni nada parecido y **podemos escoger los repositorios que queramos** (además de los oficiales de cada distribución).
- **Binarios** que se instalan directamente: Fácil; se busca el programa en Internet, se descarga y se instala según las instrucciones que nos indiquen. Normalmente los pasos son descomprimirlo y ejecutar un script de instalación.
- **Backports:** Se trata de instalar paquetes sueltos (los paquetes son, por así decir, como los “ladrillos” que construyen las aplicaciones) y puede usarse incluso con paquetes de otras distribuciones. Básicamente es una especie de parcheado de la primera manera y aunque no es complicada, no es recomendable para usuarios novatos.
- **Compilar** las fuentes: Es similar a la segunda manera, sólo que aquí en lugar de bajarnos el programa en sí, obtenemos el código fuente del mismo (las instrucciones en un lenguaje de programación determinado que forman el programa) y tenemos que compilarlo (construir el programa a partir del fuente) antes de instalarlo. Esto suena complicado, pero en realidad en la mayoría de los casos es sota, caballo y rey: siempre se hace igual y en muy pocos pasos. De todas formas es algo que normalmente un usuario medio no necesitará hacer.

Comentar por último, que en el caso de Debian los paquetes vienen comprimidos en un tipo especial de archivo binario de nombre `<nombre_del_paquete>_<versión>-<revisión>_<arquitectura>` y la extensión `.deb`, por ejemplo:

tuxkart_0.4.0-4.1_i386.deb

indica que dicho archivo es el instalable del paquete tuxkart (un juego de coches 3D) 0.4.0 revisión 4.1 para la arquitectura (tipo de procesador) i386 (un PC normal, vaya). No preocuparse, si usamos únicamente el sistema de repositorios sólo necesitaremos conocer el nombre del paquete y nada más.

Instalación por repositorios: Uso de apt, aptitude y synaptic

Apt es la herramienta básica para instalar aplicaciones en Debian y conviene conocerla, aptitude es la versión avanzada de la misma y la que los desarrolladores de Debian recomiendan usar, y synaptic una interfaz gráfica para hacer lo mismo. Nuestro consejo es que conozcáis, aunque sea por encima, las tres y uséis aquella con la que os encontréis más cómodos.

ADVERTENCIA: Eso sí, **procurad no mezclar el uso de apt y aptitude**, para evitar que el sistema se haga un lío: si usáis normalmente apt para instalar/desinstalar, no uséis aptitude y viceversa.

ADVERTENCIA2: **En todos los casos, para instalar/desinstalar aplicaciones y modificar los archivos de configuración, necesitamos hacerlo como root.**

*** El APT. Uso básico. ***

El APT es una herramienta para instalar, desinstalar, buscar, actualizar... paquetes de software. Un paquete puede ser un programa, librerías, documentación, etc. Debian es la distribución pionera en desarrollar herramientas para la gestión de paquetes. La primera de estas herramientas fue el dpkg. El APT es un programa de gestión avanzada de paquetes más potente que el dpkg.

Una de las cualidades más interesante del APT es que calcula las dependencias entre paquetes. ¿Qué significa esto?. Por ejemplo: el gestor de ventanas Window Maker viene en el paquete de nombre wmaker. Sin embargo para que este gestor de ventanas funcione se necesita tener instalados los paquetes de nombre: libungif4g, libwraster2 y libxaw7. Pues bien, nosotros sólo tenemos que instalar el paquete wmaker, el APT se dará cuenta que este paquete depende de los anteriores y los instalará también. Esto nos libraré de quebraderos de cabeza como que para que este programa funcione se necesita tal o cual librería; el APT se encarga de todo.

Los repositorios (fuentes del apt). Instrucciones de apt

Existe un archivo de configuración del APT, conocido popularmente como las "fuentes del APT". Ese archivo es: `/etc/apt/sources.list` En ese archivo están escritas **todas las direcciones de Internet (también llamadas "repositorios") y opcionalmente, el nombre de los CD's o DVD's de donde vamos a obtener los paquetes** (aunque esto último está cada vez más en desuso, es muy conveniente si no disponéis de una buena conexión a Internet). APT anota los paquetes que hay en cada una de esas "fuentes" (direcciones de Internet y CD's) y crea una base de datos. Así, cada vez que queramos instalar un paquete, el APT sabrá dónde está la versión más actualizada del mismo (la mayoría de las veces en Internet) y lo instalará. Si no tenemos Internet en el momento de la instalación de ese paquete, lo instalará de uno de los CD/DVD de la distribución.

La herramienta APT consta de varias instrucciones. Vemos algunas brevemente y luego veremos ejemplos de uso.

apt-get install <nombre_del_paquete>

Instala el paquete: nombre_del_paquete

apt-get update

Actualiza la lista de paquetes.

apt-get upgrade <nombre_del_paquete>

Actualiza el paquete: nombre_del_paquete

apt-get dist-upgrade

Actualiza la distribución de Debian GNU/Linux. Es decir, actualiza TODOS los paquetes de la distribución.

apt-get remove <nombre_del_paquete>

Desinstala el paquete: nombre_del_paquete.

apt-get clean

Borra totalmente el repositorio local (/var/cache/apt/archives) que contiene los ficheros .deb descargados.

apt-get autoclean

Igual que la anterior, pero sólo borrará aquellos paquetes que ya no pueden ser descargados, o son claramente inservibles.

apt-cdrom add

Añade un CD/DVD de la distribución de Debian a la lista de repositorios.

apt-cache search <palabra/s clave/s>

Busca paquetes que tengan que ver con la/s palabra/s clave/s

apt-cache show <nombre_del_paquete>

Muestra información sobre el paquete (versión, paquetes de los que depende, para qué sirve, etc...)

EJEMPLO 1: Actualizar la lista de fuentes (los repositorios).

Bueno, lo primero de todo es que APT cree la base de datos con todos los paquetes que tenemos en los CD/DVD's de la distribución. Si tenemos una buena conexión de Internet podemos pasar de esto y bajarnos los paquetes directamente de la red. Sea como fuere, para ello podemos editar a mano el archivo /etc/apt/sources.list o hacer uso de las instrucciones "ad hoc" del APT:

apt-cdrom add

Entonces nos aparecerá un mensajito en pantalla pidiéndonos que insertemos un DVD de la distribución Debian. Repetiremos este proceso con todos ellos. Si recordáis, este paso es exactamente el mismo que el programa de instalación de Debian nos proponía cuando estábamos instalando el sistema operativo. Bueno, pues si no lo hicisteis antes, éste es el momento.

Si lo que queremos es bajarnos los paquetes de Internet, tenemos una copia "especial" del sources.list en el directorio: /usr/share/doc/apt/examples/sources.list Esta copia tiene anotadas un montón de direcciones de Internet de donde podemos bajarnos los paquetes. Sólo tendríamos que copiarla al directorio /etc/apt/. ¿Qué? ¿qué no sabes cómo se copia en GNU/Linux? Pues deberías, pero bueno, ahí te va:

```
cp /usr/share/doc/apt/examples/sources.list /etc/apt/
```

Hagamos lo que hagamos, después de escribir en el archivo de las fuentes el nombre de nuestros CD/DVD's o de las direcciones de Internet, **tenemos SIEMPRE que actualizarlo**. Para actualizar este archivo escribimos:

```
apt-get update
```

Este último paso es absolutamente imprescindible. Si queremos bajarnos los paquetes de Internet, debemos estar conectados al hacer este paso.

EJEMPLO 2: Buscar e instalar un paquete.

Supongamos que quiero instalar un programa para reproducir mp3. Pues empezáramos buscando los paquetes que tengan que ver con el tema escribiendo:

```
apt-cache search mp3
```

Y nos aparecería una lista enorme de programas en pantalla. Todos esos paquetes que aparecen en pantalla tienen asociada una pequeña explicación de lo que son y lo que hacen. Podemos afinar la búsqueda cambiando la palabra clave por mp3player, mp3-player, mp3 player, etc...

Bueno, uno de esos muchos programas que nos aparecen es el noatun que según la explicación que le acompaña no sólo reproduce mp3, sino que también reproduce mpeg. Supongamos que ese es el programa que nos hace tilín y lo queremos instalar. Pues nada más fácil, tecleamos:

```
apt-get install noatun
```

Y se empieza a instalar el programa y todos los paquetes con librerías, etc, que este programa necesita para funcionar.

Si ponemos varias palabras detrás de search, nos mostrará los paquetes que contengan ambas, así:

```
apt-cache search pdf kde
```

Si ponemos comillas en las palabras, buscará la expresión exacta:

```
apt-cache search "mp3 player"
```

Por cierto, no distingue entre mayúsculas y minúsculas y las palabras hay que ponerlas en inglés, me temo. **La búsqueda la hace tanto en el nombre como en la descripción del paquete, si queremos que sólo busque en el nombre, podemos usar la opción -n.**

Si de alguna forma se daña un paquete instalado o simplemente deseamos **reinstalar la versión más reciente** de un paquete podemos hacer:

```
apt-get --reinstall install <nombre_del_paquete>
```

A veces, aparte de cómodo, puede ser útil desinstalar un paquete a la vez que instalamos otro y viceversa:

```
apt-get install paquete1 paquete2-
```

Aquí instalaríamos paquete1 a la vez que desinstalamos paquete2.

EJEMPLO 3: Desinstalar un paquete.

Supongamos que después de probar el noatun resulta que no me gusta nada. Bueno, sin problemas, ejecuto:

apt-get remove noatun

La instrucción de desinstalar paquetes tiene una opción muy interesante:

apt-get remove --purge <nombre_del_paquete>

¿Y para que sirve? Bueno, la mayoría de los programas de GNU/Linux tienen asociado uno o varios archivos de configuración. Normalmente la labor más pesada que un usuario de GNU/Linux tiene que hacer es la de editar estos archivos de configuración. Imagínate que un día necesitas espacio en tu disco duro, y no te queda más remedio que quitar algún programa para ganar sitio. Esto te puede dar muuucha pereza, sobre todo si tenemos en cuenta que, si alguna vez vuelves a instalar ese programa, tendrás que volver a editar el susodicho archivo de configuración. Pues que no te dé pereza ninguna, que para eso se inventó el APT. Si tecleas:

apt-get remove <nombre_del_paquete> se te borrará el programa, pero no el/los archivo/s de configuración. De esta forma si alguna vez quieres volver a instalar este programa, no tendrás que volver a editar el archivo de configuración, conservas el que habías editado en su día.

Ahora bien, si quieres borrar el programa por completo porque estás seguro de que nunca más lo vas a usar, puedes teclear:

apt-get remove --purge <nombre_del_paquete>

y se os borrará el programa y los archivos de configuración del mismo. Recordad que los archivos de configuración son normalmente scripts u otros archivos de texto plano; vamos, que no ocupan casi nada.

EJEMPLO 4: Manteniendo el sistema actualizado

Tarea cotidiana. Las distribuciones modernas se actualizan (o al menos nos avisan) ellas solitas, pero si no, podemos hacer como root:

apt-get update
apt-get dist-upgrade -u

Y listo. La -u es para que nos muestre los paquetes que se van a actualizar. Si tenemos una conexión lenta u ocurre algún problema a la hora de bajarse los paquetes, podemos salir pulsando CTRL+C y luego volver a ejecutar el comando otro día y seguirá automáticamente en el punto en el que lo dejamos.

Configurar y reconfigurar paquetes.

Hasta ahora vimos que instalar un paquete es algo tan fácil como teclear:

apt-get install <nombre_del_paquete>

Pues bueno, hay que advertir una cosa. Hay muchos paquetes que al instalarlos requieren ser configurados. Para ello, después de ejecutar **apt-get install <nombre_del_paquete>**, el sistema nos irá pidiendo cierta información. El sistema nos sugerirá una opción por defecto, que es la que aparece en pantalla, por lo que, si en el momento de la instalación no sabemos qué responder, siempre podemos aceptar la opción por defecto y reconfigurar el paquete más tarde con la instrucción:

dpkg-reconfigure <nombre_del_paquete>

Aquí podemos ver un uso de la herramienta dpkg. No será el único, tiempo al tiempo.

Un ejemplo típico de paquete que requiere configuración es el xserver-xorg. Este paquete genera el archivo /etc/X11/xorg.conf que es el archivo que tiene la información sobre nuestro monitor, tarjeta de vídeo, resolución de pantalla, teclado y ratón para ser usados con los interfaces de ventanas. Cuando lo instalamos nos irá pidiendo sucesivamente toda esa información (refresco

del monitor, resolución de pantalla, marca de la tarjeta de vídeo, conector de ratón, configuración del teclado).

Otro paquete que nos puede interesar reconfigurar es el etherconf, que establece la configuración de la tarjeta de red (dirección IP, puerta de enlace, máscara de red, etc..).

Hay otros paquetes que al instalarlos nos muestran una ventana con algún mensaje de advertencia, algún recordatorio, etc. En esos casos no hay que configurar nada ni teclear nada, sencillamente aceptamos el mensaje y el proceso de instalación sigue normalmente. Es recomendable leer esos mensajes, suelen dar información útil, como en qué directorio están el/los archivo/s de configuración de dicho paquete.

El APT. Uso avanzado.

Los repositorios en detalle.

Si habéis echado un vistazo al **archivo sources.list**, tal vez os hayáis sentido un poco confusos ¿qué significan todas esas líneas? Pues bien, ha llegado el momento de explicarlo. Veamos una entrada de ejemplo:

```
deb http://ftp.es.debian.org/debian/ lenny main contrib non-free
```

En este archivo, como ya explicamos antes, **cada línea es una fuente (o repositorio) de paquetes para nuestro sistema**. La primera palabra "**deb**" indica que son paquetes normales, o sea, para instalar directamente, también podría ser "deb-src" en cuyo caso indicaría que los paquetes contienen no la utilidad o aplicación en sí, sino su código fuente. De momento, sólo nos interesan los "deb".

A continuación viene una dirección de un servidor de Internet donde están los paquetes. Normalmente, como todas las direcciones comienza por http://, que indica que se accede al servidor mediante el protocolo http, que es el que se usa habitualmente para acceder a webs, pero también podría comenzar por ftp://, para indicar que se accede mediante el protocolo ftp, especializado en transferencias de archivos. Generalmente, por razones de rendimiento, es preferible éste último, pero esto no es algo matemático, depende de cada servidor en concreto. De todas formas, si vamos a añadir un repositorio nuevo, en el caso de que nos den varias opciones ya nos dirán cuál es la más idónea.

Lo siguiente que aparece es la dirección en sí del servidor (varias palabras separadas por puntos) y la ruta o directorio dentro del servidor donde se encuentran los paquetes (aquí el separador es una barra inclinada a la dcha.)

Hasta aquí no hay mucho donde meter mano, hay que poner lo que nos es dado y punto. Únicamente comentar que en los repositorios oficiales de Debian, suelen funcionar mejor los que se encuentran fuera de España. Por ejemplo, el repositorio mencionado es el oficial para España (.es), pues bien, si alguna vez nos da problemas, basta cambiarlo por el de Alemania (.de) y seguramente vaya mejor, o sea poner:

```
deb http://ftp.de.debian.org/debian/ lenny main contrib non-free
```

Por si a alguien le entra la duda, **en todos los repositorios oficiales de Debian, hay exactamente los mismos paquetes, son sólo imágenes** ("mirrors", copias) del central. Su utilidad es evitar saturaciones y problemas si alguno se cae.

Ahora empieza lo interesante. **A continuación aparecen una serie de palabras** (separadas por un espacio en blanco) **que indican qué paquetes exactamente vamos a seleccionar del repositorio. Esto es MUY importante, ya que determinará cómo será nuestro sistema. La primera palabra es la más importante porque indica el "tipo" (o rama) de Debian que tendremos**. Actualmente Debian nos ofrece varios tipos (ramas) de distribución y depende de nosotros escoger una u otra:

stable: Es la distribución estable y oficial, **la más probada y normalmente la más recomendada para servidores, y para aquellos usuarios a los que no les importe no usar lo más nuevo en cuanto a aplicaciones y/o programas, a cambio de una mayor estabilidad. Es la mejor opción para usuarios novatos**. En vez de "stable" también puede aparecer el nombre oficial (ya sabemos que Debian saca sus nombres de los personajes de la saga "Toy Story"). Por ejemplo, la distribución estable actual (a la hora de escribir estas líneas) lleva el nombre de "lenny", así pues es lo mismo poner lenny que stable ¿no?. Pues no, hay una diferencia sutil pero importante: la distribución estable va cambiando de nombre (woody, sarge, etch, lenny, etc...) según van saliendo versiones actualizadas de Debian, pero si ponemos "stable" SIEMPRE se referirá a la estable actual. Esto implica que cuando salga la nueva versión estable nuestro sistema se actualizará automáticamente (hablaremos de esto más adelante), lo cual es una de las ventajas de Debian, pero puede que no a todo el mundo le haga gracia; puede que queramos elegir nosotros cómo y cuándo vamos a actualizar la distribución, y aunque la

actualización es bastante segura, si hay muchas novedades y cambios siempre hay un cierto riesgo de que nos dejen de funcionar algunas cosas o cambie alguna configuración. Entonces si esto no nos importa pues dejamos "stable", si somos del tipo de personas mencionado o no sabemos, mejor poner el nombre. Por último, tened en cuenta que "stable" proporciona actualizaciones de seguridad de las aplicaciones instaladas, pero no actualizaciones de versión, o sea que si sale una nueva versión de la aplicación tal, nosotros seguiremos con la anterior. Esto no es tan malo como parece ya que recordemos que "stable" prima la estabilidad y seguridad sobre otras consideraciones y que existen otras formas de actualizar la versión de las aplicaciones sin cambiar de rama.

testing: Distribución en fase de pruebas y la futura versión estable. Tanto los paquetes como las aplicaciones **están suficientemente probadas y su estabilidad es grande, pero no está tan garantizada como la de "stable"**. Si llevamos un tiempo usando Debian y necesitamos o **queremos tener las últimas versiones de los programas, ésta es nuestra distribución**. Ojo, que aquí también se aplica lo que dijimos en la "stable" respecto al nombre: una distribución es "bautizada" cuando entra en la fase "testing" y según se va comprobando, pasa a la "stable", pero mantiene su nombre. En esta rama, además de actualizaciones de seguridad (aunque no tan exhaustivas como en "stable"), también nos actualiza la versión de las aplicaciones.

unstable: También conocida como "sid". En este caso siempre se llama igual. Distribución inestable, con lo más nuevo y normalmente menos recomendado por cuestiones de estabilidad no suficiente probada. **Para los inquietos y aventureros.**

experimental: Esto no es una distribución para usar, sino para probar y ayudar a corregir fallos. Sólo para "geeks" y gente que quiera colaborar con Debian.

oldstable: Cuando una distribución estable es reemplazada por otra, la antigua pasa a denominarse así, aunque lo más común es dejar el nombre "de pila" oficial (como sarge, woody...) Durante un año aproximadamente, se siguen proporcionando actualizaciones de seguridad y después desaparece de los repositorios oficiales y pasa a quedar "aparcada" en lo que se conoce como los archivos de Debian. Esto puede ser importante si hemos decidido no actualizar la distribución (porque nuestro hardware ya no da para más y no lo vamos a cambiar, p.e.); tras el periodo de gracia, necesitaremos cambiar los repositorios que usemos y apuntar al archivo de Debian, si queremos seguir disponiendo de los paquetes de nuestra versión (cosa muy interesante si, por ejemplo, nos cargamos alguno por error). Supongamos que usamos la rama estable de "etch" y no queremos cambiarnos a la nueva versión; llegará un momento en que nuestros repositorios dejen de funcionar y tengamos que usar únicamente el repositorio siguiente:

```
deb http://archive.debian.org/debian-archive/debian/ etch main contrib non-free
```

Si en vez de etch estamos usando sarge, pondríamos "sarge" donde pone "etch" y así sucesivamente.

Por si no ha quedado claro **el sistema de nombrado**, resumimos: una versión de Debian es "bautizada" cuando está en la fase testing, luego pasa a estable, después a oldstable y por último se queda archivada. A lo largo de su ciclo de vida, cada vez va cambiando menos hasta quedar por fin "fossilizada" y sin soporte. Así pues si escogemos para nuestros repositorios una versión con su "nombre de pila", esa sería, en cierta medida, la opción conservadora. Si ponemos el nombre de la fase (del momento en su ciclo de vida), tendremos un sistema siempre en evolución, aunque el ritmo de la misma dependerá de qué fase escojamos.

Lo mejor es que una vez hayamos elegido nuestro tipo de distribución, nos atengamos a él en TODOS los repositorios. No es muy conveniente andar mezclando tipos. Puede hacerse (y más adelante explicaremos cómo) pero ten en cuenta que: tiene que hacerse con cuidado y a no ser que seas un usuario experimentado no es algo que necesites ni quieras hacer. ¿Que con qué tipo nos quedamos? Si eres nuevo en Debian o en el mundo Linux, escoge "stable", y si ya tienes experiencia al respecto y las versiones de las aplicaciones que vienen en "stable" no cubren tus necesidades, usa "testing".

Las palabras que siguen al tipo de distribución indican lo siguiente: Debian sigue una política muy estricta respecto a las licencias de los programas que incluye, estas palabras **hacen referencia a esa política de licencias**. Pueden ser hasta cuatro, y podemos poner cualquier combinación de las mismas, aunque normalmente, "main" siempre estará presente:

main: Son los paquetes principales de la distribución. Los paquetes incluidos en main, además de cumplir con las normas DFSG (Debian Free Software Guidelines), no dependen de ningún otro paquete que esté fuera de main.

contrib: Los paquetes de contrib también deberán cumplir la DFSG, pero dependen para su compilación o ejecución de algún otro paquete que esté en non-free.

non-free: Paquetes que no son software libre. Los paquetes en non-free puede que no cumplan estrictamente la DFSG, ya sea por usar algoritmos patentados, por su tipo de licencia, o por otra cuestión legal.

non-us: Actualmente en desuso, antes contenía paquetes que no podían ser legalmente alojados en servidores de EEUU. Gracias a cambios en la legislación de ese país, esto ya no es necesario.

Aparte de los repositorios oficiales de Debian podemos añadir otros que no sean oficiales y que contengan aplicaciones que no estén incluidas en nuestra distribución o versiones más actuales. Normalmente vienen indicados en las webs de dichas aplicaciones. Esto tiene como ventaja respecto a instalarlos "a pelo" que podremos manejar la aplicación con el APT igual que cualquier otro paquete y además se nos actualizará automáticamente. Como inconveniente, que puede que esa aplicación no esté pensada para nuestro tipo de distribución y nos salgan un montón de errores al intentar instalarla indicando que necesitamos una versión distinta de varios paquetes. En estos casos lo mejor suele ser no calentarse la cabeza y renunciar a instalarla, al menos por este medio ya que podemos armar un buen estropicio si intentamos mezclar paquetes de un tipo de distribución con otra.

Para terminar, decir que como en todos los archivos de configuración de GNU/Linux, un carácter "#" indica que todo lo que viene detrás en la misma línea es un comentario. Esto nos será muy útil por si queremos quitar algún repositorio que ya no nos interese o que no nos podamos conectar a él, mejor que borrarlo es comentarlo (o sea, indicamos al sistema que ya no lo vamos a usar), por si alguna vez lo volvemos a necesitar o se arreglan los problemas de conexión. Tal que así:

```
#deb http://ftp.es.debian.org/debian/ lenny main contrib non-free
```

También es muy recomendable poner comentarios si añadimos algún repositorio por nuestra cuenta (obtenido de Internet, p.e.) para un paquete o paquetes en concreto, para que no se nos olvide su utilidad:

```
#Repositorio para la última versión del paquete tal y cual  
deb .....
```

Búsqueda avanzada de paquetes

dpkg -l | more

Ver una **lista ordenada alfabéticamente de todos los paquetes del sistema** con estado, nombre, versión y descripción. El estado se indica con 3 caracteres cuyo significado podemos ver al principio de la salida del comando.

Si tenemos una idea del nombre del paquete pero no estamos seguros:

dpkg -l "*office"

Buscamos cualquier paquete que tenga la palabra "office" al final de su nombre. La opción es una L minúscula. No olvidarse de las comillas. El asterisco actúa de comodín. Tened en cuenta que el dpkg sólo busca entre los paquetes instalados o la descripción de los mismos en el cache.

Ver más información sobre un paquete:

apt-cache show kpdf
apt-cache depends kpdf
apt-cache showpkg kpdf

Con "show" nos muestra una información bastante completa del paquete, con "depends" nos muestra únicamente (pero de forma más clara) qué paquetes son necesarios para que funcione el paquete kpdf (Depends), qué paquetes nos pueden venir bien que tengamos instalados (Suggest), con qué paquetes puede haber conflictos (Conflicts) y a qué paquetes reemplaza (Replaces). "showpkg" es similar al anterior pero además nos muestra las versiones de los paquetes y los paquetes que dependen de él (reverse dependencies).

Notar que con "show", si el paquete estuviera instalado y hubiera una versión más reciente, aparecería información de ambas versiones, primero la disponible y luego la que está instalada.

Si queremos ver de forma sencilla **qué versión de un paquete tenemos instalada y si hay una más reciente** (es actualizable) una herramienta que nos puede servir de ayuda es apt-show-versions. Para instalarla hacemos:

apt-get install apt-show-versions

Uso:

apt-show-versions -p <nombre_del_paquete>

Nos dirá a qué rama pertenece el paquete, su número de versión y si ya está actualizado (uptodate) o hay una versión más reciente (upgradable). Con

apt-show-versions

nos mostrará información de todos los paquetes y si añadimos la opción -u sólo de los paquetes que pueden ser actualizados. Con la opción -a veremos información más detallada. Si vemos el mensaje "*newer than version in archive*" quiere decir que el paquete es una versión más reciente de la que en teoría está disponible. Esto puede ser debido a varias causas, desde que hemos instalado el paquete de forma independiente hasta que proviene de un repositorio con versiones más actuales y que ahora no está activo (lo hemos borrado o comentado en el sources.list)

Si queremos ver **qué paquete proporciona un determinado archivo** hacemos:

dpkg -S <nombre_del_archivo>

Si queremos ver **información sobre un paquete que no es de nuestro tipo de distribución o información detallada sobre un paquete que no tenemos instalado** (p.e. en qué paquete se encuentra un determinado fichero):

Pues vamos a esta página: <http://www.debian.org/distrib/packages> y ahí encontraremos un formulario para hacer búsquedas sobre todos los paquetes de cualquier distribución.

Experimentando con gaseosa

Todos los comandos del apt-get tanto los que instalan como los que desinstalan tienen una opción muy maja, que es -s. Con esta opción, hagamos lo que hagamos sólo lo simulará y nos mostrará información de cómo se supone que irían las cosas si lo hacemos de verdad. Normalmente, a medida que cogemos confianza en el manejo, no la usaremos, pero si alguna vez queremos hacer pruebas o algo que se sale de lo normal, viene muy bien, por si las moscas:

```
apt-get remove kpdf -s
```

Mezclando tipos de distribución

Bien, **si queremos tener alguna aplicación que no esté en nuestro tipo de distribución** o tener alguna aplicación concreta en una versión determinada **PERO NO queremos cambiar nuestra distribución**, podemos hacer lo siguiente. Suponiendo p.e. que tienes Debian estable, y quieras mezclar los paquetes de ella con los de testing, primero hacemos

```
apt-get update
apt-get dist-upgrade
```

para asegurarnos que tenemos la distribución actualizada. Luego añadimos las entradas de testing al **sources.list**:

```
# Debian testing
deb http://ftp.de.debian.org/debian/ testing main contrib non-free
deb-src http://ftp.de.debian.org/debian/ testing main contrib non-free

deb http://ftp.de.debian.org/debian-non-US testing/non-US main contrib non-free
deb-src http://ftp.de.debian.org/debian-non-US testing/non-US main contrib non-free
```

A continuación, editamos o creamos si no existe el fichero **/etc/apt/apt.conf** y ponemos esta línea:

```
APT::Default-Release "stable";
```

Ahora, aunque tengamos los repositorios de testing, el sistema no se actualizará con los paquetes de esa distribución, ya que le indicamos que la que tiene que usar por defecto es la estable. Si en un momento dado queremos pasar toda la distribución a testing, podemos hacer:

```
apt-get dist-upgrade -t testing -s
```

La opción -s es para que sólo lo simule, por si nos entra un yuyu, si vemos que no pasa nada raro, podemos volver a ejecutar el comando sin -s para que lo haga de verdad.

Bien, ahora para quedarnos más tranquilos podemos hacer

```
apt-get dist-upgrade -s
```

y vemos que efectivamente, no nos actualiza la distribución. Ahora podemos instalar ese paquete que sólo estaba en testing. Pero si ya tenemos una versión anterior del paquete instalada o éste depende de otros cuya versión sólo está disponible en testing, se va a quejar. En este caso podemos forzar a que instale los paquetes así:

```
apt-get install -t testing <nombre_del_paquete>
```

Aquí también es una buena idea usar primero la opción `-s` por si acaso. Aún así **tened en cuenta que si actualizar un determinado paquete implica la actualización de algún paquete básico del sistema** (como `libc`) **podemos encontrarnos con que hemos actualizado sin pretenderlo todo el sistema** o incluso haber montado un buen lío. **Como regla básica, usad siempre la opción `-s`** y si queremos actualizar o instalar un paquete o unos pocos y nos sale como resultado que se van a actualizar un porrón, pensadlo dos veces. También es muy conveniente tomar nota de lo que hacemos y de los paquetes que cambiamos.

Recordad que `apt-show-versions` nos dará información sobre todas las versiones disponibles de un paquete y la rama (tipo de distribución) en que se encuentran.

Instalando una versión concreta de un paquete

Si por cualquier motivo necesitamos una versión concreta de un paquete o instalar una versión anterior (por motivos de compatibilidad p.e.) lo podemos hacer así:

```
apt-get install <nombre_del_paquete>=version
```

por ejemplo:

```
apt-get install libfreetype6=2.1.7-2.4
```

Si no encuentra el paquete con la versión que necesitamos, se puede buscar en <http://snapshot.debian.net>, añadiendo al `sources.list`:

```
deb http://snapshot.debian.net/archive pool libfreetype6
```

Para evitar que el sistema actualice el paquete (p.e al hacer un `apt-get dist-upgrade`) podemos hacer:

```
echo "libfreetype6 hold" | dpkg --set-selections
```

De esta forma el paquete no se movería de la versión de la que está y si el sistema necesitara actualizarlo, no lo haría, pero nos avisaría que el paquete está retenido. Si queremos **podemos deshacer esto** tecleando:

```
echo "libfreetype6 install" | dpkg --set-selections
```

Creando un repositorio local

Un repositorio local **consiste simplemente en almacenar manualmente los archivos `.deb`** (los paquetes) **en nuestro disco duro y que funcione de forma transparente**: todo se hace igual sólo que en vez de conseguir los paquetes de Internet, los saca de nuestro propio equipo. ¿Y para qué queremos esto teniendo tantos repositorios en la red? La respuesta obvia es si no tenemos conexión o ésta es lenta o con muchos problemas; otras podrían ser tener un repositorio propio para experimentar o automatizar el manejo de paquetes sueltos (backports) etc...

Primero, creamos un directorio y colocamos los `.debs` que queramos allí. En el caso típico de un ordenador sin conexión, podemos obtener los paquetes de otro ordenador que sí tenga una buena conexión y con la misma distribución, versión y rama que en el nuestro. Basta con copiar los paquetes de `/var/cache/apt/archives` (que es donde se almacenan los paquetes que se bajan de Internet antes de instalarlos) y pasarlo al directorio que hemos creado en nuestro equipo, por ejemplo:

```
mkdir /root/mispaquetes
```

A continuación instalamos, si no la tenemos ya, la herramienta `dpkg-scanpackages`:

apt-get install apt-move feta

Si nos da problemas la instalación, podemos probar con

```
dpkg-repack dpkg-dev
```

Ahora desde el directorio superior al que vamos a almacenar los paquetes (en nuestro ejemplo sería en /root) hacemos:

```
dpkg-scanpackages mispaquetes /dev/null | gzip > mispaquetes/Packages.gz
```

Lo que estamos haciendo es crear un archivo Packages.gz el cual contiene información acerca de los paquetes. OJO, si más adelante metemos más paquetes en el directorio, hay que repetir el comando para que se actualice la información de los paquetes disponibles.

Ya sólo queda añadir al sources.list:

```
# Mi repositorio local  
deb file:/root mispaquetes/
```

Ahora hacemos **apt-get update** y nuestro repositorio local ya está funcionando.

Tener un sistema con los mismos paquetes que otro

Para tener un sistema con los mismos paquetes que tiene instalado otro, basta con hacer:

```
dpkg --get-selections > mispaquetes.txt
```

Obtendremos un archivo mispaquetes.txt con el listado de todos los paquetes instalados, y ahora para hacer que el otro sistema tenga los mismos paquetes (tanto instalados como desinstalados) lo copiamos en éste y ejecutamos:

```
dpkg --set-selections < mispaquetes.txt
```

APT seguro

Esta funcionalidad se introdujo a partir de la versión "etch" de Debian. En pocas palabras se trata de **garantizar que los sitios desde los que se descargan los paquetes** (los repositorios, vaya) **son de confianza, que los paquetes no han sido alterados** (integridad) **y que la transmisión es segura**. Se basa en el uso de criptografía fuerte y firmas digitales en la validación de los paquetes descargados. Una explicación detallada de qué son y cómo funcionan las firmas digitales basadas en los sistemas de clave pública y privada (criptografía asimétrica) excedería el alcance de este manual, para lo que aquí nos interesa, simplemente saber que si confiamos en un determinado repositorio, tenemos que conseguir su clave pública y esta clave nos garantizará la integridad de los paquetes que nos descarguemos de dicho repositorio. Si queréis más detalles del sistema, consultad la web <http://wiki.debian.org/SecureApt> (en inglés), sobre los sistemas de clave pública hay abundante información en Internet (por ejemplo http://es.wikipedia.org/wiki/Clave_p%C3%BAblica)

Por defecto, sólo se incluyen las claves con las que se firman los repositorios oficiales de paquetes de Debian. Estas claves se incluyen en el paquete "debian-archive-keyring". También se incluye la herramienta apt-key que permite añadir nuevas claves al anillo de claves de apt.

Debian guarda las claves públicas de los repositorios en el archivo `/etc/apt/trusted.gpg`. Podemos ver su contenido ejecutando el comando `sgte` como root:

apt-key list

Esto nos muestra las claves, el nivel y tipo de cifrado (normalmente 1024D) seguido de su referencia (también llamada UID) y un detalle importante: la fecha de expiración. Para mayor seguridad, las claves tienen una vigencia limitada, pasada la cual caducan y son sustituidas por otras nuevas, que nosotros deberemos incorporar. ¿Que cómo se hace esto? Muy sencillo: en su configuración por omisión, apt mostrará un aviso cuando se descarguen paquetes de fuentes que no están autenticadas o cuya clave ha caducado. **Las claves de los repositorios oficiales de Debian están vigentes durante 1 año (aprox.), cuando se presentan conflictos basta con actualizar (o descargar) el paquete "debian-keyring":**

```
apt-get install debian-keyring debian-archive-keyring
```

y añadir las claves que contiene al anillo de claves de confianza de apt:

```
apt-key add /usr/share/keyrings/debian-role-keys.gpg
```

¿Significa esto que si no tenemos las claves no podremos usar los repositorios? En absoluto. Como hemos dicho si intentamos usar repositorios sin clave o con la clave caducada, nos aparecerá un mensaje de error, algo similar a *"GPG error: http://ftp.us.debian.org testing Release: The following signatures couldn't be verified because the public key is not available"* al hacer `apt-get update`, pero esto no nos impedirá descargar los paquetes, simplemente nos dará un mensaje de aviso y nos pedirá confirmación al hacerlo. Si estos mensajes nos resultan molestos podemos evitar que aparezcan usando la opción `--allow-unauthenticated` cuando ejecutemos apt; aunque siempre será mejor, si no nos causa muchos quebraderos de cabeza, instalar las claves.

En el caso de repositorios no oficiales, para añadir sus claves, hacemos: primero, obtener la clave y luego añadirla con apt-key a las claves del apt. Deberíamos poder bajarnos el archivo donde se encuentra la clave (normalmente es un archivo de texto con la extensión `.asc` ó `.key`) del propio repositorio. La forma de hacerlo debería estar indicada en la web del repositorio; una forma podría ser:

```
wget http://ftp.unrepositorio.org/archivo_clave.asc -O - | apt-key add -
```

En este caso estamos descargando el archivo con la clave y añadiendo ésta en un sólo paso, otra forma, paso a paso sería:

tras obtener el archivo con la clave hacemos:

```
gpg --import archivo_clave.asc
```

Luego vemos cuál es la referencia de la clave con:

```
gpg --list-keys
```

y finalmente (supongamos que "E23C5FC3" es la referencia buscada) añadimos la clave a apt:

```
gpg -a --export E23C5FC3 | apt-key add -
```

Otra forma es buscar la referencia de la clave en un servidor de claves, (p.e. en el servidor de claves de RedIRIS: rediris.es/cert/servicios/keyserver/), y una vez obtenida añadirla a las claves del apt:

```
gpg --keyserver pgpkeys.mit.edu --recv-key 2D230C5F  
gpg -a --export 2D230C5F | apt-key add -
```

Aquí usamos el programa para gestionar claves gpg, "pgpkeys.mit.edu" sería la dirección del servidor de claves y "2D230C5F" la referencia de la clave que queremos.

Por último hacemos **apt-get update** y ya está. Por cierto, el archivo con la clave del repositorio oficial de Debian (por si alguna vez nos hiciera falta) está en:

<http://ftp-master.debian.org>

Una buena noticia: existe un programa para realizar el manejo de claves de forma gráfica:

gui-apt-key. Ah, y un consejo: para evitar problemas con la fecha de caducidad de las claves, asegurarnos de que la fecha y hora de vuestro sistema es correcta.

Soluciones a algunos problemas

Si da un **error al intentar instalar o desinstalar un paquete**, prueba con estos dos comandos:

```
apt-get -f install  
dpkg --configure -a
```

Y luego vuelve a probar, tal vez haya que ejecutar el segundo comando más de una vez.

Si al hacer el apt-get update os da el siguiente error:

```
Reading Package Lists... Error!  
E: Dynamic MMap ran out of room  
E: [...]
```

El error ocurre por falta de cache y se soluciona añadiendo al archivo **/etc/apt/apt.conf** la línea:

```
APT::Cache-Limit "1000000";
```

Hay que subir el valor hasta que desaparezca el mensaje (pero a intervalos grandes, de 100000 en 100000 p.e.).

* APTITUDE *

Aptitude es la nueva herramienta de instalación y manejo de paquetes, introducida con la versión "Sarge" de Debian y destinada a reemplazar al apt-get en un futuro. Aptitude presenta alguna ventaja frente a apt-get, por ejemplo, maneja mejor las dependencias entre paquetes. Tanto uno como otro te informan en el momento de instalar de si necesitas instalar otras librerías adicionales, pero después, en caso de quitar ese programa, sólo aptitude recuerda que las ha instalado, borrándolas también. apt-get permite borrar el programa, pero no las librerías adicionales.

Otra diferencia, relacionada con lo anterior, es que aptitude mantiene un log donde guarda todas las acciones realizadas, en /var/log/aptitude.

¿Que cómo se maneja? Muy fácil, **su manejo básico es similar al apt, solamente que en vez de varios comandos** (apt-get, apt-cache, ...) **hay uno sólo: aptitude y el resto es igual**. P.e. para instalar un paquete haríamos:

```
aptitude install <nombre_del_paquete>
```

y para eliminarlo:

```
aptitude remove <nombre_del_paquete>
```

..y así sucesivamente. Lamentablemente, no todo es perfecto, por ejemplo, aptitude search pdf kde no funciona igual, busca los paquetes que contengan "pdf" O "kde", pero no los que contengan ambos términos en su descripción. Bueno, hay una forma "fácil e intuitiva" (sobre todo intuitiva) de que aptitude haga lo mismo:

```
aptitude search ~dpdf~dkde
```

Aquí el "~d" indica que busquemos en la descripción del paquete, mientras que si fuera "~n", buscaría en el nombre solamente. Por cierto, el carácter este raro (~) se llama virgulilla y se obtiene pulsando AltGr+4 (en el mismo sitio en el que está el \$).

También **hay algunas funciones añadidas que apt no tenía y otras que se hacen de forma ligeramente distinta**. P.e:

```
aptitude reinstall <nombre_del_paquete>
```

reinstala un paquete, evitándonos así tener que desinstalarlo y vuelta a instalar.

```
aptitude purge <nombre_del_paquete>
```

elimina un paquete y sus ficheros de configuración.

```
aptitude hold <nombre_paquete>
```

bloquea el paquete

```
aptitude unhold <nombre_paquete>
```

desbloquea el paquete

Recordad también que no se puede usar apt y aptitude, es uno u otro. Si empezamos usando aptitude, hay que seguir usándolo siempre y viceversa para apt. Esto es porque cada programa ignora lo que hace el otro y si mezclamos su uso, cada uno tendrá un listado de la configuración de los paquetes del sistema distinto y podemos armarla buena. **La excepción serían las búsquedas**, ya que en ese caso no estamos cambiando el sistema, solamente haciendo consultas.

Con el tiempo aptitude será el sistema base estándar para manejar paquetes en Debian y un consejo juicioso sería que si tenéis una distribución reciente (a partir de Etch, p.e.) os olvidarais del apt y usaseis aptitude. Sin embargo en el momento de redactar estas líneas, la balanza aún

está nivelada y para ser completamente sinceros, nuestra experiencia con aptitude es menor que con apt. Desde luego, para un uso básico, creemos que aptitude es la mejor opción; la duda estaría si vamos a hacer un uso avanzado y queremos tener un gran control sobre lo que hace el sistema de paquetes.

* SYNAPTIC *

También conocida como "Administrador o Gestor de paquetes", **Synaptic es simplemente una interfaz gráfica que nos permite hacer la gestión de paquetes mediante el uso de ventanas y el ratón, en vez de en línea de comandos.** Como suele suceder con este tipo de herramientas, cambia facilidad de uso por flexibilidad y potencia; o sea, que **hace menos cosas que apt, pero lo que hace, lo hace de forma más sencilla para el usuario.** Nuestro consejo es que si os va la comodidad y no queréis ir más allá de lo básico, Synaptic os puede gustar, pero aún así, merece la pena echar un vistazo al menos a los fundamentos de apt; primero, porque con poco esfuerzo, se puede llegar muy lejos y segundo, porque si conocéis lo esencial sobre apt, Synaptic os resultará pan comido.

Al ejecutar Synaptic nos pide la clave de root, claro, y una vez introducida vemos una ventana en la que aparece la lista completa de los paquetes del sistema (tanto los instalados como los que no) ordenados alfabéticamente y por categorías. A la izda. aparecen las distintas categorías, si pulsamos sobre una veremos sólo los paquetes de esa categoría; también está la categoría "Todo" (la primera de la lista) que como su propio nombre indica nos sirve para ver todos los paquetes. De los paquetes podemos ver su nombre, la versión instalada (si el paquete está instalado), la última versión disponible y una descripción corta; también aparece un pequeño icono que nos indica el estado del paquete, un cuadrado en blanco significa que el paquete no está instalado y un cuadrado coloreado (en verde) que sí que lo está; podemos ver el resto de iconos y su significado en la ayuda. Pinchando sobre el nombre de cada paquete podemos ver una descripción más detallada, si además luego pinchamos en el botón "Propiedades" podremos ver más información sobre el paquete (tamaño, quién es el mantenedor, de qué paquetes depende, etc...)

Lo primero que hay que hacer es clic sobre el botón de "Recargar" para actualizar la lista de los paquetes disponibles.

Si ahora queremos ver qué paquetes están instalados o cuáles son actualizables, podemos hacer clic en el botón "Estado" que aparece debajo del listado de categorías, con lo que accedemos a una clasificación de los paquetes por su estado actual en vez de por su categoría; para volver a la vista normal, pulsamos en "Secciones".

Buscar un paquete es muy sencillo: usamos el botón "Buscar", el listado de categorías cambiará a un listado de términos de búsqueda; seleccionando el término que acabamos de introducir veremos qué paquetes se corresponden. De nuevo, para volver a la vista normal, pulsamos en "Secciones".

Instalar, desinstalar o actualizar paquetes es igual de sencillo, escogemos los paquetes (podemos usar las teclas CTRL y Shift (Mays) para escoger varios, igual que haríamos con archivos normales), los marcamos con la acción que queremos haciendo doble clic en el icono que aparece junto al nombre del paquete o escogiendo la acción en el menú "Paquete", y luego pulsamos el botón "Aplicar". Si lo único que queremos es actualizar los paquetes instalados, podemos simplemente hacer clic en el botón de "Marcar todas las actualizaciones" y luego en "Aplicar". Verdaderamente el programa es muy fácil de usar y en seguida nos haremos con él; para más detalles podemos consultar la ayuda.

Comentar por último que podemos editar la lista de los repositorios y algunas opciones más sobre la obtención e instalación de paquetes en el menú Configuración -> Repositorios.

*** Instalación por binarios. ***

Esto les sonará familiar a los que vengan del mundo Güindous, ya que la mecánica de instalación es similar en muchas aplicaciones. **Básicamente se trata de bajarse el programa de la página del mismo y seguir las instrucciones que allí nos den. Normalmente los pasos suelen ser: descomprimir el archivo y luego ejecutar un script.** Para ilustrarlo, nada mejor que un ejemplo práctico: vamos a instalar el navegador web Firefox y añadirle el plugin de Java. NOTA: Esto lo hacemos solamente a modo ilustrativo, recordad que ante cualquier problema o duda debemos acudir siempre a las instrucciones que nos darán en la página web del programa. **Tened en cuenta que los programas instalados sin proceder de un archivo .deb están fuera del control de las herramientas de administración de paquetes (apt, aptitude, etc..)**

Debian incluye al Firefox, claro (aunque por razones legales, le cambia el logo y el nombre: Iceweasel, es exactamente el mismo programa); pero debido al rápido desarrollo de la aplicación, si usamos la rama estable de Debian seguramente no tendremos la versión más actual de Firefox; no pasa nada, hacemos:

Descargamos el archivo de Firefox de su página web: <http://www.mozilla-europe.org/es/firefox/> Guardamos el archivo descargado en un directorio para su instalación. Uno especialmente adecuado sería `/usr/local`. Recordad que es mejor hacer todos los pasos de la instalación como root. Ahora atención a la extensión del fichero: si es un `.tar.gz` ó un `.tar.bz2` esto indica que es un fichero comprimido y lo primero que tendremos que hacer será descomprimirlo; en el primer caso la instrucción es

```
tar -zxvf <nombre_fichero>.tar.gz
```

y en el segundo

```
tar -jxvf <nombre_fichero>.tar.bz2
```

Al descomprimirse, los ficheros del programa se guardan automáticamente en un directorio llamado "firefox". Dentro de ese directorio suele estar el script que lanza la aplicación, llamado también "firefox" (equivalente al `.exe` del Güindous), para mayor comodidad o si nos da problemas de permisos al intentar ejecutarlo, podemos poner como propietario a nuestro usuario:

```
chown <nuestro_nombre_usuario> firefox/ -R
```

Y ya está. Ahora vamos a hacer lo mismo con el plugin Java. Lo primero lo descargamos en <http://www.java.com/es/download/>

De nuevo lo copiamos en `/usr/local`. En este caso la extensión es `.bin`, o sea, que lo podemos ejecutar directamente:

```
./jre-<lo_que_sea>.bin
```

Tras leer y aceptar (escribimos "yes" y pulsamos Enter) las condiciones, ya tenemos Java instalado. Si se niega a ejecutarse puede ser que tengamos que darle permisos de ejecución:

```
chmod +x jre-<lo_que_sea>.bin
```

Ahora hay que enlazar el plugin con Firefox, aunque esto es específico de Java y se sale un poco del tema, vamos a verlo en breve:

```
Vamos al dir. de plugins del firefox
cd /usr/local/firefox/plugins
```

Creamos un enlace al plugin de java:

```
ln -s /usr/local/<nombre_directorio_jre>/plugin/i386/ns7/libjavaplugin_oji.so .
```

(no os olvidéis del punto al final)

Comprobación sumas MD5

Una última nota: en muchas páginas de programas aparecerá un fichero .md5 junto al programa para descargar o una ristra de números y letras llamada "md5 sum" (suma md5). Pues bien **se trata simplemente de una comprobación para asegurarnos que el fichero se ha descargado correctamente y no está corrupto, pero también para garantizar que nadie ni nada ha alterado el fichero original, algo interesante para aquellos a los que les preocupe la seguridad**. Esa ristra de números y letras es el resultado de aplicar el algoritmo md5 al fichero, una especie de dni del mismo. Luego, cuando nos bajemos el programa, basta con que volvamos a calcular la suma md5 y comprobar que coincide con la que el desarrollador ha colgado en la página del programa. Para ello hacemos:

```
md5sum <nombre_fichero>
```

y comprobamos que los resultados son iguales. Como esto es algo pesado, podemos hacer que el programa compruebe el resultado por nosotros. Si tenemos el fichero .md5, lo guardamos en el mismo directorio del programa y hacemos:

```
md5sum -c <nombre_fichero_md5>
```

¿Que no tenemos el fichero md5? Pues lo hacemos nosotros mismos: basta con crear un fichero de texto normal (lo podemos llamar como queramos y le ponemos de extensión .md5) con una sola línea con la suma md5 que aparezca en la web del programa (copia y pega), DOS espacios en blanco (ni más ni menos) y el nombre del programa a comprobar (cuidad de que sea exactamente igual).

Esta comprobación no sólo sirve para binarios ejecutables, sino también para todo tipo de ficheros, como por ejemplo imágenes ISO de CD o DVD.

Backports.

Puertas traseras. O cómo conseguir lo que queremos si la puerta principal está cerrada. Estrictamente hablando "**Debian backports**" es un proyecto de Debian para dar apoyo a los usuarios de la rama estable, de forma que puedan tener ciertos paquetes más actualizados sin necesidad de mezclar ramas y evitando en lo posible problemas de dependencias. Para más información véase la página del proyecto <http://www.backports.org>. Aquí aplicaremos el concepto en un sentido amplio para incluir la instalación manual de paquetes que no pertenezcan a nuestra rama o incluso que no pertenezcan a Debian.

Usar backports.org es sencillo, primero añadimos esta línea al `sources.list`:

```
deb http://www.backports.org/debian lenny-backports main contrib non-free
```

notad que donde pone "lenny" debemos sustituirlo por el nombre propio de nuestra rama. Ahora hacemos `apt-get update` y listo. Los backports están desactivados por defecto, si queremos instalar un paquete de backports, hacemos:

```
apt-get -t lenny-backports install <nombre_del_paquete>
```

Otra forma de hacer esto es bajarse los archivos `.deb` de los paquetes que queremos y luego instalarlos "a mano" de esta manera:

```
dpkg -i <nombre_archivo>.deb
```

Podemos instalar varios paquetes de una tacada: `dpkg -i *.deb`

Aparte de la web ya indicada, **tenemos muchos otros sitios desde donde bajarse paquetes, eso sin contar que en muchas ocasiones las webs de proyectos o programas determinados nos ofrecerán también descargarse el programa en formato `.deb`**. Algunos específicamente dedicados a Debian son:

<http://www.debian.org/distrib/packages>

<http://www.apt-get.org>

<http://snapshot.debian.net>

Un buen ejemplo de instalación de una aplicación mediante backports es la conocida suite ofimática OpenOffice, que por supuesto, viene incluida en Debian, pero si queremos tener la última versión, podemos ir a su web (<http://download.openoffice.org>) y descargarnos los archivos `.deb`

Si instalamos un paquete con `dpkg -i` y las dependencias no están satisfechas, el paquete quedará sin configurar y el sistema nos avisará que se encontraron errores. Podemos o bien desinstalar el paquete (con `apt-get remove` o con `dpkg -r`) o hacer `apt-get -f install` o intentar instalar los paquetes de los que depende. Este es el principal problema que nos podemos encontrar al instalar paquetes manualmente: que dependan de otros que a su vez dependen de otros y así... eso sin contar con problemas de conflicto entre versiones de paquetes. Es por esto que debemos hacerlo con tiento, preferiblemente con pocos paquetes y sin empecinarnos demasiado si las cosas no van bien.

Por último, y aunque se sale del ámbito de este documento, comentar que **pueden instalarse** (o al menos intentarlo) **paquetes preparados para otras distribuciones de GNU/Linux**. Alien es un programa que permite convertir un paquete binario en formato no Debian (por ejemplo, en formato `.rpm` que usan RedHat, Suse y Fedora) a formato `.deb`. Esto podemos intentarlo como último recurso en el caso de que no haya forma de conseguir el paquete en formato Debian y tampoco lo podamos compilar o instalar como binario. Para convertir un paquete haríamos:

```
alien --to-deb <archivo_paquete>
```

Con la opción `-i` podemos solicitar que alien intente instalar el paquete una vez convertido. En Debian también se incluyen herramientas de instalación propias de otras distribuciones como `rpm` (para instalar paquetes `.rpm`)

Compilación.

Como comentábamos anteriormente, la mecánica es similar a la instalación por binarios, sólo que aquí hay que hacer algunos pasos más debido a que lo que obtenemos no son programas ejecutables y sus librerías, sino ficheros de texto con las instrucciones del programa (las "fuentes") y lo que tenemos que hacer es "construir" el programa (lo que se dice compilarlo). Es como elaborar un plato a partir de los ingredientes y la receta, sólo que aquí lo hará el ordenador por nosotros. DOS ADVERTENCIAS:

- Una, aquí es (aún más) **especialmente importante seguir las instrucciones y recomendaciones que nos proporcionen los autores del programa**, ya sea a través de su página web o en los ficheros "README" o similares que vienen junto con el mismo.
- Dos, **es posible que durante la compilación nos dé errores u advertencias de que faltan determinadas librerías en nuestra máquina** (o sea que le faltan ingredientes para hacer la receta), en este caso debemos instalarlas (fijémonos en el nombre e instalamos también las que tengan el sufijo "-dev" en el nombre, ya que eso indica que son las librerías de desarrollo, o sea, no hacen falta para ejecutar pero sí para compilar, que es justo lo que queremos). **Tened en cuenta que los programas instalados sin proceder de un archivo .deb están fuera del control de las herramientas de administración de paquetes** (apt, aptitude, etc..) y por tanto, no pueden ser manejados por éstas.

Los pasos típicos suelen ser:

1. Bajarse el archivo comprimido con las fuentes
2. Descomprimir
3. Configurar
4. Compilar
5. Instalar

(los tres últimos pasos suelen hacerse cada uno con una sola instrucción). Vamos a ver un **ejemplo**:

Puede que al instalar una aplicación mediante binarios o un backport, ésta se queje de que las versiones de ciertas librerías del sistema son inferiores (esto pasaba por ejemplo con el Firefox 3 en Debian etch). Ciertas librerías básicas del sistema, de las que dependen muchas aplicaciones, más vale no tocarlas (no al menos hasta que actualicemos el sistema entero) no sea que por arreglar uno, estropeemos ciento. Existe, sin embargo, una solución: compilar la nueva versión y guardarla en un directorio aparte; en el caso concreto del Firefox 3, necesitaba una versión de las GTK (librerías gráficas). Veamos cómo haríamos:

1. Bajamos el archivo comprimido con las fuentes de <http://ftp.gnome.org/pub/gnome/sources/gtk+/2.10/gtk+-2.10.0.tar.bz2>
2. Luego, **como root**, vamos al directorio `/opt` y copiamos allí el archivo `.tar.bz2` para instalarlas (este es un directorio adecuado para librerías opcionales, para programas nos podría valer `/usr/local`). Hacemos:

Creamos un directorio para esta versión de las GTK: `mkdir gtk210`
Descomprimos: `tar -jxvf gtk+-2.10.0.tar.bz2`

3. Vamos al directorio de las fuentes: `cd gtk+-2.10.0`
Configuramos: `./configure --prefix=/opt/gtk210` (notad que le pasamos un parámetro indicando el directorio de instalación)
4. Compilamos: `make`
5. Instalamos: `make install`

Y ya está. Verdaderamente no es muy difícil. Por lo general el mayor problema está en conseguir las librerías (los "ingredientes") que nos puedan faltar y a veces, ajustar las opciones de configuración. Si tenemos todos los ingredientes, en la mayoría de los casos con hacer

"./configure" (sin parámetros), "make" y "make install" ya tendremos el programa compilado e instalado. **Si al compilar con make, ocurre algún error, podemos borrar los ficheros intermedios generados con "make clean"**, para una vez corregido éste, volver a hacer "make" a partir de cero.

Ah, se me olvidaba. En el ejemplo, para ejecutar Firefox 3, quedaría por crear (lo podemos hacer con cualquier editor de texto) el script de inicio del Firefox, lo llamamos "firefox.sh" y lo guardamos en el directorio de instalación del mismo. El contenido del script sería:

```
#!/bin/sh
export LD_LIBRARY_PATH="/opt/gtk210/lib"
/usr/local/firefox3/firefox $*
```

Donde pone "/usr/local/firefox3" ponemos el directorio donde hallamos instalado el Firefox 3 y listo.

Historia:

29 Septiembre 2009: Redacción del documento “*Guía APT*” por Carlos Felipe Alvarez González.

<http://stoneheads.wordpress.com/2009/09/29/guia-apt/>

Dicho documento se ha basado en:

Enero 2004: “*GUÍA BREVE DE INSTALACIÓN Y CONFIGURACIÓN DE DEBIAN GNU/LINUX*”.
(*Capítulo 2*) versión 2.1 por Juan Antonio Silva Villar.

Anexo I. Licencia del documento: GNU Free Documentation License

(Puede encontrarse una traducción no oficial de esta licencia al castellano en <http://curso-sobre.berlios.de/gfdles/gfdles.html>)

GNU Free Documentation License

Version 1.2, November 2002
Copyright (C) 2000,2001,2002 Free Software Foundation, Inc.
51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

D. Preserve all the copyright notices of the Document.

E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

H. Include an unaltered copy of this License.

I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the

Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.

N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.

O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties--for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements."

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects. You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those

notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

Anexo II. Cláusula Mariposa/Butterfly Clause

BUTTERFLY CLAUSE

Version 1
December 2006

Copyright (C) 2006 José María Lancho Rodríguez
Author: José María Lancho Rodríguez
C/Sagasta nº 24, 6º derecha
Madrid, Spain

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Translations of this license to other languages are included below

BUTTERFLY CLAUSE AGREEMENT

The licensor excludes any use of the provided software in State Public Administrations acquired in the case the State executes an action of armed aggression against a third State or exercises actions which constitute genocide, crimes against humanity or war crimes.

The default of any of the prohibitions which are related with the granted rights will mean the immediate termination, without need for prior notice, of the main license of the granted software and the immediate annulment of the granted rights and the abstaining of its exercise within Public Administrations. In all cases it will allow for the demand of damages to the Licensor before the Pacific Technological Arbitration Court located in Paris or in Zurich, to be chosen by the plaintiff, submitting to the arbitration regulations of that Court, granting the capacity to be amicus curiae or even principal in defect of license issuer, to any disadvantaged, enabling, in any case the demandability and grantability of sufficient punitive and coercive damages to stop the unauthorized use of the, all of the above regardless of the criminal responsibilities of the implicated individuals, and only using equity as a criteria to determine the arbitral response to which both parties agree to be submitted.

The use of the licensed programs means the acceptance of these contractual terms as autonomous contractual expression of the software bearer conscience clause. The parties expressly waive a second reviewing arbitration of the arbitral decision.

END OF TERMS AND CONDITIONS

CONTRATO DE LICENCIA:

El licenciador excluye cualquier uso del software cedido en las Administraciones Públicas del Estado adquirente en los supuestos de que el Estado proceda a efectuar un acto de agresión armada contra un tercer Estado o proceda a efectuar actos constitutivos de genocidio, crímenes de lesa humanidad o crímenes de guerra.

El incumplimiento de cualquiera de estas prohibiciones que quedan conectadas con los derechos cedidos, supondrá automáticamente la terminación, sin necesidad de previa notificación, de la licencia principal del software cedido y en consecuencia la inmediata revocación de los derechos concedidos y la abstención de su ejercicio dentro de las Administraciones Públicas. En todo caso permitirá la exigencia de daños y perjuicios al Licenciario ante la Corte Arbitral de Tecnologías Pacíficas, sita en París o en Zurich, a libre elección del demandante, sometiéndose a las normas de arbitraje de esa Corte, otorgando la capacidad para ser parte coadyuvante e incluso principal en defecto del licenciante, a cualquier perjudicado, posibilitando, en todo caso la exigibilidad y otorgabilidad de daños punitivos y daños coercitivos suficientes para detener la utilización no autorizada del software, todo ello con independencia de las responsabilidades criminales de las personas naturales implicadas, y utilizando exclusivamente la equidad como criterio determinante de la respuesta arbitral a que ambas partes aceptan someterse.

La utilización del programa licenciado supone la aceptación de los términos de este contrato como expresión contractual autónoma de cláusula de conciencia del titular del software. Las partes hacen expresa renuncia de una segunda instancia revisora de la sentencia arbitral.

FIN DE LA LICENCIA

CLAUSE PAPILLON CONTRAT DE LICENCE:

Le donneur de licence exclut toute utilisation du logiciel cédé aux Administrations Publiques de l'Etat dans l'hypothèse ou cet Etat effectue un acte d'agression armée contre un Etat tiers, ou qu'il réalise des actes constitutifs de génocide, de crimes de lèse humanité ou de crimes de guerre.

Le manquement à ces interdictions qui sont en connexion avec les droits cédés, entraînera automatiquement la cessation, sans préavis, de la licence principale du logiciel cédé, et, en conséquence, la révocation immédiate des droits octroyés et l'abstention de leur exercice au sein des Administrations Publiques.

En tout cas, il permettra l'exigence de dommages et intérêts au donneur de licence devant la Cour Pacifique d'Arbitrage Technologique siégeant à Paris ou à Zurich, selon la libre élection du demandeur, se soumettant aux normes d'arbitrage de ladite Cour, en octroyant à tous ceux qui s'estimeraient lésés la capacité pour s'unir ou remplacer en son défaut le donneur de licence, rendant possible en tout cas l'octroi de dommages punitifs et de dommages coercitifs suffisants afin de mettre fin à l'utilisation non autorisée du logiciel, ceci indépendamment des responsabilités criminelles des personnes naturelles impliquée, et en utilisant exclusivement l'équité comme critère déterminant de la décision de l'arbitrage auquel les deux parties acceptent de se soumettre.

L'utilisation du logiciel licencié suppose l'acceptation des termes de ce contrat comme expression contractuelle de clause de conscience du titulaire du logiciel. Les parties renoncent de façon expresse à une seconde instance révisant la décision de l'arbitrage.

FIN DE LA LICENCE

KLAUSEL SCHMETTERLING LIZENSVERTRAG

Der Lizenzgeber schließt irgend einen Gebrauch der übertragenen Software an die öffentliche Verwaltung des empfangenden Staates aus in den Fällen in welchen der Staat eine bewaffnete Angriffshandlung gegen einen dritten Staat oder Handlungen durchführt, die einen Völkermord begründen, Straftaten von geringer Menschlichkeit oder Kriegsverbrechen.

Die Nichteinhaltung von irgend einem dieser Verbote, die mit den übertragenen Rechten verbunden bleiben, wird automatisch die Beendigung bedingen, ohne die Notwendigkeit der vorhergehenden Benachrichtigung, der Hauptlizenz der übertragenen Software und konsequenterweise die sofortige Rücknahme der gewährten Rechte und der Abstandnahme ihrer Ausübung innerhalb der öffentlichen Verwaltungen. In jedem Fall erlaubt es, vom Lizenznehmer Schäden vor dem pazifistischen technologischen Schiedsgericht gelegen in Paris oder Zürich, nach freier Wahl des Klägers, zu verlangen, wobei sich dieser den Normen dieses Schiedsgerichts unterwirft und hierbei die Fähigkeit erlaubt wird, ununterstützender Teil und sogar hauptsächlich in Abwesenheit des Lizenzgebers, jedem Benachteiligten, so ermöglicht in jedem Fall die Einklagbarkeit und die Gewährung von Schäden, die aus Straftaten stammen und von zwingenden Schäden, um die nicht erlaubte Benutzung der Software zu stoppen, dies alles mit Unabhängigkeit der strafrechtlichen Verantwortung der in dies verwickelten natürlichen Personen und exklusiv die Rechtlichkeit als determinierendes Kriterium benutzend der schiedsrichterlichen Antwort der beide Seiten akzeptieren sich zu unterwerfen.

Die Benutzung des Programms mit der Lizenz setzt die Akzeptanz der Begriffe dieses Vertrages als vertraglichen selbständigen Ausdruck der Gewissensklausel des Inhabers der Software. Die Seiten sehen ausdrücklich von einer zweiten Instanz ab, die die Entscheidung des Schiedsgerichts überprüft.

LIZENSVERTRAGSENDE