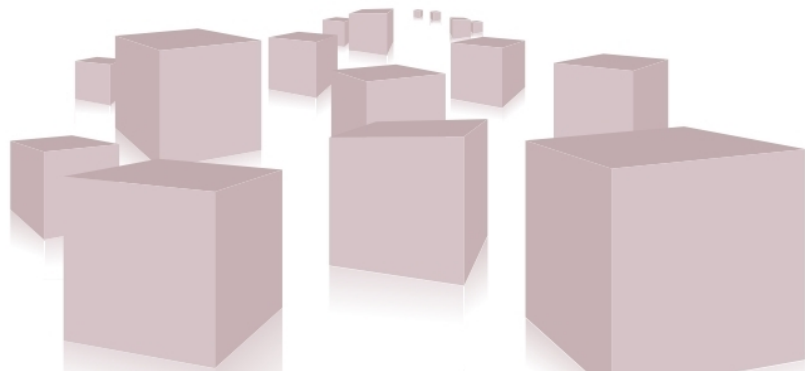


# nhopkg

The Universal Package Manager





## ÍNDICE

	Pág.
Índice.....	3
Licencia.....	4
Agradecimientos.....	4
Preguntas Frecuentes (FAQ).....	5
Manual Básico de Uso.....	7
Otras Características.....	12

## LICENCIA

Copyright (C) 2009 Jaime Gil de Sagredo Luna.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

## AGRADECIMIENTOS

Juan Carlos de la Vega Salamanca (<http://pillateunlinux.wordpress.com> y <http://putolinux.wordpress.com>).

## PREGUNTAS FRECUENTES (FAQ)

### ¿En qué distribuciones Nhopkg trabaja correctamente?

*Nhopkg* puede trabajar correctamente en cualquier distribución GNU/Linux, e incluso puede hacerlo en otros sistemas basados en Unix que cumplan con las dependencias necesarias.

Algunas distribuciones ya usan *Nhopkg* como gestor de paquetes por defecto:

- Vegnux 0.47.0 (<http://vegnux.blogspot.com>)

### ¿Qué es Nhopkg?

*Nhopkg* pretende ser un sistema de gestión de paquetes universal para cualquier sistema operativo basado en GNU/Linux o en Unix.

### ¿Qué hace Nhopkg?

*Nhopkg* es capaz de instalar paquetes binarios y de fuentes. Puede crear paquetes de fuentes (*.src.nho*) a partir de los paquetes de fuentes que suministran los desarrolladores de software en formatos tar.bz2 y tar.gz. Esta construcción de paquetes puede llevarse a cabo a partir de una serie de preguntas que realiza *Nhopkg* al usuario o bien de forma automática, ya que es capaz de reconocer qué comandos serán necesarios para instalar dicho software.

Puede compilar e instalar dichos paquetes fuentes y crear paquetes binarios (*.nho*) a partir de los archivos que se instalan en el sistema.

Además *Nhopkg* puede realizar las funciones básicas de los gestores de paquetes, como gestión de dependencias, eliminación de software, obtención de información del software instalado, instalación desde repositorios y mucho más.

### ¿Cómo se ejecuta Nhopkg?

*Nhopkg* actualmente se ejecuta a través de comandos por consola. Aun así, en próximas versiones se empezará a trabajar conjuntamente con proyectos para la creación de una interfaz gráfica.

### ¿Podemos comparar Nhopkg con rpm o dpkg?

*Nhopkg* pretende ser universal, es decir, que pueda ser usado sin problemas en cualquier sistema GNU/Linux o Unix. *Nhopkg* además puede ser complementado, ya que es capaz de chequear las dependencias de los paquetes aun si están instaladas por otros gestores.

### ¿Nhopkg es un software totalmente estable?

*Nhopkg* es un software que está en continuo desarrollo por lo que es posible que no sea 100% estable. Aun así *Nhopkg* puede ser usado sin ningún problema.

### Nhopkg es software libre, ¿Siempre lo será?

Si, *Nhopkg* es y será software libre. Actualmente está liberado bajo la Licencia GPLv3.

### ¿Quién desarrolla Nhopkg?

Actualmente solo Jaime Gil de Sagredo, su creador, aunque el desarrollo está abierto a la incorporación de nuevos desarrolladores.

### ¿En qué lenguaje está escrito Nhopkg?

*Nhopkg* está escrito en el lenguaje de script Bash. Este lenguaje posibilita que sea fácil de instalar casi en cualquier sistema que lo contenga, y permite una gran portabilidad del mismo.

### ¿Nhopkg es compatible con otros gestores de paquetes?

En parte sí y en parte no. Por ejemplo, un paquete instalado por otro gestor de paquetes no puede

ser eliminado por *Nhopkg*, ni puede obtenerse información sobre él a través de *Nhopkg*, pero sin embargo, *Nhopkg* es capaz de buscar si una dependencia está instalada en el sistema por otro gestor de paquetes, ya que no se basa en el nombre de los paquetes para gestionar dependencias, sino en los archivos instalados por las mismas.

### **Después de todo esto, ¿Por qué usar Nhopkg?**

*Nhopkg* es ligero, rápido y fácil de usar. Además tiene cierta compatibilidad con otros gestores de paquetes (como vemos en el punto anterior) y facilita en gran medida la compilación de software desde fuentes, siendo incluso capaz de hacer todo ese proceso de forma automática.

### **¿Cómo se mantiene este proyecto?**

Este proyecto se mantiene por el desarrollo del autor y las aportaciones externas de terceros, ya sea con aportes económicos en forma de donaciones, aportando código, bugs o sugerencias. Te animamos a que colabores con el proyecto.

## MANUAL BÁSICO DE USO

### Instalación local de paquetes binarios (.nho)

*Nhopkg* puede instalar uno o varios paquetes binarios de la siguiente forma.

Lo primero es descargar el paquete binario (.nho) específico para tu arquitectura, ya que no hay distinción entre distribuciones, siempre y cuando se use software del repositorio oficial. Una vez descargado, abrimos una terminal y escribimos el siguiente comando, siempre como root:

```
nhopkg -iv tu-paquete.nho tu-otro-paquete.nho
```

### Ejemplo:

```
nhopkg -iv gimp-2.6.3-2.nho vlc-0.9.8-a.nho
```

### Explicación:

El comando *-i* instala un paquete binario que esté en nuestro disco duro, es decir, de forma local. La opción *-v* hace que veamos la información que muestra *Nhopkg* al instalarlo.

### Instalación desde repositorios de paquetes binarios (.nho)

*Nhopkg* puede instalar uno o varios paquetes binarios que se encuentren en el repositorio seleccionado.

Para ello podemos listar los paquetes del repositorio y ver cuál queremos instalar.

```
nhopkg -u
```

Una vez actualizada, podemos ver qué paquetes se encuentran disponibles así:

```
nhopkg --list-repo
```

### Ejemplo:

```
nhopkg --list-repo | more
```

### Explicación:

Utilizando el comando *more* podemos ir viendo poco a poco los paquetes. Es recomendable usar esa opción si la lista es muy grande.

Instalamos los paquetes que deseamos de la siguiente manera (como root):

```
nhopkg -Sv tu-paquete tu-otro-paquete
```

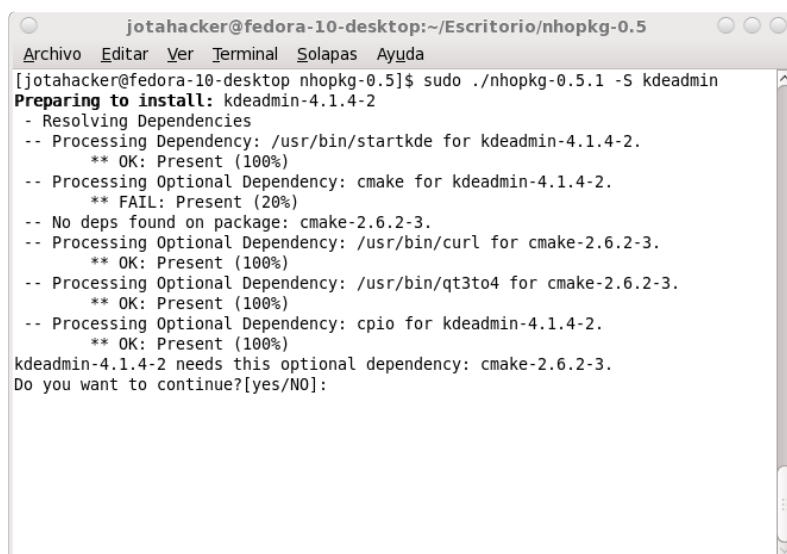
### Ejemplo:

```
nhopkg -Sv gimp-2.6.3-2 vlc
```

### Explicación:

El comando *-S* instala un paquete binario desde nuestro repositorio. Podemos especificar el paquete a instalar ya sea con su nombre completo y la versión, con únicamente su nombre o bien ofreciendo

uno de los archivos que contiene ese paquete que queremos.  
La opción `-v` hace que veamos la información que muestra *Nhopkg* al instalarlo.



```
jotahacker@fedora-10-desktop:~/Escritorio/nhopkg-0.5
Archivo Editar Ver Terminal Solapas Ayuda
[jotahacker@fedora-10-desktop nhopkg-0.5]$ sudo ./nhopkg-0.5.1 -S kheadmin
Preparing to install: kheadmin-4.1.4-2
- Resolving Dependencies
-- Processing Dependency: /usr/bin/startkde for kheadmin-4.1.4-2.
** OK: Present (100%)
-- Processing Optional Dependency: cmake for kheadmin-4.1.4-2.
** FAIL: Present (20%)
-- No deps found on package: cmake-2.6.2-3.
-- Processing Optional Dependency: /usr/bin/curl for cmake-2.6.2-3.
** OK: Present (100%)
-- Processing Optional Dependency: /usr/bin/qt3to4 for cmake-2.6.2-3.
** OK: Present (100%)
-- Processing Optional Dependency: cpio for kheadmin-4.1.4-2.
** OK: Present (100%)
kheadmin-4.1.4-2 needs this optional dependency: cmake-2.6.2-3.
Do you want to continue?[yes/NO]:
```

*Nhopkg* preguntando si instalar dependencias opcionales de *kheadmin*.

### Compilando paquetes fuente de forma local (.src.nho)

*Nhopkg* es capaz de instalar paquetes desde sus fuentes, previamente habiendo creado un paquete *.src.nho* desde el mismo. Además, en este proceso *Nhopkg* compila, instala y crea un paquete binario de dicho software. Para esto, desde consola y siempre como root, hacemos:

```
nhopkg -bv tu-paquete.src.nho tu-otro-paquete.src.nho
```

### Ejemplo:

```
nhopkg -bv gimp-2.6.3-2.src.nho vlc-0.9.8-a.src.nho
```

### Explicación:

El comando `-b` compila, instala y crea el paquete binario a partir del paquete de fuentes (*.src.nho*) suministrado de forma local.

La opción `-v` hace que veamos la información que muestra *Nhopkg* y el proceso de compilación e instalación.

### Eliminación de un paquete instalado por *Nhopkg*.

*Nhopkg* puede eliminar completamente cualquier paquete de software instalado por él, sin importar la forma en que se hizo (cualquiera de las anteriores). Antes de nada podemos ver qué paquetes están instalados por *Nhopkg*, su información, archivos que lo componen, etc.

```
nhopkg -list
```

### Ejemplo:

```
nhopkg -list | more
```

### Explicación:

Con el uso de *more* podemos mostrar los paquetes de forma más ordenada. Es recomendable en

caso de que haya muchos paquetes instalados. También se pueden usar `grep` u otros programas que nos ayuden a obtener más información.

```
nhopkg -info tu-paquete
```

**Ejemplo:**

```
nhopkg -info gimp-2.6.3-2
```

**Explicación:**

Muestra información acerca de dicho paquete. Esta información se basa en el nombre del paquete, la versión, revisión, descripción, dependencias, licencia, etc.

```
nhopkg -show tu-paquete
```

**Ejemplo:**

```
nhopkg -show vlc | more
```

**Explicación:**

Con este comando puedes ver todos los archivos que instalado dicho paquete. Es aconsejable usar también la utilidad `more`, aunque podemos usar otros como `grep`, `head`, etc

Una vez nos disponemos a eliminar el paquete que queramos, realizamos lo siguiente en consola (como usuario root):

```
nhopkg -rv tu-paquete tu-otro-paquete
```

**Ejemplo:**

```
nhopkg -rv gimp-2.6.3-2 vlc
```

**Explicación:**

Con el comando `-r` se desinstala el paquete seleccionado y todas los demás paquetes que dependan de él. Se pueden elegir los paquetes a desinstalar o bien dando el nombre completo y su versión o únicamente el nombre del paquete.

Con la opción `-v` nos muestra que paquetes se eliminan, sino, muestra una barra de progreso.

```
jotahacker@fedora-10-desktop:~/Escritorio/nhopkg-0.5
Archivo Editar Ver Terminal Solapas Ayuda
[jotahacker@fedora-10-desktop nhopkg-0.5]$ sudo ./nhopkg-0.5.1 -r cmake
Preparing to remove: cmake-2.6.2-3
The nexts packages will be deleted from your system: kdeadmin-4.1.4-2 cmake-2.6.2-3.
Do you want to continue?[yes/NO]: yes
Removing packages and inverse dependencies.
kdeadmin-4.1.4-2: 100% [=====] 00m01s ETA
cmake-2.6.2-3: 100% [=====] 00m01s ETA
-- Updating database: nhopkg.db
- cmake-2.6.2-3 has been removed from your sistem.
[jotahacker@fedora-10-desktop nhopkg-0.5]$
```

*Nhopkg después de haber eliminado cmake y los paquetes que dependen de éste.*

### Buscar paquetes y obtener información acerca de ellos.

*Nhopkg* nos permite buscar paquetes en los repositorios y comprobar si está instalado en nuestro sistema además de otra información adicional. Para ello, desde consola realizamos:

```
nhopkg -s paquete o argumento-de-busqueda
```

### Ejemplo:

```
nhopkg -s gimp
```

### Explicación:

El comando `-s` busca paquetes, con los argumentos expuesto, en nuestro repositorio, y en caso de encontrar alguna coincidencia, nos la muestra con una serie de informaciones, como el nombre, versión y revisión del paquete, la descripción, la última versión disponible, y la versión instalada (en caso de ser así).

### Creación de paquetes fuente (.src.nho) de forma automática.

En la versión 0.5, *Nhopkg* incorpora una nueva función para crear los paquetes fuente `.src.nho` de forma automática. Él solo es capaz de encontrar la variedad de información que necesita para la creación del paquete. Desde nombre, versión, licencia, descripción, hasta los comandos de compilación e instalación. Hay que destacar que por ahora solo funciona con algunos paquetes de software, y que no recoge las dependencias que necesita cada software (todavía).

```
nhopkg -A tu-tarball.tar.bz2 tu-otro-tarball.tgz
```

### Ejemplo:

```
nhopkg -AR gimp-2.6.3.tar.bz2
```

### Explicación:

El comando `-A` crea automáticamente un paquete de fuentes `.src.nho` a partir de un *tarball*.

La opción `-R` impide que *Nhopkg* te pregunte por los datos que fue incapaz de obtener. Si no se usa esa opción te preguntará por aquellos datos que no pudo obtener.

**Aclaración:**

Aunque esta opción crea paquetes completamente compatibles, es una opción todavía en fase de desarrollo constante, por lo que si queremos crear un paquete fuente más completo, deberemos usar la siguiente opción.

**Creación de paquetes fuente (.src.nho) de forma manual.**

A parte de la creación de paquetes fuente de forma automática, *Nhopkg* provee de otra opción para crearlos de forma manual, lo que provoca que el paquete sea creado con más garantía de ser útil. En esta opción, *Nhopkg*, a través de un asistente, pregunta al usuario acerca de datos importantes para la construcción del paquete. Entre otras cosas pregunta el nombre, versión, revisión, licencia, descripción, grupo, web del autor, dependencias requeridas y opcionales, tanto para compilación como para que se ejecute después. Además se pueden añadir parches o archivos externos al paquete si es necesario. Por último preguntará los comandos de compilación, instalación, configuración de la instalación y configuración de la eliminación.

Es importante responder lo más rigurosamente posible todas las preguntas que nos hace el asistente, para así crear un buen paquete.

```
nhopkg -c tu-tarball.tar.bz2 tu-otro-tarball.tgz
```

**Ejemplo:**

```
nhopkg -c gimp-2.6.3.tar.bz2
```

**Explicación:**

El comando `-c` ejecuta un asistente para crear un paquete de fuentes desde el *tarball.tar.bz2* usado.

## OTRAS CARACTERÍSTICAS

### Comandos

- i, --install**  
Instala un paquete binario.
- S, --super-install**  
Instala un paquete binario desde repositorio.
- d, --dios**  
Igual que -S, --super-install.
- r, --remove**  
Elimina un paquete instalado por *Nhopkg*.
- B, --backup**  
Crea un paquete binario desde un paquete instalado.
- b, --build**  
Compila e instala un paquete de fuentes.
- k, --check**  
Comprueba que el paquete instalado es correcto.
- c, --create-source**  
Crea un paquete de fuentes desde un *tarball*.
- A, --auto-package**  
Crea automáticamente un paquete de fuentes desde un *tarball*.
- z, --tgz2nho**  
Convierte un paquete .tgz en otro .nho.
- g, --create-repo**  
Crea un repositorio de software en el directorio seleccionado.
- U, --update**  
Actualiza la base de datos del repositorio.
- e, --clean**  
Elimina datos en cache.
- s, --search**  
Busca paquetes y sus detalles.
- t, --list-repo**  
Muestra los paquetes en el repositorio y sus detalles.
- l, --list**  
Muestra los paquetes instalados.
- n, --info**  
Muestra detalles del paquete.
- w, --show**  
Muestra archivos del paquete.
- u, --update-db**  
Actualiza la base de datos de *Nhopkg*.
- license**  
Muestra la licencia de *Nhopkg*.
- license-all**

Muestra la licencia completa de *Nhopkg*.

**--version**

Muestra la versión de *Nhopkg*.

**-h, --help**

Muestra las páginas de ayuda de *Nhopkg*.

## Opciones

**-R, --recursive**

*Nhopkg* trabaja recursivamente.

**--root**

Establece raíz de instalación.

**--no-check-deps**

No comprueba dependencias.

**--no-check-arch**

No comprueba arquitectura.

**--no-check-md5**

No comprueba suma MD5.

**--force-check-deps**

Fuerza comprobación de dependencias.

**--force-check-arch**

Fuerza comprobación de la arquitectura.

**--force-check-md5**

Fuerza comprobación de la suma MD5.

**--purge**

Elimina dependencias inversas del paquete.

**-o, --output**

Escribe log de la salida.

**-v, --verbose**

Operaciones verbosas.

**--**

Para de obtener argumentos.

