

# Instalación automatizada de Sistemas Operativos y Aplicaciones

**Rafael Jurado Sánchez**  
**2016**

## Índice

Qué necesitamos	2
Primeros pasos: Conceptos	2
Proporcionando la respuesta: semillas	3
Detalles del archivo seed	5
Preparando el archivo ISO	7
Automatización de la instalación de aplicaciones	11
Automatización de instalaciones en Windows con Ninite	11
Automatización de instalaciones en Linux	12
Bibliografía	16
Anexo I: relación de carga de trabajo	17

## Qué necesitamos

- Un sistema Linux sobre el que trabajar
  - Una imagen .ISO del sistema a instalar
  - ISO Master para la edición de la imagen .ISO
  - Editor de textos
- 
- Sistema elegido para autoinstalación: Linux Mint 18
  - Instalación sobre VirtualBox 5.1 en CentOS 7

## Primeros pasos: Conceptos

La instalación automatizadas o desatendida, tiene múltiples utilidades sobretodo en el entorno laboral. Una oficina con varios equipos similares, utilizados por personas que desempeñan funciones similares (administrativos, contables, diseñadores...), o un data center con decenas de miles de equipos, son ejemplos clásicos de utilización para las instalaciones desatendidas. No es práctico, ni viable, instalar cada equipo a mano uno a uno, si no que se precisan herramientas para que todo sea lo más paralelizable posible.

Para ello se preparan tanto las instalaciones de archivos (imágenes de instalación, scripts de configuración) como las infraestructuras necesarias (redes, distribuidores...). Sin embargo, algo de esa envergadura queda fuera de las posibilidades de este documento, ya que no entraremos en las instalaciones remotas ni los recursos compartidos en un servidor mediante particiones nfs. En este documento ejemplificaremos de la manera reducida la mecánica y funcionamiento, utilizando un sistema Linux Mint 18 de 32 bits preparado para que su instalación no requiera de más intervención humana que el introducir el cd o usb con el archivo de imagen debidamente montado.

Una instalación automatizada consiste, básicamente, en conseguir que la instalación de un sistema operativo se realice en su totalidad sin la intervención de ningún administrador, al menos en sus partes más básicas. En este documento nos centraremos en la consecución de una instalación automatizada de Linux Mint 18, un sistema operativo de código abierto basado en Ubuntu que, a su vez, se basa en Debian.

Esto último es importante porque el procedimiento que permite la instalación automatizada del sistema depende de la tecnología implementada para la instalación de los sistemas Debian: preseeding.

Preseeding consiste en una serie de órdenes que permiten dar respuesta preestablecida a las preguntas que se realizan durante el proceso de instalación, eliminando la necesidad de un interlocutor humano que las proporcione en el momento en que se formulan dichas preguntas. Además, algunas características de la configuración de la instalación, que no suelen estar

disponibles durante la misma, pueden ser definidas. Las preguntas las realiza el Instalador Debian (DebianInstaller) y las respuestas pasan a la base de datos defconf.

Posteriormente, una vez instalado el sistema operativo, realizaremos una breve configuración automática de las aplicaciones necesarias, instalando unas y desinstalando otras, simulando una instalación personalizada para unos requerimientos concretos.

## Proporcionando la respuesta: semillas

En primer lugar necesitamos una imagen .ISO del sistema elegido. Se ha descargado para ello la versión 32 bits de Linux Mint 18, pero este documento podría aplicarse a cualquier distribución basada en Ubuntu.

Una vez con la imagen en nuestro disco duro se debe crear un cd/dvd o usb de arranque. Para ello hay diversas herramientas en todos los sistemas operativos (por ejemplo Rufus en Windows y Grabador de imágenes USB incluido con cada instalación Linux Mint) y son de uso sencillo. No entraremos en este aspecto porque, para facilitar la demostración en vídeo que se adjunta al final del documento, se ha utilizado una máquina virtual de Virtual Box corriendo sobre CentOS 7.

Cabe decir que para utilizar el siguiente tutorial en un equipo real que equipe bios UEFI o EFI deberá tenerse en cuenta este aspecto a la hora de crear el dispositivo de arranque, para lo cual hay documentación amplia y detallada para todos los sistemas, aunque quizá bajo sistemas Linux sea algo más complejo.

Con el dispositivo creado, podemos pasar ahora a crear el archivo .seed que será incluido en el dispositivo de arranque posteriormente.

Preseeding podría traducirse por “antes del sembrado”, es decir, consideremos que el proceso de instalación es un proceso “siembra” en nuestro equipo que dará como fruto un sistema operativo útil. Necesitamos las “semillas” necesarias, que son las respuestas a las necesidades del instalador y que estarán contenidas en un archivo .seed.

El archivo seed se almacena en el directorio preseed del dispositivo de instalación (sea este cd, dvd o un usb). A continuación figura la configuración del archivo .seed empleado en la instalación:

```
# Preseeding only locale sets language, country and locale.
d-i debian-installer/locale string es_ES

# Keyboard selection.
d-i keyboard-configuration/xkb-keymap select es

# netcfg will choose an interface that has link if possible. This makes it
# skip displaying a list if there is more than one interface.
```

```

d-i netcfg/choose_interface select auto

# Any hostname and domain names assigned from dhcp take precedence over
# values set here. However, setting the values still prevents the questions
# from being shown, even if values come from dhcp.
d-i netcfg/get_hostname string equipo
d-i netcfg/get_domain string oficina

# If non-free firmware is needed for the network or other hardware, you can
# configure the installer to always try to load it, without prompting. Or
# change to false to disable asking.
d-i hw-detect/load_firmware boolean true

# To create a normal user account.
d-i passwd/user-fullname string Usuario Normal
d-i passwd/username string miusuario
# Normal user's password, either in clear text
d-i passwd/user-password password al286335
d-i passwd/user-password-again password al286335

# Alternatively, you may specify a disk to partition. If the system has only
# one disk the installer will default to using that, but otherwise the device
# name must be given in traditional, non-devfs format
# The presently available methods are:
# - regular: use the usual partition types for your architecture
d-i partman-auto/method string regular

# You can choose one of the three predefined partitioning recipes:
# - atomic: all files in one partition
# - home:    separate /home partition
# - multi:   separate /home, /var, and /tmp partitions
d-i partman-auto/choose_recipe select atomic

### Package selection
tasksel tasksel/first multiselect standard, cinnamon-desktop, ssh-server

# Whether to upgrade packages after debootstrap.
# Allowed values: none, safe-upgrade, full-upgrade
d-i pkgsel/upgrade select full-upgrade

# This is fairly safe to set, it makes grub install automatically to the MBR
# if no other operating system is detected on the machine.
d-i grub-installer/only_debian boolean true

# This one makes grub-installer install to the MBR if it also finds some other
# OS, which is less safe as it might not be able to boot that other OS.
d-i grub-installer/with_other_os boolean false

# Avoid that last message about the install being complete.
d-i finish-install/reboot_in_progress note

```

Como vemos, el código abunda en comentarios. No en vano en la documentación de preseed de Debian podemos encontrar un ejemplo extenso de archivo .seed, del que generamos el que hemos usado. Sin embargo, vamos a realizar una descripción breve de cada parte de este código de ejemplo, que no es si no un extracto parcial del archivo final utilizado.

## Detalles del archivo seed

- Selección de idioma local y de teclado

```
# Preseeding only locale sets language, country and locale.
d-i debian-installer/locale string es_ES

# Keyboard selection.
d-i keyboard-configuration/xkb-keymap select es
```

- Detección del dispositivo de red y establecimiento de nombre para el equipo en el dominio de red.

```
# netcfg will choose an interface that has link if possible. This makes it
# skip displaying a list if there is more than one interface.
d-i netcfg/choose_interface select auto

# Any hostname and domain names assigned from dhcp take precedence over
# values set here. However, setting the values still prevents the questions
# from being shown, even if values come from dhcp.
d-i netcfg/get_hostname string equipo
d-i netcfg/get_domain string oficina
```

- Instalación de firmware no libre para dispositivos hardware. Especialmente útil para ciertos dispositivos, como tarjetas wifi, que no disponen de un firmware libre en la base de datos de los repositorios de Linux Mint, Ubuntu o Debian.

```
# If non-free firmware is needed for the network or other hardware, you can
# configure the installer to always try to load it, without prompting. Or
# change to false to disable asking.
d-i hw-detect/load_firmware boolean true
```

- Creación de cuentas de usuario no administrador. Este paso es delicado, ya que es posible introducir las claves de usuario en texto plano, algo que hemos hecho aquí por motivos didácticos, algo totalmente desaconsejable en una instalación real dado que dichas claves, aunque en el sistema se almacenarían cifradas permanecerían en texto plano en el archivo seed, que puede ser visto por cualquiera con acceso al dispositivo de instalación. En el caso de necesitar crear una clave cifrada, puede usarse el comando `openssl passwd -crypt myPassword` para conseguir una clave cifrada.

```
# To create a normal user account.
d-i passwd/user-fullname string Usuario Normal
d-i passwd/username string miusuario
# Normal user's password, either in clear text
d-i passwd/user-password password a1286335
d-i passwd/user-password-again password a1286335
```

- Tipo de particionado para la instalación. Se pueden especificar diferentes tipos de formato para las particiones de la instalación, o dejar el formato estándar. Además se puede configurar realizar una partición separada para el directorio `/home` u otros. Sin embargo, debido al uso de Ubiquity por parte de Linux Mint 18 (a consecuencia de estar basado en

Ubuntu) no se recomienda utilizar dichas opciones puesto que pueden dar lugar a fallos, dejando en su lugar la opción `atomic` por defecto.

```
# Alternatively, you may specify a disk to partition. If the system has only
# one disk the installer will default to using that, but otherwise the device
# name must be given in traditional, non-devfs format
# The presently available methods are:
# - regular: use the usual partition types for your architecture
d-i partman-auto/method string regular

# You can choose one of the three predefined partitioning recipes:
# - atomic: all files in one partition
# - home:    separate /home partition
# - multi:   separate /home, /var, and /tmp partitions
d-i partman-auto/choose_recipe select atomic
```

- Selección de paquetes. Podemos seleccionar entre uno o varios paquetes de instalación, que incluyen lo necesario para entornos gráficos y otras funciones. El mínimo recomendable es `standard`, pero existen las siguientes opciones:
  - **standard** (standard tools)
  - **desktop** (graphical desktop)
  - **gnome-desktop** (Gnome desktop)
  - **xfce-desktop** (XFCE desktop)
  - **kde-desktop** (KDE desktop)
  - **cinnamon-desktop** (Cinnamon desktop)
  - **mate-desktop** (MATE desktop)
  - **lxde-desktop** (LXDE desktop)
  - **web-server** (web server)
  - **print-server** (print server)
  - **ssh-server** (SSH server)

```
### Package selection
tasksel tasksel/first multiselect standard, cinnamon-desktop, ssh-server
```

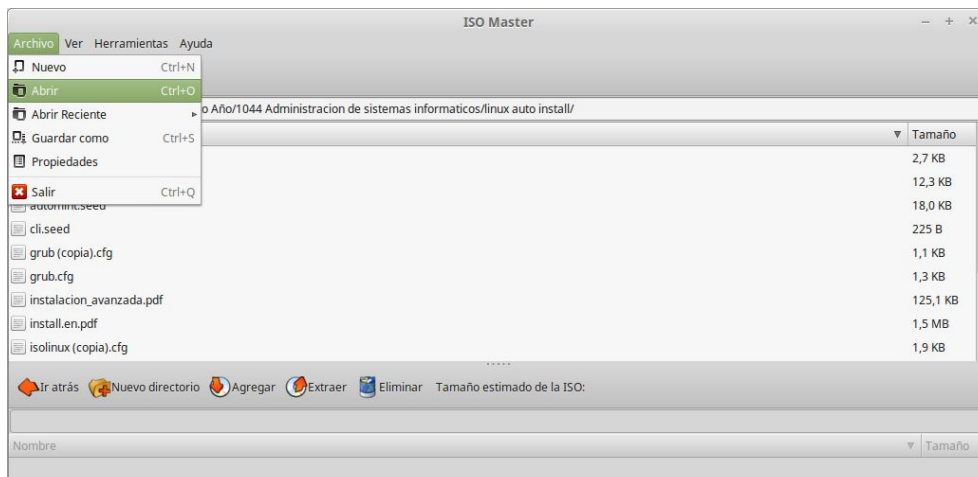
- Selección de tipo de grub. El gestor de arranque puede ser configurado, como en el ejemplo, para que detecte si se instala automáticamente sobre el MBR en solitario o si deja espacio en el sector de arranque para otro sistema operativo.

```
# This is fairly safe to set, it makes grub install automatically to the MBR
# if no other operating system is detected on the machine.
d-i grub-installer/only_debian boolean true

# This one makes grub-installer install to the MBR if it also finds some other
# OS, which is less safe as it might not be able to boot that other OS.
d-i grub-installer/with_other_os boolean false
```

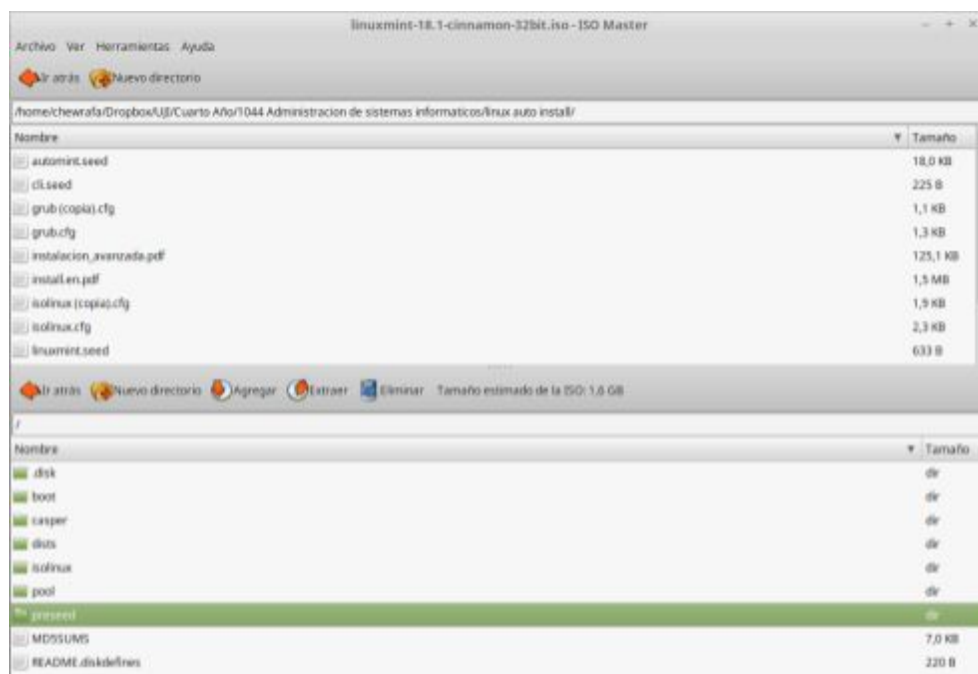
## Preparando el archivo ISO

Hemos completado el primer paso, ahora quedan algunos detalles más que completar. Vamos a emplear ahora dos herramientas diferentes pero muy sencillas: ISO Master y Gedit, ambos disponibles en los repositorios de Linux Mint y Ubuntu.



*Vista general de ISO Master*

Una vez instalado ISO Master abrimos imagen iso tal y como se ilustra en la imagen previa. Seleccionamos el archivo ISO que hemos descargado previamente y la ventana de la aplicación debería quedar tal y como puede verse en la imagen siguiente.

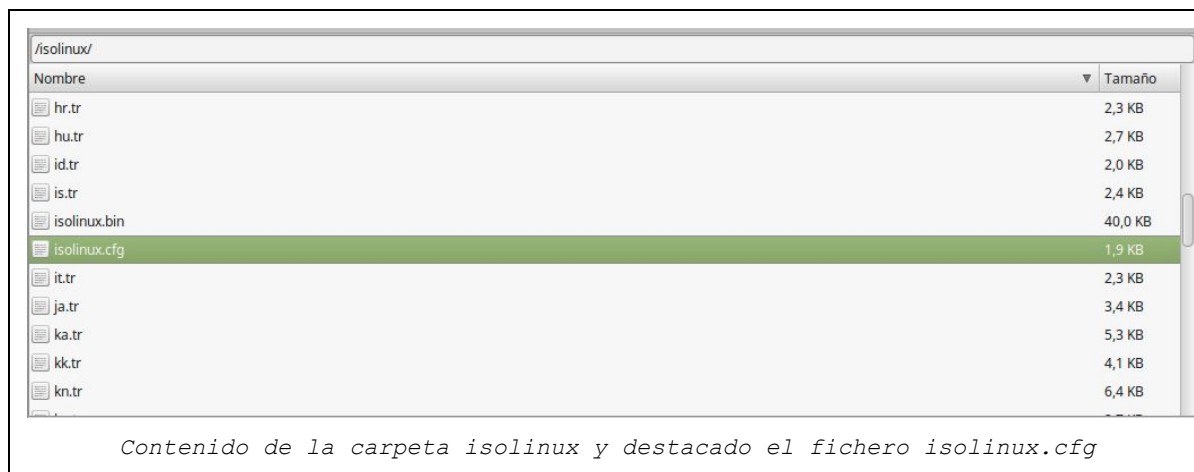


*ISO Master tras cargar el archivo ISO*



En la mitad superior tenemos el navegador de archivos del disco duro y en la inferior la navegación por la estructura de archivos de la imagen. La interfaz es sencilla de entender y de utilizar. Navegaremos en la mitad superior para encontrar los archivos que queremos añadir al archivo ISO o bien para elegir la carpeta donde queremos exportarlos desde dicho archivo.

Debido a la naturaleza de Linux Mint, hay que agregar una serie de modificaciones al archivo `isolinux.cfg` que se encuentra en la carpeta `isolinux` en la raíz del dispositivo de instalación que tenemos preparado.



Extraemos el archivo `isolinux.cfg` que se encuentra en la carpeta del mismo nombre en nuestro dispositivo y lo editamos en Gedit o cualquier otro editor de textos planos. Este archivo contiene la configuración del menú de arranque de la instalación, al que necesitamos hacer algunas modificaciones. Para empezar, localizamos la línea:

```
menu default
```

Una vez localizada la borraremos, puesto que esta línea identifica la opción por defecto que tendrá seleccionado el menú al arrancar y la que se ejecutará por defecto al acabar la cuenta atrás. Lo ideal sería eliminar dicha cuenta atrás, que en este documento se ha dejado por defecto para que pueda verse la opción añadida al menú. Si desea eliminarse el tiempo de espera hay que localizar en el inicio del documento la línea:

```
timeout 100
```

En esta ocasión si deseamos que no haya espera hemos de poner `timeout` a 0, sin eliminar la línea. Ahora tenemos que localizar la configuración de la opción que previamente era la que figuraba por defecto e insertar nuestra opción justo antes, con lo que quedaría una configuración con el siguiente aspecto:

```
label auto
  menu label AUTOINSTALL
```

```

kernel /casper/vmlinuz
append file=/cdrom/preseed/automint.seed auto=true priority=critical
debian-installer/locale=es_ES keyboard-configuration/layoutcode=es ubiquity/reboot=true
languagechooser/language-name=Spanish countrychooser/shortlist=ES
localechooser/supported-locales=es_ES.UTF-8 boot=casper automatic-ubiquity
initrd=/casper/initrd.lz quiet splash noprompt noshell--
menu default

label live
menu label Start Linux Mint
kernel /casper/vmlinuz
append file=/cdrom/preseed/linuxmint.seed boot=casper initrd=/casper/initrd.lz quiet
splash --

```

Ni siquiera es necesario conservar el resto de opciones si no queremos, se han dejado aquí como futura referencia. Sin embargo, en términos de profesionalidad, sería conveniente al menos comentar todas aquellas opciones que no se utilizan.

La línea `append` es la más interesante. En ella se especifica la ruta hacia el archivo `.seed` y una serie de comandos de Ubiquity. Ubiquity usa un subconjunto de componentes de Debian Installer que utilizan los componentes de la versión de Ubuntu y que realizan las mismas preguntas que el instalador Debian. Algunos interesantes son:

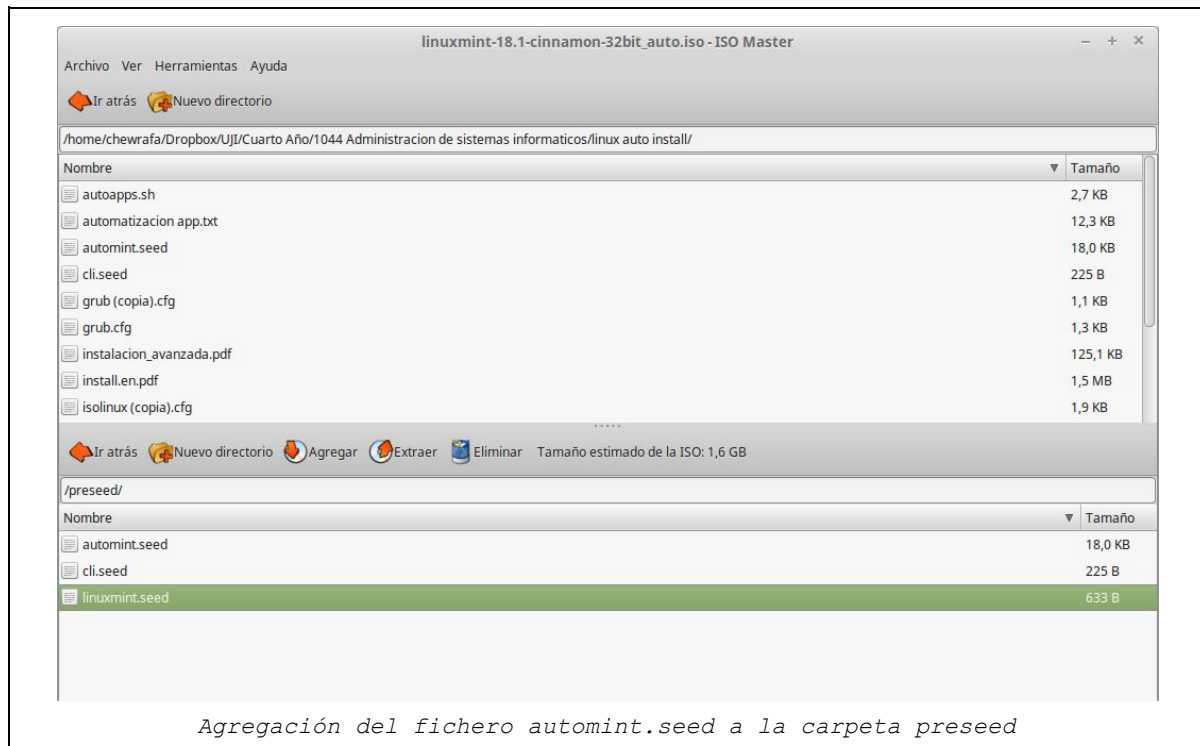
- **ubiquity/reboot:** Reinicia automáticamente cuando el instalador completa su tarea. Si se añade “`noprompt`” evitamos el mensaje de “extraiga el cd de la unidad”.
- **ubiquity/failure\_command:** Especifica un comando que se ejecutará si falla el instalador.
- **ubiquity/success\_command:** De manera similar al comando de `preseed` “`preseed/late`”, especifica un comando que se ejecutará si el instalador tiene éxito.

En este caso hemos usado otros más similares a los requeridos por Debian Installer:

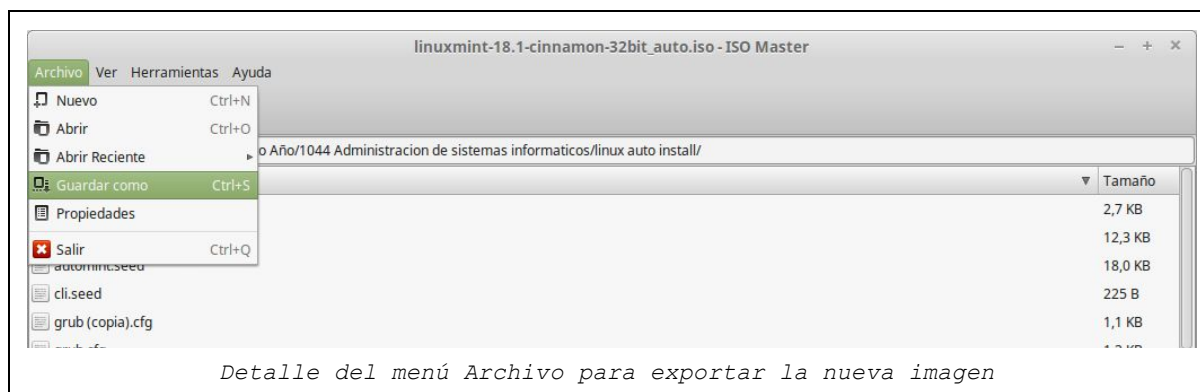
- **languagechooser/language-name:** especifica el lenguaje.
- **countrychooser/shortlist:** elección de país, territorio o área.
- **localechooser/supported-locales:** selección de conjunto de caracteres locales soportados.

Una vez finalizada la edición, volvemos a ISO Master. En la parte inferior de la aplicación eliminamos el archivo `isolinux.cfg` original, seleccionamos el modificado con el mismo nombre de la parte superior de la aplicación y lo movemos a la imagen. No hay que preocuparse por no conservar el archivo original, ya que exportaremos la nueva imagen con otro nombre para conservar la que hemos descargado sin modificar.

Queda un segundo archivo a añadir a la imagen, en este caso el `.seed`. La mayoría de su contenido lo hemos descrito anteriormente, pero el archivo completo utilizado en este documento puede desargarse para su uso. En este caso el archivo lo agregaremos dentro de la carpeta `preseed`, quedando como sigue:



El proceso ya casi está completo. Sólo queda exportar nuestra nueva imagen ISO con las modificaciones añadidas. Para ello simplemente vamos a Archivo y seleccionamos Guardar Como, introduciendo un nombre distinto al de la imagen ISO original.



Tras unos minutos (dependiendo de la potencia del equipo donde se esté trabajando) tendremos la nueva imagen lista para utilizar. En el vídeo enlazado en el Anexo puede observarse el proceso completo de instalación sobre una máquina virtual. Este se prolonga bastante debido a que el equipo donde se realiza la experiencia no está pensado para alojar un sistema exigente como es Linux Mint, pero tampoco está pensado este documento para generar imágenes destinadas a ser usadas en máquinas virtuales.

## Automatización de la instalación de aplicaciones

### Automatización de instalaciones en Windows con Ninite

Salimos momentáneamente del ámbito Linux para centrarnos brevemente en Windows. Es posible la instalación automatizada de aplicaciones sobre este sistema gracias a herramientas como los scripts de VisualBasic o PowerShell que, pese a no ser igual de potente que Bash Shell, se queda cerca. Sin embargo para el usuario medio esto no es útil ni cómodo. Sin embargo existen alternativas como Ninite.

Ninite es una aplicación de escritorio Windows que, sin embargo, se configura desde web. En la misma elegimos aquellas aplicaciones que queremos que sean descargadas e instaladas automáticamente (como vemos en este caso estamos limitados a las que nos ofrece).



Tras elegir las, pulsamos en Get Your Ninite y nos descargará un archivo ejecutable de Windows. Al utilizarlo, nos descargará las aplicaciones seleccionadas, libres de software extra (como barras de búsqueda en el navegador) y con la versión adecuada a nuestra máquina (32 ó 64 bits). Si ejecutamos el fichero en posteriores ocasiones, nos actualizará dichas aplicaciones.

Ninite es sin duda un método fácil y cómodo para necesidades generalistas porque, aunque limitado a aquellas aplicaciones que figuran en la web, cubre los requerimientos más usuales en equipos de uso general (como oficinas o personales).

### Automatización de instalaciones en Linux

El proceso de instalar aplicaciones en Linux se ha hecho sencillo con el tiempo gracias a la aparición de aplicaciones gráficas que facilitan a cualquier usuario la elección de aquellos programas que necesita. Pero por debajo, lo que está ocurriendo es lo que los usuarios más avanzados ya conocen: ejecución de comandos en consola. En este caso nos centraremos en el

caso de sistemas basados en Debian y APT. Para sistemas basados en Ubuntu hay una opción similar a Ninite llamada Ubuntu After Install, cuyo enlace puede encontrarse al final de este documento pero cuyo uso queda fuera de nuestro propósito.

Vamos a crear un script que podrá ser ejecutado en nuestro sistema Linux Mint 18 recién instalado. Aunque sería posible hacer que este script estuviera incluido en la instalación, podría ser más flexible tenerlo aparte de la misma, dado que aunque en un entorno de oficina la base instalada sea la misma, puede que varíe de unos equipos a otros, por ejemplo entre departamentos.

Un script es simplemente un archivo que contiene órdenes que se ejecutarán de manera secuencial y desatendida. En este caso crearemos uno con Gedit, aunque cualquier otro editor de textos planos serviría igualmente. Lo llamaremos `autoapps.sh`:

- En primer lugar actualizamos los paquetes del sistema y sus referencias

```
#!/bin/bash

#Autor: Rafael Jurado Sanchez
# Script para instalacion de aplicaciones automaticamente en Linux Mint 17.3 tras
una instalacion limpia

#Paso 0: actualizacion de paquetes del sistema

echo < actualizacion de paquetes >
echo -----
sudo apt-get update -y
```

- A continuación creamos una carpeta para la descarga de paquetes, por si fuera necesaria. Como ejemplo la emplearemos para descargar JDK8 y la utilizaremos como directorio para que permanezca instalado dicho paquete.

```
#Paso 1: creacion de directorio para descarga de paquetes si es necesario
clear
echo < creacion de directorio Utils >
echo -----
mkdir /home/utils

#Pausa para ver la salida por terminal, comentar para automatizacion
read -p "Pulse tecla para continuar"

#Paso 2: descarga e instalacion de jdk8 (ultima version a fecha 20-12-2016)
clear
echo < instalacion de jdk8 - v111 20-12-2016 >
echo -----
cd /home/utils
wget
http://download.oracle.com/otn-pub/java/jdk/8u111-b14/jdk-8u111-linux-x64.tar.gz
tar zxvf jdk-8u111-linux-x64.tar.gz
rm jdk-8u111-linux-x64.tar.gz
```

- Ahora agregamos repositorios que puedan ser necesarios para la futura instalación y actualización de aplicaciones. En este caso agregamos los repositorios de Filezilla y actualizamos las referencias de APT.

```
#Paso 3: agregacion de repositorios (descomentar y configurar segun se necesite)
#sudo sh -c 'echo "deb http://.../" >> /etc/apt/sources.list'
#agregacion del repositorio de filezilla
sudo add-apt-repository ppa:adabbas/1stppa
sudo apt-get update
```

- En este paso desinstalaremos las aplicaciones que no se consideran necesarias. En un equipo de oficina, por ejemplo, nos desharíamos de aplicaciones de dibujo, chat y multimedia. A la hora de generar el script es útil conocer qué aplicaciones se encuentran instaladas en el equipo, por lo que sería conveniente ejecutar `dpkg -l > listaaplicaciones.txt` que nos generará una lista de esas aplicaciones en el archivo especificado.

```
#Paso 4: desinstalacion de aplicaciones innecesarias
clear
echo < desinstalacion de aplicaciones innecesarias >
echo -----
sudo apt-get autoremove -y gimp*
sudo apt-get autoremove -y banshee*
sudo apt-get autoremove -y transmission-common*
sudo apt-get autoremove -y transmission-gtk*
sudo apt-get autoremove -y pidgin*
sudo apt-get autoremove -y pidgin-data*
```

- Tras la desinstalación, la instalación. En esta ocasión instalaremos Filezilla como ejemplo, aunque podría ser cualquier aplicación presente en los repositorios. También volvemos a actualizar las referencias de APT.

```
#Paso 5: instalacion de aplicaciones requeridas
clear
echo < instalacion de aplicaciones requeridas >
echo -----
sudo apt-get install -y filezilla

#Paso 6: nueva actualizacion de paquetes
echo < actualizacion de paquetes >
echo -----
sudo apt-get update -y
```

- Aunque se ha especificado en la instalación, puede que hayan paquetes que necesitemos para dispositivos y que no se hayan instalado. Para eso caso se incluye el siguiente comando. Volveremos a actualizar las referencias de APT.

```
#Paso 7: instalacion de firmware no libre
sudo apt-get install -y linux-firmware-nonfree

#Paso 8: nueva actualizacion de paquetes
```

```
echo < actualizacion de paquetes >
echo -----
sudo apt-get update -y
```

- Este último paso es opcional y por ello se haya comentado. Es posible configurar todos los aspectos del escritorio cinnamon instalado con Linux Mint 18 (al igual que es posible hacerlo con Gnome, KDE, etc). Ejecutando `gsettings list-schemas` es posible saber todos los parámetros configurables en el sistema. Aquí reproducimos una breve muestra de los primeros que aparecen en la lista en Linux Mint y, posteriormente, la parte final opcional de nuestro script automatizado:

```
org.gnome.gthumb.picasaweb
org.cinnamon.desktop.thumbnailers
org.gnome.system.proxy.https
org.cinnamon.settings-daemon.plugins.screensaver-proxy
org.cinnamon.desktop.default-applications.office.tasks
org.gnome.gthumb.convert-format
org.gnome.settings-daemon.plugins.cursor
org.nemo.preferences
org.cinnamon.desktop.keybindings.wm
org.cinnamon.keybindings
org.cinnamon.muffin
org.cinnamon.desktop.a11y.keyboard
org.gnome.gthumb.importer
org.cinnamon.settings-daemon.plugins.sound
org.gnome.eog.fullscreen
org.gnome.gthumb.image-viewer
org.cinnamon.settings-daemon.peripherals.touchscreen
org.cinnamon.settings-daemon.plugins.power
```

```
#Paso 9 (opcional): configuracion de cinnamon
#clear
#echo < configuracion de cinnamon >
#echo -----
#cambio de fondo de escritorio
#gsettings set org.cinnamon.desktop.background picture-uri "file:///filename"
```

Como paso final, una vez finalizado el archivo y guardado, hay que hacer que sea posible su ejecución, bien sea en el menú Propiedades del fichero o por consola estando en la carpeta donde se encuentra el fichero:

```
chmod +x autoapps.sh
```

Y para ejecutarlo manualmente:

```
./autoapps.sh
```





## Bibliografía

Preseeding: <http://d-i.alioth.debian.org/manual/en.i386/apbs01.html>

Ubiquity: <https://wiki.ubuntu.com/UbiquityAutomation>

Creación de instalaciones Linux desatendidas:

- <http://askubuntu.com/questions/806820/how-do-i-create-a-completely-unattended-install-of-ubuntu-desktop-16-04-1-lts>
- <https://community.linuxmint.com/tutorial/view/54>
- <http://hands.com/d-i/>
- [https://docs.moodle.org/dev/Table\\_of\\_locales](https://docs.moodle.org/dev/Table_of_locales)
- <https://wiki.ubuntu.com/Installer/FAQ>

Instalación automatizada de aplicaciones:

- <http://blog.self.li/post/74294988486/creating-a-post-installation-script-for-ubuntu>
- Ubuntu After Install: <https://www.thefanclub.co.za/how-to/ubuntu-after-install>
- <https://ninite.com/>

Imagen Linux Mint: <https://www.linuxmint.com/download.php>

ISO Master: <http://littlesvr.ca/isomaster/>

Archivos de este documento:

- automint.seed: <https://www.dropbox.com/s/26u1sy30uy23z8t/automint.seed?dl=0>
- isolinux.cfg: <https://www.dropbox.com/s/0kj8hjthn94wgaw/isolinux.cfg?dl=0>
- autoapps.sh: <https://www.dropbox.com/s/az6122lnz8dxr2g/autoapps.sh?dl=0>

## Anexo I: relación de carga de trabajo

Evento	Tiempo empleado
Búsqueda de documentación	15 horas
Filtrado de documentación	2 horas
Organización de la información y práctica	3 horas
Generación de código y pruebas	4 horas
Pruebas sobre máquina virtual	3 horas
<b>TOTAL</b>	<b>27 horas</b>