

## A. Tareas a realizar cuando instales Ubuntu 12/14/16.

1. Para poder cambiar la interfaz del escritorio usando el Gnome “clásico”  
`sudo apt-get install gnome-panel`
2. Para poder usar *codecs* propietarios (archivos MP3, M4A, MP4, WMV, etc.) y demás formatos  
`sudo apt-get install ubuntu-restricted-extras`
3. Para poder ver algunos videos y ver contenido web en flash en los exploradores web  
`sudo apt-get install flashplugin-installer`
4. En el servicio DNS, el resolutor (cliente DNS) lee la información del archivo `/etc/resolv.conf` para determinar los servidores DNS a los cuales preguntar para la resolución de nombres. En versiones anteriores a Ubuntu 12 la herramienta de configuración de red NetworkManager modificaba el contenido del archivo `/etc/resolv.conf` con la información de los servidores DNS que se les proporcionaba, pero a partir de Ubuntu 12 hasta Ubuntu 16 ya no lo hace por defecto.

El motivo es que estas nuevas versiones de Ubuntu vienen configuradas por defecto con un servidor DNS sólo caché en el mismo equipo (127.0.1.1) implementado por la aplicación `dnsmasq`. Y aunque con NetworkManager indiquemos cuales van a ser los servidores DNS, la aplicación mantiene que el servidor DNS es el proporcionado por `dnsmasq`.

Si no queremos que actúe de este modo, hay que deshabilitar este servidor DNS caché en la configuración de NetworkManager para que vuelva a escribir en `/etc/resolv.conf` los servidores DNS que le indiquemos. Para ello editamos el archivo de configuración de NetworkManager.conf

```
sudo gedit /etc/NetworkManager/NetworkManager.conf
```

y comentamos la línea

```
#dns=dnsmasq
```

Para que surja efecto volveremos a reiniciar el servicio de red a través del gestor NetworkManager

```
sudo restart network-manager
```

Ahora veremos que en `/etc/resolv.conf` sí aparece la IP del servidor DNS que indicamos con la aplicación NetworkManager.

5. La configuración de las interfaces de red pueden realizarse básicamente de dos formas. Una de ellas es editando de forma tradicional el archivo `/etc/network/interfaces` y luego activar o desactivar las interfaces de red. Un contenido típico del archivo `/etc/network/interfaces` sería éste:

```
auto eth0
iface eth0 inet dhcp
```

Por otro lado algunas distribuciones Linux con interfaz gráfica traen instalado la aplicación gráfica NetworkManager para configurar las interfaces de red. Por defecto, las configuraciones de las interfaces de red que gestiona NetworkManager NO son las mismas que aparecen en el archivo `/etc/network/interfaces`, sino son otras y sus datos de configuración los almacena en la carpeta `/etc/NetworkManager/system-connections`.

Este modo de trabajo que por defecto emplea NetworkManager es bastante adecuado ya que no mezcla las configuraciones manuales de las interfaces de red que tradicionalmente usa el servicio *networking* con las configuraciones del servicio proporcionado por NetworkManager.

No obstante este modo de trabajo puede cambiarse de forma que NetworkManager SI gestione y edite directamente las interfaces que se ubican en el archivo `/etc/network/interfaces`. Ahora hay que tener cuidado porque ambos servicios, el de red *networking* y el de NetworkManager trabajan sobre el mismo archivo de configuración y puede haber problemas de consistencia en los datos.

Las últimas versiones de NetworkManager están configuradas para que mediante diferentes *plugins* se pueda configurar las interfaces de red según las distintas distribuciones (Ubuntu, Red Hat, SUSE, etc.). El *plugin* para Ubuntu se denomina *ifupdown*, y podemos indicar si NetworkManager gestionará o no directamente las interfaces que aparecen en el archivo `/etc/network/interfaces`. Para ello, en el archivo `/etc/NetworkManager/NetworkManager.conf`, en la sección correspondiente a este *plugin*, se usa el parámetro *managed* con valores *true* o *false*. Por defecto, como hemos dicho, no las gestiona (*managed=false*).

Si deseamos que NetworkManager SI gestione y edite directamente las interfaces que se ubican en el archivo `/etc/network/interfaces`, debemos habilitar el plugin mediante (*managed=true*) y luego reiniciar el servicio de red a través del gestor NetworkManager

```
sudo restart network-manager
```

**IMPORTANTE. Sea cualesquiera el modo de configuración de las conexiones de red, la única información fiable de las conexiones existentes en un momento dado la proporciona el comando *ip addr show*.**

## B. Reinstalar grub2 usando grub-customizer.

Esto es sólo necesario en la máquina anfitrión para editar cómodamente el menú que presenta grub2.

- a) Instalación de grub-customizer
  - Añadimos repositorio  
`sudo add-apt-repository ppa:danielrichter2007/grub-customizer`
  - Actualizamos repositorio  
`sudo apt-get update`
  - Descarga e instalación del paquete  
`sudo apt-get install grub-customizer`
- b) Ejecuta el grub-customizer, editando las opciones que desees y termina salvando las modificaciones.
- c) Carga ahora el GRUB en el [MBR](#) del disco (es decir, instala el gestor de arranque al comienzo del primer disco duro `"/dev/sda"`).  
`grub-install --recheck /dev/sda`
- d) Actualiza el archivo de configuración de grub2 (`grub.cfg`)  
`sudo update -grub2`

## C. Cambio en los nombres de las interfaces de red.

A partir de la version 15 de Ubuntu, la nomenclatura de las interfaces de red ha cambiado de **eth0, eth1, wlan0, wlan1, a enp0s3, enp0s1, etc.**

Este cambio a *Predictable Network Interface Names*, pretende asignar identificadores estables a las interfaces de red basándose en el tipo: local Ethernet, Wlan, Wwan, etc. y evitar los problemas de la nomenclatura clásica.

El sistema empieza por intentar dar los nombres de los dispositivos en el siguiente orden:

- a) Nombres ordenados mediante un índice de los dispositivos que el Firmware/BIOS detecta en la placa base. Por ejemplo: `eno1`.
- b) Nombres ordenados mediante un índice de los dispositivos que el Firmware/BIOS detecta en las ranuras PCI Express de la placa base. Por ejemplo: `ens1`.
- c) Nombres ordenados mediante un índice según la localización del conector hardware del dispositivo. Por ejemplo: `enp0s3`.
- d) Si las asignaciones anteriores fallan, se pueden incorporar la dirección MAC como nombre. Por ejemplo: `enx78e7d1ea46da`.
- e) Si no se pueden aplicar ninguna de las políticas anteriores, se utiliza la **nomenclatura tradicional** del kernel. Por ejemplo, `eth0, eth1, wlan0, etc.`

Además con la plena implementación de *systemd* (nuevo sistema de inicio que sustituye al clásico *System V init*), el comando *ifconfig* ha quedado obsoleto y es reemplazado por el comando *ip*. Este es el cuadro comparativo de comandos:

<b>Acción</b>	<b>Ifconfig</b>	<b>IP</b>
Mostrar los dispositivos de red	<code>sudo ifconfig</code>	<code>ip addr show // ip link show</code>
Activar interface de red	<code>sudo ifconfig eth0 up</code>	<code>sudo ip link set enp0s3 up</code>
Desactivar interface de red	<code>sudo ifconfig eth0 down</code>	<code>sudo ip link set enp0s3 down</code>
Establecer dirección IP	<code>sudo ifconfig eth0 192.168.1.1</code>	<code>sudo ip address add 192.168.1.1 dev enp0s3</code>
Eliminar dirección IP		<code>sudo ip address del 192.168.1.1 dev enp0s3</code>
Añadir interface virtual	<code>sudo ifconfig eth0:1 10.0.0.1/8</code>	<code>sudo ip addr add 10.0.0.1/8 dev enp0s3 label enp0s3:1</code>
Cambiar a off dispositivo	<code>sudo ifconfig -arp eth0</code>	<code>sudo ip link set dev enp0s3 arp off</code>