



# Módulo de curso **LINUX**

Departamento de Informática

Departamento de Informática





## Departamento de Informática



### Consideraciones

- ❖ Este módulo tiene por finalidad el seguimiento de las clases a medida que el instructor desarrolla los diferentes contenidos del curso.
- ❖ Los temas enunciados en el módulo pueden ser utilizados en más de una oportunidad en el desarrollo del curso a través de ejemplos que el instructor dará.
- ❖ Este módulo NO es suficiente para la preparación del examen, ya que la explicación de cómo se realiza cada acción está a cargo del instructor.
- ❖ En caso de rendir el examen en forma libre se recomienda consultar los contenidos mínimos y la bibliografía complementaria.
- ❖ Para rendir el examen regular se cuenta con 6 (seis) meses a partir de la finalización del curso y hasta 3 oportunidades durante este período de tiempo, luego de lo cual se producirá el vencimiento del mismo.
- ❖ Le recomendamos que lea el Reglamento del Sistema de Créditos en Informática que se encuentra disponible en <http://www.ufasta.edu.ar>



Desarrollado para el Departamento de Informática

Módulo del Curso Linux

Clase 1.....	5
Objetivos.....	5
Datos sobre el curso.....	5
Introducción- Clase 01.....	6
Introducción y principios básicos.....	8
Prácticas importantes.....	9
Curso de Linux – Clase 02.....	10
Sistema Gráfico.....	10
Prácticas importantes.....	20
Curso de Linux – Clase 03.....	21
El poder de la consola.....	21
Hostname.....	21
w.....	21
id.....	21
date.....	21
man.....	21
ls.....	21
cd.....	21
mkdir.....	22
rm.....	22
cp.....	22
mv.....	23
find.....	23
clear.....	23
ps.....	23
kill.....	23
killall.....	24
sudo.....	24
passwd.....	24
su.....	24
sudo passwd.....	24
apt.....	24
aptitude.....	25
dpkg.....	26
alien.....	27
man.....	27



alias.....	27
Editores.....	28
Vi.....	28
Prácticas importantes.....	29
Curso de Linux – Clase 04.....	29
Administración de sistema.....	32
Prácticas importantes.....	32
Archivos de configuración del Sistema.....	33
Prácticas importantes.....	33
La Red.....	34
Dirección IP.....	34
Cambiar DNS.....	34
Administrar la red.....	35
Curso de Linux – Clase 05.....	36
Conexión a un servidor remoto.....	36
Prácticas importantes.....	37
Enviar u obtener archivos y carpetas con scp.....	37



## Clase 1

### Objetivos

Los objetivos de curso son:

1. Que los participantes se introduzcan en el sistema operativo Linux de forma sencilla y útil.
2. Que se logre incorporar la utilización del sistema operativo Linux en los hogares para realizar las mismas tareas de todos los días.
3. Que se conozcan las ventajas y diferencias del sistema operativo Linux

### Datos sobre el curso

El curso utilizará el sistema operativo Linux, distribución Kubuntu, desde un Live-CD, o CD booteable para la totalidad de las prácticas. El CD se puede obtener gratuitamente desde <http://www.kubuntu.org/download.php>

## Introducción - Clase 01

### ¿Qué es Linux?

Linux es un término genérico para referirse a sistemas operativos similares a Unix basados en el núcleo de Linux. Su desarrollo es uno de los ejemplos más prominentes de software libre; normalmente todo el código fuente puede ser utilizado, modificado y redistribuido libremente por cualquiera bajo los términos de la Licencia Pública General de GNU (GNU GPL) y otras licencias libres.

El nombre Linux viene del núcleo Linux, inicialmente escrito por Linus Torvalds en 1991. El software de sistema viene, normalmente, del sistema operativo GNU, iniciado por Richard Stallman en 1983 y mantenido por la FSF (Free Software Foundation). La contribución de GNU es la razón por la que algunos prefieren llamar GNU/Linux al sistema operativo formado por las herramientas de GNU y el núcleo Linux en su conjunto.

Que es Software Libre?

Software libre (en inglés free software) es la denominación del software que brinda libertad a los usuarios sobre su producto adquirido y por tanto, una vez obtenido, puede ser usado, copiado, estudiado, modificado y redistribuido libremente. Según la Free Software Foundation, el software libre se refiere a la libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software; de modo más preciso, se refiere a cuatro libertades de los usuarios del software.

De acuerdo con tal definición, el software es "libre" si garantiza las siguientes libertades:

- **Libertad 0:** la libertad de usar el programa, con cualquier propósito.
- **Libertad 1:** la libertad de estudiar cómo funciona el programa, y adaptarlo a tus necesidades.



- **Libertad 2:** la libertad de distribuir copias, con lo que puedes ayudar a tu vecino.
- **Libertad 3:** la libertad de mejorar el programa y hacer públicas las mejoras a los demás, de modo que toda la comunidad se beneficie.

Las libertades 1 y 3 requieren que esté disponible el código fuente porque estudiar y modificar software sin su código fuente es muy poco viable.

### ¿Qué son las distribuciones de Linux?

Una distribución es una variante del sistema Linux que se enfoca a satisfacer las necesidades de un grupo específico de usuarios. De este modo hay distribuciones para hogares, empresas y servidores. Algunas incorporan programas comerciales (como Mandriva PowerPack) o solamente software libre (como Debian).

Las distribuciones son ensambladas por individuos, empresas u otros organismos. Cada distribución puede incluir cualquier número de software adicional, incluyendo software que facilite la instalación del sistema. La base del software incluido con cada distribución incluye el núcleo Linux y las herramientas GNU, al que suelen añadirse también varios paquetes de software.

### ¿Qué es Ubuntu?

Ubuntu es una distribución GNU/Linux que ofrece un sistema operativo predominantemente enfocado a computadores personales, aunque también proporciona soporte para servidores. Es una de las más importantes distribuciones de GNU/Linux a nivel mundial. Se basa en Debian GNU/Linux y concentra su objetivo en la facilidad y libertad de uso, la fluida instalación y los lanzamientos regulares (cada 6 meses: las versiones .04 en abril y las .10 en octubre). El principal patrocinador es Canonical Ltd., una empresa privada fundada y financiada por el empresario sudafricano Mark Shuttleworth.

El nombre de la distribución proviene del concepto zulú y xhosa de ubuntu, que significa "humanidad hacia otros" o "yo soy porque nosotros somos". Ubuntu es un movimiento sudafricano encabezado por el obispo Desmond Tutu, quien ganó el Premio Nobel de la Paz en 1984 por sus luchas en contra del Apartheid en Sudáfrica. El sudafricano Mark Shuttleworth, mecenas del proyecto, se encontraba muy familiarizado con la corriente. Tras ver similitudes entre los ideales de los proyectos GNU, Debian y en general con el movimiento del software libre, decidió aprovechar la ocasión para difundir los ideales de Ubuntu. El eslogan de la distribución –“Linux para seres humanos” (en inglés "Linux for Human Beings")– resume una de sus metas principales: hacer de Linux un sistema operativo más accesible y fácil de usar.



## **Gnome y Kde**

Gnome y Kde son entornos de escritorio para sistemas operativos de tipo Unix/Linux.

El Proyecto GNOME, según sus creadores, provee un gestor de ventanas «intuitivo y atractivo» y una plataforma de desarrollo para desarrollar aplicaciones que se integran con el escritorio. El Proyecto pone un gran énfasis en la simplicidad, usabilidad y en hacer que las cosas funcionen. Otros objetivos del proyecto son:

La libertad para crear un entorno de escritorio que siempre tendrá el código fuente disponible para reutilizarse bajo una licencia de software libre.

El aseguramiento de la accesibilidad, de modo que pueda ser utilizado por cualquiera, sin importar sus conocimientos técnicos y discapacidad física.

Hacer que esté disponible en muchos idiomas. En el momento está siendo traducido a más de 100 idiomas.

Un ciclo regular de liberaciones y una estructura de comunidad disciplinada.

De acuerdo con su página web, «KDE es un entorno de Escritorio contemporáneo para estaciones de trabajo Unix. KDE llena la necesidad de un escritorio amigable para estaciones de trabajo Unix, similar a los escritorios de MacOSX o Windows».

KDE se basa en el principio de la personalización. Todos los componentes de KDE pueden ser configurados en mayor o menor medida por el usuario. Las opciones más comunes son accesibles en su mayoría desde menús y diálogos de configuración. Los usuarios avanzados pueden optar por editar los archivos de configuración manualmente, obteniendo en algunos casos un mayor control sobre el comportamiento del sistema.

## **Boot del Sistema y Live CD**

Un Live CD o Live DVD, más genéricamente Live Distro, es un sistema operativo (normalmente acompañado de un conjunto de aplicaciones) almacenado en un medio extraíble, tradicionalmente un CD o un DVD (de ahí sus nombres), que puede ejecutarse desde éste sin necesidad de instalarlo en el disco duro de una computadora, para lo cual usa la memoria RAM como disco duro virtual y el propio medio como sistema de ficheros.

Algunos Live CD incluyen una herramienta que permite instalarlos en el disco duro. Otra característica es que por lo general no se efectúan cambios en la computadora utilizada, aunque algunos pueden almacenar preferencias si así se desea.

Para usar un Live CD es necesario obtener uno (muchos de ellos distribuyen libremente una imagen ISO que puede bajarse de Internet y grabarse en disco) y configurar la computadora para que arranque desde la unidad lectora, reiniciando luego la computadora con el disco en la lectora, con lo que el Live CD se iniciará automáticamente.



### *Introducción y principios básicos*

1. Introducción al curso
  1. Metodología del curso, diagramación y prácticas
2. ¿Qué es Linux?
  1. La importancia de saber por qué usamos Linux. Free Software, sociedad, educación y libertad
3. ¿Qué es Debian?
  1. Nuestra distribución base, la historia de la unidad
  2. Miles de aplicaciones disponibles para lo que necesitamos
4. ¿Qué es Ubuntu?
  1. Cooperación y facilidad
5. ¿Qué es Kde?
  1. Sistemas gráficos, comparativa y diferencias
  2. Integración y relación completa entre componentes
6. ¿Qué es Kubuntu?
  1. La unión de Kde y Ubuntu
7. Boot del sistema
  1. Concepto de Live-CD
  2. Cambio de lenguaje
    1. Con F2 cambiamos el lenguaje de las aplicaciones y del teclado
8. Comenzar desde el principio. La ayuda.
  1. Khelpcenter. La Ayuda en el sistema Gráfico. Ver la Ilustración 1.
    1. Es la base de la ayuda de KDE gráficamente
  2. Ayuda en la consola. Comando 'man'. La ayuda más completa que hay.
    1. Para aprender a usar el comando man, hacer 'man man'
    2. Para aprender a usar el comando ls, hacer 'man ls'.
    3. Del comando man se sale con la tecla 'q'.
9. Multiusuario - Login en todas las consolas incluso mientras bootea
  1. Desde que bootea el equipo podemos ingresar al Linux desde diferentes consolas, estas se acceden apretando CTRL-F1 hasta CTRL-F6
  2. Desde dentro del sistema gráfico, debemos apretar CTRL-ALT-F1 hasta F6.
10. Conceptos teóricos importantes de Linux:
  1. Manejo del kernel separado de las aplicaciones
  2. Módulos del kernel
    1. Carga y descarga de drivers
  3. Manejo del sistema gráfico como una aplicación más
    1. El sistema gráfico como un proceso más del sistema
  4. Todo lo vamos a tratar de ver en consola y gráficamente
    1. La totalidad de las operaciones y aplicaciones se puede realizar en modo gráfico o en modo consola. Muchas de las aplicaciones gráficas son un Frontend de una aplicación de consola.



## 5. Filesystem

1. El filesystem ext3, journaling desde sus comienzos. Ext4, el futuro.

## 6. Acceso a los recursos

1. Dentro de Linux, todo se ve y accede como un archivo.
  1. Los archivos son archivos
  2. Los directorio son archivos
  3. Los dispositivos son archivos
  4. Las consolas son archivos
2. Transparencia en la utilización y programación.

## 7. Aplicaciones pequeñas

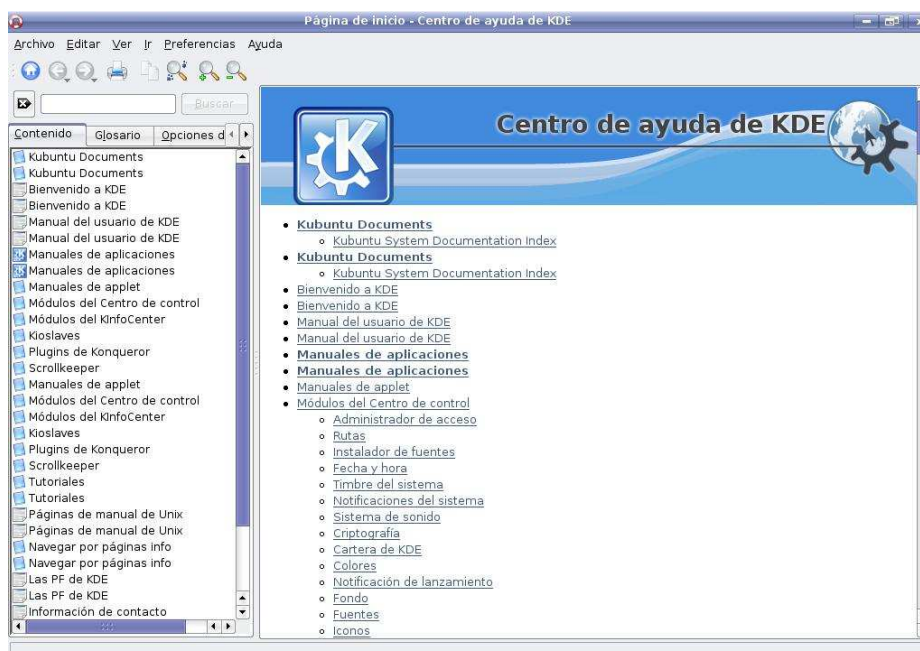
1. El concepto heredado de Unix es que las aplicaciones sean pequeñas pero que hagan las cosas realmente bien.

## 8. Sudo

1. Dentro de Kubuntu solo utilizamos un usuario sin privilegios. Para realizar tareas administrativas utilizamos el comando sudo.

## Prácticas importantes

- Boteo del sistema en varios lenguajes
- Interpretación de los mensajes del kernel a la hora de bootear
- Cambio de consolas con CTRL-ALT-F1, etc.
- Acceso a la ayuda mediante el khelppcenter
- Acceso a la ayuda por medio del comando man



*Ilustración 1: Centro de Ayuda khelppcenter*



## Jerarquía de carpetas

En un sistema Unix, todo es un archivo; si algo no es un archivo, es un proceso.

Para un usuario común no es necesario conocer esta estructura de archivos (ya que el Sistema Operativo la hace transparente a él), pero si es que desean profundizar un poco sobre Linux, pueden comenzar conociendo cómo es que está conformado.

El SO tienen la distribución de archivos con una estructura de árbol, la que tiene como inicio el directorio / (llamado raíz o root). Debajo del directorio raíz, se encuentran los principales directorios del sistema. Para verlos, pueden lanzar en consola :

```
ls / $ ls /
```

```
bin dev initrd lib media proc srv usr vmlinuz.old  
boot etc initrd.img lib64 mnt root sys var windows  
cdrom home initrd.img.old lost+found opt sbin tmp vmlinuz
```

- /bin/: comandos básicos para todos los usuarios del sistema.
- /boot/: archivos estáticos necesarios para el arranque del sistema.
- /dev/: dispositivos del sistema.
- /etc/: archivos de configuración del sistema y de las aplicaciones instaladas en el mismo.
- /home/: directorio para poner las carpetas home de los usuarios.
- /lib/: librerías esenciales para el núcleo del sistema y módulos del mismo.
- /mnt/: punto de montaje temporal para dispositivos.
- /proc/: procesos y variables del núcleo del sistema.
- /root/: directorio home para el root del sistema.
- /sbin/: comandos especiales para el root del sistema.
- /tmp/: archivos temporales. Según la distribución utilizada (o la configuración que utilicemos) se borran al arrancar el sistema o cada cierto período de tiempo.
- /usr/: segunda estructura jerárquica, utilizada para almacenar todo el software instalado en el sistema.
- /var/: directorio para los spoolers de impresión, ficheros de log, etc.

## Curso de Linux - Clase 02

### Sistema Gráfico

#### Concepto de Funcionamiento

El sistema de ventanas **X** (en inglés **X Window System**) fue desarrollado a mediados de los años 1980 en el MIT para dotar de una interfaz gráfica a los sistemas Unix. Este protocolo permite la interacción gráfica en red entre un



usuario y una o más computadoras haciendo transparente la red para éste.

Generalmente se refiere a la versión 11 de este protocolo, **X11**, el que está en uso actualmente.

**X** es el encargado de mostrar la información gráfica y es totalmente independiente del sistema operativo. El sistema de ventanas X distribuye el procesamiento de aplicaciones especificando enlaces cliente-servidor. El servidor provee servicios para acceder a la pantalla, teclado y ratón, mientras que los clientes son las aplicaciones que utilizan estos recursos para interacción con el usuario. De este modo mientras el servidor se ejecuta de manera local, las aplicaciones pueden ejecutarse remotamente desde otras máquinas, proporcionando así el concepto de transparencia de red.

La comunicación entre el cliente X y el servidor se realiza por medio de un protocolo conocido como Xprotocol, que constituye una serie bytes interpretados como comandos básicos para generar ventanas, posicionarlas, o controlar eventos. Los clientes X acceden al Xprotocol mediante el uso de una biblioteca llamada Xlib, que evita al programador de clientes X tener que lidiar con el código binario del Xprotocol. Sin embargo, los aspectos de decoración de ventana y manejos de ventanas no están definidos en esta biblioteca. **X NO ES UN** gestor de ventanas, necesita de uno para controlar el manejo de ventanas. Esto trae la ventaja de que permite al usuario instalar el administrador de ventanas que más le agrade, e incluso tener varios instalados eligiendo el más apropiado a la hora de acceder a **X**. También trae la ventaja de que hace de **X** estrictamente un sistema gráfico, de tal modo que un cliente X podría estar enviando un gráfico a una pantalla, a una impresora o a cualquier otro hardware sin darse cuenta, flexibilizando la salida gráfica.

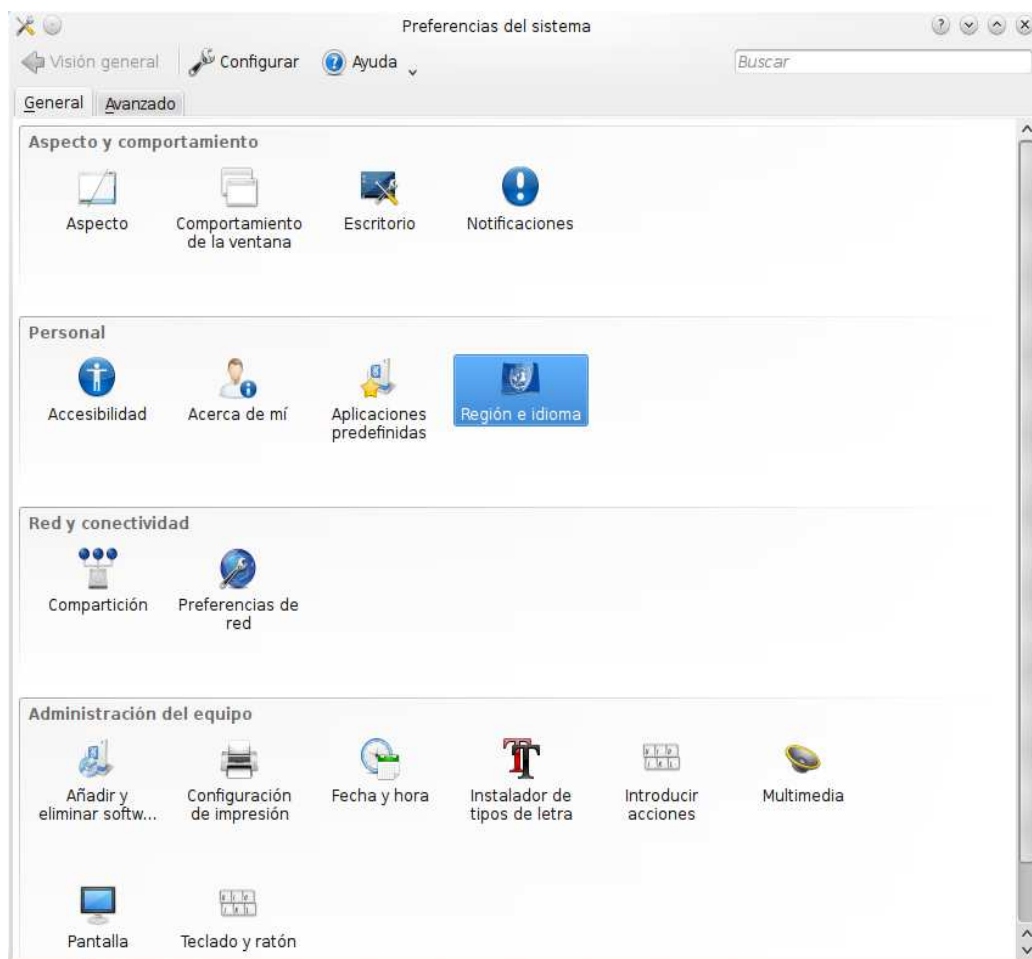
Cambiar resolución de pantalla desde teclado: Ctrl Alt + Ctrl Alt -

Cuando la resolución de pantalla es superior a 1024 x 768 y no se ve bien con los metodos anteriores se puede modificar el archivo **xorg.conf** para aumentar la resolución. Se introduce la siguiente linea en una terminal:

```
$ sudo gedit /etc/X11/xorg.conf
```

1. Conceptos de funcionamiento
  1. El sistema gráfico es un cliente servidor
2. Configuraciones generales del sistema
  1. Usamos kcontrol para todas las configuraciones gráficas
    1. Desde el menú de kde, "Configuración del sistema" está configurado para verse como en la Ilustración 2.
    2. Si ejecutamos kcontrol manualmente, se verá como en la Ilustración 3.
3. Configuración de la pantalla y del monitor
  1. Dentro de 'Monitor y Pantalla', luego 'Tamaño, Orientación y Posición'
  2. Cambiar la configuración de la pantalla
4. Configuración del lenguaje del teclado

1. Usamos control
2. Dentro de 'Región e idioma', 'Disposición del teclado'
3. Agregar o cambiar los lenguajes
5. Configuración del lenguaje de todas las aplicaciones
  1. Usamos kcontrol
  2. Dentro de 'Región e idioma', 'País/Región e idioma'
6. Copy/paste en Linux
  1. El concepto de copy paste mediante el Mouse
    1. Intentarlo en Internet
    2. Intentarlo en konsole
    3. Intentarlo en Kate
  2. El concepto tradicional, mejorado con klipper. Ver Ilustración 4
    1. Múltiples copy/pastes
    2. Operaciones
      1. Activar las operaciones desde el icono de klipper
      2. Configurar a klipper, ir a 'Operaciones'
      3. Cuando copiamos algún texto reconocible, nos permite realizar una acción sobre el mismo.



*Ilustración 2: Control desde el menú de sistema*



*Ilustración 3: kcontrol ejecutado a mano*



*Ilustración 4: Klipper y sus operaciones*



7. Konqueror, la centralización de contenidos.
  1. 'Preferencias' 'Cargar perfil de vista', diferentes formas de ver el contenido
    1. Administración de archivos. Ver la Ilustración 6
    2. Navegación en Web (por omisión). Ver la Ilustración 5
  2. loslave. Formas de acceder a diferentes datos.
    1. Una manera global de acceder a tipos de datos predefinidos.
    2. En cualquier lugar (menues de Abrir archivos, Guardar archivos, Imprimir, Ejecutar comandos, etc) podemos poner como tipo de archivos muchas variantes como:
      1. pop3://
      2. smb://
      3. ldap://
      4. audiocd://
      5. gzip://
      6. file://
      7. imap://
      8. http://
      9. tar://
  3. Panel de Navegación básico
    1. Marcadores
      1. Utilización de marcadores genéricos, ya sea de sitios Web o de directorios.
    2. Metabarra
      1. Utilización de la barra con información extendida de los archivos. Ver Ilustración 7.
    3. Network
      1. Acceso a recursos de red
    4. Services: un resumen general de las aplicaciones.
      1. Ej: Audio CD Browser, nos muestra los archivos de los CD de música tradicionales con el formato de nuestra preferencia.
    5. Iconos con la previsualización activada.
    6. Usamos desde el menú 'Ver', la opción de las carpetas que reflejan el contenido.
    7. Vista de las carpetas, cada carpeta tiene diferentes formas de visualizar su contenido.
      1. Vista del tamaño de archivo
        1. Cambiar a esta vista y ver los gráficos de tamaño
      2. Visor de imágenes
        1. Zoom hacemos con CTRL-Ruedita, o CTRL+ y CTRL-
      3. Previsualizaciones
        1. De archivos normalmente
        2. De sonidos
          1. Pararnos sobre un archivo de sonido para ver la previsualización
        3. De videos
      4. Split View



Ilustración 5: konqueror modo Web

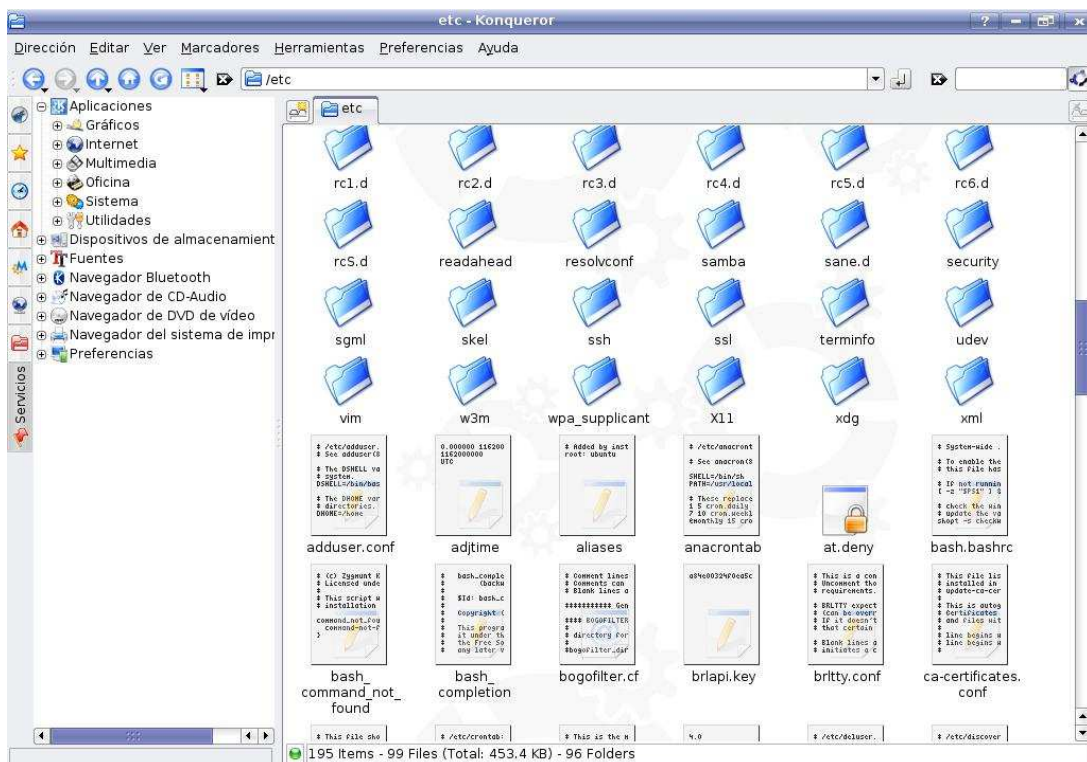


Ilustración 6: Konqueror modo archivos

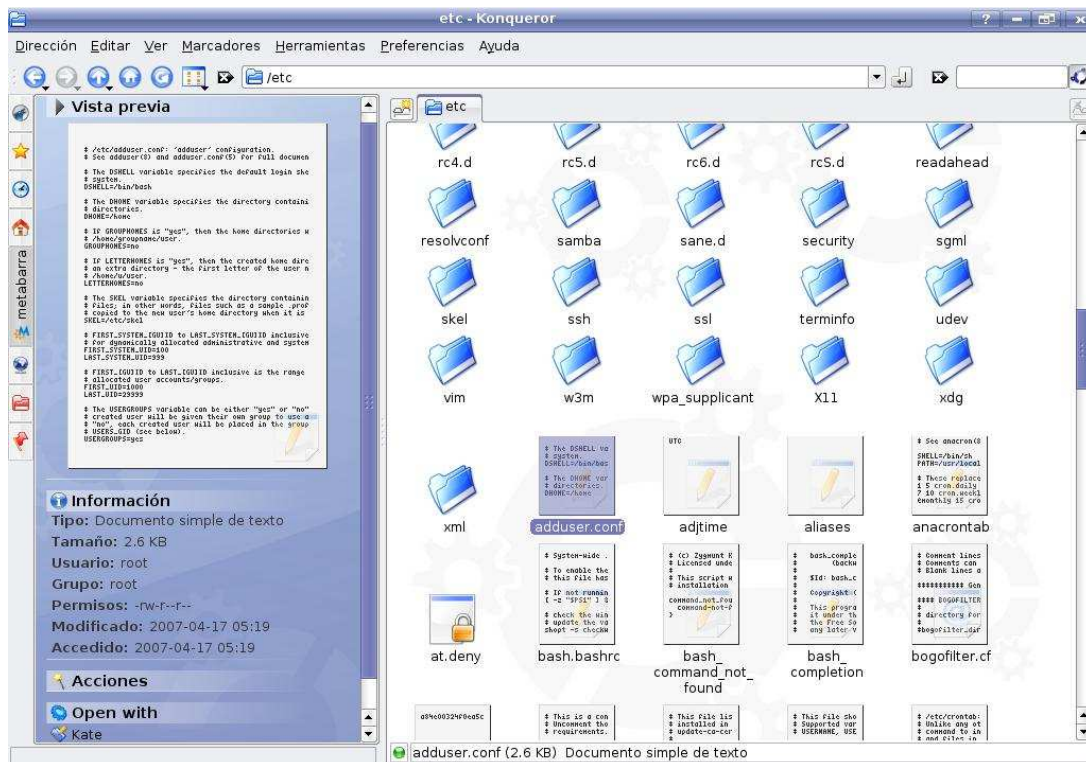


Ilustración 7: Konqueror y su metabarra

## 8. Openoffice

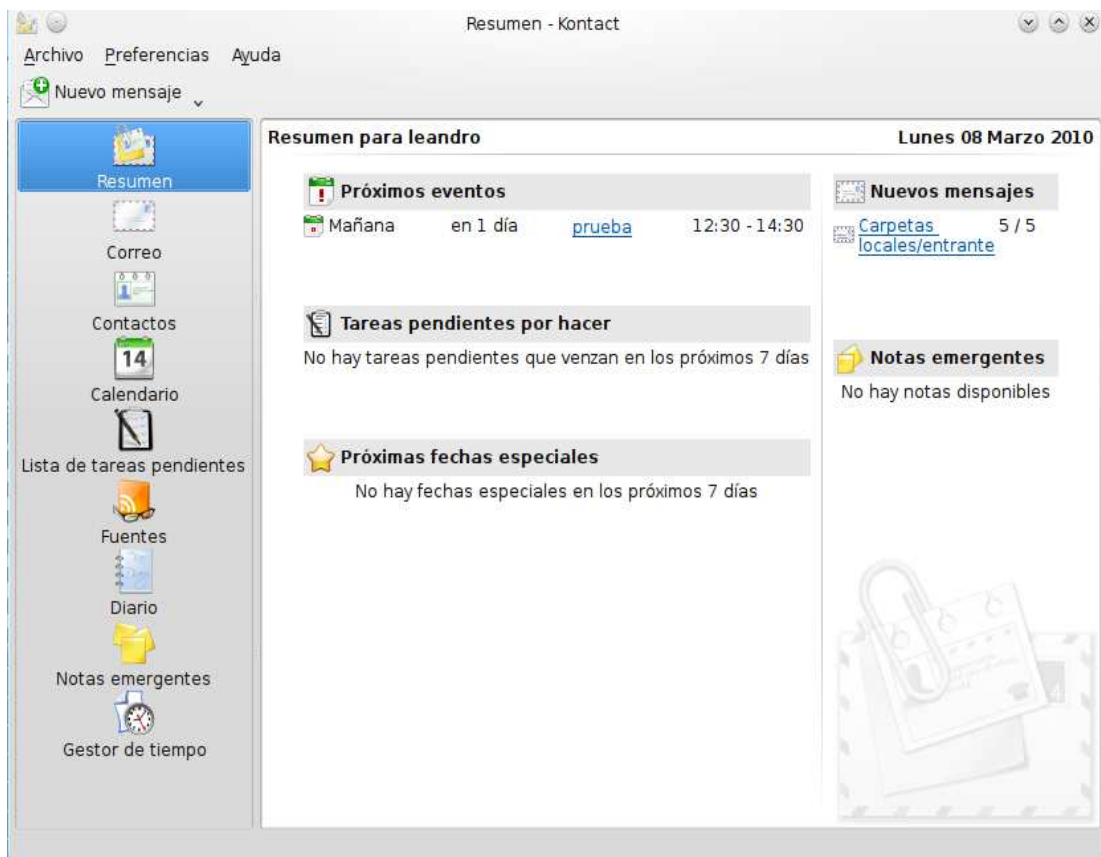
1. Suite de ofimática libre
2. Mayormente intenta guardar una similitud con los procesadores de texto actuales.
3. Exporta a pdf directamente
  1. Escribir un texto ejemplo y exportarlo como pdf.

## 9. Kontact

1. La suite de manejo de mails, contactos, agenda y organización personal. Todas las aplicaciones están separadas. Ver Ilustración 9
2. Kmail, el manejador de mails. Ver Ilustración 9
  1. Agregamos una cuenta Pop3 y Smtip de Ufasta.
  2. Configurar Kmail. Cuentas. Nueva cuenta pop3.
    1. Verificar configuración en la pagina de Ufasta.
  3. En los Extras usamos 'Comprobar qué soporta el servidor', que se conecta al servidor y muestra que opciones permite para autenticar.
  4. Verificamos que podemos enviar y recibir mails correctamente
3. Kalendar. Ver Ilustración 10
  1. Una agenda completa e integrada con el resto de Kde
  2. Eventos, repeticiones, avisos por mail desde los contactos
    1. Agregamos un evento nuevo e invitamos a otras personas por mail, verificamos que lo recibimos bien.
    3. Integración con Outlook y su agenda si reciben nuestros mails
    4. Manejo de los ToDo



4. Akregator
  1. Lector de noticias de Kde
  2. Ya está configurado para ciertos sitios
  3. Accedemos a las noticias directamente
5. Contactos
  1. Archivo de contactos integrado con el resto de Kde: kpgp, kmail, etc.
  2. Agregado y modificación de datos de los contactos
  3. De cada mail se pueden agregar los contactos
    1. De algún mail recibido agregar los contactos



*Ilustración 8: Resumen de Kontact*

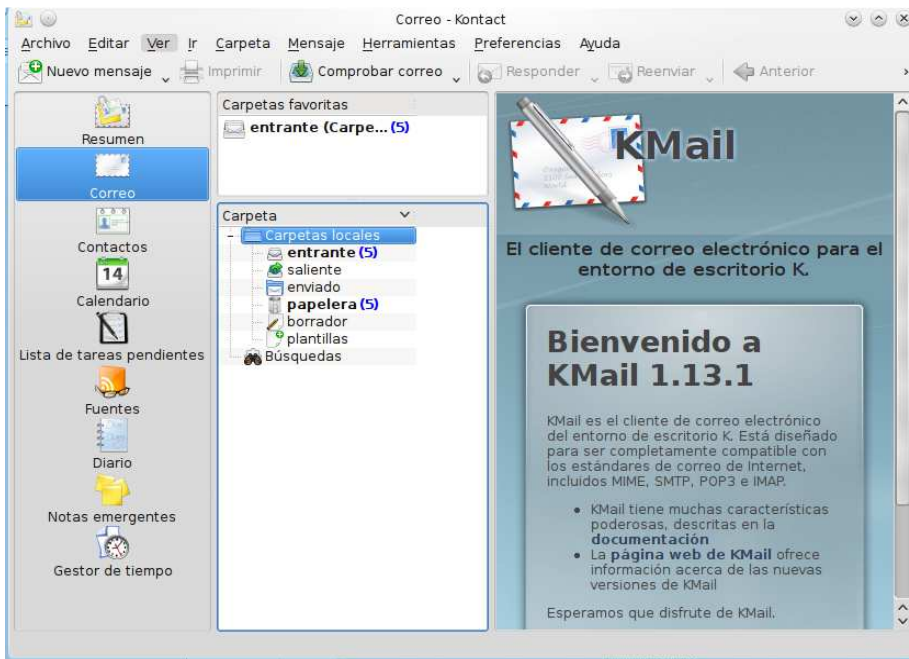


Ilustración 9 : Kmail

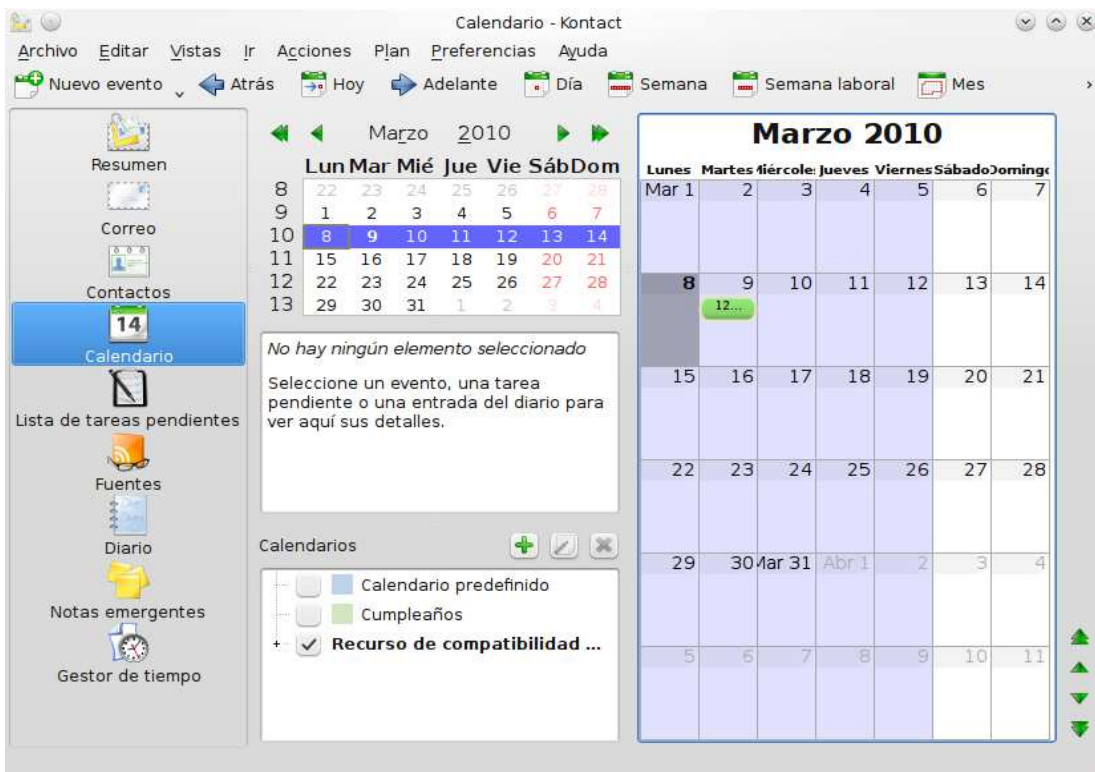


Ilustración 10: Kalendar

## 10. Kate

1. El editor gráfico de kde más potente en modo texto
2. Resaltado de diversos formatos
  1. Abrir un archivo html
  2. Abrir un archvo perl
  3. Abrir un archivo C
3. Configurar Kate
  1. Los plugins del Editor
    1. Probar de activar el completado de palabras
  2. Zoom del texto
  3. Multiples archivos simultáneos

## 11. Amarok. Ver Ilustración 11

1. El reproductor de sonido preferido de kde
2. Manejo de colecciones
3. Descarga de portadas de los discos de Internet
4. Descarga de las letras de las canciones
5. Descarga de información de wikipedia
6. Visualizaciones.
7. Conexión con magnatune directa

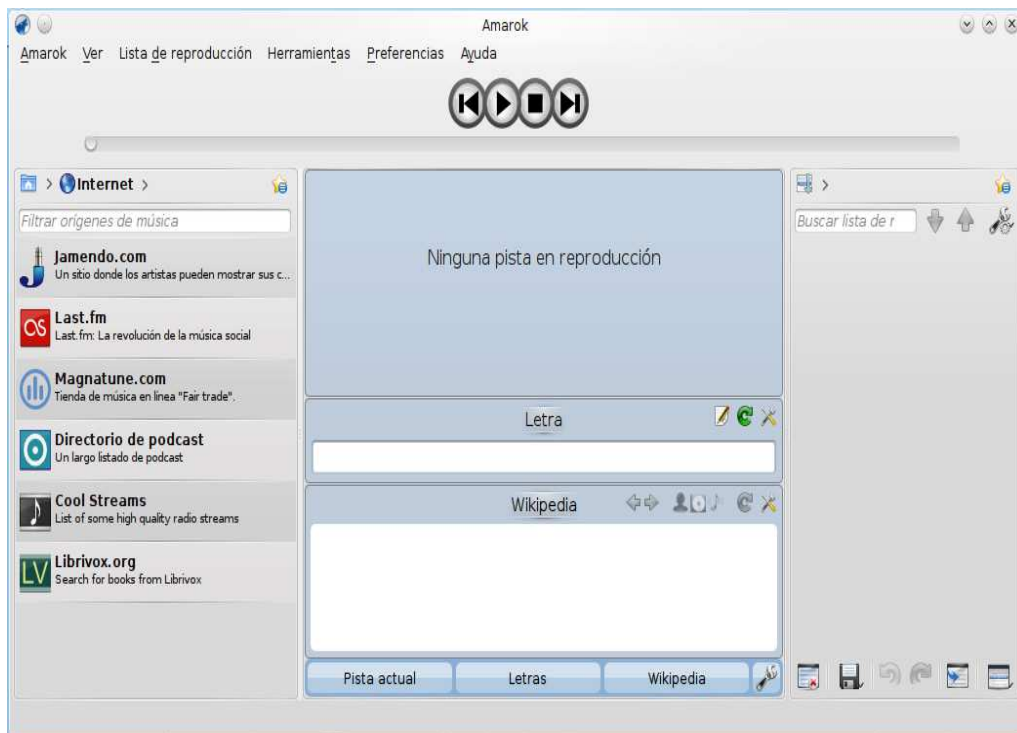


Ilustración 11: Amarok



12. Gwenview. El visualizador de gráficos
  1. Acceder al directorio /usr/share/wallpapers para ver algunos gráficos
  2. Complementos
    1. Exportación a flickr...
13. Okular. El visualizador de pdf
  1. Visualización de pdf en modo presentación
14. Ksnapshot
  1. Capturar pantallas en diferentes formatos gráficos.
15. konversation. El cliente IRC de kde
  1. Conexión al canal de ayuda kubuntu
  2. Chat con los otros usuarios

### **Prácticas importantes**

1. Cambiar la configuración de la pantalla
2. Cambiar la configuración del teclado
3. Cambiar la configuración del lenguaje de las aplicaciones
4. Navegación de archivos con konqueror
5. Crear Carpetas con Konqueror
6. Navegación Web con konqueror
7. Con konqueror acceder a la red
8. Utilización de los ioslaves para acceder a diferentes tipos de datos
9. Agregamos un evento nuevo e invitamos a otras personas por mail, verificamos que lo recibimos bien.
10. Previsualizaciones de Audio
11. Creación de un documento y exportarlo a pdf con el Openoffice
12. Creación y activación de una cuenta Pop3 y Smtip de Ufasta
13. Creación de un evento en la agenda con invitados por mails (que denieguen o acepten la invitación)
14. Lectura de las noticias con RSS de Akregator
15. Usar kate para abrir un archivo en perl o C o cualquier otro formato.
16. Utilizar amarok para escuchar música
17. Visualización de gráficos con Gwenview
18. Utilización de okular para ver un pdf y copiar texto
19. Realizar capturas de pantallas con ksnapshot (captura por Área o diferentes ventanas)
20. Conectarse con konversation al IRC y chatear con los usuario pidiendo ayuda



## **Curso de Linux - Clase 03**

### **El poder de la consola**

#### **hostname**

Nos muestra el nombre del equipo, host.. \$ hostname -i muestra la dirección ip

#### **W**

Muestra los usuarios conectados a la máquina.

#### **id**

Muestra el id del usuario y los grupos a los que pertenece.

#### **date**

Muestra la hora actual del sistema para cambiar la hora: \$date --set15:30

#### **man**

El comando man toma como argumento el nombre del programa del que se quiere ayuda, y busca en una base de datos ayuda sobre ese comando. Si no lo encuentra, te dice que no existe.

#### **ls**

list: listar. Nos muestra el contenido de la carpeta que le indiquemos después. Por ejemplo. Si queremos que nos muestre lo que contiene /etc:

```
$ ls /etc
```

Si no ponemos nada interpretará que lo que queremos ver es el contenido de la carpeta donde estamos actualmente:

```
$ ls
```

Además acepta ciertos argumentos que pueden ser interesantes. Para mostrar todos los archivos y carpetas, incluyendo los ocultos:

```
$ ls -a
```

Para mostrar los archivos y carpetas junto con los derechos que tiene, lo que ocupa, etc:

```
$ ls -l
```

Además se pueden solapar los argumentos. Si quisiéramos mostrar los archivos de la misma forma que antes, pero que muestre también los ocultos:

```
$ ls -la
```

#### **cd**

change directory: cambiar directorio. Podemos usarlo con rutas absolutas o relativas. En las absolutas le indicamos toda la ruta desde la raíz (/). Por ejemplo, estemos donde estemos, si escribimos en consola:

```
$ cd /etc/apt
```

nos llevará a esa carpeta directamente. Del mismo modo si escribimos:

```
$ cd /
```

nos mandará a la raíz del sistema de ficheros.



Las rutas relativas son relativas a algo, y ese algo es la carpeta donde estemos actualmente. Imaginad que estamos en /home y queremos ir a una carpeta que se llama temporal dentro de vuestra carpeta personal. Con escribir...

```
$ cd tu_carpeta/temporal
```

nos situará allí. Como véis hemos obviado el /home inicial ya que si no lo introducimos toma como referencia el directorio donde estamos, que es ese.

¿Y qué sucede si escribimos tan sólo:

```
$ cd
```

Sí, sólo "cd". Esto lo que hace es que te lleva a tu carpeta personal directamente y estemos donde estemos. Es algo realmente muy práctico, muy simple y que no todos conocen.

### **mkdir**

make directory: hacer directorio. Crea una carpeta con el nombre que le indiquemos. Nuevamente podemos usar rutas absolutas y relativas. Podemos indicarle toda la ruta que le precede al directorio que queremos crear, o si estamos ya en la carpeta que lo va a contener basta con poner tan sólo el nombre:

```
$ mkdir /home/tu_cuenta/pepito
```

Si ya estamos en /home/tu\_cuenta...

```
$ mkdir pepito
```

### **rm**

remove: borrar. Borra el archivo o la carpeta que le indiquemos. Como antes se puede indicar la ruta completa o el nombre del archivo. Esto a partir de ahora lo vamos a obviar, creo que ya ha quedado claro con los dos comandos anteriores.

Para borrar un archivo:

```
$ rm nombre_archivo
```

Para borrar una carpeta vacía:

```
rm nombre_carpeta
```

Para borrar una carpeta que contiene archivos y/o otras carpetas que pueden incluso contener más:

```
rm -r nombre_carpeta
```

Otras opciones: "-f" no te pide una confirmación para eliminar o "-v" va mostrando lo que va borrando.

### **cp**

copy: copiar. Copia el archivo indicado donde le digamos. Aquí podemos también jugar con las rutas, tanto para el fichero origen, como en el del destino. También podemos poner el nombre que le queramos poner a la copia. Por ejemplo, si estuviéramos en /etc/X11 y quisiéramos hacer una copia de seguridad de xorg.conf en nuestra carpeta personal:

```
$ cp xorg.conf /home/tu_carpeta/xorg.conf.backup
```

Para copiar un directorio se usa cp -r

```
$ cp -r /etc /backup/
```

Tendremos /backup/etc, y dentro de ese directorio tendremos lo mismo que hay en /etc



## **mv**

move: mover. Es igual que el anterior, sólo que en lugar de hacer una copia, mueve directamente el archivo con el nombre que le indiquemos, puede ser otro distinto al original:

```
$ mv /etc/pepito.html /home/tu_carpeta/ese_pepito.html
```

Otro uso muy práctico que se le puede dar es para renombrar un archivo. Basta con indicar el nuevo nombre en el segundo argumento con la misma ruta del primero. En este ejemplo suponemos que ya estamos en la carpeta que lo contiene:

```
$ mv pepito.html ese_pepito.html
```

También permite mover directorios o renombrarlos

```
$ mv /home/yo/misdocumentos /home/yo/mis_documentos
```

## **find**

find: encontrar. Busca el archivo o carpeta que le indiques:

```
$ find / -name pepito
```

El comando anterior buscaría en todos los sitios las carpetas y archivos que se llamen pepito. Si tuviéramos la seguridad de que se encuentra en /var por ejemplo, se lo indicaríamos:

```
$ find /var -name pepito
```

Si no estamos muy seguros del nombre podemos indicárselo con comodines. Supongamos que el nombre de lo que buscamos contiene "pepi", en la misma carpeta de antes:

```
$ find /var -name *pepi*
```

Tiene otras opciones. Por ejemplo podemos decirle que encuentre los archivos/carpetas de más de 1500 KB:

```
$ find / -size +1500
```

O los archivos/carpetas contienen el nombre "pepi" y tienen menos de 1000 KB:

```
$ find / -name *pepi* -size -1000
```

## **clear**

clear: despejar. Limpia la pantalla/consola quedándola como si acabáramos de abrirla.

```
$ clear
```

## **ps**

process status: estado de los procesos. Nos muestra lo que queramos saber de los procesos que están corriendo en nuestro sistema. Cada proceso está identificado con un número llamado PID. Si hacemos:

```
$ ps -A
```

nos mostrará un listado de todos los procesos, su PID a la izquierda y su nombre a la derecha. Si queremos más información:

```
$ ps aux
```

## **kill**

kill: matar. Elimina el proceso que le indiquemos con su PID:

```
$ kill
```



En ocasiones el proceso no “muere” del todo, pero se le puede forzar al sistema para que lo mate con seguridad del siguiente modo:

```
$ kill -9
```

### **killall**

Permite matar por nombre de proceso, y los procesos llamarse siempre igual (no así sus pid), permite no tener que hacer un ps para saber qué tienes que poner.

```
$ killall firefox-bin
```

### **sudo**

super-user do: hacer como superusuario. La cuenta de usuario en Ubuntu es relativamente normal. Tiene derechos de administrador a medias. Cada vez que se haga algo importante y de riesgo para el sistema, hay que hacerlo mediante el prefijo “sudo” y escribiendo después la contraseña.

Siempre que necesitemos hacer un apt-get/aptitude update o install y acciones de este tipo, tendremos que poner antes el “sudo”.

### **passwd**

password: contraseña. Con este comando podremos cambiar la contraseña de nuestra cuenta. Primero nos pedirá la contraseña actual como medida de seguridad. Después nos pedirá que introduzcamos dos veces seguidas la nueva contraseña.

```
$ passwd
```

### **su**

super-user: superusuario. Mediante su podremos loguearnos como superusuario. Tras escribirlo nos pedirá la contraseña de root y estaremos como administrador. Podremos hacer todo lo que queramos.

```
$ su
```

Este comando también nos permite hacer login con otra cuenta distinta. Por ejemplo, imaginemos que tenemos otra cuenta, además de root y la nuestra, llamada “invitado”. Para hacer login como tal bastaría con poner:

```
$ su invitado
```

y después escribir la contraseña de esa cuenta.

### **sudo passwd**

No es un comando propiamente dicho, pero es interesante conocerlo. Gracias a la unión de estos dos comandos se puede cambiar la contraseña de root (la del super-usuario).

```
$ sudo passwd
```

### **apt**

advanced packets tool: herramienta avanzada de paquetes. Es uno de los comandos más útiles que se han desarrollado en los sistemas GNU/Linux debian o basados en esta distro. Nos permite comprobar actualizaciones, actualizar todo el sistema. También nos ofrece funcionalidad para buscar, descargar e instalar paquetes con una sola orden.





Tenemos variantes, las más usadas son las siguientes:

\$ apt-cache search nombre\_paquete

Busca nombre\_paquete para ver si existe literal o aproximadamente ofreciéndonos los paquetes que pudieran ser en caso de que hayamos puesto un nombre aproximado.

\$ apt-get update

Actualiza los repositorios que son los que contienen los paquetes. Los repositorios son como las direcciones que contienen nuestros paquetes. apt-get update lo que hace es actualizar el listado de todos esos paquetes, con la dirección de dónde obtenerlos para que a la hora de hacer la búsqueda y su posterior descarga sea más rápida haciéndolo en nuestro ordenador.

\$ apt-get upgrade

Actualiza nuestro sistema con todas las posibles actualizaciones que pudiera haber. La actualización no se realiza sólo sobre el propio sistema operativo, sino también sobre las aplicaciones que estén contenidas en los repositorios. Una útil forma de estar siempre al día.

\$ apt-get install nombre\_paquete

Localizado el nombre del paquete que queremos descargar e instalar, este comando se encargará del proceso. Buscará en nuestro índice (el que se crea con update) de dónde tiene que descargarse el paquete, lo hace y posteriormente lo instala.

\$ apt-get remove [--purge] nombre\_paquete

Elimina el paquete especificado del sistema. Dame el argumento “--purge” (corchetes = opcional) para que borre también los ficheros de configuración.

\$ apt-get autoremove

Elimina paquetes que han quedado inservibles tras realizar algún apt-get remove, los llamados huérfanos. Normalmente tras hacer este último te avisa en un mensaje que lo realices.

Todos estos comandos necesitan tener privilegios de administración, así que si no los usáis como root, debéis agregar primero el conocido “sudo”.

## **aptitude**

aptitude: aptitud, habilidad. En el fondo juega con las siglas de apt para crear aptitude. Es una versión mejorada de apt.

aptitude nació como un front-end de apt, es decir, como una especie de aplicación gráfica y en modo texto para realizar todo lo que hace apt. Pero lo cierto es que sus características son mejores.

apt cuando instala algo te puede realizar una sugerencia para que instales algo más que te podría venir bien, pero no lo hace. Hay programas que a la vez usan otros para algunas de sus funciones u opciones. apt no instalaría los segundos, como mucho te avisaría. Sin embargo aptitude sí que lo instalará porque sabe que de alguna forma es indispensable para el que has pedido.

De la misma forma, si con apt instalas luego ese programa que es usado por otro, cuando desinstalas el principal, no se desinstalará el secundario, aunque éste ya no tenga mucho sentido que esté instalado, y lo mismo sucede con librerías. aptitude está capacitado para desinstalar lo que él mismo ha instalado como recomendación. Te deja el sistema más limpio tras las desinstalaciones.

Para abrir el interfaz gráfico de aptitude, tan sólo hay que teclearlo:



\$ aptitude

Sin embargo, también se puede usar exactamente igual que apt, pero con las características que he comentado de aptitude:

\$ aptitude search nombre\_paquete

\$ aptitude install nombre\_paquete

\$ aptitude remove nombre\_paquete

\$ aptitude purge nombre\_paquete

\$ aptitude update

\$ aptitude upgrade

Y al igual que antes, necesitaréis usarlo con el sudo delante si no estáis como administrador.

### dpkg

depackage: despaquetar. Los paquetes cuando se instalan sufren un proceso de despaquetaje. En el fondo un paquete .deb contiene una serie de scripts de pre-instalación, post-instalación y los archivos en cuestión del paquete.

Este comando lo usaremos para instalar un paquete .deb que ya tengamos descargados en nuestro sistema. En muchas ocasiones hay una aplicación que no está en los repositorios y nos hemos bajado el .deb para instalarlo con el interfaz gráfico que corresponda (GDebi en el caso de GNOME).

En el fondo estas interfaces gráficas están basadas en dpkg. Si queremos instalar un paquete ya descargado mediante consola usaremos el argumento '-i' (i=install):

\$ dpkg -i nombre\_paquete

Para desinstalarlo '-r' (r=remove):

\$ dpkg -r nombre\_paquete

Para desinstalar el paquete y los ficheros de configuración "--purge" (purgar):

\$ dpkg -r --purge nombre\_paquete

### alien

Alien: de otro país, de otro planeta. Aunque Debian -y por extensión Ubuntu- dispone de una ingente cantidad de paquetes en sus repositorios, puede que alguien tenga algún problema en encontrar una aplicación específica empaquetada como le interesa aunque ha visto el paquete que quiere para otras distros.

alien es bastante práctico para estas situaciones ya que nos permite transformar un paquete de un gestor de paquetes determinado en otro. Por ejemplo podemos pasar de un .deb (Debian) a un .rpm (Red Hat) y viceversa.

Las extensiones soportadas son:

\* deb (Debian)

\* rpm (Red Hat)

\* slm (Stampede)

\* tgz (Slackware)

\* pkg (Solaris)

Su uso es sencillo. Lo que debemos saber es el argumento que transformará el paquete original en la extensión objetivo:

\* "--to-deb" o "-d" para transformar a .deb

\* "--to-rpm" o "-r" para transformar a .rpm



- \* “-to-tgz” o “-t” para transformar a .tgz
- \* “-to-pkg” o “-p” para transformar a .pkg
- \* “-to-slp” para transformar a .slp

Como ejemplo, pasaremos un supuesto paquete de Red Hat llamado “pepito.rpm” a “pepito.deb”:

```
$ alien -d pepino.rpm
```

### **man**

manual: manual. Es otro de los comandos de gran potencia en linux. Normalmente queda programa o comando viene con un archivo de ayuda muy completo sobre su uso y sus argumentos. Cuando se desconoce cómo se usa y qué argumentos tiene un comando o aplicación tan sólo tenemos que escribir en consola:

```
$ man nombre
```

En ocasiones la información que nos ofrece man puede llegar a ser excesiva. Casi todos los comandos y aplicaciones aceptan el argumento “-help” para que muestre cierta ayuda más resumida. Por ejemplo con aptitude:

```
$ aptitude -help
```

Todos los comandos que os he mostrado tienen muchos más argumentos. Os he puesto los más usados o necesarios, así que si hay alguno que os interesa particularmente conocer más de él, tenéis “man” o “-help” para obtener más detalles.

### **alias**

Permite crear atajos para comandos:

```
$ alias ll="ls -l"
```

```
$ alias instala="sudo aptitude install"
```

también se pueden dejar permanentes, para agregar los alias permanentemente editan el archivo:

```
$ gedit /home/usuario/.bashrc
```

y agregan al final del archivo sus alias

```
alias actualiza="sudo aptitude update"
```

```
alias upgrade="sudo aptitude upgrade"
```

```
alias instala="sudo aptitude install"
```

```
alias purge="sudo aptitude purge"
```

```
alias show="sudo aptitude show"
```

```
alias busca="sudo aptitude search"
```

```
alias apagate="sudo shutdown -h now"
```

luego cierran la terminal, abren otra y listo.

Para instalar la aplicación "pepe"

para instalar:

```
$ sudo aptitude install pepe
```

o si tenemos el alias:

```
$ instala pepe
```



## Editores

### Vi

vi es un editor de texto elemental que conviene conocer porque a veces es el único a nuestro alcance.

Para aclararse con vi, tienes que tener en cuenta que vi puede encontrarse en dos estados o modos:

- En el modo de comandos, vi está esperando que le des alguna orden (por tanto, interpreta lo que escribas como órdenes).
- En el modo de edición, vi está esperando que escribas el texto del fichero (por tanto, interpreta lo que escribas como texto).

Cuando entras en vi, está en modo de comandos. Para pasar al modo de edición puedes pulsar i (insertar), a (añadir), o (añadir una línea). Para pasar al modo de comandos, puedes pulsar Escape o Suprimir.

### Comandos más habituales de vi

comando	descripción
i	insertar antes del cursor
a	añadir detrás del cursor
o	añadir una línea en blanco
x	borrar un carácter
j	borrar el final de línea (une dos líneas)
dd	borra la línea completa
u	deshacer la última edición
:q	salir
:q!	salir sin guardar
:w	guardar
:wq	guardar y salir
:set nu	muestra números de línea
:set nonu	oculta números de línea
:n copy m	copia la línea n detrás de la línea m
:n1, n2 copy m	copia desde la línea n1 hasta la línea n2 detrás de la línea m

## Prácticas importantes

1. Abrir múltiples consolas en konsole
2. Utilizar todos los comandos con sus variantes.
3. Editar un archivo en vi, agregando texto, buscando, reemplazando, borrando y saliendo.
4. Editar un archivo en kate, agregando texto, buscando, borrando y saliendo.

## Curso de Linux - Clase 04

### Administración del sistema

1. Manejo de dispositivos Usb gráficamente
  1. Hal y la detección automática de dispositivos. Ver Ilustración 14
  2. Agregar un dispositivo y desconectarlo en la consola
    1. Comando mount



Ilustración 12: Reconocimiento de USbs

2. Administración de Firewalls corporativos y personales
  1. Conceptos fundamentales, ¿por qué un firewall?
  2. Administración
    1. iptables, la consola de administración
    2. Configuración del firewall para un equipo en una casa con Internet.



### 3. GFW. Administración gráfica de Firewalls.

#### 1. Instalación

1. `sudo apt-get install gufw`

#### 2. Ejecución como root

#### 3. Por consola:

#### 4. Para activarlo y que se inicie en el arranque ejecutamos:

```
sudo ufw enable
```

Para denegar todas las conexiones

```
sudo ufw default deny
```

Para abrir puertos (sustituyendo el número de puerto por el de tu elección):

```
sudo ufw allow 14002/tcp
```

```
sudo ufw allow 15002/udp
```

Permitir la comunicación entre la red local (sustituye la ip por la que se ajuste a tu configuración):

```
sudo ufw allow from 192.168.1.0/16
```

### 3. Sistema de logs

#### 1. Administración del sistema de logs

#### 2. Ksystemlog, Visualización de logs. Ver Ilustración 14

##### 1. Selección de los archivos de log

##### 2. Interpretación y verificación

#### 3. Syslog server

#### 4. Log remoto o centralización de logs

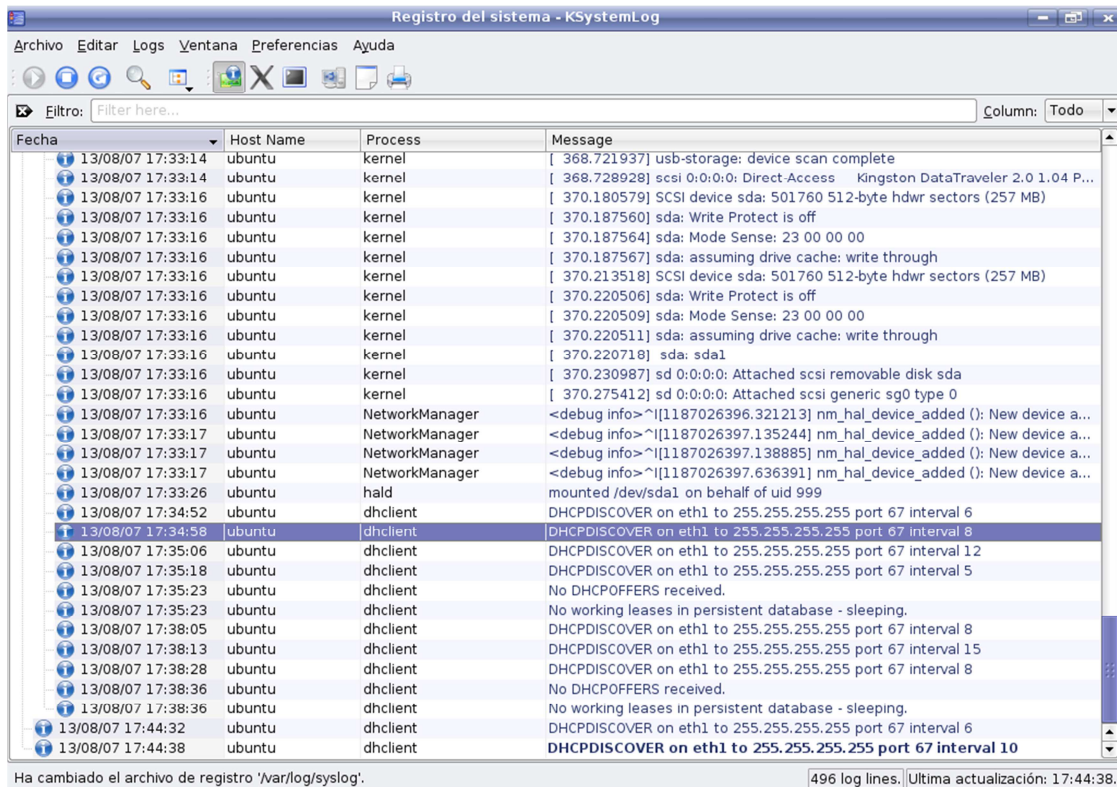


Ilustración 13: Ksystemlog

4. Impresoras
  1. Agregado de impresoras
  2. Configuración de impresión
5. Reloj impreciso
  1. Configuración del reloj en diversos modos
6. Adept (aptitude, apt-get, apt-cache)
  1. Administración de los paquetes de ubuntu-debian
    1. Característica esencial del mantenimiento
  2. Actualizar la base de paquetes conocidos, lo más importante antes de instalar
    1. `sudo apt-get update`
  3. Agragado de programas
    1. `sudo apt-get install xxxx`
  4. Borrado de programas
  5. Configuración de fuentes de repositorios
    1. Ingresar diferentes repositorios
7. Kcontrol
  1. Administración de Usuarios
  2. Agregado de usuarios



### Prácticas importantes

1. Agregado y desconexión de dispositivos.
2. Agregado de reglas al firewall para una conexión hogareña
3. Visualización de logs, interpretación
4. Configuración del syslog
5. Agregado de programas desde Internet a kubuntu
6. Borrado de paquetes
7. Agregado de usuarios gráficamente
8. Agregado de usuarios manualmente

### Archivos de configuración del Sistema

NIVEL	Funcionalidad
0	El nivel de ejecución 0 está configurado para parar el sistema.
1	Este nivel es denominado como <i>single user</i> , ya que sólo permite la entrada al sistema al root del mismo. Se arrancan los daemons mínimos y sirve para tareas de mantenimiento.
2~5	Los niveles del 2 al 5 están destinados para ser configurados según las necesidades de cada instalación. Al instalar el sistema, por defecto todos son iguales. Estos niveles también se llaman multiusuario, ya que, por defecto, permiten que más de un usuario trabaje en el sistema.
6	El último nivel está preparado para reiniciar el sistema. Es muy parecido al 0 pero se añade una función de reinicio.

1. Orden de booteo del sistema
  1. Niveles de ejecución
    1. *Generalmente se definen en inittab pero Ubuntu cambio a Upstart, directamente obtiene las configuraciones de /etc/init/*
    2. *Ubuntu Cambio de Forma de Niveles de Ejecución:*  
 Runlevel 0 – Halt  
 Runlevel 1 – Modo texto multiusuario  
 Runlevel 2 – Modo gráfico multiusuario  
 Runlevel 6 – Reboot
  - *Probar desde la línea de comandos el comando runlevel*
  - *Diferentes niveles de ejecución en los directorios /etc/rcX.d*
  - *Listar los Demonios que se cargan en el runlevel initctl list*
2. Archivos de passwords y datos de los usuarios
  1. /etc/passwd





Una línea ejemplo en este fichero:

```
usuario1:FXWUuZ.vwXttg:500:501:usuario pepito:/home/usuario1:/bin/bash
```

Los diferentes campos (7) están separados por dos puntos (:) y el significado de los mismos es el siguiente:

<b>usuario1:</b>	Nombre de la cuenta (Login)
<b>FXWUuZ.vwXttg:</b>	Clave de acceso encriptada (password)
<b>500:</b>	UID de esta cuenta
<b>501:</b>	GID del grupo principal al que pertenece la cuenta
<b>usuario pepito:</b>	Nombre del usuario
<b>/home/usuario1:</b>	Directorio de trabajo de usuario1
<b>/bin/bash:</b>	Interprete de comando (shell) de usuario pepito

## 2. /etc/shadow

<b>slice:\$1\$NLJJ6\$ow5g1l1NgYITqqQQy5D21:14234:0:99999:7: : :</b>	
<b>Contraseña</b>	Contraseña encriptada. La forman entre 13 y 24 caracteres (a-z, A-Z, 0-9, \, /). Si comienza por el carácter \$, indica que la contraseña se ha encriptado usando un algoritmo distinto de DES. Si comienza por \$1\$, el algoritmo de cifrado está basado en MD5.
<b>Nombre de usuario</b>	Nombre que identifica al usuario en el sistema. Debe tener entre 1 y 32 caracteres.
<b>Último cambio</b>	Días que han pasado desde la última vez que la contraseña fue cambiada contados desde el 1 de enero de 1970.
<b>Mínimo</b>	Días que deben pasar como mínimo para que el usuario pueda cambiar la contraseña.
<b>Máximo</b>	Días durante los que la contraseña es válida. Al terminar el usuario tiene que cambiar la contraseña.
<b>Aviso</b>	Días a los que el usuario será avisado de que debe cambiar la contraseña antes de que ésta caduque.
<b>Inactivo</b>	Días a los que se deshabilita la cuenta después de que caduque la contraseña.
<b>Caducidad</b>	Días a los que se deshabilita la cuenta contados desde el 1 de enero de 1970.

## 3. Archivos de grupos

1. sudo addgroup grupo
2. sudo adduser usuario grupo
4. Configuraciones personales de cada usuario
  1. Directorios home

## Prácticas importantes

1. Modificación de la configuración de los programas a la hora de booteo.
2. Agregado y borrado de grupos de usuarios
3. Agregado y borrado de usuarios manualmente
4. Agregado de un usuario a un grupo creado



## La Red

### Dirección IP

Una **dirección IP** es un número que identifica de manera lógica y jerárquica a una interfaz de un dispositivo (habitualmente una computadora) dentro de una red que utilice el protocolo IP (*Internet Protocol*), que corresponde al nivel de red o nivel 3 del modelo de referencia OSI. Dicho número no se ha de confundir con la dirección MAC que es un número hexadecimal fijo que es asignado a la tarjeta o dispositivo de red por el fabricante, mientras que la dirección IP se puede cambiar.

Es habitual que un usuario que se conecta desde su hogar a Internet utilice una dirección IP. Esta dirección puede cambiar cada vez que se conecta; y a esta forma de asignación de dirección IP se denomina una *dirección IP dinámica* (normalmente se abrevia como *IP dinámica*).

Los sitios de Internet que por su naturaleza necesitan estar permanentemente conectados, generalmente tienen una *dirección IP fija* (se aplica la misma reducción por *IP fija* o *IP estática*), es decir, no cambia con el tiempo. Los servidores de correo, DNS, FTP públicos, y servidores de páginas web necesariamente deben contar con una dirección IP fija o estática, ya que de esta forma se permite su localización en la red.

### Cambiar DNS

Tenemos que modificar el archivo:

/etc/resolv.conf

Buscamos las líneas nameserver 80.58.0.33 nameserver 195.235.113.3  
Sustituimos las dos direcciones por las que deseemos.

Administrar la red

1. Conceptos fundamentales
  1. Direcciones IP
  2. Protocolo TCP e IP
  3. Ruteo
  4. Dns
2. Configuración de la red
  1. Cambio de direcciones Ip
    1. Comando 'ifconfig'
    2. Cambio de la dirección ip del equipo
  2. Cambio de Rutas
    1. Comando 'route'
    2. Cambio de las rutas en el equipo
    3. Cambio del gateway por omisión
  3. DHCP
    1. Tomar una dirección IP con DHCP
    2. Re-pedir una dirección con DHCP
  4. Cambio de DNSs
    1. Cambio de los dns del sistema

3. Verificación de la conexión de la red
  1. ping y sniffing
4. Knetworkmanager. Ver Ilustración 16

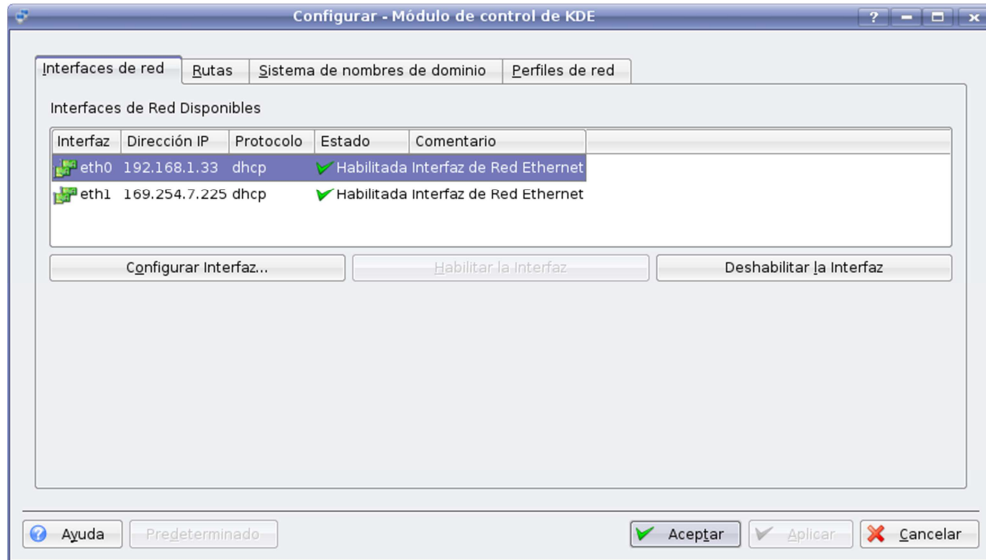


Ilustración 14: Knetworkmanager

1. Conexión a otros equipos con SSH
  1. ¿Qué es SSH?
    1. **SSH** (acrónimo del inglés **Secure SHell**, cuya traducción sería *intérprete de comandos seguro*) es un protocolo de comunicación para controlar un ordenador en remoto a través de una CLI (Command Line Interface -Interfaz de Línea de Comandos- también llamada: "shell"). Sirve para conectarnos con un ordenador ante el cual no estamos físicamente, bien porque está en una sala de servidores refrigerada, bien porque no tiene teclado ni pantalla, por ejemplo los que están apilados en un rack. Instalar servidor "open ssh server". Para instalarlo ejecuta el siguiente comando: `apt-get install openssh-server` levantar el servidor `sudo /etc/init.d/ssh start`
  2. ¿Para qué sirve?
  3. Configuración de ssh
  4. Levantar el servidor ssh
  5. Acceso remoto con el cliente



## Curso de Linux - Clase 05

### Conexión a un servidor remoto

Para conectarnos con un servidor SSH remoto desde Ubuntu tenemos un cliente por defecto. En Windows no, hay que descargar un cliente (por ejemplo, el programa Putty). Usar el cliente es muy sencillo: basta con teclear desde una consola o terminal lo siguiente:

```
$ ssh host_remoto
```

Donde *host\_remoto* es la IP del servidor SSH o el nombre de este. Eso hará que nos conectemos con nuestro nombre de usuario. Si queremos conectar como un usuario remoto teclearemos:

```
$ ssh usuario_remoto@host_remoto
```

Luego nos pide la contraseña del usuario. La primera vez que nos conectemos a un servidor tarda un poco más y nos pide confirmación tecleando "yes" con todas sus letras, las subsiguientes ya no. Sabemos que estamos conectados porque el prompt cambia y aparece en lugar del nombre de nuestro host el nombre del host remoto.

Los comandos, programas y scripts que lancemos tras conectarnos se ejecutarán en la máquina a las que nos hayamos conectado, utilizando los recursos del host remoto (CPU, memoria, disco, etc.). Esta arquitectura puede utilizarse, por ejemplo, para tener un servidor más potente y varios clientes que ejecutan aplicaciones en dicha máquina.

Para ejecutar aplicaciones gráficas en la máquina a la que nos conectamos tenemos dos opciones. La primera consiste en definir la variable \$DISPLAY apuntando a la máquina desde la que nos conectamos.

```
$ export DISPLAY=host_local:0.0
```

Este mecanismo no se recomienda por motivos de seguridad (el protocolo X11 no se encuentra cifrado) y, además, pueden encontrarse problemas porque cortafuegos intermedios bloqueen ese tráfico (puertos 600x TCP).

Una solución mejor es utilizar un túnel SSH para encapsular el protocolo X11, lo que transmite la información de manera segura y, además, no suele dar problemas con los cortafuegos intermedios.

Para poder ejecutar aplicaciones gráficas en el host remoto de forma segura, necesitamos dos cosas. La primera, que en la configuración del servidor SSH del host remoto (/etc/ssh/sshd\_config) se encuentre activada la siguiente opción:

```
X11Forwarding yes
```

Para aprovechar esta característica, hemos de conectarnos usando el parámetro -X, lo que exportará la configuración de la variable \$DISPLAY con lo que podremos ejecutar aplicaciones gráficas de forma remota:

```
$ ssh -X usuario_remoto@host_remoto
```



Ahora si ejecutas el programa *xclock* verás que la ventana sale en tu escritorio:

```
$ xclock
```

### **Prácticas importantes**

1. Cambio de las rutas del sistema (por consola y *knetworkmanager*)
2. Cambio de los DNS del sistema (por consola y *knetworkmanager*)
3. Utilización de *ping* para comprobar conexiones
4. Utilización del *sniffing* para comprobar conexiones
5. Administración gráfica de los datos de red
6. Conexión a otro equipo por *ssh*

### ***Enviar u obtener archivos y carpetas con scp***

En una consola o terminal tecleamos:

```
$ scp -r usuario@maquina:/home/carpeta .
```

luego ponemos la password del usuario y nos copiará la carpeta *"/home/carpeta"* remota en el directorio actual *"."* naturalmente siempre que usuario tenga permisos sobre la carpeta y su cuenta esté entre las de los que pueden hacer *ssh*. La opción *"-r"* significa recursivo, es decir, copia la carpeta y todo su contenido, incluidas las subcarpetas y el contenido de éstas. Si no lo ponemos la orden para copiar todos los archivos de una carpeta sería:

```
$ scp usuario@maquina:/home/carpeta/* .
```

Si lo que queremos es enviar una carpeta con su contenido, utilizaremos la orden:

```
$ scp /home/carpeta/* usuario@maquina:/carpeta/
```

### **Práctica**

#### **Dirección ip del equipo donde está trabajando**

```
ifconfig
```

#### **Lista de los procesos del sistema completo**

```
ps -A
```

```
ó
```

```
ps aux
```

#### **Listado del contenido del directorio /**

```
ls /
```

#### **Cree un directorio**

```
mkdir pepe
```

#### **Guardar el contenido del listado de / en el archivo pepito.txt**

```
ls / > pepito.txt
```



**Mostrar que usuarios están conectados a su equipo actualmente.**

w

**Mostrar la fecha y hora del sistema.**

date

**Copiar archivos o carpetas**

cp archivoOrigen archivoDestino  
cp -r dirOrigen dirDestino

**Cambiarle el nombre al archivo**

mv nombreViejo nombreNuevo

**Cuente como agregaría los repositorios universe y multiverse**

Sistema --> Administracion --> Origenes del Software  
ó  
Sistema --> Administracion --> Gestor de Paquetes

**Nombre dos formas de instalar el software Abiword utilizando el modo grafico utilizando "Agregar o Quitar Programas" o "Gestor de Paquetes"**

Escriba el comando para instalarlo usando la consola.

sudo apt-get install nombrePrograma

screenshot de los gráficos de Recurso dentro del Monitor de Sistema

Swap

Es el espacio dentro del disco rígido que se utiliza como intercambio entre el disco rígido y la memoria ram, cuando usamos el live cd de ubuntu, no utilizamos el espacio de intercambio

**Conectarse con el cliente ssh a la dirección IP x.x.x.x**

ssh host\_remoto  
ssh usuario\_remoto@host\_remoto  
ssh -X usuario\_remoto@host\_remoto