

Física Computacional I

**Licenciatura en Física
menciones en Astronomía,
Ciencias Atmosféricas y
Computación Científica
2018**

Profesor: Graeme Candlish
Email: ***graeme.candlish@ifa.uv.cl***

- **Descripción y objetivo:**

Asignatura teórica y práctica que entrega herramientas computacionales básicas para modelar sistemas físicos. Se introduce el ambiente **Linux** y sus principales características además de introducir al estudiante el lenguaje **Python** y sus aplicaciones más simples a la física mecánica.

- Unidad I: **Introducción a Linux**
- Unidad II: **Introducción a Python**





- **Hardware**



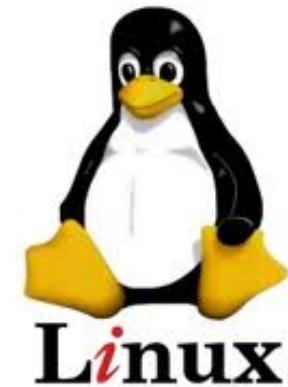
- **Software**



Conceptos básicos:

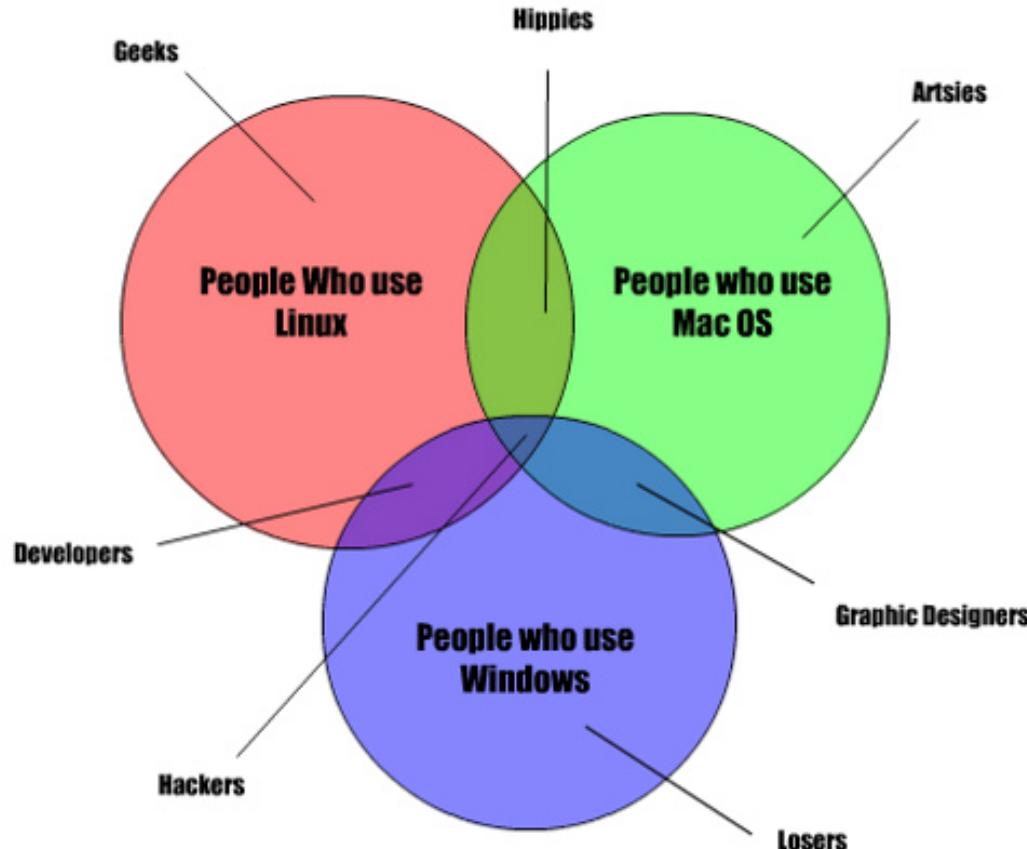
¿Qué es un Sistema Operativo (SO)?

- “Un sistema operativo es un programa que permite al usuario interactuar con el ordenador y sus componentes (monitor, disco duro, impresora, etc.) y que facilita la realización de tareas básicas como copiar o mover ficheros de un sitio a otro, editar archivos de texto, establecer una conexión a internet o hacer copias de seguridad” (Sánchez, 2009) .
- El sistema operativo es el primer programa que se ejecuta al encender el ordenador.





Operating Systems



Linux (geeks):

- Libre: Se puede descargar de internet, se puede copiar y distribuir sin que por ello se incurra en ningún tipo de delito.
Licencia GNU (www.gnu.org)
- Hecho por voluntarios: Linux no se creó para obtener beneficios con él sino para satisfacer una serie de necesidades a la hora de trabajar con el ordenador.
- Multiusuario: Varios usuarios pueden conectarse y usar el mismo ordenador a la vez.
- Multitarea: Pueden funcionar varios programas al mismo tiempo en la misma máquina.
- Multiplataforma: Hay versiones de Linux para gran cantidad de plataformas: todos los PCs basados en procesadores Intel o AMD, ordenadores Digital/Compaq con procesadores Alpha, ordenadores Apple, todos los smartphones usan una versión de Linux.

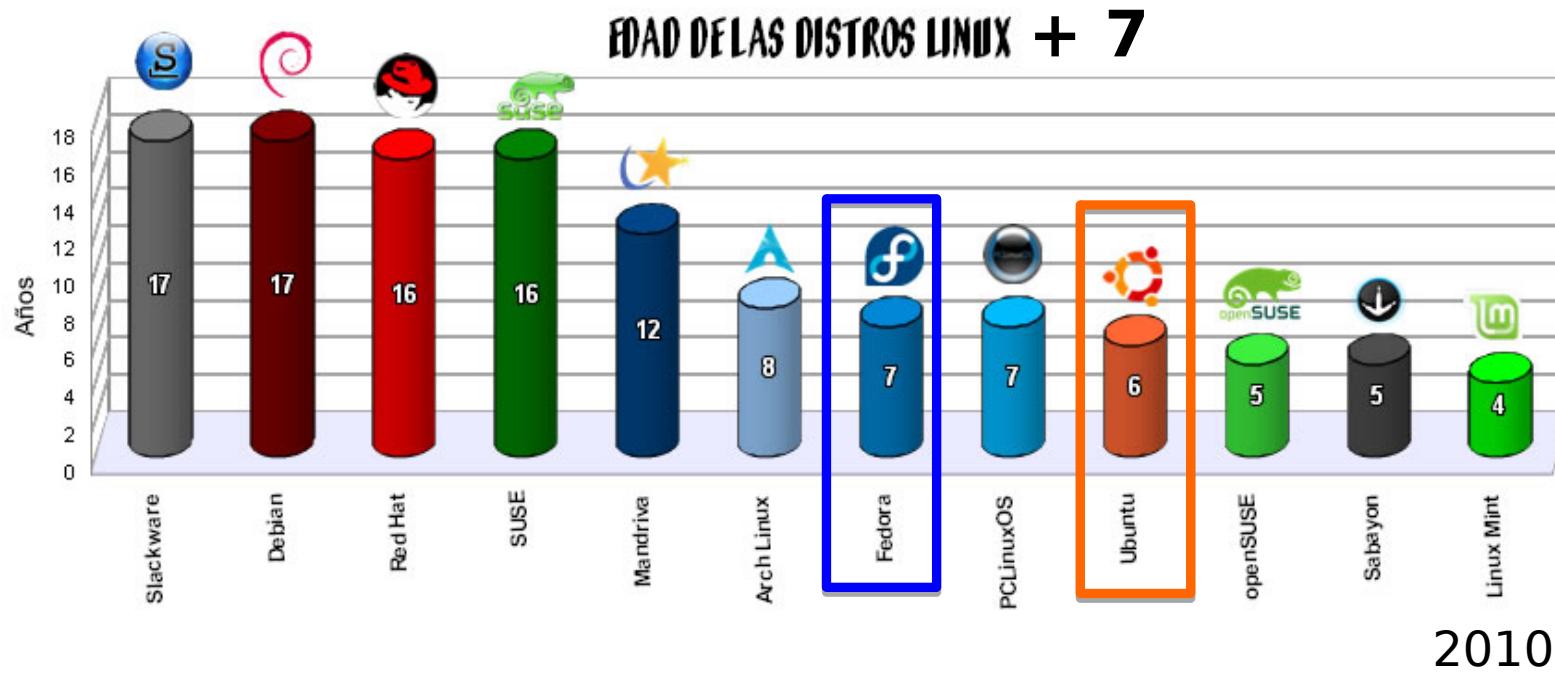
- Estable: Linux es un sistema operativo muy maduro, probado durante mucho tiempo. Hay muchos servidores que llevan funcionando bajo Linux de forma ininterrumpida muchos años sin un solo cuelgue.
- Eficiente: Linux aprovecha bien los recursos hardware. Incluso los viejos Pentium pueden funcionar bien con Linux y servir para alguna tarea.
- Hay miles de programas libres: Hay una gran cantidad de programas, desde procesadores de texto hasta programas de dibujo pasando por todo tipo de servidores, totalmente libres y gratuitos que se pueden descargar e instalar desde el propio entorno de Linux.



Distribución Linux:

Distribución = Núcleo de Linux + Programa de instalación + Aplicaciones

Kernel



¿Qué se hace en Linux aplicado a ciencias?

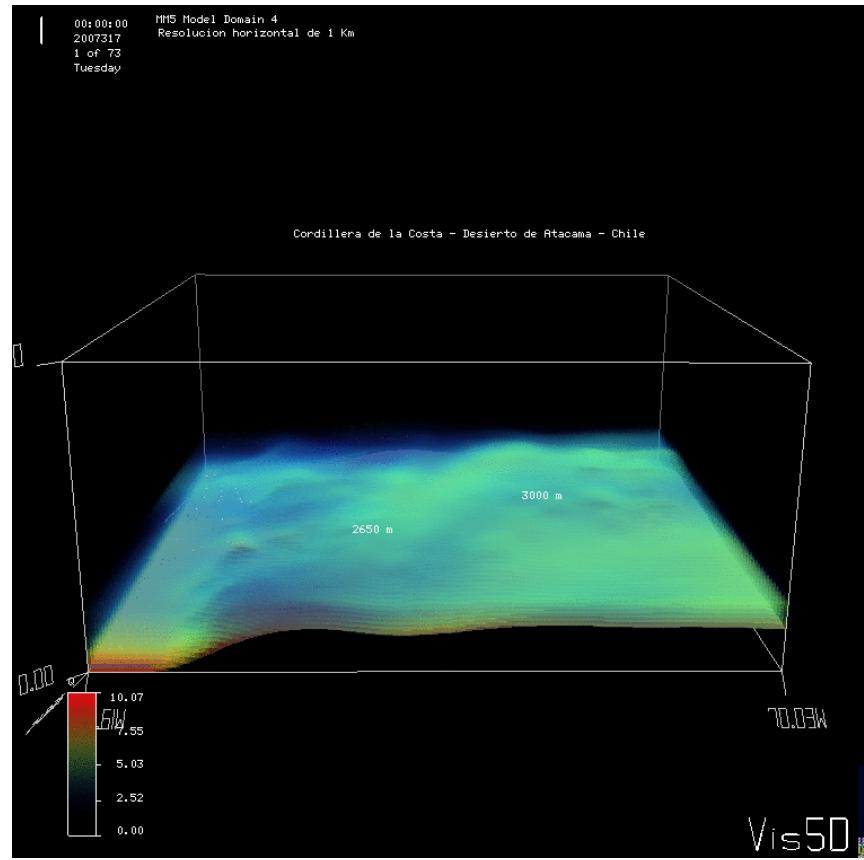
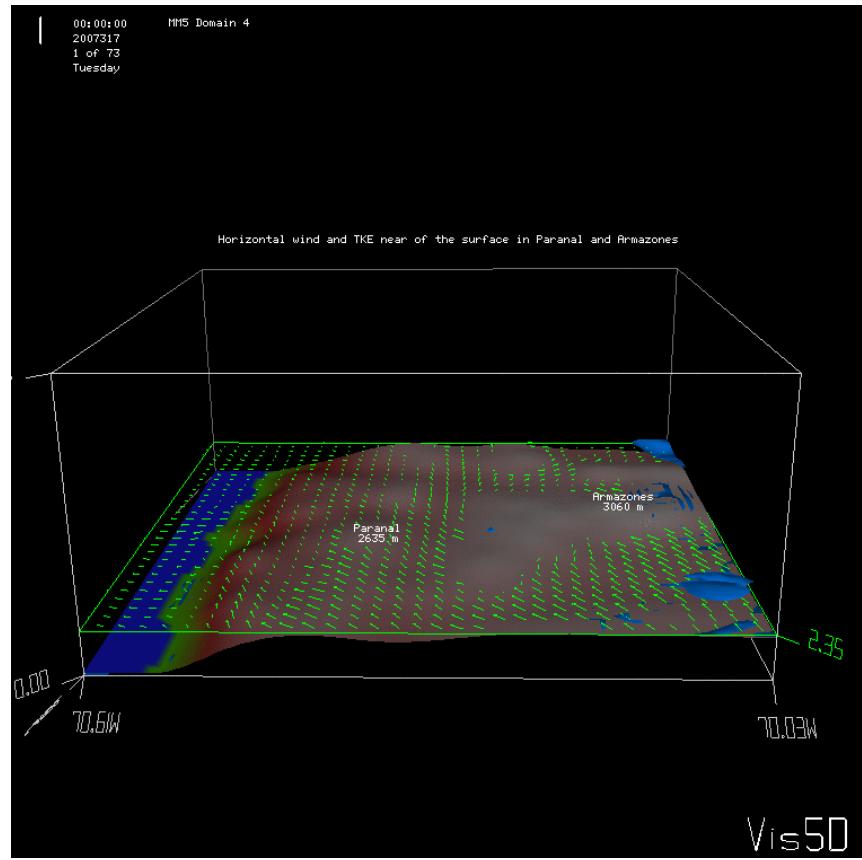
Instituto de Física y Astronomía (IFA)

Software para análisis, cálculos y simulaciones en:

- **Física:** dinámica, hidrodinámica, física cuántica, etc.
- **Astronomía y Astrofísica:** análisis de datos, simulaciones del Universo y las galaxias
- **Meteorología:** simulaciones de la atmósfera



AstroMeteorología



Conexión remota: ssh (Secure Shell)

Linux permite fácilmente establecer una conexión remota hacia un computador, servidor o cluster para realizar tareas.

- Conexión a través de línea de comandos para ejecutar tareas, editar archivos, realizar cálculos, monitorear procesos, etc.
- Muy útil para uso de supercomputadores, servidores y computo de alto nivel. Ejemplo: supercómputo en un cluster.

Supercómputo

El Supercómputo también conocido como HPC (High Performance Computing) es una práctica que consigue alcanzar niveles muy altos de rendimiento en la ejecución de procesos computacionales, lo que nos permite solucionar problemas tan complejos que serían imposibles de llevar a cabo en un equipo convencional o tan grandes que el tiempo de solución supondría un nuevo problema. Esto se logra a través de 2 factores principales:

1. Infraestructura:

Un equipo de HPC es un arreglo de servidores que tienen la finalidad de sumar recursos de procesamiento, memoria y almacenamiento.

2. Análisis de optimización:

La configuración del proceso, antes de su ejecución, es esencial en el aprovechamiento de los recursos de HPC.

Características:

1. Velocidad de procesamiento : miles de millones de instrucciones de punto flotante por segundo.
2. Usuarios a la vez: hasta miles, en entorno de redes amplias.
3. Tamaño: requieren instalaciones especiales y aire acondicionado industrial.
4. Dificultad de uso: solo para especialistas.
5. Clientes usuales: grandes centros de investigación.
6. Penetración social: prácticamente nula.
7. Impacto social: muy importante en el ámbito de la investigación, ya que provee cálculos a alta velocidad de procesamiento, permitiendo, por ejemplo, calcular en secuencia el genoma humano, número π , desarrollar cálculos de problemas físicos dejando un margen de error muy bajo, etc.
8. Parques instalados: menos de un millar en todo el mundo.
9. Hardware : Principal funcionamiento operativo

Principales usos:

1. Mediante el uso de supercomputadoras, los investigadores modelan el clima pasado y el clima actual y predicen el clima futuro .
2. Los astrónomos, astrofísicos y los científicos del espacio utilizan las supercomputadoras para estudiar el Sol, simular explosiones de supernovas en el espacio.
3. Los científicos usan supercomputadoras para simular de qué manera un tsunami podría afectar una determinada costa o ciudad.

Principales usos:

4. Las supercomputadoras se utilizan para probar la aerodinámica de los más recientes aviones militares.
5. Las supercomputadoras se están utilizando para modelar cómo se doblan las proteínas y cómo ese plegamiento puede afectar a la gente que sufre la enfermedad de Alzheimer, la fibrosis quística y muchos tipos de cáncer.
6. Las supercomputadoras se utilizan para modelar explosiones nucleares, limitando la necesidad de verdaderas pruebas nucleares.



Universidad
de Valparaíso
CHILE



**¿Y que tenemos en nuestro
instituto (IFA)?**



→ **MAD**

- **carlfriedrich**
- **hiparco**
- **stellarwind**
- **beast**
- **astromet**
- **astromet 1 y 2**
- **astromet 3 y 4**
- **math 0...4**
- **windenergy**
- **kosmos**
- **(front end)**

Cluster kosmos
512 cores

→ **UPS (APC)**

CPUs Total:

416

Hosts up:

15

Hosts down:

0

Current Load Avg (15, 5, 1m):

52%, 59%, 62%

Avg Utilization (last hour):

45%

Utilization heatmap



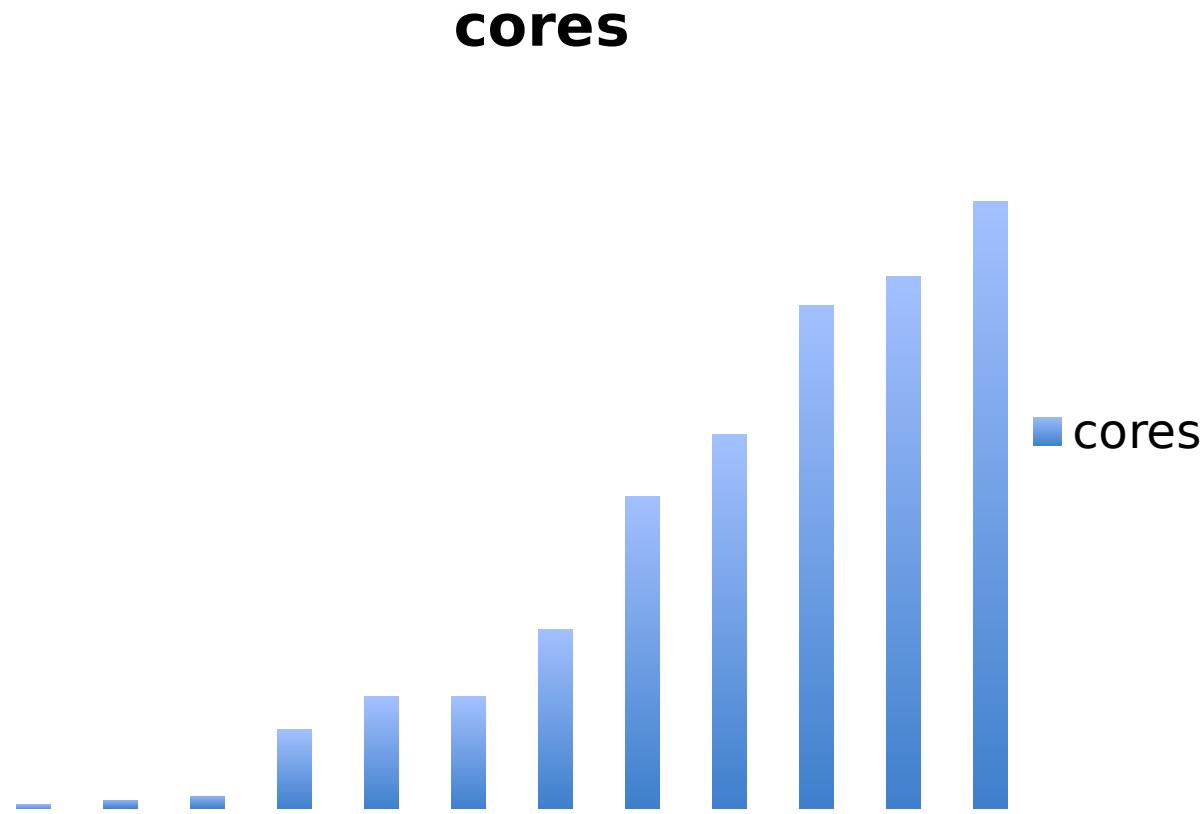
Stacked Graph - load_one



Características:

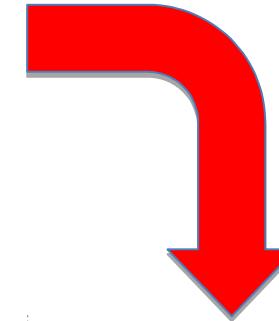
SO Linux: Rocks cluster, con Centos (free de Red Hat)

Storage de 20 TB (~20.000 GB)



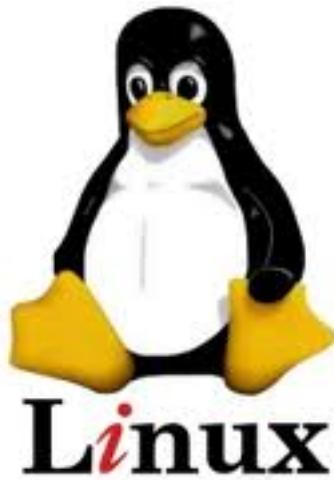
¿Cuando comenzamos a usar el cluster?

```
root@comet:~          root@comet:~          root@:/usr/portage/lo...          root@comet:/lib/rc/net [ /home/donnie ]  
->>> Downloading 'http://www.kernel.org/pub/software/scm/git/git-manpages-1.5.5.tar.bz2'  
--2008-04-14 12:46:26-- http://www.kernel.org/pub/software/scm/git/git-manpages-1.5.5.tar.bz2  
Resolving www.kernel.org... 204.152.191.5, 204.152.191.37  
Connecting to www.kernel.org|204.152.191.5|:80... connected.  
HTTP request sent, awaiting response... 200 OK  
Length: 162609 (159K) [application/x-bzip2]  
Saving to: `/usr/portage/distfiles/git-manpages-1.5.5.tar.bz2'  
  
100%[=====] 162,609      127K/s   in 1.2s  
2008-04-14 12:46:27 (127 KB/s) - `/usr/portage/distfiles/git-manpages-1.5.5.tar.bz2' saved [162609/162609]  
  
>>> Creating Manifest for /usr/portage/local/overlay/dev-util/git  
donnie@comet $ cp /usr/portage/dev-util/git/files/  
20080322-git-1.5.5.4-noperl.patch git-1.5.0-symlinks.patch      git-daemon.conf  
70git-gentoo.el          git-1.5.3.8-t9101.patch      git-daemon.initd  
71git-gentoo.el          git-1.5.3.8-t9106.patch      git-daemon.xinetd  
72git-gentoo.el          git-1.5.3-symlinks.patch  
donnie@comet $ cp /usr/portage/dev-util/git/files/20080322-git-1.5.5.4-noperl.patch files/  
donnie@comet $ cp /usr/portage/dev-util/git/files/git-1.5.3-symlinks.patch files/  
donnie@comet $ ebuild git-1.5.5.ebuild digest  
>>> Creating Manifest for /usr/portage/local/overlay/dev-util/git  
donnie@comet $ cd  
donnie@comet $ rsync -avzP rsync://rsync.gentoo.org/gentoo-portage .  
Welcome to crane.gentoo.org  
  
Server Address : 134.68.220.74  
Contact Name   : mirror-admin@gentoo.org  
Hardware       : 2 x Intel(R) Xeon(TM) CPU 1700MHz, 2019MB RAM  
  
Please note: common gentoo-netiquette says you should not sync more  
than once a day. Users who abuse the rsync.gentoo.org rotation  
may be added to a temporary ban list.  
  
MOTD autogenerated by update-rsync-motd on Thu Apr  3 18:13:41 UTC 2008  
  
receiving file list ...  
^Crsync error: received SIGINT, SIGTERM, or SIGHUP (code 20) at rsync.c(541) [receiver=3.0.2]  
donnie@comet $ man wiggle  
donnie@comet $ ./etc/profile && . ~/.bashrc  
donnie@comet $ sudo su -  
Last login: Mon Apr 14 14:20:55 PDT 2008 from comet on pts/6  
comet ~# . ~/.bashrc  
comet # pwd  
/root  
comet # exit  
logout  
donnie@comet $
```



**Cuando dominemos
el terminal (consola,
modo texto, línea de
comandos) !!**

get started





Estructura de directorios

/ directorio raíz	/bin	Contiene programas ejecutables básicos para el sistema.
	/boot	Contiene los ficheros necesarios para el arranque del sistema.
	/dev	Contiene los ficheros correspondientes a los dispositivos: sonido, impresora, disco duro, lector de cd/dvd, video, etc.
	/etc	Contiene ficheros y directorios de configuración.
	/home	Contiene los directorios de trabajo de los usuarios. Cada usuario tiene su propio directorio en el sistema dentro de /home/.
	/lib	Contiene las librerías compartidas y los módulos del kernel
	/media	Dentro de este directorio se montan los dispositivos como el CD-ROM, memorias USB, discos duros portátiles, etc
	/opt	Directorio reservado para instalar aplicaciones.
	/sbin	Contiene los ficheros binarios ejecutables del sistema operativo.
	/srv	Contiene datos de los servicios proporcionado por el sistema.
	/tmp	Directorio de archivos temporales.
	/usr	Aquí se encuentran la mayoría de los archivos del sistema, aplicaciones, librerías, manuales, juegos... Es un espacio compartido por todos los usuarios.
	/var	Contiene archivos administrativos y datos que cambian con frecuencia: registro de errores, bases de datos, colas de impresión, etc.
	/root	Directorio de trabajo del administrador del sistema (usuario root).
	/proc	Aquí se almacenan datos del kernel e información sobre procesos.

Entrada al sistema: login

```
bash          omar@Cn2: ~ (ssh)      bash      bash      bash
Omars-iMac:~$ ssh omar@10.100.90.150
omar@10.100.90.150's password:
Linux Cn2 3.2.0-4-amd64 #1 SMP Debian 3.2.54-2 x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
You have mail.

Last login: Wed Mar 12 14:48:08 2014 from omars-imac.local
omar@Cn2:~$
```

Nombre usuario

Nombre del computador (host)

Prompt consola –
bash (shell)

Visualización, creación y cambio de directorio

pwd

El comando **pwd** muestra cuál es el directorio de trabajo actual, en otras palabras, le dice al usuario dónde se encuentra dentro de la estructura de directorios del sistema. Es muy útil cuando estamos perdidos.



A screenshot of a terminal window with a grey header bar. The header bar has three colored circles (red, yellow, green) on the left, followed by the word "bash". In the center, it shows "omar@Cn2: ~ (ssh)". On the right, it shows "bash" again. The main window area is white and contains the following text:

```
omar@Cn2:~$ pwd
/home/omar
omar@Cn2:~$
```

ls

El comando **ls** muestra el contenido del directorio actual. Por defecto, los archivos ocultos no se muestran. Éste es seguramente el comando que más se utiliza.



```
1. omar@Cn2: ~ (ssh)
bash                                     bash
omar@Cn2:~$ ls
Desktop  Documents  Downloads  Music  Pictures  Public  Soft  Templates  Videos
omar@Cn2:~$ █
```

Opciones:

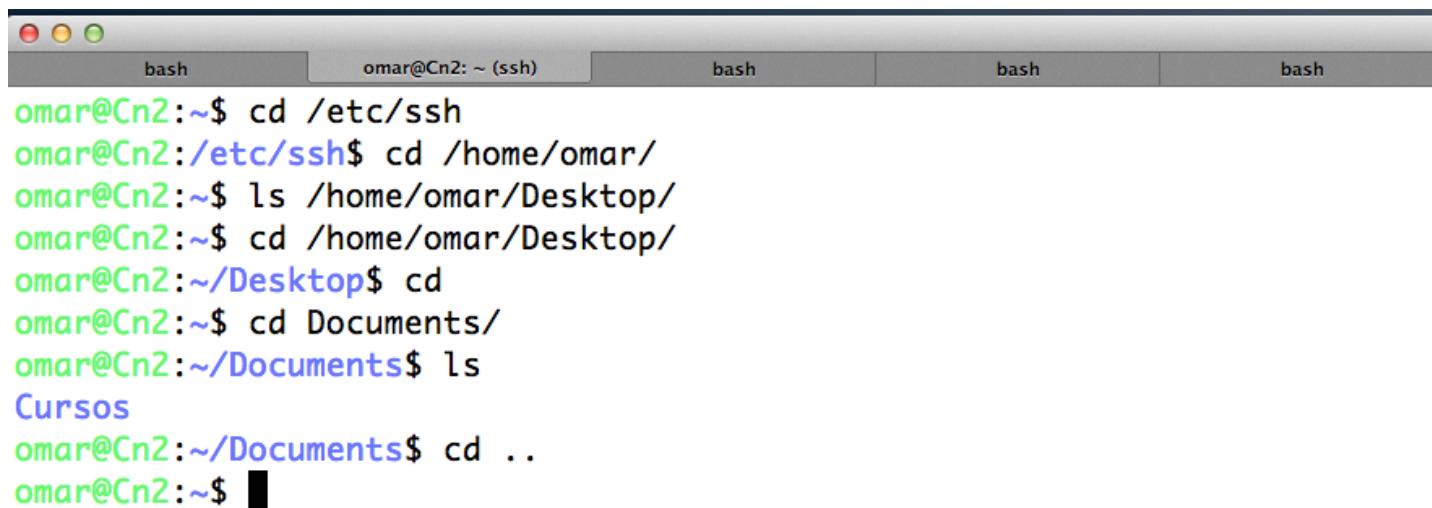
ls -a : muestra todos los archivos, incluyendo los ocultos (cuyo nombre comienza por un punto)

ls -l : muestra un listado detallado, con la última fecha de modificación de cada archivo, el tamaño, etc.

ls -h : muestra el tamaño de los ficheros en bytes, Kb, Mb, etc.

cd

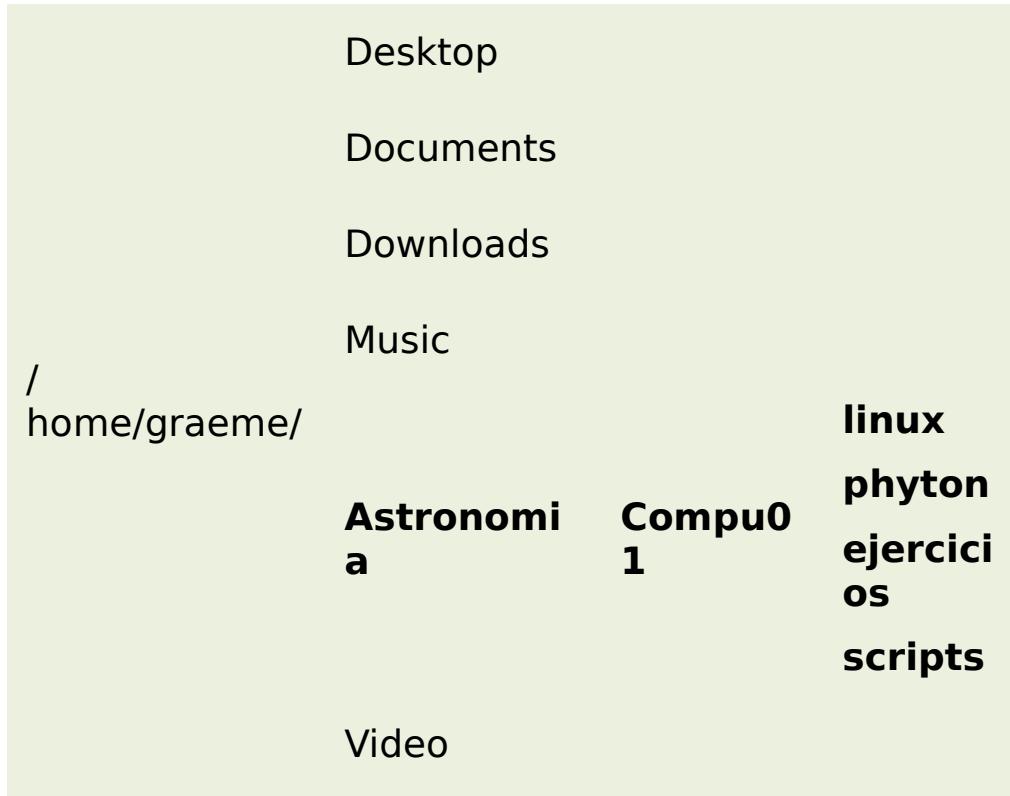
El comando **cd (change dir)** permite cambiar de directorio. Si se utiliza tal cual, sin ningún tipo de argumento, cambia al directorio de trabajo personal. Si se utiliza seguido de una ruta, cambia al directorio que se indica.



```
bash omar@Cn2: ~ (ssh) bash bash bash
omar@Cn2:~$ cd /etc/ssh
omar@Cn2:/etc/ssh$ cd /home/omar/
omar@Cn2:~/home/omar/Desktop/
omar@Cn2:~/home/omar/Desktop/
omar@Cn2:~/Desktop$ cd
omar@Cn2:~/Documents$ cd Documents/
omar@Cn2:~/Documents$ ls
Cursos
omar@Cn2:~/Documents$ cd ..
omar@Cn2:~$
```

mkdir

Se pueden crear directorios con el comando **mkdir**. Por ejemplo, para crear una estructura de carpetas donde un estudiante guardará información sobre sus asignaturas según el siguiente esquema:



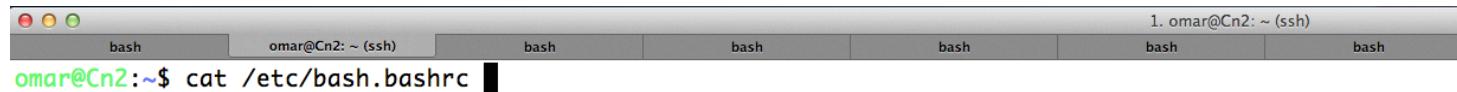


```
omar@Cn2:~$ mkdir Fisica
omar@Cn2:~$ mkdir Fisica/Compu01
omar@Cn2:~$ ls
Desktop Documents Downloads Fisica Music Pictures Public Soft Templates Videos
omar@Cn2:~$ cd Fisica/
omar@Cn2:~/Fisica$ ls
Compu01
omar@Cn2:~/Fisica$ cd Compu01/
omar@Cn2:~/Fisica/Compu01$ ls
omar@Cn2:~/Fisica/Compu01$ mkdir linux
omar@Cn2:~/Fisica/Compu01$ mkdir phyton
omar@Cn2:~/Fisica/Compu01$ mkdir ejercicios
omar@Cn2:~/Fisica/Compu01$ mkdir scripts
omar@Cn2:~/Fisica/Compu01$ ls
ejercicios linux phyton scripts
omar@Cn2:~/Fisica/Compu01$ pwd
/home/omar/Fisica/Compu01
omar@Cn2:~/Fisica/Compu01$ █
```

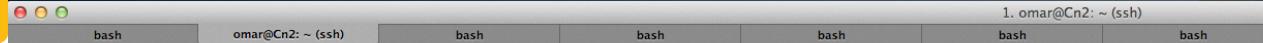
Visualización de ficheros o archivos

cat

El comando **cat** muestra por pantalla el contenido de un fichero y, cuando termina, el usuario está otra vez de vuelta en la línea de comandos.



A screenshot of a terminal window with multiple tabs. The active tab shows the command `omar@Cn2:~$ cat /etc/bash.bashrc` being typed. The terminal is running on a Mac OS X system, as indicated by the window title bar which reads "1. omar@Cn2: ~ (ssh)".



Visualización

cat

```
# set variable identifying the chroot you work in (used in the prompt below)
if [ -z "${debian_chroot:-}" ] && [ -r /etc/debian_chroot ]; then
    debian_chroot=$(cat /etc/debian_chroot)
fi

# set a fancy prompt (non-color, overwrite the one in /etc/profile)
PS1='${debian_chroot:+($debian_chroot)}\u@\h:\w\$ '
```

El comando cat muestra el contenido de un archivo. El usuario está

na, el

```
# Commented out, don't overwrite xterm -T "title" -n "icontitle" by default.
# If this is an xterm set the title to user@host:dir
#case "$TERM" in
#xterm*|rxvt*)
#    PROMPT_COMMAND='echo -ne "\033]0;${USER}@${HOSTNAME}: ${PWD}\007"'
#    ;;
#*)
#    ;;
#esac
```

omar@Cn2:~\$

```
# enable bash completion in interactive shells
#if ! shopt -oq posix; then
#  if [ -f /usr/share/bash-completion/bash_completion ]; then
#    . /usr/share/bash-completion/bash_completion
#  elif [ -f /etc/bash_completion ]; then
#    . /etc/bash_completion
#  fi
#fi

# if the command-not-found package is installed, use it
if [ -x /usr/lib/command-not-found -o -x /usr/share/command-not-found/command-not-found ]; then
    function command_not_found_handle {
        # check because c-n-f could've been removed in the meantime
        if [ -x /usr/lib/command-not-found ]; then
            /usr/bin/python /usr/lib/command-not-found -- "$1"
            return $?
        elif [ -x /usr/share/command-not-found/command-not-found ]; then
            /usr/bin/python /usr/share/command-not-found/command-not-found -- "$1"
            return $?
        else
            printf "%s: command not found\n" "$1" >&2
            return 127
        fi
    }
fi
omar@Cn2:~$
```



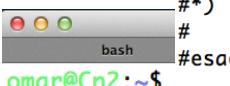
```
# set variable identifying the chroot you work in (used in the prompt below)
if [ -z "${debian_chroot:-}" ] && [ -r /etc/debian_chroot ]; then
    debian_chroot=$(cat /etc/debian_chroot)
fi

# set a fancy prompt (non-color, overwrite the one in /etc/profile)
PS1='${debian_chroot:+($debian_chroot)}\u@\h:\w\$ '
```

cat

El comando **cat** muestra el contenido del fichero bash.bashrc que está dentro del directorio /etc. Se visualiza en la terminal:

```
# Commented out, don't overwrite xterm -T "title" -n "icontitle" by default.
# If this is an xterm set the title to user@host:dir
#case "$TERM" in
#xterm*|rxvt*)
#    PROMPT_COMMAND='echo -ne "\033]0;${USER}@${HOSTNAME}: ${PWD}\007"'
#    ;;
#*)
#    ;;
#esac
```



omar@Cn2:~\$

muestra el contenido del fichero bash.bashrc que está dentro del directorio /etc. Se darán cuenta que es imposible ver todo el contenido de este fichero, porque ha pasado por pantalla muy rápido. Por eso **cat** se suele utilizar para visualizar el contenido de archivos pequeños.

```
# if the command-not-found package is installed, use it
if [ -x /usr/lib/command-not-found -o -x /usr/share/command-not-found/command-not-found ]; then
    function command_not_found_handle {
        # check because c-n-f could've been removed in the meantime
        if [ -x /usr/lib/command-not-found ]; then
            /usr/bin/python /usr/lib/command-not-found -- "$1"
            return $?
        elif [ -x /usr/share/command-not-found/command-not-found ]; then
            /usr/bin/python /usr/share/command-not-found/command-not-found -- "$1"
            return $?
        else
            printf "%s: command not found\n" "$1" >&2
            return 127
        fi
    }
fi
```

omar@Cn2:~\$ █

more

El comando **more** hace lo mismo que cat, a diferencia de que muestra el fichero pantalla a pantalla, es decir, llena de texto la pantalla y se espera a que el usuario pulse la tecla <espacio> para pasar a la siguiente:



bash omar@Cn2: ~ (ssh) bash bash bash bash bash bash

1. omar@Cn2: ~ (ssh)

```
omar@Cn2:~$ more /etc/bash.bashrc
# System-wide .bashrc file for interactive bash(1) shells.

# To enable the settings / commands in this file for login shells as well,
# this file has to be sourced in /etc/profile.

more
# If not running interactively, don't do anything
[ -z "$PS1" ] && return

El comando # check the window size after each command and, if necessary,
pantalla, es # update the values of LINES and COLUMNS.
| shopt -s checkwinsize
<espacio> p # set variable identifying the chroot you work in (used in the prompt below)
if [ -z "${debian_chroot:-}" ] && [ -r /etc/debian_chroot ]; then
    debian_chroot=$(cat /etc/debian_chroot)
fi

# set a fancy prompt (non-color, overwrite the one in /etc/profile)
PS1='${debian_chroot:+($debian_chroot)}\u@\h:\w\$ '

# Commented out, don't overwrite xterm -T "title" -n "icontitle" by default.
# If this is an xterm set the title to user@host:dir
#case "$TERM" in
#xterm*|rxvt*)
#    PROMPT_COMMAND='echo -ne "\033]0;${USER}@${HOSTNAME}: ${PWD}\007"'
#    ;;
#*)
#    ;;
#esac

# enable bash completion in interactive shells
#if ! shopt -oq posix; then
#    if [ -f /usr/share/bash-completion/bash_completion ]; then
#        . /usr/share/bash-completion/bash_completion
#    elif [ -f /etc/bash_completion ]; then
#        . /etc/bash_completion
#    fi
#fi

# if the command-not-found package is installed, use it
if [ -x /usr/lib/command-not-found -o -x /usr/share/command-not-found/command-not-found ]; then
    function command_not_found_handle {
        # check because c-n-f could've been removed in the meantime
        if [ -x /usr/lib/command-not-found ]; then
--More--(81%)
```

talla a
a

less

El comando **less** es el más versátil de los tres, ya que permite moverse hacia delante y hacia atrás dentro del fichero, utilizando los cursores o las teclas de “AvPág” y “RePág”.

En cualquier momento se puede interrumpir la visualización y volver al símbolo del sistema pulsando la letra “q”.

head y **tail**

Los comandos **head** y **tail** permiten mostrar de forma parcial el contenido de un fichero. Como su nombre indica, **head** muestra las primeras líneas del fichero (la cabecera) y **tail** muestra las últimas líneas (la cola).

less

El comando **less** es el más versátil de los tres, ya que permite moverse hacia delante y hacia atrás dentro del fichero, utilizando los cursores o las teclas de “AvPág” y “RePág”.

En cualquier momento se puede interrumpir la visualización y volver al símbolo del sistema pulsando la letra “q”.

head y tail

Los comandos **head** y **tail** nos permiten visualizar el principio y final de un fichero respectivamente. Como veremos en el siguiente ejemplo, ambos comandos nos muestran las primeras y últimas líneas de un fichero.

```
1. omar@Cn2: ~ (ssh) o. Como
   bash                         muestra las
omar@Cn2:~$ head /etc/bash.bashrc
bash                         últim
# System-wide .bashrc file for interactive bash(1) shells.

# To enable the settings / commands in this file for login shells as well,
# this file has to be sourced in /etc/profile.

# If not running interactively, don't do anything
[ -z "$PS1" ] && return

# check the window size after each command and, if necessary,
# update the values of LINES and COLUMNS.
omar@Cn2:~$                                omar@Cn2:~$
omar@Cn2:~$                                omar@Cn2:~$
omar@Cn2:~$ tail /etc/bash.bashrc
        return $?
    elif [ -x /usr/share/command-not-found/command-not-found ]; then
        /usr/bin/python /usr/share/command-not-found/command-not-found -- "$1"
        return $?
    else
        printf "%s: command not found\n" "$1" >&2
        return 127
    fi
}
fi
omar@Cn2:~$
```

Edición de ficheros

touch, nano, vi

```
1. omar@Cn2: ~/Fisica/Compu01/linux (ssh)
bash                                bash                                bash                                bash                                bash                                bash                                bash
omar@Cn2:~/Fisica/Compu01/linux$ touch ejemplo1.txt
omar@Cn2:~/Fisica/Compu01/linux$ nano ejemplo1.txt
omar@Cn2:~/Fisica/Compu01/linux$ vi ejemplo1.txt
omar@Cn2:~/Fisica/Compu01/linux$ vi ejemplo1.txt
omar@Cn2:~/Fisica/Compu01/linux$
omar@Cn2:~/Fisica/Compu01/linux$ ls
ejemplo1.txt
omar@Cn2:~/Fisica/Compu01/linux$
```



```
GNU nano 2.2.6          File: ejemplo1.txt          Modified
Mi primer archivo creado en linux (geek)
```

Para guardar: “ctrl” “o”
Para salir: “ctrl” “x”

^G Get Help
^X Exit

^O WriteOut
^J Justify

^R Read File
^W Where Is

^Y Prev Page
^V Next Page

^K Cut Text
^U Uncut Text

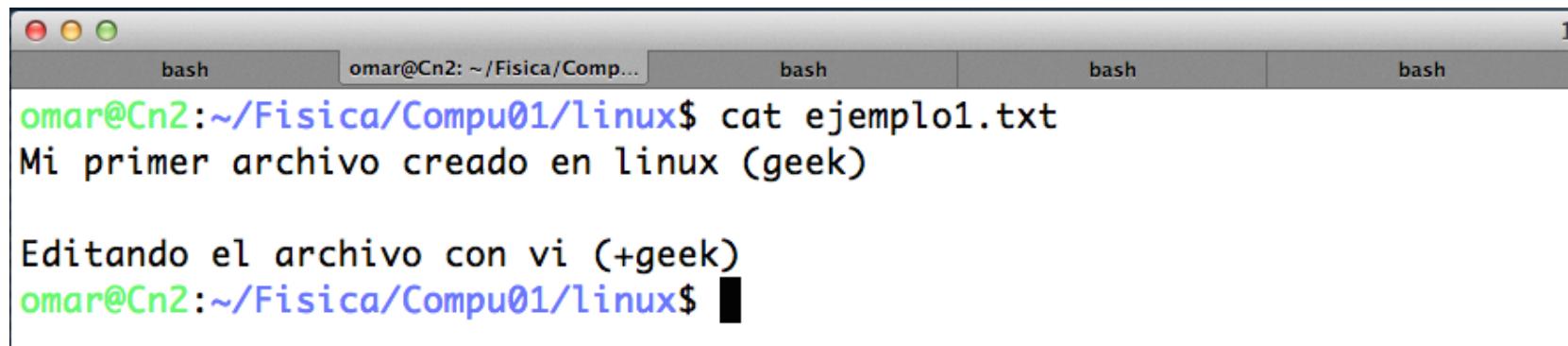
^C Cur Pos
^T To Spell

Vi

Hemos dejado para el final al editor de Linux por excelencia, se trata de **vi**. A primera vista es el más difícil de utilizar, lo cual es cierto, y parece que tiene menos opciones, pero muy al contrario se trata de un programa muy potente. Cualquier “linuxero” que se precie debe saber manejar bien este programa. Añadiremos una línea más al archivo ejemplo1.txt. Para ello, seguiremos los siguientes pasos:

- Pulsar la letra “i” para entrar en modo “edición”.
- Escribiremos el texto.
- Pulsar la tecla ESC para salir del modo “edición”.
- Teclear “:” + “w” + INTRO para grabar los cambios.
- Teclear “:” + “q” + INTRO para salir del programa.

Comprobamos que el archivo quedo bien editado:



```
omar@Cn2:~/Fisica/Compu01/linux$ cat ejemplo1.txt
Mi primer archivo creado en linux (geek)

Editando el archivo con vi (+geek)
omar@Cn2:~/Fisica/Compu01/linux$ █
```

Es más que recomendable realizar el tutorial llamado ***vimtutor***

Resumen:

- Todo usuario necesita un nombre y una contraseña para entrar en el sistema.
- La información se almacena en directorios y subdirectorios (carpetas y subcarpetas).
- Hay una serie de directorios predefinidos como /bin, /dev, /home, /etc, /var, etc. para todos los sistemas Linux.
- Hay rutas absolutas, que comienzan por el carácter “/”, y que definen una ruta efectiva completa y rutas relativas, que no comienzan por el carácter “/”, y cuya ruta efectiva sería la concatenación del directorio actual con esa misma ruta relativa.

Comandos vistos hasta ahora:

<i>Comando</i>	<i>Acción</i>	<i>Ejemplo</i>	
pwd	muestra el directorio actual	pwd	
ls	lista ficheros y directorios	ls -l	
cd	cambia de directorio	cd mp3/wim_mertens	
mkdir	crea uno o varios directorios	mkdir cartas facturas	
cat	visualiza un fichero	cat /var/log/dmesg	
more	visualiza un fichero pantalla a pantalla	more /var/log/dmesg	
less	visualiza un fichero pantalla a pantalla y permite retroceder	less /var/log/dmesg	
head	visualiza las primeras filas de un fichero	head -n5 /var/log/dmesg	
tail	visualiza las últimas filas de un fichero	tail /var/log/dmesg	
touch	crea un fichero vacío	touch listado.txt	
*	ee	editor de textos muy simple	ee listado.txt
*	mcedit	editor de textos que forma parte de Midnight Commander	mcedit listado.txt
*	vi	editor de textos muy potente	vi listado.txt
*	apt-get	instala y desinstala programas	apt-get install mc
!!!	man	muestra ayuda sobre un determinado comando	man ls

Ejercicios:

1. En la carpeta de usuario cree el siguiente directorio

Fisica/LFIS-116/Tarea1

2. Dentro del directorio Tarea1 cree un archivo usando **vi** que contenga

Nombre:

Año de ingreso:

Mención: