



Universidad de Murcia

Facultad de Informática

Universidad Politécnica de Cartagena Escuela Técnica Superior de Ingeniería de Telecomunicación

# Título de Grado en Ciencia e Ingeniería de Datos **Fundamentos de Computadores**

Práctica 1: Introducción al shell de Linux.

Boletines de prácticas

CURSO 2022 / 23

## Índice general

I.	Boletines de prácticas	2
	B1.1. Introducción	2
	B1.2. Boletín 0: Máquina Virtual de Linux	5
	B1.2.1. Objetivos	5
	B1.2.2. Plan de trabajo	5
	B1.2.3. Ejercicios a realizar durante la sesión	5
	B1.3. Boletín 1: Primer contacto con el <i>shell</i>	8
	B1.3.1. Objetivos	8
	B1.3.2. Plan de trabajo	8
	B1.3.3. El prompt	8
	B1.3.4. El directorio de trabajo actual	8
	B1.3.5. El directorio raíz	9
	B1.3.6. El directorio de usuario	9
	B1.3.7. Rutas absolutas y relativas	10
	B1.3.8. Sintaxis general de un comando	10
	B1.3.9. Autocompletar rutas mediante la tecla tabulador	11
	B1.3.10Comandos básicos de manejo de ficheros	11
	B1.3.11El comando script	12
	B1.3.12Ejercicios a realizar durante la sesión	12
	B1.3.13Referencias	18
	B1.4. Boletín 2: Gestión de paquetes en Linux	19
	B1.4.1. Objetivos	19
	B1.4.2. Gestión de paquetes en Linux	19
	B1.4.3. Plan de trabajo	20
	B1.4.4. Ejercicios a realizar durante la sesión	20
	B1.5. ANEXO: Instalación de Máquina Virtual de Linux con archivo ISO	21
	B1.5.1. Objetivos	21
	B1.5.2. Plan de trabajo	21
	B1.5.3. Ejercicios a realizar durante la sesión	21



### **Boletines de prácticas**

### **B1.1.** Introducción

Un **Sistema Operativo** (**SO**) es el software intermediario entre el usuario de un ordenador y el hardware de éste. El SO proporciona un entorno necesario para que el usuario pueda ejecutar programas de forma conveniente. Linux es un SO de código abierto, gratuito, que ha sido desarrollado de forma colaborativa por programadores de todas las partes del mundo. A diferencia de Windows y MacOS, que han sido diseñados con código cerrado para uso comercial, Linux se puede distribuir, personalizar e incluso modificar de forma libre. Esta característica ha sido un factor clave para su gran crecimiento y aceptación desde su nacimiento en 1991 hasta hoy en día.

Todo SO ofrece una serie de servicios tales como ejecución de programas, operaciones de entrada/salida (e.g., imprimir, leer un CD, etc.), manipulación de ficheros, comunicación entre procesos en el mismo ordenador o entre procesos externos, y funciones de seguridad, entre otros. Estos servicios proporcionan a programas y usuarios las herramientas para poder llevar a cabo sus tareas cometidas. Como se muestra en la Figura I.1, un SO ofrece estos servicios a través de llamadas al sistema que se pueden acceder mediante diferentes tipos de interfaces con el usuario, entre las que destacan la *interfaz de usuario gráfica* (GUI, por sus siglas en inglés, *graphic user interface*) y la interfaz textual o de línea de comandos (CLI, por sus siglas en inglés, *command line interface*).



Figura I.1: Arquitectura de servicios de sistema operativo

#### Un poco de historia

En los comienzos de la industria informática, uno de los primeros sistemas operativos se llamaba Unix. Estaba diseñado para funcionar como un sistema multiusuario en ordenadores centrales llamados *mainframes*, cuyos usuarios se conectaban a él de forma remota a través de terminales individuales. Estos terminales eran bastante básicos para los estándares modernos: sólo un teclado y una pantalla, sin poder ejecutar programas localmente. En su lugar, se limitaban a enviar las pulsaciones al servidor y a mostrar los datos que recibían en la pantalla. No había ratón, ni gráficos sofisticados, ni siquiera se podía elegir el color. Todo se enviaba como texto y se recibía como texto. Obviamente, cualquier programa que se ejecutara en el *mainframe* tenía que producir texto como salida y aceptar texto como entrada.

En comparación con los gráficos, el texto consume muy pocos recursos. Incluso en las máquinas de los años 70, que ejecutaban cientos de terminales a través de conexiones de red increíblemente lentas según los estándares actuales, los usuarios podían interactuar con los programas de forma rápida y eficiente. Los comandos también se mantenían muy concisos para reducir el número de pulsaciones necesarias, lo que aceleraba aún más el uso del terminal por





parte de los usuarios. Esta velocidad y eficiencia es una de las razones por las que esta interfaz de texto sigue siendo ampliamente utilizada hoy en día.

Pese a que ha transcurrido medio siglo, los usuarios de un *mainframe* Unix utilizaban el terminal para realizar el mismo tipo de tareas de gestión de ficheros que hoy día se puede realizar con un ratón y un par de ventanas. Ya sea creando ficheros, renombrándolos, poniéndolos en subdirectorios o moviéndolos en el disco, los usuarios de los años 70 podían hacerlo todo con una interfaz textual. Cada una de estas tareas requería su propio programa o comando: uno para cambiar de directorio (cd), otro para listar su contenido (ls), un tercero para renombrar o mover ficheros (mv), etc.

Para coordinar la ejecución de cada uno de estos programas, el usuario se conectaba a un único programa maestro que podía utilizarse para lanzar cualquiera de los otros programas. Al *envolver* los comandos del usuario, a este programa se le conocía como *shell* (en inglés, concha, caparazón). El *shell* podía proporcionar capacidades comunes a cualquiera de ellos, como la posibilidad de pasar datos de un comando directamente a otro, o utilizar caracteres comodín especiales para trabajar con muchos ficheros de nombre similar a la vez. Los usuarios podían incluso escribir código sencillo (llamados *shell scripts*) que podían utilizarse para automatizar largas series de comandos del shell con el fin de facilitar tareas complejas.

El programa shell original de Unix se llamaba simplemente sh, pero ha sido ampliado y superado a lo largo de los años, por lo que en un sistema Linux moderno lo más probable es que utilices un shell llamado bash, aunque actualmente existen otros programas *shell* tales como sh, dash, zsh, ash, etc., todos ellos prácticamente idénticos en los aspectos fundamentales ya que siguen en mayor o menor medida las especificaciones del estándar POSIX.

Linux es una especie de descendiente de Unix. El núcleo de Linux está diseñado para comportarse de manera similar a un sistema Unix, de tal manera que la mayoría de los viejos shells y otros programas basados en texto se ejecutan en él con bastante facilidad. En teoría, incluso se podría conectar uno de esos viejos terminales de los años 70 a un equipo Linux moderno y acceder al shell a través de él. Pero hoy en día es mucho más común usar una terminal de texto emulada en software: esa misma interfaz de texto de estilo Unix, pero ejecutada en una ventana junto a tus programas gráficos.

#### ¿Por qué aprender a usar la interfaz de línea de comandos de Linux?

En la actualidad, la inmensa mayoría de usuarios utilizan cotidianamente una GUI para interactuar con el ordenador a través de ventanas, botones, etc., al resultar una forma intuitiva y fácil de realizar sus tareas. Mientras que una GUI resulta suficiente para cubrir las necesidades a nivel de usuario, aquellos profesionales que desarrollan su trabajo en los diferentes ámbitos de la Informática —desde programadores o administradores de sistemas a científicos de datos— encuentran en la interfaz de línea de comandos una valiosa herramienta, dado el enorme potencial que ofrece a la hora de mejorar su productividad mediante la automatización de tareas recurrentes en el día a día. Por esta razón, a lo largo de las prácticas de la asignatura de Fundamentos de Computadores aprenderemos a utilizar la interfaz de línea de comandos. En este boletín vamos a introducir su manejo básico a través de la práctica de sencillos comandos, lo cual nos servirá de base para el resto de prácticas de la asignatura.

La línea de comandos o *shell* de Linux es una interfaz basada en texto para manejar el ordenador, la cual se suele conocer como terminal o consola de texto. En realidad, el *shell* no es más que un programa que ofrece una ventana de texto en la que, desde el teclado y línea a línea, podemos introducir órdenes o *commandos*. Dichas órdenes son interpretadas por el shell, quien se encarga de analizarlas sintácticamente y procesarlas, lo cual generalmente resulta en la ejecución de otros programas. Los comandos introducidas pueden ser muy sencillas, como por ejemplo para obtener un listado de los ficheros existentes en un directorio, o bien comandos mucho más complicadas que permitan hacer tareas complejas o tediosas.

Los intérpretes de comandos (y de manera muy especial el *shell* de Linux) suelen permitir configurar, personalizar y monitorizar el sistema de una manera muy avanzada, así como automatizar tareas repetitivas y tediosas con el empleo de los llamados *guiones shell (shell scripts)*. Los guiones shell son conjuntos de comandos que se podrían teclear secuencialmente pero que, en lugar de ello, se agrupan dentro de un fichero para posteriormente ejecutarlas cuantas veces queramos como una sola orden, correspondiente al propio nombre del fichero de texto que contiene el guión shell. En un guión shell se usan variables y estructuras de control (*if/while/*etc.), permitiendo que convertirlos



en "pequeños" programas que son ampliamente utilizados en el mundo de Unix/Linux.

Es por ello que, a pesar de que introducir comandos manualmente desde el teclado pueda parecer dificultoso y/o intimidante, el usuario medio de Linux antes o después trabajará habitualmente con una ventana de la consola abierta. Las ventajas de usar el *shell* son, por lo tanto, muchas y merece la pena el esfuerzo inicial que supone su uso. Entre ellas:

- Algunas aplicaciones se lanzan exclusivamente o más cómodamente desde la línea de comandos.
- La labor reiterada de producir distintas salidas de un proceso se puede automatizar con el uso de un *script*.
- Instalar paquetes de programas.
- Hacer copias de seguridad o llevar un repositorio de ficheros.
- Administrar y gestionar usuarios y grupos, y sus correspondientes permisos en ficheros y directorios.
- Personalizar los comandos del SO (añadiendo más, modificando sus aspectos y funcionalidades).
- Velar por el buen estado del sistema (*backups*, agregar y quitar usuarios, instalar, quitar, actualizar y configurar aplicaciones, etc.).
- El control de los procesos que se empeñan en dejar el sistema bloqueado hace al uso del intérprete de comandos prácticamente imprescindible.
- Etc.

#### ¿Y qué sucede si nuestro sistema operativo no es Linux?

En este caso, podemos crear un entorno que ejecute Linux dentro del SO que nuestro ordenador esté usando. Este entorno es conocido como **Máquina Virtual**.

Una máquina virtual nos permite abstraer el hardware de un ordenador (es decir, su procesador, memoria, disco duro, etc) entre diferentes entornos, creando la ilusión de que cada uno de ellos por separado se está ejecutando en su propio ordenador. Cada uno de estos entornos virtuales ejecutando su propio SO encima del SO nativo de nuestro ordenador puede ser visto como una *computadora virtual*.

El uso de máquinas virtuales tiene varias ventajas:

- Poder usar un SO concreto sin tener que instalarlo de forma completa o mediante particiones en nuestro ordenador.
- Poder utilizar varios sistemas operativos de forma simultánea.
- Tener disponible de forma inmediata un SO pre-configurado con todo el software y herramientas que necesitamos sin necesidad de realizar todos los pasos de instalación por nosotros mismos.

Cada computadora virtual tendrá su propios recursos asignados (CPU, almacenamiento, RAM, etc.) y limitaciones, de forma que todas las máquinas virtuales de un mismo ordenador (conocido como *host*) puedan funcionar de forma segura y eficiente. El encargado de crear, gestionar y proteger las máquinas virtuales de un mismo ordenador es un software conocido como *Gestor de Máquinas Virtuales* o *hypervisor* (ver Figura I.2).





Figura I.2: Arquitectura de sistema operativo con virtualización

### B1.2. Boletín 0: Máquina Virtual de Linux

#### B1.2.1. Objetivos

El objetivo de este boletín inicial de prácticas es el de familiarizar al alumno con el hypervisor **VirtualBox** a través de su instalación, y la importación de una máquina virtual de Linux a través de un archivo OVA. Este boletín se realizará en el ordenador personal del alumno, como parte de su trabajo autónomo en casa. Un archivo OVA (Open Virtualization Appliance)contienen una máquina virtual comprimida que es directamente instalable en un hypervisor. Cuando se importa un fichero OVA, se extrae la máquina virtual con su SO y todo el software ya instalado y configurado en él. También puedes optar por instalar una distribución de Ubuntu e instalar por tí mismo el software necesario para los ejercicios de prácticas. Para hacer este tipo de instalación, sigue el anexo al final de este documento.

#### B1.2.2. Plan de trabajo

El plan de trabajo de esta sesión será el siguiente:

- 1. Lectura del boletín por parte del alumno.
- 2. Realización de forma individual en casa de los ejercicios propuestos en el boletín.

#### B1.2.3. Ejercicios a realizar durante la sesión

- 1. **Descarga el archivo OVA preparado para la asignatura.** Descarga el archivo OVA disponible en el siguiente enlace: Ubuntu 22.04 (LTS). Este archivo tiene un tamaño de más de 7 GB, por lo que la descarga tardará unos minutos. Mientras se descarga el archivo, pasa al siguiente punto para instalar VirtualBox.
- 2. Descarga e instala VirtualBox. Descargaremos la última versión de este hipervisor desde su página Web *https://www.virtualbox.org/wiki/Downloads*.

Durante la instalación, sigue las recomendaciones por defecto seleccionando "Next". Antes de proceeder a instalar, probablemente el instalador te avisará de que tu conexión a Internet se va a desconectar temporalmente. Procede con la instalación de todas formas.

3. **Importa la máquina virtual comprimida en el archivo OVA que has descargado.** Una vez que has instalado e iniciado VirtualBox, vas a crear una máquina virtual con una distribución de Ubuntu. Pero no vas a crearla





Figura I.3: Página Web de VirtualBox

desde cero, lo cual supondría tener que instalar todo el software necesario para las prácticas de esta asignatura. En vez, vas a importar una máquina virtual ya configurada y con todo el software necesario. Para ello, en el menu de VirtualBox, selecciona "Archivo" y haz click en "Importar servicio virtualizado...". Se abrirá un diálogo en el que debemos indicar el archivo OVA que nos hemos descargado. Para ello, haz click en el icono de carpeta de la derecha y selecciona el archivo en tu sistema de ficheros (Figura I.4a). Otro diálogo listará la configuración del sistema virtual que estamos importando. No modifiques esta configuración y haz click en "Importar" (Figura I.4b).

Oracle VM VirtualBox Administrador	
§rchivo Máquina Ayuda	
r X	? ×
— Importar servicio virtualizado	- Importar servicio virtualizado
Servicio a importar	- Preferencias de servicio
Please choose the source to import appliance from. This can be a local file system to import OVF archive or one of known cloud service providers to import cloud VM from.	Estas son las máquinas virtuales contenidas en el servicio y las preferencias sugeridas de las máquinas virtuales importadas de VirtualBox. Puede cambiar varias de las propedades mostradas haciendo doble cir en los elemenos y deshabilitar otras usando las casilias de verticación de abajo.
Fuente: Sistema de archivos local	Nit Sintama virtual 1
Seleccione un archivo desde el que importar el servicio virtualizado. Virtualillox actualmente soporta importar servicios guardados en Open Virtualization Format (OVF). Para continuar, seleccione el archivo a importar abajo	Nombre Ubuntu 22.04 LTS (FC)
Archivo:  sers\/vidoria\Documents\/UPCT\ASIGNATURAS\GRADO DE DATOS UMU-UPCT\FUNAMENTOS DE COMPUTADORES-GCD\/Uburtu 22.04 LTS (FC).ova	
	RAM 1024 MB
	OVD 🗹
	Controlador USB
	ine in Tarjeta de sonido in importando servicio virtualizado: importing appliance C:\Users\victo 🚿
	Adaptador de red
	Controlador de aln
	Controlador de alm
	Controlador de ain
	Imagen de disco maain ooranie ceur co co oranie oranie international de disco maain and a construction of the disco orange of the construction of the disco orange of the disco of the
	Grapeta base CLUSers/victoria/virtualbox VVis
	a Grupo primano 7
	Carpeta base de máquina: 🔋 C:\Users\Victoria\VirtualBox VMs 🗸 🗸
	Politica de dirección MAC: Incluir solo las direcciones NAT de adaptador de red 🔹
	Opciones adicionales:
	Servicio virtualizado no firmado
Modo experto Next Cancelar	
	Restaurar valores predeterminados Importar Cancelar

(a) Importar un archivo OVA

(b) Instalación de un archivo OVA

Figura I.4: Importación de máquina virtual en VirtualBox

Ahora que hemos importado nuestra máquina virtual, podemos ver todas las opciones que hemos configurado (Figura I.15). Ya tienes la máquina virtual de Ubuntu completamente funcional y con todo el software necesario para los ejercicios de prácticas.

4. **Cierra tu máquina virtual.** Para apagar tu máquina virtual de Ubuntu, en la barra superior de ésta, selecciona *Archivo/Cerrar*. Aparecerá la ventana mostrada en la Figura I.23. Aquí puedes seleccionar 'Guardar el estado de la máquina' si deseas, al volver a iniciarla, continuar justo en el punto donde se cerró. En este caso, todo lo que has estado haciendo se pausará hasta que vuelvas (no se pierde ningún tipo de información), pero la máquina virtual no consumirá recursos. Si quieres apagar la máquina totalmente, lo que equivaldría a pulsar





Oracle VM VirtualBox Administrador Archivo Máquina Ayuda		- o ×
Herramientas	Image: Configuration     Image: Configuration     Image: Configuration       News     Configuration     Descritzring	
Ubuntu 22.04 LTS (rC)	General Hombie: Ubuntu 22.04 LTS (FC) Statema operative: Ubuntu ((x+0+t))	Previsualización
	Instrume         Instrume           Memorita bases         1024 MB           Orders da errangues         Despetis, Disco duro           Acaleración:         VirzyAMO-V, Paginación anidota, Parsvirtualización KVM	
	Pantala Memoin de video: 16 MB Cantrolador gréfico: VMSVGA Servidor de escitorio remoto: Inhabilitado Crabación: Inhabilitado	
	Advacenamiento     Controlotar, III,     Berginaria enpetato:     Descinaria enpetato:	
	Audio Controldor de anfitririe: Windows DirectSound Controldor : ICH ACS7	
	Red Adaptador 1: Intel PRD/1000 MT Desktop (IKT)	
	Image: Controlled US:         0HCI           Filtres de diagositive:         0 0 active)	
	Carpetas compartidas Ninguno	
	Bescripckin Ninguro	

Figura I.5: Máquina virtual creada

el botón de apagado/encendido de una máquina física, selecciona 'Enviar señal de apagado'. No selecciones 'Apagar la máquina', puesto que esto es un apagado abrupto, equivalente a tirar del cable de alimentación, por lo que el SO no se cerrará adecuadamente y podrían haber perdidas de información.





### B1.3. Boletín 1: Primer contacto con el shell

#### B1.3.1. Objetivos

El objetivo de este boletín de prácticas es introducir a los alumnos en el uso elemental del SO Linux, haciendo especial hincapié en el manejo de éste desde el intérprete de comandos o *shell*. Se introducirán una serie de comandos básicos para dar los primeros pasos en este SO. Más concretamente, se tratarán los siguientes temas:

- Acceso al sistema.
- Interfaz gráfico. Lanzamiento de programas.
- Intérprete de comandos o *shell*. Formato general de una orden, comandos básicos.

De aquí en adelante, utilizaremos la máquina virtual de Ubuntu que está preconfigurada en VirtualBox en los ordenadores del laboratorio de prácticas. Esta máquina tiene un usuario por defecto cuyo nombre es alumno y su contraseña es la misma que su nombre de usuario, es decir, alumno.

#### B1.3.2. Plan de trabajo

El plan de trabajo de esta sesión será el siguiente:

- 1. Lectura del boletín por parte del alumno.
- 2. Realización de forma individual en clase de los ejercicios propuestos en el boletín.

#### B1.3.3. El prompt

El terminal del *shell* permite al usuario introducir comandos para utilizar los servicios del SO. La línea de comandos está indicada por lo que se conoce como *prompt*, que no es más que el conjunto de caracteres que muestra el intérprete para anunciar que se encuentra a la espera una orden o comando del usuario, que será ejecutado a continuación. El *shell* suele mostrar un *prompt* como el siguiente:

usuario@nombre\_máquina:directorio\_actual\$

donde

- usuario es el nombre del usuario conectado al sistema que hace uso del *shell*.
- nombre\_máquina es el nombre del computador que se está usando.
- directorio\_actual es el directorio de trabajo en el que se encuentra actualmente el shell.

Como vemos, el *prompt* nos indica en todo momento el directorio de trabajo donde estamos situados, que puede verse como el lugar de partida por defecto a la hora de ejecutar el siguiente comando, tal y como veremos a continuación.

#### B1.3.4. El directorio de trabajo actual

Un concepto importante que es necesario comprender desde el comienzo es que el shell tiene una noción de una ubicación por defecto en la que tendrá lugar cualquier operación con ficheros: el **directorio de trabajo actual**. Si intentas crear nuevos ficheros o directorios, ver ficheros existentes o incluso borrarlos, el shell asumirá que los estás buscando en el directorio de trabajo actual a menos que especifiques lo contrario. Por lo tanto, es muy importante tener siempre presente en qué directorio se encuentra el shell en cada momento; después de todo, borrar ficheros del directorio equivocado podría ser desastroso. Si alguna vez tienes dudas, el comando pwd te dirá exactamente cuál es el directorio de trabajo actual del shell (*pwd* es la abreviatura de "print working directory", imprimir el directorio de



trabajo). No obstante, lo normal es que el *prompt* del shell muestre el directorio de trabajo actual, por lo que una vez te acostumbres a tener presente dónde te encuentras en cada momento, raramente necesitarás ejecutar dicho comando.

El comando que utilizamos para movernos de un directorio a otro (es decir, para cambiar el directorio de trabajo actual), es  $\boxed{cd}$ :

- cd ruta\_directorio : Nos mueve de nuestro directorio actual al indicado por la ruta dada.
- cd (sin parámetros), o bien cd ~ : Nos mueve al directorio de inicio del usuario, /home/usuario).
- cd : Nos mueve al directorio último en el que habíamos estado con anterioridad al actual.

#### B1.3.5. El directorio raíz

Para Linux todo el conjunto de dispositivos de almacenamiento es considerado como un único sistema de ficheros jerárquico que parte desde el directorio raíz (/). No existen, por tanto, las unidades lógicas (C:, etc.) que usa Windows. Linux organiza el espacio en disco en subdirectorios con permisos y privilegios propios de cada usuario. Los ficheros y directorios se organizan jerárquicamente a partir de su directorio raíz (/) con una estructura predeterminada que se muestra en la Figura I.6, que merece la pena recordar y respetar para encontrar aplicaciones, librerías, archivos de configuración, datos de usuario y demás tipos de archivos con rapidez y eficacia.



Figura I.6: Jerarquía de directorios en Linux.

#### B1.3.6. El directorio de usuario

En Linux, como en otros SOs multiusuario, cada usuario posee un **directorio de usuario**, a veces también llamado **directorio personal** o **directorio de inicio**, en el que poder almacenar sus propios ficheros. Lo habitual es que, dentro del directorio /home, cada usuario tenga un subdirectorio llamado como el propio nombre de usuario (p.ej., el usuario alumno tendrá su directorio personal en /home/alumno. Para referirnos al directorio /home (leído "barra home"), que alberga los datos de todos los usuarios. El carácter ~ es sinónimo de la ruta abosluta a *nuestro home* o directorio personal, de forma que siempre que estemos dentro del mismo, el *prompt* mostrará como directorio de trabajo actual una ruta que empieza por ~.



#### **B1.3.7.** Rutas absolutas y relativas

La ruta a los directorios o los ficheros que los comandos necesitan se puede indicar por medio de su <u>ruta absoluta</u> (cualquiera de las que parten del directorio raíz) o por medio de una <u>ruta relativa</u> (cualquiera de las que parten del directorio de trabajo actual). Una ruta absoluta comienza por lo general con la barra inclinada (carácter '/'), aunque las rutas que comienzan por '~' son también absolutas (ya que ~ es en realidad una abreviatura de /home/usuario). Por contra, las rutas relativas suelen comenzar por el nombre del fichero/directorio dentro del directorio de trabajo actual al que nos referimos. En el caso de que necesitemos referirnos mediante una ruta relativa a un fichero/directorio que no esté contenido en el directorio de trabajo actual (ni directamente ni tampoco en ninguno de sus subdirectorios), es posible referirnos al directorio padre a uno dado mediante un par de puntos seguidos de la barra (caracteres . . /), lo cual se puede repetir cuantas veces sea necesario para *subir* niveles en el árbol de directorios. Una ruta relativa también puede comenzar con el punto y la barra (caracteres . . /) para hacer referencia al directorio actual, aunque esto resulta innecesario por estar implícito, salvo en casos muy concretos que veremos en posteriores boletines de prácticas. He aquí algunos ejemplos:

Ruta absoluta para cambiar al directorio /usr/lib:

alumno@pc:~\$ cd /usr/lib

• Ruta relativa desde el directorio /usr para listar el fichero sudo\_plugin.h que está dentro del subdirectorio include

alumno@pc:/usr\$ ls include/sudo\_plugin.h

o lo que es equivalente

alumno@pc:/usr\$ ls ./include/sudo\_plugin.h

• Ruta absoluta para listar el fichero sudo\_plugin.h del ejemplo anterior:

```
alumno@pc:~$ ls /usr/include/sudo_plugin.h
```

• Ruta relativa desde el directorio /usr/include para cambiar al directorio /usr/local/etc

alumno@pc:/usr/include\$ cd ../local/etc

Por ejemplo, supongamos que el usuario alumno está en /usr/local/etc y quiere ir a /home/alumno. El usuario, podrá cambiar a este directorio usando una ruta relativa o una absoluta:

- Usando una ruta relativa: alumno@pc:/usr/local/etc\$ cd ../../home/alumno
- Usando una ruta absoluta: alumno@pc:/usr/local/etc\$ cd /home/alumno

#### **B1.3.8.** Sintaxis general de un comando

La gran mayoría de comandos del shell sigue una sintaxis similar, que puede resumirse en tres partes claramente diferenciadas: orden, opciones y parámetros. El nombre de la orden o comando es el único campo obligatorio, mientras que las opciones y los parámetros son opcionales:

```
user@host:~$ comando [opciones] [parámetros]
```



- Comando. Los comandos introducidos pueden ser *comandos internas*, que son aquellas propias del programa *shell* (p.ej., exit para terminar la sesión, o cd para cambiar de directorio), o bien pueden ser *comandos externas* en las que el *shell* actúa como pasarela para lanzar a ejecución otros programas (ficheros ejecutables) que hayan sido previamente instalados y estén disponibles en el sistema, tales como ls, rm o firefox. El intérprete de comandos buscará estos ficheros ejecutables en ciertos directorios por defecto. Cualquier aplicación instalada en el sistema, sin importar si hace uso de la CLI o la GUI, se puede ejecutar desde la línea de comandos.
- Opciones. Las opciones sirven para modificar el comportamiento del comando. Las opciones siempre han de ir a continuación del nombre del comando, antes de parámetros. Pueden ponerse en cualquier orden y pueden agruparse, pero siempre antes de los parámetros de la orden. En la mayoría de comandos, con contadas excepciones, las opciones se suelen indicar bien mediante un guión y una letra (p.ej. -r), bien mediante doble guión y una palabra descriptiva (p.ej. -recursive). Es común encontrar comandos que aceptan ambas formas, abreviada y verbosa, para cada opción.

ls -a -l -h = ls -lha = ls -l --all --human-readable

Parámetros. Van al final de la orden, tras las opciones si las hubiese. Un gran número de comandos de manipulación de ficheros admite un número variable de parámetros, arbitrariamente grande (p.ej., cp o rm). En los comandos de manipulación del sistema de ficheros, las parámetros son rutas a ficheros o directorios. En caso de comandos que no requieren obligatoriamente un número mínimo de parámetros, suele tomarse el directorio de trabajo actual como argumento implícito (p.ej., si a ls no se le pasa ningún parámetro, mostrará el contenido del directorio de trabajo actual).

#### **B1.3.9.** Autocompletar rutas mediante la tecla tabulador

Una de las características más interesantes de cualquier *shell* moderno es la capacidad de autocompletar mediante la pulsación tecla del tabulador, [Tab]. Una vez acostumbrados a utilizar esta funcionalidad clave del shell, veremos dramáticamente incrementada nuestra productividad al ahorrar miles de pulsaciones de teclado diariamente. En función de la posición en que nos encontremos dentro de la estructura general de un comando (nombre del comando vs. argumentos), la pulsación del tabulador tratará de autocompletar bien el nombre del comando (por ejemplo, buscando entre los programas instalados disponibles) o bien la ruta a los ficheros/directorios que buscamos pasar como parámetros al comando. Así, al presionar [Tab] el shell completará la ruta cuando hayamos escrito suficientes caracteres de ésta como para poder distinguir una ruta de otras que comiencen por los mismos caracteres. En caso de ambigüedad, no será posible autocompletar por completo el nombre del fichero/directorio, sino sólo los caracteres que constituyen el prefijo común. Presionando dos veces seguidas, el shell mostrará las posibles rutas cuyo nombre empiece por los caracteres que ya hemos tecleado, de forma que sólo tendremos que teclear los caracteres adicionales para eliminar la ambigüedad y después volver a presionar [Tab] para autocompletar el resto del nombre.

#### B1.3.10. Comandos básicos de manejo de ficheros

El terminal de Linux nos ofrece muchas posibilidades de listado y manejo de ficheros y directorios de un modo más personalizado que un interfaz gráfico. Por ahora, vamos a aprender los comandos más básicos:

- 1s : Lista el contenido del directorio actual.
- touch FICHERO: Crea un nuevo fichero vacío en la ruta indicada, en caso de que no exista. En otro caso, le cambia la fecha y hora de modificación a la actual sin modificar su contenido.
- Cat FICHERO ... : Muestra el contenido de los ficheros indicados.
- rm FICHERO ... : Borra los ficheros indicados. También puede utilizarse para borrar directorios, para lo cual se usa la opción -r.





- mkdir DIRECTORIO : Crea un directorio vacío en la ruta designada.
- rmdir DIRECTORIO: Borra el directorio indicado, siempre que esté vacío. Si no, hay que borrar primero todas sus entradas, o bien simplemente usar el comando rm con la opción -r.

Para finalizar, dejamos aquí un listado misceláneo de otros comandos de interés:

- exit : Termina la sesión del usuario actual.
- date : Devuelve la fecha y la hora actuales.
- Clear : Limpia el terminal.
- reset : Resetea el terminal (para cuando queda "tonto", con caracteres extraños; por ejemplo, tras hacer cat de un fichero binario).

#### B1.3.11. El comando script

El comando script se utiliza para grabar toda la actividad de la terminal, lo cual resulta idóneo para estudiantes que necesitan un registro de una sesión interactiva como prueba de una tarea. Después de ejecutar el comando, script comienza a grabar todo lo impreso en la pantalla, incluyendo los comandos tecleados y las salidas mostradas, hasta que se termina la sesión mediante el comando exit. Por defecto, si no se le pasa ningún argumento al invocarlo, toda la información de la terminal se guardará en un fichero llamado typescript, en el directorio actual. Es posible ir registrando sucesivas sesiones con script en un mismo fichero, para lo cual se le pasa la opción –a, y el fichero, en su caso. En caso de ejecutar el comando sin ningún argumento, debemos tener cuidado para no sobreescribir accidentalmente una sesión previamente grabada. Finalmente, podemos mostrar el contenido del fichero generado por script con el comando less –r typescript.

En esta asignatura, los alumnos utilizarán script para registrar su trabajo autónomo en el proceso de aprendizaje de la interfaz de línea de comandos de Linux. Tanto el trabajo en el laboratorio durante las sesiones de prácticas como los ejercicios realizados en casa se deben registrar mediante script. Cada alumno deberá añadir los ficheros generados con script a un repositorio git personal, cuyo propósito es servir como bitácora para el seguimiento de la asignatura; en la próxima sesión de prácticas veremos, además del uso básico de la herramienta git, cómo crear dicho repositorio en la nube y compartirlo con el profesorado. La bitácora de cada alumno será la herramienta utilizada para la supervisión del proceso de aprendizaje y como instrumento para la evaluación continua.

#### B1.3.12. Ejercicios a realizar durante la sesión

1. Accede a la máquina virtual de Ubuntu pre-configurada. Abre la aplicación *VirtualBox* en tu ordenador del laboratorio de prácticas e inicia la máquina virtual de Ubuntu que ya está configurada en esta aplicación. Utiliza el usuario por defecto de esta computadora virtual. Para ello, introduce el nombre de usuario alumno con contraseña alumno.

Una vez iniciada sesión, te encuentras en el interfaz gráfico de Ubuntu, **GNOME**, el cuál es un entorno de escritorio gratuito y de código abierto para sistemas operativos Linux.

Vamos a abrir una terminal del *shell*, para lo cual puedes usar la combinación de teclas Ctrl + Alt + T. También, puedes abrir una terminal, buscando el programa *"Terminal"* en el menú de aplicaciones (Figura I.7). Una vez lanzado, desde esta ventana de terminal gráfico iremos ejecutando el resto de comandos del boletín.

#### 2. Primer contacto con el terminal de texto.

Prueba a teclear los comandos a continuación. A modo de prueba, vamos a utilizar el comando script para registrar los comandos que practicamos.

a) script, para comenzar a registrar tu actividad con el terminal.





(a) Búsqueda del programa Terminal



(b) Ventana que emula un terminal de línea de comandos desde el escritorio gráfico

#### Figura I.7: Consola del shell en Ubuntu

- b) ### EJERCICIO 2 ###, no es un comando sino un simple comentario para documentar en nuestro fichero typescript el ejercicio del boletín en el que nos encontramos (el shell interpreta el carácter especial almohadilla como delimitador de comentarios e ignora todo lo que viene tras él). Por convenio, utilizaremos tres almohadillas consecutivas para indicar el comienzo de un nuevo ejercicio.
- c) date, para mostrar la fecha.
- d) whoami, para mostrar tu nombre de usuario.
- e) 1s, para mostrar el contenido del directorio de trabajo actual.
- f) clear, para borrar el terminal.
- g) exit, para cerrar el shell lanzado por el comando script del apartado a.
- h) exit una vez más, para cerrar el terminal.
- *i*) Vuelve a lanzar el terminal gráfico con Ctrl+Alt+T.
- *j*) less -r typescript para mostrar la actividad del terminal registrada con script.
- *k*) script –a para continuar registrando tu actividad con el terminal en el mismo fichero.
- 3. Lanzamiento de programas. Existen varias formas de lanzar comandos en Linux: a través del menú o icono correspondiente (al estilo clásico de Windows), llamando al comando correspondiente desde el terminal, o usando el menú de lanzamiento rápido (Alt-F2). Como ejercicio, lanza la calculadora de GNOME (gnome-calculator) usando estos tres métodos. También podemos lanzar por ejemplo el famoso programa firefox para navegar por la web. Antes de comenzar, incluye el comentario ### EJERCICIO 3 ### para documentar tu bitácora. Prueba a mostrar la ayuda del programa gnome-calculator pasándole la opción --help, y luego modifica su funcionamiento lanzándolo en diferentes modos (--mode=xxx).
- 4. **Directorio de trabajo actual.** Como hemos visto, para cambiar el directorio de trabajo se utiliza el comando cd, una abreviatura de "cambiar de directorio". Recuerda escribir ### EJERCICIO 4 ### antes de reproducir los siguientes pasos.
  - *a*) Prueba a escribir lo siguiente en un terminal:

cd / pwd

Comprueba con pwd que tu directorio de trabajo es '/', es decir, el directorio raíz del sistema de ficheros.







(a) Desde el icono del programa

(b) Desde la terminal con comando



(c) Desde el lanzamiento rápido

Figura I.8: Tres métodos para abrir un programa en Linux

*b*) Desde el directorio raíz, el siguiente comando te moverá al directorio "home", que es un subdirectorio inmediato de '/':

```
cd home
pwd
```

c) Para subir al directorio padre del directorio de trabajo actual, en este caso de vuelta a "/", se utiliza la sintaxis especial de dos puntos (..). No olvides el espacio entre "cd" y "..", respetando siempre la estructura general de un comando (comando + opciones + parámetros, separados por espacios).

cd .. pwd

d) Recuerda que escribir cd por sí solo es un atajo rápido para volver a tu directorio de usuario:

```
cd
pwd
```

e) También puedes usar .. más de una vez si tienes que subir por varios niveles de directorios padre:

```
cd ../..
pwd
```

f) La ruta .../.. significa "partiendo del directorio de trabajo actual, muévete al directorio padre, y desde esa nueva ubicación muévete a su vez a su padre". Así que, si quisiéramos ir directamente desde nuestro directorio de usuario hasta el directorio etc que está en el directorio raíz, podríamos utilizar esta ruta:





```
cd
pwd
cd ../../etc
pwd
```

- 5. **Rutas relativas y absolutas.** La mayoría de los ejemplos que hemos visto hasta ahora utilizan rutas relativas. Es decir, el lugar en el que terminas depende de tu directorio de trabajo actual.
  - a) Considera la posibilidad de intentar entrar en la carpeta etc. Si ya estás en el directorio raíz, esto funcionará bien:

```
cd /
pwd
cd etc
pwd
¿Pero qué pasa si estás en tu directorio de usuario?
```

```
cd
pwd
cd etc
pwd
```

Verás un error diciendo No existe el archivo o el directorio antes de que puedas ejecutar el último pwd. El cambio de directorio especificando el nombre del directorio, o usando .., tendrá diferentes efectos dependiendo del lugar desde el que empiece, es decir, la ruta sólo tiene sentido en relación con el directorio de trabajo en que te encuentres en ese momento.

b) Hemos visto dos comandos que son absolutos (tienen el mismo efecto sin importar cuál sea el directorio de trabajo actual). El primero es cuando ejecutas cd sin argumentos para ir directamente a tu directorio de inicio de usuario. El segundo es cuando usas cd / para cambiar al directorio raíz. De hecho, cualquier ruta que comience con una barra '/' es una **ruta absoluta**. Una manera de ver esto es como si dijeras "cambia al directorio raíz, luego sigue la ruta desde allí". Esto nos da una manera mucho más fácil de cambiar al directorio etc, sin importar en qué parte del sistema de ficheros nos encontremos:

```
cd
pwd
cd /etc
pwd
```

c) También nos da otra forma de volver a tu directorio de inicio, e incluso a las carpetas dentro de él. Supongamos que quieres ir directamente a tu carpeta "Escritorio" desde cualquier lugar del disco (fíjate en la "E" mayúscula). En el siguiente comando tendrás que sustituir alumno por tu propio nombre de usuario. El comando whoami te recordará tu nombre de usuario, en caso de que no estés seguro:

```
whoami
cd /home/alumno/Escritorio
pwd
```

*d*) Hay otro atajo útil que funciona como una ruta absoluta. Como has visto, usar '/' al principio de tu ruta significa "empezar desde el directorio raíz". Utilizar el carácter de tilde (~) al principio de la ruta significa igualmente "empezar desde mi directorio de usuario".





```
cd ~
pwd
cd ~/Escritorio
pwd
```

e) Ahora ese texto extraño en el prompt puede tener un poco de sentido. ¿Has notado que cambia a medida que te mueves por el sistema de ficheros? En la mayoría de sistemas Linux muestra tu nombre de usuario, el nombre de red de tu ordenador y el directorio de trabajo actual. Pero si estás en algún lugar dentro de tu directorio personal, utilizará ~) como abreviatura. Vamos a pasearnos un poco por el sistema de ficheros, fijándote en el prompt mientras lo haces:

```
cd
cd /
cd ~/Escritorio
cd /etc
cd /var/log
cd ..
cd
```

f) En realidad, las rutas relativas funcionan porque todos los directorios del sistema de ficheros siempre contienen dos subdirectorios *especiales*, llamados '.' y '..' que hacen referencia, respectivamente, al propio directorio y a su directorio padre. De esta forma, las rutas que mostramos en los siguientes ejemplos son igualmente correctas, y aunque en este caso puedan carecer de utilidad práctica, se muestran para clarificar los mencionados conceptos de directorio actual y directorio padre:

cd /etc/	#	Ruta absoluta al directorio raíz, padre de etc
pwd	#	Estamos en el raíz (/)
cd ./././././	#	Ruta relativa, referencia repetida al directorio actual
pwd	#	Seguimos en el mismo directorio
cd etc	#	Ruta relativa a etc desde el directorio actual (raíz)
cd /	#	Ruta absoluta al directorio raíz
cd ./etc	#	Ruta relativa a etc desde el directorio actual. Nótese
	#	que teclear ./ es innecesario (equivale a cd etc)
cd /etc	#	Ruta absoluta a etc
cd /etc//home	e/.	/usr # Ruta absoluta al directorio usr, hijo del raíz
		<pre># y hermano de etc y home (equivale a cd /usr)</pre>

- 6. Uso del tabulador para autocompletar rutas. Es importante acostumbrarse desde el primer día a usar el tabulador para navegar por el sistema de ficheros, de forma que podamos autocompletar rutas al mismo tiempo que tenemos una noción exacta de en qué punto del sistema de ficheros nos encontramos.
  - a) Situado en el directorio ~/Escritorio, prueba a navegar por el sistema de ficheros presionando dos veces el tabulador para mostrar el contenido de cada directorio de nivel superior en la ruta relativa hacia el directorio /etc. Para ello, sin teclear ningún carácter tras cada separador '/', observa cómo el shell muestra las alternativas disponibles para autocompletar: , cd [tab][tab]../[tab][tab]../[tab][tab]../etc
  - *b*) Experimenta con la compleción de rutas con el tabulador tecleando, por ejemplo, cd /us[tab]/bi[tab]/X[tab], etc.





- *c*) Experimenta tecleando las rutas *extravagantes* mostradas como ejemplo en el último apartado del ejercicio anterior, pero esta vez pulsando el tabulador dos veces para tratar de autocompletar en cada nivel: cd /e[tab]/../[tab][tab]home/[tab][tab]../[tab]usr,
- 7. Movimiento básico por el sistema. Desde el terminal abierto en el apartado anterior, realiza los siguientes ejercicios:
  - a) Muestra la ruta absoluta del directorio en el que te encuentras con el comando pwd.
  - b) Cámbiate al subdirectorio bin bajo el directorio /usr que está en la raíz, usando el comando cd con una *ruta absoluta*. Comprueba de nuevo cuál es tu directorio de trabajo actual (pwd) para ver que el comando se ha ejecutado sin errores.
  - c) Muestra los ficheros en este directorio mediante ls.
  - *d*) Compara los contenidos de este directorio con los contenidos mostrados por el explorador de ficheros gráfico (Figura I.9). Para ello, haz click en el icono de carpeta de la barra de herramientas lateral izquierda para abrir el explorador de ficheros. Selecciona *Otras localizaciones* en el panel izquierdo de esta ventana y primero selecciona la carpeta *usr*, y dentro de ésta, la carpeta *bin*, de tu ordenador.



Figura I.9: Listado de ficheros en /usr/bin

- *e*) Vuelve al directorio de inicio de sesión (directorio *home*) usando el comando cd sin parámetros. Comprueba en todo momento tu situación en la jerarquía de directorios usando pwd.
- f) Cámbiate, con un sólo comando cd [ruta], al directorio /etc/X11, pero usando esta vez una *ruta relativa* (que NO comience con una /).
- g) Vuelve a tu directorio home, con varios comandos cd, cada uno indicando como parámetro un único directorio relativo, es decir, la ruta del comando cd no puede contener el carácter /. Desde el directorio /etc/X11 a tu directorio home necesitarás ejecutar 4 comandos cd.

#### 8. Creación y borrado de ficheros y directorios.

*a*) En tu directorio de trabajo ejecuta el comando ls. A continuación, mediante el ratón, selecciona con el botón izquierdo la salida del comando ls y pégala en la ventana de un editor de texto inicialmente vacío usando el botón izquierdo del ratón o Ctrl+V (Figura I.10). Guarda el fichero generado en un fichero llamado salida.txt en tu directorio actual. Para abrir el editor de texto, puedes buscar la aplicación en el gestor de aplicaciones de Ubuntu. Para ello, haz click en el icono de la esquina inferior izquierda y escribe *edit* en el buscador.







Figura I.10: Copia y pegado en el editor de texto

- b) Ejecuta de nuevo el comando ls para comprobar que tu fichero creado es mostrado en el terminal.
- c) Crea en tu directorio personal el directorio pracFC (usando mkdir)
- d) Muévete al directorio creado con cd y la ruta relativa de este directorio.
- e) Crea un fichero que se llame *prueba.txt* en tu directorio actual, pracFC, mediante el comando touch.
- *f*) Desde el explorador de ficheros gráfico, abre el fichero *prueba.txt*, escribe "*Mi primer fichero creado con touch*" y guárdalo.
- g) Vuelve a tu directorio home con una ruta relativa.
- h) Mediante el comando cat , muestra el contenido del fichero *prueba.txt* que acabas de editar, utilizando un ruta relativa.
- *i*) Intenta borrar el directorio pracFC usando el comando rm. ¿Puedes? Si no, usa el comando adecuado para hacer el borrado (utiliza *rmdir* u otra posible opción es añadir el parámetro -*r* al comando rm).

#### B1.3.13. Referencias

Traducción adaptada del tutorial: https://ubuntu.com/tutorials/command-line-for-beginners



### B1.4. Boletín 2: Gestión de paquetes en Linux

#### B1.4.1. Objetivos

El objetivo de este boletín de prácticas es familiarizar al estudiante con la gestión de paquetes en línea de comandos a través de la herramienta APT, así como de instalar varias herramientas necesarias para los sucesivos boletines de prácticas de la asignatura.

#### B1.4.2. Gestión de paquetes en Linux

La gestión de actualizaciones de software ya existente en el sistema operativo (SO) así como de instalaciones de nuevo software en una tarea fundamental en cualquier SO. Las nuevas actualizaciones pueden incluir versiones más recientes de software, nuevas características, correcciones de seguridad o mejoras de calidad del SO. La instalación de nuevo software nos permite añadir a nuestro ordenador las aplicaciones y funcionalidades que no vienen por defecto en el SO y necesitamos para alguna tarea.

En Ubuntu (y Linux en general), no es necesario buscar el programa por Internet, descargarlo e instalarlo manualmente. Para eso existen los repositorios, una especie de almacén centralizado que contiene todo el software disponible y que siempre está actualizado. Nosotros tan sólo deberemos elegir la aplicación y el sistema ya se encarga de descargarla e instalarla.

Ubuntu proporciona aplicaciones para poder gestionar la instalación y actualizaciones de software a través de la interfaz de usuario gráfica (*Software y Actualizaciones, Ubuntu Software*, etc.). Sin embargo, es conveniente saber manejar la herramienta en línea de comandos para la gestión de paquetes, llamada APT (*Advanced Packaging Tool*). APT es una forma sencilla, cómoda y rápida de gestionar los paquetes instalados en Ubuntu y de instalar nuevos paquetes. A continuación, listamos los comandos fundamentales de apt :

- apt update : Actualiza la base de datos local de paquetes disponibles, que en realidad es un índice de todos los paquetes instalables desde *apt*. Nótese que esto no lleva a cabo ninguna actualización en el software que tenemos instalado.
- apt list upgradable : Muestra una lista de los paquetes actualizables.
- apt upgrade : Actualiza los paquetes actualizables. Para ello, compara las versión instalada de cada paquete con la versión disponible del paquete en la base de datos local de paquetes. Si la versión instalada es menor, el comando nos pregunta si actualizar el paquete a la versión más reciente disponible.
- apt list nombre-paquete : Busca el paquete con nombre *nombre-paquete* en los repositorios disponibles, o entre los paquetes instalados. Es posible usar comodines tales como \* en el nombre del paquete a la hora de buscar. Con la opción --installed busca únicamente entre los paquetes instalados.
- apt install nombre-paquete : Instala el paquete con nombre *nombre-paquete* desde los repositorios disponibles. Una vez ejecutemos el comando es posible que el sistema nos pregunte si deseamos instalar el paquete y sus dependencias, mostrándonos ciertos detalles del mismo como su nombre completo, versión o tamaño.

A la hora de utilizar este último comando, debemos tener en cuenta que necesitamos poseer privilegios de superusuario, así que necesitamos anteponer el comando sudo para indicar que queremos ejecutar el comando subsiguiente con privilegios de administrador. Por ejemplo para actualizar APT ejecutaremos el comando sudo apt update, y *sudo* nos pedirá nuestra contraseña para verificar nuestra identidad antes de otorgarnos privilegios de superusuario.



#### **B1.4.3.** Plan de trabajo

El plan de trabajo de esta sesión será el siguiente:

- 1. Lectura del boletín por parte del alumno.
- 2. Realización en clase de los ejercicios propuestos en el boletín.

#### B1.4.4. Ejercicios a realizar durante la sesión

- 1. Actualiza APT. Inicia una consola de línea de comandos y ejecuta *apt update* para actualizar el índice de paquetes.
- 2. Actualiza los paquetes instalados en el sistema. Ejecuta el comando *apt upgrade* para actualizar los paquetes instalados en tu SO Ubuntu.
- 3. Busca el paquete git. Utiliza el comando *apt list git* para buscar el paquete *git*.
- 4. **Comprueba la versión de** git **instalada.** Utiliza el comando *git –version* para comprobar la versión de git que ha sido instalada.
- 5. Comprueba si tienes instalados otros paquetes. Comprueba si están instalados otros paquetes tales como *okteta, glances, htop y vlc.*
- 6. **Añadiendo software.** Si algunos de los paquetes en el punto anterior no está instalado, utiliza el comando *apt install* para instalarlo.





### B1.5. ANEXO: Instalación de Máquina Virtual de Linux con archivo ISO

#### B1.5.1. Objetivos

El objetivo de este anexo de prácticas es el de familiarizar al alumno con la configuración de una máquina virtual de Linux genérica. Los archivos ISO, también conocidos como imágenes ISO, almacenan una copia exacta de un sistema de ficheros y son el formato más utilizado para distribuir copias de sistemas operativos. Cuando utilizamos una imagen ISO para crear una máquina virtual, tendremos una copia exacta de la distribución del SO contenida en el archivo ISO. Por lo tanto, nuestro SO estará "vacíoz tendremos que instalar el software necesario.

#### B1.5.2. Plan de trabajo

El plan de trabajo de esta sesión será el siguiente:

- 1. Lectura del boletín por parte del alumno.
- 2. Realización de forma individual en casa de los ejercicios propuestos en este anexo.

#### B1.5.3. Ejercicios a realizar durante la sesión

- Descarga la distribución de Ubuntu. Vamos a crear una máquina virtual de Linux, en particular, de una distribución de Ubuntu, uno de los SO Linux más aceptados y populares de hoy en día gracias a su sencillez de manejo. Descarga la versión 22.04 LTS (Ubuntu Desktop) de su página oficial: https://ubuntu.com/download/desktop/. Se descargará un fichero Disc Image File (ISO) de 3.4 GB. Mientras se descarga el archivo, pasa al siguiente punto para instalar VirtualBox.
- 2. Descarga e instala VirtualBox. Descargaremos la última versión de este hipervisor desde su página Web *https://www.virtualbox.org/wiki/Downloads*.



Figura I.11: Página Web de VirtualBox

Durante la instalación, sigue las recomendaciones por defecto seleccionando "Next". Antes de proceeder a instalar, probablemente el instalador te avisará de que tu conexión a Internet se va a desconectar temporalmente. Procede con la instalación de todas formas.

3. **Crea una máquina Virtual.** Una vez que has instalado e iniciado VirtualBox, vas a crear una máquina virtual para ejecutar la distribución de Ubuntu que has descargado en el primer paso. Para ello, pulsa el icono de "Nueva" o selecciona "Máquina/Nueva" en la barra de herramientas superior. Se abrirá un diálogo en el que debemos indicar el tipo y versión de máquina virtual que queremos crear. En nuestro caso, el tipo es "Linux" y









(b) Nombrar máquina virtual



la versión "Ubuntu" (32 o 64 bits). Además, indicaremos el nombre con el que identificaremos nuestra máquina Ubuntu, y dejaremos la carpeta por defecto dónde se alojarán todos los ficheros de nuestra máquina virtual.

En el siguiente paso, indicaremos la cantidad de memoria RAM de nuestra máquina virtual. Aunque el tamaño recomendado por VirtualBox es 1 GB, incrementa el tamaño a 4 GB (4096 MB) ya que éste es un tamaño más apropiado para obtener un rendimiento óptimo del SO virtual.

	3	× ×
← Crear máquina virtual		
Tamaño de memoria		
Seleccione la cantidad de memoria (RAM) máquina virtual.	en megabytes a ser reservac	la para la
El tamaño de memoria recomendado es 20	48 MB.	
••••••	4	096 韋 MB
4 MB	16384 MB	
	Next	Cancelar

Figura I.13: Tamaño de memoria RAM de máquina virtual

En el siguiente paso (Figura I.14), configuramos el disco duro de nuestra máquina virtual. Claramente, este disco será virtual, simulado por un archivo. Deja seleccionado 'VDI' que es el formato de disco duro predeterminado por VirtualBox. En el siguiente paso, selecciona 'reservado dinámicamente' para que el espacio reservado por la máquina para el disco duro se ajuste al espacio utilizado y así no ocupar espacio innecesario. En el último paso, indica el tamaño del disco. Aunque Ubuntu recomienda usar al menos 25GB, puesto que el uso de disco de nuestra máquina virtual durante las prácticas de esta asignatura será limitado, dejamos un tamaño de 20 GB.

Ahora que hemos creado nuestra máquina virtual, podemos ver todas las opciones que hemos configurado (Figura I.15).

Todavía nos falta un paso clave que es la instalación del SO (Figura I.16). Pulsamos el icono de Configuración de la máquina virtual y seleccionamos 'Almacenamiento' en la barra vertical de la izquierda. Debajo de 'Controlador: IDE', pulsa donde pone vacío. En el panel de la derecha, en 'Unidad óptica', pulsa en el icono de disco azul y haz click en 'Seleccionar un archivo de disco'. Selecciona el archivo de Ubuntu que has descargado en el primer paso.



	?
	Crear de disco duro virtual
? ×	Tipo de archivo de disco duro
Crear máquina virtual	Selecione el tipo de archivo que quiere usar para el nuevo disco duro virtual. Si no necesita usar con otro software de virtualización puede dejar esta configuración sin cambiar.
Disco duro	VDI (VirtualBox Disk Image)     VHD (Virtual Hard Disk)
Si desea puede añadir un disco duro virtual a la nueva máquina. Puede crear un nuevo archivo de disco duro o seleccionar uno de la lista o de otra ubicación usando el icono de la carpeta.	VMDK (Virtual Machine Disk)
<ul> <li>Si necesita una configuración de almacenamiento más compleja puede omitir este paso y hacer los cambios a las preferencias de la máquina virtual una vez creada.</li> </ul>	
El tamaño recomendado del disco duro es 10.00 GB.	
🔿 No añadir un disco duro virtual	
<u>Crear un disco duro virtual ahora</u>	
◯ <u>U</u> sar un archivo de disco duro virtual existente	
Vacío 👻 🗔	
Crear Cancelar	Modo <u>e</u> xperto <u>N</u> ext Cancela
(a) Crea disco ahora	(b) Tipo de archivo de disco duro
(u) ereu uisee uneru	
? ×	? >
Crear de disco duro virtual	← Crear de disco duro virtual
Almacenamiento en unidad de disco duro física	Ubicación del archivo y tamaño
Seleccione si el nuevo archivo de unidad de disco duro virtual debería crecer según se use (reserva dinámica) o si debería ser creado con su tamaño máximo (tamaño fijo).	Escriba el nombre del archivo de unidad de disco duro virtual en el campo debajo o haga clic en icono de carpeta para seleccionar una carpeta diferente donde crear el archivo.
Un archivo de disco duro reservado dinámicamente solo usará espacio en su disco físico a	C:\Users\Victoria\VirtualBox VMs\test\test.vdi
medida que se llena (hasta un máximo tamaño fijo), sin embargo no se reducirá de nuevo automáticamente cuando el espacio en él se libere.	- Seleccione el tamaño de disco duro virtual en megabytes. Este tamaño es el límite para el archi de datos que una máquina virtual podrá almacenar en el disco duro.
Un archivo de disco duro de <b>tamaño fijo</b> puede tomar más tiempo para su creación en algunos	20.00
Bernende, però non mannence es mas rapido ai asano.	
	4.00 MB 2.00 TB
U Tamano tijo	
Next Cancelar	Crear Cancela
(c) disco duro dinámico	(d) Tamaño de disco

Figura I.14: Configuración de disco duro de máquina virtual

r Gracie vm virtualisox Administrador Archivo Máquina Avuda		- • ×
Herramientas	Neva Configuración Descatar Iniciar	
WUbuntu 🕐 Apagada	General Nembre: MVUbuntu Sistems operative: Ubuntu (44-bit)	Previsualización
	Sistema Memoria base: 1024 M8 Orden de arranque: Diquite, Óptics, Disco duro Aceleración: Paravitualización KVM	MVUbuntu
	Pantala Memoria de Video: 16 MB Controledor gráfico: VMSVGA Servidor de escritorio remois: Inhobilidado Grabación: Inhobilidado	
	Almacenamiento     Controlodor: IDE     BOT Secondrom mestro:     Controlodor: SATA     Porto SATA     Porto SATA     Porto SATA	
	Audio     Cerronador de anfitrán: Windows DirectSound     Cerronador: ICH ACS7	
	Adaptador 1: Intel PRO/1000 MT Desktop (NAT)	
	USB     Controledor USB: OHCI     Filtros de dispositivos: 0 (0 activo)	
	Carpetas compartidas	
	Ninguno  Secripción	
	- Ninguno	

Figura I.15: Máquina virtual creada

4. **Instala el SO en tú máquina virtual.** Para comenzar la instalación de Ubuntu en tu máquina virtual, pulsa el icono "Iniciar" de tu máquina. Se abrirá una ventana de inicio del sistema operativo.

Ten en cuenta que para poder usar máquinas virtuales en tu ordenador, tu equipo tiene que permitir la



👽 Oracle VM VirtualBox Administrador Archivo Māduina Avuda					- 0 ×
Herramientas	Nueva Configuraci	ón Descartar Iniciar			
WVUbuntu (1) Apogada	General Nombre: Sistema operativo:	MVUbuntu Ubuntu (64-bit)			Previsualización
	Sistema				
	MVUbuntu - Configuraciór	1	?	×	MVUbuntu
	General	Almacenamiento			
	Sistema	Dispositivos de <u>a</u> lmacenamiento	Atributos		
	Pantalla	Controlador: IDE	Unidad <u>ó</u> ptica: IDE secundario maestro		
		- 💽 Vacio	CD/DVD vivo	Seleccionar/crear un disco óptico vir	tual
	Almacenamiento	Controlador: SATA	Información	Seleccionar un archivo de disco	
	🕩 Audio	MVUbuntu.vdi	Tamaño:	Eliminar disco de la unidad virtual	
	Red Red		Ubicación:		
	Nuertos serie		Conectado a:		
	💕 USB				
	Carpetas compartidas				
	Interfaz de usuario				
		🔶 💮 🔛			
			Aceptar Ca	ancelar	

Figura I.16: Configuración SO de Máquina virtual

virtualización. Normalmente esto viene activado por defecto, pero de caso contrario puedes activarlo en la BIOS en la opción 'Intel Virtualization Technology' si tu ordenador cuenta con un procesador Intel o 'SVM Mode' si tu ordenador cuenta con un procesador AMD.

En el desplegable inicial, selecciona 'Try or install ubuntu' (Figura I.17). Entonces, aparecerá el entorno gráfico que nos pregunta si queremos probar o instalar Ubuntu. Selecciona "Probar Ubuntu", lo cual te permitirá cambiar la resolución de la pantalla para una instalación posterior más comoda.

		Jul 9 16:04 Instalar	
	Le damos la bie	envenida	
GNU GRUB version 2.06	English Español Esperanto		
ry or Install Ubuntu buntu (safe graphics) EM install (for manufacturers) est memory	Euskara Français Gaeilge Galego	$\odot$	
	Hrvatski Íslenska Italiano Kurdî	Probar Ubuntu	Instalar Ubuntu
	Latviski Lietuviškai Magyar	Puede probar Ubuntu sin hacer ningún cambio er este CD.	n su equipo, directamente desde
Use the 1 and 4 keys to select which entry is highlighted. Press enter to boot the selected OS, 'e' to edit the commands before booting or 'c' for a command-line.	Nederlands No localization (UTF-8) Norsk bokmål	O si está listo, puede instalar Ubuntu junto a (o e actual. Esto no tardará demasiado.	n lugar de) su sistema operativo
	Norsk avaarsk	Es posible que quiera leer las <u>notas de publicació</u>	<u>on</u> .

(a) Arranque de instalador

(b) Selección de probar Ubuntu

Figura I.17: Selección de prueba de Ubuntu

Se iniciará Ubuntu en modo prueba, es decir, que lo podemos usar pero ningún dato sería guardado. Para hacer más cómoda la instalación, cambia la resolución de la ventana para poder verla a un tamaño normal (Figura I.18). Para ello, haz click en los iconos de la esquina superior derecha de la ventana. En el menú desplegable, selecciona 'Configuración'. Una vez se abra la configuración, en el menú de la izquierda, baja hasta encontrar el apartado 'Monitores'. En el panel de la izquierda, selecciona Resolución y elige la resolución más conveniente según tu monitor. Tras cambiar la resolución, indica "Mantener cambios" si todo se ve correctamente.

Ahora, instala Ubuntu haciendo click en el icono del escritorio "Instalar Ubuntu 22.04.1 LTS". En el instalador, selecciona el idioma y la configuración del teclado (Figura I.19). En el siguiente paso nos preguntará qué





(a) Configuración en Ubuntu



Figura I.18: Inicio de instalación de Ubuntu

tipo de instalación queremos hacer (Figure I.20a). En este caso seleccionaremos la instalación normal, con actualizaciones de Ubuntu. Puede que al realizar este paso, la ventana se vuelva pequeña de nuevo. Si esto es así, sin cerrar la ventana de instalación, vuelve a realizar los pasos que se indicaron antes para cambiar la resolución.

Activities	Install Ubuntu 22.04 LTS	Jul 9 17:06	eni	A 🗣 🗎	Activities 🛈 Instal
() ()					
• 💯		Install			
	Welcome				<b>a</b>
	Welcome				
	English				0
	Español	You may wish to read the <u>release notes</u> .			
	Esperanto				
	Euskara				
	Gaeilge				
0	wthmbox jo				
	Hrvatski				
-	Italiano				
	Kurdî				
	Latviski				
	Lietuviškai				
	Nederlands				
	No localization (UTF-8)				
	Norsk bokmål				
			Quit Back Cont	inue	
		• • • • • • • • • • • • • • • • • • • •			
••••					

(a) Selección de idioma



(b) Configuración de teclado

Figura I.19: Selección de idioma en instalación de Ubuntu

Tras esto, el instalador nos pregunta dónde queremos instalar Ubuntu. Aquí seleccionaremos que queremos 'Borrar el disco e instalar Ubuntu' (Figure I.20b). Cabe destacar que esto no borrará nada del disco de tu ordenador personal, si no que borrará el disco virtual que creamos previamente en VirtualBox.

El instalador nos pedirá que indiquemos dónde nos encontramos para ajustar nuestra zona horaria. Bastará con pulsar el país en el que te encuentres, para que este ajuste la zona horaria pertinente.

Como siguiente y último paso antes del inicio de la instalación, Ubuntu nos solicitará el nombre del equipo, el nombre de usuario y la contraseña (Figura I.22). Rellenaremos el formulario e indicaremos si queremos o no poner la contraseña al iniciar una sesión. Recuerda esta contraseña ya que se necesitará para instalar programas y administrar tu sistema operativo.

Ya completado este paso, comenzará la instalación. Ubuntu nos mostrará ahora ciertas configuraciones. Tendremos que esperar hasta que finalice y cuando esta termine, reiniciar la máquina virtual. Al reiniciar la







(a) Tipo de instalación





#### Figura I.20: Configuración de la instalación

Figura I.21: Configuración de zona horaria de Ubuntu

máquina virtual, nos pedirá que extraigamos el medio de instalación. Como es una máquina virtual y no tenemos medio de instalación simplemente pulsaremos la tecla "Intro".

Una vez reiniciado el equipo, probablemente la resolución se haya desconfigurado. Sigue los mismos pasos que hemos indicado antes para establecer la resolución que mejor vaya a tu tamaño de pantalla. Además, Ubuntu nos mostrará ahora configuraciones, las cuales vamos a omitir. En la esquina superior derecha, pulsa sobre "Omitir" y "Siguiente".

Una vez llegado a este punto, ya tendrás la máquina virtual de Ubuntu completamente funcional.

5. Cierra tu máquina virtual. Para apagar tu máquina virtual de Ubuntu, en la barra superior de ésta, selecciona Archivo/Cerrar. Aparecerá la ventana mostrada en la Figura I.23. Aquí puedes seleccionar 'Guardar el estado de la máquina' si deseas, al volver a iniciarla, continuar justo en el punto donde se cerró. En este caso, todo lo que has estado haciendo se pausará hasta que vuelvas (no se pierde ningún tipo de información), pero la máquina virtual no consumirá recursos. Si quieres apagar la máquina totalmente, lo que equivaldría a pulsar el botón de apagado/encendido de una máquina física, selecciona 'Enviar señal de apagado'. No selecciones 'Apagar la máquina', puesto que esto es un apagado abrupto, equivalente a tirar del cable de alimentación, por lo que el SO no se cerrará adecuadamente y podrían haber perdidas de información.





Activitie	Install Ubuntu 22.04 LTS	Jul 10 18:21		en	A (	) <b>+</b>
. 🀑		Install				
6	Who are you?					
9	Your name:	vbm	•			
	Pick a username	The name it uses when it talks to other computers.				
0	Choose a password:	Deeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeee				
	comminged password.	Log in automatically     Require my password to log in				
A		Use Active Directory You'll enter domain and other details in the next step.				
?						
6			Back	Continue		
	Show Applications	•••••				

Figura I.22: Configuración de usuario de Ubuntu



Figura I.23: Cierre de máquina virtual

6. Instalar las guest additions. Consisten en controladores de dispositivos y aplicaciones del sistema que optimizan el sistema operativo huésped para mejorar el rendimiento y la usabilidad, habilitando funcionalidades avanzadas tales como integración del ratón, soporte gráfico mejorado, portapapeles y carpetas compartidas entre *host* y guest. Las guest additions deben instalarse dentro de la máquina virtual después de que el sistema operativo huésped haya sido instalado. Existe algunos prerrequisitos que hay que instalar, ejecutando el siguiente comando desde un terminal en el huésped: sudo apt install build-essential dkms linux-headers-\$(uname -r). Después, en el menú "Dispositivos" de la máquina virtual, haz clic en *Insertar la imagen de CD de las* "Guest additions", y verás que te aparece un disco llamado VBox\_GAs\_... Tras esto, ejecuta sudo /media/nombreusuario/VBox\_GAs\_x.y.z/VBoxLinuxAdditions.run, reemplazando nombreusuario por el nombre de usuario que hayas elegido, y x.y.z por tu versión concreta de VirtualBox.