

Uso de máquinas virtuales y Shell de comandos

© EDICIONES ROBLE, S.L.

Índice

I. Introducción	3
II. Objetivos	4
III. El concepto de máquina virtual. Virtual Box	5
IV. Creación y configuración de una máquina virtual	9
4.1. Instalación de Virtual Box	9
4.2. Creación y configuración de una máquina virtual	13
V. Carga de una máquina virtual	39
VI. La shell de comandos de Linux. Creación de scripts	50
6.1. La shell de comandos	50
6.2. Creación de scripts	71
VII. Resumen	90
VIII. Caso práctico	91
Recursos	93
Enlaces de Interés	93
Bibliografía	93
Glosario.	93

I. Introducción

Una de las principales dificultades que surge en el ámbito del desarrollo de software es la incompatibilidad de las herramientas que se van a utilizar con respecto a los diferentes sistemas operativos. Así, en muchas ocasiones, el software no es multiplataforma, es decir, se ha creado para ejecutarse en un sistema operativo concreto. Esto produce problemas si el sistema operativo sobre el que se va a trabajar no coincide con el que es necesario para el software que se quiere utilizar, por lo que en estos casos sería necesario instalar el sistema operativo requerido por el software.

Otro problema más particular, y ligado al ámbito del análisis de datos y del Big Data, es la complejidad de la instalación y configuración de las herramientas informáticas que se utilizan. En muchos casos, el proceso de instalar y configurar la herramienta con los parámetros adecuados requiere de unos conocimientos de un perfil administrador muy diferente al del usuario de la herramienta concreta. A estas dificultades habría que añadir el tiempo requerido para realizar estas operaciones.

En esta unidad del módulo se van a estudiar las máquinas virtuales como solución a estos problemas. Más concretamente, se analizará la herramienta Virtual Box para crear y gestionar máquinas virtuales. Asimismo, en esta unidad se realizará una introducción sobre el uso de la Shell de comandos de Linux, ya que muchas de las herramientas que se utilizan en el ámbito del Big Data se ejecutan sobre entornos de tipo Linux y la manera de interactuar con dichas herramientas es a través del uso de comandos de la Shell.

La unidad se estructura de la siguiente forma. En primer lugar, se realizará una introducción al concepto de máquina virtual y su necesidad de uso. A continuación, se describirá cómo instalar Virtual Box y cómo se puede crear y configurar de manera básica una máquina virtual. En la siguiente sección se mostrará cómo cargar y utilizar una máquina virtual previamente creada. La última sección de la unidad se centra en revisar el uso de los comandos de la Shell de Linux.

II. Objetivos

Los objetivos que los alumnos alcanzarán tras el estudio de esta unidad son:



- Entender el concepto y la necesidad del uso de máquinas virtuales.
- Saber instalar la herramienta Virtual Box.
- Saber crear y configurar de manera básica una máquina virtual en Virtual Box.
- Saber cómo se carga una máquina virtual dada en Virtual Box.
- Conocer y saber utilizar de manera básica la Shell de comandos de Linux.

III. El concepto de máquina virtual. Virtual Box

La existencia de diferentes sistemas operativos acarrea, como una de las principales desventajas en el ámbito del desarrollo de software, la necesidad de tener que crear para un mismo software diferentes versiones adecuadas para cada uno de los sistemas operativos.

Sin embargo, esto no ocurre siempre, ya que la creación de una versión del mismo software para diferentes plataformas supone un esfuerzo económico para las empresas que no siempre es rentable. Esta situación produce problemas en el momento que se quiere utilizar un determinado software que solo se ejecuta para un determinado sistema operativo que no coincide con el que dispone el usuario.

Soluciones anteriores

Antes de la aparición de las máquinas virtuales, la única solución posible consistía en instalar en la misma máquina los sistemas operativos que eran necesarios para el software que se iba a utilizar, compartiendo de esta manera el disco duro entre ellos. Se trata de una solución costosa tanto en tiempo —el necesario para instalar y poner a punto el sistema operativo correspondiente—, como en la cantidad de memoria del disco que deben compartir los diferentes sistemas operativos instalados.

La solución de la máquina virtual

En este sentido, las máquinas virtuales aparecen como una solución a la necesidad de tener que usar diferentes sistemas operativos en una misma máquina.

La idea que sostiene el funcionamiento de las máquinas virtuales consiste en usar el sistema operativo instalado por defecto en la máquina, denominado sistema anfitrión, para ejecutar una simulación de otro sistema operativo denominado sistema huésped.

El sistema huésped se ejecuta como si fuera un programa más y se puede utilizar como si realmente estuviera instalado en la máquina —aunque realmente no se encuentra instalado y lo único que se está ejecutando es una simulación—.

Crear y ejecutar una máquina virtual

Para poder crear y ejecutar una máquina virtual, es necesario un software especial encargado de llevar a cabo la simulación. Actualmente, las herramientas más utilizadas para la creación y gestión de máquinas virtuales son Virtual Box y VMware.

La herramienta VMware dispone de una versión gratuita con las características y funciones necesarias para un usuario particular, tales como la creación de una máquina virtual, y una versión de pago orientada a empresas que añade funciones y características más avanzadas tales como la posibilidad de virtualizar sistemas operativos y gestionarlos a través de la red como si se tratase de una nube.

Por otro lado, Virtual Box es una herramienta de código abierto, totalmente gratuita y disponible para cualquier sistema operativo, propiedad de Oracle. No ofrece las mismas características avanzadas orientadas hacia las empresas de VMware, pero sí dispone de todas las funciones necesarias para crear y ejecutar máquinas virtuales. E incluso dispone de algunas funcionalidades propias como la capacidad de ejecutar varias máquinas virtuales a la vez.

Con respecto a la eficiencia y rendimiento, VMware presenta un rendimiento mejor que Virtual Box en el uso de la memoria y de la CPU. Sin embargo, con respecto al rendimiento del disco duro ambas herramientas son similares.

Indicaciones para su uso

Cuando se utilizan máquinas virtuales, hay que tener en cuenta lo siguiente:

- Todas las operaciones que se realicen sobre y desde el sistema operativo virtualizado no afectarán al sistema operativo anfitrión. De hecho, esta es una de las ventajas de usar máquinas virtuales: si ocurre algún problema en el sistema operativo virtualizado, el sistema operativo anfitrión quedará resguardado y bastará con interrumpir la ejecución de la máquina virtual.
- Cada vez que se ejecuta una máquina virtual, en el momento en que se apaga, es posible guardar el estado de la máquina virtual. Es decir, es posible mantener el sistema tal como estaba antes de apagarse, de manera que cuando se vuelve a ejecutar, el usuario podrá continuar en el mismo estado en que lo dejó.
- En general, es posible el intercambio de documentos entre los sistemas operativos anfitrión y huésped, lo que facilita la reutilización de todo lo generado en un sistema operativo virtualizado y el uso de recursos que se encuentran en el sistema operativo anfitrión.
- Desde un sistema operativo virtualizado se pueden utilizar recursos auxiliares como el acceso a la red de comunicación, impresoras u otros dispositivos que se encuentran conectados a la máquina sobre la que se ejecuta.

Principales casos de uso

Hay dos casos de uso principales cuando se utilizan este tipo de herramientas:

Crear una máquina virtual

El primer caso surge cuando existe la necesidad de preparar un entorno virtual para ejecutar determinado software y no es posible hacerlo en el sistema operativo instalado por defecto en la máquina.

Cargar una máquina virtual

El otro caso de uso consiste en utilizar una máquina virtual que alguien previamente ha creado y configurado y que se desea utilizar para poder acceder a determinado software instalado en la misma.

Obsérvese, con respecto a este último caso, que existen sitios web que actúan como repositorios de máquinas virtuales que han sido creadas por otros usuarios y que ponen a libre disposición de otros usuarios para que puedan descargarlas y usarlas.

Servicios en la nube

También se llama la atención sobre el hecho de que, además de las herramientas comentadas anteriormente, existe otra forma alternativa de crear y utilizar máquinas virtuales en el contexto de los denominados servicios en la nube.



La computación en la nube —también denominado *cloud computing*— es un modelo de prestación de servicios en el que los usuarios pueden acceder a servicios o recursos que se encuentran en Internet mediante el pago por el consumo efectuado y, en algunos casos, de forma gratuita. Entre los servicios ofrecidos en la nube por parte de las empresas que se dedican a ello, se encuentra la creación de máquinas virtuales, también denominadas instancias.

La creación de una instancia consiste en definir las características de la máquina virtual: sistema operativo, capacidad de almacenamiento, CPU, memoria, capacidad de Red...

A la instancia se le asocia una IP pública que servirá para realizar la conexión a la misma. Existe una relación entre las herramientas de virtualización y las instancias de máquinas virtuales de los servicios de la nube que consiste en la posibilidad de importar y exportar máquinas virtuales de un servicio en la nube a una máquina virtual local y viceversa.



Por ejemplo, en la página web de Amazon se describe la importación y exportación de máquinas virtuales entre los servicios en la nube de Amazon (AWS) y la herramienta de virtualización VMware (Amazon. VM Import/Export. [En línea] URL disponible en <https://aws.amazon.com/es/ec2/vm-import>)

Virtual Box

Esta unidad se va a centrar en el uso de Virtual Box, ya que cubre las necesidades esenciales requeridas en este ámbito, que son esencialmente evitar las incompatibilidades del software que se va utilizar con el sistema operativo instalado en una máquina —asegurando así que se podrá hacer uso del software con independencia de la máquina utilizada— y evitar al usuario tener que realizar la instalación y configuración del software, usando para ello máquinas virtuales que ya lo tienen instalado y configurado, lo que solo requerirá cargar la máquina virtual correspondiente.

IV. Creación y configuración de una máquina virtual



Virtual Box es un programa de software libre que permite la instalación de otro sistema operativo, dentro del sistema existente, y la interacción con el mismo. De esta forma, es posible instalar programas diseñados para los sistemas operativos clientes. En esta sección se va a describir cómo se instala Virtual Box, además de cómo se puede crear y configurar de una manera básica una máquina virtual.

4.1. Instalación de Virtual Box

Sitio de descarga de Virtual Box

Para instalar Virtual Box, en primer lugar, hay que descargar el software desde la siguiente dirección (figura 1.1.):

- Página web de descargas de Virtual Box, <https://www.virtualbox.org/wiki/Downloads>

The screenshot shows the 'Download VirtualBox' section of the VirtualBox website. At the top, there's a logo for Oracle VM VirtualBox. Below it, the text 'Download VirtualBox' is prominently displayed. A subtext says 'Here, you will find links to VirtualBox binaries and its source code.' Underneath, there's a heading 'VirtualBox binaries' with a note: 'By downloading, you agree to the terms and conditions of the respective license.' There are three main bullet points:

- **VirtualBox 5.1.22 platform packages.** The binaries are released under the terms of the GPL version 2.
 - Windows hosts
 - OS X hosts
 - Linux distributions
 - Solaris hosts
- **VirtualBox 5.1.22 Oracle VM VirtualBox Extension Pack** [All supported platforms](#)
Support for USB 2.0 and USB 3.0 devices, VirtualBox RDP, disk encryption, NVMe and PXE boot for Intel cards. See [this chapter from the User Manual](#) for an introduction to this Extension Pack.
The Extension Pack binaries are released under the [VirtualBox Personal Use and Evaluation License \(PUEL\)](#).
Please install the extension pack with the same version as your installed version of VirtualBox:
If you are using [VirtualBox 5.0.40](#), please download the extension pack [here](#).
- **VirtualBox 5.1.22 Software Developer Kit (SDK)** [All platforms](#)

Figura 1.1. Sitio de descarga de Virtual Box.

Descarga de Virtual Box

Durante esta unidad, se os enseña cómo descargar e instalar la versión 5.1.22. Sin embargo, es aconsejable que os descarguéis la última versión disponible. En las explicaciones siguientes se utilizará la versión para Windows —para el resto de sistemas operativos el proceso es similar—. En la página de descargas, se pulsa sobre el enlace correspondiente al sistema operativo sobre el que se quiere instalar —en este caso sobre Windows hosts— y comienza la descarga (figura 1.2.).



Figura 1.2. Descarga de Virtual Box.

Instalador de Virtual Box

Una vez que se ha bajado el archivo ejecutable, se pulsa sobre el instalador para que comience la ejecución del asistente (figura 1.3.).

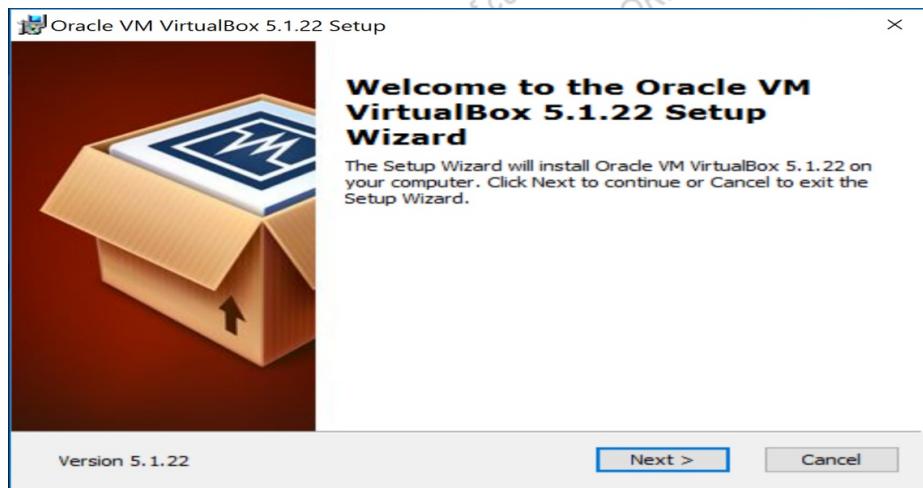


Figura 1.3. Instalador e Virtual Box.

Ventana "Custom Setup"

Se pulsa sobre “Next” y aparece la ventana “Custom Setup”. Se deja la selección “VirtualBox Application” y se comprueba que la carpeta de instalación elegida por el instalador es C:\Program Files\Oracle\VirtualBox (figura 1.4.).

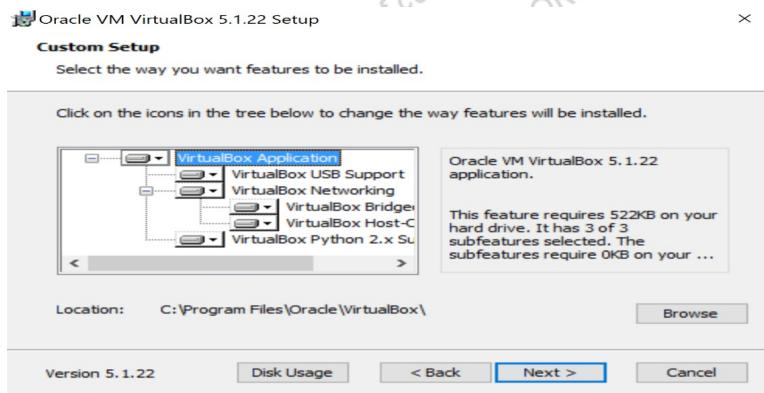


Figura 1.4. Ventana "Custom Setup".

Opciones de configuración

Se pulsa sobre “Next” en la siguiente ventana, sin cambiar las opciones que aparecen por defecto (figura 1.5.).

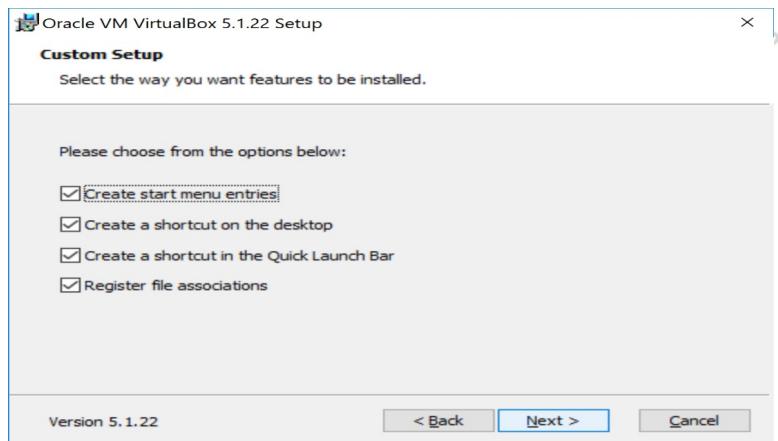


Figura 1.5. Opciones de configuración.

Ventana de desconexión de la red

En la siguiente ventana que aparece, se indica que el ordenador se va a desconectar momentáneamente de la red. Se pulsa sobre “Yes” (figura 1.6.).

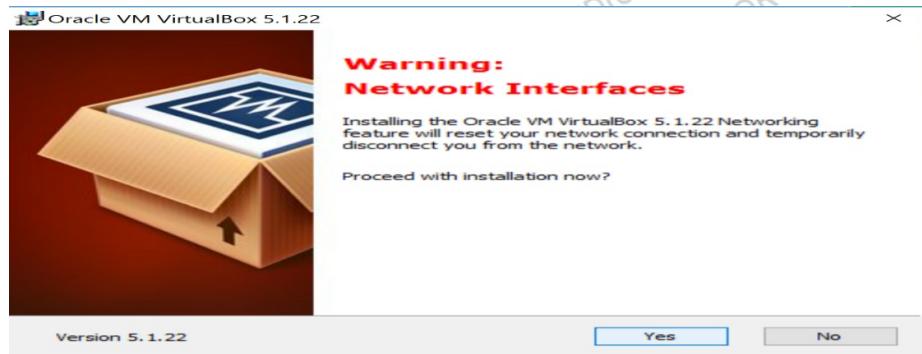


Figura 1.6. Ventana de desconexión de la red.

Ventana de instalación

Por último, en la siguiente ventana se pulsa sobre “Install” para que comience la instalación (figura 1.7.).

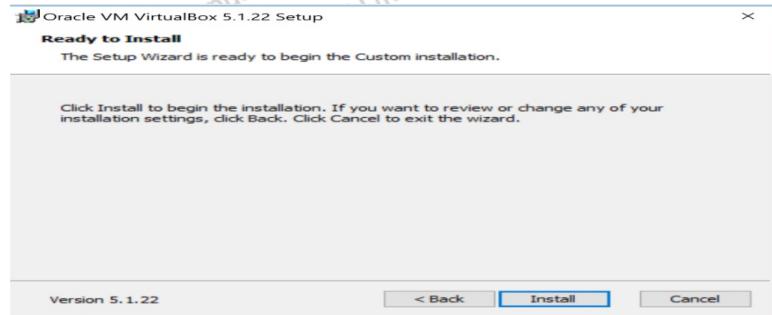


Figura 1.7. Ventana de instalación.

Ventana de finalización de instalación

A continuación, aparecerá un aviso de Windows para preguntar si el programa puede hacer cambios en el sistema. Se pulsa sobre "Yes" y se deja que el instalador se ejecute hasta que finalice la instalación. En ese momento, aparece la pantalla de fin de instalación. Se pulsa sobre "Finish", que ejecutará Virtual Box (figura 1.8.).

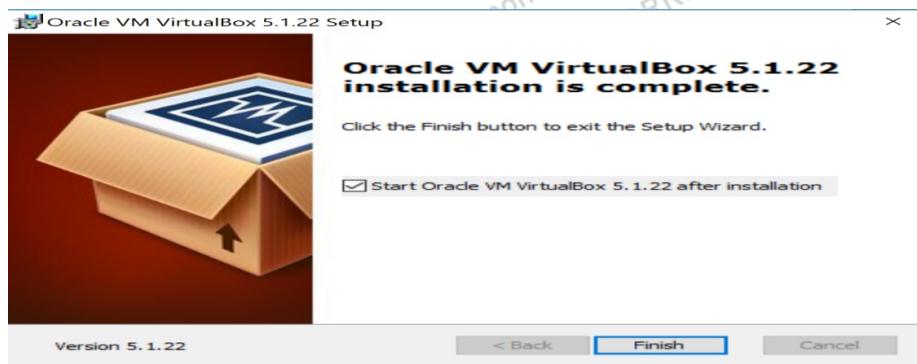


Figura 1.8. Ventana de finalización de instalación.

Interface principal de Virtual Box

En la figura 1.9., se puede observar la interface principal de Virtual Box.

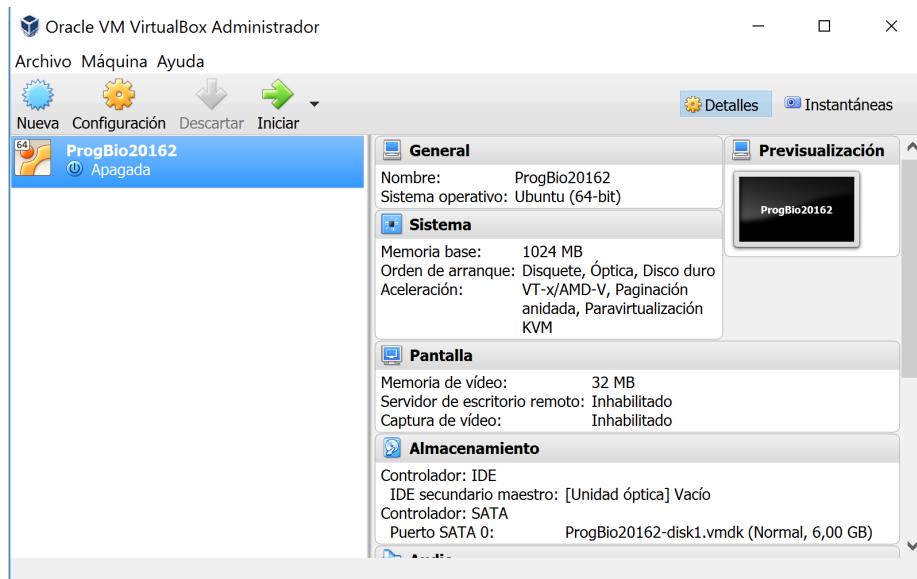
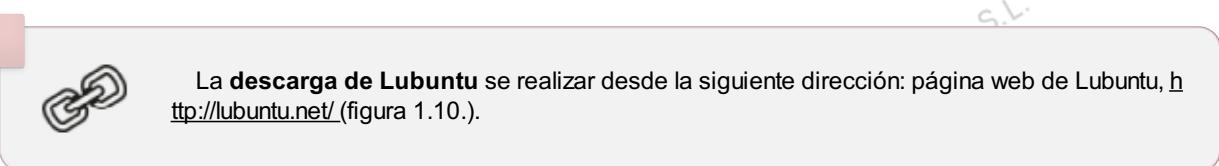


Figura 1.9. Interface principal de Virtual Box.

4.2. Creación y configuración de una máquina virtual

Para crear una máquina virtual, lo primero que se necesita es una imagen del sistema operativo que se desea virtualizar. Para ilustrar la creación de la máquina virtual, se va a utilizar Lubuntu. Se trata de una distribución de Linux pensada para máquinas que disponen de pocos recursos software.



Página web de descargas de Lubuntu

The screenshot shows the Lubuntu download page. At the top, there's a navigation bar with links for download, wiki, forum, documentation, and contact. Below the navigation, there are two main sections: 'get lubuntu' and 'lubuntu social'. The 'get lubuntu' section contains a logo, a call to action to get Lubuntu, and download links for 'lubuntu x86 CD' and 'lubuntu 64-bit CD'. The 'lubuntu social' section contains a logo, a call to action for promotion, and links to Lubuntu's Facebook page and Twitter account. At the bottom, there's a note about choosing the right download for Windows 8 or UEFI firmware, and links for 'Download lubuntu (Intel x86) desktop CD', 'Download Torrent', 'Download lubuntu 64-bit (AMD64) desktop CD', and another 'Download Torrent' link.

Figura 1.10. Página de descargas de Lubuntu.

Descarga de Lubuntu

En la página de descargas, se pulsa sobre el enlace más adecuado a la máquina que se posea (figura 1.11.).

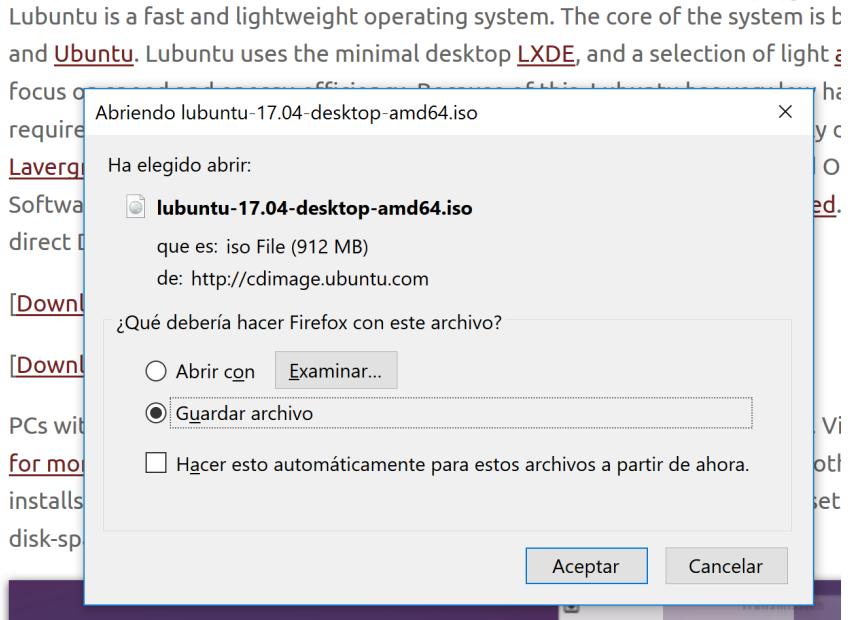


Figura 1.11. Descarga de Lubuntu.

Asistente para crear una máquina virtual

Una vez descargado Lubuntu, se va a realizar la creación de la máquina virtual. Para ello, se abre Virtual Box pulsando sobre el ícono que habrá en el escritorio, o bien en la lista de programas, o bien sobre el archivo VirtualBox.exe que se encuentra en la carpeta de instalación C:\Program Files\Oracle\VirtualBox. En la interface principal de Virtual Box, se pulsa sobre el ícono “Nueva” para crear una nueva máquina virtual. Como resultado, se abre el asistente (figura 1.12.).

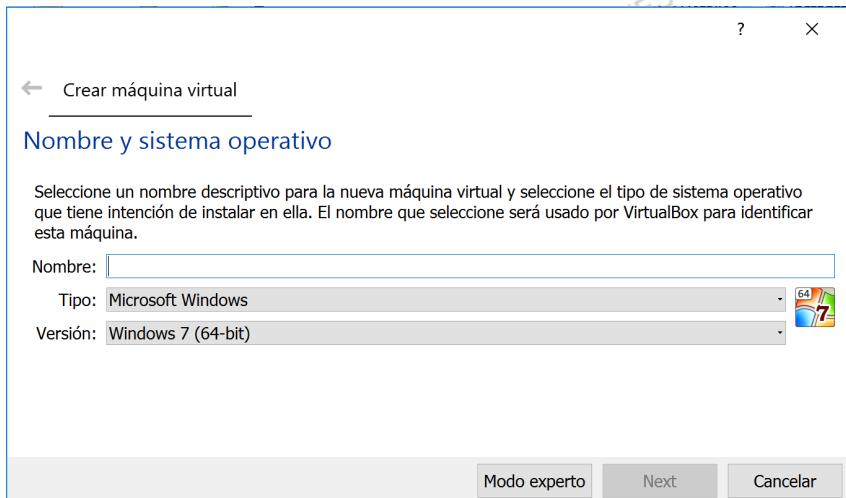


Figura 1.12. Asistente para crear una máquina virtual.

Datos de la máquina virtual

En el asistente, se inserta un nombre para la máquina virtual “Lubuntu-Virtualizado”, se elige como tipo “Linux” y como versión “Ubuntu (64-bit)”. Por último, se pulsa sobre el enlace “Next” (figura 1.13.).

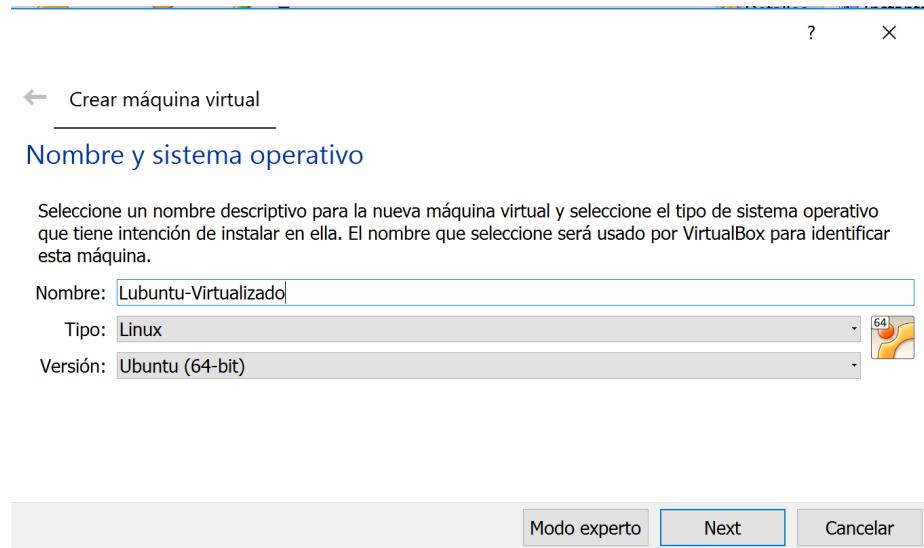


Figura 1.13. Datos de la máquina virtual.

Tamaño de la memoria de la máquina virtual

En la siguiente ventana, se debe elegir cuánta memoria se va asignar al sistema virtualizado (memoria que se quita a Windows), y se pulsa sobre “Next”. Se puede seleccionar la recomendación que indica el asistente (figura 1.14.).

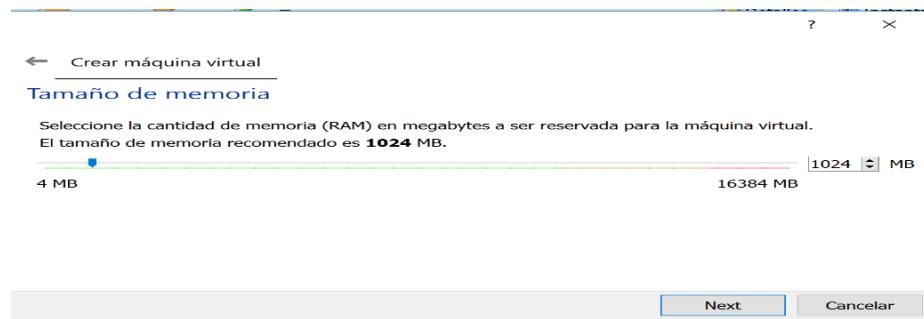


Figura 1.14. Tamaño de la memoria de la máquina virtual.

Creación de máquina virtual

En la siguiente ventana, se verifica que se ha seleccionado la opción de “Crear disco virtual ahora” y se pulsa sobre “Crear” (figura 1.15.).

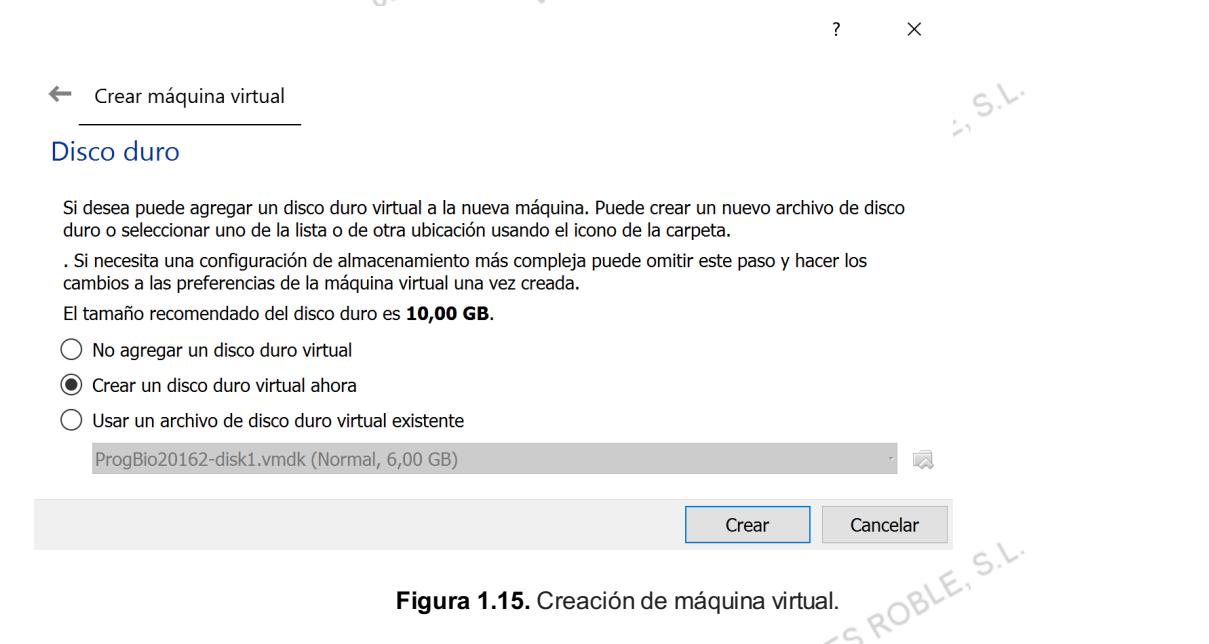


Figura 1.15. Creación de máquina virtual.

Selección del tipo de archivo del disco duro

En la siguiente ventana, se elige el tipo de archivo que se desea utilizar para el disco duro virtual. Se puede dejar la recomendación que indica el asistente (figura 1.16.) y se pulsa sobre “Next”.

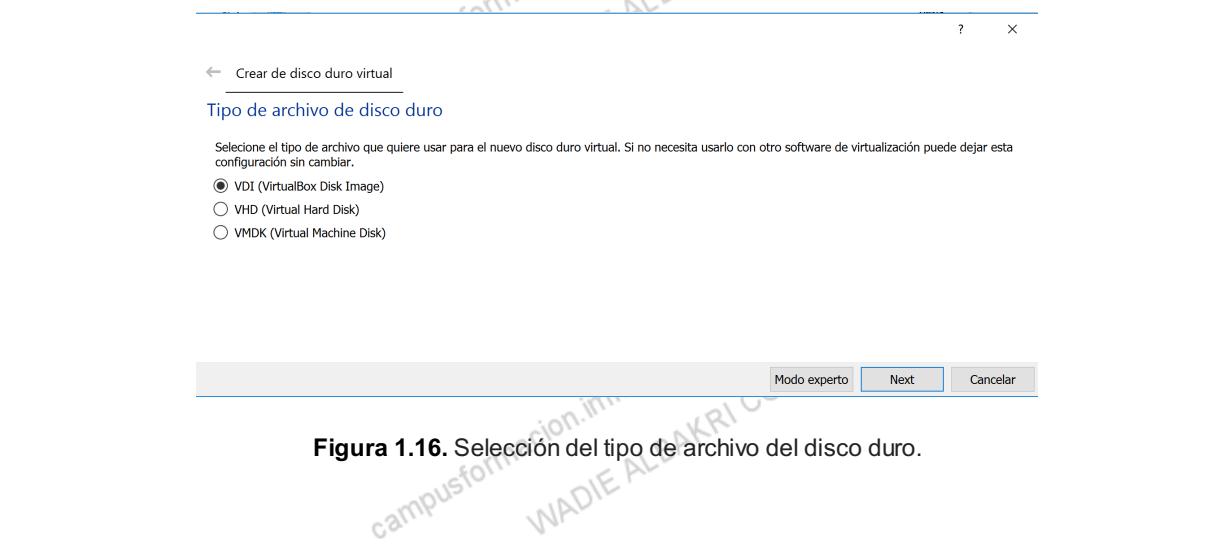


Figura 1.16. Selección del tipo de archivo del disco duro.

Selección del tipo de almacenamiento en el disco duro

En la siguiente ventana, se selecciona cómo se realizará el almacenamiento en el disco duro. Se mantiene la recomendación que indica el asistente (figura 1.17.) y se pulsa sobre “Next”.

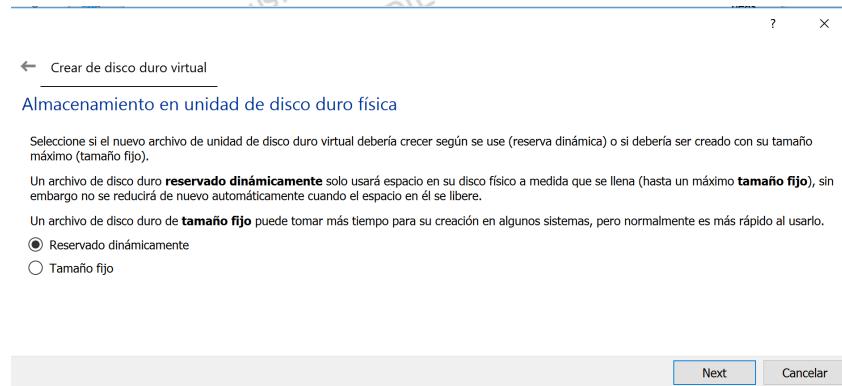


Figura 1.17. Selección del tipo de almacenamiento en el disco duro.

Ubicación del archivo del disco duro

En la siguiente ventana, se pregunta sobre dónde se debe ubicar el archivo. Por defecto, se guardará en: C:\Users\nombre_usuario\VirtualBox VMs\Nombre-Máquina Virtual.

Se puede cambiar pulsando sobre el ícono en forma de carpeta. Asimismo, en esta ventana se puede configurar el tamaño máximo del disco duro. Se puede dejar el que indica por defecto el asistente (figura 1.18.).

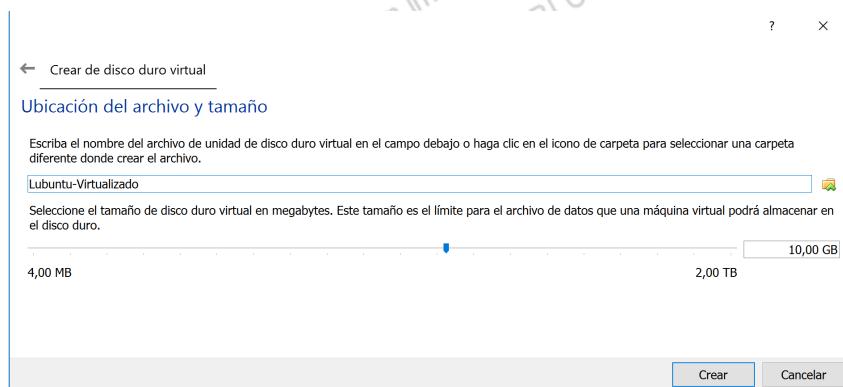


Figura 1.18. Ubicación del archivo del disco duro.

Máquina virtual creada

Al pulsar sobre “Crear”, aparece la ventana de Virtual Box actualizada con la máquina que se ha configurado (figura 1.19.).

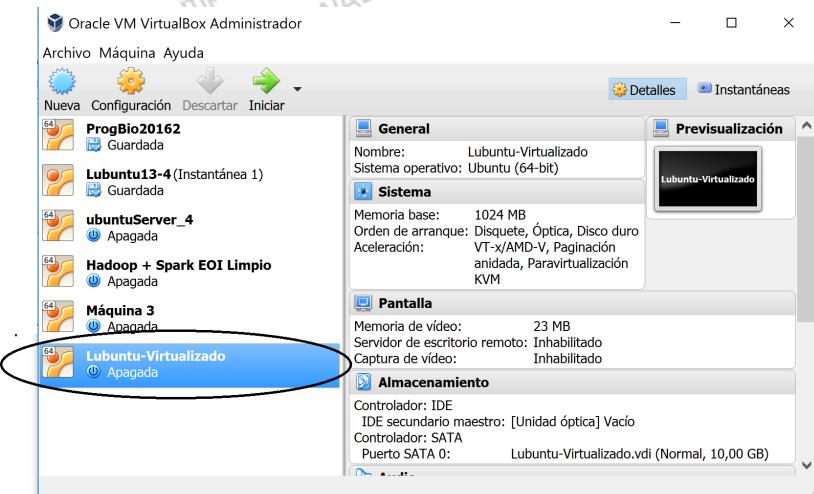


Figura 1.19. Máquina virtual creada.

Configuración de la máquina virtual

La máquina virtual aún no se ha creado, lo que se ha hecho es configurar las características de la máquina. Antes de crear la máquina es necesario configurar algunas más, lo cual se realiza seleccionando la máquina y pulsando sobre el enlace “Configuración” que genera la ventana mostrada en la figura 1.20.

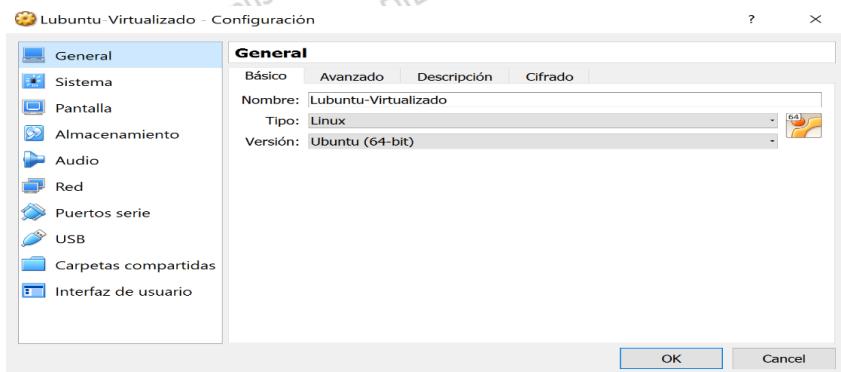


Figura 1.20. Configuración de la máquina virtual.

Pestaña de almacenamiento del asistente de configuración

En esta ventana, se pulsa sobre el ícono “Almacenamiento” y aparece la pestaña que se muestra en la figura 1.21.

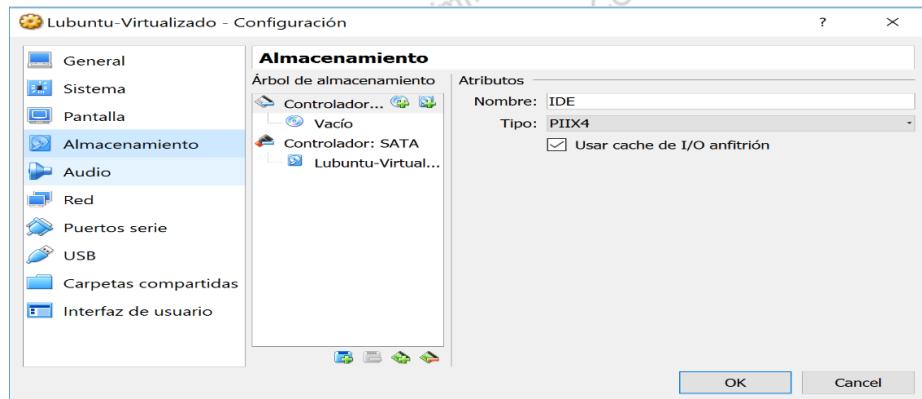


Figura 1.21. Pestaña de almacenamiento del asistente de configuración.

Configuración del controlador

En esta pestaña, se pulsa en el ícono “CD” del cuadro “Árbol de almacenamiento” del “Controlador: IDE” que aparece vacío (figura 1.22).

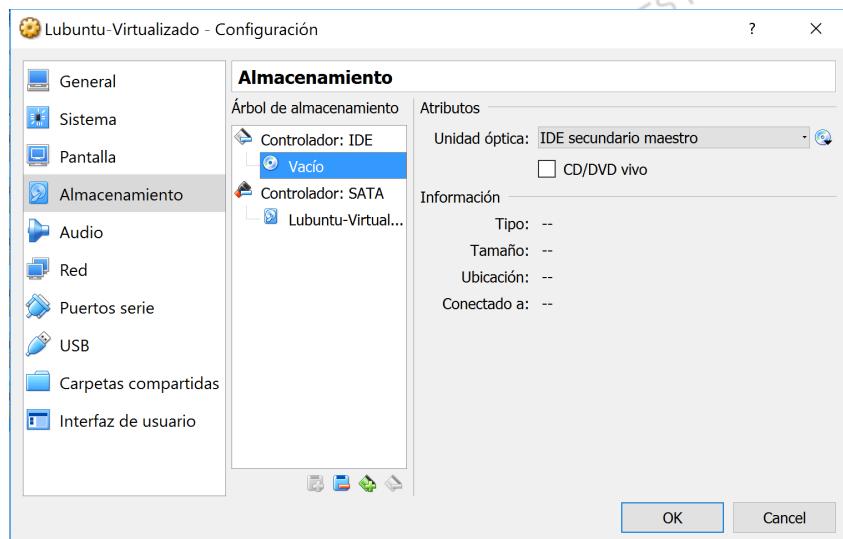


Figura 1.22. Configuración del controlador.

Selección del archivo

En el cuadro “Atributos”, se pulsa sobre el icono del “CD” y aparece una lista desplegable. Se selecciona “Seleccione archivo de disco óptico virtual...” (figura 1.23.).

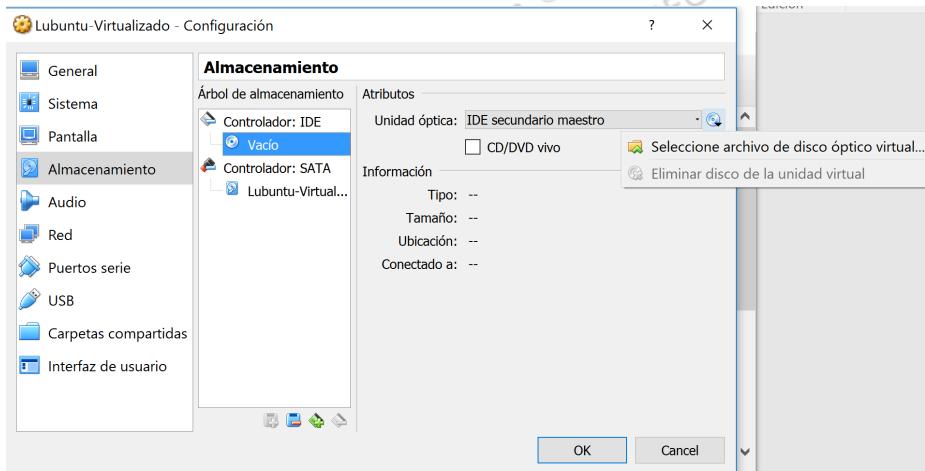


Figura 1.23. Selección del archivo.

Selección del archivo de Lubuntu

Aparece un navegador de archivos y se selecciona el archivo de “Lubuntu” que se ha descargado previamente (figura 1.24.).

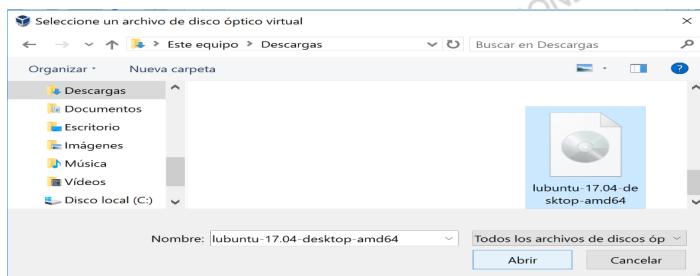


Figura 1.24. Selección del archivo de Lubuntu.

Pantalla resultante de la configuración

Tras realizar la selección, la pantalla aparece como en la figura 1.25.

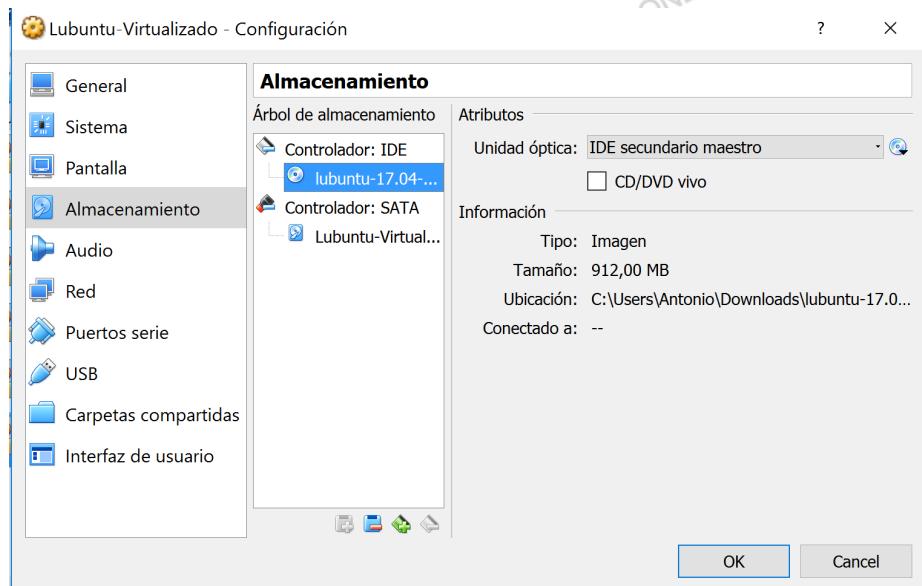


Figura 1.25. Pantalla resultante de la configuración.

Interface principal actualizada

A continuación, se pulsa sobre “OK” para volver a la interface principal de Virtual Box en la que se puede observar que se ha actualizado el cuadro almacenamiento (figura 1.26.).

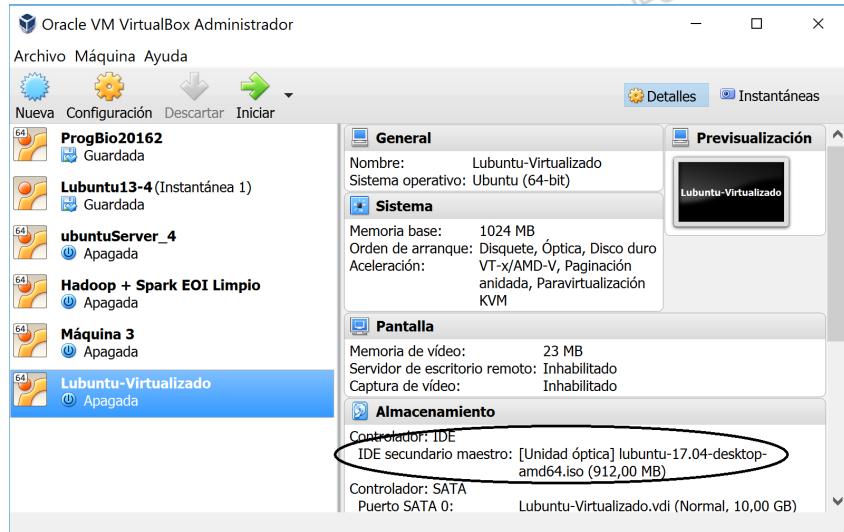


Figura 1.26. Interface principal actualizada.

Pantalla de instalación del sistema

Ahora, se va a crear el sistema virtualizado. Desde la pantalla principal, se pulsa sobre el icono de “Iniciar” para que se cargue el sistema. Aparecerán diferentes mensajes y pantallas de instalación (figura 1.27.). Son mensajes similares a los que se mostrarían si se estuviera instalando realmente el sistema operativo. Estos mensajes solo aparecerán la primera vez que se cargue el sistema virtualizado, no lo harán en las restantes ocasiones.

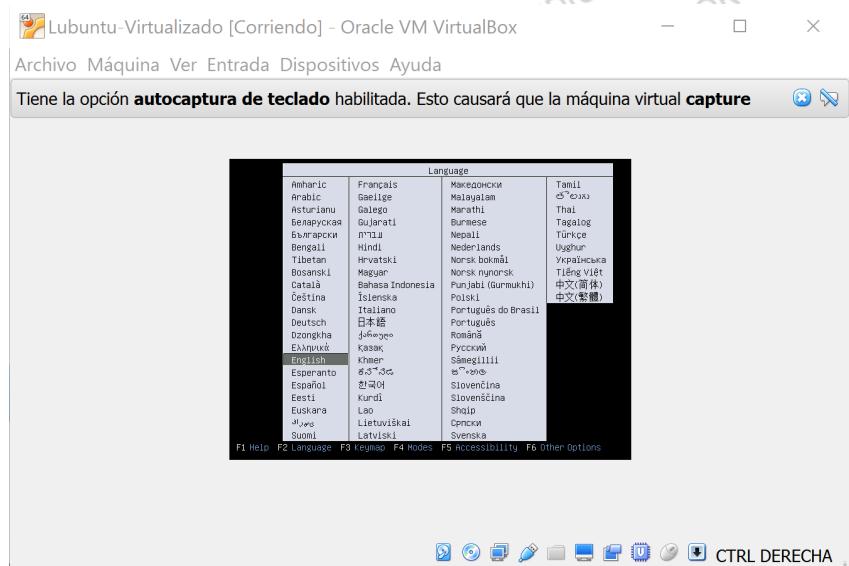


Figura 1.27. Pantalla de instalación del sistema.

Instalación de Lubuntu

Se selecciona “Instalar Lubuntu” (figura 1.28.).



Figura 1.28. Instalación de Lubuntu.

Selección del idioma

Se selecciona el idioma (figura 1.29.).

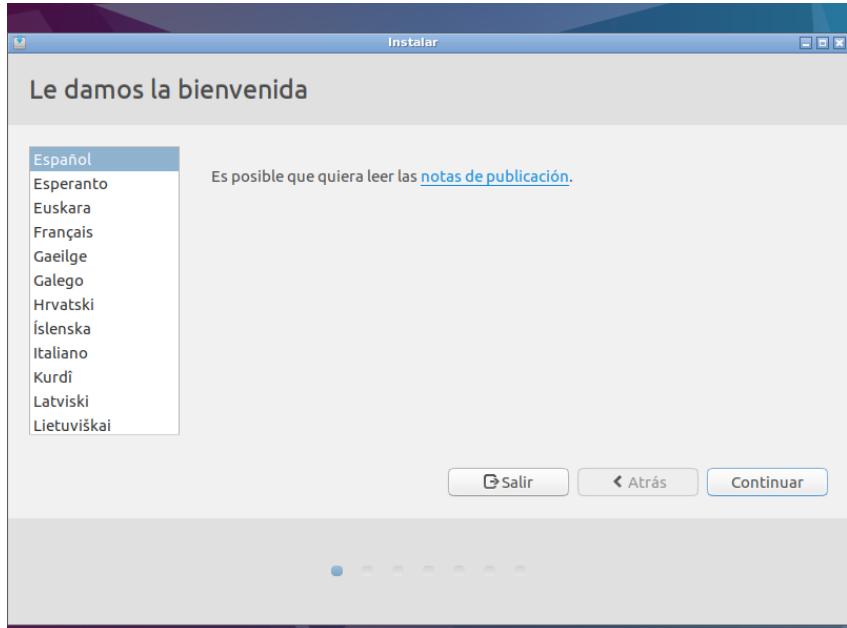


Figura 1.29. Selección del idioma.

Selección de software adicional

Se seleccionan paquetes de software adicional para su instalación (figura 1.30.).

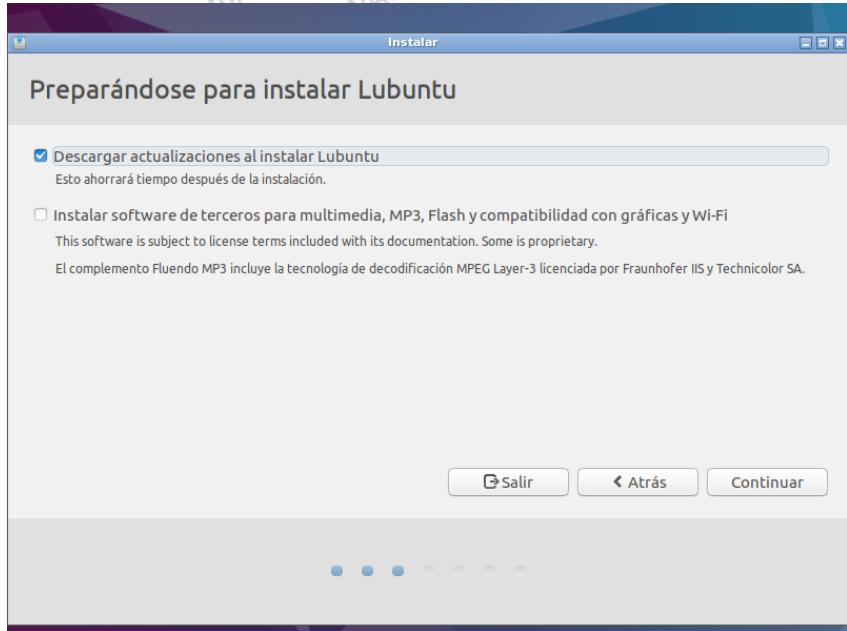


Figura 1.30. Selección de software adicional.

Tipo de instalación

Y el tipo de instalación (figura 1.31.).

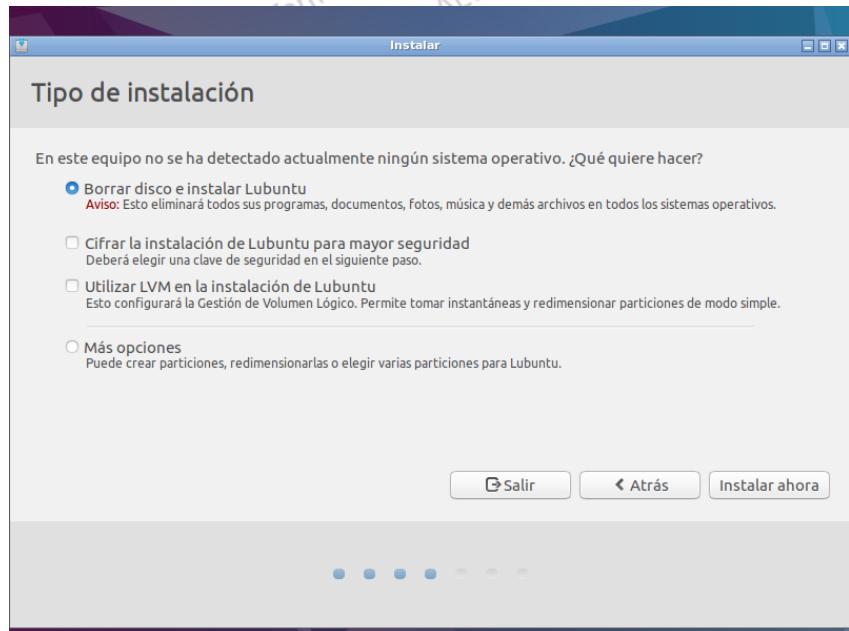


Figura 1.31. Tipo de instalación.

Pantalla consecutiva

Se pulsa sobre “Continuar” en la pantalla que aparece a continuación (figura 1.32.).

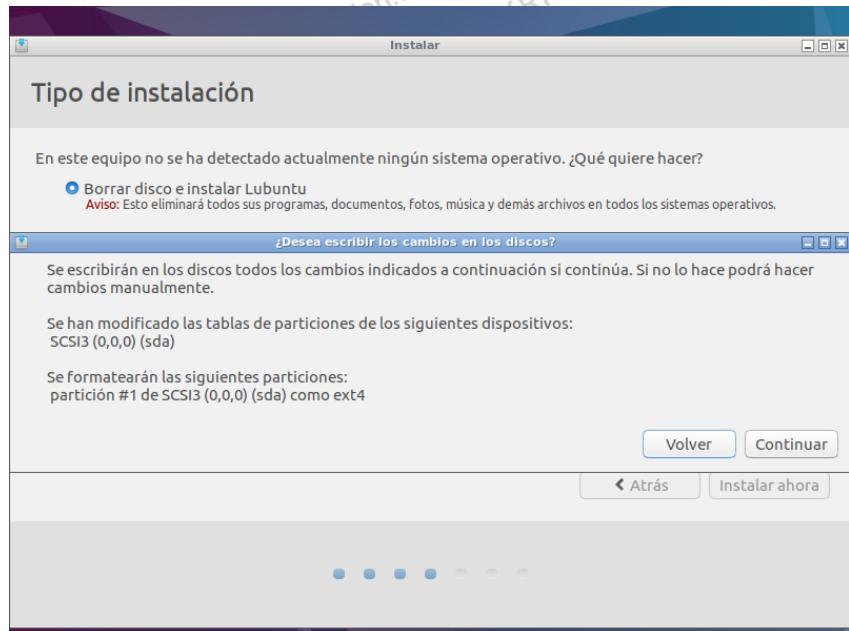


Figura 1.32. Pantalla consecutiva.

Pantalla de localización

Se selecciona nuestra localización (figura 1.33.).

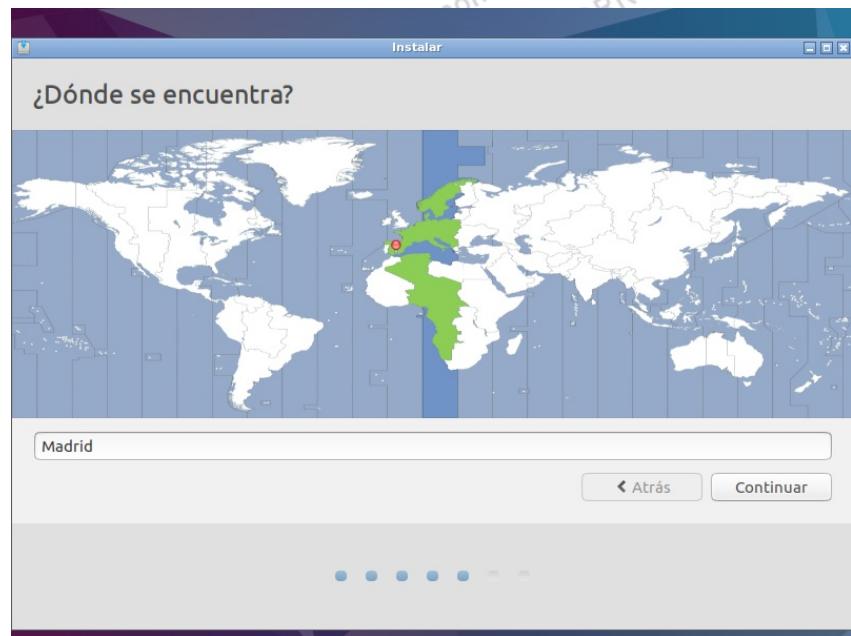


Figura 1.33. Pantalla de localización.

Disposición del teclado

Y la disposición del teclado (figura 1.34.).

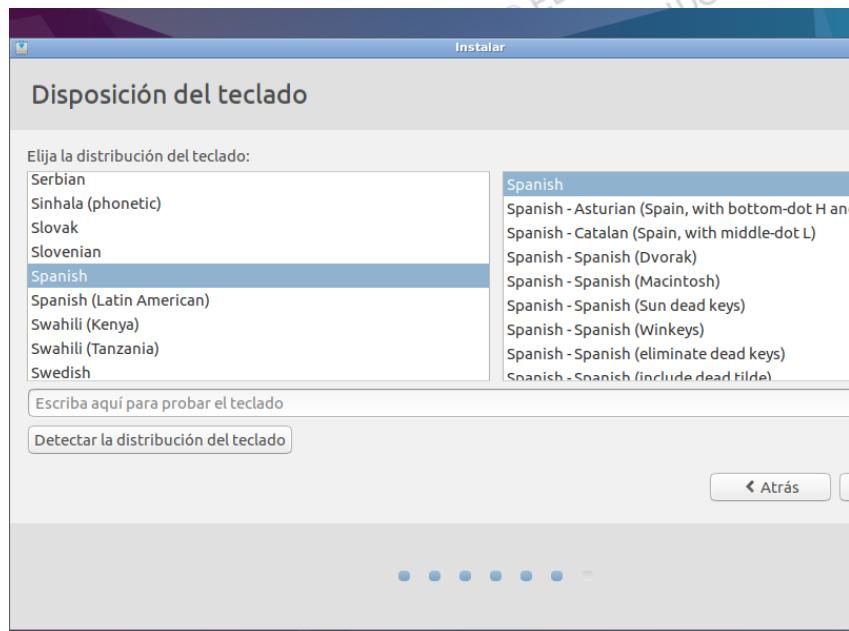


Figura 1.34. Disposición del teclado.

Configuración del usuario

Se configura el usuario (figura 1.35.).

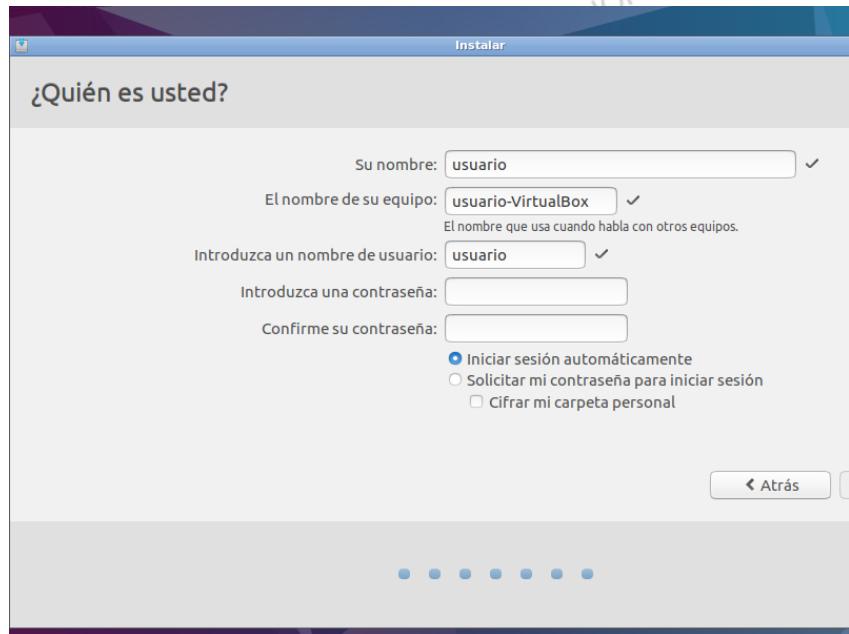


Figura 1.35. Configuración del usuario.

Instalación de Lubuntu

Se realiza la instalación de Lubuntu (figura 1.36.).

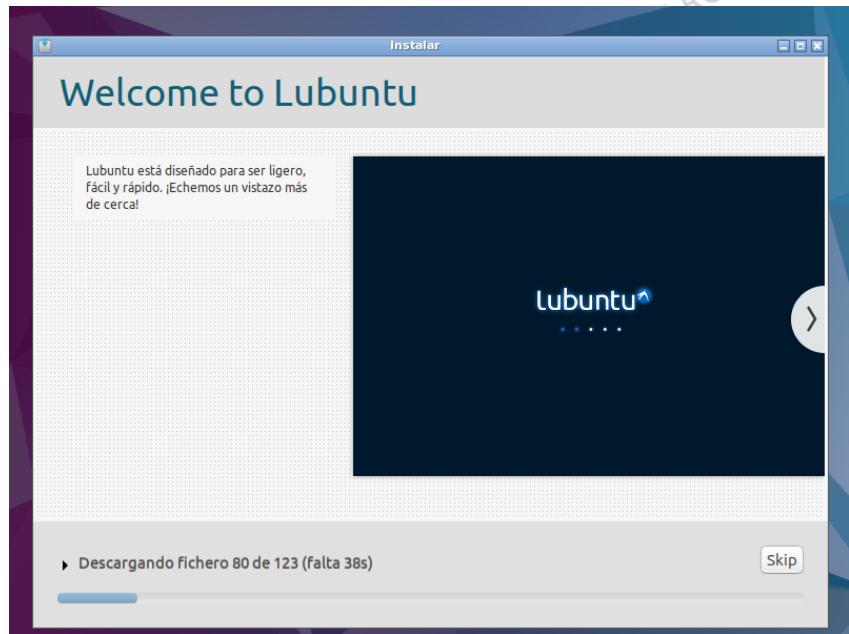


Figura 1.36. Instalación de Lubuntu.

Reinicio del sistema

Se completa la instalación y se pulsa sobre “Reiniciar ahora” (figura 1.37.).

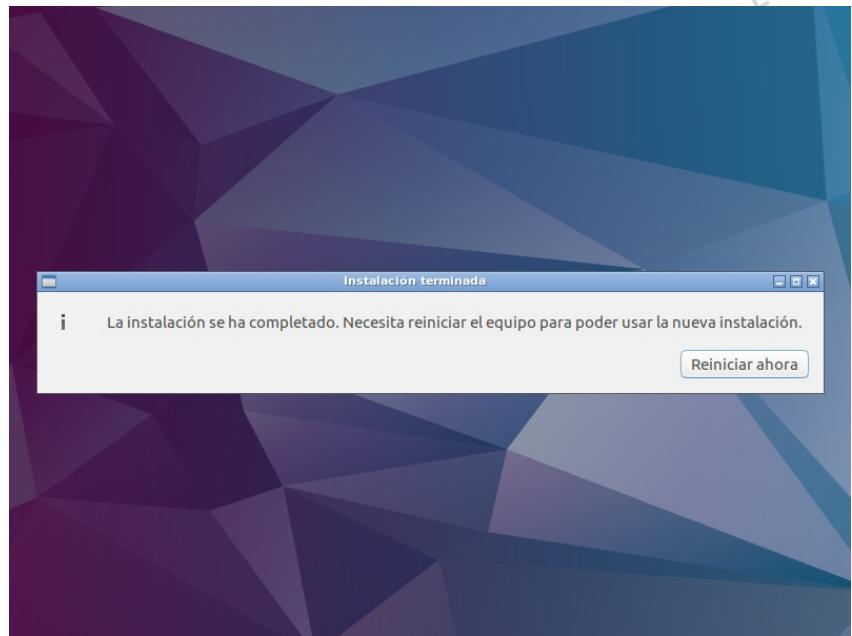


Figura 1.37. Reinicio del sistema.

Carga del sistema operativo

Se carga el sistema operativo (figura 1.38.).

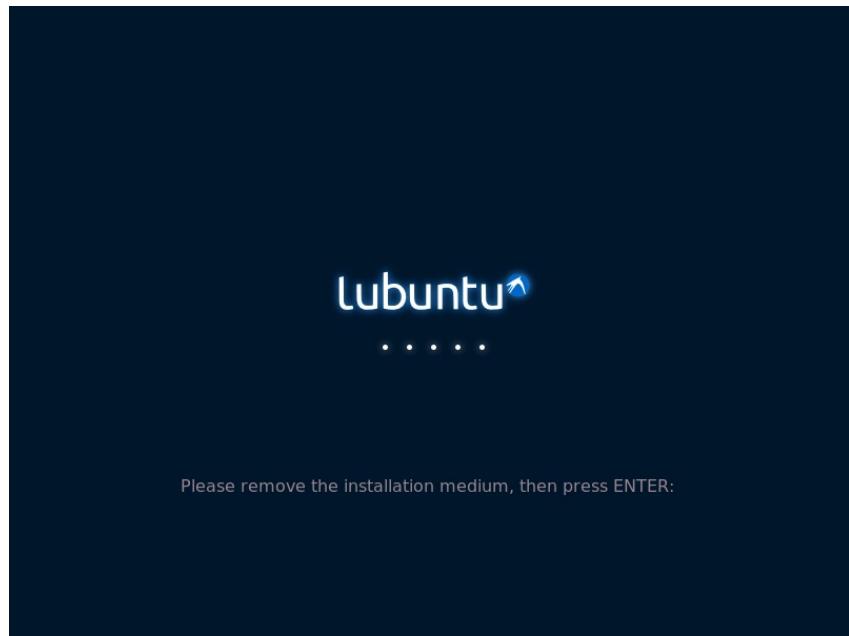


Figura 1.38. Carga del sistema operativo.

Máquina virtual cargada

En la figura 1.39., se puede observar el sistema operativo virtualizado.

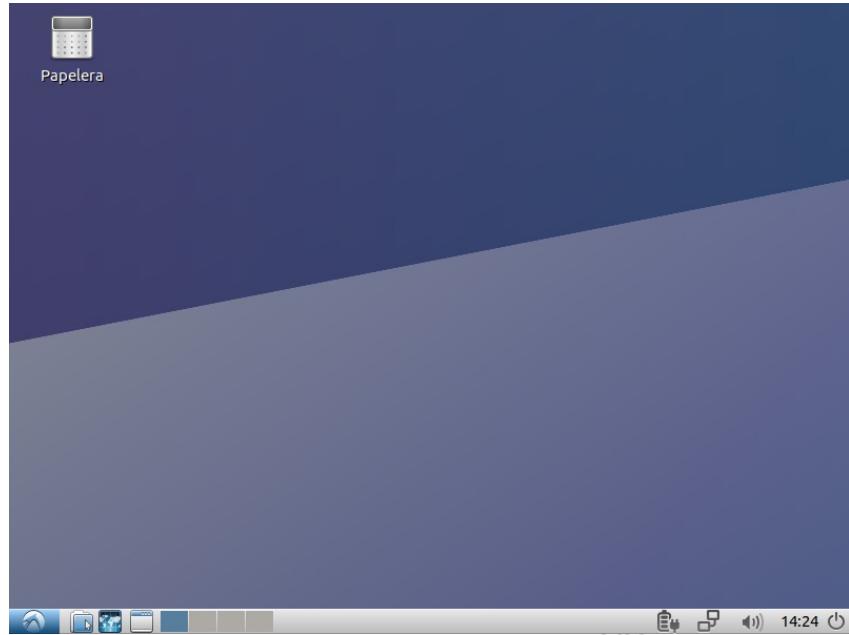


Figura 1.39. Máquina virtual cargada.

Opciones de cierre de la máquina virtual

Cuando se cierra la máquina virtual, aparecen varias opciones (figura 1.40.):

- Guardar el estado de la máquina: cierra la MV guardando el estado en el que se encuentra en ese momento, de modo que, al volver a iniciarla, continuará justo en el punto donde se cerró. No se pierde ningún tipo de información.
- Enviar señal de apagado: es el apagado ACPI.
- Apagar la máquina: equivale a cerrar la máquina sin guardar el estado, por lo que el SO no se cerrará correctamente y al iniciar la MV posteriormente, el SO hará los chequeos previstos cuando este no se cierre correctamente. Es posible que se pierdan datos, por lo que debe usarse como último recurso cuando la MV se ha bloqueado.

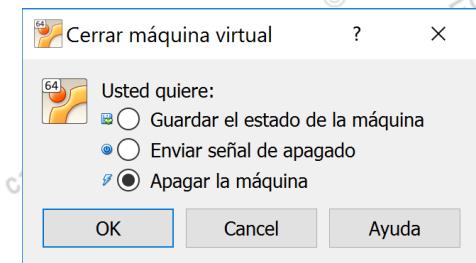


Figura 1.40. Opciones de cierre de la máquina virtual.

La próxima vez que se quiera ejecutar el sistema virtual bastará con seleccionar la máquina virtual y pulsar sobre “Iniciar” en la pantalla principal de Virtual Box.

V. Carga de una máquina virtual

Las máquinas virtuales son archivos que tienen el software de un sistema operativo virtualizado y permiten ejecutarlo en un sistema de virtualización. Físicamente son archivos que pueden tener varias extensiones, dependerán de la forma en la que se ha exportado la máquina virtual: ova, .vdi, etc. Hay repositorios gratuitos con máquinas virtuales ya preparadas para descargar e instalar:



- OSBoxes, <http://www.osboxes.org/>
- VirtualBoxes, <https://virtualboxes.org/>

Importación de una máquina virtual de tipo .ova

Descarga de una máquina virtual

Para ilustrar cómo cargar una máquina virtual, es posible descargar una de una distribución de Lubuntu desde el siguiente enlace (figura 1.41.):

- SourceForge, <https://sourceforge.net/projects/virtualimages/files/Lubuntu/>

Home / Browse / System Administration / OS distribution / Virtual Images / Files

Virtual Images
free virtual images for VirtualBox and VMWare
Brought to you by: vincentzac

Summary | **Files** | Reviews | Support

Home / Lubuntu

Name	Modified	Size	Downloads / Week
Lubuntu13-4.ova	2013-08-22	1.1 GB	1
Totals: 1 Item		1.1 GB	1

Figura 1.41. Descarga de una máquina virtual.

Importar una máquina virtual

Una vez descargada, se importa en Virtual Box. Para ello, se selecciona en el menú la opción: "Archivos">>"Importar Servicio Virtualizado" y se selecciona la máquina virtual que se ha descargado (figura 1.42.).

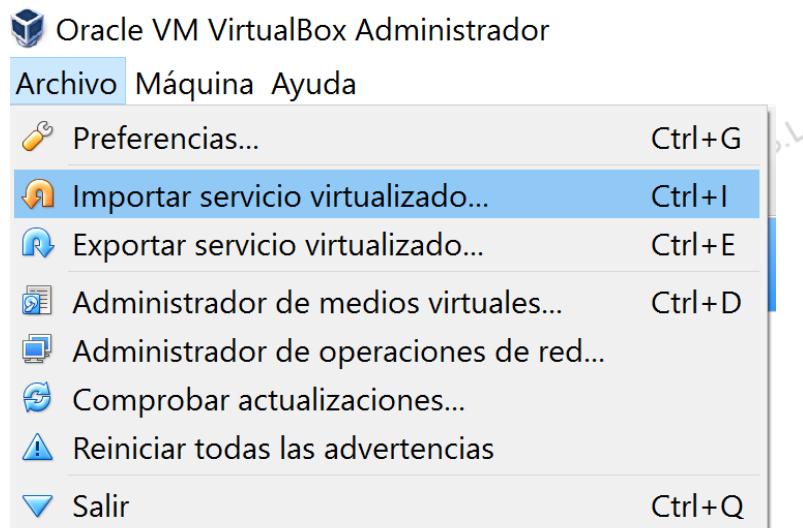


Figura 1.42. Importar una máquina virtual.

Selección de máquina virtual

A continuación, se selecciona la máquina virtual que se quiere importar (figura 1.43.).

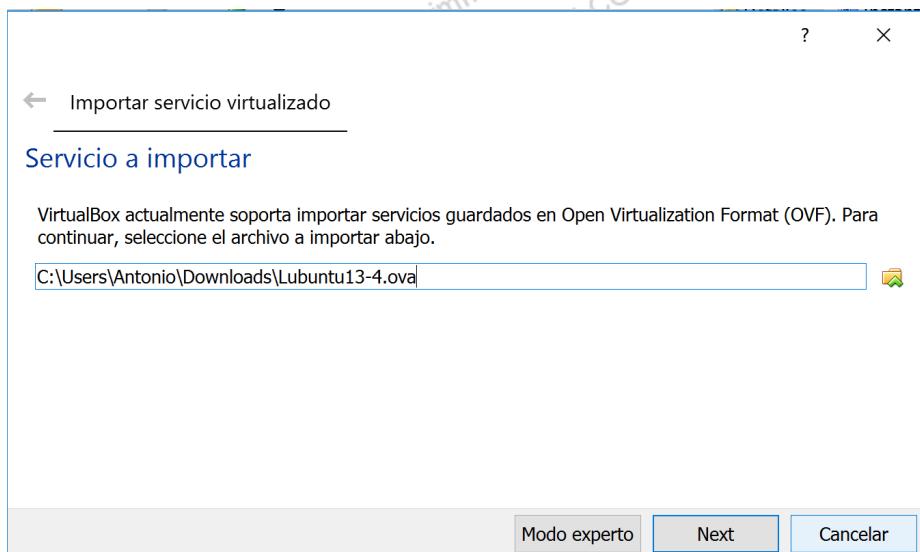


Figura 1.43. Selección de la máquina virtual a importar.

Configuración de la importación

A continuación, se pulsa sobre “Next” y aparecerá un resumen de la máquina virtual que se va a importar. Se debe marcar la opción “Reiniciar la dirección MAC de todas las tarjetas de red” y pulsar sobre “Importar” (figura 1.44.).

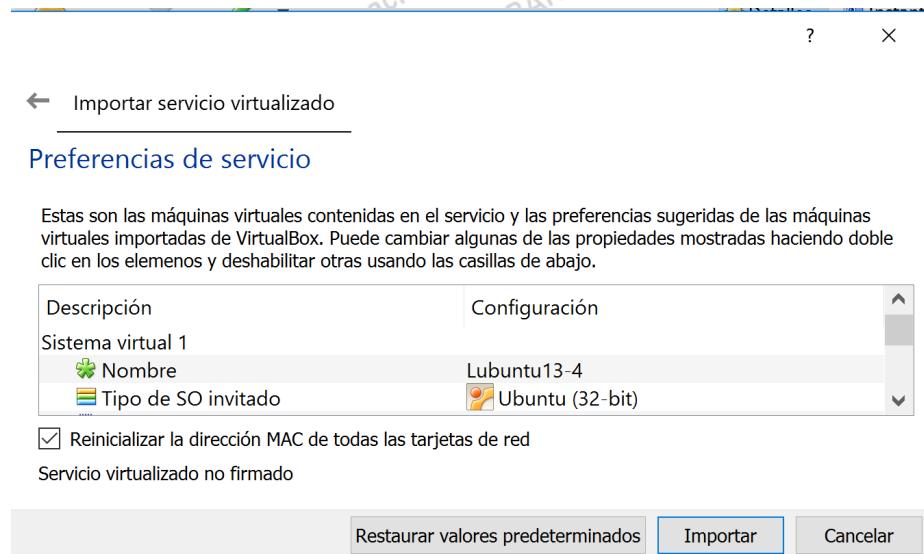


Figura 1.44. Configuración de la importación.

Importación de la máquina virtual

Se muestra una ventana con el estado del proceso de importación. El tiempo varía dependiendo de las características de la máquina (figura 1.45.).

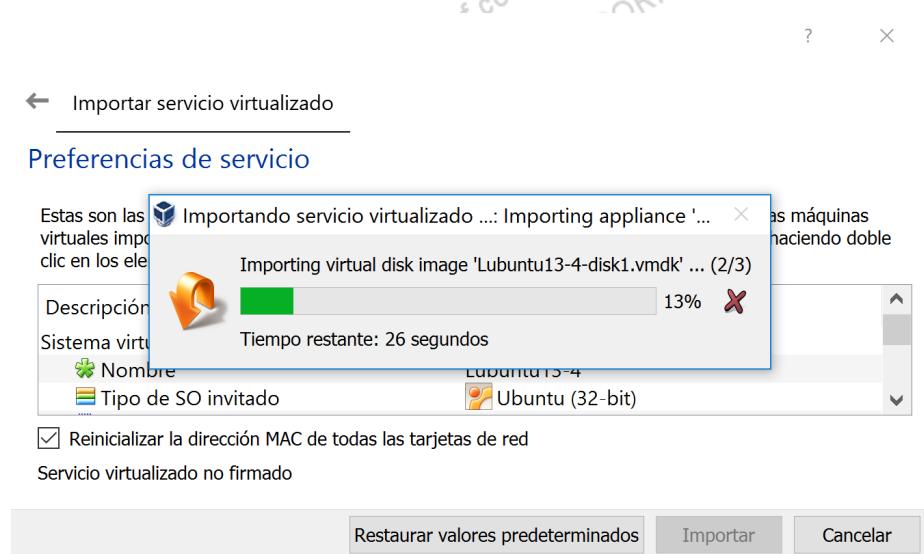


Figura 1.45. Importación de la máquina virtual.

Máquina virtual importada

Una vez importada la máquina virtual, debería aparecer en la parte de la izquierda de la pantalla una nueva máquina virtual de nombre Lubuntu13-4 (figura 1.46.).

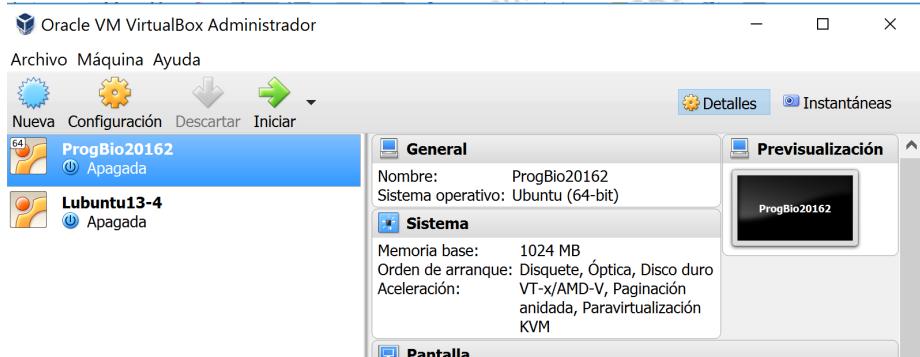


Figura 1.46. Máquina virtual importada.

Configuración de la máquina virtual importada

Se pulsa sobre Lubuntu 13-4 y se selecciona la opción de “Configuración” (figura 1.47.).

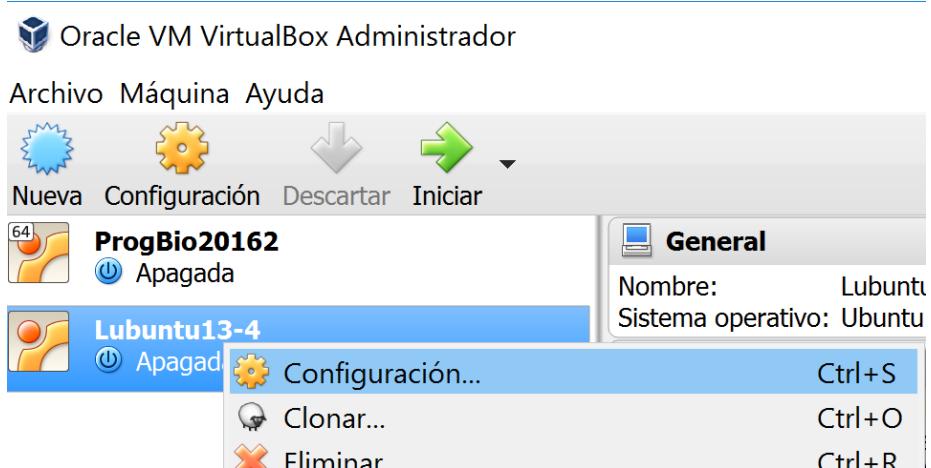


Figura 1.47. Configuración de la máquina virtual importada.

Configuración de la memoria RAM

A continuación, se selecciona en “Sistema” la opción “Placa Base” para fijar la cantidad de memoria de la máquina virtual (figura 1.48.). Dependiendo de la memoria RAM que tenga el ordenador que se está usando, se puede aumentar o disminuir el valor. 512 Mb deberían ser suficientes, pero se recomiendan 1024 Mb para que esta funcione con fluidez.

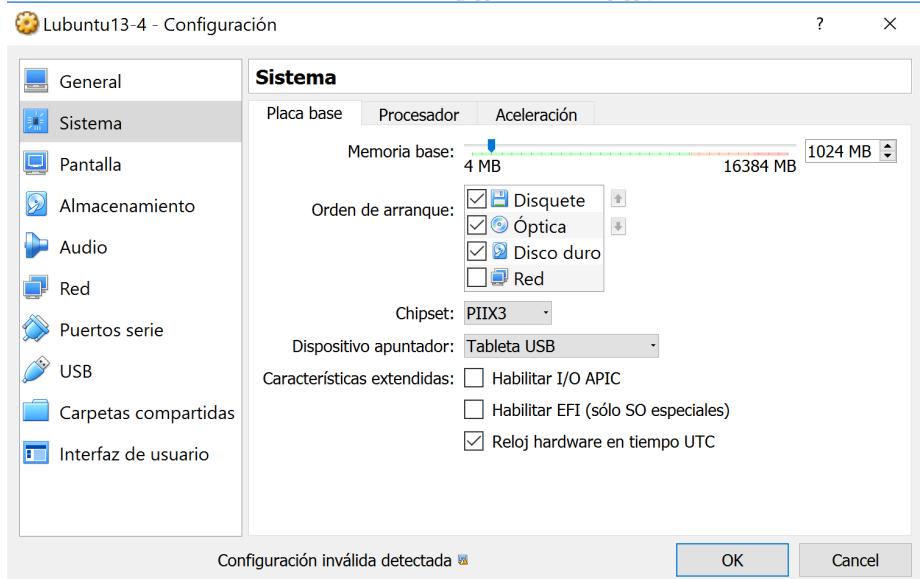


Figura 1.48. Configuración de la memoria RAM.

Configuración de la tarjeta gráfica

Se selecciona la opción “Pantalla” y, en esta pestaña, se ajusta la cantidad de memoria para la tarjeta gráfica: 32 Mb deberían ser suficientes (figura 1.49.).

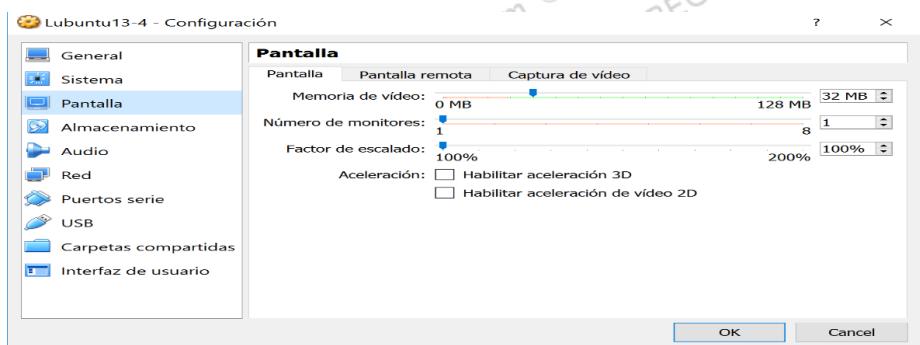


Figura 1.49. Configuración de la memoria para la tarjeta gráfica.

Inicio de la máquina virtual

A continuación, se pulsa en el botón “OK”. En la pantalla principal de Virtual Box, se selecciona la máquina virtual “Lubuntu 13-4” y se pulsa en el botón “Iniciar” (figura 1.50.).

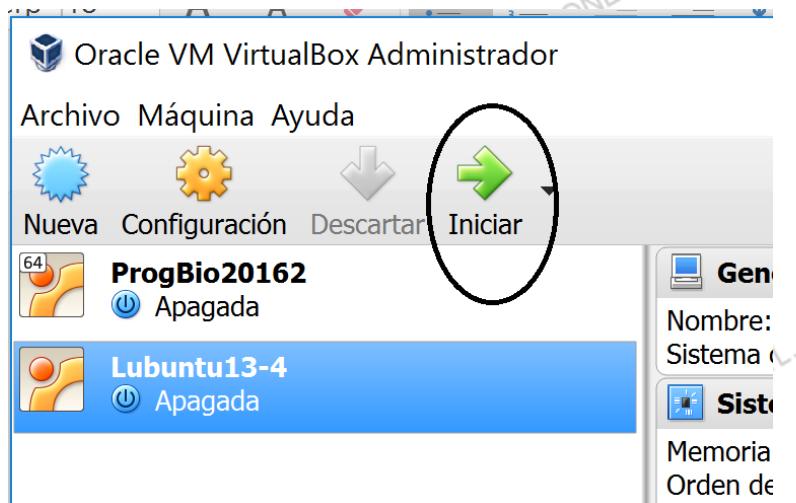


Figura 1.50. Inicio de la máquina virtual importada.

Pantalla de login

A continuación, se ejecuta la máquina virtual “Lubuntu 13-4” y ya se podrá trabajar con ella. Se selecciona la cuenta “Guest Account” y se pulsa sobre “Login” (figura 1.51).

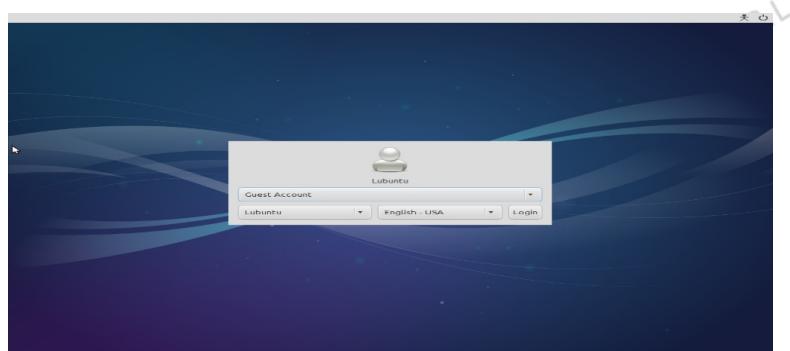


Figura 1.51. Login en máquina virtual importada.

Escritorio de la máquina virtual importada

A continuación, aparece el escritorio del sistema operativo virtualizado (figura 1.52.).

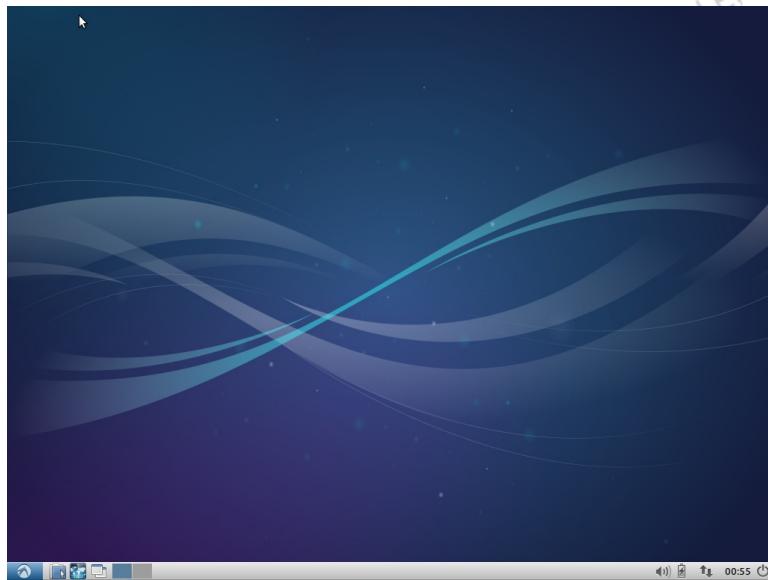


Figura 1.52. Escritorio de la máquina virtual importada.

Observaciones importantes

Algunas observaciones que se deben tener en cuenta:

1. Para que funcionen las máquinas virtuales es necesario que el PC permita la "Virtualización Hardware". Esta es una opción que suele encontrarse en la BIOS del equipo. Puede llamarse "Virtualización Hardware" o similar en función del modelo del PC.
2. Para confirmar que el PC soportará la virtualización, se puede usar la siguiente página: <https://www.grc.com/securable.htm>
3. Si la importación da un error estilo ERROR_ZIP_CODE, muy posiblemente el archivo .ova esté corrupto. Para solucionarlo, se debe descargar de nuevo la máquina virtual.
4. Una vez importada la máquina, si al intentar encenderla no lo hiciera, se debe revisar el punto 1.
5. También es importante que la máquina virtual esté preparada para el sistema operativo anfitrión, es decir, saber si es de 64 bits o 32 bits.

Se acaba de describir cómo cargar una máquina virtual mediante la opción de "Importar un servicio virtualizado", puesto que la máquina se encontraba en un archivo de tipo .ova. Sin embargo, las máquinas virtuales pueden distribuirse en otros formatos diferentes

Importación de máquina virtual de tipo .vmdk

Importación de una máquina virtual tipo .vmdk

A continuación, se muestra cómo importar una máquina virtual que se encuentra en un archivo de tipo .vmdk. Para ello, se pulsa en la opción “Nueva” (figura 1.53.).

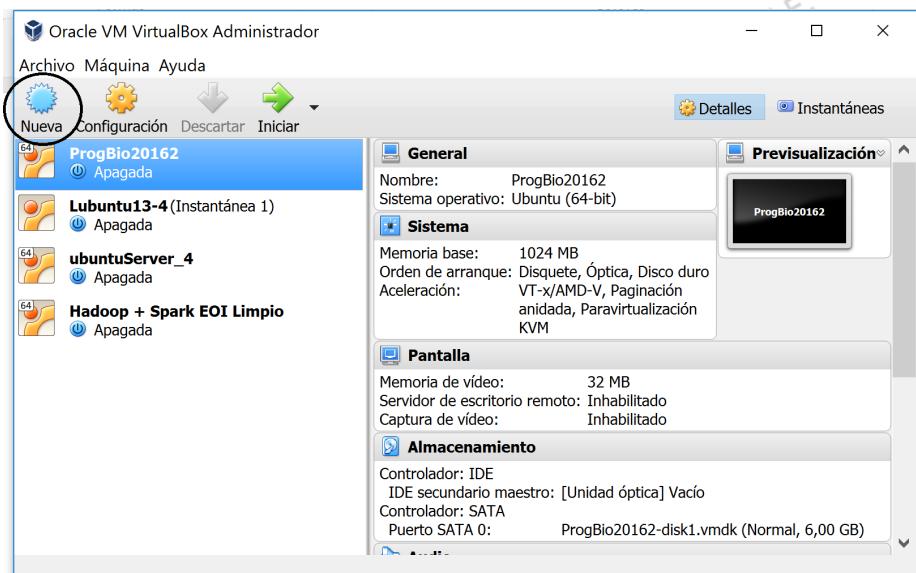


Figura 1.53. Importación de una máquina virtual tipo .vmdk.

Nombre de la máquina virtual

Aparece una nueva ventana en la que se debe configurar la máquina virtual. Se rellena la pestaña “Nombre” con uno cualquiera (por ejemplo, “Máquina 3”), en “Tipo” se selecciona “Linux” y en “Versión” se selecciona “Ubuntu (64-bit)”, tal como se ve en la figura 1.54.

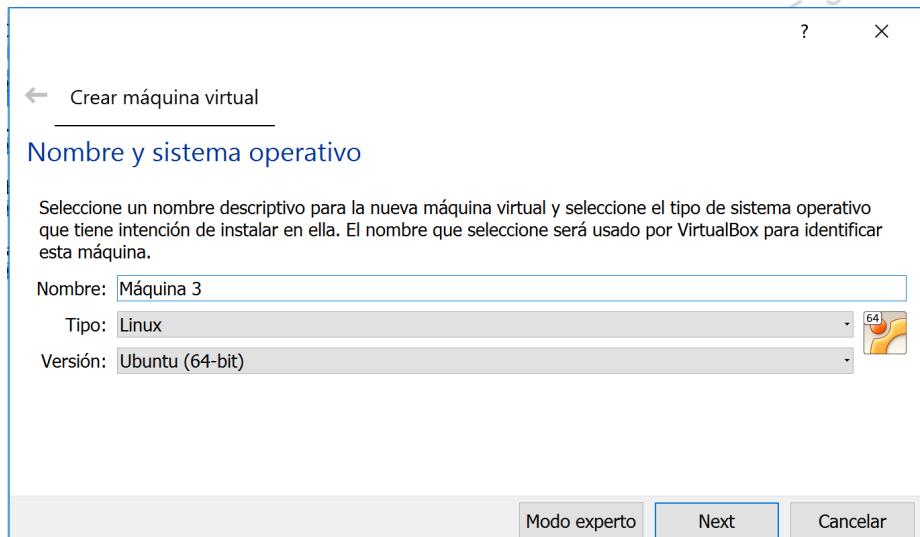


Figura 1.54. Nombre de la máquina virtual.

Configuración de la memoria

Se pulsa sobre “Next” y, en la ventana siguiente, se configura el tamaño de la memoria. 1024 Mb es un buen valor (figura 1.55.).

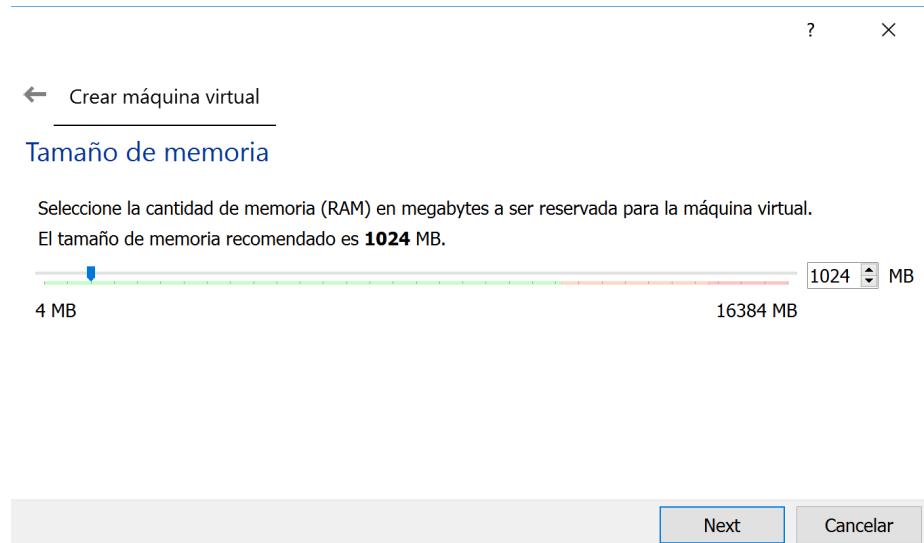


Figura 1.55. Configuración de la memoria.

Selección de opciones

A continuación, se pulsa sobre “Next” y en la ventana siguiente se selecciona la opción “Usar un archivo de disco duro virtual existente” y se pulsa sobre el ícono en forma de carpeta (figura 1.56.).

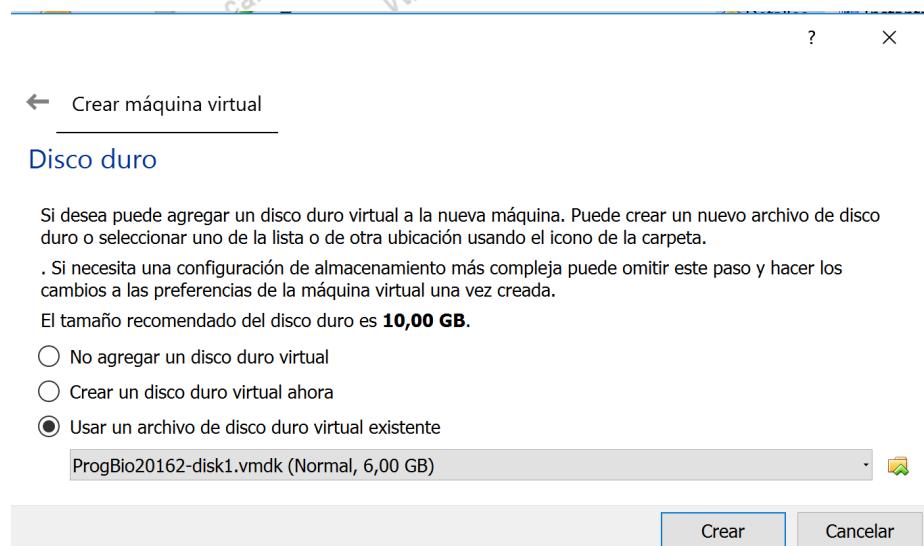


Figura 1.56. Selecciona "Usar disco ya existente".

Seleccionar la máquina virtual

Cuando se pulsa sobre el ícono de la carpeta, aparece un navegador de archivos, en el que se buscará la máquina que se quiere cargar (figura 1.57.).

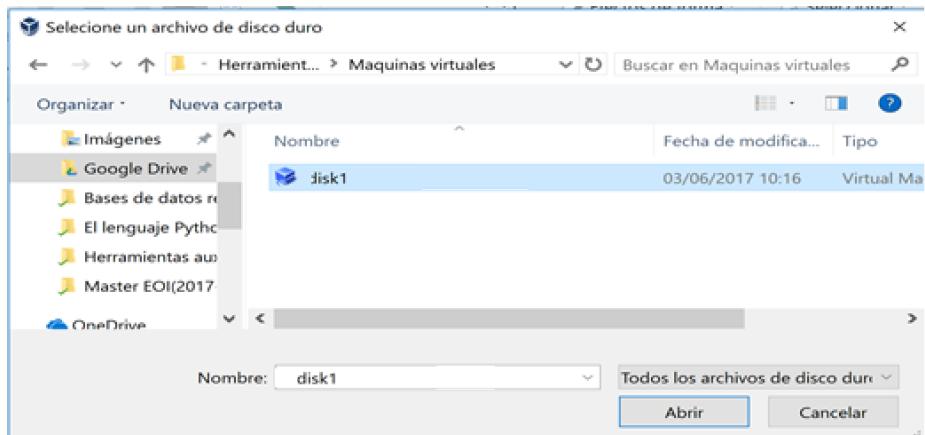


Figura 1.57. Búsqueda de la máquina virtual.

Máquina virtual seleccionada

Se pulsa sobre “Abrir” y se vuelve a la ventana anterior (figura 1.58.).

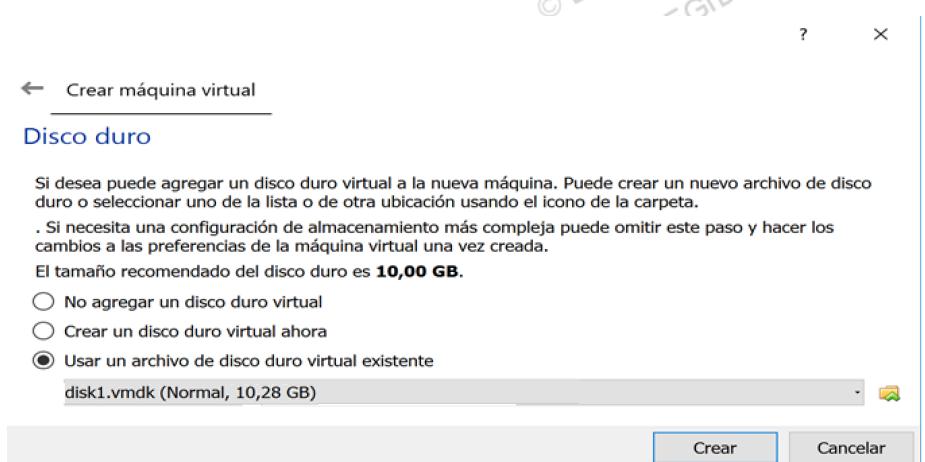


Figura 1.58. Máquina virtual seleccionada.

Máquina virtual importada

A continuación, se pulsa sobre “Crear” y aparecerá en el listado de máquinas virtuales (figura 1.59.).



Figura 1.59. Máquina virtual importada.

Tutorial de descarga

[Tutorial para descarga e importación de máquinas virtuales VMDK](#)

VI. La shell de comandos de Linux. Creación de scripts

6.1. La shell de comandos

Linux es un sistema operativo similar a Unix que se distribuye como software libre.



Para ilustrar los contenidos de esta sección, se va utilizar una distribución de Linux denominada Lubuntu que se caracteriza por exigir pocos recursos hardware para su ejecución. Linux dispone de una aplicación denominada intérprete de comandos o Shell de comandos que permite interactuar al usuario con el sistema operativo mediante la ejecución de comandos o sentencias.

Acceso

Hay diferentes tipos de intérpretes que se diferencian en la sintaxis de los comandos y en la forma de interaccionar con el usuario. Para acceder al intérprete de comandos, se pulsa sobre Ctrl+Alt+T o bien se pulsa en un ícono que aparece en la parte inferior izquierda, dentro de la barra de herramientas inferior del escritorio de Lubuntu (figura 1.60.).

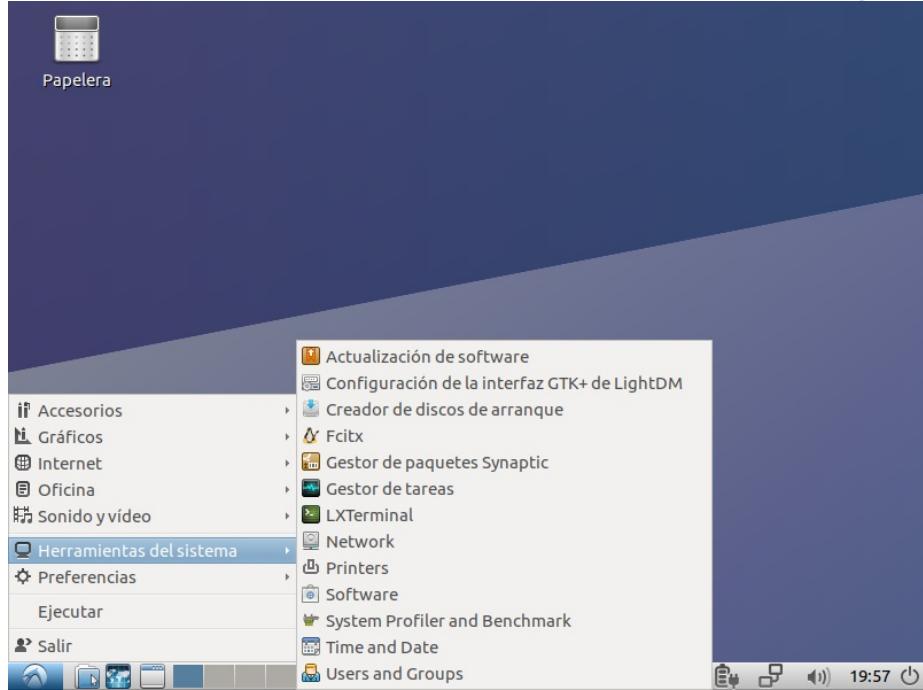


Figura 1.60. Acceso al intérprete de comandos de Linux.

Características generales

Algunas características generales son:

- El terminal muestra en pantalla un indicador de línea de órdenes esperando que el usuario introduzca una. El indicador finaliza con un carácter \$ en el caso de usuarios normales y con # en el caso del superusuario.
- Al comienzo de la línea de órdenes aparece el usuario de la forma usuario@nombre: ~\$ (figura 1.61.)
- Cuando se escribe un comando para que se ejecute, hay que pulsar la tecla “Enter”.
- Los comandos hay que teclearlos exactamente. En este sentido, las letras mayúsculas y minúsculas se consideran caracteres diferentes.
- El terminal siempre está dentro de una carpeta específica y puede navegar hasta otras carpetas y gestionar los archivos. Siempre se abre desde la carpeta personal del usuario.

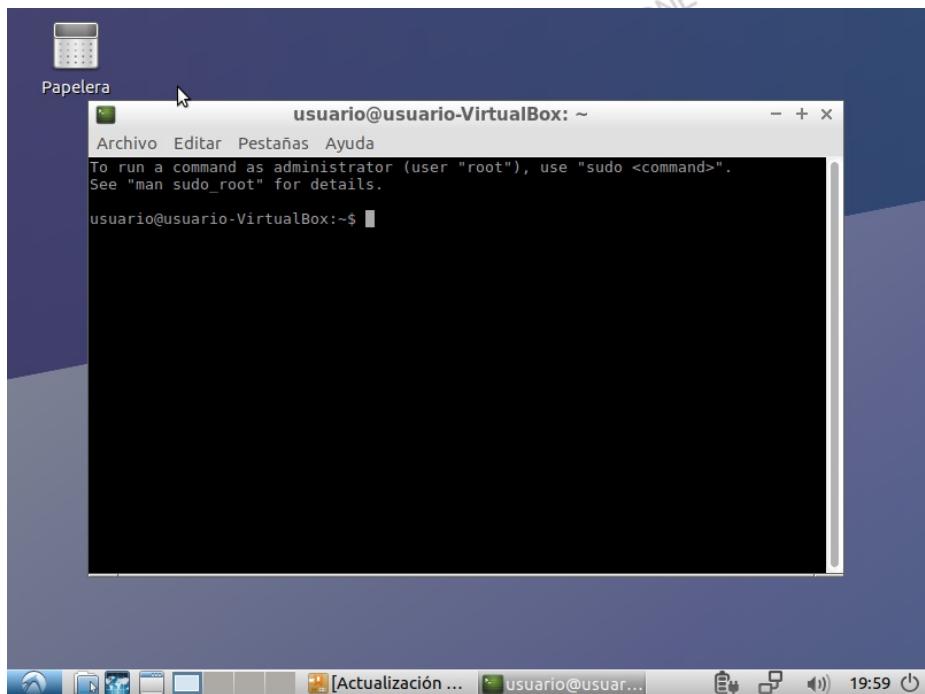


Figura 1.61. Intérprete de comandos Linux.

Comandos principales

A continuación, se van a revisar los principales comandos:

Cambio de directorio

Para cambiar de directorio se usa el comando **cd directorio_destino**. En particular para ir al directorio superior se usa **cd ..** (figura 1.62.) y **cd** sin argumentos para volver a la carpeta personal.

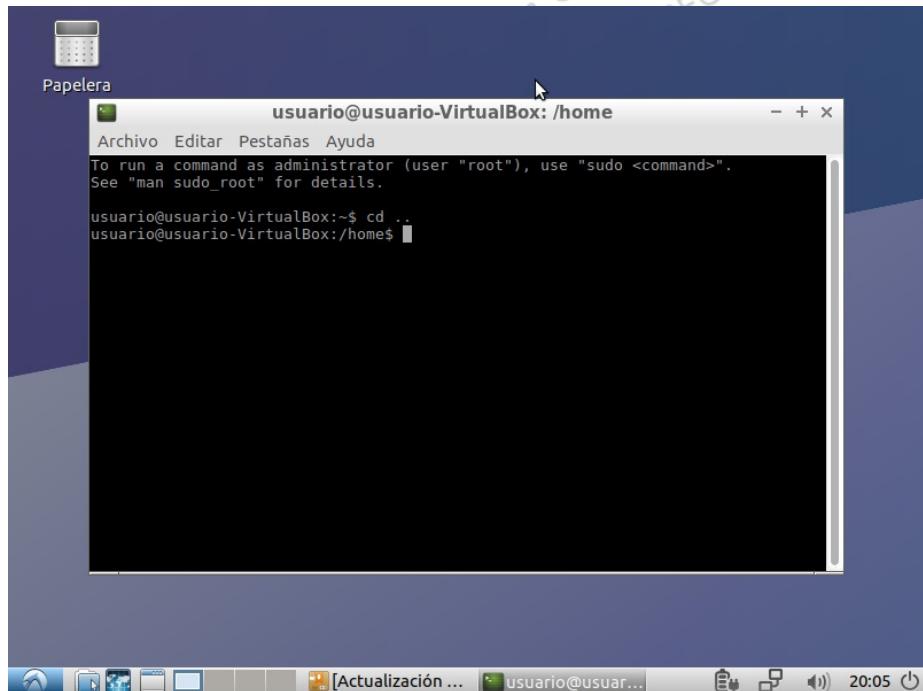


Figura 1.62. Cambio de directorio.

Situación actual

El comando **pwd** imprime la ruta del directorio actual (figura 1.63.).

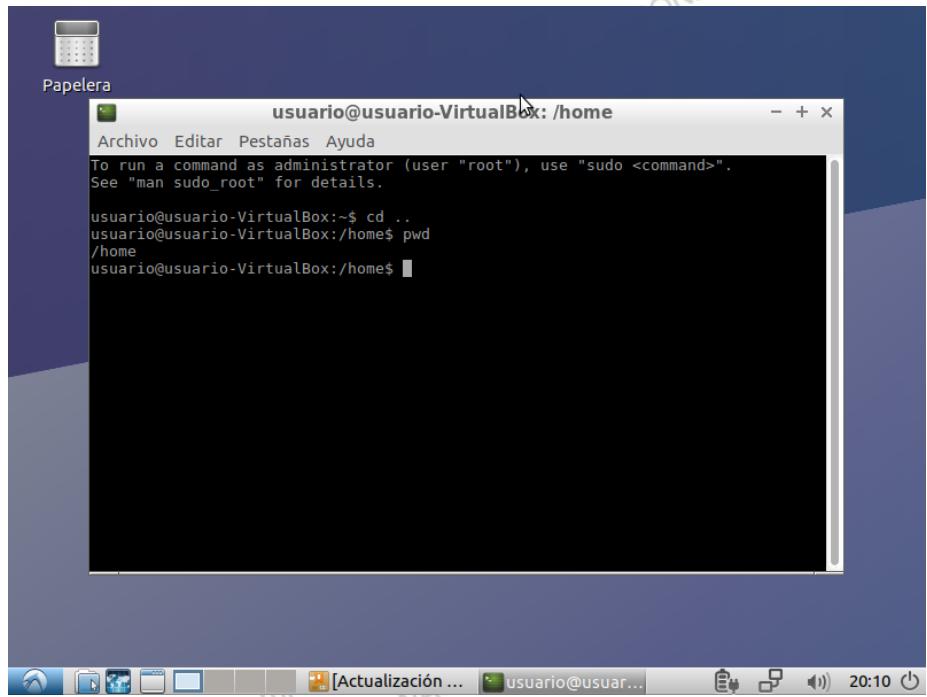


Figura 1.63. Uso del comando **pwd**.

Listado del contenido de un directorio

El comando **ls** muestra los nombres de los archivos y subdirectorios contenidos en el directorio en el que se está (figura 1.64.).

Uso básico

Solo se obtienen los nombres de los archivos, sin ninguna otra información.

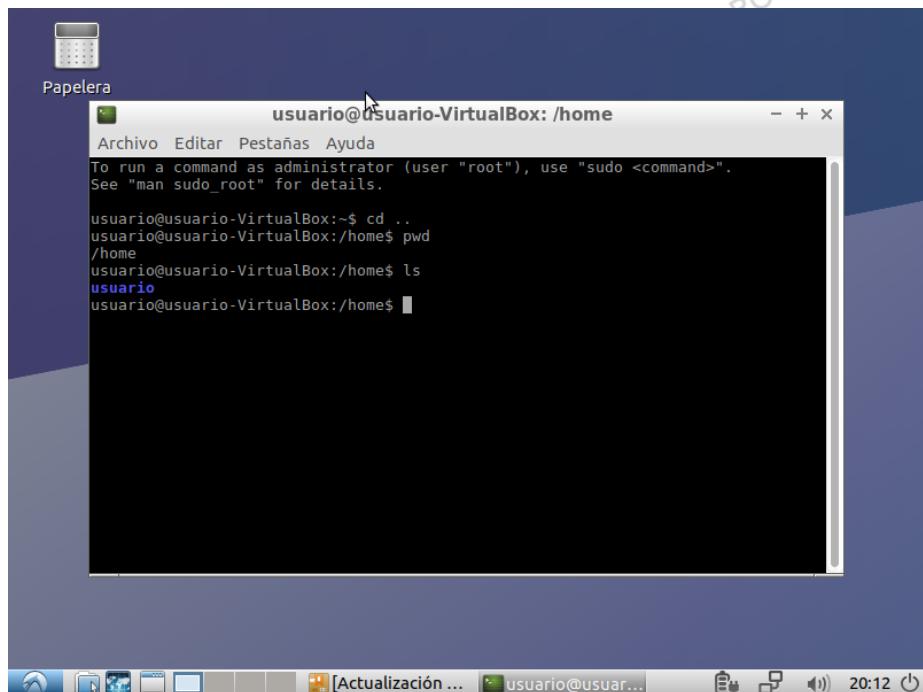


Figura 1.64. Uso del comando ls.

Conjunto de opciones

Admite un conjunto de opciones:

- **ls -R** lista recursivamente los subdirectorios encontrados.
- **ls -a** muestra todos los archivos, incluyendo algunos que están ocultos para el usuario —aquejellos que comienzan por un punto—. El archivo punto . indica el directorio actual y el doble punto .. el directorio padre, que contiene al actual.
- **ls -c** muestra ordenando por día y hora de creación.
- **ls -t** muestra ordenando por día y hora de modificación.
- **ls -r** muestra el directorio y lo ordena en orden inverso.
- **ls subdir** muestra el contenido del subdirectorio subdir.
- **ls -l filename** muestra toda la información sobre el archivo.
- **ls -color** muestra el contenido del directorio coloreado: verde para los ejecutables, azul para las carpetas, fucsia para las imágenes, rojo para los comprimidos, etc.
- **ls -l** muestra toda la información de cada archivo, incluyendo protecciones, tamaño y fecha de creación o del último cambio introducido, etc.

Combinación de opciones

Las opciones anteriores pueden combinarse. Por ejemplo, **ls -cr** muestra el directorio ordenando inversamente por fechas (figura 1.65.).

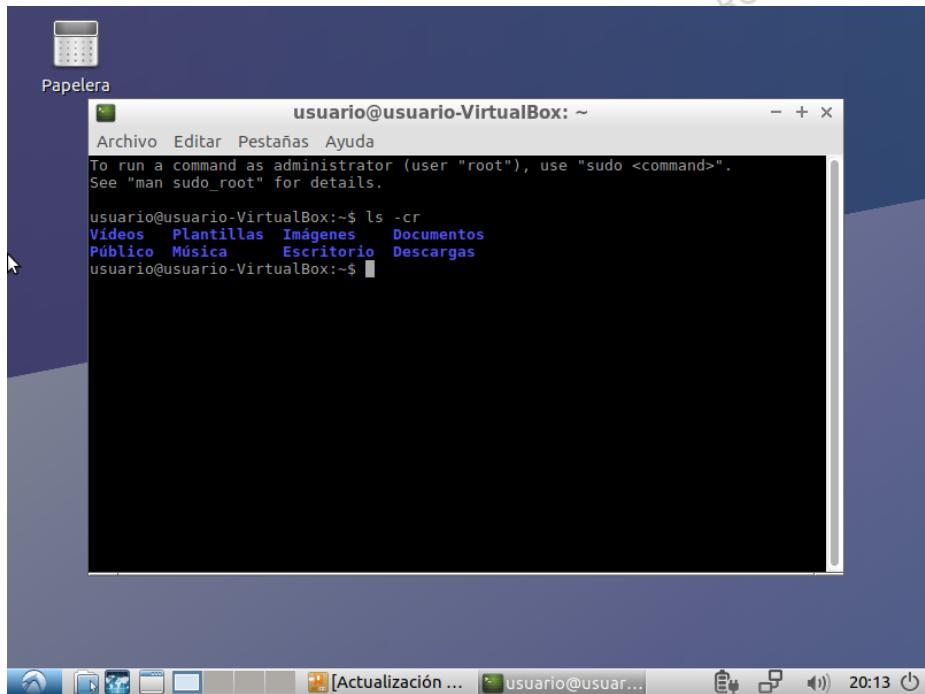


Figura 1.65. Uso del comando ls con varias opciones.

El comando **ls** admite los caracteres de sustitución * que representa cualquier conjunto o secuencia de caracteres y ? que representa cualquier carácter, pero solo uno.



Por ejemplo, **ls *.gif** muestra todos los nombres de archivos que acaben en .gif y **ls file?** muestra todos los archivos cuyos nombres empiecen por file y tengan un nombre de cinco caracteres.

Existe otro comando denominado **dir** que tiene la misma función que **ls**, pero sin mostrar tanta información.

Creación de ficheros y creación de subdirectorios

- Creación de ficheros:

El comando **touch** actualiza los registros de fecha y hora con la fecha y hora actual de los ficheros indicados como argumento. Si el fichero no existe, el comando touch lo crea. Su uso más frecuente es para crear archivos.

La sintaxis del comando touch sigue la forma: **touch fichero**

Además, **touch ejemplo.txt** crea el archivo ejemplo.txt en el directorio actual, si este no existiera.

- Creación de subdirectorios:

El comando **mkdir** permite crear un nuevo subdirectorio. La sintaxis es **mkdir subdir1** donde subdir es el nombre del directorio que se va a crear (figura 1.66.).

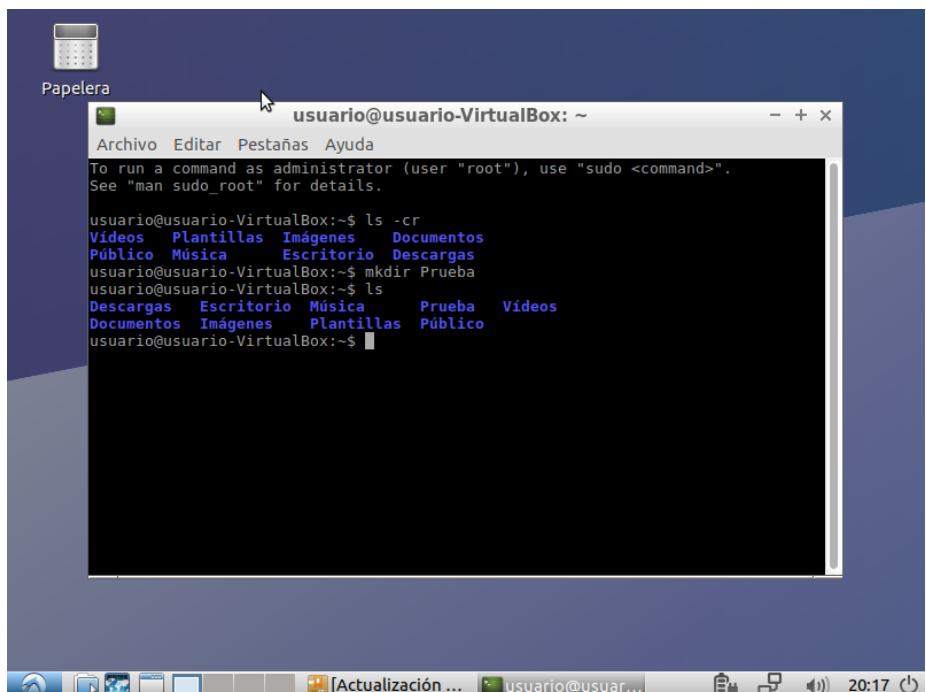
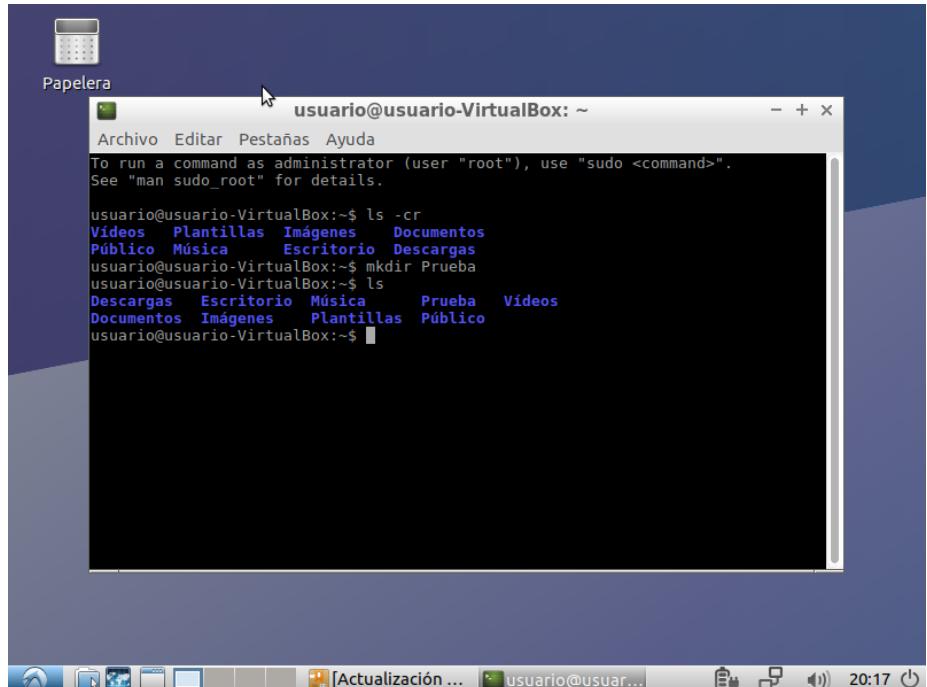


Figura 1.66. Creación de un subdirectorio.

Borrado de subdirectorios

El comando borra uno o más directorios del sistema siempre que estos subdirectorios estén vacíos. La sintaxis es **rmdir subdir1** donde subdir es el nombre del directorio que se va a eliminar (figura 1.67.).



The screenshot shows a terminal window titled "usuario@usuario-VirtualBox: ~". The window contains the following text:

```
Archivo Editar Pestañas Ayuda
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

usuario@usuario-VirtualBox:~$ ls -cr
Vídeos Plantillas Imágenes Documentos
Público Música Escritorio Descargas
usuario@usuario-VirtualBox:~$ mkdir Prueba
usuario@usuario-VirtualBox:~$ ls
Descargas Escritorio Música Prueba Vídeos
Documentos Imágenes Plantillas Públ...
usuario@usuario-VirtualBox:~$
```

Figura 1.67. Borrado de un subdirectorio.

Copia de archivos

La sintaxis del comando es **cp file1 file2** (figura 1.68.) donde file1 y file2 indican el nombre de un archivo o la ruta al mismo. Su ejecución hace una copia de file1 y lo llama file2. Si file2 no existía, lo crea con los mismos atributos de file1 y en caso de existir su contenido es sustituido por el de file1. El archivo file2 estará en el mismo directorio que file1.

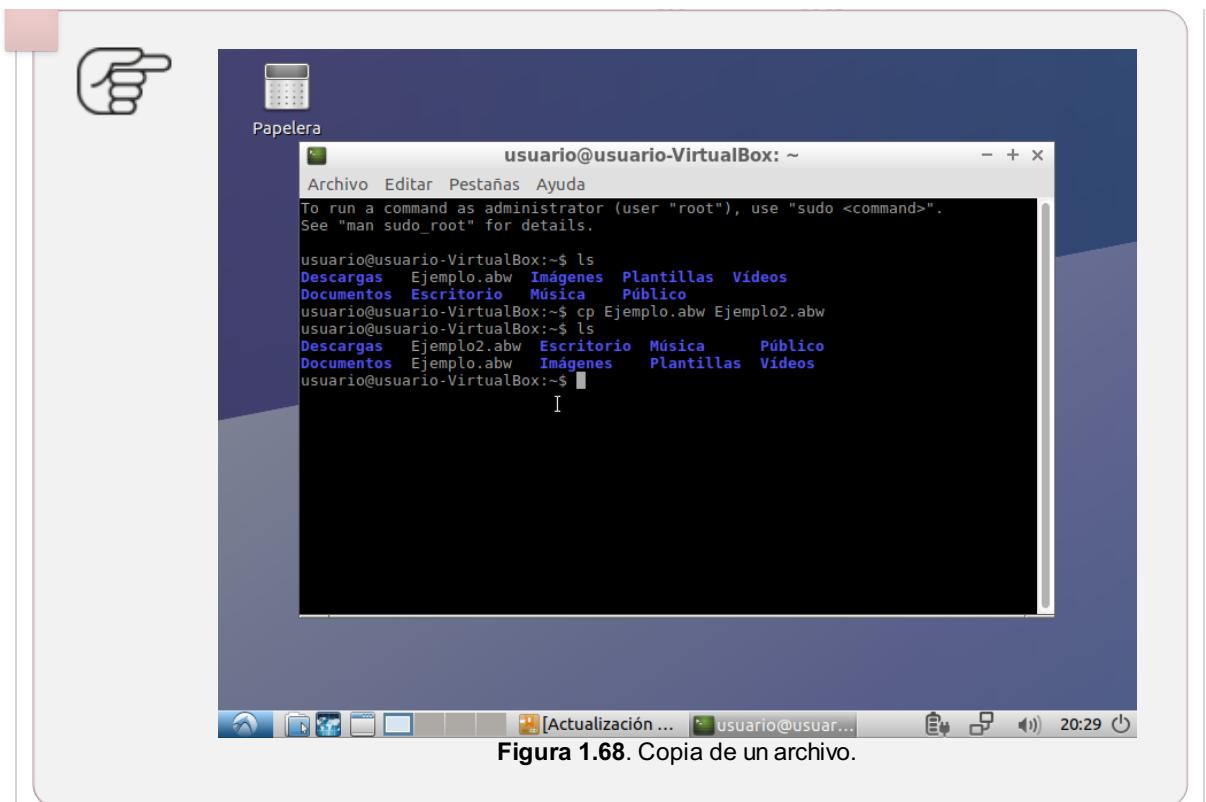


Figura 1.68. Copia de un archivo.

También se puede usar como **cp file1 file2 namedir** que hace copias de file1 y file2 en el directorio namedir (figura 1.69.).

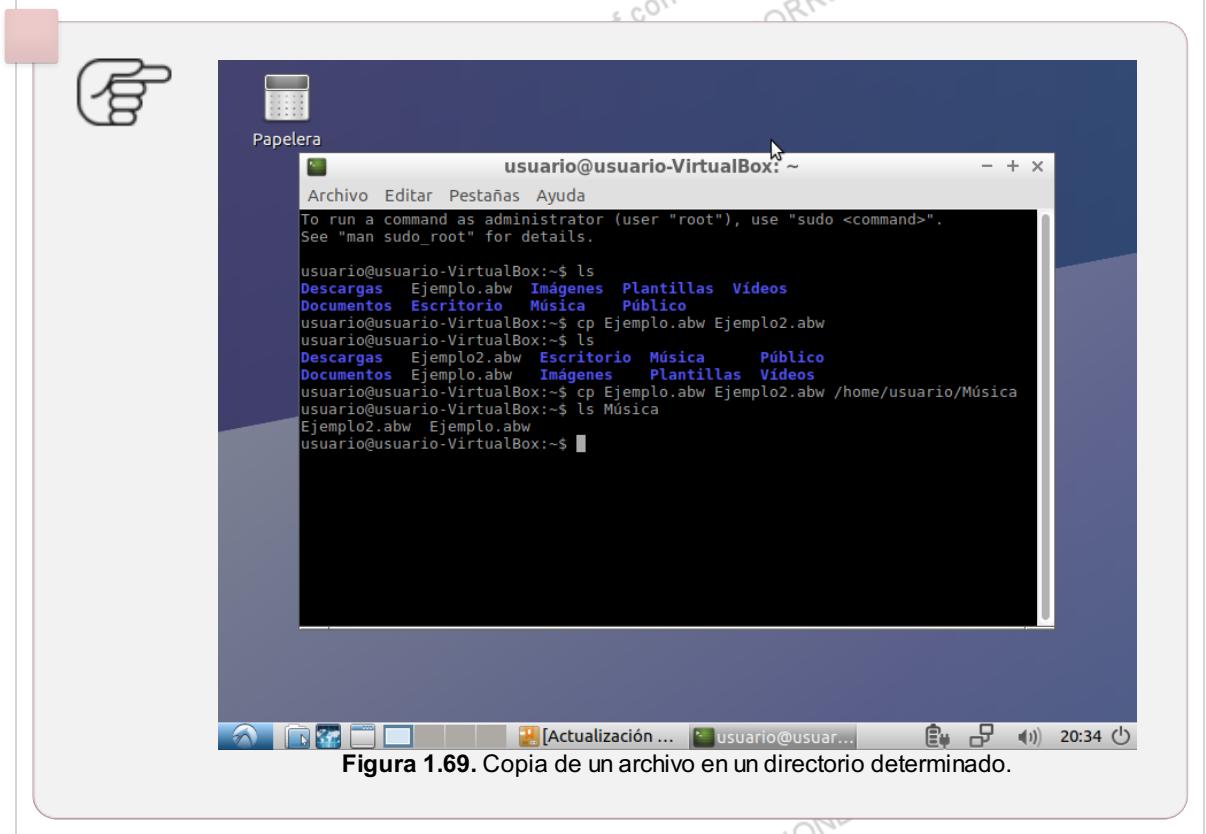


Figura 1.69. Copia de un archivo en un directorio determinado.

Traslado y cambio de nombre de archivos

La sintaxis es **mv file1 file2** y realiza la misma función que cp, pero eliminando el archivo original. Desde la visión del usuario, se cambia el nombre a file1 por file2 (figura 1.70.).

The screenshot shows a terminal window titled "Papelera" (Recycle Bin) with a blue header bar. The window title bar says "usuario@usuario-VirtualBox: ~". The terminal content is as follows:

```
To run a command as administrator (user "root"), use "sudo <command>".  
See "man sudo_root" for details.  
  
usuario@usuario-VirtualBox:~$ ls  
Descargas Ejemplo.abw Imágenes Plantillas Vídeos  
Documentos Escritorio Música Público  
usuario@usuario-VirtualBox:~$ cp Ejemplo.abw Ejemplo2.abw  
usuario@usuario-VirtualBox:~$ ls  
Descargas Ejemplo2.abw Escritorio Música Público  
Documentos Ejemplo.abw Imágenes Plantillas Vídeos  
usuario@usuario-VirtualBox:~$ cp Ejemplo.abw Ejemplo2.abw /home/usuario/Música  
usuario@usuario-VirtualBox:~$ ls Música  
Ejemplo2.abw Ejemplo.abw  
usuario@usuario-VirtualBox:~$ mv Ejemplo2.abw Ejemplo3.abw  
usuario@usuario-VirtualBox:~$ ls  
Descargas Ejemplo3.abw Escritorio Música Público  
Documentos Ejemplo.abw Imágenes Plantillas Vídeos  
usuario@usuario-VirtualBox:~$
```

The terminal window has a dark background with light-colored text. The bottom of the screen shows the desktop environment's taskbar with icons for the application menu, file manager, terminal, and system status.

Figura 1.70. Cambio de nombre de un archivo.

- Si los nombres que aparecen son de directorios, entonces el comando **mv namedir1 namedir2** cambia el nombre del subdirectorio namedir1 por namedir2.
- **mv file1 file2 namedir** traslada uno o más archivos (file1, file2, etc.) al directorio namedir conservando el nombre.

Borrado de archivos

El comando **rm** elimina uno o más archivos de un directorio en el cual tengamos permiso de escritura. La sintaxis es **rm file1 file2** (figura 1.71.). En el comando, se pueden utilizar los caracteres de sustitución * y ?

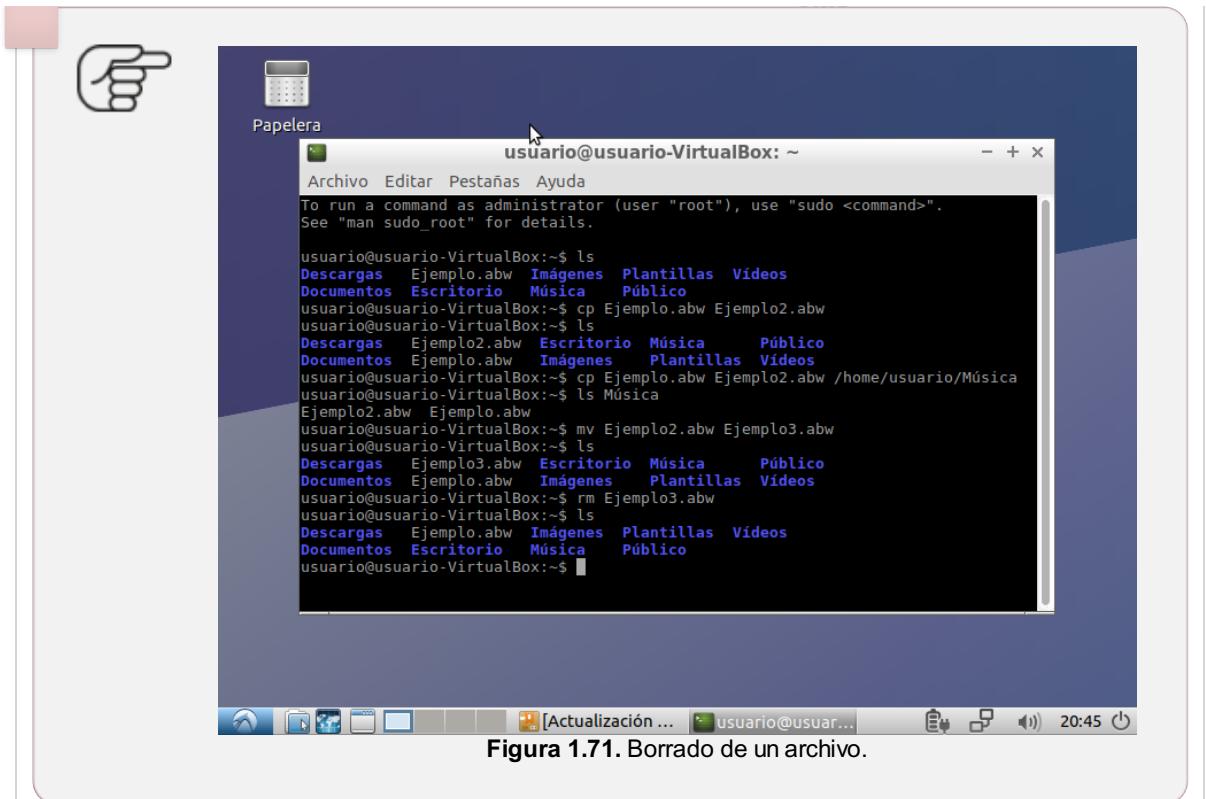


Figura 1.71. Borrado de un archivo.

Si se usa la opción **-i**, se pide confirmación para borrar cada archivo de la lista especificada: **rm -i file1 file2** (figura 1.72.).

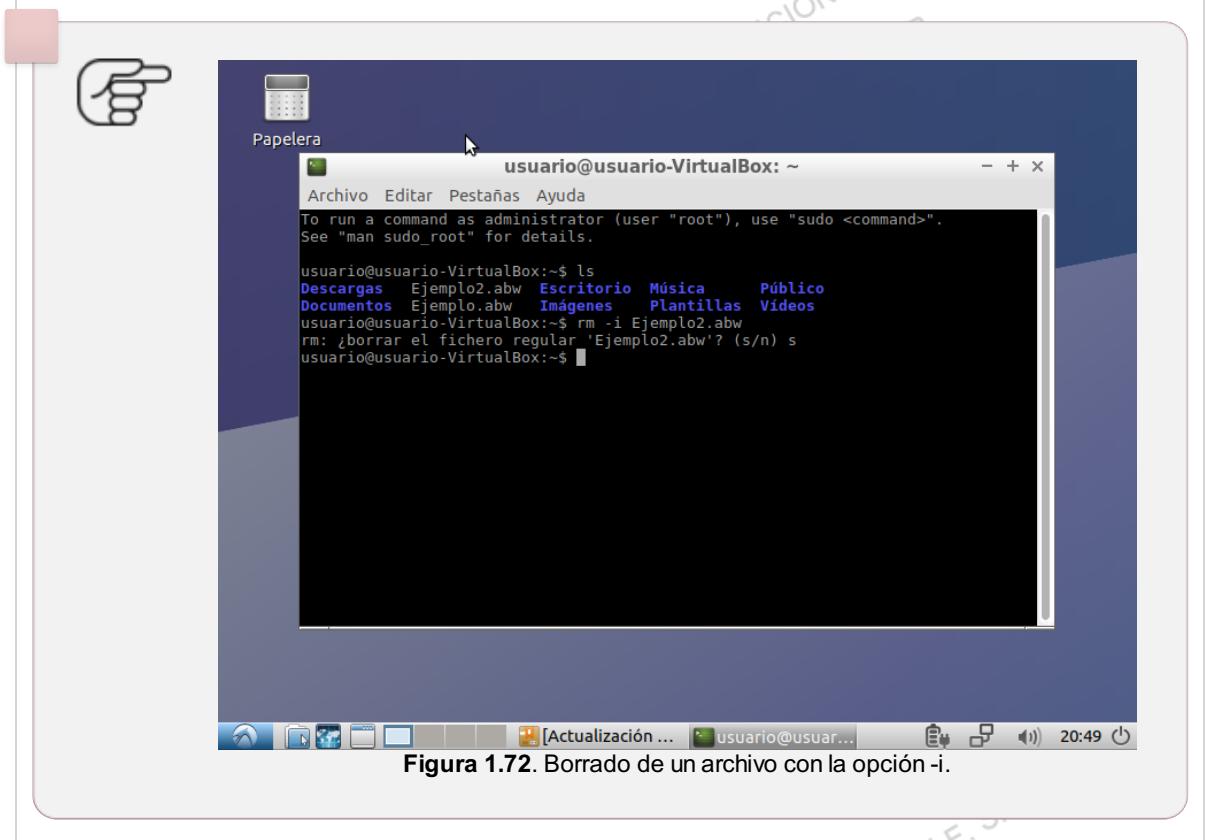


Figura 1.72. Borrado de un archivo con la opción **-i**.

Enlace a archivos

Un mismo archivo puede estar repetido con más de un nombre. También es posible tener un mismo archivo con varios nombres distintos y poder acceder a él desde más de un directorio. Esto último se denomina enlaces múltiples a un archivo y la forma de crearlo es mediante el comando **ln: ln file1 file2**.

De esta forma el archivo file1 tiene dos nombres: file1 y file2. Además, este comando advierte previamente si el nombre file2 está ocupado y en este caso no se ejecuta. En el siguiente ejemplo se crea un enlace al archivo Ejemplo.abw (figura 1.73.).

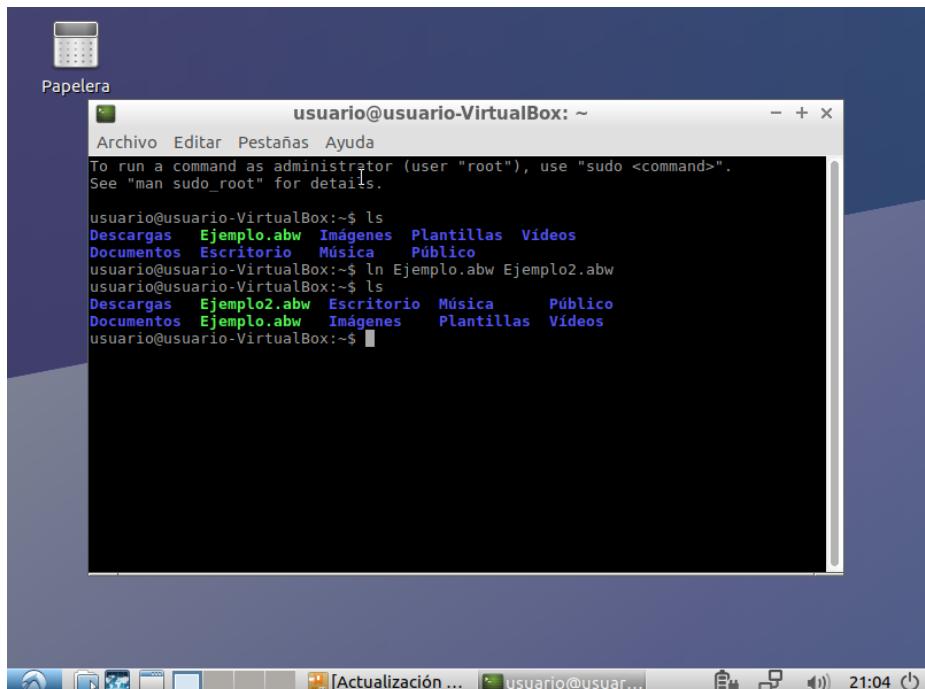


Figura 1.73. Enlace a un archivo.

Los archivos enlazados a otro se borran como los archivos normales, de manera que si se borra el archivo original permanece su contenido en los archivos enlazados.

Características de un archivo

Mediante el comando **file** se puede obtener información acerca de un archivo: **file fich** (figura 1.74.).

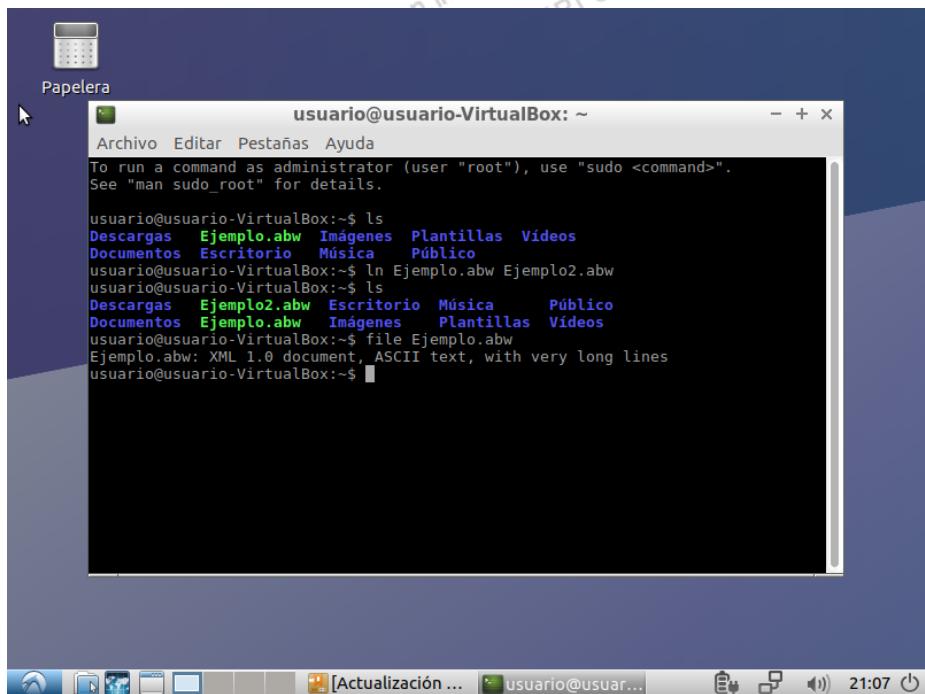


Figura 1.74. Características de un archivo.

Cambio de modo en los archivos

En Linux, los archivos tienen asociados permisos que indican las operaciones que le está permitido realizar a un usuario sobre los mismos. Los permisos de cada archivo se pueden ver con el comando **ls -l**. Para cambiar los permisos de un archivo se emplea el comando **chmod [quién] oper, permiso, files** donde:

quién:

Indica a quién afecta el permiso que se desea cambiar. Es una combinación de las letras **u** para el usuario, **g** para el grupo del usuario, **o** para los otros usuarios, y **a** para todos los anteriores. Si no se indica nada, se supone **a**.

oper:

Indica si el permiso se da usando un **+** o se quita usando un **-**.

permiso:

Indica el permiso que se quiere dar o quitar. Es una combinación de las letras: **r** (leer), **w** (escribir), **x** (ejecutar). Estos permisos son diferentes cuando se aplican a directorios: **r** da la posibilidad de ver el contenido del directorio con el comando **ls**; el permiso **w** da la posibilidad de crear y borrar archivos en ese directorio; el permiso **x** autoriza a buscar y utilizar un archivo concreto.

files:

Nombres de los archivos o directorios cuyos modos de acceso se quieren cambiar.

En el siguiente ejemplo, se cambian los permisos al archivo Ejemplo.abw (figura 1.75.)



```
usuario@usuario-VirtualBox:~$ ls
Descargas Ejemplo.abw Imágenes Plantillas Vídeos
Documentos Escritorio Música Público
usuario@usuario-VirtualBox:~$ ln Ejemplo.abw Ejemplo2.abw
usuario@usuario-VirtualBox:~$ ls
Descargas Ejemplo2.abw Escritorio Música Público
Documentos Ejemplo.abw Imágenes Plantillas Vídeos
usuario@usuario-VirtualBox:~$ file Ejemplo.abw
Ejemplo.abw: XML 1.0 document, ASCII text, with very long lines
usuario@usuario-VirtualBox:~$ chmod a-r Ejemplo.abw
usuario@usuario-VirtualBox:~$ ls -l
total 40
drwxr-xr-x 2 usuario usuario 4096 jun 29 14:24 Descargas
drwxr-xr-x 2 usuario usuario 4096 jun 29 14:24 Documentos
--wx-w--w- 2 usuario usuario 2717 jun 29 20:28 Ejemplo2.abw
--wx-w--w- 2 usuario usuario 2717 jun 29 20:28 Ejemplo.abw
drwxr-xr-x 2 usuario usuario 4096 jun 29 14:24 Escritorio
drwxr-xr-x 2 usuario usuario 4096 jun 29 14:24 Imágenes
drwxr-xr-x 2 usuario usuario 4096 jun 29 20:34 Música
drwxr-xr-x 2 usuario usuario 4096 jun 29 14:24 Plantillas
drwxr-xr-x 2 usuario usuario 4096 jun 29 14:24 Público
drwxr-xr-x 2 usuario usuario 4096 jun 29 14:24 Vídeos
usuario@usuario-VirtualBox:~$
```

Figura 1.75. Cambio de propiedades de un archivo.

Cambio de propietario a un conjunto de archivos

El comando **chown** se emplea para cambiar de propietario a un determinado conjunto de archivos: **chown newowner file1 file2...**



Solo lo puede emplear el actual propietario de los mismos. Los nombres de propietario se encuentran almacenados en el archivo /etc/passwd.

Cambio de grupo de un archivo

El comando **chgrp** se emplea para cambiar el grupo al que pertenece un archivo: **chgrp newgroup file1 file2...** Los grupos de usuarios están almacenados en el archivo /etc/group.

Visualización sin formato de un archivo

El comando **cat filename** permite visualizar el contenido de uno o más archivos de forma no formateada y también permite copiar uno o más archivos como apéndice de otro ya existente:

- **cat filename**: saca por pantalla el contenido del archivo filename.
- **cat file1 file2 ...**: saca por pantalla, secuencialmente y según el orden especificado, el contenido de los archivos indicados.
- **cat file1 file2 >file3**: El contenido de los archivos file1 y file2 es almacenado en file3.
- **cat file1 file2 >>file3**: el contenido de file1 y file2 es añadido al final de file3.
- **cat >file1**: Acepta lo que se introduce por el teclado y lo almacena en file1 (se crea file1). Para terminar, se emplea <ctrl>d

Editor vi

Alternativamente, se puede usar el editor **vi**. El editor puede encontrarse en dos estados o modos:

- En el modo de comandos, vi está esperando alguna orden (por tanto, interpreta lo que se escribe como órdenes).
- En el modo de edición, vi está esperando que se escriba el texto del fichero (por tanto, interpreta lo escrito como texto).

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



Algunos comandos de vi son:

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

Visualización de archivos con formato

Este comando **pr file** imprime por consola el contenido de los archivos de una manera formateada, por columnas, controlando el tamaño de página y poniendo cabeceras al comienzo de las mismas:

pr file

Produce una salida estándar de 66 líneas por página, con un encabezamiento de 5 líneas: 2 en blanco, una de identificación y otras 2 líneas en blanco.

pr -In file

Produce una salida de n líneas por página.

pr -F file

Hace una pausa para presentar la página, hasta que se pulsa <return> para continuar.

pr -t file

Suprime las 5 líneas del encabezamiento y las del final de página.

pr -wn file

Ajusta la anchura de la línea a n posiciones.

pr -d file

Lista el archivo con espaciado doble.

pr -h `caracteres` file

El argumento o cadena de caracteres `caracteres` se convertirán en la cabecera del listado.

pr +n file

Imprime el archivo a partir de la página n.

Se pueden combinar varias opciones en un mismo comando, como por ejemplo **pr -dt file** o cambiar la salida con **pr file1 > file2**, que crea un archivo nuevo llamado file2 que es idéntico a file1, pero con formato por páginas y columnas.

Visualización de datos pantalla a pantalla

Los comandos **more** y **less** permiten visualizar un archivo pantalla a pantalla. El número de líneas por pantalla es de 23 de texto más una última línea de mensajes, donde aparecerá la palabra **more**. Cuando se pulsa la barra espaciadora se visualizará la siguiente pantalla. Para salir de este comando se pulsa **<ctrl>d** o **q**. La sintaxis es **more file**.



El comando **less** permite el desplazamiento a lo largo del texto empleando las teclas de cursores pudiendo ir hacia arriba o abajo de un archivo. La sintaxis es **less file**.

Búsqueda en archivos

El comando **grep 'conjuntocaracteres' file1 file2 file3** busca una palabra, clave o frase en un conjunto de directorios, indicando en cuáles de ellos la ha encontrado. Busca archivo por archivo, por turno, imprimiendo aquellas líneas que contienen el conjunto de caracteres buscado. Si el conjunto de caracteres está compuesto por dos o más palabras separadas por un espacio, se escribe entre apóstrofes '.



Se pueden utilizar expresiones regulares de la forma: **grep [-opcion] expresión_regular [referencia...]**

Las opciones que admite son:

- **c** escribe el número de las líneas que satisface la condición.
- **i**, no se distinguen mayúsculas y minúsculas.
- **I** escribe los nombres de los archivos que contienen líneas buscadas.
- **n**, cada línea es precedida por su número en el archivo.
- **s**, no se vuelcan los mensajes que indican que un archivo no se puede abrir.
- **v**, se muestran solo las líneas que no satisfacen el criterio de selección.



Algunos ejemplos:

- **grep "d" text** recupera las líneas que comienzan por d.
- **grep "[^d]" text** recupera las líneas que no comienzan por d.
- **grep -v "C" file1 > file2** quita las líneas de file1 que comienzan por C y lo copia en file2

Impresión

El comando **lpr nombre_archivo** imprime por defecto el archivo indicado. Si se usa sin argumentos imprime el texto que se introduzca a continuación en la impresora.

Compresión de archivos

El comando **tar -cvf nombre_archivo.tar archivo1 archivo2...** agrupa varios archivos en uno solo archivo tar. Con **gzip archivo** se comprime archivo (que es borrado) y se crea un archivo con nombre archivo.gz.

Se pueden realizar las operaciones inversas, de manera que se pueden extraer los archivos mediante **tar -xvf nombre_archivo.tar archivo1...** y descomprimir un archivo mediante **gzip -d archivo.gz**.

Dado que suele emplearse tar y gzip de forma consecutiva obteniéndose archivos con extensión tar.gz o tgz, entonces tar incluye la opción z para extraer archivos con esta extensión **tar -zxf archivo.tar.gz**.

Redirecciones

Los comandos tienen una entrada estándar (número 0) y dos salidas estándar (número 1 para la salida normal y número 2 para la salida con errores). Por defecto, la entrada y la salida estándar de un comando es la terminal, a no ser que el comando especifique los nombres de archivos que hagan de entrada y de salida, como por ejemplo el comando **cp file1 file2**.

Sin embargo, se puede redirigir la salida de un comando usando los operadores:

- (>) redirige la salida estándar hacia el archivo indicado y en caso de no existir se crea.
- (<) redirige la entrada estándar desde un determinado archivo.
- (>>) redirige la salida estándar hacia otro archivo, pero añadiendo dicha salida al final de ese archivo, sin sobrescribir el contenido original.



Por ejemplo:

- **date >>archivo** el archivo **archivo** contendrá información sobre todas las veces que se ha entrado en el sistema.
- **cat file1 file2 >file3** añade al archivo **file2** al final de **file1** y al conjunto lo llama **file3**.
- **cat file2 >>file1** realiza la misma operación que el anterior, pero al resultado lo llama **file1**.

Tuberías

Una tubería (**/**) permite comunicar la salida estándar de un comando con la entrada estándar de otro. Por ejemplo, **ls | mail juan** envía a juan una lista de los archivos del sistema. Con el operador de tubería se pueden empalmar tantos comandos como se deseé.

Bifurcación

Permite redirigir la salida de un comando a un determinado archivo y que también se bifurque hacia la terminal. Para ello se usa el operador **tee**, como por ejemplo **ls | tee file**, que muestra la salida por el terminal y además la envía al archivo **file**. En este ejemplo, si se quiere que la salida se añada al final de **file**, se usaría la opción **-a**: **ls | tee -a file**.

Redirección de los errores



Los mensajes de error se dirigen a la salida número 2, que normalmente es también la terminal. En algunos casos puede interesar redirigir estos mensajes a otra salida como, por ejemplo, cuando se ejecuta un programa en segundo plano.

Véanse a continuación los siguientes ejemplos:

- **gcc prueba.c 2>errores** se redirige la salida 2 hacia el archivo errores.
- **program <datos.d >resultados.r 2> & 1** se redirige la salida estándar de errores al mismo archivo que la salida estándar.

Ejecución de un programa



Existen varios comandos para gestionar la ejecución de un programa:

- El carácter **&** permite realizar una ejecución en segundo plano recuperando inmediatamente el control del terminal. Para ello se añade el carácter **&** al final del comando de ejecución **program <datos.d >resultados.r &**. Como resultado aparece en el terminal el número de proceso de la ejecución de este programa.
- Para detener la ejecución de un proceso se puede utilizar el comando **kill númerodeproceso**.
- Cuando se sale del sistema, si hay algún proceso ejecutándose en segundo plano se para, salvo que se use el comando **nohup program**. En este caso si no se utilizan redirecciones todas las salidas del programa se dirigen a un archivo llamado **nohup.out**.
- El comando **nice** permite realizar ejecuciones con baja prioridad de la forma:

 - **nice program &**
 - **nice nohup program &**

Para darle al programa la prioridad mínima habría que invocarlo como **nice -19 program &** donde el -19 indica la mínima prioridad.

- El comando **time**, precediendo a cualquier otro comando, suministra información acerca del tiempo total empleado en la ejecución, del tiempo de CPU utilizado por el programa del usuario y del tiempo de CPU consumido en utilizar recursos del sistema. Por ejemplo, **time gcc prueba.c** ofrece información sobre el tiempo de compilación y montaje del programa prueba.c.
- El comando **top** muestra una lista de los procesos que se están ejecutando. Sus principales opciones son **u**, que muestra los procesos que pertenecen a un determinado usuario), **k**, que elimina un proceso, y **h**, que muestra la ayuda del programa.

Otros comandos

En las siguientes tablas (Tabla 1), se muestran otros comandos interesantes del intérprete de comandos de Linux.



Comando	Significado
date	Muestra el día y la hora.
cal	Muestra el calendario. Tiene diversas opciones. Por ejemplo, cal 1945 mostraría el calendario del año 1945.
who	Indica qué usuarios tiene el ordenador en ese momento, en qué terminal están y desde qué hora.
whoami	Indica cuál es la terminal y la sesión en la que se está trabajando.
uptime	Muestra el tiempo que lleva encendido nuestro ordenador.
hostname	Muestra el nombre de la máquina.
bc	Abre una calculadora de texto. Para salir se introduce quit .
lshw	Muestra todas las características del hardware.
history	Muestra los comandos usados por el usuario en orden cronológico.
fc -l	Muestra los últimos comandos utilizados por el usuario.

man comando	Muestra el manual de un comando con sus modificadores y argumentos.
lspci	Muestra los diferentes dispositivos PCI.
umask	Muestra los permisos con los que el usuario creará sus archivos por defecto.
uname -r	Muestra la versión del kernel.
lsusb	Muestra información de los dispositivos que tengo conectados en los puertos USB.
xkill	Permite cerrar un programa bloqueado.
groups	Muestra los grupos del sistema a los que pertenece un usuario.
clear	Limpia la consola.
head	Indicando un número n y un nombre de fichero, muestra las n primeras líneas del fichero indicado.

Tabla 1.1. Comandos de Linux.

6.2. Creación de scripts

Un guión o script es un archivo de texto que contiene comandos de la Shell que se interpretarán cuando se ejecute el script.

Características

Se caracterizan por:

- Pueden tener cualquier nombre.
- El archivo debe tener permiso de ejecución.
- Para ejecutarlo se debe indicar el nombre del intérprete y el nombre del guión.
- Las instrucciones del guión se procesan por orden, una por línea, de manera que los saltos de línea se interpretan como un “INTRO”.
- Las órdenes podrían aparecer en una misma línea separadas por punto y coma.

Los guiones pueden tener una línea inicial que indica el tipo de intérprete que se quiere utilizar para ejecutar los comandos.



Por ejemplo `#!/bin/bash` indica que debe usarse el intérprete Bourne-Again Shell.

Asimismo, un guión puede contener comentarios que se marcan con el carácter # al inicio del texto del comentario.

Principales elementos

Los principales elementos de un script son:

Variables y parámetros

Definición de una variable

Para definir una variable se usa la estructura variable=valor donde variable es el nombre de la misma. Hay que tener en cuenta lo siguiente:

1. No puede haber un espacio entre el nombre de la variable, el signo = y el valor.
2. Si se desea que el valor contenga espacios, es necesario utilizar comillas.
3. Para recuperar el valor de una variable hay que anteponerle a su nombre el carácter \$. Por ejemplo, para visualizar el valor de una variable echo \$variable.

En el siguiente ejemplo, se define la variable Prueba="ls -l" y a continuación se invoca con \$Prueba (figura 1.76.):



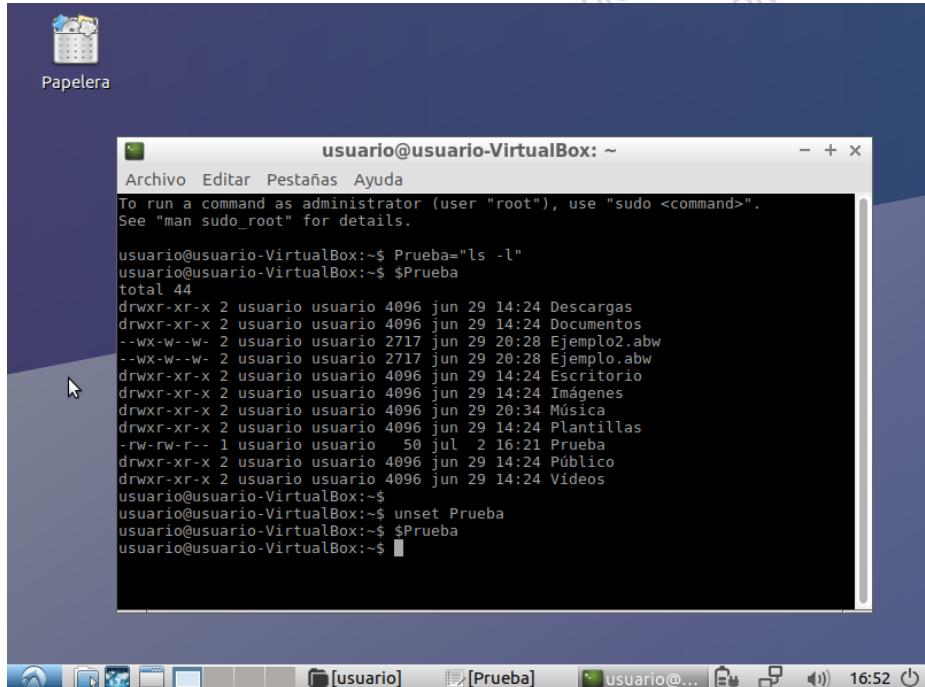
```
usuario@usuario-VirtualBox: ~
Archivo Editar Pestañas Ayuda
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

usuario@usuario-VirtualBox:~$ Prueba="ls -l"
usuario@usuario-VirtualBox:~$ $Prueba
total 44
drwxr-xr-x 2 usuario usuario 4096 jun 29 14:24 Descargas
drwxr-xr-x 2 usuario usuario 4096 jun 29 14:24 Documentos
--wx-w--w- 2 usuario usuario 2717 jun 29 20:28 Ejemplo2.abw
--wx-w--w- 2 usuario usuario 2717 jun 29 20:28 Ejemplo.abw
drwxr-xr-x 2 usuario usuario 4096 jun 29 14:24 Escritorio
drwxr-xr-x 2 usuario usuario 4096 jun 29 14:24 Imágenes
drwxr-xr-x 2 usuario usuario 4096 jun 29 20:34 Música
drwxr-xr-x 2 usuario usuario 4096 jun 29 14:24 Plantillas
-rw-rw-r-- 1 usuario usuario 50 jul 2 16:21 Prueba
drwxr-xr-x 2 usuario usuario 4096 jun 29 14:24 Público
drwxr-xr-x 2 usuario usuario 4096 jun 29 14:24 Vídeos
usuario@usuario-VirtualBox:~$
```

Figura 1.76. Ejemplo de variable.

Eliminación de una variable

Para eliminar una variable, se usa el comando **unset**. Por ejemplo (figura 1.77.):



```
usuario@usuario-VirtualBox: ~
Archivo Editar Pestañas Ayuda
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

usuario@usuario-VirtualBox:~$ Prueba="ls -l"
usuario@usuario-VirtualBox:~$ $Prueba
total 44
drwxr-xr-x 2 usuario usuario 4096 jun 29 14:24 Descargas
drwxr-xr-x 2 usuario usuario 4096 jun 29 14:24 Documentos
--wx-w--w- 2 usuario usuario 2717 jun 29 20:28 Ejemplo2.abw
--wx-w--w- 2 usuario usuario 2717 jun 29 20:28 Ejemplo.abw
drwxr-xr-x 2 usuario usuario 4096 jun 29 14:24 Escritorio
drwxr-xr-x 2 usuario usuario 4096 jun 29 14:24 Imágenes
drwxr-xr-x 2 usuario usuario 4096 jun 29 20:34 Música
drwxr-xr-x 2 usuario usuario 4096 jun 29 14:24 Plantillas
-rw-rw-r-- 1 usuario usuario 50 jul 2 16:21 Prueba
drwxr-xr-x 2 usuario usuario 4096 jun 29 14:24 Público
drwxr-xr-x 2 usuario usuario 4096 jun 29 14:24 Vídeos
usuario@usuario-VirtualBox:~$ 
usuario@usuario-VirtualBox:~$ unset Prueba
usuario@usuario-VirtualBox:~$ $Prueba
usuario@usuario-VirtualBox:~$ 
```

Figura 1.77. Eliminación de una variable.

Variables definidas en la Shell

Existen algunas variables definidas en la Shell. En la siguiente tabla se muestran las principales variables (tabla 1.2.).

Nombre de variable	Significado
LOGNAME	Nombre del usuario.
HOME	Directorio de trabajo del usuario actual.
PATH	Caminos usados para ejecutar órdenes o programas.
PWD	Directorio activo.
TERM	El tipo de la terminal actual.
SHELL	Shell actual.
\$?	Contiene el valor de salida de la última orden ejecutada. El valor de '0' indica que no ha habido errores, mientras que otros valores indican errores.

Tabla 1.2. Algunas variables predefinidas.

Parámetros

Un guión puede recibir parámetros en la línea de órdenes para considerarlos durante su ejecución. Los parámetros recibidos se guardan en una serie de variables que el script puede consultar cuando lo necesite. Los nombres de estas variables son: \$1 \$2 \$3 ... \${10} \${11} \${12} ... de manera que:

- La variable \$0 contiene el nombre con el que se ha invocado al script, es decir, el nombre del programa.
- \$1 contiene el primer parámetro.
- \$2 contiene el segundo parámetro.
- ...

A continuación, en el siguiente ejemplo de un guión Shell se muestran los valores de los parámetros (figura 1.78.).

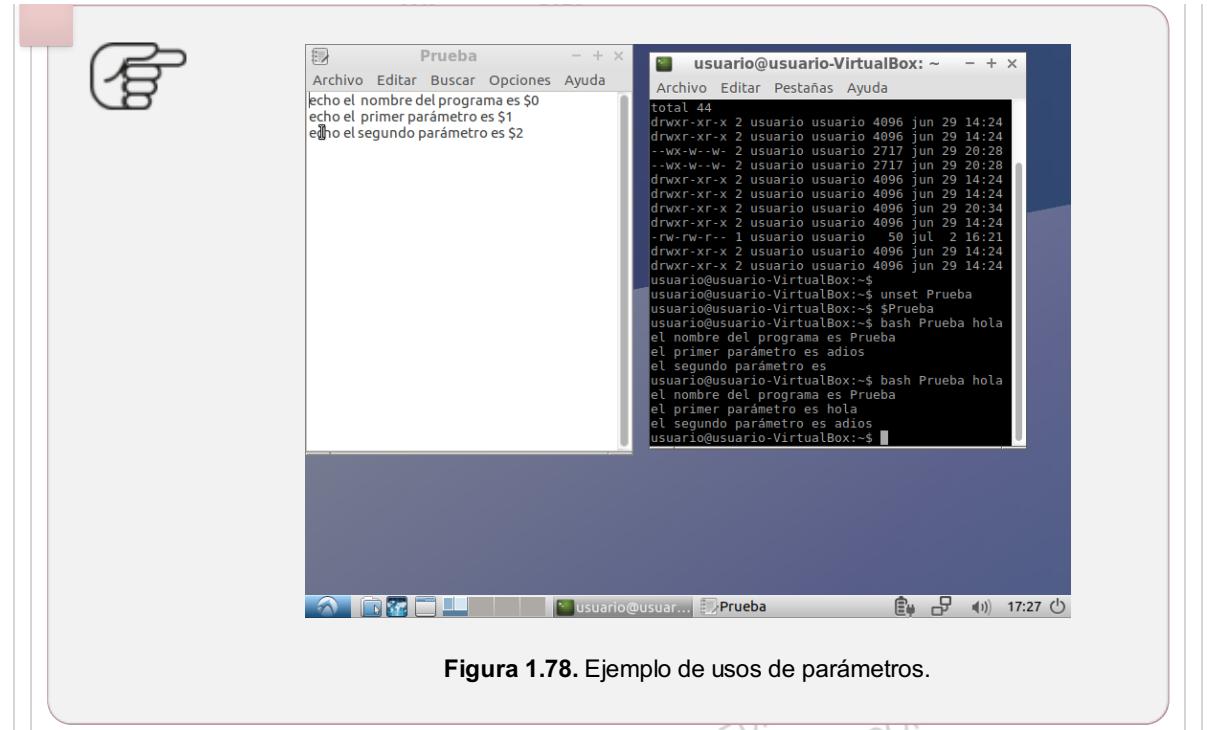


Figura 1.78. Ejemplo de usos de parámetros.

La orden shift mueve todos los parámetros una posición a la izquierda, esto hace que el contenido del parámetro \$1 desaparezca y sea reemplazado por el contenido de \$2, \$2, que es reemplazado por \$3, etc. Además, existen algunas variables especiales tales como \$# que contienen el número de parámetros que ha recibido el script, y \$* o \$@ contienen todos los parámetros recibidos.

Evaluación de variables

A continuación, se describen algunas reglas de la evaluación de variables:

\$var o \${var}

Representa el valor de la variable o nada si la variable no está definida.

\${var-val}

Devuelve el valor de var si está definida, si no val.

\${var=val}

Valor de var si está definida, si no val y el valor de var pasa a ser val.

\${var?message}

Devuelve \$var si está definida y si no, imprime el mensaje en el terminal del Shell. Si el mensaje está vacío imprime uno estándar.

\${val+var}

Devuelve \$val, si está var definida, si no, nada.

Ejemplo

El siguiente ejemplo muestra cómo podemos usar una variable, asignándole un valor en caso de que no esté definida (figura 1.79.).

The screenshot shows two terminal windows side-by-side. The left window is titled 'Prueba' and contains the following text:

```
echo el valor de var1 es $var1
echo el varlo de var1 es ${var1=5}
echo el valor de var1 es $var1
```

The right window is titled 'usuario@usuario-VirtualBox: ~' and contains the following text:

```
--wx-w--w- 2 usuario usuario 2717 jun 29 20:28
drwxr-xr-x 2 usuario usuario 4096 jun 29 14:24
drwxr-xr-x 2 usuario usuario 4096 jun 29 14:24
drwxr-xr-x 2 usuario usuario 4096 jun 29 20:34
drwxr-xr-x 2 usuario usuario 4096 jun 29 14:24
-rw-rw-r- 1 usuario usuario 50 jul 2 16:21
drwxr-xr-x 2 usuario usuario 4096 jun 29 14:24
drwxr-xr-x 2 usuario usuario 4096 jun 29 14:24
usuario@usuario-VirtualBox:~$ unset Prueba
usuario@usuario-VirtualBox:~$ $Prueba
usuario@usuario-VirtualBox:~$ bash Prueba hola
el nombre del programa es Prueba
el primer parámetro es adios
el segundo parámetro es
usuario@usuario-VirtualBox:~$ bash Prueba hola
el nombre del programa es Prueba
el primer parámetro es hola
el segundo parámetro es adios
usuario@usuario-VirtualBox:~$ bash Prueba
el valor de varl es
el varlo de varl es 5
el valor de varl es 5
usuario@usuario-VirtualBox:~$
```

Figura 1.79. Ejemplo de usos de parámetros

Arrays

Un array es una colección de elementos del mismo tipo, dotados de un nombre, y que se almacenan en posiciones contiguas de memoria.



El primer elemento del array está numerado con el 0.

No hay un tamaño límite para un array y la asignación de valores se puede hacer de forma alterna.

Sintaxis

La sintaxis para declarar un array es la siguiente:

- Crear e inicializar un array: nombre_array=(val1 val2 val3 ...).
- Asignar un valor al elemento x del array: nombre_array[x]=valor.
- Acceder al elemento x: \${nombre_array[x]}.
- Consultar todos los elementos: \${nombre_array[*]} o \${nombre_array[@]}.

Aspectos importantes

Es preciso tener en cuenta lo siguiente:

- Si al referenciar a un array no se utiliza subíndice se considera que se está referenciando a su primer elemento.
- Para conocer el tamaño en bytes del array se utiliza #\${nombre_array[x]}, donde x puede ser un subíndice, o bien los caracteres * o @.

Estructuras de control

IF-CASE

Indica que determinadas órdenes solo se ejecuten cuando se cumplan unas condiciones concretas. Para ello, se utilizan las estructuras if o case.

Estructura if

```
if [expresión]
then
    órdenes a ejecutar si se cumple la condición
elif [expresión]
then
    órdenes a ejecutar si se cumple la condición
else
    órdenes a ejecutar en caso contrario
fi
```



Por ejemplo:

```
if grep -q set prueba.bash  
then  
    echo encontrada la palabra clave set  
else  
    echo no encontrada la palabra clave set  
fi
```

Estructura case

```
case $var in  
    v1) ... #Acción a realizar si var toma el valor v1  
        ;;  
    v2|v3) ...#Acción a realizar si var toma el valor v2 o v3  
        ;;  
    *) ...# Caso por defecto  
        ;;  
esac
```



Por ejemplo:

```
case $var in
```

```
    1) echo La variable var es un uno
```

```
    ::
```

```
    2) echo La variable var es un dos
```

```
    ::
```

```
    *) echo La variable var no es ni un uno ni un dos
```

```
    ::
```

```
esac
```

BUCLES

Permite ejecutar bloques de órdenes de forma iterativa dependiendo de una condición. Para ello se utilizan las estructuras if o case.

Estructura while

```
while [ expresión ] # Mientras la expresión sea cierta ...
```

```
do
```

```
...
```

```
done
```

Estructura until

```
until [ expresión ]
```

```
# Mientras la expresión sea falsa ...
```

```
do ...
```

```
done
```

Estructura for

```
for var in lista#Por cada valor en la estructura lista se ejecuta una iteración.
```

```
do  
órdenes a ejecutar  
done
```



Por ejemplo:

```
for i in 10 30 70  
do  
echo Mi número favorito es $i  
done
```

Break y continue

Sirven para interrumpir la ejecución secuencial del cuerpo del bucle:

- **break** transfiere el control a la orden que sigue a done, haciendo que el bucle termine antes de tiempo.
- **continue** transfiere el control a done, haciendo que se evalúe de nuevo la condición, prosiguiendo el bucle.

En ambos casos, las órdenes del cuerpo del bucle siguientes a estas sentencias no se ejecutan. Lo normal es que formen parte de una sentencia condicional, como if.

Órdenes internas de la Shell

Una orden interna del Shell es una orden que el intérprete implementa y que ejecuta sin llamar a programas externos.

echo

Envía una cadena a la salida estándar.



Por ejemplo:

```
echo Esto es una cadena
```

read

Lee una cadena de la entrada estándar.



Por ejemplo:

```
echo -n "Introduzca un valor para var1: "
read var1
echo "var1 = $var1"
```

La orden read puede leer diferentes variables a la vez. También se puede combinar el uso de read con echo para mostrar un prompt que indique qué es lo que se está pidiendo.

Caracteres especiales

Hay una serie de caracteres especiales para usar en echo y que permiten posicionar el cursor en un sitio determinado:

\b: retrocede una posición (sin borrar)

\f : alimentación de página

\n: salto de línea

\t: tabulador

Para que echo reconozca estos caracteres es necesario utilizar la opción .e. y utilizar comillas dobles:

```
$ echo -e "Hola \t ¿cómo estás?"
```

```
hola como estás
```

cd

Cambia de directorio.

pwd

Devuelve el nombre del directorio actual.

let arg [arg]

Cada arg es una expresión aritmética a ser evaluada: \$ let a=\$b+7. Si el último arg se evalúa a 0, let devuelve 1; si no, devuelve 0.

exit

Finaliza la ejecución del guión, recibe como argumento un entero que será el valor de retorno.

true y false

Devuelven 0 y 1 siempre, respectivamente. El valor 0 se corresponde con true y cualquier valor distinto de 0 con false.

La orden test

Test es una orden que permite evaluar si una expresión es verdadera o falsa. Se usan en la estructura if/then/else/ para determinar qué parte del script se va a ejecutar.

La sintaxis de test puede ser una de las dos que se muestran a continuación:

test expresión

[expresión]

Algunos operadores que pueden aparecer en las condiciones a evaluar:

Operadores para números

-eq	Igual a
-ne	Distinto de
-lt	Menor que
-le	Menor o igual que
-gt	Mayor que
-ge	Mayor o igual que

Figura 1.80. Operadores para números.



Por ejemplo:

```
if [ $e -eq 1 ]
then
echo Vale 1
else
echo No vale 1
fi
```

Operadores para cadenas

-z cadena	Verdad si la longitud de cadena es cero
-n cadena o cadena	Verdad si la longitud de cadena no es cero
cadena1 == cadena2	Verdad si las cadenas son iguales. Se puede emplear = en vez de ==
cadena1 != cadena2	Verdad si las cadenas no son iguales
cadena1 < cadena2	Verdad si cadena1 se ordena lexicográficamente antes de cadena2 en la localización en curso
cadena1 > cadena2	Verdad si cadena1 se ordena lexicográficamente después de cadena2 en la localización en curso

Figura 1.81. Operadores para cadenas.



Por ejemplo:

```
#!/bin/bash

S1='cadena'

S2='Cadena'

if [ $S1!=$S2 ];

then

echo "S1(''$S1') no es igual a S2(''$S2')"

fi

if [ $S1==$S1 ];

then

echo "S1(''$S1') es igual a S1(''$S1')"

fi
```

Operadores sobre ficheros

-e fichero	El fichero existe
-r fichero	El fichero existe y tengo permiso de lectura
-w fichero	El fichero existe y tengo permiso de escritura
-x fichero	El fichero existe y tengo permiso de ejecución
-f fichero	El fichero existe y es regular
-s fichero	El fichero existe y es de tamaño mayor a cero
-d fichero	El fichero existe y es un directorio

Figura 1.82. Operadores sobre ficheros.

Por ejemplo:

```
if test -f "$1" # ¿ es un fichero ?  
then  
more $1  
elif test -d "$1" # ¿ es un directorio ?  
then  
(cd $1;ls -l|more)  
else # no es ni fichero ni directorio  
echo "$1 no es fichero ni directorio"  
fi
```

Operadores lógicos

-o	OR
-a	AND
!	NOT
\(Paréntesis izquierdo
\)	Paréntesis derecho

Figura 1.83. Operadores lógico

Funciones

Declaración

Para declarar una función:

function nombre:

{ mi_código }

Llamar a la función es como llamar a otro programa, solo hay que escribir su nombre.



Por ejemplo:

Se define la función hola

```
function hola {
```

```
    echo ¡Hola!
```

```
}
```

```
hola # Se llama a la función hola
```

Definición con parámetros

Se pueden definir funciones con parámetros:

```
#!/bin/bash
```

```
function e {
```

```
    echo $1
```

```
}
```

Se podría invocar como:

```
e Hola
```

Ejemplos

Algunos ejemplos de scripts:



Ejemplo 1. Guión que solicita confirmación si va a sobrescribir un archivo cuando se use la orden cp.

```
#!/bin/bash

if [ -f $2 ]
then
echo "$2 existe. ¿Quieres sobreescibirlo? (s/n)"
read sn
if [ $sn = "N" -o $sn = "n" ]
then
exit 0
fi
fi
cp $1 $2
```



Ejemplo 2. Guión que liste los directorios existentes en el directorio actual.

```
#!/bin/bash

for archivo in *
do
test -d $archivo && ls $archivo
done
```



Ejemplo 3. Guión que borra con confirmación todos los archivos indicados como argumentos en la línea de comandos.

```
#!/bin/bash

while test "$1" != ""

do

rm -i $1

shift

done
```



Ejemplo 4. Guión que evalúa la extensión de un archivo. Si esta se corresponde con .txt., copia el archivo al directorio ~/copias. Si es otra la extensión presenta un mensaje.

```
case $1 in

*.txt)

cp $1 ~/copias/$1

;;

*)

echo "$1 extensión desconocida"

;;

Esac
```



Ejemplo 5. Guión que pone el atributo de ejecutable a los archivos pasados como argumento.

```
for fich in $@  
do  
    if test -f $fich  
    then  
        chmod u+x $fich  
    fi  
done
```



Ejemplo 6. Guión que lee dos números del teclado e imprime su suma.

```
#!/bin/bash  
  
echo "Introduzca un número \n"  
  
read numero1  
  
echo "Introduzca otro número \n"  
  
read numero2  
  
let respuesta=$numero1+$numero2  
  
echo "$numero1 + $numero2 = $respuesta \n"
```

campusformacion.imf.com © EDICIONES ROBLE
WADIE ALBAKRI CORREGIDOR

EDICIONES ROBLE, S.L.

VII. Resumen

En esta unidad, se ha introducido el concepto de máquina virtual como un mecanismo que facilita la ejecución de una simulación de un sistema operativo diferente al que se tiene instalado por defecto. Para poder crear y ejecutar una máquina virtual, se requiere de un software específico. Las dos herramientas más importantes para este fin son Virtual Box y VMware.

En esta unidad se ha descrito cómo usar la herramienta Virtual Box para crear, ejecutar y configurar máquinas virtuales. Para ello, se ha creado una máquina virtual a partir de una distribución de Linux denominada Lubuntu.

Asimismo, se ha descrito cómo se puede importar una máquina virtual creada por terceros. Para este caso, se ha utilizado una máquina virtual de una distribución de Linux de tipo Lubuntu.

La última parte de la unidad se ha dedicado a describir la Shell de comandos de Linux, habiéndose revisado aquellos que permiten una administración básica del sistema.

VIII. Caso práctico

Se pide crear un script que muestre el siguiente menú:

Ver directorio actual.....[1]

Copiar archivos.....[2]

Editar archivos.....[3]

Imprimir archivo.....[4]

Salir del menú.....[5]"

Donde:

- Si elige la primera opción, se mostrarán los archivos del directorio actual.
- Si elige la segunda opción, se le pedirá el nombre del archivo que se quiere copiar y el nombre del archivo donde se quiere copiar.
- Si elige la tercera opción, se le pedirá el nombre del archivo que se desea editar y se abrirá el editor vi para editarlo.
- Si elige la cuarta opción, se le pedirá el nombre del archivo que se desea imprimir y se imprimirá el mismo.
- Si elige la quinta opción, se saldrá del script.

Solución

```
while true  
do  
clear  
echo "
```

Ver directorio actual.....[1]

Copiar ficheros.....[2]

Editar ficheros.....[3]

Imprimir fichero.....[4]

Salir del menú.....[5]"

read i

case \$i in

1) ls -l|more; read z

;;

2) echo "Introduzca [desde] [hasta]"

```
read x y
```

```
cp $x $y
```

```
read x
```

```
:
```

3) echo "¿Nombre de fichero a editar?"

```
read x;
```

```
vi $x
```

```
:
```

4) echo "¿Nombre de fichero a imprimir?"

```
read x
```

```
lpr $x
```

```
:
```

5) clear; break

```
:
```

```
esac
```

```
done
```

Recursos

Enlaces de Interés



<https://aws.amazon.com/es/ec2/vm-import/>

<https://aws.amazon.com/es/ec2/vm-import/>

Amazon. VM Import/Export

Bibliografía

- *Guía oficial de primeros pasos con VirtualBox.* [En línea] URL disponible en <https://www.virtualbox.org/manual/ch01.html> :
- Garrels, Machtelt. *Bash Guide for Beginners.* 2008. [En línea] URL disponible en <http://www.tldp.org/LDP/Bash-Beginners-Guide/Bash-Beginners-Guide.pdf> :
- Love, Robert. *Linux Kernel Development.* Addison Wesley; 2010, 3.^a ed. :
- Matthew, Neil y Stones, Richard. *Beginning Linux Programming.* Ed. Wiley; 2007, 4.^a ed. :
- Mitchell, Mark et al. *Advanced Linux Programming.* Indianapolis, IN: New Riders Publishing; 2001. [En línea] URL disponible en: http://richard.esplins.org/static/downloads/linux_book.pdf :

Glosario.

- **Crear una máquina virtual:** es un sinónimo de virtualizar.
- **Exportar una máquina virtual:** es la acción de exportar una máquina virtual creada para distribuirla entre otras personas.
- **Importar/Cargar máquina virtual:** es la acción de importar una máquina virtual que ha sido creada por un tercero a un sistema de gestión de máquinas virtuales para poder usarla.
- **Lubuntu:** es una distribución de Linux que requiere pocos recursos hardware.
- **Máquina virtual:** es una simulación de un sistema operativo. Físicamente es un archivo con varias extensiones posibles tales como .ova, .vmdk, etc.
- **Script o guión:** es un archivo de texto que contiene comandos de la Shell que se interpretarán cuando se ejecute el script.
- **Shell o intérprete:** aplicación que permite interactuar al usuario con el sistema operativo mediante la ejecución de comandos o sentencias.
- **Sistema operativo anfitrión:** sistema operativo sobre el que se ejecuta una máquina virtual.
- **Sistema operativo huésped:** sistema operativo virtualizado que se ejecuta sobre el sistema operativo anfitrión.
- **Sistema operativo virtualizado:** es un sinónimo de máquina virtual.

- **Software multiplataforma:** es un software que puede ejecutarse en más de un sistema operativo.
- **Virtual Box:** software gratuito para crear máquinas virtuales.
- **Virtualizar:** es la acción de crear una máquina virtual.
- **VMware:** software con diferentes versiones de licencias para crear máquinas virtuales.