

## **UNIDAD 2: INSTALACIÓN DE SISTEMAS OPERATIVOS**

### **MÓDULO PROFESIONAL:**

**Sistemas Informáticos**

# Índice

RESUMEN INTRODUCTORIO.....	4
INTRODUCCIÓN.....	4
CASO INTRODUCTORIO .....	5
1. PRINCIPALES SISTEMAS OPERATIVOS .....	6
1.1 Sistemas operativos Windows .....	6
1.2 Sistemas operativos Macintosh.....	12
1.3 Sistemas operativos Unix/Linux.....	13
1.3.1 Distribuciones comerciales de Linux.....	14
1.3.2 Distribuciones gratuitas de Linux.....	16
2. VIRTUALIZACIÓN .....	18
2.1 VirtualBox.....	19
3. INSTALACIÓN DE UN SISTEMA OPERATIVO. UBUNTU.....	23
3.1 Requisitos mínimos de Ubuntu .....	23
3.2 Proceso de instalación.....	24
3.2.1 Creación de la máquina virtual.....	24
3.2.2 Instalación de Ubuntu sobre la máquina virtual.....	29
4. GESTORES DE ARRANQUE .....	38
4.1 Secuencia de arranque del sistema .....	38
4.2 El gestor de arranque .....	39
4.3 El gestor de arranque GNU GRUB .....	40
4.4 Ejecución de GNU GRUB.....	41
5. MAN, LA AYUDA DEL SISTEMA.....	43
5.1 Posibles problemas con el idioma.....	45
5.2 Configuración de las locales .....	46
5.3 Contenido de man.....	49
5.4 Secciones de man .....	51

5.5 Opciones de man .....	52
6. ACTUALIZACIÓN DEL SISTEMA.....	54
RESUMEN FINAL .....	56

## RESUMEN INTRODUCTORIO

A lo largo de esta unidad revisaremos los principales sistemas operativos existentes en el mercado, centrándonos fundamentalmente en las familias Windows y Linux.

Presentaremos a continuación uno de los conceptos más importantes en los sistemas informáticos en la actualidad: la virtualización. Veremos qué ventajas aporta, y estudiaremos cómo llevarla a cabo de forma práctica mediante el programa VirtualBox.

Sobre una máquina virtual veremos el proceso de instalación de un sistema operativo, concretamente Ubuntu.

Mostraremos también una forma alternativa a la virtualización para tener varios sistemas operativos instalados en un mismo equipo: los gestores de arranque.

Y terminaremos hablando de las herramientas que ofrecen los sistemas operativos para ayudarnos con su utilización y para realizar las actualizaciones periódicas necesarias para un correcto funcionamiento.

## INTRODUCCIÓN

Tal como se comentaba al inicio de la unidad anterior, para desarrollar una aplicación es preciso conocer el sistema informático sobre el que posteriormente se va a ejecutar. La Unidad 1 se centraba en la componente hardware de dichos sistemas, mientras que en esta Unidad 2 vamos a revisar a fondo la componente software. En especial los sistemas operativos, que en última instancia es sobre los que trabajarán nuestras aplicaciones.

Se hará un recorrido por los sistemas operativos que podemos encontrar en nuestro trabajo cotidiano, analizándolos y comparándolos para ver sus principales diferencias. Esto nos permitirá tener unos criterios claros cuando vayamos a decidir sobre qué plataforma queremos desarrollar, y a qué plataforma irán dirigidas nuestras aplicaciones.

Para facilitar las pruebas con diferentes sistemas operativos, y entendiendo que por regla general contaremos con un solo ordenador para nuestro uso personal, veremos dos técnicas que permiten instalar en dicho ordenador tantos sistemas operativos como necesitemos: la virtualización y los gestores de arranque.

Obviamente, aunque con la experiencia mejorará nuestro conocimiento de los sistemas operativos, a menudo será necesario consultar el funcionamiento concreto de una determinada orden, y sus distintas posibilidades. Por ello es preciso conocer la organización de la ayuda de dichos sistemas.

## **CASO INTRODUCTORIO**

Comienzas a trabajar en una empresa de servicios informáticos y te piden instalar una distribución Linux en un disco duro en el que ya se tiene instalado Windows 8. Es decir, se desea tener varios sistemas operativos en el mismo disco duro, ¿es eso posible?

Al finalizar la unidad el alumnado conocerá los principales sistemas operativos, identificará las características de las principales distribuciones Linux, será capaz de realizar la instalación de una distribución Linux habiendo comprobado previamente si se cumplen los requisitos mínimos, será capaz de instalar varios sistemas operativos en un mismo equipo y conocerá los mecanismos de ayuda que proporciona Linux.

# 1. PRINCIPALES SISTEMAS OPERATIVOS

En el tema anterior se hablaba de la importancia de los Sistemas Operativos, uno de los componentes software fundamentales dentro de un Sistema Informático.

Dentro de los Sistemas Operativos existen varias familias:

- **Windows.** Sistemas operativos propietarios, de la compañía Microsoft,
- **Mac OS.** Sistemas operativos propietarios, de la compañía Apple
- **Unix/Linux.** Sistemas operativos libres. En este tema, después de revisar brevemente las dos primeras familias, vamos a centrarnos de forma práctica en la última, la de los sistemas Unix/Linux, concretamente en una de las distribuciones Linux más populares a día de hoy, Ubuntu.

## 1.1 Sistemas operativos Windows

A su vez tiene dos ramas principales, los sistemas de escritorio (dirigidos a usuarios particulares) y los sistemas de servidor (dirigidos fundamentalmente a empresas).

Por una parte, tenemos los **sistemas operativos Windows de escritorio** más populares:

- **Windows 98:** es una versión mejorada de Windows 95, aún basada en MS-DOS, el sistema operativo de consola de Microsoft. Incluía el sistema de archivos FAT32, que permitía particiones de mayor tamaño que las soportadas por Windows 95.  
Cabe destacar su **segunda edición**, que introdujo la capacidad de compartir entre varios equipos una conexión a Internet a través de una sola línea telefónica.



Imagen: Logotipo de Windows 98

- **Windows XP:** pese a que salió al mercado en 2001 y a que Microsoft ya no lo mantiene, aún se encuentra funcionando en muchos equipos antiguos, que no disponen de un hardware lo bastante potente como para utilizar un sistema más moderno.

Fue el primer sistema de esta familia que dejaba de estar basado en MS-DOS para pasar a utilizar la arquitectura de Windows NT (la de la gama de sistemas de servidores), lo que mejoró notablemente sus prestaciones respecto a sus predecesores.

- **Windows Vista:** pese a su largo tiempo de desarrollo, y a las muchas expectativas que generó, resultó un sistema bastante decepcionante para sus usuarios, ya que sus requerimientos de hardware eran bastante altos, y sufrió muchos problemas de compatibilidad, tanto con los controladores hardware como con los programas existentes.



**Imagen: Logotipo de Windows Vista**

- **Windows 7:** versión mejorada de Windows Vista, que solucionaba parte de los problemas de compatibilidad de este. Prestó especial atención a la interfaz, para volverla más cómoda para los usuarios. Resultó un sistema más ligero, estable y rápido que su predecesor.
- **Windows 8:** su cambio más llamativo fue la eliminación del menú Inicio, que formaba parte de la interfaz gráfica de los sistemas Windows desde Windows 95. Esta decisión provocó bastante polémica entre los usuarios, hasta que finalmente fue implementado de nuevo en una de las revisiones posteriores. Añadió características interesantes, como el soporte para procesadores ARM (añadido a los ya clásicos de Intel y AMD), o una interfaz compatible con el uso de pantallas táctiles.



**Imagen: Logotipo de Windows 8**

- **Windows 10:** es el último de esta rama de la familia, y el que está vigente en la actualidad. Su lanzamiento, para animar a los usuarios a realizar el cambio desde sistemas anteriores, vino precedido de una fuerte campaña publicitaria. Y se ofertó de forma gratuita para los usuarios de Windows 7 y 8 durante su primer año. Su interfaz incluyó un menú Inicio actualizado. También incorporó otras características interesantes, tanto de tipo administrativo como de seguridad. Así como el nuevo navegador de Microsoft, denominado Edge.



**Imagen: Logotipo de Windows 10**



#### **ENLACE DE INTERÉS**

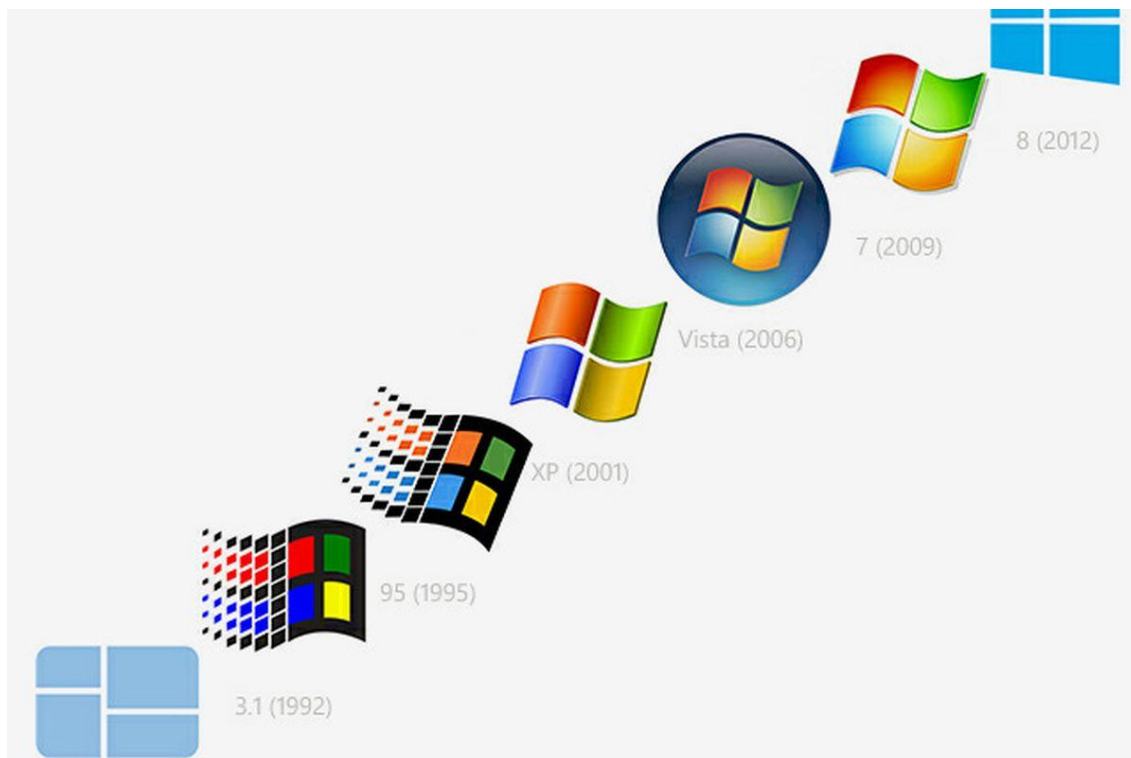
En el siguiente enlace se ofrece una breve historia de los sistemas operativos Windows, desde sus orígenes a la actualidad:

- [El Sistema Operativo WINDOWS: desde sus inicios a la actualidad](#)

En el siguiente enlace se ofrece un análisis más detallado de Windows 10:

- [Windows 10, análisis: el sistema operativo más completo de la historia de Microsoft](#)





**Imagen: La familia de sistemas operativos Windows de escritorio**

Por otra parte, se encuentran los **sistemas Windows de servidor**:

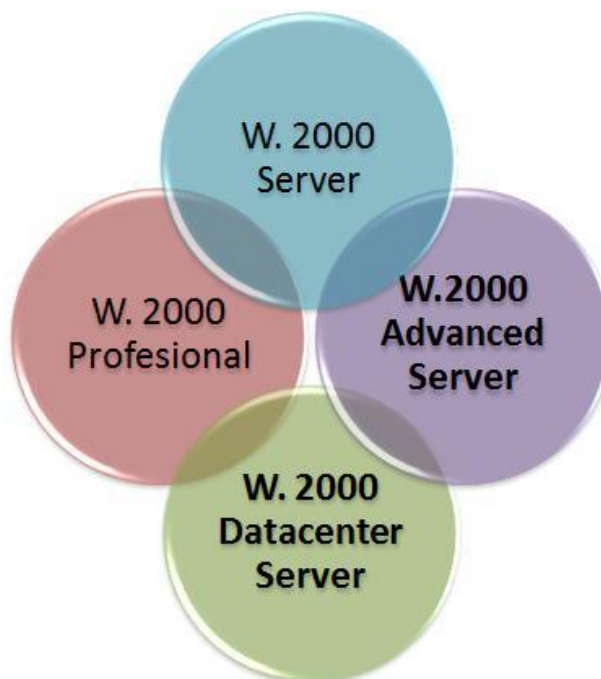
- **Windows 2000 Server**: basado en la tecnología de Windows NT, surge en febrero de 2000 y se destinaba a ser el servidor de archivos, impresión, web, FTP para empresas medianas y pequeñas.



**Imagen: Logotipo de Windows 2000 Server**

Este sistema, como los posteriores, se presenta en varias versiones, que ofrecen diferentes funcionalidades.

Las versiones de Windows 2000 son:



**Imagen: Versiones de Windows 2000 Server**

- **Windows Server 2003:** esta versión apareció en el año 2003 para sustituir a Windows 2000 Server. Desarrollado a partir de Windows XP Profesional, al que se le han añadido nuevos servicios convirtiéndolo en un Sistema Operativo para servidores.



**Imagen: Logotipo de Windows Server 2003**

- **Windows Server 2008:** se basa en el núcleo Windows NT 6.0 Service Pack 1. Las mejoras más destacables que introduce son:
  - Nuevas funcionalidades para el Active Directory.
  - Nuevas prestaciones de virtualización y administración de sistemas.
  - Servidor web IIS 7.5
  - Soporte para más de 256 procesadores.

- Cierre limpio de servicios.
- Server Core.
- PowerShell.
- Virtualización de Windows Server.



**Imagen: Logotipo de Windows Server 2008**

- **Windows Server 2012:** ofrece a las empresas y proveedores de servicios una infraestructura escalable, dinámica, para varios usuarios y optimizada para la nube. Windows Server 2012 permite que las organizaciones se conecten de manera segura entre las diferentes instalaciones y sirve para que los profesionales de TI respondan en forma más rápida y eficaz a las necesidades de negocio. Características destacadas:



- **Windows Server 2016:** mantiene las principales características de Windows Server 2012 y presenta una interfaz muy similar, aunque añade algunas prestaciones interesantes como:

- **Nano Server.** Una nueva opción de instalación, con incluso menos requisitos de hardware que Server Core.
- **Windows Server Containers e Hyper-V Containers.** Pretende aislar las aplicaciones del entorno del sistema operativo e incluso entre ellas, mejorando la estabilidad y seguridad.
- **Virtualización anidada.** Permite ejecutar Hyper-V en un equipo virtual, y crear y ejecutar equipos virtuales dentro del mismo permitiéndose un segundo nivel de virtualización.
- **PowerShell Direct.** Permite ejecutar comandos de PowerShell contra un equipo virtual sin preocuparse por la configuración de administración remota, la configuración de red o la configuración del firewall de Windows.



**Imagen: Logotipo de Windows Server 2016**

## 1.2 Sistemas operativos Macintosh

El sistema operativo de la compañía Apple, creado para sus equipos Macintosh, se denomina de forma genérica **Mac OS**.

Es conocido por haber sido uno de los primeros sistemas dirigidos al gran público en contar con una interfaz gráfica compuesta por la interacción del ratón con ventanas, iconos y menús.

Su primera versión, desarrollada completamente por Apple, fue lanzada en 1985. La última versión con estas características sería la versión 9, de 1999. A partir de la versión 10 (denominada **Mac OS X**), el sistema cambió su arquitectura totalmente y pasó a basarse en Unix, aunque su interfaz gráfica mantiene muchos elementos de las versiones anteriores.



**Imagen: Logotipo de Mac OS X**



#### ENLACE DE INTERÉS

En el siguiente enlace se ofrece información más detallada acerca del sistema operativo de Macintosh:

- [Mac OS](#)

## 1.3 Sistemas operativos Unix/Linux

Es en el segmento de los servidores donde Linux exhibe toda su potencia y fiabilidad como sistema operativo. Para estudiar los sistemas operativos en red, se puede empezar estudiando Linux. De hecho Linux está basado en Unix que fue el primer sistema operativo donde se implementaron los protocolos TCP/IP y que ofreció los primeros servicios de red basados en dichos protocolos, como telnet, ftp, smtp, etc... A diferencia de otros sistemas operativos, Linux, desde su nacimiento, está asociado inevitablemente con el software libre. De hecho las diferentes versiones que se han ido desarrollando de su principal componente, el núcleo o kernel, se ofrecen bajo la conocida licencia GPL que permite su libre distribución.

**GPL (General Public License):** es una de las licencias de software libre más utilizadas. El software distribuido bajo esta licencia permite su redistribución tanto binaria como del código fuente y permite llevar a cabo modificaciones sin ninguna restricción.

El **kernel** de Linux se encarga fundamentalmente de gestionar los recursos del equipo, se comunica con los diferentes dispositivos, gestiona programas en ejecución, administra la memoria, el acceso a los sistemas de almacenamiento y el uso del microprocesador... Sin embargo, para dotar al sistema de una funcionalidad mínima, es necesaria la existencia, junto con el kernel, de otros programas y aplicaciones de usuario.

Durante el primer período de la existencia de Linux, se podían encontrar en multitud de servidores web todos los archivos y programas necesarios para el funcionamiento de Linux. Sin embargo, y debido a que la tarea de reunir dichos archivos y programas e instalarlos correctamente podía ser bastante compleja, aparecieron empresas que se dedicaron a hacer este trabajo. Y aquí surgió el concepto de distribución.

### 1.3.1 Distribuciones comerciales de Linux

Como hemos visto, una **distribución** no es más que una recopilación de programas y archivos (incluyendo siempre la última versión estable del núcleo o kernel), organizados y preparados para su instalación. La mayoría del software de todas estas distribuciones tiene licencias GPL.

Actualmente existen un gran número de distribuciones, la mayor parte de ellas orientadas al uso de sistema operativo de escritorio o monopuesto, que incluyen entornos gráficos de escritorio y aplicaciones destinadas al usuario final, como navegadores web, herramientas ofimáticas, reproductores de música, video...

Sin embargo, y debido a la alta modularidad que tiene el software instalado en los sistemas Linux, realmente cualquier distribución se podría adaptar a su uso como sistema operativo orientado a proporcionar servicios en red. Simplemente habría que instalar y configurar el software que proporcione los servicios de red requeridos.

Por otra parte, algunas de las empresas que comenzaron desarrollando distribuciones Linux han reorientado su negocio precisamente al segmento de los servidores, ofreciendo productos basados en Linux pero con diferentes aplicaciones propias orientadas a su uso como servidores.

Los dos ejemplos más representativos de esta situación son Red Hat y SUSE. Ambas empezaron ofreciendo sus propias distribuciones de Linux, que incluían software propio, sin embargo en la actualidad, su actividad está basada en productos del segmento de los servidores orientados a proporcionar servicios de red.

- **Red Hat:** es una de las distribuciones más populares e importantes de Linux, La empresa Red Hat Inc., responsable de la distribución nació en 1994 y durante mucho tiempo ha sido una de las distribuciones de referencia en el software libre.



**Imagen: Logotipo de Red Hat**

A partir del año 2003, la estrategia de Red Hat ha sido volcar sus esfuerzos en las versiones más comerciales orientadas a empresas. En esta línea, Red Hat se ha centrado en su versión para empresas

Red Hat Enterprise Linux, mientras que el desarrollo de la parte no comercial dedicada al software libre ha sido asumido por un proyecto independiente llamado Fedora Project.

Las versiones de Red Hat son:

- Red Hat Enterprise Linux Server: diseñada para implantaciones de pequeño tamaño.
  - Red Hat Enterprise Linux Advanced Platform: proporciona un entorno más rentable, flexible y escalable.
  - Red Hat Enterprise Linux for Mainframes: versión para macrocomputadores.
  - Red Hat Enterprise Linux for HPC (High Performance Computing): versión para clusters de computación de alto rendimiento. Los sistemas de clusters son asociaciones de servidores (nodos) trabajando en paralelo. Algunas configuraciones de cluster funcionan como espejo, es decir, un nodo funciona como servidor y otro es una réplica del primero. Esta característica ofrece una alta disponibilidad del sistema.
- **Suse:** es una distribución cuyo enfoque ha sido desde sus inicios claramente comerciales, pero siempre desde el marco del software libre. Una práctica habitual de SUSE ha sido publicar versiones comerciales que se diferenciaban de la versión libre por incluir una extensa documentación impresa y un período de soporte.



**Imagen: Logotipo de SuSe**

En el año 2004, la empresa Novell compró SUSE y a partir de ese momento comienza una política parecida a la llevada a cabo en Red Hat, es decir, SUSE se centra en distribuciones comerciales destinadas a empresas y libera una distribución para que la comunidad de desarrolladores de Linux sea la encargada de futuros desarrollos. Esta última toma el nombre de openSUSE.



La versión para empresas que ofrece SUSE se llama SUSE Linux Enterprise Server.

- **Mandriva:** la distribución Mandrake Linux apareció en 1998 y estaba basada a su vez en la distribución Red Hat. Posteriormente la distribución cambió su nombre a Mandriva.



**Imagen: Logotipo de Mandriva**

Su filosofía inicial era ofrecer un sistema robusto, flexible y fácil de utilizar aunque sin perder toda la potencia de un sistema Linux. Esto se ha mantenido, convirtiéndose actualmente en una de las distribuciones más fáciles y preferida por los usuarios nóveles. Actualmente, esta distribución tiene abierta una línea de negocio para empresas, otra para socios con software comercial y otra de libre distribución.

Algunas de sus versiones son Mandriva Linux Free y Mandriva Enterprise Server.

### ***1.3.2 Distribuciones gratuitas de Linux***

Algunas de las más importantes son:

- **Debian:** el Proyecto Debian parte de una asociación de personas que han hecho causa común para crear un sistema operativo libre. Utiliza el núcleo Linux (el corazón del sistema operativo), pero la mayor parte de las herramientas básicas vienen del Proyecto GNU; de ahí el nombre GNU/Linux.

Debian GNU/Linux ofrece más que un S.O. puro; viene con 25000 paquetes, programas precompilados distribuidos en un formato que hace más fácil la instalación en su computadora.



**Imagen: Logotipo de Debian**



- **Ubuntu:** cualquier distribución Linux es susceptible de convertirse en un sistema operativo de servidor, basta con instalar el software adecuado que proporcione los servicios deseados. Por ejemplo, para que un sistema Linux se convierta en un servidor web sólo hay que instalar el software que proporcione este servicio, por ejemplo, Apache.

Ubuntu es una distribución basada en Debian y que ha conseguido una gran popularidad gracias a mantener las características de robustez y fiabilidad de Debian pero intentando ser una distribución más amigable que esta.

Ubuntu nace de la iniciativa de algunos programadores de Debian y GNOME que fundan la empresa Canonical Ltda.



Imagen: Logotipo de Ubuntu



#### ENLACE DE INTERÉS

En el siguiente enlace se ofrece una comparativa de las distribuciones Linux más populares, que puede servir como guía a la hora de elegir una de ellas:

- [31 distribuciones Linux para elegir bien la que más necesitas](#)

## 2. VIRTUALIZACIÓN

Cuando se plantea la necesidad de instalar un nuevo sistema operativo, como en el caso introductorio, resulta muy útil poder probarlo previamente sin alterar la configuración de nuestro sistema. Como se verá a continuación, cualquier instalación tradicional de un sistema operativo implica cambios importantes en el sistema (particiones, sistema de archivos...). Cambios que, si no se gestionan de manera adecuada, pueden llegar a inutilizar dicho sistema.

Por eso, una práctica cada vez más extendida es el uso de la **virtualización**, una técnica que nos permite instalar y ejecutar un sistema operativo completo dentro de otro sistema ya instalado (que se denomina anfitrión), accediendo a sus recursos pero sin modificar la configuración básica de este.

Son numerosas las **ventajas** que se pueden encontrar a la hora de crear una máquina virtual en nuestro equipo. Entre ellas destacamos:

- Aprovechar los recursos de la máquina anfitriona.
- Usar más de dos sistemas operativos a la vez en el mismo ordenador y al mismo tiempo.
- Pruebas de aplicaciones en la máquina virtual, sin dañar la anfitriona, previniendo de posibles virus o programas maliciosos.
- Aprender el manejo de diferentes sistemas operativos.
- Simular situaciones con diferentes máquinas son tener que comprar más recursos.
- Tener varios sistemas operativos sin particionar el disco duro.
- Están aisladas entre sí. Si entra un virus en la máquina virtual no afectará al resto de máquinas o a la máquina anfitriona.
- Se comporta como una máquina física.
- Compatible con muchos sistemas operativos.
- Es independiente del hardware físico. Mientras el anfitrión dispone de un disco duro físico, la máquina virtual puede disponer de varios discos duros virtuales (**IDE** o **SCSI**)

En cuanto a los **inconvenientes** de las máquinas virtuales:

- Existe una limitación de recursos, ya que se basan en los recursos de la máquina anfitriona.
- Suele ser más lenta que la anfitriona.
- Suele consumir muchos recursos de la máquina anfitriona.
- Ocupa mucho espacio en el disco duro de la máquina anfitriona.

Sin duda las ventajas son mayores que los inconvenientes, así que se va a emplear esta técnica en las instalaciones que llevemos a cabo en adelante,

pudiendo disponer así en nuestras máquinas de un conjunto de sistemas operativos sin que haya ningún conflicto entre ellos.

## 2.1 VirtualBox

Se trata de un software libre patrocinado por Oracle. Es un potente producto de virtualización X86, AMD64 / Intel64 para la empresa y uso doméstico. Está disponible como software de código abierto bajo los términos de la Licencia Pública General de GNU (GPL).

Puede ejecutarse en Windows, Linux, Macintosh y OpenSolaris, destacando su fácil instalación en Windows y Linux. Soporta un gran número de sistemas operativos invitados incluyendo los diferentes sistemas operativos Windows, tanto de escritorio como de servidor, así como la mayoría de las distribuciones Linux, Solaris y OpenSolaris, OS / 2, y OpenBSD.



**Imagen: Logotipo de Virtual Box**

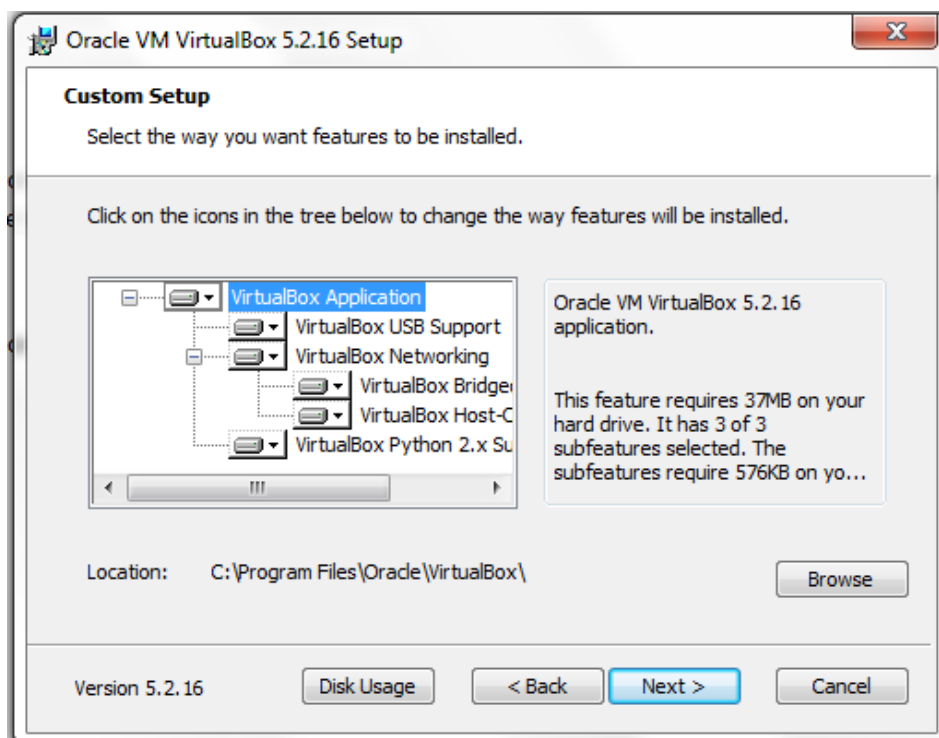
Antes de nada hay que realizar la instalación de VirtualBox, que se va a describir a continuación. En nuestro caso se usará la versión 5.2.16, que se puede descargar desde la web de Oracle.

Se ejecuta el archivo de instalación que se ha descargado



**Imagen: Asistente de instalación de Virtual Box**

Se aceptan las opciones por defecto:



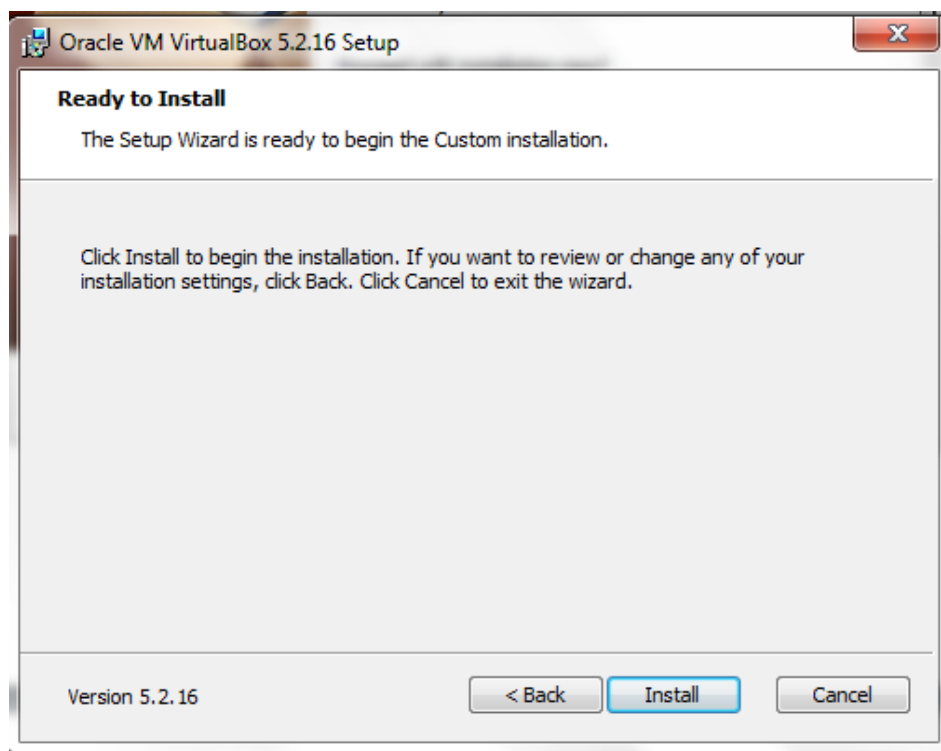
**Imagen: Selección del tipo de instalación y la carpeta de destino**

Se nos avisa de que perderemos momentáneamente la conexión de red, ya que se precisa resetearla:



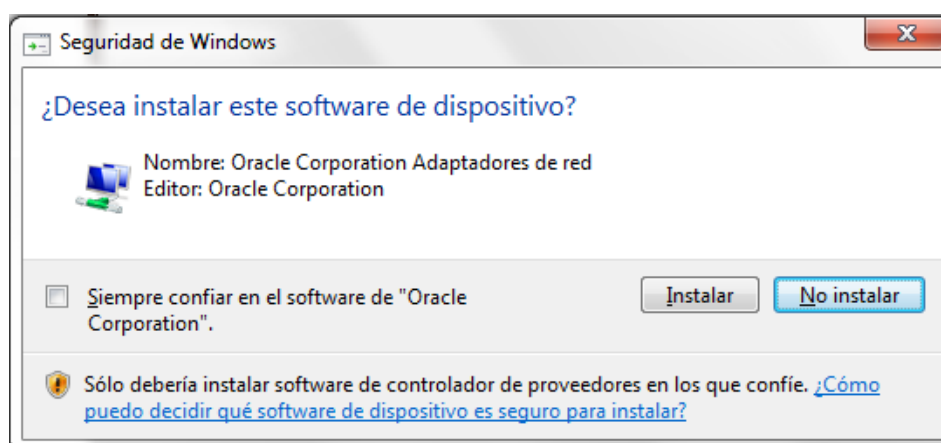
**Imagen: Aviso acerca de la conexión de red**

Se procede a instalar:



**Imagen:** Comienzo de la instalación

Durante el proceso, se nos puede solicitar la instalación de varios controladores, necesarios para la operativa de Virtual Box, que se aceptarán:



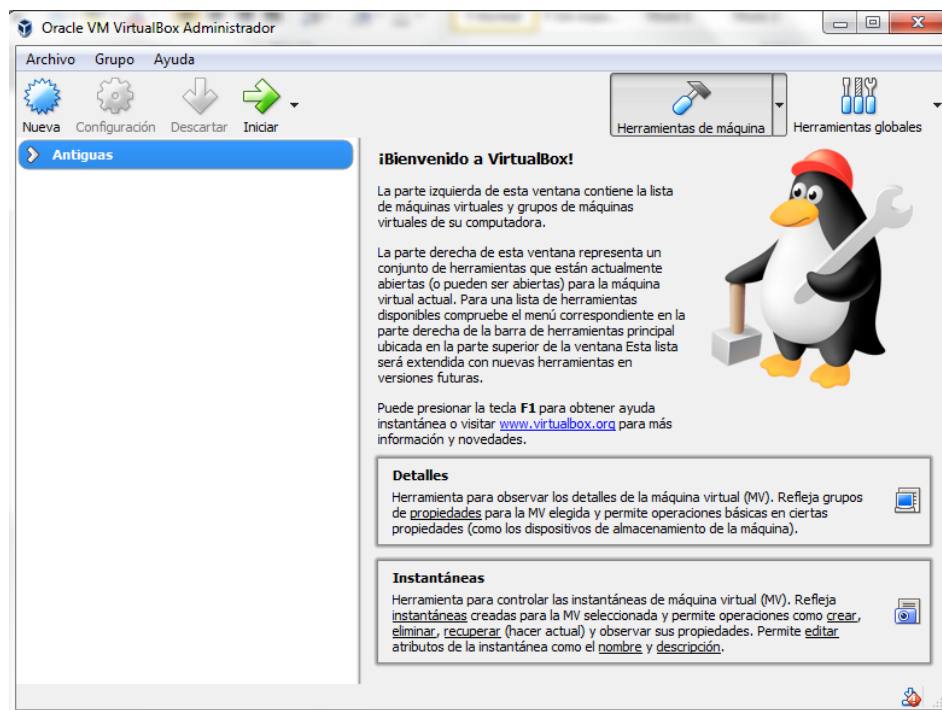
**Imagen:** Instalación de controladores de red

Tras unos minutos, la instalación queda completada



**Imagen: Fin de la instalación**

Se puede comprobar la correcta instalación accediendo al programa:



**Imagen: Pantalla de inicio de VirtualBox**

## 3. INSTALACIÓN DE UN SISTEMA OPERATIVO. UBUNTU

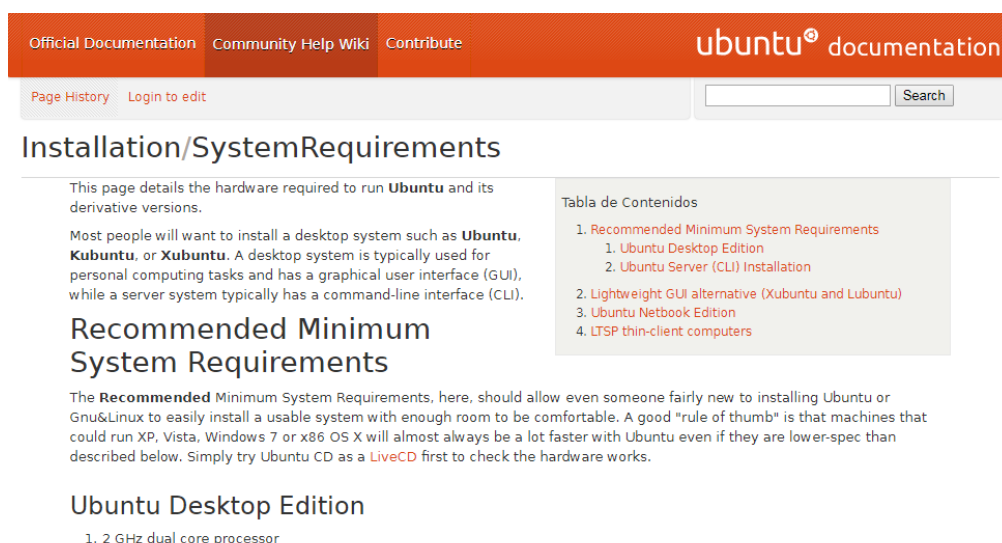
Uno de los factores que siempre hay que considerar antes de la instalación de un sistema operativo, bien sea directamente en un ordenador, bien mediante una máquina virtual, es la especificación de sus requisitos mínimos, que indican el hardware necesario para que dicho sistema funcione correctamente.

### 3.1 Requisitos mínimos de Ubuntu

GNU/Linux, como hemos visto, no es un SO como tal, es el nombre que recibe el kernel que da sustento a una serie de distribuciones, que son lo que sí que podríamos llamar un sistema operativo. Por esta razón, hablar de requerimientos mínimos de Linux en cuanto a hardware es tan inexacto como hablar de requerimientos mínimos de Windows en cuanto a Hardware, ya que no es lo mismo los requerimientos de un Windows XP que los requerimientos de un Windows 7, y mucho menos los de un Windows 10 en sus versiones superiores.

A estas alturas nadie duda de las ventajas de los sistemas basados en Linux, ya que son por lo general muy fiables, muy seguros, cada vez más fáciles de manejar, con unas interfaces gráficas cada vez mejores, con un buen aprovechamiento de los recursos de hardware y, sobre todo, son de software libre y, en la mayoría de los casos, gratuitos. Pero de esto a decir que funcionan en cualquier máquina hay una buena diferencia.

Las distribuciones de Linux necesitan, como todo SO, unos requisitos mínimos de hardware, que dependerán de esta. Por regla general, lo mejor es consultarlos en la propia web del fabricante del sistema.



The screenshot shows the Ubuntu documentation website. The top navigation bar includes links for 'Official Documentation', 'Community Help Wiki', and 'Contribute'. The main header features the 'ubuntu® documentation' logo. Below the header, there are links for 'Page History' and 'Login to edit', and a search bar. The main content area is titled 'Installation/SystemRequirements'. It contains a paragraph explaining that the page details hardware requirements for Ubuntu and its derivative versions. It then lists 'Recommended Minimum System Requirements' and provides a detailed explanation of these requirements, noting that they should allow even someone new to installing Ubuntu or GNU/Linux to easily install a usable system. The page also includes a 'Tabla de Contenidos' (Table of Contents) on the right side, listing the following sections: 1. Recommended Minimum System Requirements, 1. Ubuntu Desktop Edition, 2. Ubuntu Server (CLI) Installation, 2. Lightweight GUI alternative (Xubuntu and Lubuntu), 3. Ubuntu Netbook Edition, and 4. LTSP thin-client computers. The main content area also includes a section for 'Ubuntu Desktop Edition' with a list of requirements, starting with '1. 2 GHz dual core processor'.

**Imagen: Indicación de requerimientos en la web de Ubuntu**



Esto es lo que se especifica en el caso de Ubuntu.

<b>Procesador</b>	2 Ghz
<b>Memoria RAM</b>	2 GB
<b>Disco duro</b>	25 GB
<b>Tarjeta gráfica</b>	VGA con resolución mínima de 1024x768
<b>Dispositivos extraíbles</b>	Unidad CD/DVD Puerto USB
<b>Conexión a Internet</b>	No imprescindible, pero de mucha utilidad

## 3.2 Proceso de instalación

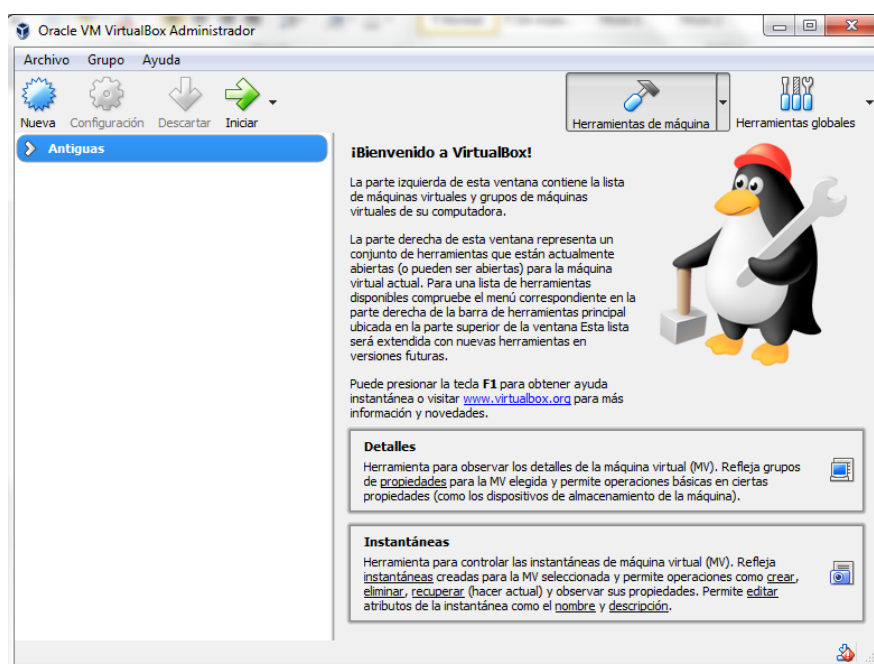
En este caso se va a instalar Ubuntu como una máquina virtual en VirtualBox, para así disfrutar de todas las ventajas comentadas en apartados anteriores.

Para obtener una distribución actualizada de Ubuntu, lo más recomendable es descargar una imagen desde su web, adecuada al equipo en el que se desea realizar la instalación. En nuestro caso se va a instalar la versión 16.04.

### 3.2.1 Creación de la máquina virtual

Para la creación de la máquina virtual, se siguen los siguientes pasos:

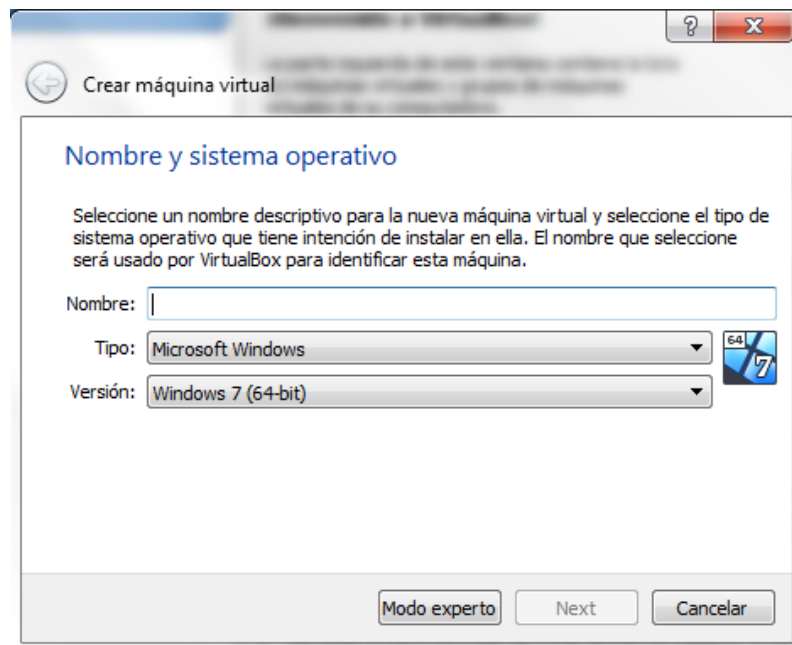
Se comienza por acceder a VirtualBox



**Imagen: Pantalla de inicio de VirtualBox**

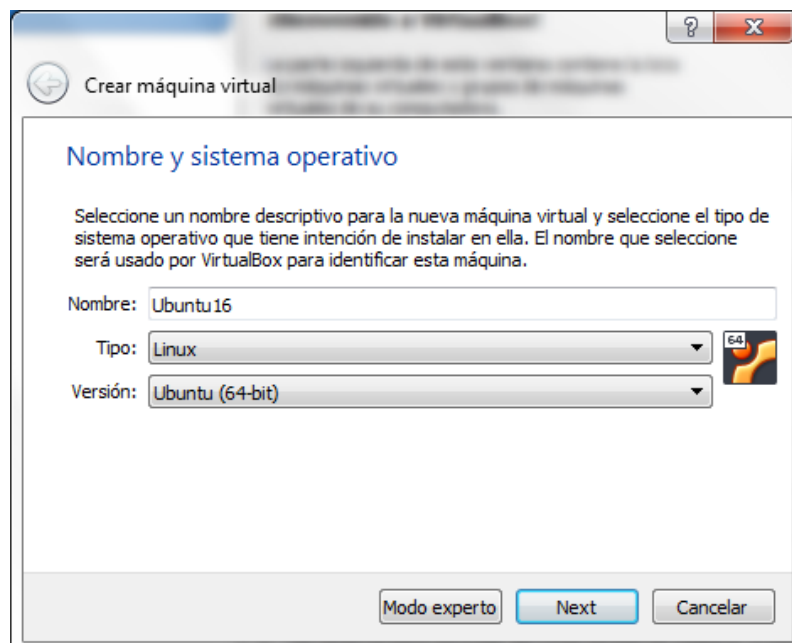


Desde la pantalla de inicio ya se puede crear una máquina virtual para Ubuntu utilizando el icono **"Nueva"**



**Imagen: Creación de nueva máquina virtual**

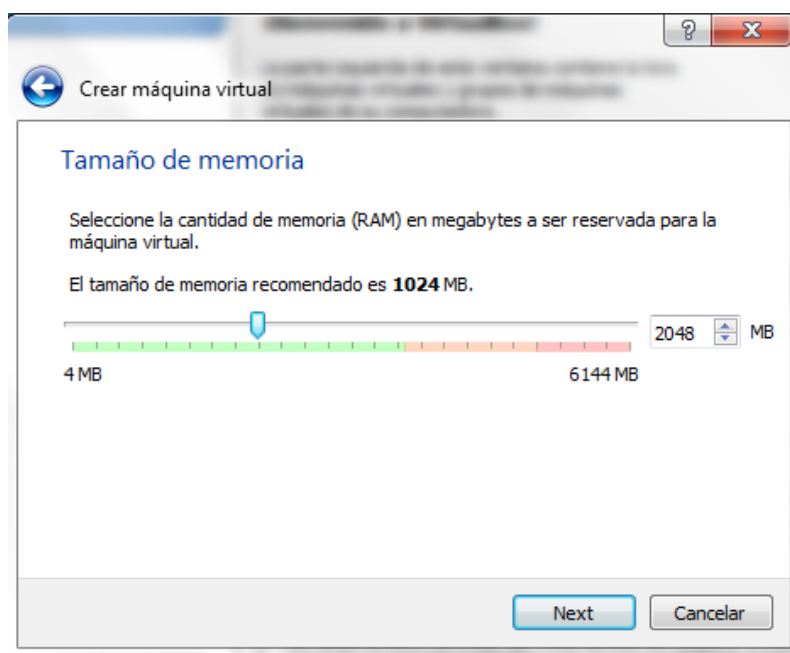
Se nombra la máquina virtual a crear, y se elige el tipo de sistema operativo y la versión deseada. En nuestro caso, como se comentó al inicio, se va a instalar la versión 16.04 de Ubuntu.



**Imagen: Elección de nombre y tipo de sistema operativo**

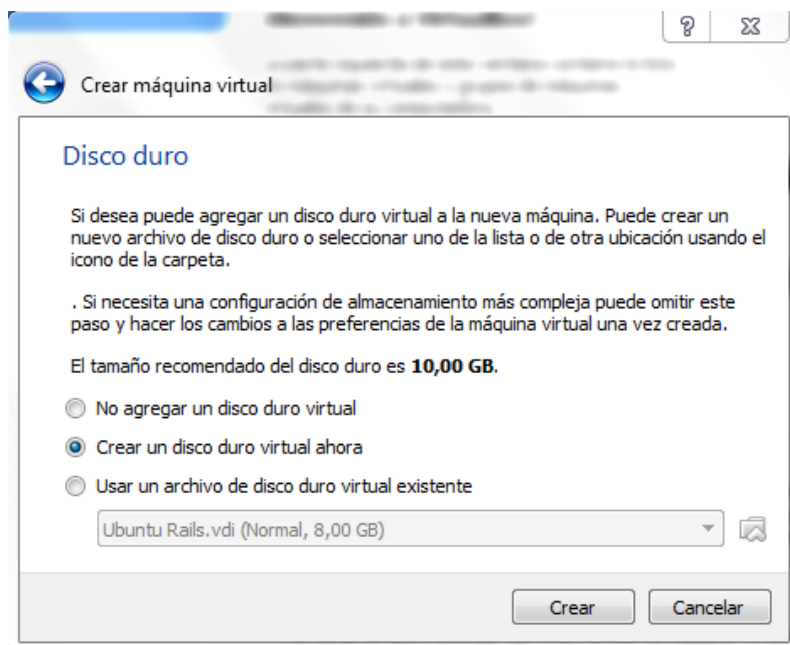
A continuación un paso fundamental: la asignación de memoria RAM a la máquina virtual. La cantidad elegida dependerá del tamaño de la memoria RAM del equipo anfitrión. Cuanta más memoria se asigne, más rápido funcionará la máquina virtual. Pero nunca se debe tomar toda la memoria disponible, pues en ese caso el anfitrión tendrá serias dificultades para seguir trabajando.

Se eligen 2GB para la nueva máquina, lo que ofrecerá un equilibrio razonable en el rendimiento de la máquina virtual y el anfitrión.



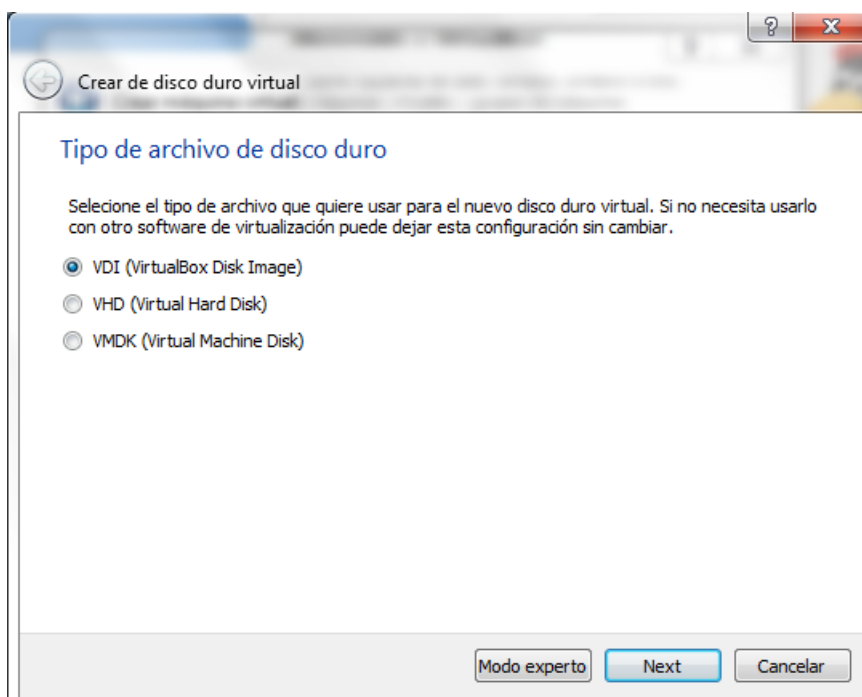
**Imagen:** Asignación de memoria RAM

El siguiente elemento a configurar es el disco duro que se va a asignar a la máquina. Como en caso anterior, dependerá de las características del sistema anfitrión, pero como norma general hay que buscar un equilibrio entre dicho anfitrión y las máquinas virtuales que contiene. En este caso se va a elegir la opción recomendada, 10 GB.



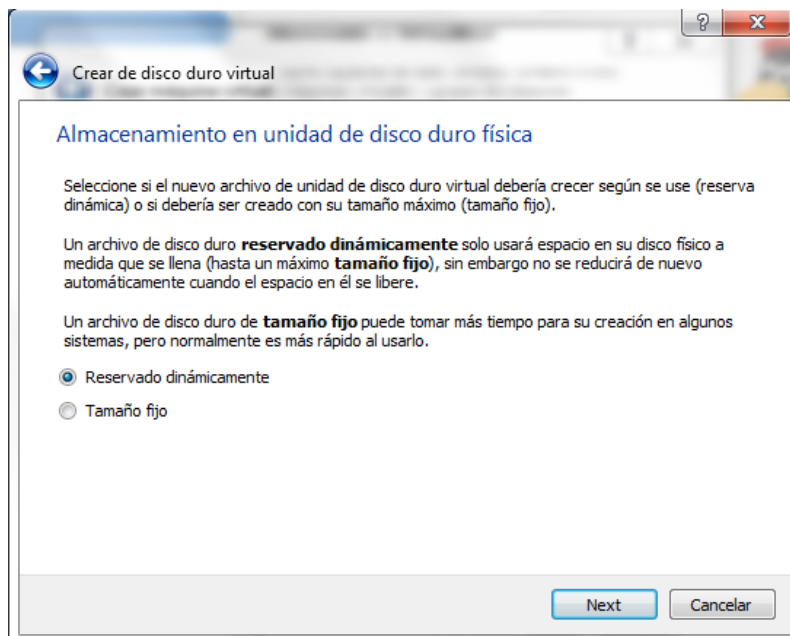
**Imagen: Creación del disco duro para la máquina virtual**

Se elige el tipo de disco a crear, optando también con la opción por defecto, VDI:



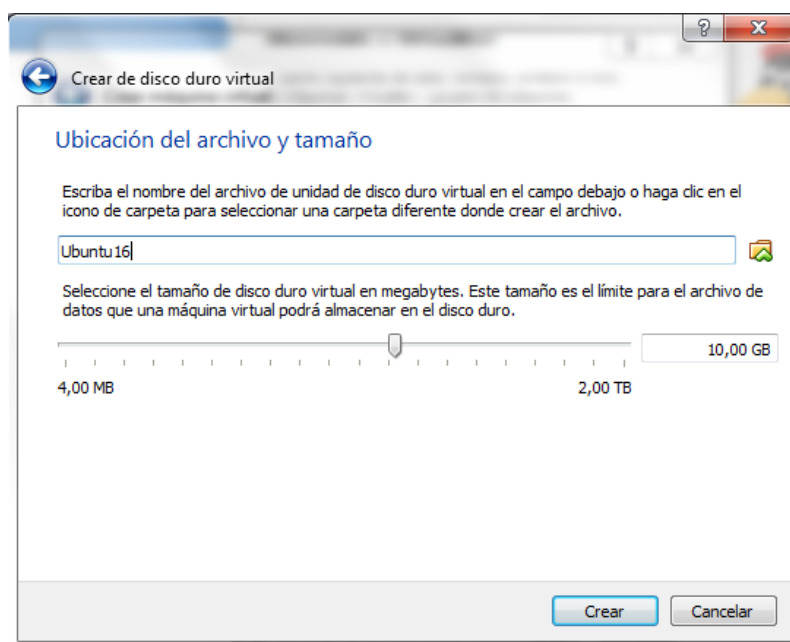
**Imagen: Elección del tipo de disco duro virtual**

Se indica que se desea un disco dinámico, de forma que sólo se vaya ocupando espacio en el disco duro del anfitrión conforme se vaya necesitando:



**Imagen: Características del disco duro virtual**

Se selecciona el nombre y ubicación para el archivo del disco duro (nuevamente se aceptan las opciones por defecto). Y también el tamaño, que como se ha indicado previamente será de 10 GB.



**Imagen: Nombre, ubicación y tamaño desde disco duro virtual**

Se pulsa **Crear**, y la máquina virtual Ubuntu 16 ya está generada.

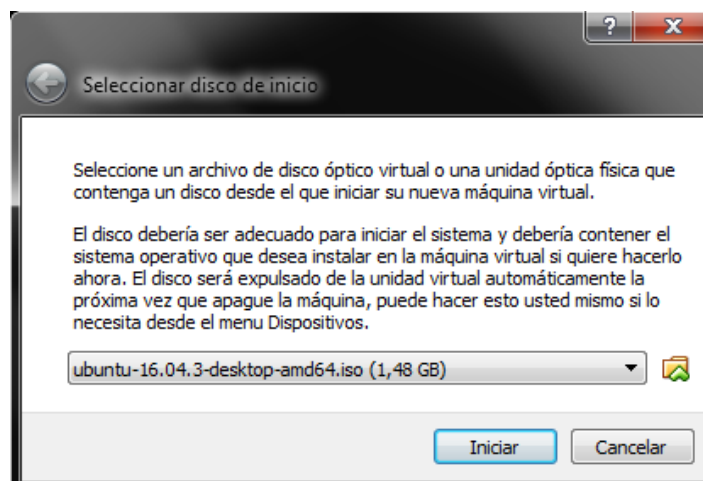


**Imagen: Máquina virtual Ubuntu16 creada**

### ***3.2.2 Instalación de Ubuntu sobre la máquina virtual***

En el apartado anterior se había terminado de crear la máquina virtual Ubuntu16. Ahora se va a poner en funcionamiento mediante el botón **Iniciar** de VirtualBox.

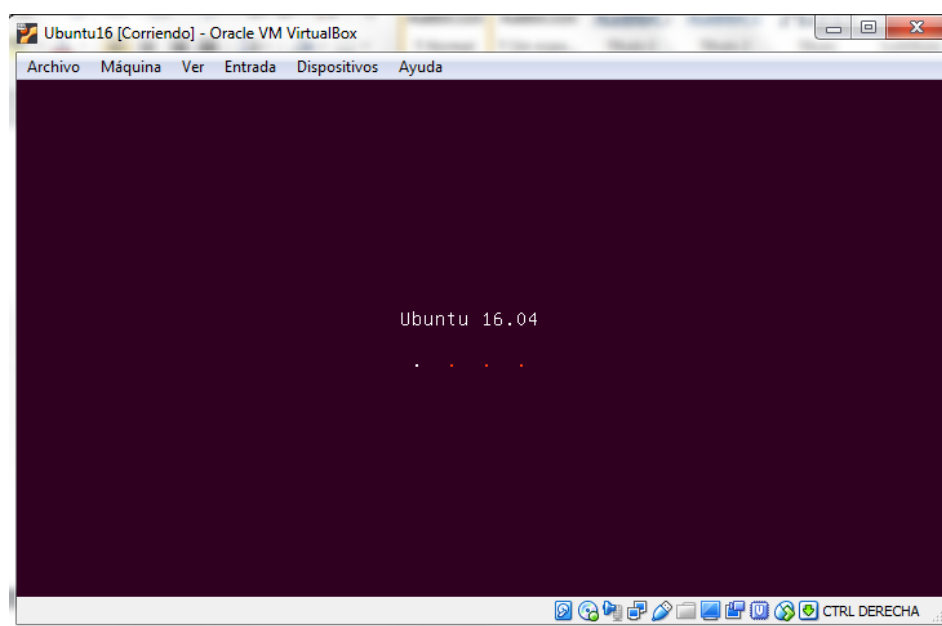
Hay que tener en cuenta que el sistema operativo deseado, Ubuntu 16, aún no ha sido instalado, sólo se ha creado una máquina virtual para él. Luego el siguiente paso será instalar dicho sistema operativo. Para ello, como ya se comentó al inicio, se necesita disponer de un disco (CD o DVD) que contenga el sistema. O, más cómodo, se puede descargar una versión ISO a partir de la cual realizar la instalación. En nuestro caso se emplea esta segunda opción, localizando en el disco duro del equipo anfitrión el archivo que contiene la citada imagen ISO.



**Imagen: Selección de la ISO de Ubuntu**

Los pasos para la instalación son los siguientes:

Se da comienzo al proceso de instalación mediante el botón **Iniciar**



**Imagen: Comienzo de la instalación de Ubuntu**

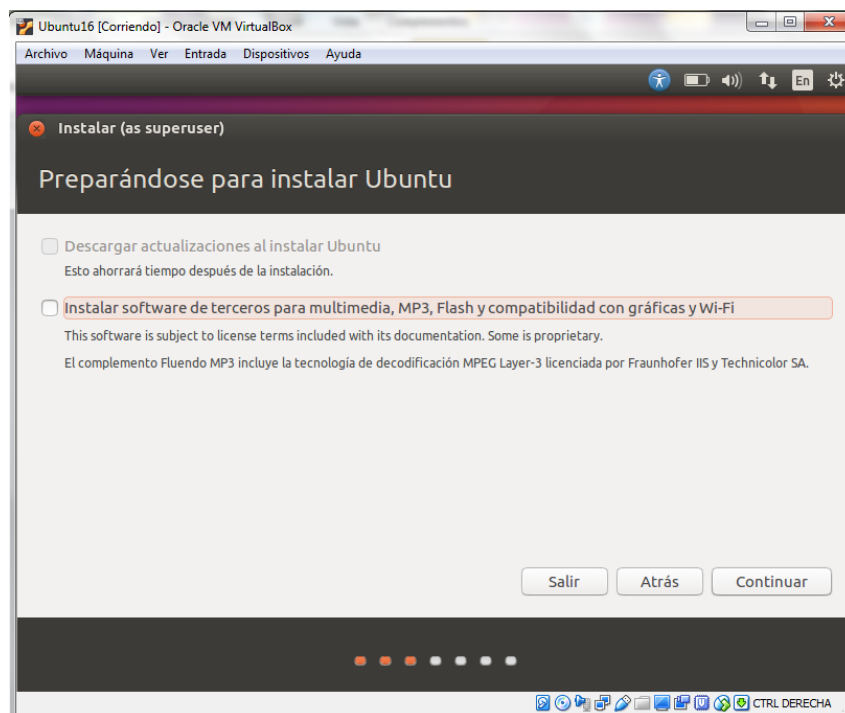
Y se obtiene una instalación idéntica a la que se realizaría directamente en un equipo cualquiera, sin virtualizar.

Se elige el idioma Español, y la opción de **Instalar Ubuntu**.



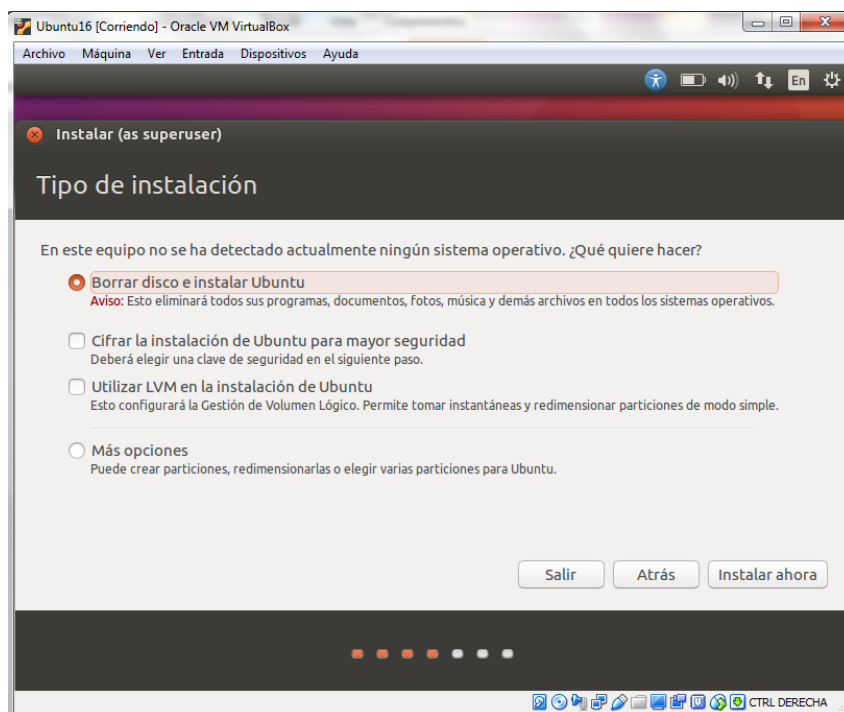
**Imagen: Pantalla inicial de la instalación**

Se va a hacer toda la instalación con las opciones por defecto, así que se pulsa **Continuar**.



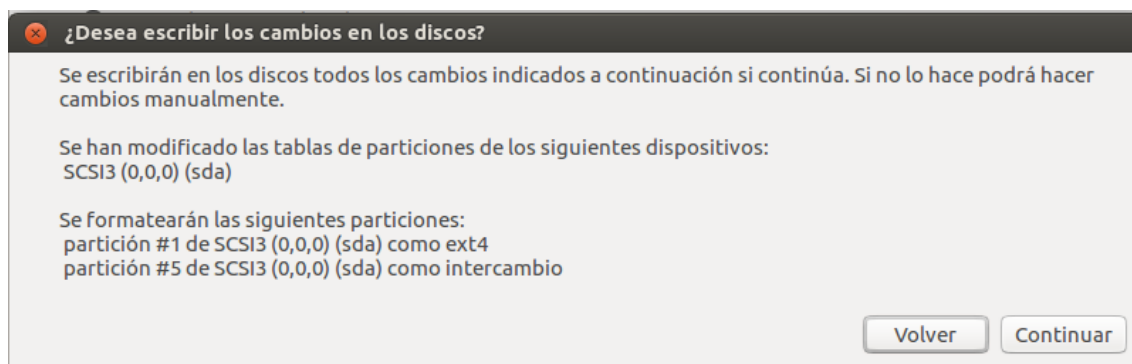
**Imagen: Preparación de la instalación**

Se borra el disco de la máquina virtual (donde ahora mismo no hay nada instalado), para instalar Ubuntu en él.



**Imagen: Selección del tipo de instalación**

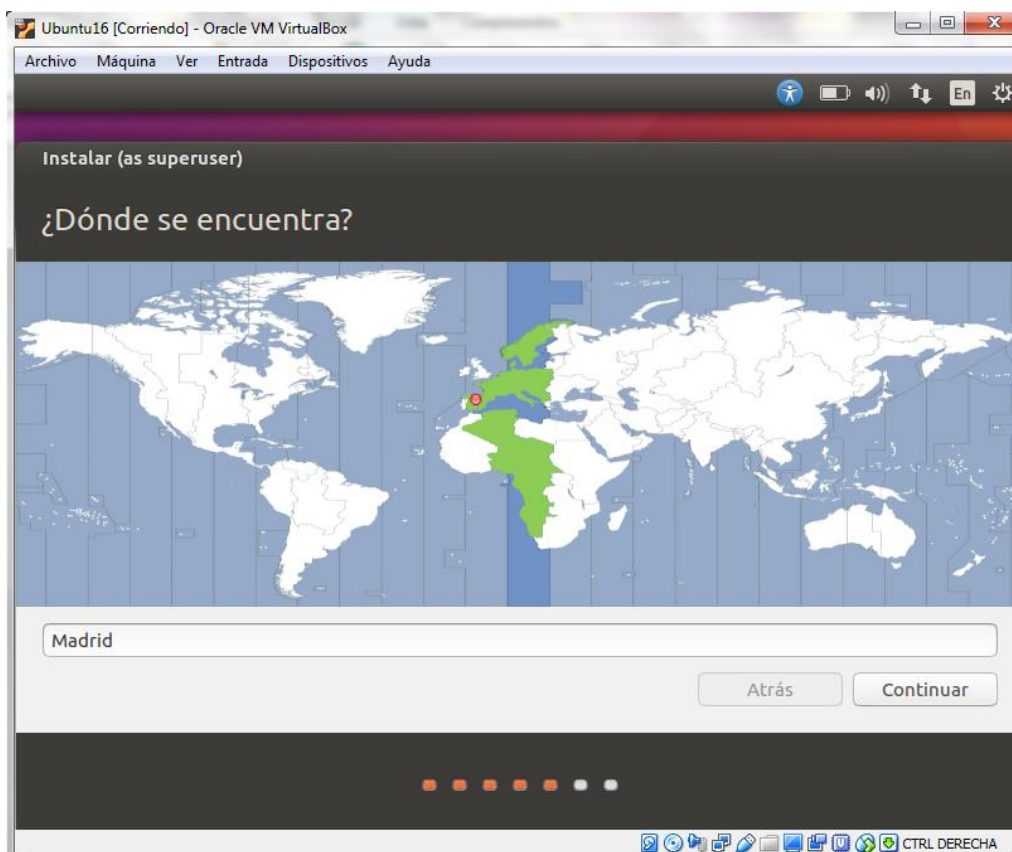
Se nos indican los cambios que se van a realizar en el disco. Concretamente se va a crear una partición de intercambio (swap), y otra principal, de tipo ext4. Se hablará de los sistemas de archivos en la siguiente unidad. Se pulsa Continuar.



**Imagen: Indicación de los cambios en el disco**

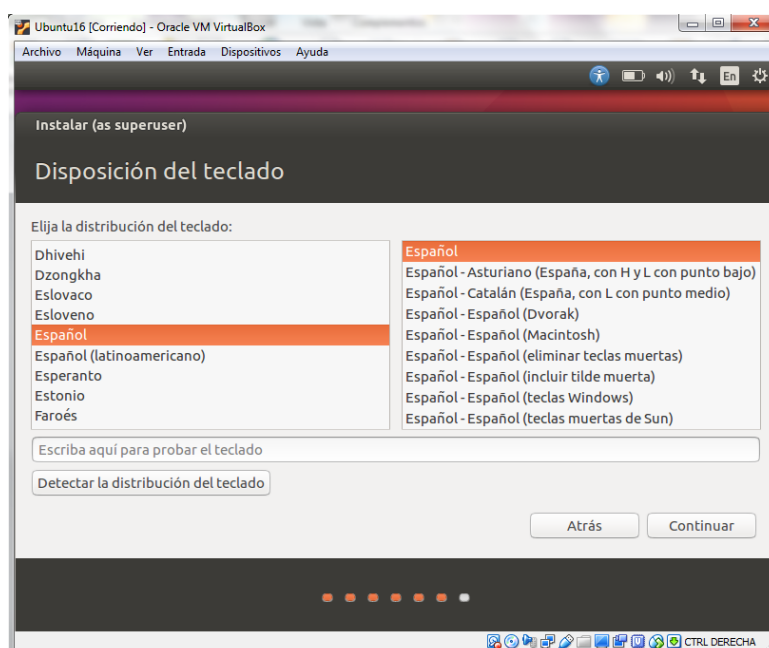
A continuación se van solicitando algunas opciones iniciales de configuración del sistema operativo Ubuntu. Se comienza indicando nuestra localización, que servirá al sistema para configurar varias opciones locales.





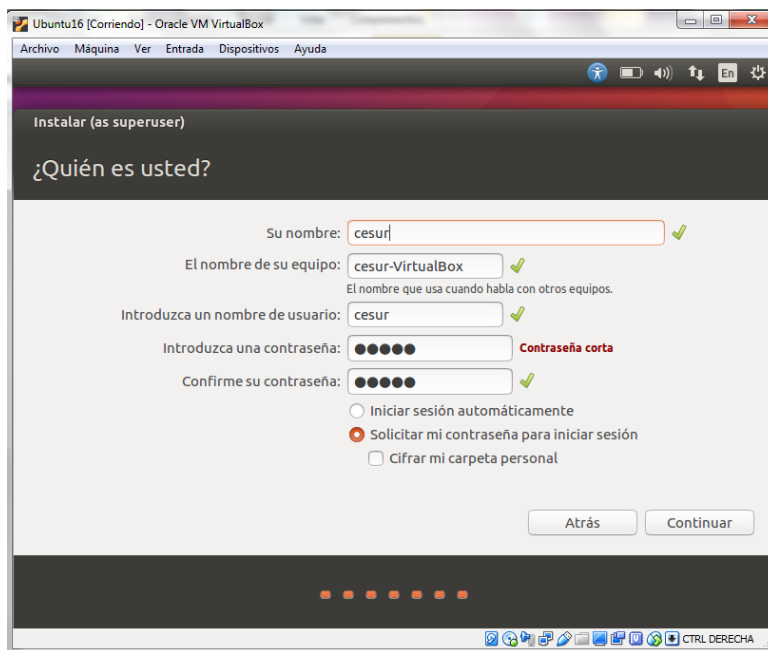
**Imagen: Primeras opciones de configuración del sistema**

Entre dichas opciones locales está la configuración del teclado, que se configura en la pantalla posterior. Es importante elegir **Español** para luego no tener problemas con las tildes y otros caracteres especiales:



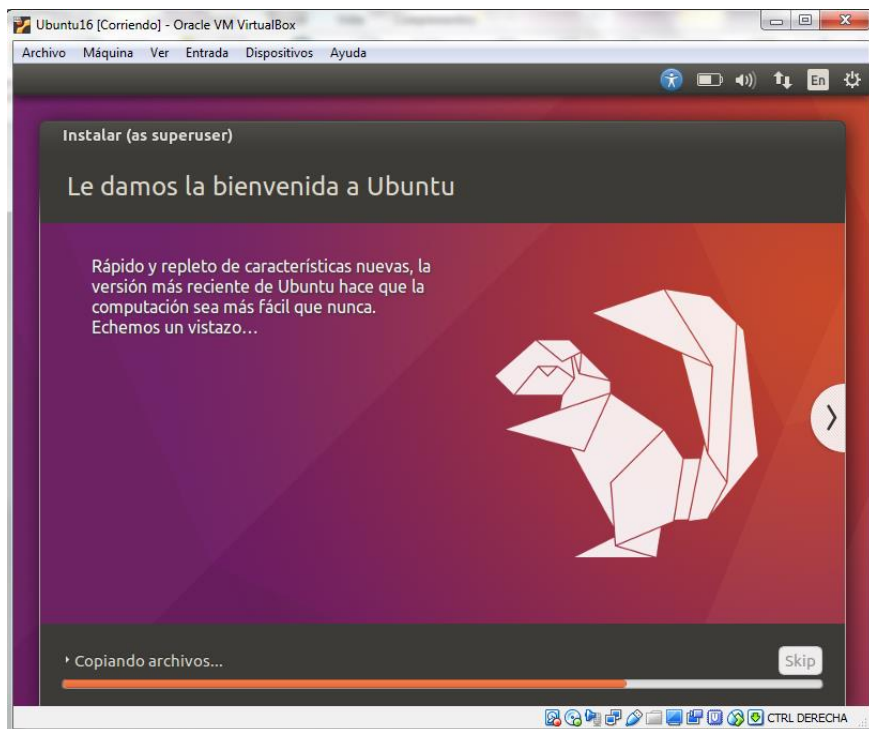
**Imagen: Configuración del teclado**

El siguiente es otro paso muy importante: la identificación. Se va a crear un usuario en el sistema, con el que luego se trabajará. En nuestro caso dicho usuario será **cesur**, que será el texto que utilizemos también como contraseña. Esto no sería recomendable en un sistema real, pero por comodidad a la hora de hacer pruebas se van a elegir usuario y contraseña iguales.



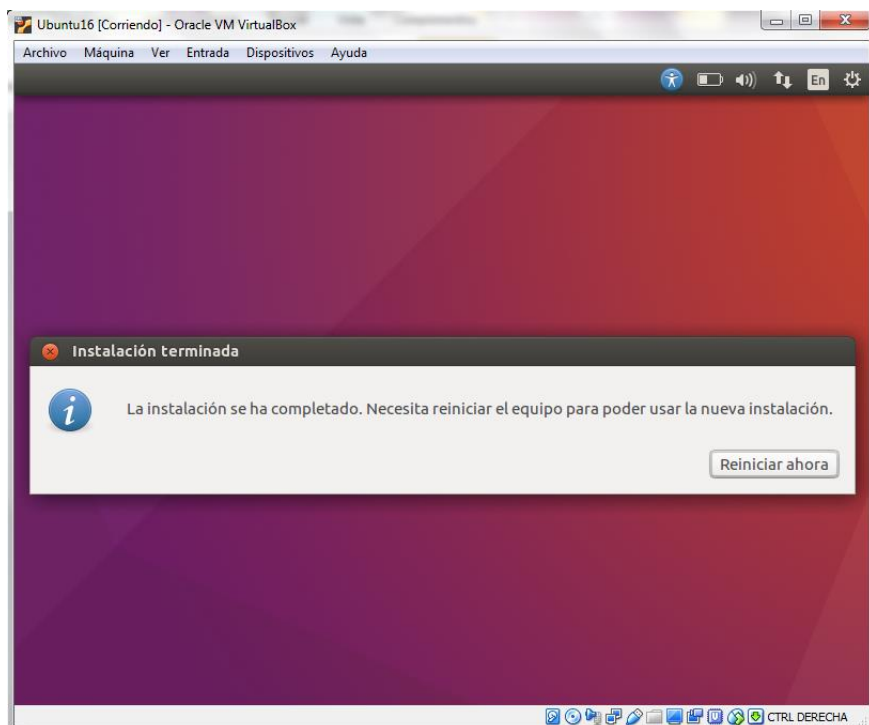
**Imagen: Identificación en el sistema. Creación de un primer usuario**

Comienza la copia de archivos, mientras se recuerdan algunas características del sistema operativo que se está instalando.



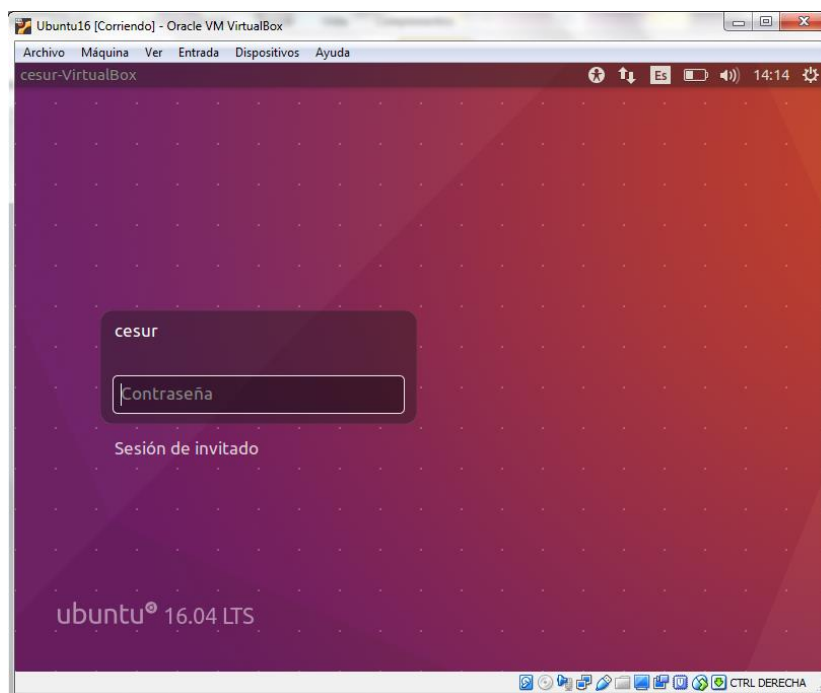
**Imagen: Comienzo de la copia de archivos**

Después de unos minutos, la instalación se completa y se solicita reiniciar el equipo (la máquina virtual en este caso).



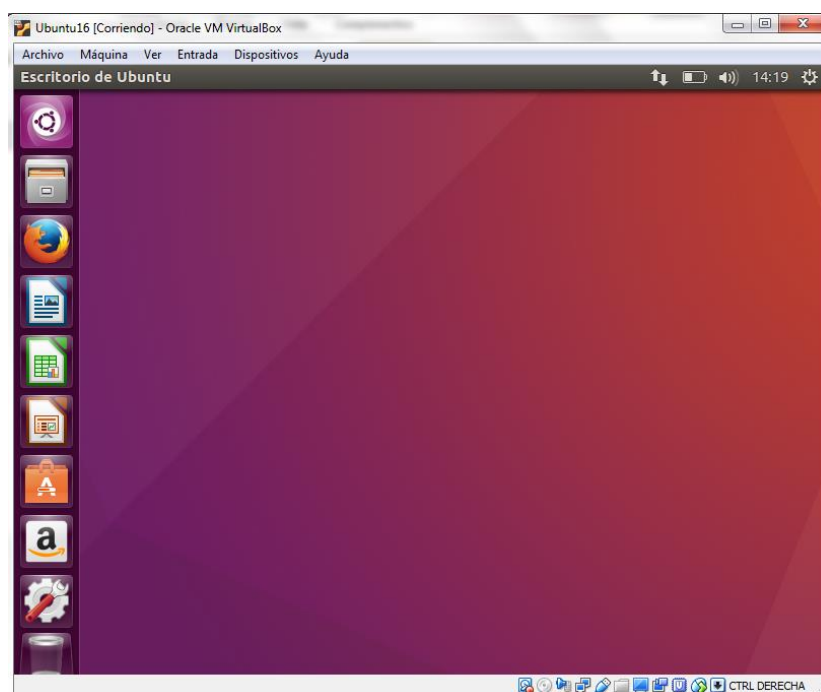
**Imagen: Instalación completada**

Reiniciada la máquina, ya se puede iniciar una sesión con el nuevo usuario y contraseña.



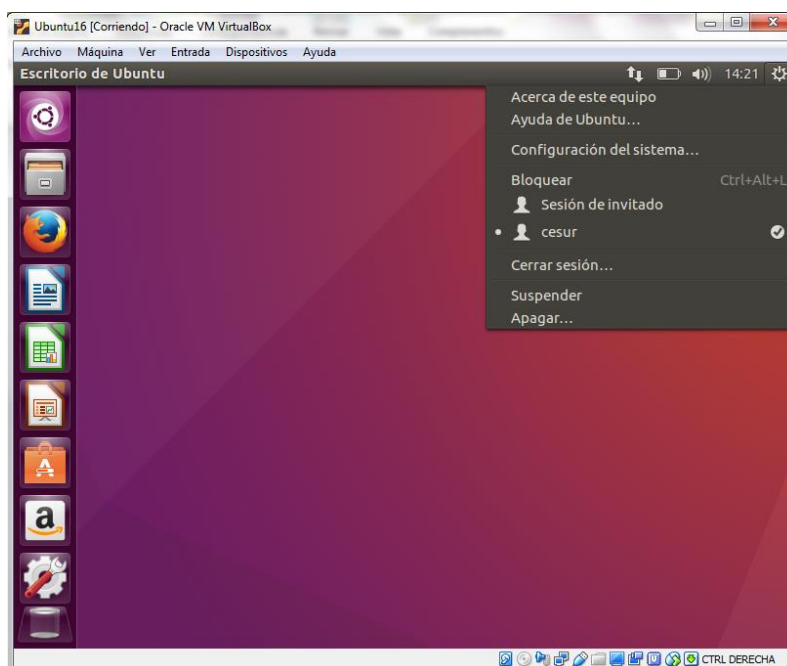
**Imagen: Primer inicio de sesión en la máquina virtual Ubuntu**

Una vez identificados, se accede a la interfaz gráfica de Ubuntu, denominada **Unity**, desde la que se podrán realizar las operaciones habituales de cualquier sistema operativo. Operaciones que se irán desglosando en las siguientes unidades.



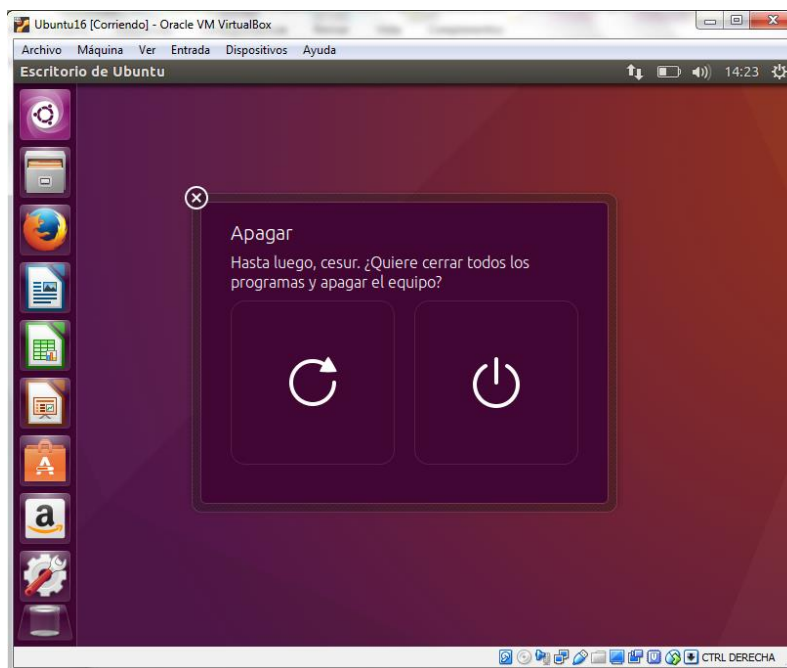
**Imagen: Interfaz Unity de Ubuntu**

Cuando se desee terminar el trabajo con Ubuntu, se seleccionará **Apagar** la máquina virtual (esta es la forma correcta de salir del sistema).



**Imagen: Salir del sistema**

Se confirma.



**Imagen: Confirmación de apagado del equipo**

Con lo que se cerrará la máquina virtual, y se vuelve al panel de administración de Virtual Box.

Y ya está instalado el nuevo sistema Ubuntu16 en su correspondiente máquina virtual.

## 4. GESTORES DE ARRANQUE

Ya se ha visto que cuando se desea tener instalados en un ordenador dos o más sistemas operativos, una opción sencilla para conseguirlo es usar la virtualización. Pero hay otra alternativa clásica para ello, que consiste en emplear un **gestor de arranque**. Gestor que nos permite, en el momento de arrancar del equipo, elegir con cuál de dichos sistemas queremos iniciar sesión.

En concreto, y a modo de ejemplo, se va a presentar el gestor de arranque utilizado por prácticamente todas las distribuciones GNU/Linux, que es GNU GRUB.

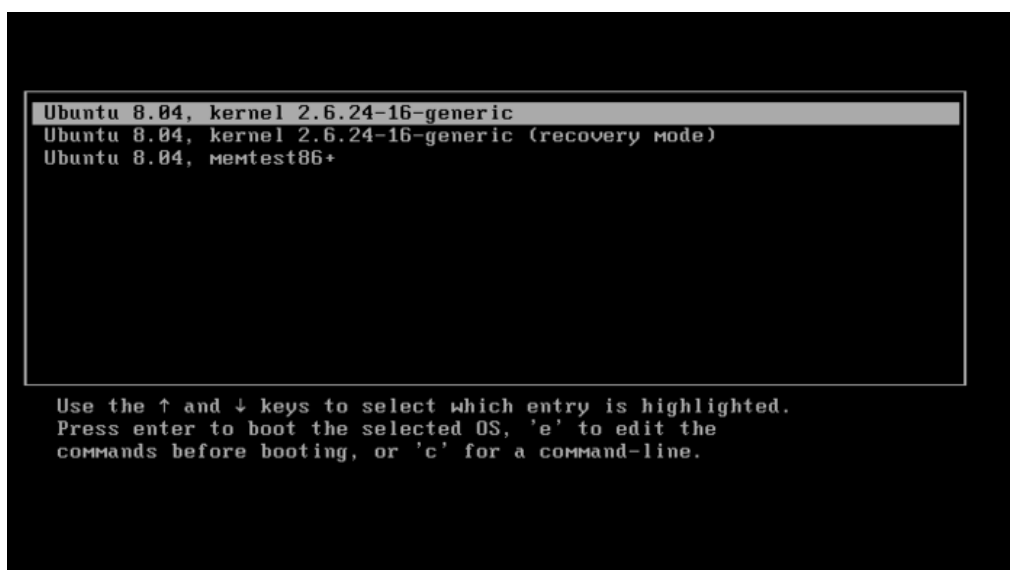


Imagen: Panel de arranque de GRUB

### 4.1 Secuencia de arranque del sistema

La secuencia de arranque del sistema, simplificada en una serie de pasos, es la siguiente:

1. Cuando el usuario enciende el ordenador (Power o n) la BIOS realiza un chequeo de los componentes hardware y utiliza la configuración establecida para comprobar determinados aspectos del equipo, como pueden ser la hora del sistema, secuencia de arranque (orden en que serán probados los periféricos disponibles y que permiten el arranque de la máquina, como discos duros, CD-ROM/DVD, memorias USB,...).
2. A continuación la BIOS carga en memoria el programa que se encuentra almacenado en el primer sector (sector 0, cuyo tamaño es de 512 bytes) del primer dispositivo en la secuencia de arranque. Se pasa el control de la máquina a dicho programa, llamado gestor de arranque, que contiene las instrucciones, en código máquina, que arrancan el ordenador. Este sector se llama MBR (Master Boot Record).

3. Si el gestor de arranque es multiarranque (soporta el arranque de diferentes sistemas operativos) muestra un menú donde el usuario debe seleccionar el sistema operativo a arrancar. Una vez elegida la opción, el gestor transfiere el control al primer sector de la partición del disco duro, donde está el programa cargador de dicho sistema operativo.
4. El programa cargador del sistema operativo carga el núcleo (kernel), que es ahora el que continua la secuencia hasta quedar listo para el inicio de sesión, por parte del usuario.
5. El usuario entra en el sistema introduciendo un nombre de usuario válido para dicho sistema y su contraseña correspondiente. Si ambos son correctos se permite la entrada. Si alguno de ellos no es válido, se vuelve a repetir la identificación completa del usuario.

## 4.2 El gestor de arranque

De la secuencia de arranque descrita en el apartado anterior, se va a prestar especial atención a los pasos 2 y 3, ambos relativos al gestor de arranque. Se supone, para los ejemplos y las órdenes, que el ordenador arranca desde una unidad de disco duro IDE o SATA.

Todo disco duro tiene un sector 0 llamado **Master Boot Record** (MBR) que es el sector de arranque del disco duro. En él se aloja un programa encargado de pasar el control, en secuencia de arranque, al sector cero de la partición que contiene el sistema operativo seleccionado.

Es decir, toda partición primaria o extendida tiene su sector 0, también llamado sector de arranque de la partición. En este sector se aloja, a su vez, un programa encargado de arrancar el sistema operativo instalado en dicha partición. En ocasiones este programa es una parte o etapa del gestor instalado en el MBR.

Se puede definir un **gestor de arranque** como una aplicación que se carga en memoria al encender el ordenador y que permite al usuario elegir el Sistema Operativo con el que quiere trabajar.

Cuando en el disco duro sólo hay instalado un sistema operativo como Windows, el gestor de arranque correspondiente suele ser transparente al usuario y el usuario puede que no se dé cuenta de que existe.

Pero si el usuario quiere alojar en su disco duro más de un sistema operativo, por ejemplo Windows y GNU/Linux, necesitará disponer de un gestor de arranque que permita al usuario elegir, en cada arranque, el sistema operativo que desea.



Existen varios **gestores multiarranque** que, instalados en el MBR, permiten al usuario seleccionar el sistema operativo saltando al sector cero de la partición (primaria o extendida) que lo aloja y, ejecutando a su vez el gestor de arranque propio de dicho sistema.

Ejemplos de gestores de multiarranque son los siguientes:

- GNU GRUB (Grand Unified Bootloader)
- LILO (LIinux Loader) algo obsoleto
- SYSLINUX
- PXELINUX, derivado de SYSLINUX y utilizado para arranque de GNU/Linux por red. Requiere la existencia de un servidor PXE en la red.



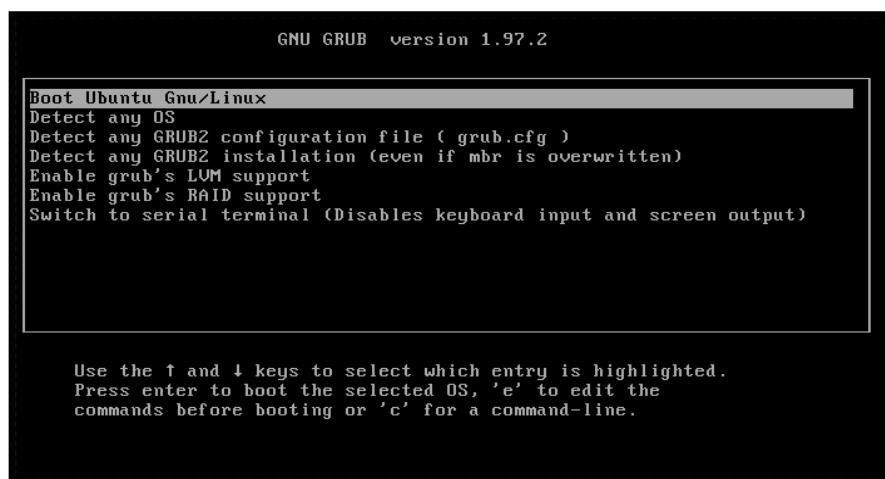
#### ENLACES DE INTERÉS

En los siguientes enlaces puedes acceder a los gestores multiarranque citados:

- [SYSLINUX](#)
- [PXELINUX](#)

## 4.3 El gestor de arranque GNU GRUB

GNU GRUB es un gestor de arranque capaz de iniciar diferentes tipos de sistemas operativos libres, así como sistemas operativos privativos, a través del arranque en cadena que luego se explica.



**Imagen: Ejemplo de arranque con GRUB**

Además de ser flexible, las principales características de GNU GRUB son las siguientes:

- Entiende sistemas de archivos como: ext2/ext3 (típicos de GNU/Linux), VFAT y NTFS (típicos de la familia Microsoft Windows), JFS (IBM), XFS



(desarrollado por la antigua Silicon Gráficos y el primero en disponer de sistema de journaling2, también para GNU/Linux),...

- Entiende diferentes tipos de núcleos, y por tanto, puede arrancar un sistema operativo cualquiera sin conocer la posición física del núcleo de dicho sistema en el disco duro. Lo único que necesita es conocer el nombre del archivo, el disco y partición donde se encuentra.
- GNU GRUB dispone de un archivo de configuración sobre el que se puede actuar y hacer modificaciones 'en caliente' antes de arrancar un sistema operativo. Esta característica de edición de entradas específicas, antes de ser utilizadas, es muy interesante para hacer pruebas cuando se presentan problemas en el arranque.
- GNU GRUB tiene 2 modos de trabajo:
  - **La carga directa:** el sistema operativo (núcleo) se carga directamente sin ningún tipo de intermediarios. Modo utilizado para los sistemas GNU/Linux.
  - **La carga encadenada:** se utiliza para cargar otros sistemas operativos y significa que el MBR apunta al primer sector de la partición que tiene el sistema operativo y en él están los archivos necesarios para arrancarlo. Es decir, el GRUB carga el cargador de arranque de otro sistema operativo. Modo utilizado por los sistemas de la familia Microsoft Windows.
- No tiene límite en el número de núcleos de sistemas operativos arrancables. El menú que presenta al usuario puede tener tantas entradas disponibles como necesite el usuario.
- GNU GRUB no distingue entre discos IDE (/dev/hda) y SATA/SCSI (/dev/sda). La nomenclatura utilizada para ambos es la misma.
- Cuenta las unidades desde cero sin importarle su tipo.
- GNU GRUB permite el arranque a través de la red, ya que puede cargar imágenes de arranque de sistemas operativos utilizando el protocolo TFTP 3.

## 4.4 Ejecución de GNU GRUB

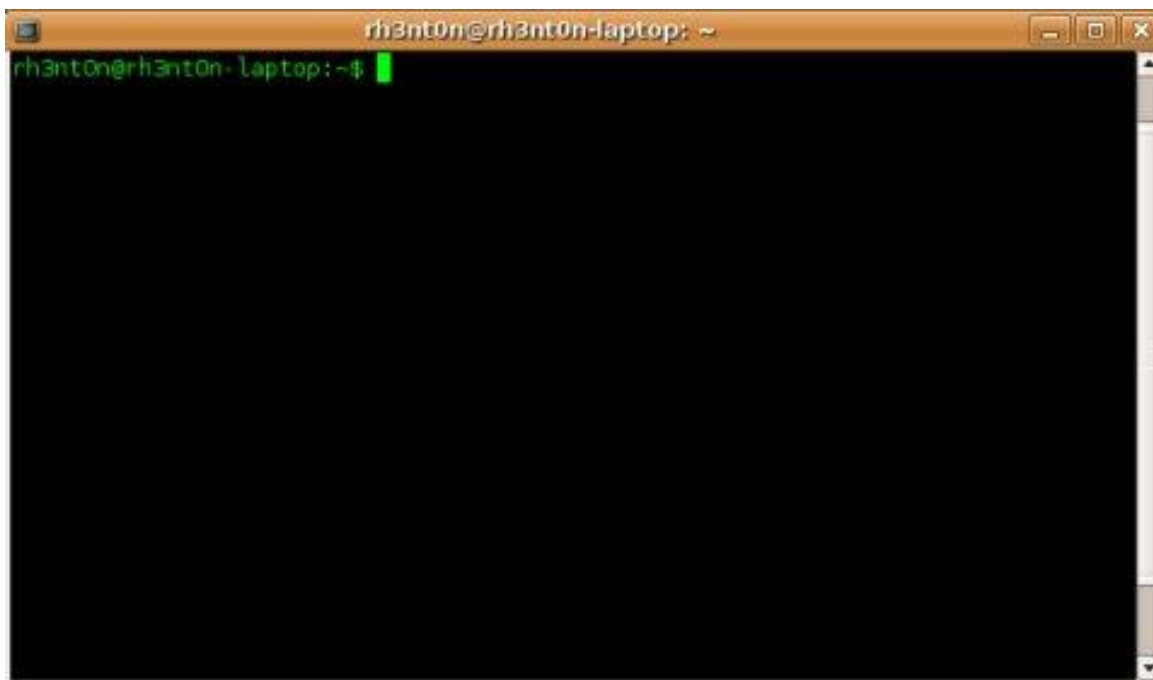
En general, un gestor de arranque para equipos con procesadores x86 o AMD tiene al menos dos etapas. La primera consiste en un pequeño programa en el MBR, cuya única función es localizar el gestor de arranque de la segunda etapa y cargar la primera parte de éste en memoria. En concreto, la ejecución de GNU GRUB está dividida en dos etapas. Cada etapa es una fase de ejecución de GNU GRUB.

- **Etapas 1:** la BIOS carga el GRUB en memoria, desde el MBR.
- **Etapas 2:** visualiza el menú de GRUB para seleccionar el sistema operativo a iniciar y carga en memoria el núcleo de dicho sistema. A partir de este momento es el núcleo el que se encarga de continuar la secuencia de arranque. En esta etapa el GRUB dispone de una interfaz para editar la entrada del sistema operativo y poder realizar modificaciones sobre ella antes de proceder a su carga y ejecución.

En ocasiones, cuando la partición que contiene el núcleo ( /boot ) está más allá del cilindro 1024 del disco duro o se está utilizando discos en modo LBA4, debe existir una etapa intermedia (etapa 1.5) que sirve de puente entre ambas y que depende del sistema de archivos. Esta etapa 1.5 es un pequeño archivo (~10Kb) y suele estar en /boot .

## 5. MAN, LA AYUDA DEL SISTEMA

En general, todo usuario que se inicia en Linux termina utilizando la línea de comandos (también llamada Shell), que es prácticamente indispensable para la administración del sistema a pesar de lo poco amigable que pueda resultar.



**Imagen: Shell de Linux**

La mayoría de los programas y aplicaciones (ejecutables desde la shell) incluyen gran número de opciones y argumentos, con la consiguiente desventaja de la dificultad de recordar todos y cada uno de ellos.

Por ello, se hace necesario algún tipo de documentación que sea fácilmente accesible y comprensible.

Man (abreviatura de Manual) es el manual en línea para todos los sistemas Linux, y también para los sistemas Unix.

Man es un paginador de manuales que formatea y muestra las páginas del manual solicitadas. Estas páginas de ayuda ofrecen la sintaxis y descripción detallada de comandos, utilidades y funciones, incluyendo todas sus opciones y argumentos.



## EJEMPLO PRÁCTICO

Si tecleamos la orden: *man mkdir* obtendremos información sobre el comando *mkdir*.

```

cesur@cesur-VirtualBox: ~
MKDIR(1)                                User Commands                                MKDIR(1)

NAME
    mkdir - make directories

SYNOPSIS
    mkdir [OPTION]... DIRECTORY...

DESCRIPTION
    Create the DIRECTORY(ies), if they do not already exist.

    Mandatory arguments to long options are mandatory for short options
    too.

    -m, --mode=MODE
        set file mode (as in chmod), not a=rwx - umask

    LibreOffice Impress ts
    no error if existing, make parent directories as needed

    -v, --verbose
        print a message for each created directory

Manual page mkdir(1) line 1 (press h for help or q to quit)
  
```

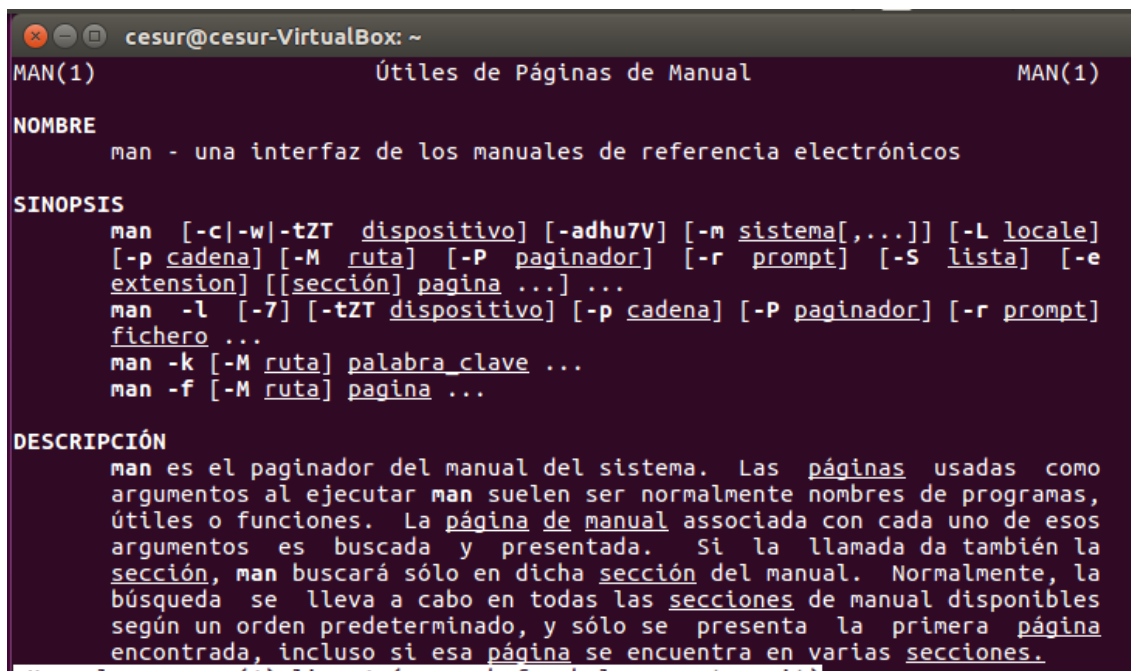
La descripción puede incluir ejemplos de utilización y otra información como archivos utilizados y comandos relacionados.

Las páginas del manual se ubican en directorios en función del idioma y a su vez, se organizan en secciones.

Como es evidente, las páginas Man están traducidas a muchos idiomas, entre ellos el español. Actualmente hay soporte para alemán, finlandés, francés, italiano, japonés, coreano, etc.

Como nota interesante, cabe señalar que el proyecto para la documentación oficial de sistemas Linux llamado tldp (The Linux Documentation Project, tldp.org) se encarga de coordinar la creación, mantenimiento y desarrollo de las páginas Man en Inglés; y su correspondiente versión española es.tldp.org (denominado comúnmente tldp-es/LUCAS, servicios editoriales para la documentación libre en español), que lleva a cargo la coordinación para la traducción de las mismas, además de saber cómo traducirlas al español e incluso participar en la traducción. Antes de seguir con las siguientes secciones del artículo, comprobaremos si efectivamente tenemos el programa Man instalado en nuestro sistema.

Lo comprobamos así: usuario@miMaquina:~\$ man man



```

cesur@cesur-VirtualBox: ~
MAN(1)                               Útiles de Páginas de Manual                               MAN(1)

NOMBRE
man - una interfaz de los manuales de referencia electrónicos

SINOPSIS
man [-c|-w|-tZT dispositivo] [-adhu7V] [-m sistema[,...]] [-L locale]
[-p cadena] [-M ruta] [-P paginador] [-r prompt] [-S lista] [-e
extension] [[sección] página ...] ...
man -l [-7] [-tZT dispositivo] [-p cadena] [-P paginador] [-r prompt]
fichero ...
man -k [-M ruta] palabra_clave ...
man -f [-M ruta] página ...

DESCRIPCIÓN
man es el paginador del manual del sistema. Las páginas usadas como
argumentos al ejecutar man suelen ser normalmente nombres de programas,
útiles o funciones. La página de manual asociada con cada uno de esos
argumentos es buscada y presentada. Si la llamada da también la
sección, man buscará sólo en dicha sección del manual. Normalmente, la
búsqueda se lleva a cabo en todas las secciones de manual disponibles
según un orden predeterminado, y sólo se presenta la primera página
encontrada, incluso si esa página se encuentra en varias secciones.

```

Imagen: Comprobación de que Man está instalado

Si se obtiene el siguiente mensaje de error, no lo tenemos instalado:

```
usuario@miMaquina:~$ man man
```

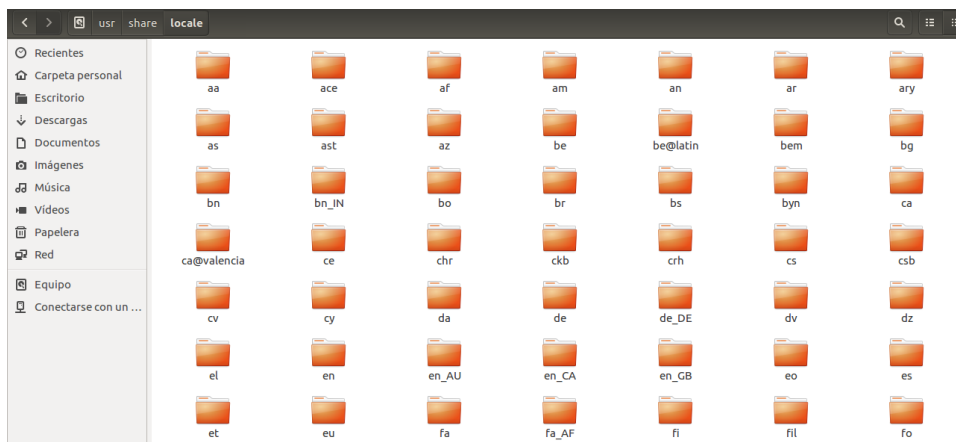
```
bash: man: command not found
```

## 5.1 Posibles problemas con el idioma

Como se ha mencionado, seguramente tengamos instalado el programa Man de manera correcta. Si no es así, ya sea porque no lo tengamos instalado o porque las páginas del manual no estén en castellano (incluidos caracteres como la "ñ" o las vocales acentuadas), seguiremos las siguientes pautas.

En sistemas GNU/Linux y UNIX, para configurar los estándares y formatos específicos del país, como son el idioma, las fechas, la moneda, la coma decimal, etc. (lo que es para Windows la "configuración regional") se hace uso de las "**locales**" o códigos de idioma. Realmente las "locales" son un mecanismo que permite a las aplicaciones mostrar una salida adecuada y una funcionalidad conforme a las convenciones locales que hemos mencionado (el conjunto de caracteres, el formato del tiempo y de la fecha, los símbolos monetarios, etc.). Éstas utilizan variables de entorno para determinar el comportamiento apropiado. Por ejemplo, ponemos definir diferentes "locales" para distintos usuarios del sistema y otras para el súper usuario.

Las “locales” y sus datos son parte de las bibliotecas del sistema y se pueden encontrar en /usr/share/locale por lo general.



**Imagen:** Directorio que contiene las “locales” en Ubuntu

Si exploramos el directorio, un “locale” (o un código de idioma) es generalmente nombrado como “aa” y “aa\_BB”. Esto quiere decir que existen dos niveles. El primer nivel representa el código del lenguaje (aa) y el segundo el código del país (BB).



#### EJEMPLO PRÁCTICO

es\_AR, lenguaje español, país Argentina.  
 en\_GB, lenguaje inglés, país Reino Unido.  
 en\_US, lenguaje español, país Estados Unidos.  
 fr, lenguaje francés (únicamente).

## 5.2 Configuración de las locales

En la configuración de las locales se debe llevar a cabo:

Lo primero sería comprobar que tenemos instaladas las locales (es muy probable que sí).

- Para sistemas con gestión de paquetes Debian:

```
usuario@miMaquina:~$ dpkg -l | grep locales
```

```
ii locales 2.6.1-1 GNU C Library: National Language (locale)
```

- Para sistemas con gestión de paquetes Rpm:

```
[usuario@miMaquina~]$ rpm -qa | grep locales
```

```
locales-2.3.2
```

Si no es así, se instalan ejecutando como root.

- Para sistemas con gestión de paquetes Debian:

```
usuarioRoot@miMaquina:~# apt-get install locales
```

- Para sistemas con gestión de paquetes Rpm:

```
[usuarioRoot@miMaquina~]# rpm -ivh locales-version-arquitectura.rpm
```

Ahora, se comprueba que el idioma es el correcto.

Se ejecuta el fichero /usr/bin/locale (que obtiene la configuración local):

```
usuario@miMaquina:~$ /usr/bin/locale
```

```
LANG=es_ES.UTF-8
```

```
LC_CTYPE="es_ES.UTF-8"
```

```
LC_NUMERIC="es_ES.UTF-8"
```

```
LC_TIME="es_ES.UTF-8"
```

```
LC_COLLATE="es_ES.UTF-8"
```

```
LC_MONETARY="es_ES.UTF-8"
```

```
LC_MESSAGES="es_ES.UTF-8"
```

```
LC_PAPER="es_ES.UTF-8"
```

```
LC_NAME="es_ES.UTF-8"
```

```
LC_ADDRESS="es_ES.UTF-8"
```

```
LC_TELEPHONE="es_ES.UTF-8"
```

```
LC_MEASUREMENT="es_ES.UTF-8"
```

```
LC_IDENTIFICATION="es_ES.UTF-8"
```

```
LC_ALL=
```

Vemos que LC\_TIME es para la configuración de la hora, LC\_MONETARY para la moneda, etc. Para idioma español es conveniente poner como "locales" es\_ES.UTF-8.

Con el comando locale -a se ven los códigos de idioma disponibles en el sistema (el "locale" por defecto del sistema es "C", que equivale al idioma inglés).

```
usuario@miMaquina:~$ locale -a
```

```
C
```

```
es_ES
```

```
es_ES.utf8
```

```
POSIX
```

Para ver cuál de ellos es el que está activo:

```
usuario@miMaquina:~$ echo $LANG
```

```
es_ES.UTF-8
```

También se pueden ver los locales activados en nuestro sistema editando el fichero `/etc/locale.gen`:

```
usuario@miMaquina:~$ cat /etc/locale.gen
```

```
# This file lists locales that you wish to have built. You can find a list
```

```
# of valid supported locales at /usr/share/i18n/SUPPORTED. Other
```

```
# combinations are possible, but may not be well tested. If you change
```

```
# this file, you need to rerun locale-gen.
```

```
#
```

```
es_ES.UTF-8 UTF-8
```

Además, como curiosidad, con `locale -m` vemos todos los mapas de caracteres disponibles (además de los activados).

```
usuario@miMaquina:~$ locale -m
```

```
ANSI_X3.110-1983
```

```
ANSI_X3.4-1968
```

```
ARMSCII-8
```

```
ASMO_449
```

```
BIG5
```



## 5.3 Contenido de man

Es interesante saber que este paquete instala en el sistema una serie de aplicaciones que también sirven para buscar documentación, entre otras cosas.

Veamos una breve descripción de las más interesantes:

- **whatis [opciones] <comando>**

Realiza búsquedas del comando especificado en las cabeceras de todas las páginas Man almacenadas y muestra una breve descripción del mismo junto con su nombre y sección entre paréntesis.

Por ejemplo:

```
usuario@miMaquina:~$ whatis chattr
```

```
chattr (1) - change file attributes on a Linux second extended file system
```

En este sentido, el funcionamiento es muy similar a Man, solo que whatis muestra la pequeña descripción incluida en la sección "Nombre" de las páginas del manual.

- **apropos [opciones] <cadena>**

Idéntico cometido que el comando whatis, pero con ventaja de especificar un patrón, no el nombre del comando exacto. Igual que whatis, apropos buscará el patrón en las cabeceras de todas las páginas Man almacenadas.

Por ejemplo:

```
usuario@miMaquina:~$ apropos clea
```

```
cleanlinks (1) - remove dangling symbolic links and empty directories
```

```
cleanup-info (8) - clean up the mess that bogus install-info may have done
```

```
clear (1) - clear the terminal screen
```

```
clear_console (1) - clear the console
```

```
clearenv (3) - clear the environment
```

```
.....
```

```
.....
```

sigreturn (2) - return from signal handler and cleanup stack frame

syslog (2) - read and/or clear kernel message ring buffer; set console\_loglevel

tidy (1) - Utility to clean up and pretty print HTML/XHTML/XML

apropos también hace uso de la base de datos utilizada por whatis.

- **mandb**

Crea o actualiza la base de datos que utiliza el comando whatis para las búsquedas.

Por ejemplo:

```
usuario@miMaquina:~$ mandb
```

Purgando entradas antiguas en la base de datos en /usr/share/man...

Procesando las páginas de manual bajo /usr/share/man...

Actualizando el caché de índices para la ruta `/usr/share/man/man8'.

Aguarde...listo.

Controlando si hay páginas cat sin fuentes bajo /usr/share/man...

Controlando si hay páginas cat sin fuentes bajo /var/cache/man...

Purgando entradas antiguas en la base de datos en /usr/local/man...

.....

Procesando las páginas de manual bajo /usr/local/share/man...

2 subdirectorios man tenían páginas de manual más recientes.

Se añadieron 5 páginas de manual.

Se añadieron 0 «cats» sin fuentes.

Se han eliminado 0 entradas antiguas en la base de datos.

**Importante:** Algunas distribuciones Linux utilizan el comando makewhatis en vez del mandb.

- **manpath**

Muestra el path o la ruta donde debe buscar el programa Man sus páginas.

- **catman**

Crea o actualiza las páginas de manual con formato. Es decir, guarda las páginas del manual ya formateadas en directorios establecidos al efecto (normalmente en /var/cache/man/cat(nº\_de\_sección)).

Recordemos que las "manpages" inicialmente vienen comprimidas y sin formatear. El programa Man se ocupa de mostrarlas convenientemente.

Esta solución elimina la necesidad de descomprimir y formatear una pagina del manual la próxima vez que un usuario la solicite.

También catman hace uso de la base de datos index.db para determinar qué ficheros tienen que ser formateados.

Estas páginas son llamadas "páginas cat".

## 5.4 Secciones de man

La información de las páginas Man de Linux están agrupadas en un conjunto de archivos. Estos ficheros están distribuidos en secciones y, cada sección contiene un tipo específico de información. Ejecutando el comando manpath podemos saber dónde están ubicadas estas secciones.

Descripción de secciones:

1 – Comandos y aplicaciones de usuario.

2 – Llamadas al sistema (funciones del núcleo) y errores del kernel.

3 – Llamadas a librerías del sistema.

4 – Drivers de dispositivos y protocolos de red.

5 – Formatos estándar de archivos.

6 – Juegos y demos.

7 – Ficheros y documentos misceláneos.

8 – Comandos de administración del sistema.

9 – Especificaciones e interfaces del kernel.

Cuando ejecutamos Man, éste busca en todas las secciones el comando pasado como argumento, siguiendo un orden específico en las mismas y, devuelve la primera coincidencia, aunque aparezca en más de una sección.

Por ejemplo:

```
usuario@miMaquina:~$ whatis passwd
```

```
passwd (1) - change user password
```

```
passwd (5) - the password file
```

Vemos que existen dos páginas del manual, la 1 y la 5.

Con esto, ejecutando `man passwd` veríamos solamente la ayuda en línea para la primera sección (Comandos y aplicaciones de usuario).

## 5.5 Opciones de man

Se explican las más importantes:

**man -w  
<comando>**

- Indica que se visualice la ruta completa (path) de las páginas Man en lugar de la página de ayuda asociada al comando.

**man -k  
<comando>**

- El funcionamiento de Man es idéntico al del comando `apropos`.

**man -h**

- Muestra la ayuda para Man.

**man -f  
<comando>**

- El funcionamiento de Man es idéntico al del comando `whatis`.

**man -a  
<comando>**

- Busca por todas las secciones de Man y muestra todas las coincidencias y no solo la primera página del manual encontrada.

**man [numero \_  
sección]  
<comando>**

- Si queremos dirigirnos a una sección en concreto sin pasar secuencialmente por ellas.

**man -P  
<paginador>  
<comando>**

- Para visualizar una la página Man con el paginador que pasamos como argumento.

**man -L <locales>  
<comando>**

- Establecemos manualmente las "locales". Por ejemplo, con la siguiente instrucción veríamos las páginas Man con la codificación de caracteres ISO-8859-15\*:  
usuario@miMaquina:~\$ man -L es\_ES.iso88591 ls

Una vez “dentro” de una página, se pueden realizar diversas acciones.

TECLA	SIGNIFICADO
Q	Salir de la pantalla
H	Despliega la pantalla de ayuda
E	Avanza una línea
Y	Regresa una línea
[Barra espaciadora]	Avanza una pantalla
D	Avanza media pantalla
U	Regresa media pantalla
[Flecha a la derecha]	Mueve media pantalla a la derecha
[Flecha a la izquierda]	Mueve media pantalla a la izquierda
¿[cadena]	Busca una cadena de texto hacia arriba
/[cadena]	Busca una cadena de texto hacia abajo
G	Avanza al inicio del manual

## 6. ACTUALIZACIÓN DEL SISTEMA

Todo sistema operativo necesita actualizarse periódicamente para garantizar su correcto funcionamiento, ya que sus fabricantes trabajan de forma continua en mejoras y soluciones de los problemas que se van detectando.

En Ubuntu se puede realizar esta actualización mediante comandos, desde el Terminal, haciendo uso de la herramienta apt. Apt es un sistema de gestión de paquetes creado por el proyecto Debian, que simplifica mucho el proceso de instalación y desinstalación de programas en sistemas Linux.

Básicamente se necesita realizar dos operaciones (para ello harán falta privilegios de administrador):

- **apt-get update.** Actualiza, tomando como referencia los servidores de repositorios que se tengan configurados, la lista de paquetes disponibles y sus versiones.

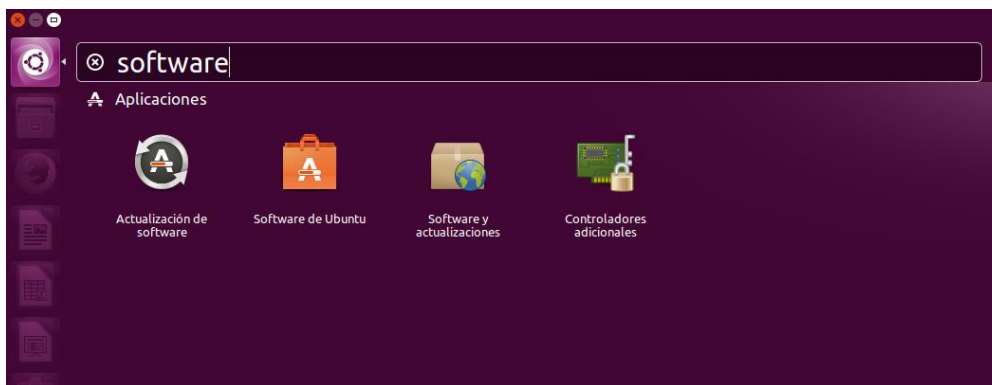


```
cesur@cesur-VirtualBox: ~  
To run a command as administrator (user "root"), use "sudo <command>".  
See "man sudo_root" for details.  
  
cesur@cesur-VirtualBox:~$ sudo apt-get update  
[sudo] password for cesur:  
Obj:1 http://security.ubuntu.com/ubuntu xenial-security InRelease  
Obj:2 http://es.archive.ubuntu.com/ubuntu xenial InRelease  
Obj:3 http://es.archive.ubuntu.com/ubuntu xenial-updates InRelease  
Obj:4 http://es.archive.ubuntu.com/ubuntu xenial-backports InRelease  
AppStream cache update completed, but some metadata was ignored due to errors.  
Leyendo lista de paquetes... 27%
```

Imagen: Ejecución de apt-get update

- **apt-get upgrade.** Con la lista actualizada gracias al comando anterior, instala las nuevas versiones de los programas que tengamos instalados tratando, siempre que sea posible, de mantener la configuración de estos.

Si no se desea utilizar comandos, Ubuntu dispone de una herramienta gráfica, similar al Windows Update de los sistemas Windows, que nos permite controlar las actualizaciones del sistema de manera sencilla. Se trata de **Software y actualizaciones**, que se puede localizar desde el buscador de Unity.



**Imagen: Acceso a Software y actualizaciones**

Mediante esta herramienta se puede indicar qué tipo de actualizaciones se desea descargar, y desde dónde hacerlo



**Imagen: Herramienta Software y actualizaciones**

Y, más interesante aún, se pueden automatizar dichas instalaciones, de manera que se realicen de forma periódica sin que se tenga que estar atento a ello. Para ello utilizamos la pestaña **Actualizaciones**, en la que podemos indicar qué tipo de actualizaciones se desea realizar, y cuándo hacerlas.



**Imagen: Actualizaciones automáticas**

## RESUMEN FINAL

En esta unidad se han comenzado a revisar los principales sistemas operativos que podemos encontrarnos en nuestro puesto de trabajo. De ellos, nos hemos centrado en los sistemas Unix/Linux, explicando en qué consisten sus distribuciones y quedándonos con una de ellas, Ubuntu, para los ejemplos prácticos.

A continuación se ha hablado de la virtualización, una técnica que nos permite tener instalados con facilidad varios sistemas operativos en un mismo equipo. Se ha presentado, a modo de ejemplo, un software de virtualización muy popular, VirtualBox.

Más tarde nos hemos centrado en el proceso de instalación de un sistema operativo, concretamente Ubuntu. Para ello se ha creado una máquina virtual con VirtualBox.

Se ha hablado también de los gestores de arranque, otra alternativa para tener varios sistemas operativos en un mismo ordenador.

Se ha descrito con detalle la ayuda de los sistemas Linux, Man.

Por último, se ha revisado el proceso de actualización de un sistema operativo, imprescindible para que siga funcionando de forma óptima.