

## 2.4. Configuraciones multisistema



# Índice

---

Objetivos.....	4
Gestión de varios sistemas operativos en un ordenador .....	5
Requisitos previos. Particionado y dimensionado .....	5
Problemas con el MBR. Elegir gestor de arranque.....	6
Problemas con el registro de arranque (MBR) .....	6
Elegir el gestor de arranque adecuado .....	7
¿Qué tipo de BIOS tiene mi sistema?.....	8
Preparar las particiones para permitir su arranque.....	9
Analizar el orden en la instalación de los sistemas operativos.....	10
Los gestores de arranque .....	11
Código de arranque maestro ("Máster Boot Code") .....	11
Proceso de arranque del sistema:.....	12
Configuración de gestores de arranque en S.O. libres y propietarios .....	13
Gestores de arranque en sistemas Windows.....	13
Gestores de arranque en sistemas Linux.....	14
Ejemplo de arranque dual .....	15
Configuración del arranque dual en la máquina virtual .....	15
Reparar el gestor de arranque.....	21
Reparar gestores de arranque en sistemas Windows.....	21
Reparar gestores de arranque en sistemas Unix/Linux .....	22
Sustitución del gestor de arranque.....	23
Ficheros necesarios para el arranque.....	24
Instalaciones múltiples y actualizaciones .....	25
Requisitos, versiones y licencias .....	25
Actualización de sistemas operativos y aplicaciones.....	26
Actualizar a una versión superior (update) .....	27
Cambiar a una versión inferior (downgrade) .....	27
Uso de instalaciones desatendidas.....	29
Instalación desatendida en Windows .....	30
Instalación desatendida en Linux .....	30

Herramientas - Instaladores .....	31
Controladores de dispositivos.....	33
Herramientas de gestión de drivers.....	34
Gestión de los drivers .....	35
Gestión de controladores en entornos GNU/Linux .....	35
Cómo hacer un "live USB" de Ubuntu.....	36
Pasos para crear el "live USB".....	37
Despedida .....	38
Resumen.....	38

Estudiaremos ahora el caso de que queramos tener varios sistemas operativos en un mismo equipo, y cuáles son los principales gestores de arranque que lo permiten.

Aunque pueda parecer una situación no muy frecuente en la práctica, la realidad es que a veces nos vemos forzados a trabajar con diferentes sistemas y tener la posibilidad de instalarlos en un mismo equipo puede ser muy provechoso.

Ya sabemos que podemos recurrir a la virtualización de sistemas, pero a veces necesitamos que el sistema se ejecute directamente sobre el hardware y no deseamos destruir el S.O. previo que ya tenemos.

## Objetivos

1. Aprender cómo instalar dos sistemas operativos sobre el mismo equipo.
2. Comprender cómo funcionan los gestores de arranque de los sistemas modernos.
3. Conocer los requisitos que debemos contemplar para instalar aplicaciones en el sistema.

# Gestión de varios sistemas operativos en un ordenador

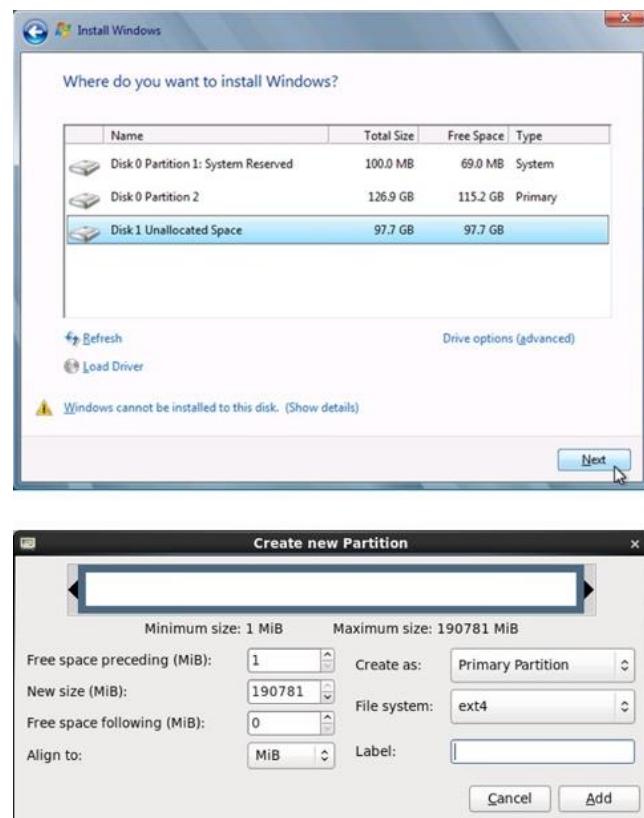
## Requisitos previos. Particionado y dimensionado

Cuando vamos a instalar varios sistemas operativos sobre un mismo ordenador y disponemos de un solo disco duro tendremos que **repartir su espacio** para ambos sistemas, además de las aplicaciones y los datos de usuario que vayan a cargarse.

Es importante **verificar que nuestro disco tiene el espacio suficiente** para albergar varias particiones, al menos con el tamaño mínimo recomendado por el proveedor del S.O.

En muchos casos el segundo sistema operativo querremos instalarlo después de haber instalado el primero y entonces, normalmente al introducir el nuevo S.O. en el dispositivo (CD/DVD, pendrive, ...) e iniciar el proceso de instalación, el programa detectará las particiones existentes en el disco y nos ofrecerá la creación de una nueva partición o la modificación de las existentes (para crear espacio para el nuevo S.O.) y dar formato a la nueva partición.

Hemos de tener en cuenta que **algunos sistemas operativos requieren más de una partición para funcionar**, y en ese caso habrá que crearlas también durante la instalación.



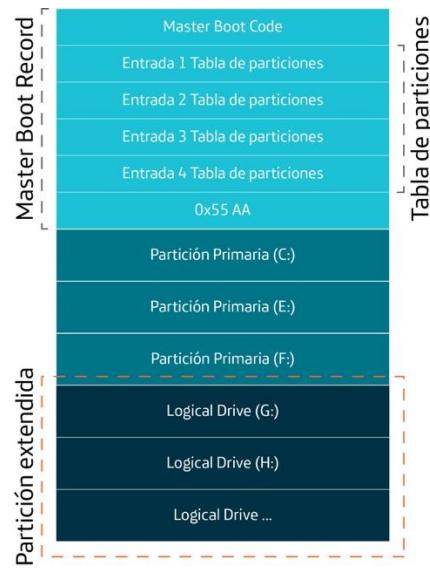
# Problemas con el MBR. Elegir gestor de arranque

El **registro maestro de arranque (MBR - Master Boot Record)** es el primer sector ("sector cero") de un dispositivo de almacenamiento de datos, como por ejemplo puede ser un disco duro.

El **MBR** se puede emplear para almacenar la tabla de particiones del dispositivo y también para el arranque del sistema cuando es llamado desde la BIOS.

Dentro del MBR hay una porción de código que en el arranque se encarga de lanzar el "**Volume Boot Record (VBR)**" o "**bootloader**" del S.O. que se encuentra al principio de la partición primaria activa.

En este punto, el MBR puede asumir que se arranca desde la partición activa del disco actual o bien puede programarse como un MBR de "**arranque dual**". En ese caso debe interactuar con el usuario para preguntarle desde qué partición se arranca, y si fuese necesario pasarle el control a un MBR de otro disco.



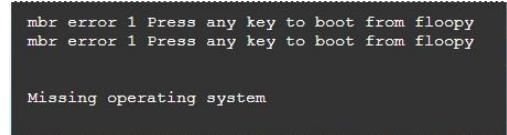
*Estructura del disco.*

## Problemas con el registro de arranque (MBR)

Si por alguna razón se corrompe el contenido del MBR, al intentar arrancar el ordenador es común que nos salga algún mensaje del tipo: "Sistema Operativo no encontrado" / "Falta BOOTMGR presione Ctrl + Alt + Del para reiniciar el sistema." / "Inserte un dispositivo de arranque.". En este caso, si sabemos que tenemos instalado nuestro S.O. y antes de proceder a una reinstalación, podemos intentar reparar el MBR, como veremos más adelante.

Por otra parte, **los problemas con el MBR pueden venir dados por sus limitaciones:** como por ejemplo que solo pueden definirse 4 particiones primarias o 3 primarias + 1 partición extendida (y dentro de ella las particiones lógicas que queramos), o que los metadatos de las particiones lógicas (en la partición extendida) se almacenan como una lista enlazada y si se pierde algún enlace, se perderán todas las particiones lógicas existentes después de los metadatos.

Otros posibles problemas, ya no debidos a limitaciones, pueden ser los "**MBR rootkit**", que son una clase de "**malware**" que ataca directamente al "master boot record", reemplazando normalmente el MBR original por uno que ellos mismos generan, y al mismo tiempo abren "puertas traseras" ("**backdoors**") y establecen conexiones no autorizadas para volcado de datos.

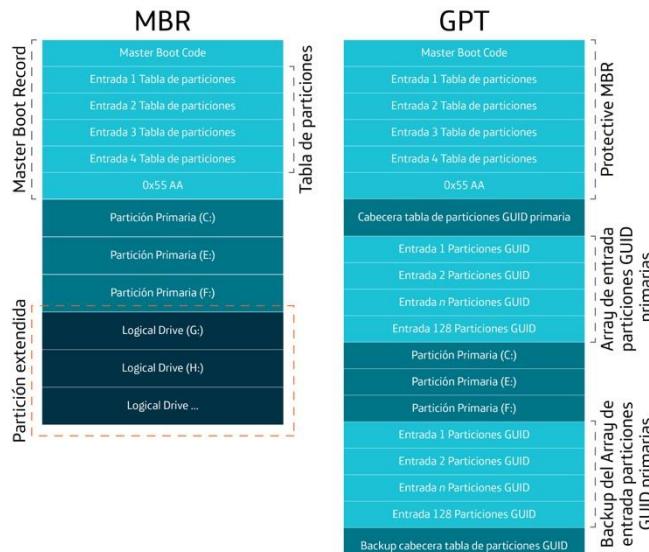


Mensaje de error en el MBR.

## Elegir el gestor de arranque adecuado

A la hora de dar formato a un dispositivo (disco duro) el estilo de registro maestro de arranque más conocido es el **MBR**, ya descrito antes. Modernamente, junto con la **sustitución de la BIOS por sistemas UEFI**, se está implantando también un nuevo estilo de particiones denominado **GPT ("GUID Partition Table")**, con menos limitaciones y más seguro, donde a cada partición se le asocia un único identificador global (GUID).

Por ejemplo, mientras que en los MBR la tabla de particiones se almacena solo en los primeros sectores del disco, **en los GPT se crean copias redundantes a lo largo de todo el disco**, por lo que en caso de fallo o algún problema, la tabla de particiones se puede recuperar automáticamente desde cualquiera de estas copias.

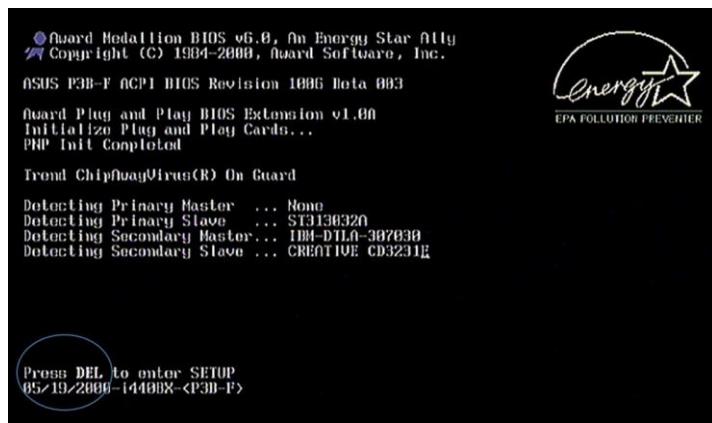


A la hora de elegir un gestor de arranque debemos conocer los requerimientos de los sistemas operativos que vamos a utilizar. Por ejemplo, Windows solo puede arrancar desde discos GPT en sus sistemas de 64 bits (desde Vista en adelante), pero los sistemas de 32 bits, aunque pueden leer y escribir, no son capaces de arrancar desde discos GPT. Las versiones modernas de Linux también son compatibles GPT, e incluso Apple ha empezado a usar GPT como tabla de particiones.

# ¿Qué tipo de BIOS tiene mi sistema?

Seguro que ya lo sabes, pero por si acaso, ¿dónde puedes ver el tipo de rutina de arranque inicial que tiene tu sistema?

¿Es de tipo BIOS o UEFI?

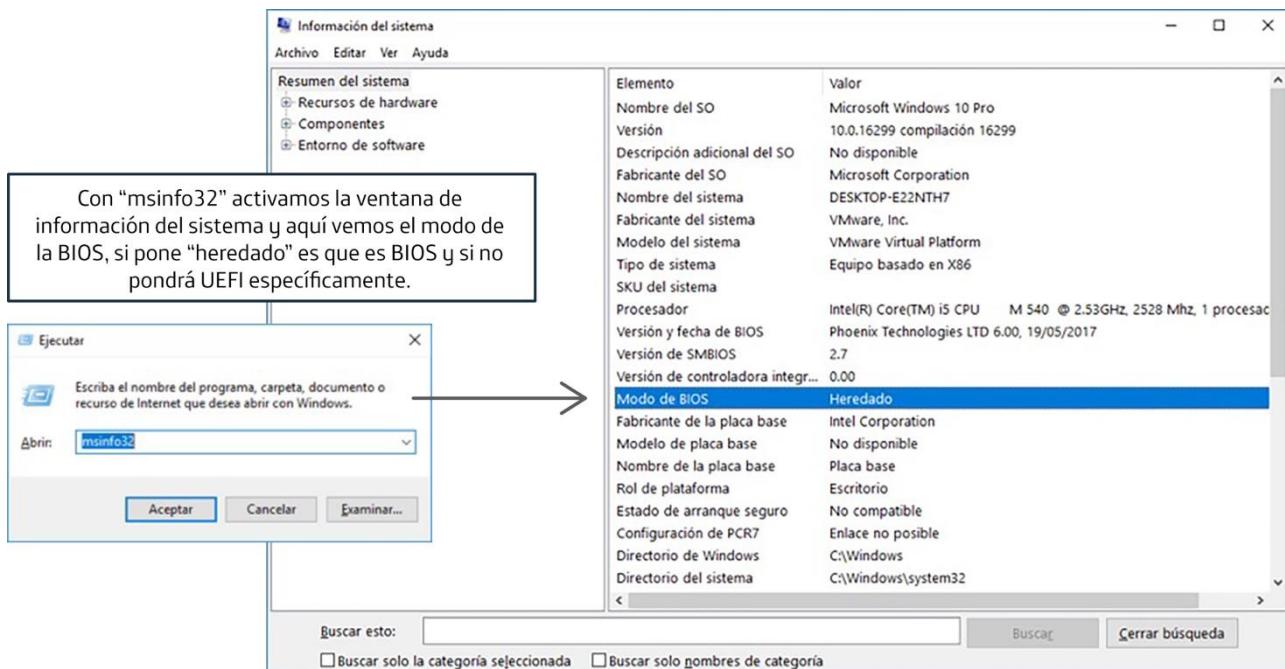


Evidentemente siempre puedes reiniciar tu sistema y durante el arranque pulsar la tecla de entrada en la rutina de gestión del arranque (esto es diferente de un ordenador a otro y suele aparecer durante un tiempo en la pantalla en el arranque).

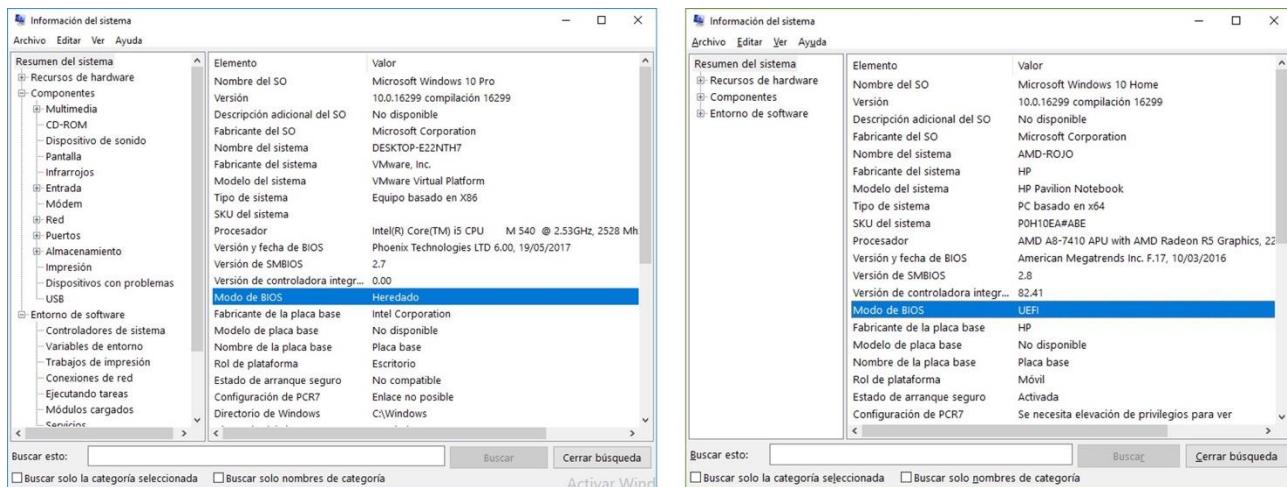
Pero vamos a ver cómo averiguar el tipo de BIOS sin tener que reiniciar el PC.

Pulsa la "tecla de windows + r" y en el cuadro de diálogo de "ejecutar" escribe "msinfo32" y luego da "enter".

Te saldrá la ventana de información del sistema y en ella verás que especifica el tipo de BIOS.



Fíjate que cuando pone "Heredado" significa que tienes el sistema más antiguo (BIOS) y cuando el equipo tiene UEFI lo pone específicamente. Te mostramos la información de dos equipos diferentes, uno con BIOS y otro con UEFI:



Ahora te pedimos que realices esta sencilla operación sobre tu equipo anfitrión y sobre la máquina virtual de Windows 10.

## Preparar las particiones para permitir su arranque

Como ves, debemos elegir el formato de gestor de arranque (MBR / GPT) adecuado para el S.O. que vayamos a utilizar, y ello determinará la estructura de las particiones en el disco, aunque no su número (el máximo sí en el caso de MBR), pero sí cómo se gestionan.

Además, si queremos instalar más de un sistema operativo en el mismo disco deberemos tener en cuenta los requisitos de todos ellos en conjunto. Por ejemplo, **si queremos instalar dos S.O. en el mismo disco:**

1. Debemos **particionar el disco creando dos o más particiones primarias**.
2. Las particiones deben tener al menos el **tamaño mínimo requerido por cada S.O.**, y si hay espacio suficiente en el disco es aconsejable que sean mayores.
3. Sobre las **particiones primarias** creadas lo aconsejable es **instalar un S.O., comenzando primero por el más antiguo** y después el más moderno (que soportará más funcionalidades).
4. También es aconsejable **configurar la partición para memoria de intercambio (SWAP)** del orden del **doble de la RAM** que tenga nuestro equipo.
5. Si alguno de los sistemas operativos a instalar **necesita más de una partición** debemos tenerlo en cuenta.

6. En el caso de que ya exista un S.O. instalado y vayamos a instalar otro, es aconsejable **desfragmentar el disco duro** antes de hacerlo y hacer **copia de seguridad** de los datos de usuario.

## Analizar el orden en la instalación de los sistemas operativos

Cuando vamos a instalar varios sistemas operativos sobre un mismo equipo, y después de asegurarnos de que cumplimos con todos los requisitos para cada uno de ellos, debemos **planificar** (si es posible) **el proceso** de instalación. Para ello tendremos varios criterios:



- Si es posible, **instalar primero los sistemas más antiguos** y luego los más modernos.
- Si instalamos sistemas Windows y sistemas Linux debemos **instalar primero Windows y luego Linux**, puesto que estos últimos pueden detectar Windows y configurar un sistema de arranque dual incluyendo Windows, cosa que no ocurre al contrario.

Por ejemplo, si quisiéramos instalar tres sistemas operativos como Windows 7, Windows 10 y Ubuntu, el orden a seguir sería: primero instalar los paquetes Windows, comenzando por el más antiguo (Windows 7) y luego el más moderno (Windows 10), y luego instalar Linux Ubuntu.

# Los gestores de arranque

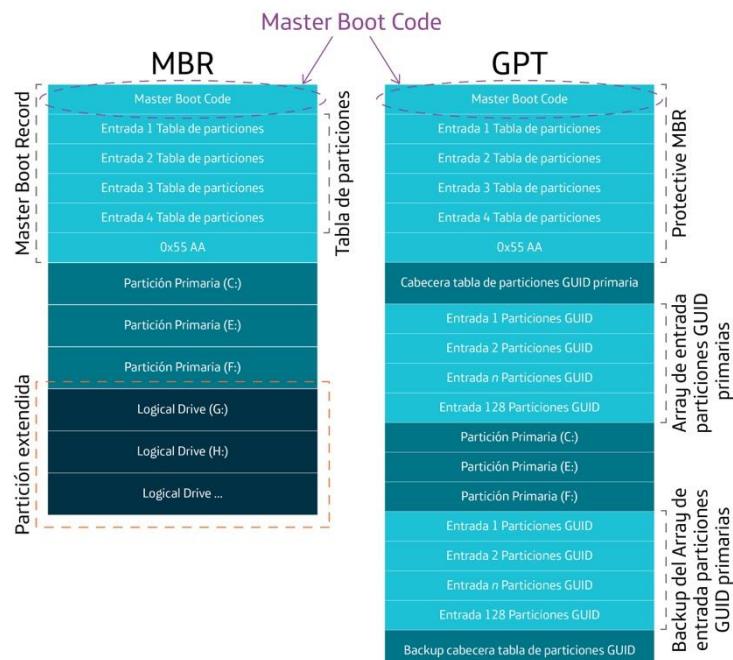
## Código de arranque maestro (“Master Boot Code”)

El “*Master Boot Code*” (también llamado “*bootstrap loader*” o cargador inicial) es una parte del registro maestro de arranque (MBR o GPT) donde se encuentran las primeras instrucciones a ejecutar tras el arranque de la máquina (*boot*).

Si estamos usando MBR el “*Master Boot Code*” ocupará 446 bytes del total de 512 del MBR. El resto es usado por las tablas de partición y una firma de disco.

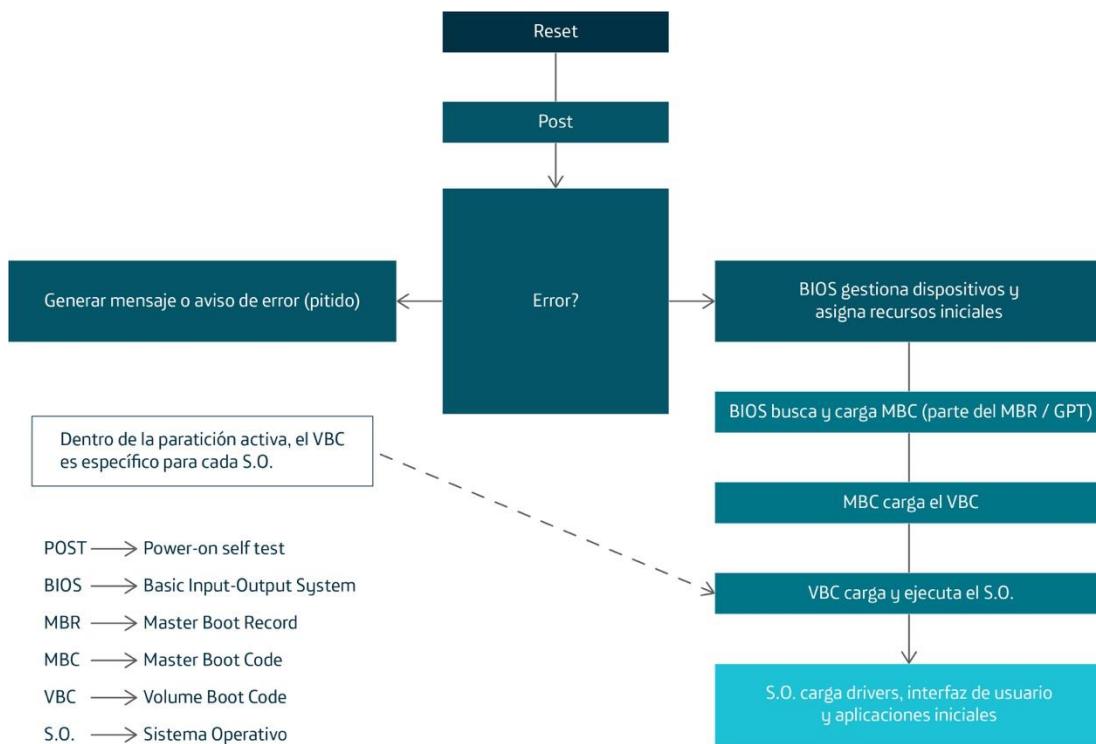
**El Master Boot Code** es llamado por la BIOS en el arranque y al ejecutarse revisa la tabla maestra de particiones para encontrar una partición activa. Si la encuentra busca su sector inicial, carga su código en memoria y le transfiere el control.

Este código de la partición activa es capaz de cargar y ejecutar cualquier otro programa situado en cualquier partición del disco, y lo que hará será cargar directamente un S.O. o bien un “gestor de arranque” que nos permita elegir entre varios de ellos.



## Proceso de arranque del sistema:

Veamos gráficamente cómo sería el proceso de arranque del sistema:



Como ves en la figura, primero se efectúa una comprobación del HW del propio equipo. Si esta comprobación falla normalmente el equipo genera un pitido como aviso de error. Si el equipo está bien pasa a leerse y ejecutarse la BIOS (que está grabada en firmware), y esta a su vez lee y ejecuta el **MBC (Master Boot Code)**.

El MBC (que ya está en disco) lee la tabla de particiones y pasa el control al **VBC (Volume Boot Code)** de la partición activa, que **es quien carga el sistema operativo** y cuando ya lo tiene activo le pasa el control para que cargue los programas del sistema y adopte la configuración que tiene preestablecida.

**Nota:** fíjate que tendremos solamente un MBC (cargador inicial) dentro del "Registro Maestro de Arranque" (MBR o GPT), pero luego en cada partición que tenga un sistema operativo instalado y listo para arrancar tendremos un VBC (gestor de arranque de ese volumen/partición, es decir, de ese S.O.).

No debes agobiarte por tantas siglas, pero es importante que al menos te suenen y comprendas a grandes rasgos cómo se realiza el proceso.

# Configuración de gestores de arranque en S.O. libres y propietarios

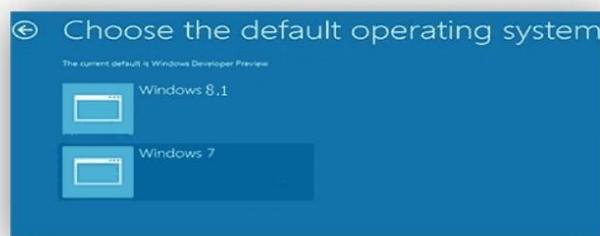


Has visto como, dependiendo del S.O., podemos tener un MBR u otro (en función de los que admitan), y aunque suelen ser comunes a un grupo o familia de sistemas hay diferencias si son libres o propietarios.

Por ejemplo, **el gestor de arranque tradicional de Windows/DOS no puede arrancar una partición Linux**, porque no está diseñado para cargar el *kernel* de Linux, y solo puede atender a una partición activa (y primaria).

## Gestores de arranque en sistemas Windows

Windows, durante la instalación, configura su gestor de arranque en nuestro sistema, pero si solamente tenemos un S.O. instalado en el equipo este se carga automáticamente y el gestor de arranque no nos pregunta.



Si instalamos varios Windows (por ejemplo Windows 7 y Windows 8) debemos instalar primero el más antiguo (Windows 7) y después el más reciente (Windows 8). De esta forma, al instalar Windows 8 el proceso de instalación detectará una partición previa de Windows 7 y dará la opción de añadir una entrada al gestor de arranque para que cuando arranquemos la máquina podamos elegir el S.O. que queremos utilizar.

## Gestores de arranque en sistemas Linux

Existen varios gestores de arranque que se instalan según la distribución que estemos usando, pero **GRand Unified Bootloader (GRUB)** es prácticamente el gestor de arranque estándar (el más utilizado) para sistemas GNU/Linux, y al instalarlo en el MBR permite arrancar cualquier partición (primaria o lógica).

Para instalar **GRUB** simplemente instalamos primero los S.O. que queramos tener en el ordenador, dejando para ser el último instalado el Linux que incorpora GRUB (por ejemplo **Ubuntu**).

GRUB detectará automáticamente todos los sistemas operativos instalados en todas las particiones del disco y añadirá una entrada a su menú para cada uno de ellos, ofreciéndonos la elección de cargar uno u otro en el arranque. Si más tarde eliminamos alguno de los S.O. o instalamos uno nuevo, podemos actualizar GRUB simplemente arrancando el S.O. Linux y tecleando el comando “**sudo update-grub**” y se actualizará su configuración con los nuevos sistemas.



**Nota:** no te preocupes si nunca has visto el comando, más adelante hay un capítulo dedicado a ponernos al día sobre Linux Ubuntu.

**Nota:** existen gestores de arranque comerciales y válidos en principio para cualquier S.O., como por ejemplo es el caso de “BootIt”, del que podemos descargar una versión de prueba en:

<https://www.terabyteunlimited.com/bootit-bare-metal.htm>

## Ejemplo de arranque dual

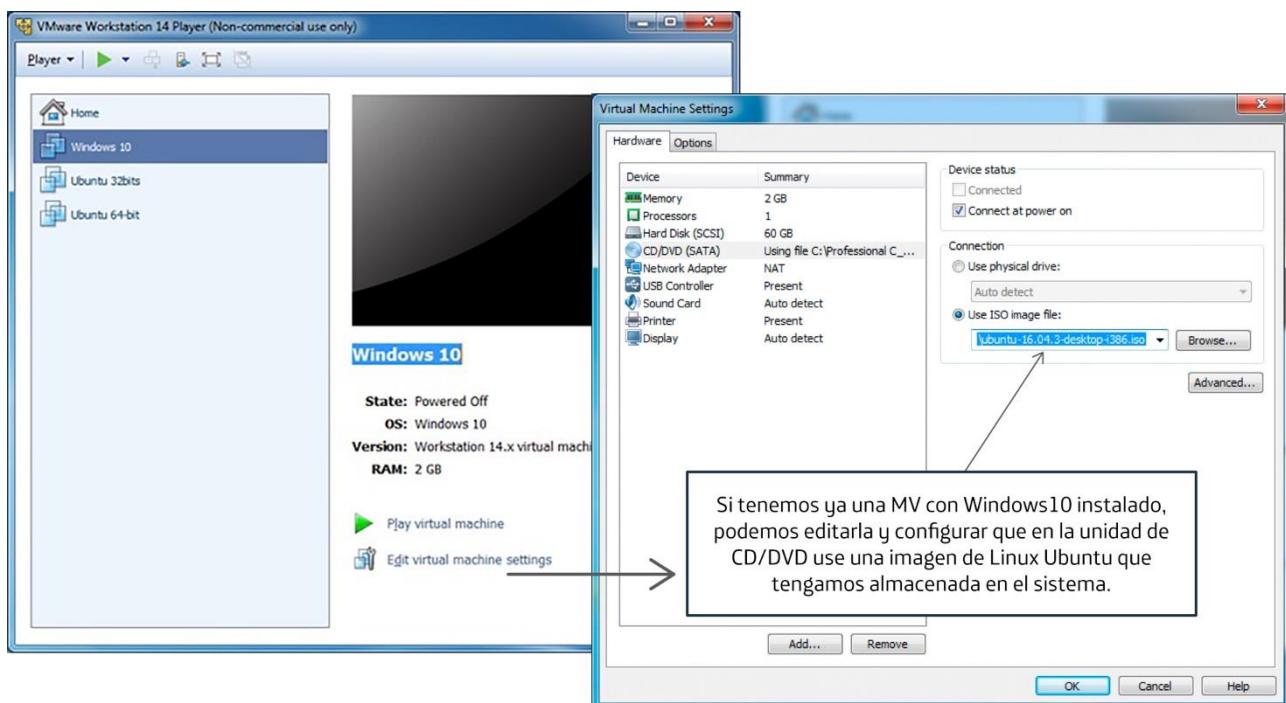
Para que veas un ejemplo de arranque dual, donde el gestor GRUB nos ofrece la posibilidad de elegir entre arrancar un sistema operativo u otro (y los dos ya están instalados en el equipo), te mostramos cómo sería dentro de una máquina virtual en el siguiente [vídeo](#).

Como has visto, GRUB nos permite elegir cuál de los dos sistemas operativos instalados en el equipo (virtual) deseamos arrancar. Además nos ofrece otras opciones, como realizar un test de memoria o entrar en funciones avanzadas del arranque de Ubuntu.

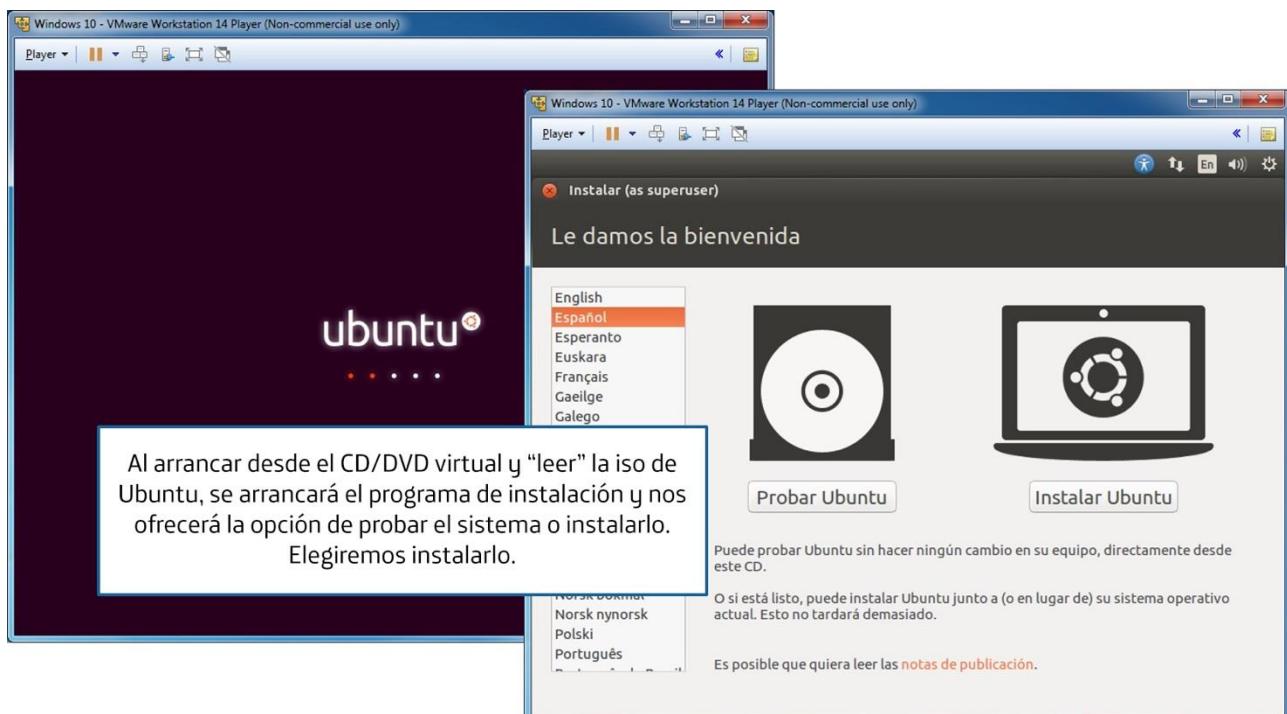
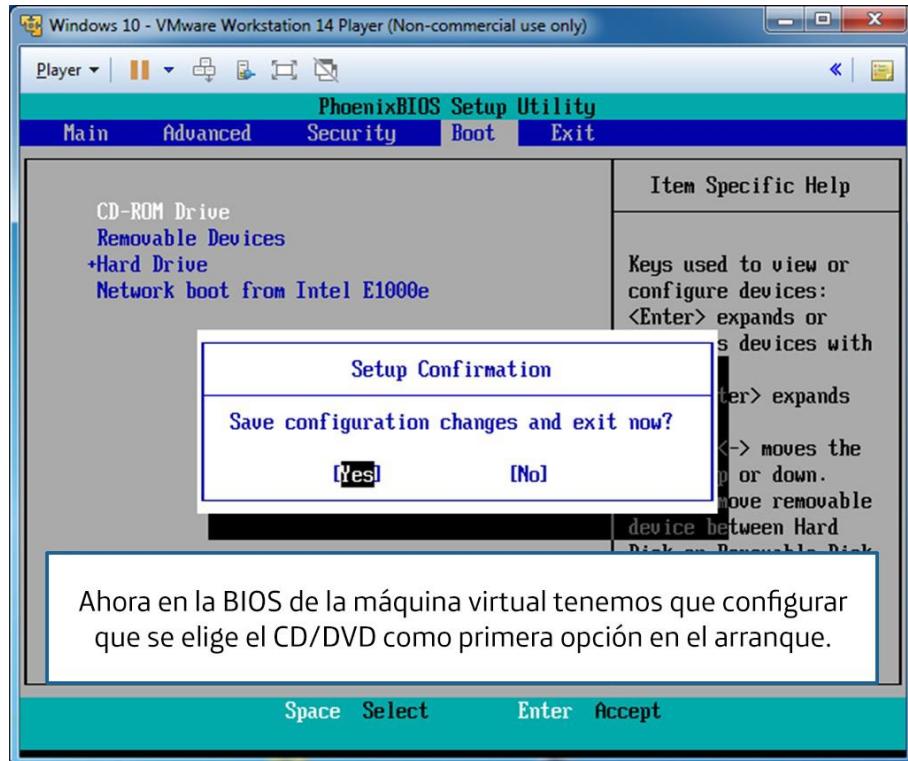
También habrás observado que hemos usado otro SW de virtualización (VirtualBox), pero tampoco queremos distraerte con ello. Nosotros de momento seguiremos usando VMware.

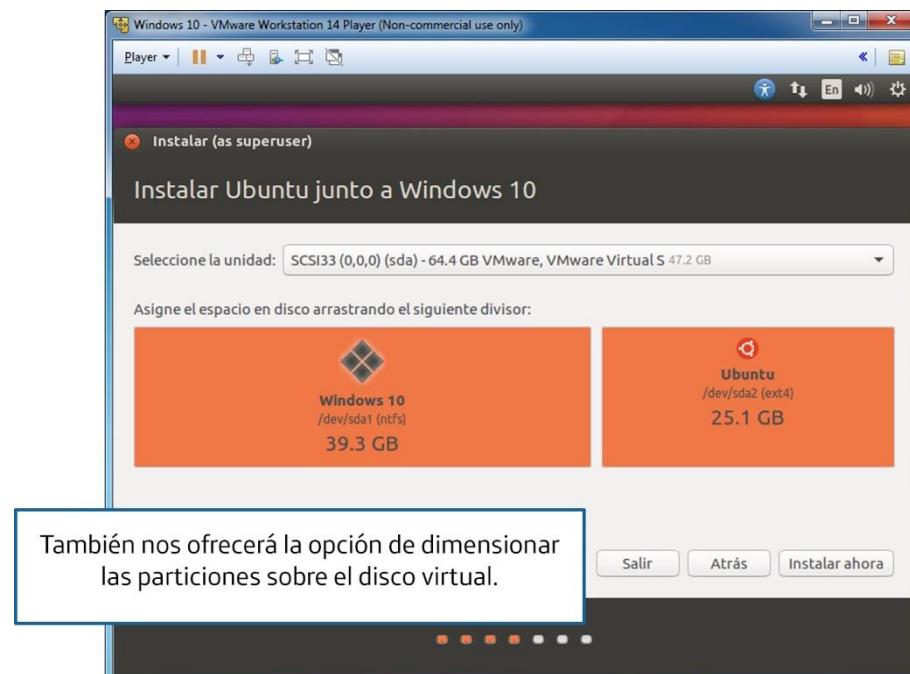
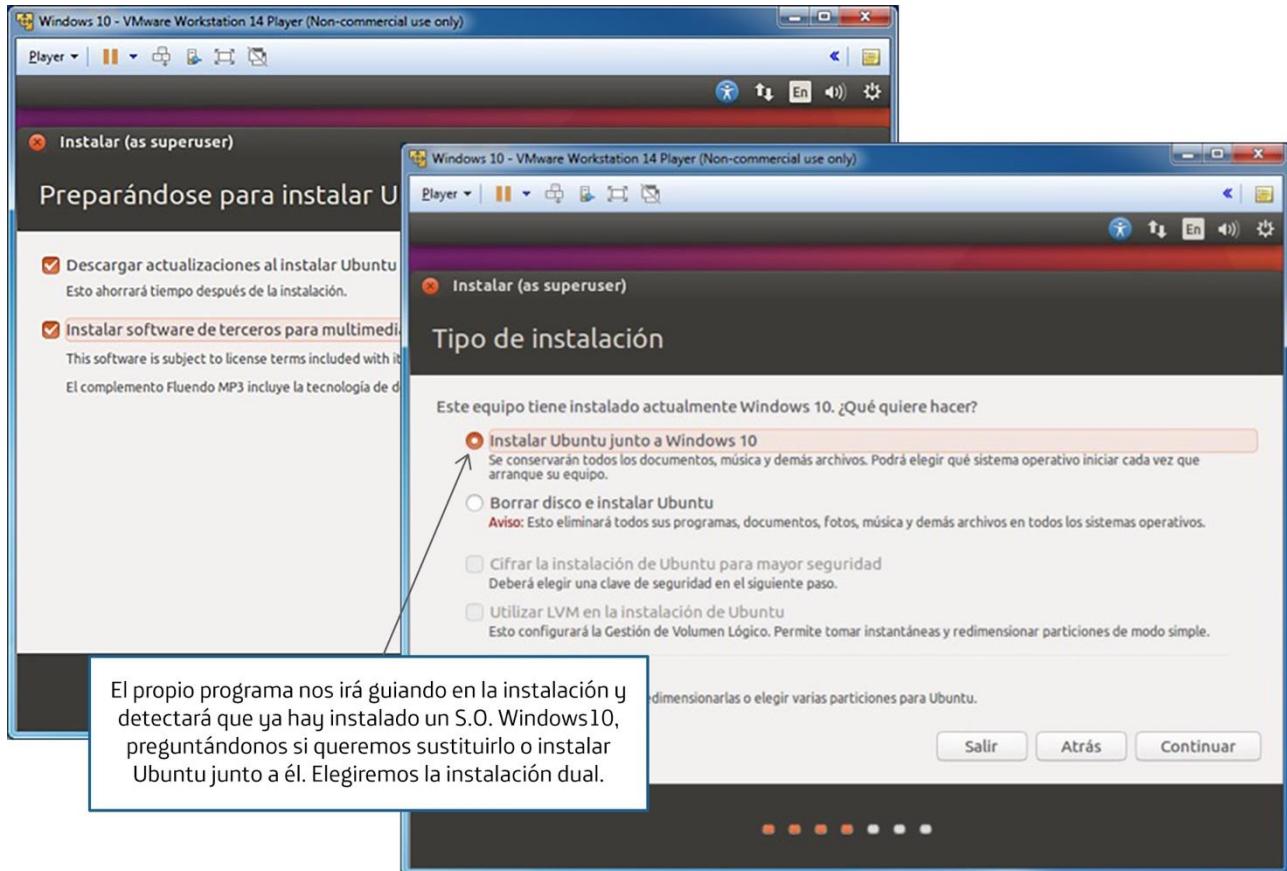
## Configuración del arranque dual en la máquina virtual

Vamos a mostrarte cómo crear un arranque dual con Ubuntu sobre una máquina virtual con Windows 10 ya instalado. Te mostramos la secuencia de pantallas con los comentarios sobre ellas. El proceso lleva cierto tiempo, pero si sigues los pasos indicados debería realizarse sin problemas.

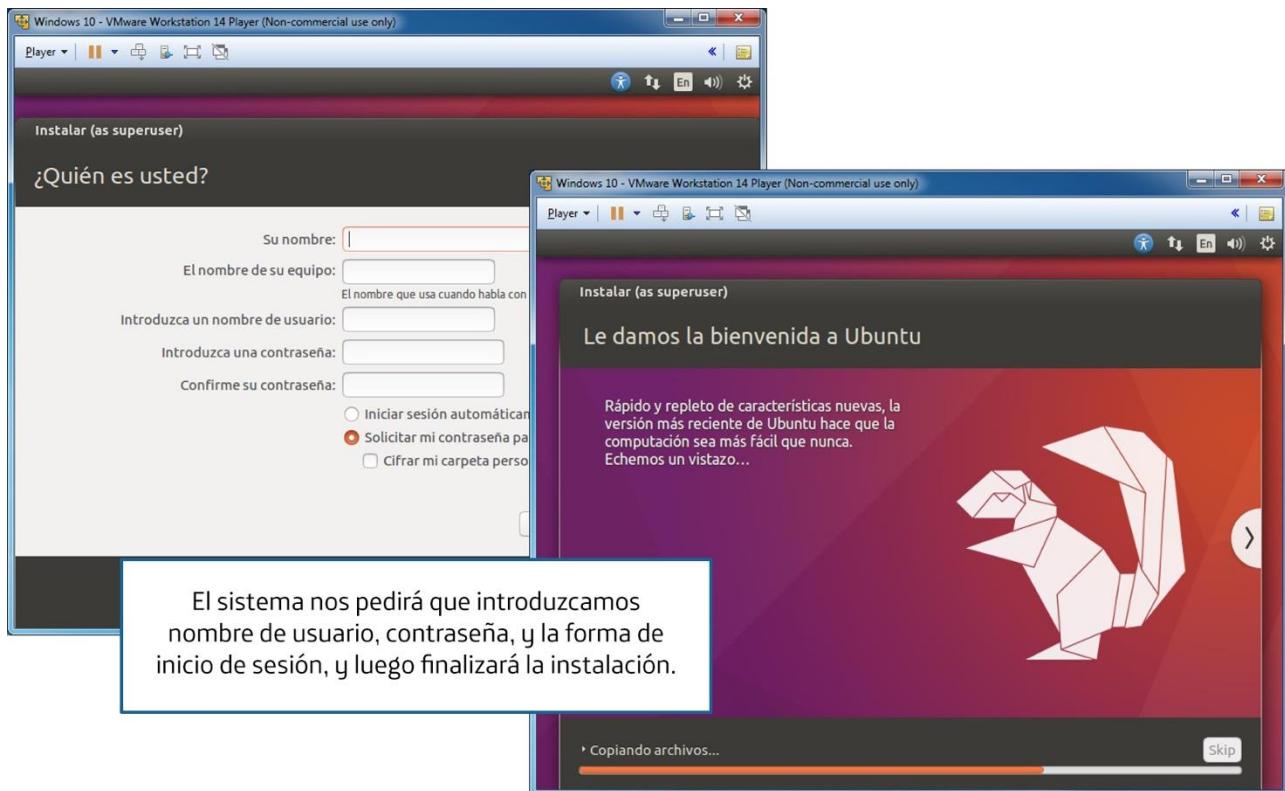
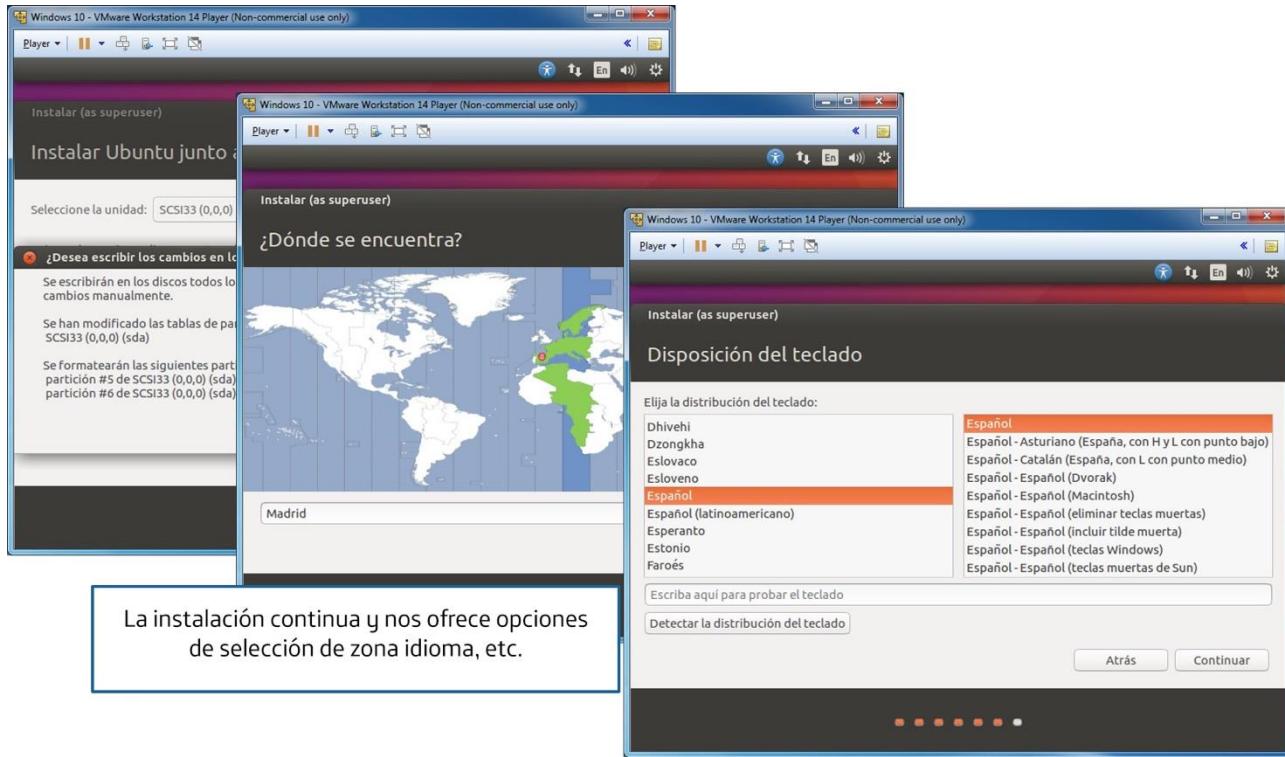


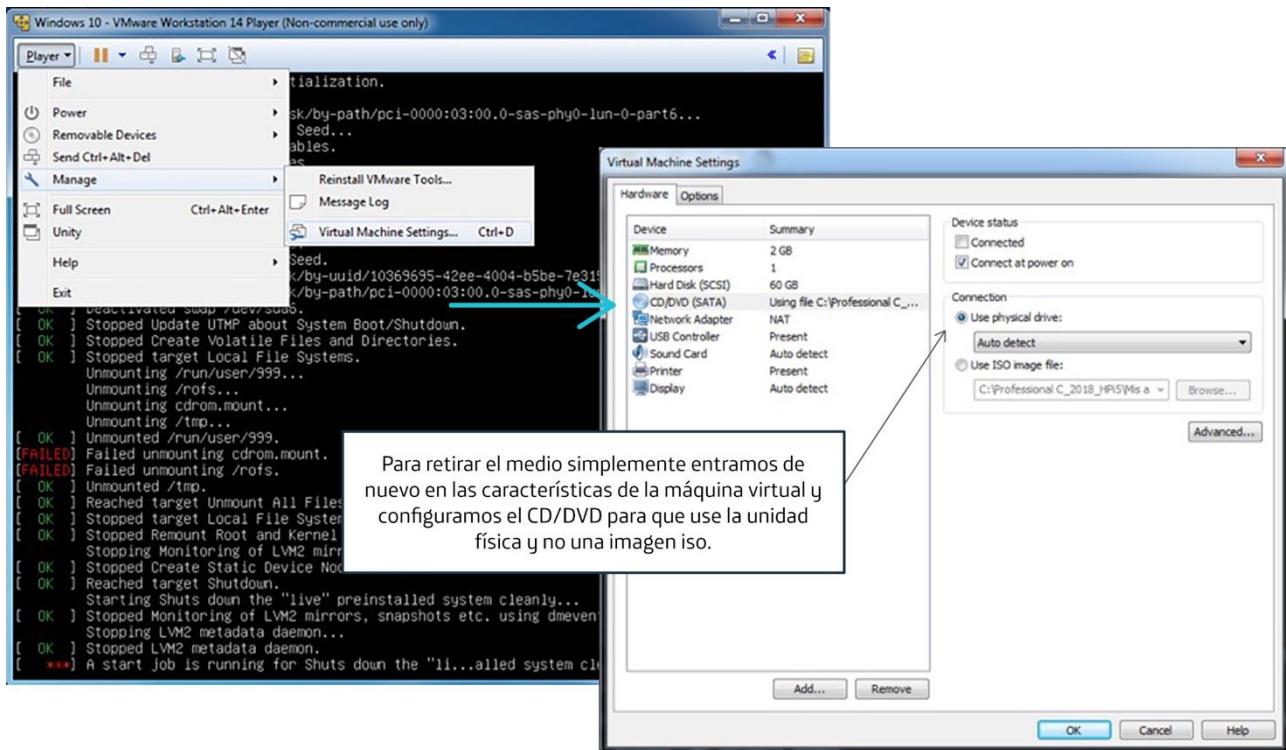
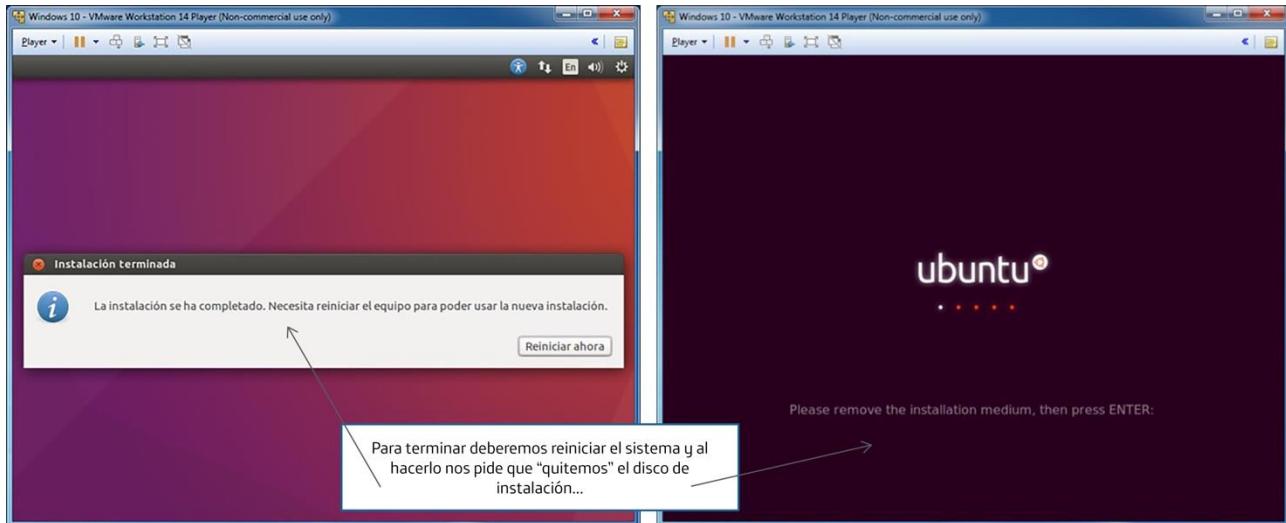
Secuencia de creación de arranque dual sobre MV VMWare con Windows 10

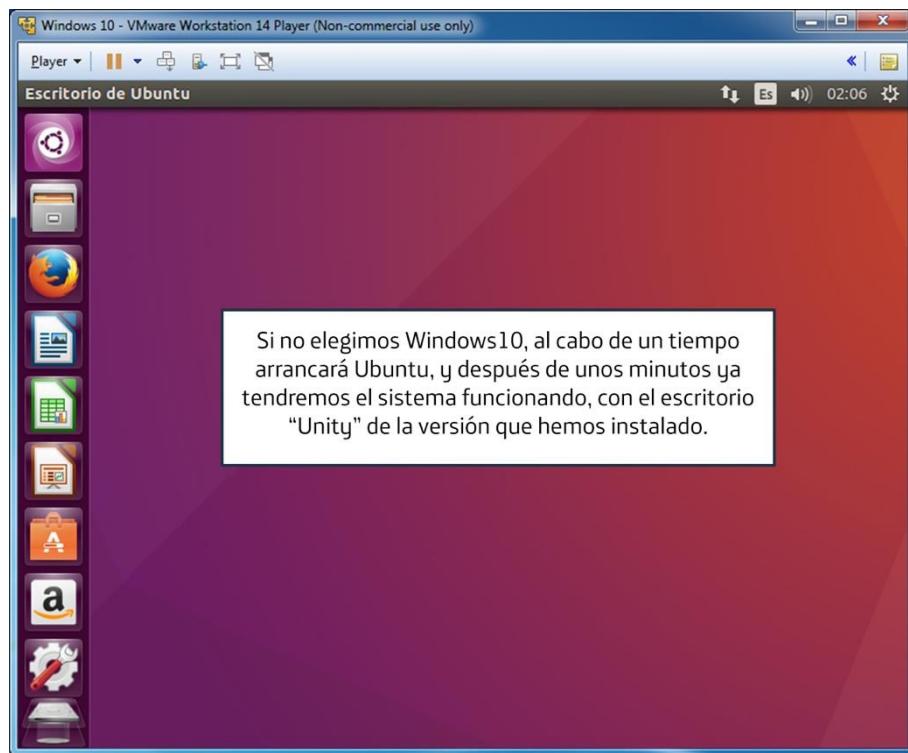
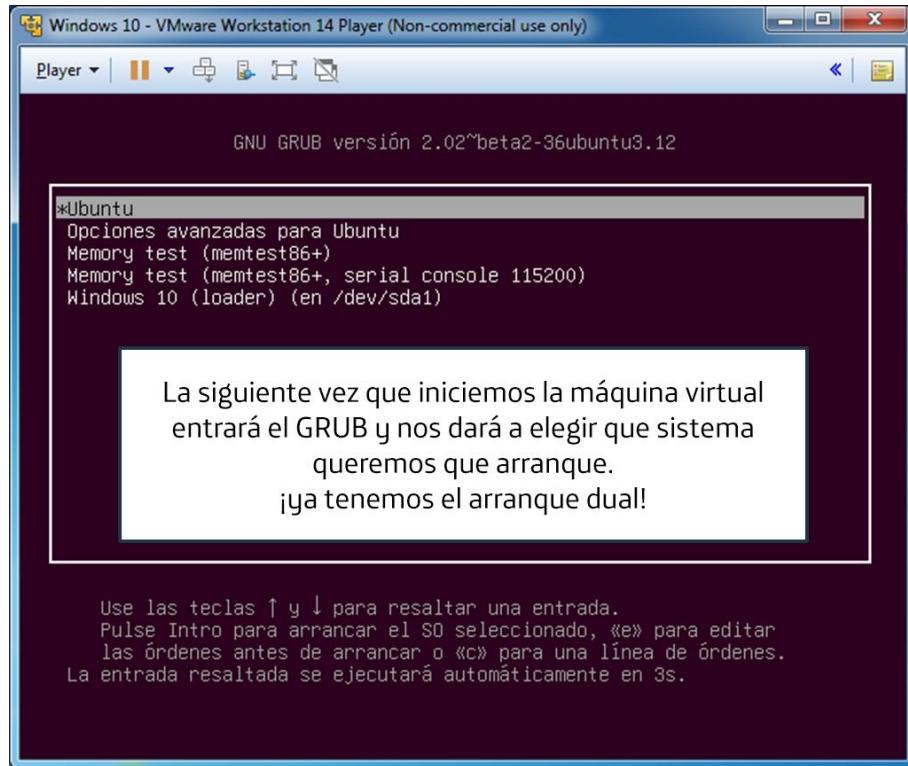




## 2.4. Configuraciones multisistema







**Nota:** no te preocupes si ahora no entramos en los detalles de Linux Ubuntu. Dedicaremos un tema específicamente a ello. En este ejemplo solo pretendemos que veas cómo funciona un arranque dual con GRUB.

# Reparar el gestor de arranque

Si necesitamos reparar el gestor de arranque normalmente tenemos un problema “serio”, puesto que lo más probable es que nuestro sistema no esté siendo capaz de arrancar.

Disponer de **copias de seguridad** (del sistema y de los datos) y de **imágenes de instalación** del sistema operativo que estemos usando es algo esencial. También podemos disponer de algún dispositivo configurado para autoarrancar un S.O. desde alguna unidad externa (lo que llamamos un “**Live CD**” o un “**Live USB**”) con un S.O. igual al que estemos usando u otro con funcionalidades de reparación o análisis forense.

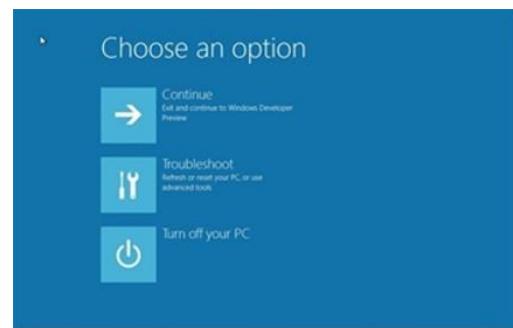
Además, si tenemos “montados” varios sistemas operativos sobre nuestra máquina, reparar el sector de arranque puede ser más complejo, pues no todos los sistemas admiten el mismo formato de MBR.

Veamos algunas de las diferencias principales entre sistemas Windows y sistemas Unix/Linux.

## Reparar gestores de arranque en sistemas Windows

Normalmente **Windows se instala en la primera partición del disco** y coloca su tabla de particiones y el enlace a su gestor de arranque en el primer sector de la partición. Si por cualquier razón se daña el registro de arranque (*boot record*) tendremos que repararlo para que pueda arrancar el sistema.

Para reparar el MBR en sistemas Windows una opción puede ser insertar el CD/DVD del S.O. y arrancar el equipo, pero en vez de elegir “reinstalar”, escoger la opción de “**reparar el equipo**”. Las opciones de reparación varían según la versión de Windows que tengamos, y antes de proceder conviene leer el manual para conocerlas.



Por otro lado, en el CD de recuperación (y a veces en los de instalación), Microsoft suele incluir un **comando para reparar el “boot sector”** llamado “**FIXBOOT**” (para reparar la referencia al sector de arranque de la primera partición en el archivo *bootloader*) y otro para reparar el MBR llamado “**FIXMBR**” (para reparar la referencia en el MBR de la primera partición).

El comando **FIXBOOT** asocia el MBR al sector de arranque de la primera partición, con lo cual podremos iniciar automáticamente el sistema operativo Windows. Si estamos usando un gestor de arranque dual como GRUB, después de usar los comandos de Windows tendremos que reinstalarlo en el MBR para seguir efectuando el arranque dual y configurarlo de nuevo para tener la opción de cargar el "bootloader" de Windows o el del otro sistema.

## Reparar gestores de arranque en sistemas Unix/Linux

Como el MBR se encuentra en el disco **es aconsejable hacer copias de seguridad** para poder recuperarlo más tarde, aunque debemos tener en cuenta que si restauramos el MBR con una tabla de particiones diferente sus datos serán ilegibles y será casi imposible recuperarlo.

Para hacer una copia del MBR podemos usar el comando:

```
dd if=/dev/sda of=/path/mbr-backup bs=512 count=1
```

Y para restaurarlo:

```
dd if=/path/mbr-backup of=/dev/sda bs=512 count=1
```

Borrar el MBR también puede ser útil por ejemplo para hacer una reinstalación completa de otro sistema operativo, y para borrarlo es suficiente con borrar los primeros 446 bits MBR (el resto es la tabla de particiones, como hemos visto). Podemos hacerlo con el comando:

```
dd if=/dev/zero of=/dev/sda bs=446 count=1
```



**Nota:** en los ejemplos anteriores se está usando el comando "**dd**" (Dataset Definition), con el cual prácticamente se puede modificar lo mismo sobre discos y particiones que con algunos programas comerciales como Norton Ghost o libres como CloneZilla.

Existen disponibles programas comerciales diversos (propietarios y libres) entre cuyas utilidades está el poder recuperar el gestor de arranque del equipo, algunas de ellas especializadas en la gestión del MBR y otras con mayor alcance pero con la función de recuperación incorporada, como Norton Ghost.

## Sustitución del gestor de arranque

Como hemos visto, una de las formas de sustituir el gestor de arranque es durante la instalación de una distribución Linux que tenga GRUB y hacerlo en último lugar sobre los S.O. ya instalados, de forma que los reconozca y los incorpore al menú inicial de arranque.

Sin embargo, en algunos casos podemos necesitar disponer de otros métodos para sustituirlo, sobre todo en función de la configuración de nuestro equipo, si tiene BIOS o UEFI, etc., y en este caso debemos tener en cuenta algunas cosas:



1. Los gestores de arranque múltiple de sistemas propietarios pueden a veces dar problemas al intentar incluir sistemas de distribución libre (tipo Linux por ejemplo).
2. Algunas configuraciones pueden hacer fallar la instalación de GRUB y no arrancar con opción múltiple, manteniendo el arranque automático del sistema más antiguo.
3. Si elegimos instalar un gestor de arranque independiente (por ejemplo “**rEFInd**”) el proveedor del SW normalmente nos dará **varios modos de instalarlo**: mediante un CD/DVD o un USB autoarrancable, con un archivo comprimido, ejecutando un programa instalador, etc.
4. Algunos paquetes SW que nos proporcionan gestores de arranque pueden tener limitaciones, por ejemplo pueden no ser “**multi-fs**” (capaces de acceder a archivos contenidos en una partición diferente de la que están instalados).

**Por todo ello, debemos estudiar bien las características concretas del gestor de arranque que deseamos instalar y, como siempre se recomienda, hacer un *backup* de nuestro sistema antes de iniciar cualquier cambio.**

## Ficheros necesarios para el arranque

Aunque la estructura y diferencias de las distintas familias de sistemas operativos son bastantes, y cada uno de ellos tendrá su propia configuración de módulos y ficheros necesarios para funcionar, podemos establecer de forma general los siguientes **elementos (ficheros y código) necesarios** para la instalación y arranque de un S.O.:

1. **Programa cargador:** “*boot program*”, que como vimos está en los primeros sectores del disco (MBR) y tiene un tamaño definido. Dará paso al gestor de arranque.
2. **Kernel del sistema operativo:** la mayoría se encuentran compilados en módulos que se van cargando cuando se necesitan, y entre ellos estará un módulo de comprobación del HW y un cargador inicial del resto de módulos.
3. **Ficheros de inicialización,** parametrización y configuración con los datos necesarios para establecer los niveles de ejecución y la personalización del sistema.
4. **Un programa finalizador** del proceso de arranque que deja al sistema disponible para el usuario, y que entre otras cosas puede realizar:
  - Verificar los sistemas de ficheros.
  - Montaje de los sistemas de ficheros y directorios permanentes.
  - Activar las áreas de “*swapping*” e intercambio.
  - Activar los procesos en segundo plano y los accesos a las redes de comunicación.
  - Habilitar la interfaz de entrada (“*login*”) para los usuarios.

# Instalaciones múltiples y actualizaciones

## Requisitos, versiones y licencias

Cualquier programa o aplicación que usemos, tanto a nivel personal como empresarial, está normalmente sujeto a unas condiciones de uso.

En el caso del SW libre estas pueden ser más flexibles y permisivas, pero igualmente debemos dedicar esfuerzo a conocerlas y saber bajo qué condiciones el proveedor nos autoriza a usar ese SW.

En el caso del uso profesional/corporativo esto es todavía más importante, pues puede conllevar repercusiones legales de alto impacto para el negocio, y además suele resultar muy compleja la gestión de las versiones y licencias de todo el SW manejado por una compañía.

Esta gestión del SW que empleamos es en los últimos tiempos cada vez más compleja y, sobre todo en el ámbito empresarial, se suele englobar bajo el término **“gestión de los activos SW”** (**“Software Asset Management – SAM”**) y suele abarcar los siguientes procesos:

1. **Realización de un inventario del software empleado** (sistemas y aplicaciones) tanto en número de usuarios como en instalaciones, entorno de ejecución, etc., identificando todos los programas instalados y/o en uso.
2. **Verificación de la correspondencia entre los programas instalados y las licencias adquiridas** para comprobar que el uso que se está haciendo se corresponde con los derechos legales comprados.
3. **Revisión de políticas y procedimientos de gestión legal del software** a nivel de toda la empresa.
4. **Desarrollo de un plan de acción de gestión de activos de software**, incorporando esta actividad como una más en las operaciones de la empresa.

**Esta metodología SAM incluso tiene un estándar internacional que es la norma UNE-ISO/IEC 19770-1:2008, y AENOR proporciona un “Certificado para la Gestión del Software Original”.**

## Detalles de instalación

En general será necesario hacer una **auditoría sobre el SW** empleado para conocer los detalles de instalación y el uso de cada aplicación, tanto si son usadas localmente como en un entorno de ejecución en red, para poder **asegurar que se están utilizando de forma adecuada** y según las condiciones establecidas con el proveedor de la licencia (tanto si es SW propietario como si es SW libre), que también puede estar sujeto a ciertas condiciones de uso. Por ejemplo, un determinado programa puede ser "libre" solamente para uso no comercial.



El control de las licencias y las versiones del SW instalado no solamente se hace por "imperativos legales", sino también porque es necesario para **optimizar el rendimiento de la organización**, controlar y minimizar los costes, prevenir riesgos derivados del uso no autorizado de software, incompatibilidades entre diferentes versiones de un programa, o no poder establecer unas directivas de seguridad homogéneas para todos los entornos de trabajo.

## Actualización de sistemas operativos y aplicaciones

La actualización del sistema operativo y las aplicaciones instaladas en el ordenador resulta un aspecto esencial del mantenimiento del equipo informático.

1. **MEJORA DE LA SEGURIDAD.** Cualquier sistema operativo o aplicación puede presentar fallos y errores que deben ser corregidos. También pueden aparecer **problemas de vulnerabilidad** que pongan en riesgo la seguridad de nuestro sistema y nuestros datos. Las actualizaciones suelen incorporar parches correctores y mecanismos de protección frente a las amenazas que se van descubriendo. Un ejemplo claro lo tenemos en los antivirus, ¿a alguien le sirve de algo un programa antivirus que no esté actualizado?
2. **NUEVAS FUNCIONALIDADES.** Las actualizaciones sirven también para **agregar nuevas funcionalidades** al SW, hacerlo compatible con nuevos protocolos o formatos adoptados por el mercado y aumentar en general la productividad del sistema.
3. **SITIOS DE CONFIANZA.** Es sumamente importante **descargar las actualizaciones de sitios de confianza**, a ser posible de los propios fabricantes del SW, y huir de instalar versiones sin referencias o ilegales por mucha funcionalidad que ofrezcan.

## Actualizar a una versión superior (update)

Mantener actualizadas las aplicaciones que utilizamos es una práctica importante, que además de proporcionarnos la funcionalidad más moderna (y generalmente más completa) de los programas, suele ayudar a mejorar la seguridad del sistema y la compatibilidad entre ellas.

Los fabricantes de SW nos ofrecen a menudo mecanismos de actualización que **difieren mucho de unas aplicaciones a otras**, desde la automatización de las actualizaciones (como nos ofrece por ejemplo Microsoft para Office), hasta la reinstalación completa a una nueva versión si es necesario.



En otros casos en los que no disponemos de aviso automático de actualización es el usuario el que normalmente se encarga de hacerlo manualmente, casi siempre al detectar que queremos una nueva funcionalidad que no está en la versión que tenemos instalada.

## Cambiar a una versión inferior (downgrade)

Algo menos común, pero también necesario a veces, es el cambio a una versión más antigua de alguna aplicación.

En general las versiones más modernas del SW son “**compatibles hacia atrás**”, es decir, soportan la funcionalidad y los formatos de trabajo de sus ediciones anteriores, pero hay algunos casos en los que no es así o, que aún siéndolo, se hace necesario reinstalar la versión antigua.

Así como en los procesos de “**update**” (**actualización**) solemos contar con ayudas para que la propia aplicación se actualice a la versión moderna, al contrario esto no suele suceder. En muchos casos el proceso pasa por la desinstalación de la versión actual y la instalación de la versión antigua en su lugar (o en otro entorno de trabajo, directorio, etc.).

Existen algunas formas de poder hacer el “**downgrade**” sin desinstalar la aplicación actual, pero dependen del entorno y S.O. que estemos empleando. En entornos heterogéneos, como puede ser Android, incluso encontramos aplicaciones específicas que nos ayudan a hacer el cambio a la versión anterior de una aplicación sin perder los datos actuales.



Recuerda que para poder probar versiones sin desinstalar las actualmente en servicio puede ser muy útil la creación de **entornos de trabajo virtuales (máquinas virtuales)** que nos permitan, bien probar la nueva versión antes de actualizarla, o bien trabajar temporalmente con la versión antigua sin desinstalar la más actual.

## Uso de instalaciones desatendidas

Los administradores que necesiten instalar un sistema operativo en un gran número de equipos tienen la opción de recurrir a las “**instalaciones desatendidas**”, en las cuales, como hemos comentado, se recurre a **ficheros (“scripts”)** de configuración y herramientas de implantación que realizan acciones de forma automática y aportan datos al proceso de instalación en vez de hacerlo el propio administrador.

El hecho de que la instalación sea desatendida no implica que la carga del S.O. deba hacerse desde una red, pero esta opción es una de las más ventajosas para los administradores.

Los **scripts de configuración** deben aportar al **proceso los datos y órdenes necesarios para llevar a cabo la instalación** correcta del S.O., y aunque ello dependerá de cada sistema, estos parámetros pueden ser, por ejemplo:

- Usuario y *password* de administración del equipo.
- Zona horaria y región.
- Organización y nombre del equipo.
- Número de licencia.
- Comandos para la instalación de aplicaciones y protocolos, por ejemplo:
  - Instalación de un paquete ofimático.
  - Instalación de los componentes de red y la tarjeta de red si es necesario.
  - Datos para la configuración de los protocolos TCP/IP (mecanismo de obtención de la IP, máscara de red, puerta de enlace y DNS predeterminados, etc.).

## Instalación desatendida en Windows

Una forma de poder realizar instalaciones desatendidas en Windows es usando el archivo de instalación llamado "**Unattend.xml**".

Como hemos comentado, este archivo contendrá los parámetros de respuesta que necesita el asistente de instalación, de forma que cuando se arranca el proceso de instalación de Windows 10, este verifica que el dispositivo de almacenamiento desde el que se realiza (disco, USB) contiene este archivo en su directorio raíz.

Existen herramientas que nos ayudan a generar estos archivos en un formato (XML) adecuado, pero también podemos crearlos con un editor de texto para programación.

```
<?xml version="1.0" ?>
<unattend xmlns="urn:schemas-microsoft-com:unattend">
  <settings pass="windowsPE">
    <component name="Microsoft-Windows-Setup" publicKeyToken="31bf3856ad364e35"
language="neutral" versionScope="nonSxS" processorArchitecture="x86">
      <WindowsDeploymentServices>
        <Login>
          <WillShowUI>OnError</WillShowUI>
          <Credentials>
            <Username>username</Username>
            <Domain>Fabrikam.com</Domain>
            <Password>my_password</Password>
          </Credentials>
        </Login>
        <ImageSelection>
          <WillShowUI>OnError</WillShowUI>
          <InstallImage>
            <ImageName>Windows Vista with Office</ImageName>
            <ImageGroup>ImageGroup1</ImageGroup>
            <Filename>Install.wim</Filename>
          </InstallImage>
          <InstallTo>
            <DiskID></DiskID>
            <PartitionID>1</PartitionID>
          </InstallTo>
        </ImageSelection>
      </WindowsDeploymentServices>
      <DiskConfiguration>
        <WillShowUI>OnError</WillShowUI>
        <Disk>
          <DiskID>0</DiskID>
          <WillWipeDisk>false</WillWipeDisk>
        </Disk>
      </DiskConfiguration>
    </component>
  </settings>
</unattend>
```

*Ejemplo de archivo para instalación desatendida.*

## Instalación desatendida en Linux

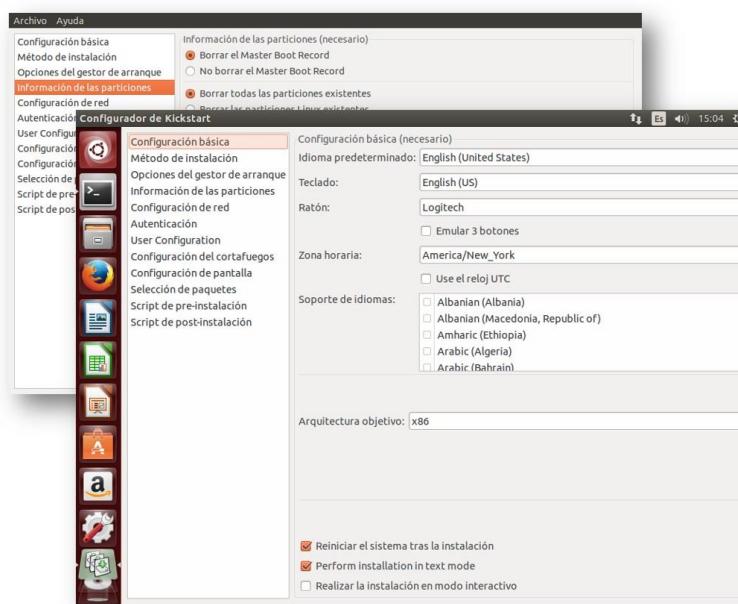
Las instalaciones desatendidas en la familia GNU/Linux también necesitan de la creación del fichero de respuestas para el programa instalador. En ese fichero básicamente le proporcionarán información de:

- Tamaño y ubicación de las particiones a crear.
- Paquetes a instalar.
- Idioma a seleccionar para la interfaz de usuario.
- Parámetros de conexión a servicios externos.
- Script de post-instalación para personalizar la instalación.

En el entorno GNU/Linux las diferentes distribuciones tienen, en principio, diferentes sistemas de auto-instalación. Los más comunes son:

- **Kickstart** en sistemas RedHat y Fedora.
- **AutoYast** en sistemas SuSe.
- **Preseed** en Debian / Ubuntu.

Aunque esto no es totalmente generalizable y distribuciones de Ubuntu pueden hacerse desatendidas con otro procedimiento como Kickstart por ejemplo, como se muestra en la imagen.



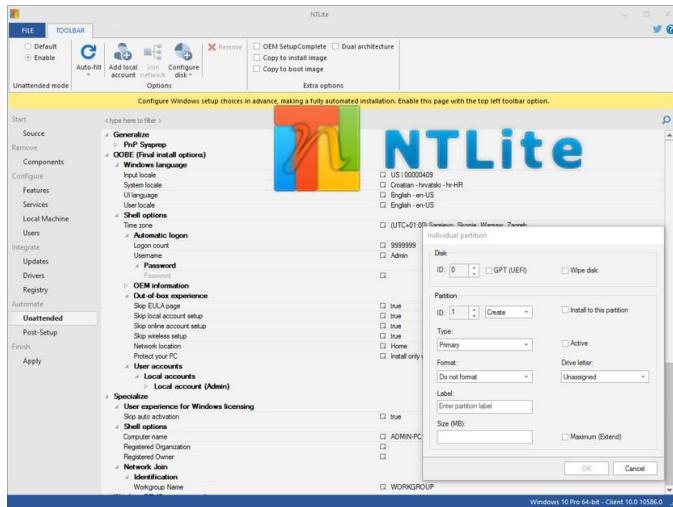
### Instalación desatendida en Linux, proceso a seguir:

5. Arranque del programa “cliente” de la autoinstalación, bien en un CD o desde la red.
6. Se le pasa como **parámetro** del *kernel* **la ubicación (servidor) del script** de autoinstalación.
7. El instalador se **descarga el fichero de respuestas** (*script*), que incluirá también la ubicación de la imagen del S.O. (servidor).
8. Se **realiza la instalación del S.O.** con los parámetros especificados en el *script* y descargando los paquetes de instalación desde el servidor indicado.
9. Al finalizar la instalación se produce un **reinicio del sistema** y (si todo ha ido bien) arranca ya disponible para el usuario.

## Herramientas - Instaladores

Las herramientas de instalación desatendida más comunes lo que suelen hacer es proporcionar **ayuda para la generación de los ficheros (scripts)** que han de acompañar a la instalación del S.O. para automatizarla. Existen herramientas externas en forma de programas ejecutables o de interfaz *online*, propietarias o libres, etc., sobre todo para entornos Windows, porque para el entorno GNU/Linux lo más común son los tres procedimientos vistos antes (Kickstart, AutoYast y Preseed).

Ejemplos de herramientas para el entorno Windows pueden ser:



**NTLite** es una herramienta que nos permite crear discos de instalación de Windows personalizados y desatendidos, configurando la instalación del S.O., los controladores y parámetros de configuración y los programas asociados necesarios.

Además de los discos de instalación, la herramienta nos permite gestionar particiones, remover componentes, administrar *drivers*, etc.

**Windows Answer File Generator** es una herramienta en línea (web) que nos permite crear el fichero XML para instalaciones desatendidas de los diferentes paquetes de Windows.

Eligiendo la versión de Windows para la que queremos el fichero, la web nos ofrecerá un menú a llenar y creará el fichero XML que luego podemos descargar.

The screenshot shows the Windows Answer File Generator website. At the top, it says 'Windows Answer File Generator Inspired by Windows System Image Manager.' Below that is a navigation bar with links for Home, Desktop, Server, Office, Services, Tutorial, and Contact. The main content area is titled 'Windows 10 Answer File Generator for x86 and x64 processors'. It says 'Below are basic settings that can be configured during and after installation. What you type will automatically be updated in the "Output" area.' There are sections for 'Installation Notes' (with a text input field), 'Notes' (with a text input field), and 'General Settings' (with dropdown menus for Product Key, Accept EULA, Skip automatic activation, Skip License Realm, Setup Language, Computer Name, and Organization Name).

# Controladores de dispositivos

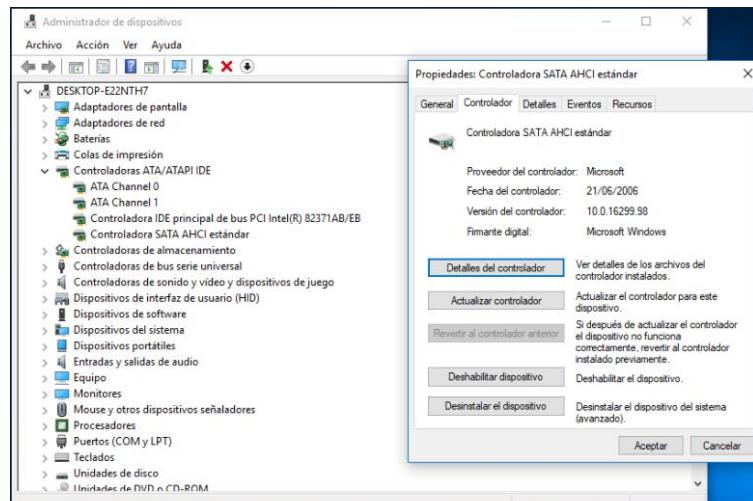
Como sabemos, la función de un controlador de dispositivo es servir de intermediario entre el S.O. y el dispositivo físico, permitiendo por ejemplo que la CPU del sistema siga ejecutando procesos durante los tiempos de espera de respuesta del dispositivo físico (impresora, teclado, etc.).

Estas funciones de los controladores pueden resumirse entonces en:

- Proporcionar **interfaces SW lo más simples y estándar posibles** en la gestión de las operaciones de E/S, ocultando los detalles últimos del HW.
- **Optimizar al máximo el tiempo de la CPU**, permitiendo que siga ejecutando acciones mientras se realiza la operación de E/S.

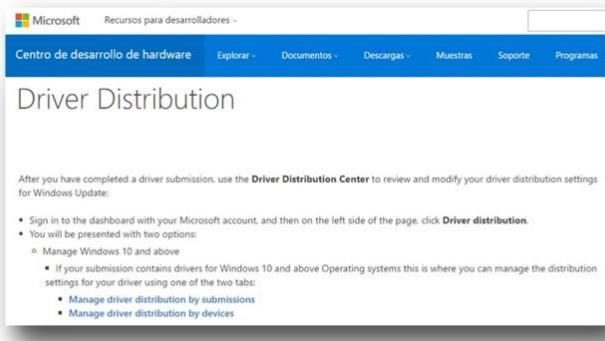
La distribución de los controladores de dispositivos es realizada normalmente por el propio fabricante del dispositivo, y normalmente se generan los *drivers* diferenciados para cada S.O., especificando además las condiciones de instalación y requisitos del mismo, aunque a menudo pueden surgir problemas de compatibilidades y versiones de "**packs**" necesarias para el buen funcionamiento del *driver*.

Para los dispositivos antiguos que pueden estar ya fuera del mercado o haber sido descontinuados en su soporte, a menudo existe el problema de encontrar el *driver* adecuado. Además algunos S.O. exigen que los controladores dispongan de firma digital para permitir que sean instalados, como ocurre en los sistemas Windows desde Windows Vista y posteriores versiones de Windows en los equipos de 64 bits.



Normalmente, una vez que el paquete SW del controlador está preparado para su distribución por parte del fabricante, esta puede hacerse de varias formas:

- Las versiones del SW del controlador pueden transferirse a un **ordenador de prueba**.
- El paquete SW puede **grabarse en un CD/DVD** u otro medio de almacenamiento e incluido en el paquete de entrega del dispositivo físico en su venta.
- Puede **alojarse el SW en algún sitio web disponible para su descarga** (esto es especialmente útil si hay que distribuir nuevas versiones y actualizaciones).



En el caso de Microsoft, por ejemplo, los *drivers* que han pasado por el proceso de pruebas "**Windows Logo Program**" pueden distribuirse a través del programa de actualización de Windows Update.

## Herramientas de gestión de drivers

Existen muchas herramientas que nos permiten una adecuada administración de los *drivers* instalados en nuestro equipo.

Algunas de ellas están incluidas en los propios paquetes del sistema operativo o son accesibles vía consola de comandos y otras (como suele suceder) son utilidades desarrolladas por terceros.



Por ejemplo, en el entorno Windows podemos hacer una exportación de los controladores a través de la **PowerShell de Windows**, y si trabajamos con Windows Server, Microsoft nos proporciona una buena guía para la gestión de los controladores del servidor o una herramienta para verificar el estado de los controladores de dominio.

## Gestión de los drivers

Existen también multitud de **herramientas (gratuitas y de pago)** para ayudarnos en la gestión (diagnóstico, actualización y *backup*) de los drivers de nuestro equipo.

Algunas de las que se pueden encontrar en la red para entorno Windows pueden ser:

- [Intel Driver Update Utility](#)
- [AMD Driver Autodetect](#)
- [DUMo Update Monitor](#)
- [Double Driver](#)
- [Snappy Driver Installer](#)
- [Device Doctor](#)
- [DriverMax](#)

Todas ellas normalmente nos ofrecen una interfaz gráfica con más o menos posibilidades que nos evitan la gestión vía comando. Algunas son herramientas gratuitas y otras no, algunas nos garantizan que están libres de “malware”, pero como con cualquier SW que vayamos a instalar en el equipo, antes de hacerlo debemos informarnos sobre su procedencia y en este sentido debe ser siempre más fiable aquel más cercano al fabricante del equipo o del S.O.

## Gestión de controladores en entornos GNU/Linux

En el caso de entornos GNU/Linux la gestión de los controladores a menudo se realiza a través del **entorno de comandos del propio sistema operativo** cuando el usuario tiene conocimiento experto, y las aplicaciones y utilidades dedicadas a facilitar el trabajo suelen ir ligadas a cada distribución GNU/Linux y no siempre van incluidas en el paquete inicial.

A menudo los desarrolladores que proporcionan ciertas utilidades GNU/Linux las empaquetan y ofrecen una nueva distribución dirigida a algún área de trabajo específica. Muchas de las utilidades desarrolladas pueden correr en el entorno de distribuciones similares, pero suele ser necesario un conocimiento más profundo de los sistemas para instalarlas correctamente.

En muchos casos lo que se nos ofrecen son listas de repositorios de controladores y herramientas de comprobación en línea que requieren también interacción vía comandos con el S.O.

Por ejemplo:

- [Debian HLC](#)
- [Linux drivers](#)



## Cómo hacer un "live USB" de Ubuntu

Cuando hablamos de un "live CD" o un "live USB" nos referimos a un dispositivo externo (CD/DVD o una memoria USB) preparado para servir como "disco de arranque" del sistema, es decir, que tendrá un sistema operativo instalado y un MBR que permitirá que la BIOS lo dirija y le pase el control de arranque del sistema operativo.

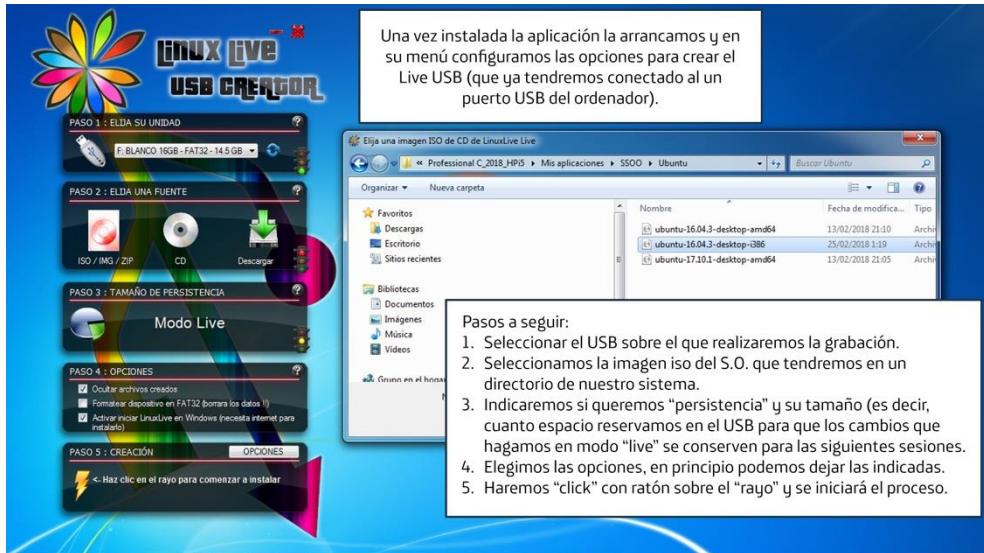
Esto quiere decir que no se trata simplemente de una memoria USB con una imagen ISO de un S.O. almacenada dentro, sino que debe estar formateada para permitir el arranque del sistema.

Lo más sencillo es hacerlo con alguna de las herramientas de creación que están disponibles en la red. Puedes elegir muchas de ellas (Rufus USB Creator, UnetBootin, LiLi USB Creator, etc.) y más o menos cada una ofrece una interfaz bastante sencilla para crear el dispositivo.

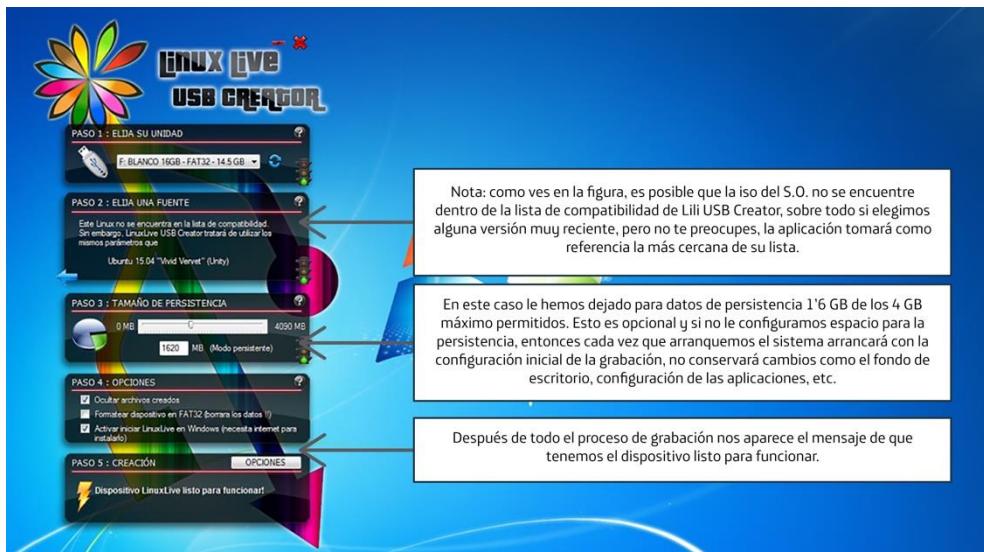
Te ofrecemos un ejemplo realizado con una de ellas, en este caso "Linux Live (Lili) USB Creator", que es muy fácil de utilizar. Lo primero será ir a la página del proveedor y descargarlo (<https://linuxliveusb.com>):



## Pasos para crear el "live USB"



### 1. – Configurar los parámetros de la aplicación.



### 2. – Activar la grabación hasta obtener el mensaje de final de proceso.

Para comprobar el funcionamiento del Live USB simplemente al encender el equipo activa las opciones de la BIOS y modifica el orden de búsqueda de S.O. para el arranque, haciendo que elija en primer lugar el USB.

Verás que arranca Ubuntu y te da la opción de instalar o probarlo sin modificar tu equipo. Debes elegir probar Ubuntu sin instalarlo. Al arrancar tendrás un sistema operativo Ubuntu totalmente funcional y cuando lo cierres y apagues tu equipo seguirá con el Windows que tenías sin haber sido modificado.

# Despedida

## Resumen

Has finalizado esta lección.

En esta lección nos hemos centrado en la problemática de instalar más de un sistema operativo sobre un mismo equipo. Como ves, además de disponer de las particiones necesarias y demás recursos, necesitamos tener un gestor de arranque que detecte la existencia de varios sistemas operativos y nos ofrezca la opción de elegir el que queremos arrancar.

Además debes recordar que es conveniente contar con los controladores adecuados instalados, y que con un poco de experiencia puedes realizar instalaciones "desatendidas" configurando previamente los parámetros de la instalación en ficheros adicionales (scripts).

Recuerda también que las acciones sobre el sistema operativo implican actuar sobre el "corazón" de la máquina, que sostiene y controla el funcionamiento de todo el sistema, por lo cual siempre que puedas probar las acciones sobre un entorno virtualizado será más conveniente, y en un entorno normal de trabajo deberemos actuar con mucho cuidado.