
TEMA 12: ARRANQUE DEL SISTEMA OPERATIVO

Índice

1. Introducción	2
2. BIOS y UEFI.....	2
3. Proceso de inicio común a Windows y Linux.....	5
Búsqueda del cargador de inicio en BIOS/MBR.....	7
Búsqueda del cargador de inicio en UEFI/GPT	9
4. Arranque de Windows.....	15
Opciones de arranque de Windows	16
Comando MSConfig.....	21
5. Arranque de Linux	25
Gestor de arranque GRUB.....	26
6. Sistemas arrancables desde dispositivos extraíbles.....	30
Crear un dispositivo USB de instalación de Windows desde Windows	31
Crear un dispositivo USB de instalación de Ubuntu (Live CD) desde Windows	31
Crear un dispositivo USB de instalación de Ubuntu (Live CD) desde Ubuntu	32

1. Introducción

En este tema vamos a ver cómo arranca un sistema operativo, desde que comienza el proceso de verificación del hardware por parte de la BIOS (o UEFI en los sistemas actuales) en el momento de encender el ordenador, hasta el momento en que el usuario inicia sesión en el equipo.

La primera parte del proceso de arranque es similar independientemente de si arranca un sistema Windows, Linux o un equipo con multiarranque. Pero una vez detectado el sistema operativo a arrancar, el proceso es diferente dependiendo del sistema operativo y la versión de éste.

Como aspecto importante, hay que diferenciar si el equipo incorpora las antiguas **BIOS** o por el contrario viene con el nuevo firmware **UEFI**. Dependiendo de este tipo de firmware básico, el arranque de los distintos sistemas operativos difiere bastante, pues también es diferente la instalación del propio sistema, el particionado de los discos y la seguridad en el proceso de inicio y carga del sistema operativo.

2. BIOS y UEFI

La **BIOS** (*Basic Input/Output System*, Sistema Básico de Entrada/Salida) es un conjunto de programas muy elementales, grabados en un chip de la placa base denominado ROM BIOS que se encarga de realizar las funciones necesarias para que el ordenador arranque. Se podría decir que es el firmware que lleva la placa base para poder arrancar el sistema, reconocer los dispositivos, cargar el sistema operativo y transferirle el control una vez cargado.

La BIOS proporciona una comunicación a bajo nivel permitiendo que el equipo funcione cuando se enciende, además de reconocer y comprobar los dispositivos del equipo (procesador, memoria RAM, discos duros, teclado, ...) durante el arranque del mismo en el proceso denominado **POST** (*Power On Self Test*, Auto Diagnóstico Al Encender). En caso de existir algún fallo durante el POST, normalmente advierte al usuario mediante unos pitidos que según el tipo de BIOS, el número de pitidos y la duración, el código de error puede ser uno u otro. Por último, la BIOS hace que el

Sistema Operativo se cargue en la memoria RAM del equipo en un proceso que vamos a estudiar en los puntos siguientes.

Debido a que la tecnología utilizada en la BIOS ha quedado obsoleta, y a que los equipos cada vez más potentes necesitaban de nuevos requerimientos durante el arranque, se empezó a trabajar en un nuevo estándar que reemplazara la BIOS.

Este nuevo estándar, denominado **EFI** (Extensible Firmware Interface) tomó forma sobre el año 2005 y ha ido poco a poco mejorándose e implantándose cada vez más en los nuevos equipos fabricados, siendo hoy el estándar incorporado en las placas bases nuevas.

Actualmente se le conoce como **UEFI** (Unified Extensible Firmware Interfaz), que fue el nombre que se le dio al consorcio formado por más de 140 compañías tecnológicas.

UEFI es una interfaz de firmware estándar para PCs, diseñada para reemplazar la BIOS. Se ha diseñado para mejorar la interoperabilidad del software y solucionar las limitaciones de la BIOS con muchísimas ventajas respecto a ésta.

Una de las principales características que incorpora UEFI es **Secure Boot**. Secure Boot es una tecnología incorporada en la UEFI que ayuda a proteger al equipo frente a software malicioso del tipo *bootkit*. El *bootkit* es un tipo de software que ataca al arranque del equipo, sustituyendo el código arrancable y llegando incluso a parchear el *kernel* del sistema operativo instalado para permitir ejecutar ciertos programas o quitar comprobaciones de seguridad (así funcionaría el *jailbreak* de los dispositivos Apple, por ejemplo).

Para prevenir estos ataques maliciosos intencionados, se incorporó el Secure Boot, que sólo permite arrancar aquellos sistemas operativos firmados digitalmente.

Esta medida de seguridad puede afectar de manera negativa a instalaciones de Linux. Aunque se puede desactivar el modo Secure Boot desde la configuración de UEFI, puede seguir dando problemas con equipos y sistemas operativos libres, sobre todo si

llevan un arranque dual con un sistema operativo propietario que necesita de la activación del Secure Boot.

A la hora de estudiar el proceso de arranque de un sistema operativo hemos de diferenciar si el equipo en el que está instalado tiene una placa con BIOS o con UEFI. En el primer caso el disco de arranque será obligatoriamente MBR, pues BIOS no permite el arranque con discos GPT. (Ya hemos estudiado en temas anteriores la diferencia entre discos con particionado MBR y discos con particionado GPT).

En el caso de tener UEFI (lo habitual en los equipos modernos) habrá que tener en cuenta si la instalación se hace con la nueva BIOS en modo UEFI (lo normal) o en *Legacy Mode* (modo BIOS heredado, es decir, emulando una antigua BIOS).

Si se hace en modo UEFI, los discos han de ser necesariamente GPT. Si es en *Legacy Mode* (por ejemplo para instalar un Windows XP), los discos serán MBR o GPT (si éste es soportado por el sistema operativo).

Windows fuerza el tipo de partición en función del firmware utilizado, es decir, si Windows se inicia en el modo UEFI, es que viene instalado en un disco GPT. Si Windows se inicia en modo *BIOS legacy*, significa que sólo se puede instalar en un disco MBR. Esta es una limitación impuesta por el instalador de Windows. Así Windows soporta bien el arranque UEFI-GPT o solo arranque BIOS-MBR, pero no el arranque UEFI-MBR o BIOS-GPT.

Estas limitaciones no existen en el kernel de Linux, sino que depende del gestor de arranque utilizado. Sin embargo, esta limitación de Windows se debe tener en cuenta si el usuario quiere arrancar Windows y Linux en el mismo disco, ya que la configuración del gestor de arranque en sí depende del tipo de firmware y de la partición del disco utilizada. En el caso de arranque dual de Windows y Linux en el mismo disco, es recomendable seguir el método utilizado por Windows, ya sea el arranque UEFI-GPT o solo arranque BIOS-MBR, no los otros dos casos.

Así que por simplificar vamos a centrarnos en dos modelos opuestos. Por un lado **BIOS/MBR** y por otro **UEFI/GPT**.

3. Proceso de inicio común a Windows y Linux

A partir de este punto, cuando hagamos referencia a firmware nos referiremos indistintamente a la antigua BIOS o al actual UEFI.

Para comenzar el proceso de inicio, primero se enciende el ordenador, lo que se conoce como arranque en frío. No es igual un arranque del ordenador cuando está apagado, que un reseteo. Al resetear un equipo, parte del firmware ya está cargado en la memoria RAM y hay fases que se pueden obviar puesto que ya han sido comprobadas. Las fases que se van a describir en este punto son aquellas que son comunes independientemente del sistema operativo a arrancar.

Veamos el proceso común de arranque en distintos pasos.

PASO 1

Al arrancar el equipo en frío lo primero que ocurre es una estabilización de la tensión, finalizando el proceso cuando la señal *Power Good* aparece. Durante este tiempo el microprocesador se encuentra en estado de espera gracias a una señal de reseteo generada por el chipset. Una vez habilitado el micro, éste accede al lugar donde se encuentra el inicio del programa de ejecución del firmware.

PASO 2

A continuación, el ordenador realiza el autodiagnóstico al encender (*POST, Power On Self Test*), esto es, un chequeo de todos los dispositivos que forman el equipo. Debido a que la adaptadora de vídeo no se ha inicializado aún, cualquier error que se produzca durante esta etapa del proceso de inicio se notificará mediante una serie de pitidos llamados códigos bip (*beep*).

PASO 3

Una vez finalizado el POST, se carga el firmware en memoria para poder ser ejecutado en el procesador y se inicializa el hardware necesario para el arranque del sistema (disco, controladores del teclado, etc.).

Se habilitará el monitor, mostrando el conteo de memoria del sistema y se realizará un “inventario” de los componentes del PC, para habilitar un chequeo por cada uno de los dispositivos instalados. Es ahora cuando se inicia el proceso de Plug&Play mostrando en pantalla todos los elementos encontrados, como un resumen de los datos.

PASO 4

El firmware (BIOS o UEFI) localiza y lee los parámetros de su configuración almacenados en la memoria CMOS. Uno de estos parámetros de configuración es el orden en el cual se chequean los dispositivos para saber si cuentan con un sistema operativo para arrancar. El orden de arranque (**boot sequence**) puede hacer referencia al lector de CD/DVD, unidades USB, discos duros, discos duros externos, particiones e incluso tarjetas de red.

Dependiendo de si el firmware es BIOS o UEFI, este proceso difiere bastante.

PASO 5

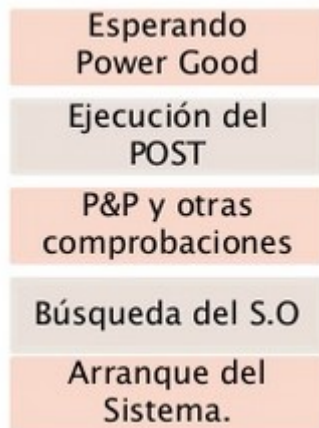
Una vez localizada la unidad con la que hay que arrancar, el firmware localiza el **cargador de inicio** del sistema operativo en dicha unidad y le transfiere el control a éste para que sea él quien continúe el proceso de carga del sistema.

El *cargador de inicio* o *boot manager* es un pequeño programa que permite cargar el sistema operativo. Su función es cargar el sistema operativo elegido por el usuario en el caso de que haya varios sistemas instalados en dicha máquina. Se le conoce también como *cargador de arranque* o *gestor de arranque*.

Existen múltiples programas de este tipo dependiendo del sistema operativo. Cada sistema operativo tiene un cargador de inicio distinto. Los más conocidos son:

- Sistema operativo Linux:
 - LILO
 - GRUB
- Sistema operativo Windows:
 - NTLDR (Windows XP).
 - BCD (Windows 7 y posteriores).

En caso de evaluar todas las unidades y no encontrar el arranque del sistema operativo, aparecerá un error en pantalla indicativo del problema del tipo “*No boot device available*” (ningún dispositivo de arranque disponible) y se solicitará la inserción de un disco de arranque. Este mensaje variará según el firmware.



De esta manera finaliza el procedimiento común de arranque. A partir de este momento, dependiendo del sistema operativo, el proceso difiere. Será el cargador de inicio (*boot manager*) particular del sistema quien tome el control.

Hemos visto que el **PASO 4** de búsqueda del cargador de arranque difiere dependiendo del tipo de firmware (BIOS o UEFI). Vamos a detallar ahora el **PASO 4** distinguiendo si el ordenador utiliza BIOS/MBR o UEFI/GPT.

Búsqueda del cargador de inicio en BIOS/MBR

Llega el momento de buscar una unidad de arranque de sistema operativo, según la secuencia definida en la propia BIOS. Por defecto esta búsqueda comenzará en la unidad de CD-ROM, pasará al disco duro y, posteriormente, al resto de las unidades del sistema. Esta secuencia de arranque (**boot sequence**) es fácilmente modificable por el usuario pudiendo conseguir el acceso al sistema operativo desde una unidad de CD-ROM, una red e, incluso, desde una memoria USB.

Cuando encuentra una unidad de disco arrancable, se accede a la ubicación física concreta donde se encuentra el sector de arranque **Master Boot Record (MBR)**: cilindro 0, cabeza 0, sector 1 (siempre es el primer sector de la unidad en cuestión).

Si este sector está preparado con las rutinas del sistema operativo oportunas, el sistema arrancará, en caso contrario se continuará la búsqueda con el resto de las unidades.

El sector de arranque es el primer sector¹ de todo disco. Cada sector contiene 512 bytes y el primer sector o sector 1 es el sector de arranque del disco. Es aquí donde buscará la BIOS para arrancar el sistema operativo. Normalmente en esos 512 bytes se encuentra un pequeño programa denominado *boot manager* que, como ya sabemos, realiza la carga del sistema operativo. Como el boot manager ocupa más de 512 bytes, en el sector MBR del disco sólo se graba una primera fase del cargador de inicio y el resto se guarda en una de las particiones del disco (dependiendo de si es Windows o Linux lo hará en una ubicación u otra). La primera fase se encarga de llamar a la segunda.

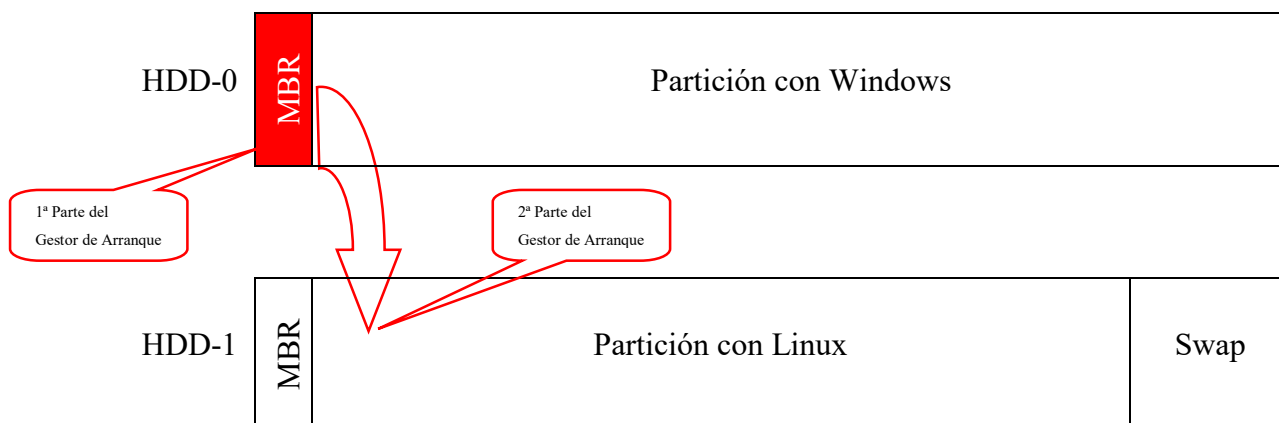
Supongamos que tenemos un ordenador con un firmware BIOS y dos discos con tabla de particiones MBR en los que tenemos instalados dos sistemas operativos (Windows y Linux). En la secuencia de arranque (boot sequence) de la BIOS indicamos que debe buscar primero un cargador en la unidad de CD-ROM, después en el disco 0 y por último en el disco 1.



Si encendemos el ordenador sin insertar un CD válido de arranque (un Live-CD o un instalador de un sistema operativo) en la unidad de CD-ROM, irá a buscar al segundo dispositivo indicado. En este caso en el disco duro 0 (HDD-0).

¹ No hay que confundir los sectores de disco con las particiones. Los sectores son divisiones físicas de los discos de tamaño fijo, de tan solo 512 bytes. Un disco duro de 1TB tiene casi 2.000 millones de sectores. El primer sector del disco es el MBR.

La BIOS examinará el primer sector del disco (MBR) en busca de un programa válido para iniciar. Si encuentra un cargador de inicio, éste se ejecutará tomando el control del procesador y liberando a la BIOS del resto del proceso. Este cargador está dividido en 2 fases, la primera almacenada en los 512 bytes del MBR y el resto (el grueso del programa) almacenado en el disco duro, que puede estar en una unidad distinta, como en el siguiente ejemplo de un cargador de arranque Linux (el Grub).



Búsqueda del cargador de inicio en UEFI/GPT

El sistema UEFI/GPT de arranque difiere bastante respecto al BIOS/MBR en la forma, pero no en el fondo. El fin es el mismo: buscar en los dispositivos arrancables un gestor de arranque y transferirle el control para cargar el sistema operativo. El procedimiento es muy diferente.

Para empezar, UEFI necesita de una partición especial en el disco duro donde almacenar el/los cargador/es de arranque. Esta partición especial UEFI debe ser una partición con formato FAT32 (aunque vale cualquier formato FAT) y se le conoce como **ESP** (*EFI System Partition*). Dentro de la tabla GPT del disco esta partición se la identifica con el GUID C12A7328-F81F-11D2-BA4B-00A0C93EC93B².

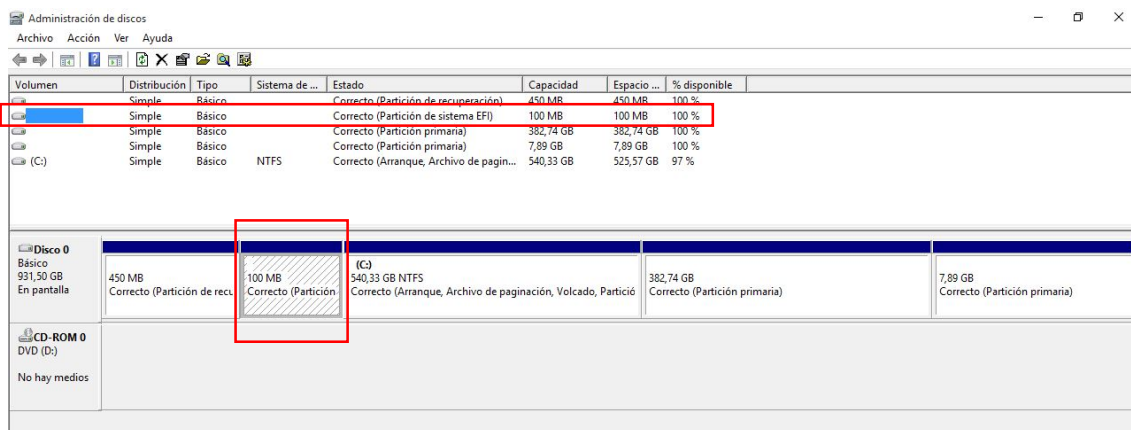
Esta partición ESP se va a utilizar para almacenar los cargadores de arranque de los distintos sistemas operativos que tengamos instalados en el sistema. Si disponemos de

² Ver el tema de Instalación de Sistemas Operativos, apartado Particionado del Disco.

varios, todos tendrán su cargador de inicio en la misma partición ESP pero en distintos directorios.

Esto es una ventaja considerable respecto al antiguo sistema BIOS en el que sólo se podía almacenar una parte pequeña de un único cargador de inicio en el sector MBR.

Lo normal es que en un disco haya sólo una partición EFI, y en ella estén todos los programas cargadores de inicio de cada sistema operativo instalado. El tamaño de esta partición no ha de ser muy grande, con 100MB suele bastar. Algunos sistemas necesitan que sea al menos de 350MB.



Volumen	Distribución	Tipo	Sistema de ...	Estado	Capacidad	Espacio ...	% disponible
	Simple	Básico		Correcto (Partición de recuperación)	450 MB	450 MB	100 %
	Simple	Básico		Correcto (Partición de sistema EFI)	100 MB	100 MB	100 %
	Simple	Básico		Correcto (Partición primaria)	382,74 GB	382,74 GB	100 %
(C:)	Simple	Básico	NTFS	Correcto (Partición primaria)	7,89 GB	7,89 GB	100 %
				Correcto (Arranque, Archivo de pagin...	540,33 GB	525,57 GB	97 %

Disco 0	Disco 0	Disco 0	Disco 0	Disco 0
Básico	450 MB	100 MB	540,33 GB NTFS	382,74 GB
931,50 GB	Correcto (Partición de recu...	Correcto (Partición)	Correcto (Arranque, Archivo de paginación, Volcado, Partición)	Correcto (Partición primaria)
En pantalla				

Estos programas almacenados en esta partición ESP son unos programas ejecutables con un formato particular definido en el estándar UEFI. Nosotros veremos estos programas en la partición con una extensión *.efi*

Puesto que esta partición especial está en formato FAT32 podemos montarla tanto en sistemas Windows como Linux para poder ver su contenido. En Linux se hace automáticamente, pues se monta dentro del directorio */boot/efi*.

En Windows podemos utilizar el mandato *mountvol* seguido de la letra de la unidad.

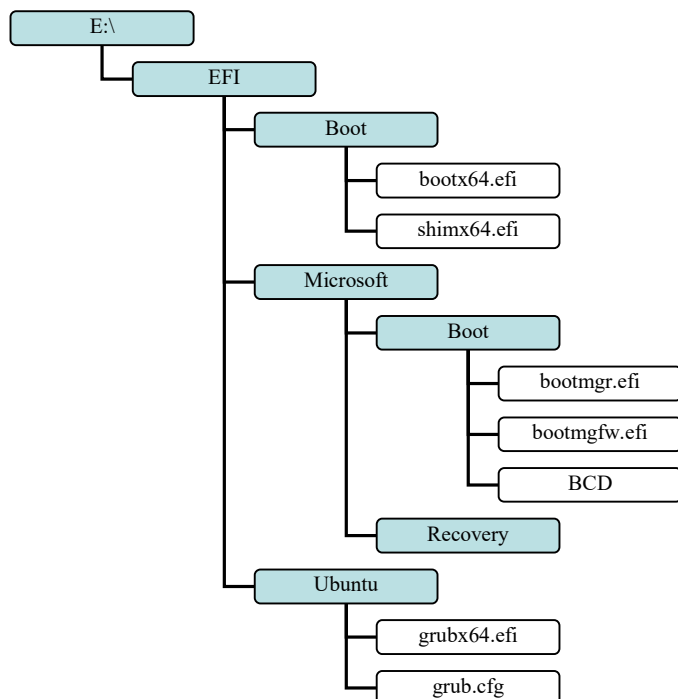
```
> mountvol E: /s
```

O bien (recomendado) ejecutar el mandato *diskpart*:

```
> diskpart
DISKPART> select disk 0
```

```
DISKPART> list volume
DISKPART> select volume 3      #seleccionamos el volumen correspondiente
                                a la partición ESP
DISKPART> assign
```

Si visualizamos el contenido de la nueva unidad nos encontraremos una estructura de directorios parecida a esta (en este ejemplo hay instalados un sistema Windows y uno Linux con sus respectivos cargadores de arranque):



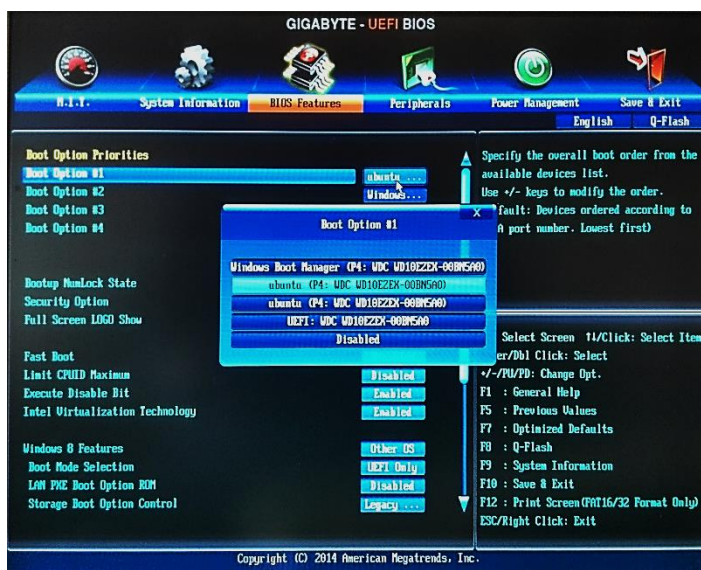
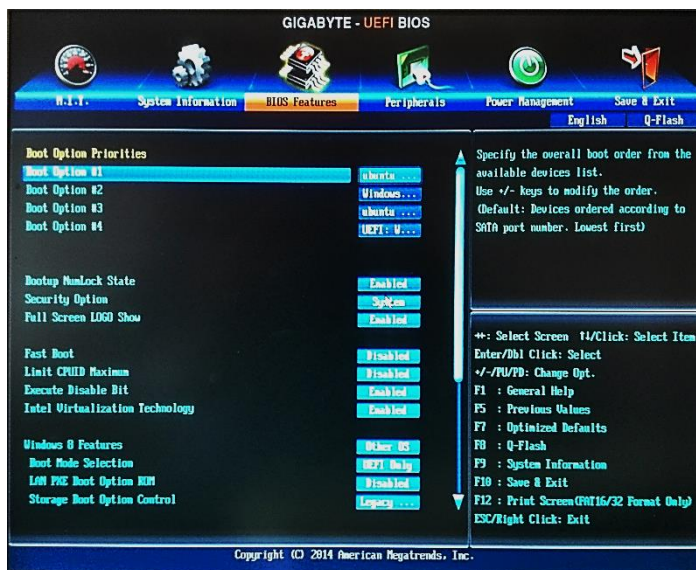
Los ficheros con extensión *.efi* son los programas ejecutables de los distintos cargadores de inicio.

El cargador de inicio de Windows será el fichero *bootmgfw.efi* y el del Ubuntu será el *grubx64.efi*.

Existe un directorio llamado *Boot* que contiene un fichero llamado *bootx64.efi* (x64 indica la arquitectura, en otras arquitecturas el fichero se llama de manera distinta) que proporciona un arranque alternativo (de reserva) para sistemas operativos no instalados de forma permanente. Está diseñado para arrancar medios de conexión en caliente, independiente del dispositivo, al igual que las imágenes de LIVE-CD y los CDs de instalación de un sistema operativo.

Para gestionar todas estas entradas, la especificación UEFI define el llamado **UEFI Boot Manager** (gestor de arranque UEFI) que podemos interpretar como un menú de arranque gestionado por el propio firmware UEFI (y no por los gestores de arranque almacenados en disco del modelo BIOS).

UEFI Boot Manager se encarga de leer todas las entradas que hay en la partición EFI (en todos los directorios) y mostrar las distintas opciones de arranque en la clásica *Boot Sequence* de la pantalla de gestión del firmware.



A diferencia de la BIOS antigua, aquí las entradas no corresponden a discos o unidades, sino a entradas generadas automáticamente por el UEFI Boot Manager a partir de los datos contenidos en la partición EFI y de los dispositivos conectados en caliente (CD/DVD, USBs, ...).

Podemos visualizar (y gestionar) estas entradas mediante programas especiales, tanto desde el entorno Linux como en Windows.

Para ello, en Linux vamos a utilizar la utilidad de la línea de comandos *efibootmgr* y en Windows podemos utilizar programas tales como *EasyUEFI* o el *Visual BCD Editor*.

El mandato *efibootmgr* de Linux nos proporciona la siguiente salida con las 3 entradas existentes en la UEFI:

```
# efibootmgr
BootCurrent: 0000
Timeout: 1 seconds
BootOrder: 0000,0002,0003
Boot0000* ubuntu
Boot0002* Windows Boot Manager
Boot0003* ubuntu
```

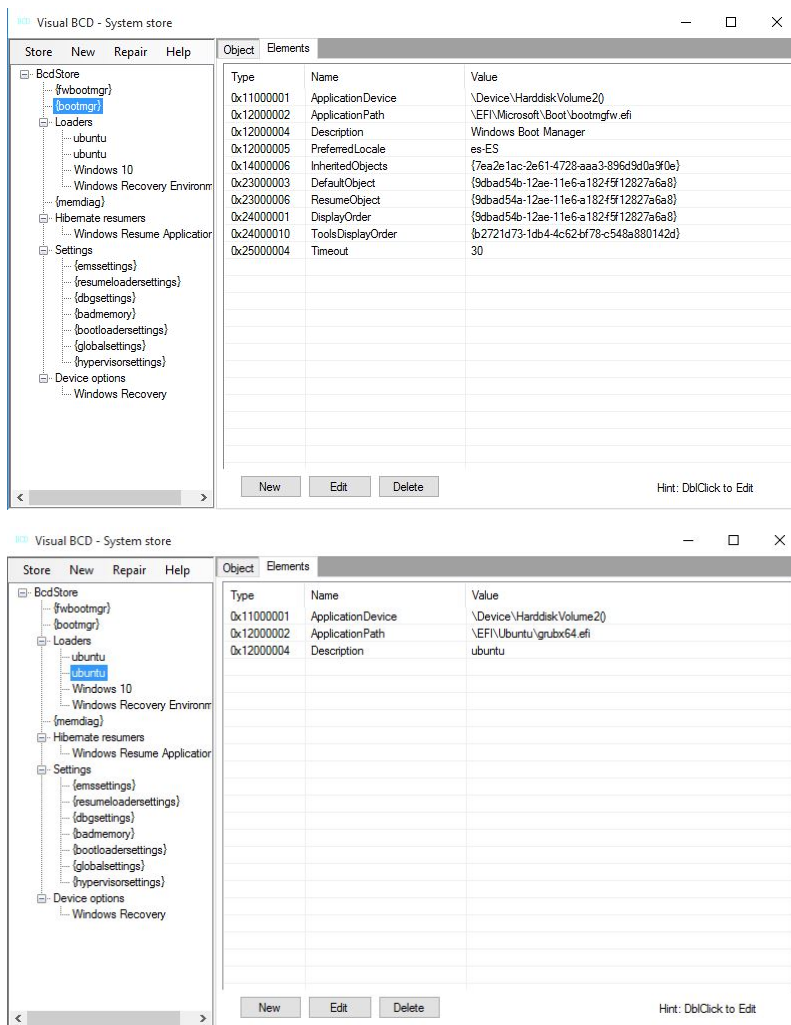
Si ampliamos la información con el parámetro *-v* la información mostrada es la siguiente:

```
# efibootmgr -v
BootCurrent: 0000
Timeout: 1 seconds
BootOrder: 0000,0002,0003
Boot0000* ubuntu HD(2,e1800,32000,6d5b5f52-7f10-4ee9-93b8-
0ec62560ba69)File(\EFI\ubuntu\shimx64.efi)
Boot0002* Windows Boot Manager HD(2,e1800,32000,6d5b5f52-7f10-
4ee9-93b8-0ec62560ba69)File(\EFI\Microsoft\Boot\bootmgfw.efi)
Boot0003* ubuntu HD(2,e1800,32000,6d5b5f52-7f10-4ee9-93b8-
0ec62560ba69)File(\EFI\Ubuntu\grubx64.efi)
```

Podemos apreciar 3 entradas en las que se apunta a los distintos ficheros *.efi* almacenados en la partición ESP.

La primera y la última corresponden a Ubuntu (hay 2 entradas distintas por el tema del *secure boot* que no vamos a ver). La segunda corresponde al cargador de Windows.

En Windows podemos gestionar esta misma información mediante el programa *Visual BCD Editor* (por ejemplo):



Para eliminar una entrada de la pantalla *boot sequence* del firmware deberemos hacerlo desde estas aplicaciones y **además borrando el directorio correspondiente de la partición ESP** (para ello habría que montarla si estamos utilizando el sistema operativo Windows).

4. Arranque de Windows

Ya hemos visto cómo el firmware (BIOS o UEFI) ha detectado dónde buscar el cargador de inicio, bien en el MBR del disco o bien la partición UEFI. El cargador de inicio de Windows se le conoce como Windows Boot Manager (*bootmgr*).

Si el firmware es BIOS, se carga la primera parte del cargador y después se llama a la segunda parte, que es el fichero oculto **bootmgr** que se encuentra en la raíz del disco de sistema (C:\).

Si el firmware es UEFI, se carga y ejecuta el **bootmgr** (ficheros *bootmgfw.efi* o *bootmgr.efi*) que, como ya hemos visto en el punto anterior, se encuentra en la partición ESP.

Independientemente del firmware utilizado, el Windows Boot Manager utiliza una base de datos para la configuración del proceso de arranque. En esta base de datos se almacena, entre otras cosas, la partición donde se encuentran los distintos sistemas operativos a arrancar. Esta base de datos se llama *Boot Configuration Data (BCD)*. Estos archivos están cifrados y protegidos, y para gestionar el BCD se deberá utilizar el programa **bcdedit** desde la línea de comandos.

Para ejecutar el *bcdedit* hay que hacerlo con permisos de administrador, es decir, habrá que ejecutar una consola como administrador.

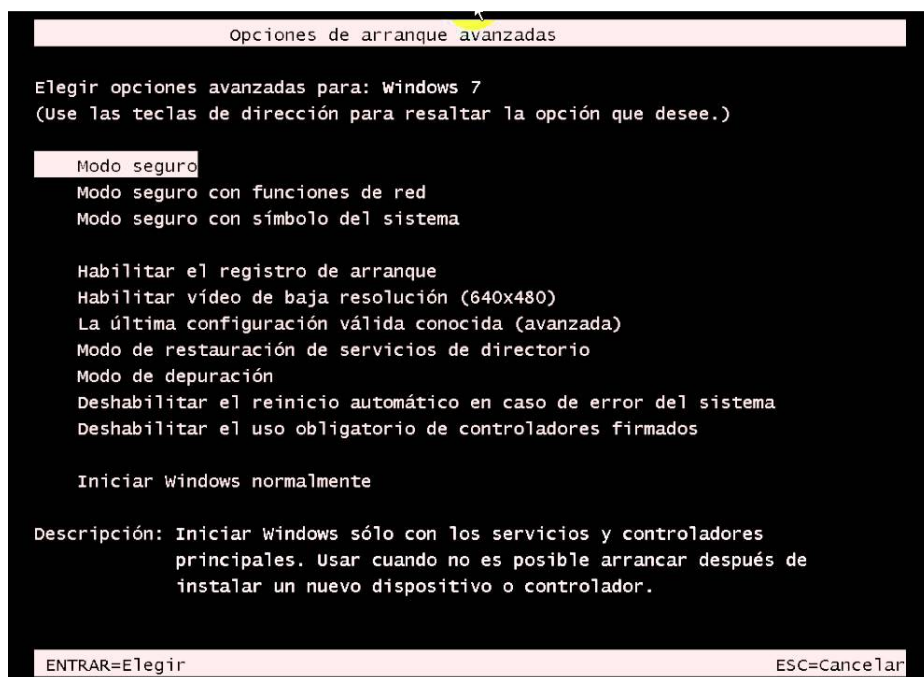
El comando *bcdedit* es un poco complicado de utilizar, lo más fácil es utilizar aplicaciones de terceros, algunas de ellas gratuitas, como el **EasyBCD** (o el *Vistart* o el *Vista Boot Pro* o incluso el *Visual BCD Editor* que hemos visto antes para gestionar las entradas UEFI) que permite de forma gráfica gestionar la configuración de arranque del sistema.

Una vez seleccionado el sistema operativo mediante el gestor de arranque a partir del menú del BCD, el *bootmgr* llama al archivo **winload.exe** (se encuentra en \Windows\system32) o al **winload.efi** en el caso de firmware UEFI (que se encuentra en

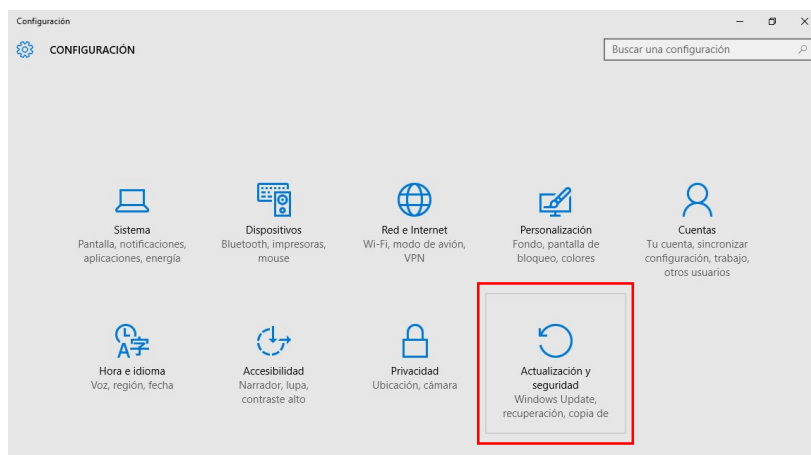
la partición ESP) que es el cargador del sistema operativo propiamente dicho y que por último da paso al archivo **ntoskrnl** (el kernel) que se encargará del resto del arranque.

Opciones de arranque de Windows

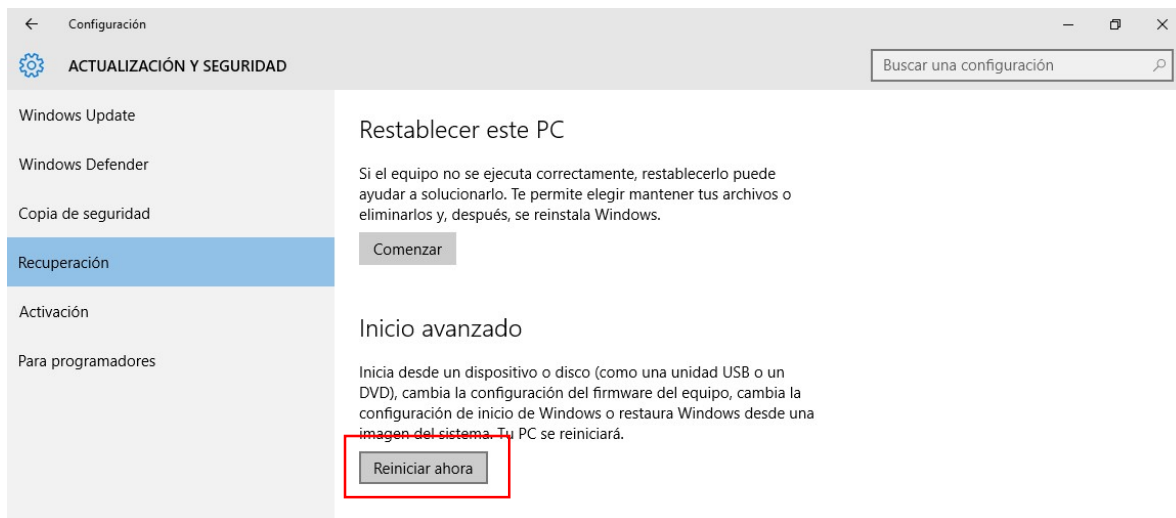
Al arrancar Windows 7, o versiones anteriores, el sistema permite al usuario poder elegir entre varios modos de arranque. Para ello, justo en el inicio del arranque, si el usuario presiona la tecla F8 el sistema muestra un menú como el siguiente (**opciones avanzadas del arranque**):



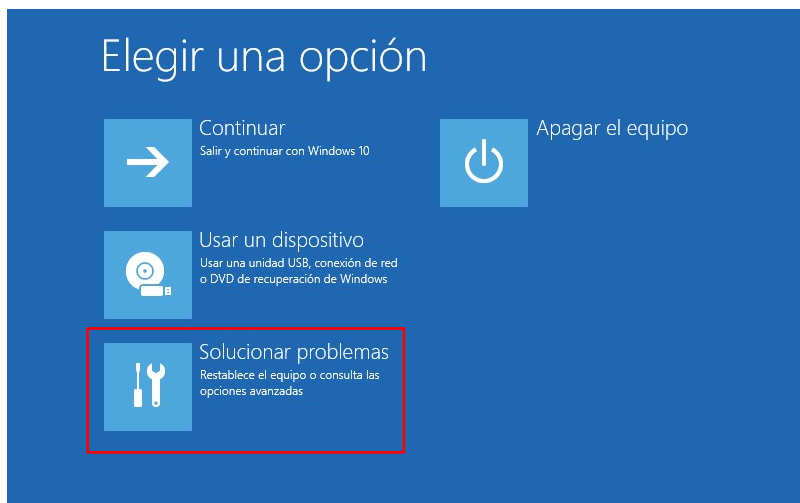
En Windows 10 es un poco más complicado acceder a este menú de opciones avanzadas. Para ello hay que dar una serie de pasos. En primer lugar, con el sistema arrancado de manera normal, accedemos al menú **configuración**:



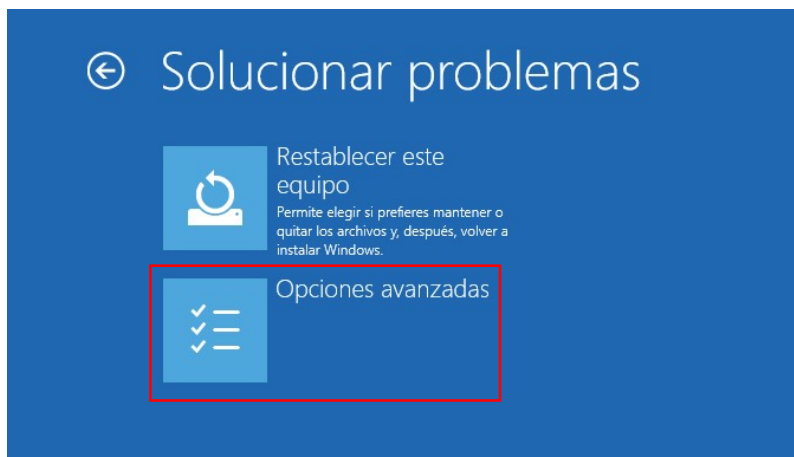
Seleccionamos la opción **Actualización y Seguridad**.



En el apartado **Recuperación** pulsamos en **Inicio avanzado** → **Reiniciar Ahora**.



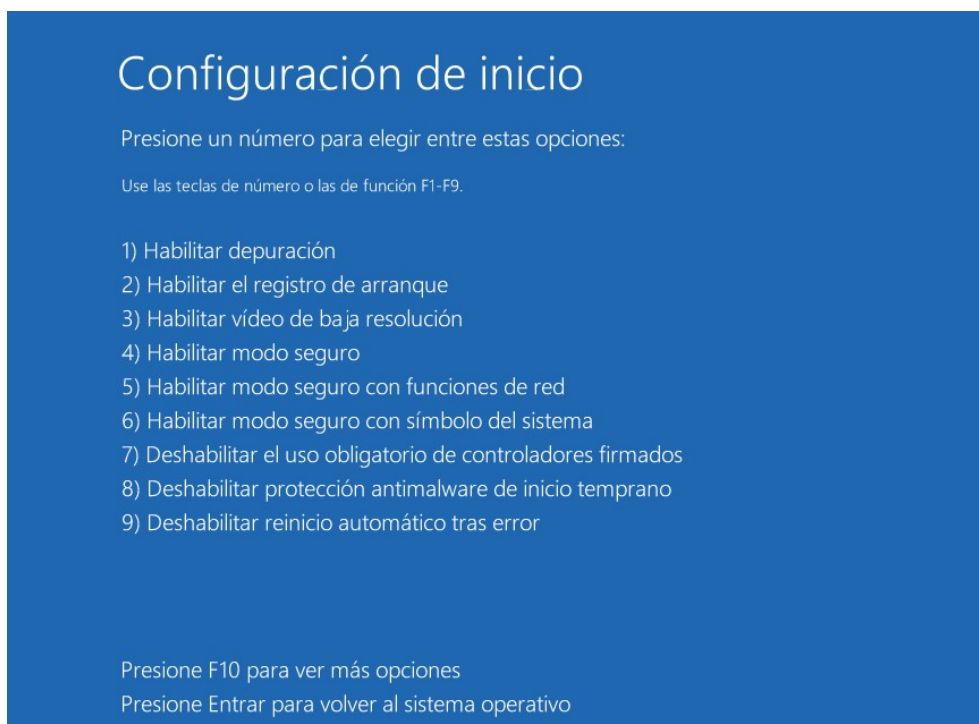
En la siguiente pantalla que nos aparece pulsamos sobre **Solucionar Problemas**.



A continuación pulsamos en **Opciones Avanzadas**.



Por último pulsamos sobre **Configuración de Inicio**, reiniciamos y ya nos sale el menú de las **opciones avanzadas del arranque**:



Este es el menú de **opciones avanzadas del arranque** de Windows. Estas opciones de inicio son muy útiles, sobre todo para solucionar problemas aparecidos en nuestro PC.

Vamos a ver cuáles son estas opciones, qué es lo que hacen y para qué nos pueden servir.

Modo seguro (SAFEBOOT_OPTION=Minimal):

Esta opción inicia Windows con un conjunto mínimo de controladores de dispositivos y servicios.

Es una opción muy apropiada tanto para reparar errores causados por una mala instalación (ya sea de software como de hardware) como para la eliminación de algunos programas o virus.

No obstante, hay algunos programas que no permiten ni su instalación ni su desinstalación en este modo de inicio.

En un arranque en Modo seguro debemos abrir sesión con un usuario con privilegios de Administrador.

Modo seguro con funciones de red (SAFEBOOT_OPTION=Network):

Es igual a la opción anterior, solo que carga los controladores necesarios para establecer las comunicaciones de red.

Al permitir conectarse a Internet es ideal para la eliminación de virus, ya que entre otras cosas nos da la posibilidad de efectuar un escaneo o desinfección online de nuestro sistema.

Modo seguro con símbolo del sistema (SAFEBOOT_OPTION=Minimal (AlternateShell)):

Es igual a Modo seguro, pero en vez de iniciar en el Explorador de Windows inicia en el editor de comandos (cmd).

Este modo de inicio es necesario cuando tenemos que ejecutar alguna utilidad que solo es accesible desde el editor de comandos, tales como algunas utilidades de disco.

Habilitar el registro de inicio:

Esta opción habilita el registro cuando el equipo se inicia con cualquiera de las opciones de modo seguro excepto La última configuración buena conocida.

El texto del registro de inicio se registra en el archivo Ntbtlog.txt en la carpeta %SystemRoot%, y nos sirve para comprobar errores en este.

Habilitar vídeo en baja resolución:

Al iniciar en Modo seguro o en Modo seguro con funciones de red el sistema carga el controladorVga.sys en lugar del controlador que estemos utilizando, con una resolución de 640 x 480 y 16 colores. Pues bien, si iniciamos en esta opción hacemos un inicio normal (es decir, cargamos todos los controladores y servicios), pero no cargamos el controlador Vga.sys, en su lugar cargamos el controlador que tengamos instalado, pero en una resolución forzada a 640 x 480 y 16 colores, que es soportada por todos los monitores existentes.

Habilitar esta opción nos puede permitir detectar y solucionar problemas relacionados con nuestra gráfica, como por ejemplo haber elegido una frecuencia de refresco que no soporta nuestro monitor.

La última configuración buena conocida (configuración más reciente que funcionó):

Esta opción inicia Windows con la última configuración buena que funcionó.

Modo de restauración de SD (sólo controladores de dominio de Windows):

Este modo sólo es válido para los controladores de dominio basados en Windows.

Realiza una reparación del servicio de directorio (SD).

Modo de depuración:

Esta opción activa el modo de depuración en Windows. La información de depuración puede enviarse por un cable serie a otro equipo que esté ejecutando un depurador.

Deshabilitar el reinicio automático si hay un error en el sistema:

Windows reinicia automáticamente el sistema si detecta un error. Esto por un lado protege al sistema, pero por otro no nos permite conocer la causa de ese error. Si elegimos esta opción deshabilitamos ese reinicio automático, lo que en la mayoría de los casos nos va a permitir ver el mensaje de error que se ha producido y solucionarlo si es posible.

Deshabilitar el uso obligatorio de controladores firmados.

Permite la carga de controladores que no están firmados por Microsoft.

Iniciar Windows normalmente:

Esta opción inicia Windows en modo normal (como si no hubiéramos pasado por esta pantalla).

Reiniciar:

Esta opción reinicia el equipo.

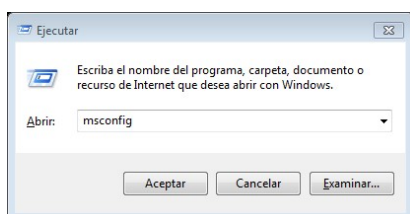
Regresar al menú de opciones del SO:

En equipos configurados para trabajar con más de un sistema operativo, esta opción vuelve al menú de inicio de arranque.

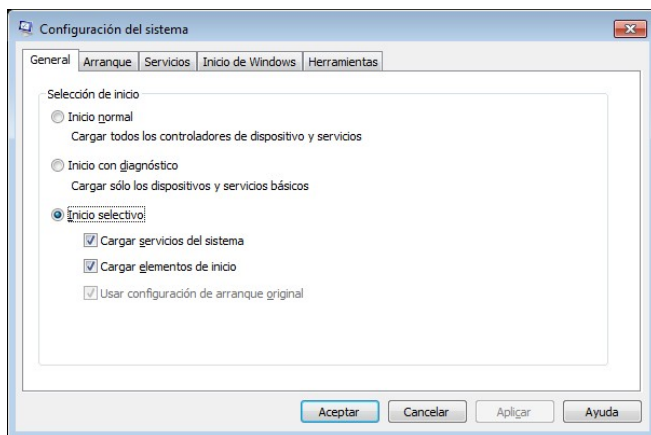
Comando MSConfig

Existe una utilidad en Windows llamada *msconfig* cuyo propósito es controlar, de una forma fácil, los programas y servicios que se arrancan en el inicio de Windows; también elegir el modo de inicio de Windows.

Para ejecutarlo es preciso pulsar el botón de Inicio - Ejecutar - MSCONFIG (e Intro)



Esta es la pantalla principal del comando:



Tiene 5 pestañas, y en cada una la posibilidad de elegir un tipo de proceso u otro elemento.

La pantalla principal es la pestaña *General*.

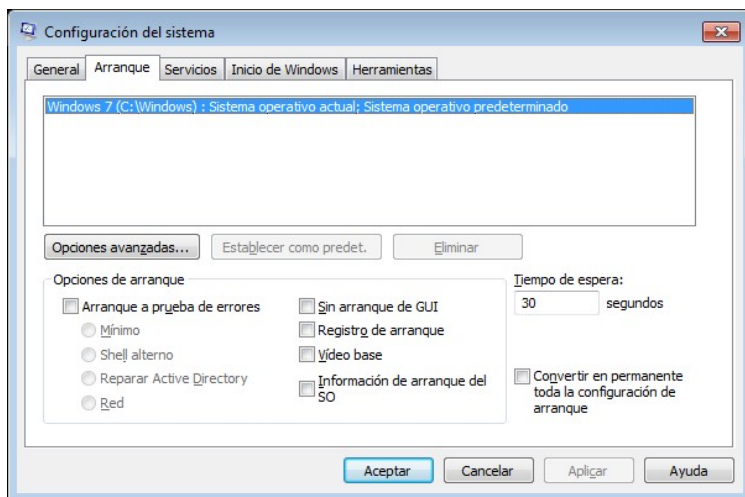
Selección de inicio

Existen 3 tipos de inicio:

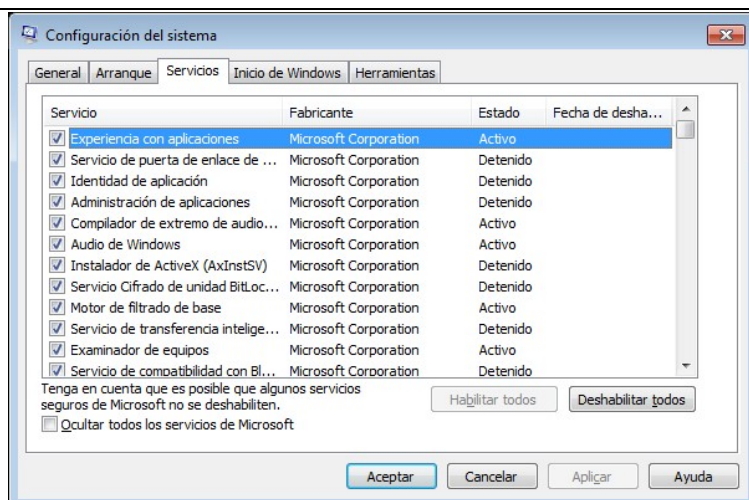
- **Normal.** Se cargan todos los elementos definidos en Inicio.
- **Con diagnósticos.** Sólo dispositivos y servicios básicos. Este se puede utilizar para iniciar Windows cuando tenemos un problema con algún módulo del inicio.
- **Selectivo.** Este es el inicio más utilizado; permite escoger exactamente los elementos que se van a iniciar en el arranque del Sistema. Para ello podemos marcar o desmarcar los epígrafes indicados más abajo (Cargar servicios del sistema, Cargar elementos de Inicio, Usar configuración de arranque original).

Pestaña de Arranque

En caso de que el gestor de arranque de Windows detecte varios sistemas operativos (normalmente sólo de la familia Microsoft Windows) podemos configurar las opciones de arranque de cada uno de ellos desde esta ventana, así como algunos parámetros del menú de inicio.



Servicios

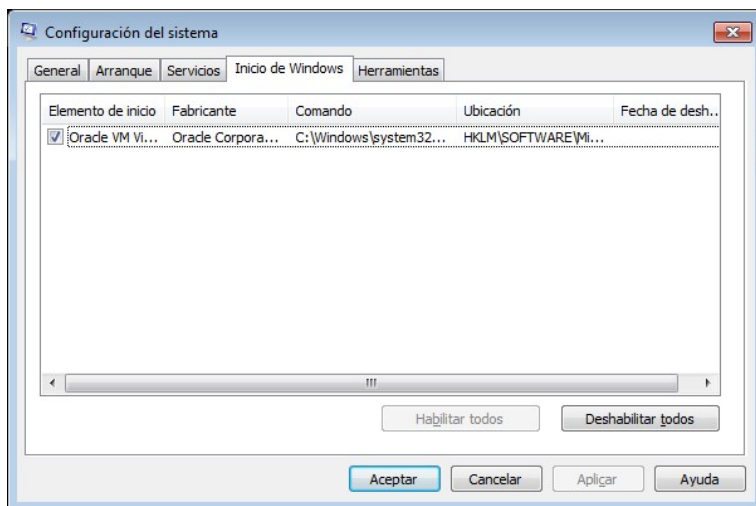


Está la relación de servicios que se inician en el arranque del Sistema; en la columna Estado nos indica si está Activo o Detenido. Si desmarcamos alguno, no se inicia.

Se puede marcar la casilla Ocultar todos los servicios de Microsoft para ver sólo los de fabricantes externos al Sistema; de esta forma vemos los que han instalado otros programas.

Inicio de Windows

Es quizás la pestaña más importante.



Permite habilitar o deshabilitar los procesos que se inician al arrancar el Sistema.

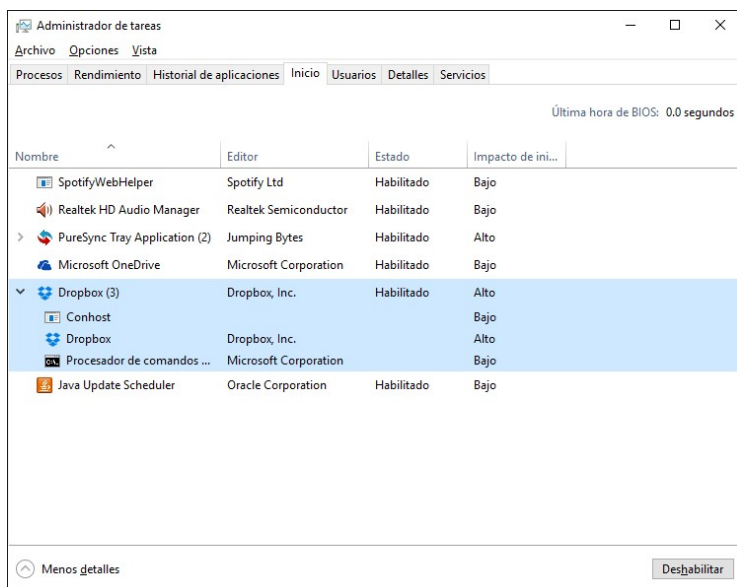
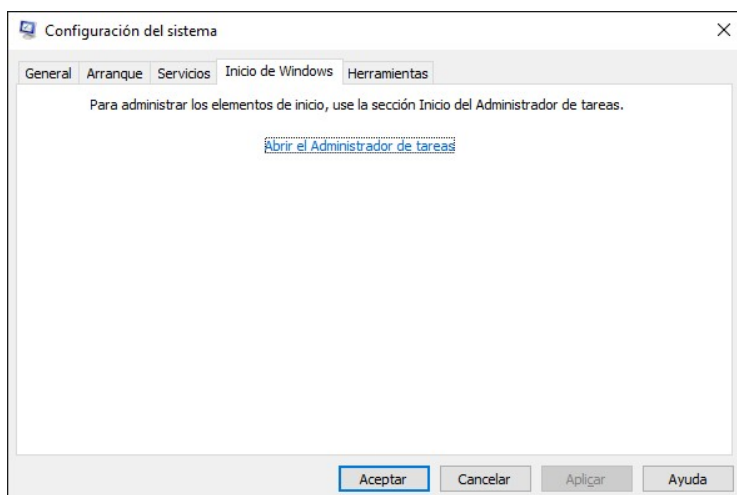
Estos programas ralentizan el inicio del Sistema, es preciso de vez en cuando hacer un repaso y deshabilitar los que no sean estrictamente necesarios. También conviene hacerlo ya que muchos virus se introducen en el Inicio y es preciso deshabilitarlos.

Si desactivamos un programa de inicio desde esta ventana, veremos en qué clave del registro de Windows está almacenada la llamada inicial. Para que desaparezca completamente de esta ventana hay que ir a dicha clave del registro y eliminar la entrada.

Es importante tener controlados qué procesos y servicios se inician automáticamente en el sistema. Puede ser que haya software malicioso o no deseado que se esté iniciando y podamos desde esta utilidad desactivar esa amenaza potencial.

Hay muchas webs que nos informan de qué son cada proceso y nos ayudan a decidir si deseamos o no que se ejecuten al inicio.

En Windows 10, esta pestaña de *Inicio de Windows* ha sido trasladada al *Administrador de Tareas*

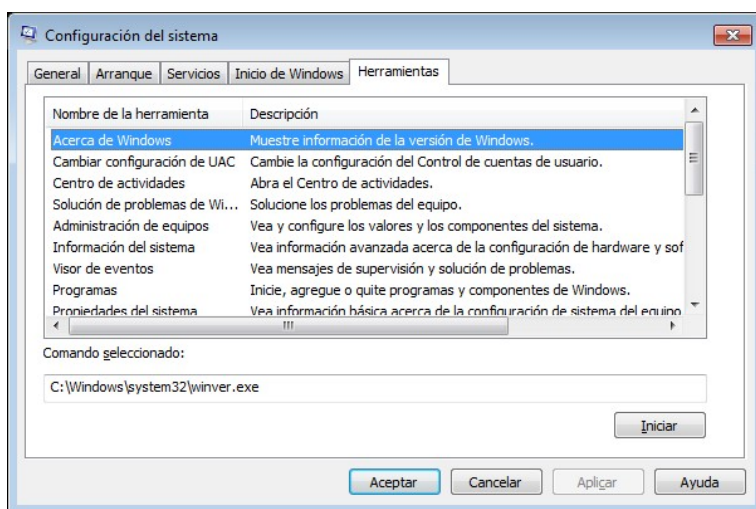


En el administrador de tareas hay una nueva pestaña llamada *Inicio* y desde ahí podemos proceder del mismo modo que en la antigua pestaña del *msconfig*.

Como curiosidad, ya nos permite mediante el botón secundario del ratón, situarnos sobre el proceso y buscar ayuda en línea sobre el mismo.

Herramientas

La última pestaña del *msconfig* es una ventana que recoge las principales herramientas del sistema agrupadas. Muy útil para la administración.



5. Arranque de Linux

Del mismo modo que hemos visto en Windows, ya sabemos cómo el firmware (BIOS o UEFI) ha detectado dónde buscar el cargador de inicio, que bien puede ser en el MBR del disco o bien en la partición UEFI.

Hemos visto que en Windows el cargador de inicio se le conoce como Windows Boot Manager (*bootmgr*). En cambio, en Linux, no hay un solo gestor de arranque, sino que dependiendo de la distribución o incluso de la personalización que haya elegido el usuario, el gestor de arranque será distinto. Hay 2 gestores de arranque principales utilizados por la mayoría de distribuciones Linux, son el **LILO** (Linux Loader) y el **GRUB**. Además, para complicarlo, existen distintas versiones de estos cargadores, diferentes unas de otras en funcionamiento.

Cuando se ejecuta el gestor de arranque, se inicia fase 1 del gestor de arranque y ésta carga la fase 2, que es la que contiene la mayor parte del código del gestor de arranque.

El hecho de dividirlo en fases corresponde a la limitación de los sectores MBR (512 bytes) en los discos que utilizan el firmware BIOS. La fase 1 se almacenaría en estos 512 bytes del MBR y la fase 2 en la partición donde se almacene el directorio */boot* de Linux (recordemos que el directorio */boot* puede estar almacenado en una partición distinta al directorio raíz del sistema Linux).

El gestor de arranque a menudo presenta al usuario un menú de opciones posibles de arranque. A continuación, si el sistema operativo seleccionado es el propio Linux, el cargador de arranque continúa con la ejecución de la función *start_kernel()*, antes de continuar por separado el proceso inactivo y planificador, y el proceso de Init (que se ejecuta en el espacio de usuario).

El planificador toma control efectivo de la gestión del sistema, y el núcleo queda dormido (inactivo).

Por último, el proceso Init ejecuta secuencias de comandos (Scripts) necesarios para configurar todos los servicios y estructuras que no sean del sistema operativo, a fin de permitir que el entorno de usuario sea creado y pueda presentarse al usuario con una pantalla de inicio de sesión.

Por ser GRUB el gestor de arranque de la distribución que estamos estudiando (Ubuntu), vamos a verlo con detalle.

Gestor de arranque GRUB

GRUB (GRand Unifier Bootloader) permite tener diferentes sistemas operativos, y diferentes versiones de ellos, en el mismo disco duro. Por ejemplo podemos tener Windows y GNU/Linux en el mismo ordenador, GRUB se cargará antes que cualquiera de éstos permitiéndonos elegir cuál iniciar.

El gestor de arranque GRUB viene preinstalado en la mayoría de las distribuciones de GNU/Linux modernas, entre ellas Debian, Ubuntu y sus derivadas. Anteriormente, el gestor de arranque más usado era LILO.

Existen 2 versiones de GRUB, la actual es la 2.0 (o simplemente Grub 2).

GRUB 1

Los archivos necesarios para el arranque de GRUB se encuentran en el directorio */boot/grub* y es aquí donde se encuentra el fichero *menu.lst* que contiene todas las opciones de arranque y que es fácilmente editable.

En este fichero se pueden añadir o cambiar las distintas opciones de arranque, así como el tiempo de espera, el SO por defecto, formato del menú, imágenes de fondo (de 640x480, de 14 colores y formato xpm.gz), contraseña cifrada, ...

GRUB 2

Grub 2.0 es el gestor de arranque predeterminado de algunas de las últimas versiones de Linux, entre ellas la Ubuntu. El funcionamiento es similar al GRUB 1, pero tiene diferencias sustanciales en el procedimiento de configuración y en los ficheros que utiliza y es mucho más flexible y potente.

El fichero */boot/grub/menu.lst* del GRUB 1 ha desaparecido y su equivalente es el */boot/grub/grub.cfg*. Este fichero se puede modificar directamente y configurar las opciones al igual que se hacía en el *menu.lst*, pero esto no es recomendable puesto que este fichero se genera de manera automática por el sistema y las modificaciones que se hagan de manera manual podrían ser borradas o sobrescritas.

Para configurar las opciones del grub2, los principales archivos a modificar son:

- Los contenidos en la carpeta */etc/grub.d/*
 - */etc/grub.d/10_linux*
 - Este archivo contiene comandos y scripts que se encargan del kernel de linux en la particion principal.
 - */etc/grub.d/30_os-prober*

- Este archivo contiene comandos y scripts que se encargan de otros sistemas operativos.
 - El archivo tiene 4 secciones. Los cambios que realicemos en una sección no afectarán al resto de las secciones.
 - Las cuatro secciones son Windows, otras particiones Linux, OSX y Hurd.
- *El archivo /etc/default/grub*
 - Tiene el grueso de las opciones configurables.
 - Los de la carpeta /usr/sbin/ (solamente los que llevan grub*, son los comandos que se ejecutan en terminal)

Un contenido típico del fichero */etc/default/grub* sería:

GRUB_DEFAULT= 0/saved, con la opción 0 hacemos que se seleccione por defecto la primera entrada, con la opción 1, la segunda, y así sucesivamente. Con la opción saved, hacemos que siempre seleccione la última entrada que se ejecutó.

GRUB_TIMEOUT=10, le ponemos el tiempo de espera hasta ejecutar la entrada que tenemos puesta como default.

GRUB_HIDDEN_TIMEOUT=0, esconde el menú de entradas del grub, si le ponemos un tiempo mas alto lo que hace es esconder el menú, pero esperarse un tiempo hasta continuar. Para hacerlo visible tenemos que comentar (escribir # al principio) la línea y que queda de la siguiente manera

```
#GRUB_HIDDEN_TIMEOUT=0
```

GRUB_HIDDEN_MENU_QUIET=true/false, si está a "true" oculta la cuenta atrás, mientras que si está a "false" muestra la cuenta atrás.

GRUB_DISTRIBUTOR=lsb_release -i -s 2> /dev/null || echo Debian , determina el nombre de la entrada del menú.

GRUB_CMDLINE_LINUX="opciones", similar al altopoptions del antiguo grub.

GRUB_CMDLINE_LINUX_DEFAULT="quiet splash" quiet sirve para agrupar las entradas iguales, mientras que splash sirve para que nos muestre la imagen de carga en vez de los mensajes del kernel

#GRUB_GFXMODE=640x480 descomentar (escribir # al principio) la línea activa el grub gráfico. Mediante el comando vbeinfo escrito en la línea de comandos del grub vemos las posibilidades.

#GRUB_DISABLE_LINUX_RECOVERY="true" Descomenta esta línea para que no aparezca la opción de recovery mode en el menú

Cualquier cambio que hagamos en este fichero o en los otros, para que tengan efecto y se actualice el fichero */boot/grub/grub.cfg*, hay que ejecutar el mandato:

```
$ sudo update-grub2
```

Al ejecutar este mandato se actualiza y sobrescribe el `/boot/grub/grub.cfg`

Para **añadir un nuevo sistema operativo** a la lista del grub basta con tenerlo instalado y hacer que el mandato `update-grub2` lo reconozca.

La información de los sistemas operativos se agrupa en distintos archivos de la carpeta `/etc/grub.d/` **Para eliminar un sistema operativo**, simplemente eliminamos el archivo o le cambiamos el nombre (por ejemplo introduciendo un asterisco antes del nombre). Después ejecutamos `update-grub2`

Para que **no aparezca memtest** en el menú, le quitamos los permisos de ejecución con el siguiente comando:

```
$ sudo chmod -x /etc/grub.d/20_memtest86+
```

Si escribimos `+x` en lugar de `-x` volvemos a darle permisos de ejecución y nos volverá a aparecer. En cualquier caso, después hay que ejecutar `update-grub2`

Para **eliminar un kernel** de la lista, podemos mover los archivos de ese kernel de la carpeta `/boot` a una carpeta creada por nosotros. Primero debemos crear la carpeta:

```
$ sudo mkdir /boot/kernels
```

A continuación movemos la imagen del kernel (cambiar `2.6.31-14` del ejemplo por el kernel que queramos eliminar del menú):

```
$ sudo mv /boot/vmlinuz-2.6.31-14-generic-pae /boot/kernels/
```

Y también podemos mover la imagen de que se utiliza para entrar en el modo recuperación (cambiar `2.6.31-14` del ejemplo por el kernel que queramos eliminar del menú):

```
$ sudo mv /boot/initrd.img-2.6.31-14-generic-pae /boot/kernels/
```

Después actualizamos `grub.cfg` con el mandato ya conocido:

```
$ sudo update-grub2
```

6. Sistemas arrancables desde dispositivos extraíbles

Hasta ahora hemos estado hablando en todo momento de cómo se inician los sistemas operativos que están instalados en un disco duro.

Hemos visto en las opciones de arranque del firmware que se podía arrancar también desde otros dispositivos, no sólo desde discos duros. Esto es especialmente importante a la hora de instalar sistemas operativos o ejecutar software mediante LIVE-CDs. El firmware nos permite elegir entre varias fuentes de arranque:

- Discos duros. La opción más habitual. Contiene sistemas operativos. Estudiada en los puntos anteriores.
- Lectores CD/DVD:
 - Con instaladores de sistemas operativos (los DVDs que contienen los sistemas a instalar).
 - Live-CD o Live-DVD, que son discos de arranque que permiten ejecutar un sistema operativo o un software sin necesidad de instalarlo ni de arrancar con un sistema operativo ya instalado en nuestro disco duro. Esta opción es habitual encontrarla en los CDs de instalación de Linux (opción *probar* sin instalar). Se suele utilizar cuando deseamos solucionar problemas con los sistemas instalados, gestionar particiones de los discos duros o cualquier otra acción relacionada con la seguridad.
- Discos duros externos. El funcionamiento es similar a un disco duro normal. La única diferencia es la interfaz de conexión (USB, eSATA).
- Dispositivos extraíbles. Tales como memorias flash conectadas por USB. Es una opción muy útil para llevar un sistema operativo en un USB arrancable en modo “live” o una imagen arrancable de un instalador de un sistema operativo (por ejemplo Windows).
- Arranque por red. Permite arrancar un ordenador desde una red en lugar de un disco. Para ello se utiliza el entorno PXE. El ordenador intentará conectarse a un servidor que le proporcionará por red el programa de arranque o NBP (network bootstrap program). Este programa se comportará como un gestor de arranque. Esta opción es muy útil para entornos de *clientes ligeros* conectado a servidores de terminales.

Para crear dispositivos USB arrancables podemos utilizar varias aplicaciones. Dependiendo del sistema operativo que estemos utilizando estas aplicaciones pueden venir ya incorporadas como utilidades del sistema.

Veamos algunos ejemplos.

Crear un dispositivo USB de instalación de Windows desde Windows

Se puede crear un disco de instalación de Windows en un pendrive. Esto nos permitirá instalar este sistema operativo en equipos que no disponen de lectores CD/DVD o que lo tienen estropeado.

Existen dos alternativas. La primera es tan sencilla como grabar desde una imagen ISO a un pendrive utilizando programas específicos de grabación (Nero, ImgBurn, ...).

La segunda alternativa es mediante un programa oficial de Microsoft que nos facilita la tarea que se puede descargar desde la misma web de Microsoft, el **Windows USB/DVD Download Tool**.

Es muy trivial utilizarlo, hay que seleccionar la imagen ISO del Windows a instalar e indicarle el dispositivo USB donde queremos que la copie.

Crear un dispositivo USB de instalación de Ubuntu (Live CD) desde Windows

Si el sistema que deseamos copiar al USB no es un Windows, no nos vale la aplicación de Microsoft vista en el epígrafe anterior.

En su lugar podemos descargar el programa *Windows Universal USB Installer (WUI)* desde la web.

Previamente el pendrive ha de estar formateado (FAT32 o NTFS). Se le indica la imagen del sistema Linux a instalar y de manera opcional, se puede marcar la opción de

persistencia para que almacene en el dispositivo los cambios, documentos y demás realizados durante la ejecución en modo *live*.

Crear un dispositivo USB de instalación de Ubuntu (Live CD) desde Ubuntu

En este caso, Ubuntu ya incorpora una utilidad para realizar este proceso llamada **Creador de disco de arranque**.

Funciona igual que las anteriores.

En todas estas opciones vistas, el gestor de arranque del sistema operativo se almacena en el propio dispositivo USB, sin particiones.

Material elaborado a partir de:

- Comunidad de Ubuntu en español <http://www.ubuntu-es.org/>
- AdamW on Linux and more <https://www.happyassassin.net/2014/01/25/uefi-boot-how-does-that-actually-work-then/>
- Wiki ArchiLinux <https://www.archlinux.org/>
- Microsoft Technet
- Wikipedia