



**PROYECTO INTEGRADO  
2012-2013**

**Autor: José Felipe González Postigo**

C.F.G.S. Administración de Sistemas Informáticos en Red

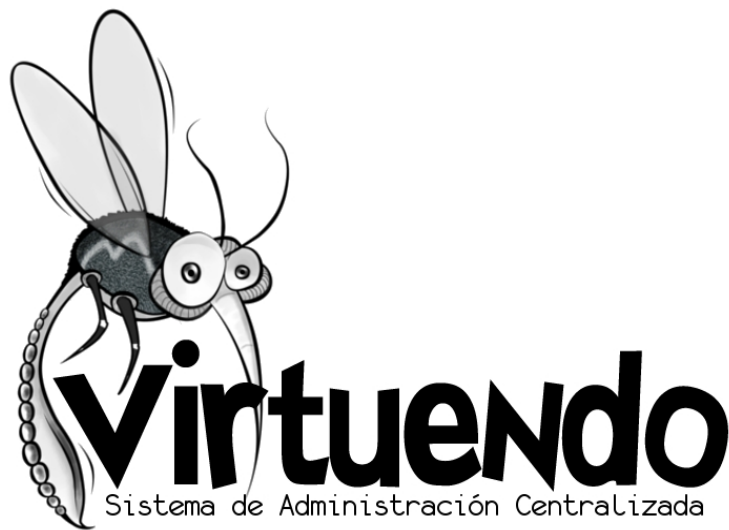
I.E.S. Miguel Romero Esteo

Tutor docente del proyecto: Alejandro de la Torre

# Índice de Contenidos:

<b>1. Presentación del proyecto.....</b>	<b>Pág 3</b>
<b>1.1 Descripción del objetivo.....</b>	<b>Pág 4</b>
<b>2. Análisis de las soluciones disponibles.....</b>	<b>Pág 5</b>
<b>2.1 Arranque en Red.....</b>	<b>Pág 5</b>
<b>2.2 Esquema del protocolo PXE.....</b>	<b>Pág 7</b>
<b>2.3 Métodos de Instalación de Sistemas Operativos.....</b>	<b>Pág 8</b>
<b>2.4 Elección del Sistema Operativo del servidor.....</b>	<b>Pág 9</b>
<b>3. Topología de la Red.....</b>	<b>Pág 10</b>
<b>4. Configuración del servidor.....</b>	<b>Pág 11</b>
<b>4.1 Requisitos de Hardware.....</b>	<b>Pág 11</b>
<b>4.2 Instalación del Sistema Operativo.....</b>	<b>Pág 11</b>
<b>4.3 Configuración inicial de Debian 6.....</b>	<b>Pág 12</b>
<b>4.4 Instalación del servidor DHCP.....</b>	<b>Pág 13</b>
<b>4.5 Instalación del servidor TFTP.....</b>	<b>Pág 14</b>
<b>4.6 Instalación del servidor de Imágenes PARTIMAGE.....</b>	<b>Pág 14</b>
<b>5. El menú de arranque PXELINUX.....</b>	<b>Pág 18</b>
<b>6. Las imágenes INITRD.....</b>	<b>Pág 21</b>
<b>6.1 Creación de imágenes INITRD.....</b>	<b>Pág 21</b>
<b>6.2 Administración de las imágenes INITRD.....</b>	<b>Pág 24</b>
<b>7. Opciones del menú de arranque.....</b>	<b>Pág 25</b>
<b>7.1 Opciones con llamadas a los programas desarrollados.....</b>	<b>Pág 25</b>
<b>7.2 Opciones de arranque de sistemas operativos.....</b>	<b>Pág 26</b>
<b>7.3 Otras opciones.....</b>	<b>Pág 27</b>
<b>8. Arranque en red de SystemRescueCd.....</b>	<b>Pág 28</b>
<b>9. Esquema de funcionamiento general del sistema.....</b>	<b>Pág 30</b>
<b>10. Creación de imágenes de disco.....</b>	<b>Pág 31</b>
<b>10.1 Particionado del disco.....</b>	<b>Pág 31</b>
<b>10.2 Creación de imagen maestra de Windows XP SP3.....</b>	<b>Pág 32</b>
<b>10.3 Creación de imagen maestra de Ubuntu 13.04.....</b>	<b>Pág 35</b>
<b>11. Habilitar arranque en red.....</b>	<b>Pág 37</b>
<b>12. Marketing.....</b>	<b>Pág 38</b>
<b>13. Bibliografía y Recursos.....</b>	<b>Pág 39</b>

# 1. Presentación del proyecto



El presente proyecto, surge de la necesidad de solucionar una serie de problemas derivados del uso continuado de los equipos informáticos en las aulas del ciclo de A.S.I.R. del I.E.S Miguel Romero Esteo, aunque su implantación es viable en cualquier aula que disponga de una red de ordenadores.

En un ambiente de trabajo, en el que los ordenadores del aula se utilizan para realizar todo tipo de prácticas de clase, a menudo nos encontramos con que las máquinas no disponen de un sistema operativo instalado para su uso inmediato.

La instalación de cualquier sistema operativo, supone una pérdida de tiempo considerable, además de su posterior configuración y la instalación del software necesario para empezar a trabajar.

Con este proyecto se pretende resolver este problema, además de aportar otras funcionalidades añadidas.

*“Virtuendo es un sistema capaz de proporcionar al usuario que se sienta delante de un ordenador, un mecanismo de instalación de sistemas operativos ágil, rápido y simple.”*

## 1.1 Descripción del objetivo

Lo que se pretende conseguir con este proyecto es un Sistema de Administración Centralizada.

Habrà uno o varios servidores que, mediante un conjunto de tecnologías, proporcionen a las máquinas cliente un menú de arranque en red.

Dicho menú debe aparecer en pantalla siempre que un alumno o usuario encienda un ordenador, de manera que el paso por el sistema sea “cuasi” obligatorio.

El servidor tendrá almacenados los datos del menú, los programas y todo lo necesario para dotar a los clientes de instalaciones limpias de sistemas operativos.

Mediante el menú de arranque en red, dispondremos de una serie de opciones que nos permitirán configurar la máquina, o simplemente arrancar un sistema operativo instalado en la misma.

Seleccionando las opciones del menú se lanzarán una serie de programas que harán los trabajos de instalación y configuración.

Los objetivos principales que deben cubrir estos programas son:

- Particionar de forma automática el disco duro de la máquina.
- Instalar un sistema operativo de forma desatendida y con la máxima brevedad posible.
- Arrancar un sistema operativo directamente sin pasar por otros gestores de arranque.
- Redirigir el flujo del arranque hacia el MBR del disco duro principal, para permitir el arranque de otros sistemas instalados en la máquina.
- Permitir la subida de nuevos sistemas operativos al servidor.
- Arrancar un sistema operativo de rescate (live CD) proporcionado por el servidor, eliminando la necesidad de introducir el CD en la unidad lectora.

## 2. Análisis de las soluciones disponibles

Para lograr los objetivos, he realizado un trabajo de investigación sobre las tecnologías disponibles, siempre buscando software Open Source.

De esta manera, cualquier persona interesada puede implementar el proyecto en su empresa o institución sin coste alguno, siguiendo los pasos que se detallan en este documento, que pretende ser, además de un proyecto integrado, una guía de instalación y posterior administración del sistema.

### 2.1 Arranque en Red

Para empezar, necesitamos un procedimiento capaz de arrancar los clientes a través de la red. En este sentido nos encontramos con una serie de protocolos estandarizados que cumplen con el objetivo propuesto:

**PXE, Preboot eXecution Environment** (Entorno de ejecución de prearranque), es un entorno para arrancar e instalar el sistema operativo en ordenadores a través de una red, de manera independiente de los dispositivos de almacenamiento de datos disponibles (como discos duros) o de los sistemas operativos instalados.

Para su funcionamiento, es necesario que la tarjeta de red del cliente tenga instalado un firmware especial.

Hoy en día, prácticamente todas las tarjetas de red poseen esta capacidad.

El firmware del cliente trata de encontrar un servicio de redirección PXE en la red para recabar información sobre los servidores de arranque PXE disponibles. Tras analizar la respuesta, el firmware solicitará al servidor de arranque apropiado el *file path* de un *network bootstrap program* (NBP), lo descargará en la memoria RAM del ordenador mediante TFTP, probablemente lo verificará, y finalmente lo ejecutará.

**BOOTP (Bootstrap Protocol)** es un protocolo de red UDP, utilizado por los clientes de red para obtener su dirección IP automáticamente. Normalmente se realiza en el proceso de arranque de los ordenadores o del sistema operativo.

Este protocolo permite a los ordenadores sin disco obtener una dirección IP antes de cargar un sistema operativo avanzado. Históricamente ha sido utilizado por las estaciones de trabajo sin disco basadas en UNIX (las cuales también obtenían la localización de su imagen de arranque mediante este protocolo) y también por empresas para introducir una instalación preconfigurada de Windows en PC recién comprados (típicamente en un entorno de red Windows NT).

El proyecto **PXELINUX** hace uso de estos protocolos para conseguir el arranque de sistemas operativos mediante la tarjeta de red.

Es una derivación de **SYSLINUX**, un proyecto que abarca un conjunto de gestores de arranque ligeros. Entre sus derivados más conocidos está **ISOLINUX**, que nos permite crear CD's de arranque con un menú de opciones personalizable.

PXELINUX se apoya en varios servicios para su funcionamiento:

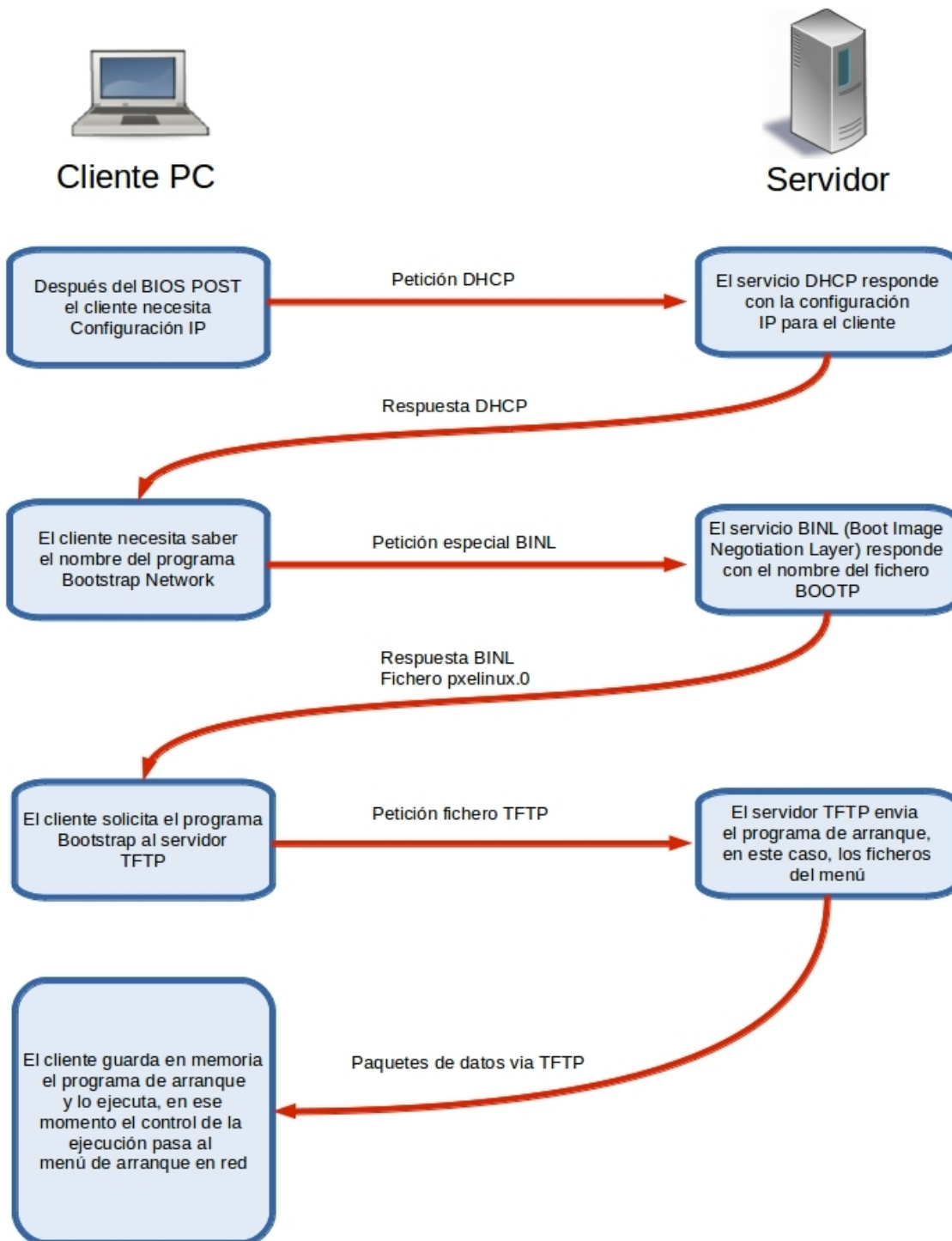
**DHCP** (sigla en inglés de *Dynamic Host Configuration Protocol*, en español «protocolo de configuración dinámica de *host*») es un protocolo de red que permite a los clientes de una red IP obtener sus parámetros de configuración automáticamente.

**TFTP** son las siglas de **Trivial file transfer Protocol** (Protocolo de transferencia de archivos trivial).

Es un protocolo de transferencia muy simple semejante a una versión básica de FTP. TFTP a menudo se utiliza para transferir pequeños archivos entre ordenadores en una red, como cuando un terminal X Window o cualquier otro cliente ligero arranca desde un servidor de red.

El servidor TFTP es el encargado de transferir al host cliente el menú de arranque en red.

## 2.2 Esquema del protocolo PXE



## 2.3 Métodos de Instalación de Sistemas Operativos

Mediante PXELINUX es posible iniciar la instalación de un sistema operativo alojado en el servidor, pero sería similar a la que haríamos introduciendo un CD de instalación.

Esta solución no es viable, pues buscamos un tipo de instalación desatendida, además necesitamos que el sistema esté listo para usar recién instalado, incluyendo drivers, configuración y software adicional.

Existen muchos métodos para realizar instalaciones desatendidas, como **WAIK** (Windows Automated Installation Kit) de Microsoft, **Kickstart** de Red-Hat o **Unattended**, por citar algunos. Pero estos métodos quedan descartados para este proyecto por algunas o varias de las siguientes razones:

- ✗ Sólo permiten instalar un Sistema Operativo concreto.
- ✗ No son Open Source.
- ✗ No son compatibles con el arranque en red.
- ✗ Necesitan intervención del usuario para las instalaciones.
- ✗ Necesitan de un entorno gráfico para funcionar.

La solución a estos problemas la encontramos recurriendo a un software capaz de respaldar y restaurar imágenes de disco completas. Con éste método podemos hacer una instalación limpia de cualquier sistema operativo, instalar los drivers y todo el software necesario y crear una imagen del disco o de una partición, que se puede almacenar en un servidor.

Posteriormente esa imagen se puede restaurar en cualquier ordenador cliente a través de la red.

Hay varios programas que resuelven esta tarea, por ejemplo **Norton Ghost**, **Acronis** o **Clonezilla**, pero sólo he encontrado uno que cumpla los requisitos deseados.

Para este proyecto he elegido utilizar **Partimage** por las siguientes razones:

- ✓ Arquitectura cliente-servidor.
- ✓ El servidor consume muy pocos recursos.
- ✓ El cliente se puede usar en línea de comandos y puede lanzarse sin intervención del usuario.
- ✓ Incluye sistema de autenticación para escribir en el servidor.
- ✓ Soporta sistemas de ficheros NTFS y ext2/3/4.
- ✓ Comprime las imágenes.
- ✓ Sólo copia la parte utilizada del disco, reduciendo el tamaño de las imágenes al mínimo.

Ya tenemos un método para mostrar un menú de arranque en red (pxelinux) y un servidor de imágenes de disco (Partimage) esperando peticiones, pero ahora necesitamos una forma de ejecutar el cliente de partimage en el host, así como los programas necesarios para el particionado del disco, etc.



La solución la encontramos en las imágenes **Initramfs** (versión mejorada de Initrd).

El **disco RAM inicial** o **Initrd**, por sus siglas en inglés es un sistema de archivos temporal usado por el núcleo Linux durante el inicio del sistema. Normalmente, el núcleo lo utiliza para hacer los arreglos necesarios antes de que el sistema de archivos raíz pueda ser montado.

Initrd.gz es un archivo comprimido que contiene un sistema de ficheros, los módulos necesarios para el funcionamiento del kernel y una serie de scripts de inicio.

Así como para cargar el código necesario para preparar el arranque del sistema, un disco RAM (ya sea initrd o initramfs) puede ser útil como disco de rescate, para utilizar en actualizaciones de seguridad, copias de seguridad de archivos, realizar análisis forenses, depurar problemas de hardware, u obviar un disco duro, por ejemplo para arrancar imágenes del sistema operativo por red.

Es decir, que disponemos de un micro sistema operativo con unas funciones básicas cargado en la memoria RAM del ordenador durante el arranque, por lo tanto si conseguimos modificar el comportamiento de los scripts de inicio, podremos lanzar comandos en la máquina para configurarla.

Posteriormente se describirá en detalle como se consigue dicho comportamiento.

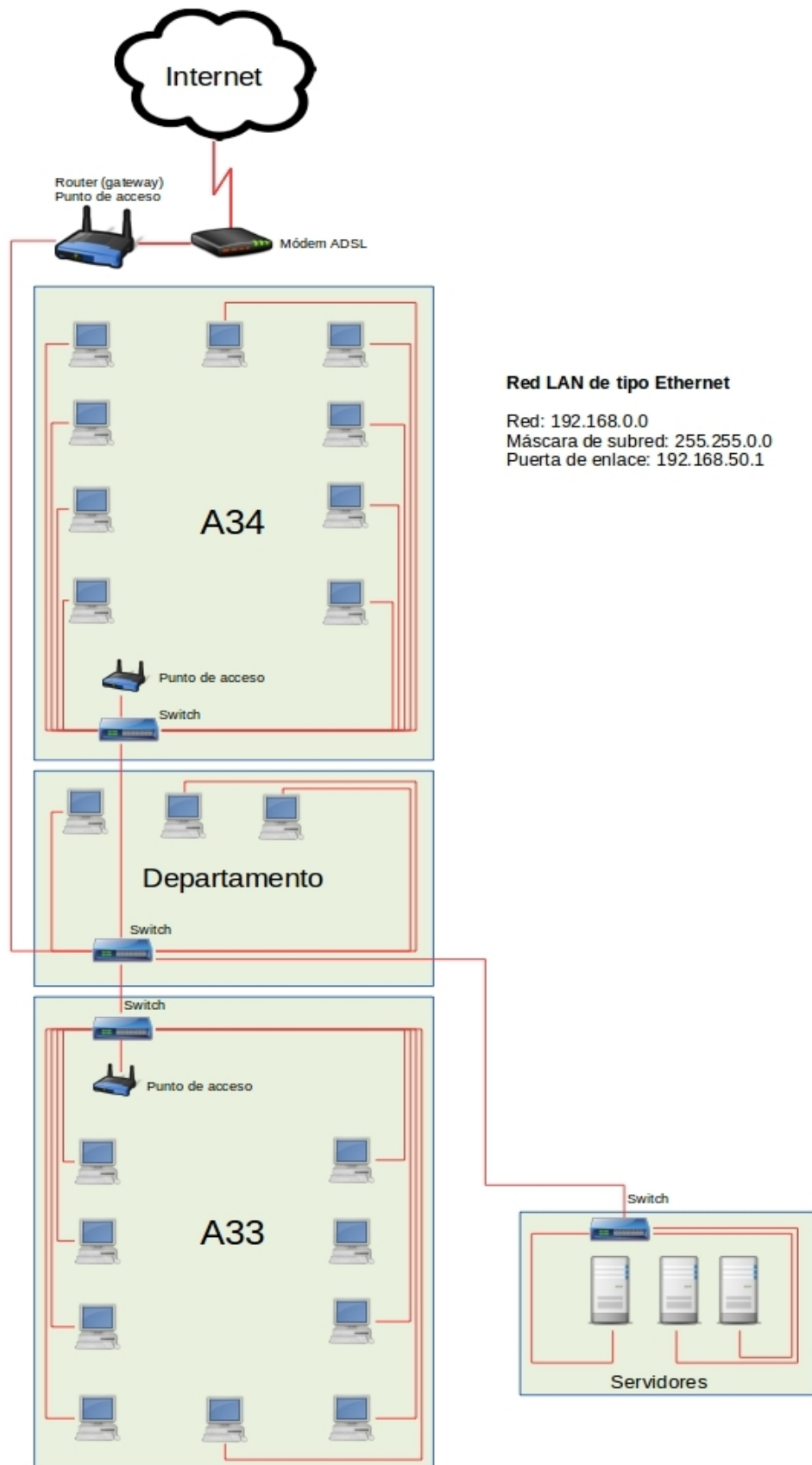
## 2.4 Elección del Sistema Operativo del servidor.

Para instalar el servidor he elegido un sistema operativo GNU/Linux. La razón es simple, todo el software en el que se basa el proyecto corre sobre este sistema. Esto no es casualidad, pues el software libre es mucho más versátil que el software privativo en este sentido, nos permite personalizar, modificar y desarrollar sobre el software existente, lo que hubiera sido imposible en otras arquitecturas disponibles en el mercado.

**Debian** es un producto maduro, orientado a servidores y con una gran comunidad de desarrolladores dando soporte continuo.

Gracias a su extenso repositorio de software, la instalación y configuración de servicios se hace más simple, por lo que se convierte en la opción perfecta para este proyecto, además es un sistema operativo de sobra conocido en el I.E.S Miguel Romero Esteo.

### 3. Topología de la Red



## 4. Configuración del servidor

### 4.1 Requisitos de Hardware

Para la instalación de los servicios necesarios para este sistema, no se precisa un servidor especializado con gran capacidad de procesamiento. En realidad, el límite de velocidad de las transferencias vendrá marcado por la infraestructura de red de que dispongamos, sobre todo cuando hagamos descargas recursivas sobre varios clientes, donde tendremos un “*cuello de botella*” en la red.

Una vez esté el sistema en producción, el servidor actuará como un repositorio de datos, por lo que es en la velocidad de lectura del disco donde deberíamos poner más atención.

Para aumentar este rendimiento, podríamos instalar varios discos montados en RAID 0, por ejemplo, aunque esto es opcional.

A priori, cualquier equipo de escritorio moderno podría hacer el trabajo para una red de unos 30 ordenadores sin problemas.

Sobre los requisitos de instalación de Debian 6, sólo se necesitan 256 Mbytes de memoria RAM, aunque es recomendable al menos 2 Gb.

En cuanto al procesador, sería recomendable que tuviera dos o más núcleos.

Necesitaremos calcular el suficiente espacio en disco para almacenar las imágenes comprimidas, que pueden ocupar de 1 a 2 Gb cada una dependiendo del S.O.

Más adelante se describe este proceso en detalle.

### 4.2 Instalación del Sistema Operativo

Para instalar Debian 6, nos dirigimos a la web oficial de Debian:

<http://www.debian.org>

Descargamos la última versión estable disponible del producto.

Tendremos que grabar un CD con la imagen descargada y colocarlo en la unidad lectora del servidor para iniciar la instalación.

Durante la instalación seleccionaremos no instalar el entorno gráfico pues no nos sirve para nada en este servidor.

Como medida de seguridad, podemos instalar el sistema operativo en una partición, y reservar otra para ubicar el directorio que servirá de repositorio para el servidor de imágenes partimage, este paso es opcional.

## 4.3 Configuración inicial de Debian 6

Una vez terminada la instalación hacemos login con el usuario root en la máquina y empezamos a configurarla.

Lo primero que debemos hacer es configurar la tarjeta de red del servidor con una IP estática. Para ello editamos el fichero `/etc/network/interfaces` con el siguiente contenido:

```
auto eth0
iface eth0 inet static
address 192.168.50.45
netmask 255.255.0.0
gateway 192.168.50.1
```

En este caso, hemos configurado la primera tarjeta de red (eth0) y la dirección IP del servidor es la 192.168.50.45.

Ahora hay que editar el fichero `/etc/apt/sources.list` para poder instalar software desde los repositorios oficiales de Debian.

Comentamos las líneas que hagan referencia a la instalación de software mediante el cd de instalación y añadimos o descomentamos, según el caso, las siguientes líneas:

```
deb http://ftp.fr.debian.org/debian/ squeeze main non-free
deb-src http://ftp.fr.debian.org/debian/ squeeze main non-free

deb http://security.debian.org/ squeeze/updates main contrib
deb-src http://security.debian.org/ squeeze/updates main contrib
```

Ya está el sistema preparado para instalar todo el software necesario por repositorios.

## 4.4 Instalación del servidor DHCP

La instalación se hace con los siguientes comandos:

```
apt-get update
apt-get install isc-dhcp-server
```

Ahora configuramos el servicio.

Editamos el fichero `/etc/default/isc-dhcp-server` modificando la siguiente directiva:

```
INTERFACES="eth0"
```

Lo siguiente es editar el fichero `/etc/dhcp/dhcpd.conf` donde se define el rango de direcciones IP que se servirán a los host conectados a la red.

```
allow bootp;
subnet 192.168.0.0 netmask 255.255.0.0 {
    range 192.168.20.10 192.168.20.20;
    option routers 192.168.50.1;
    filename "pxelinux.0";
}
```

La directiva **allow bootp** indica que vamos a permitir el arranque por red.

La directiva **filename "pxelinux.0"** indica qué fichero se descargará en el cliente en el momento del arranque.

Este fichero se ejecutará en el cliente para que pueda encontrar los ficheros de configuración del menú, los cuales tendremos que crear posteriormente.

Es obligatorio que el fichero `pxelinux.0` esté ubicado en la misma máquina donde hemos instalado el servidor DHCP, por lo tanto, el servicio TFTP también estará instalado en dicha máquina.

Por último, podemos arrancar o parar el servicio respectivamente con los siguientes comandos:

```
/etc/init.d/isc-dhcp-server start
/etc/init.d/isc-dhcp-server stop
```

## 4.5 Instalación del servidor TFTP

Para instalarlo basta con ejecutar los siguientes comandos en la consola:

```
apt-get update  
apt-get install tftpd-hpa
```

El asistente de instalación nos preguntará cuál va a ser el directorio raíz para el servidor tftp, dejamos el que viene por defecto: /srv/tftp

En el directorio **/srv/tftp** pondremos todos los ficheros necesarios para que se descargue el menú de arranque en los clientes, así como los programas (imágenes initrd.gz) necesarias para el funcionamiento de cada selección.

## 4.6 Instalación del servidor de Imágenes PARTIMAGE

A fecha de la redacción de este proyecto, la versión disponible en el repositorio de Debian 6 es la 0.6.8.

Para conseguir la última versión de partimage es necesario instalarlo en el sistema compilando el código fuente.

La versión 0.6.9 incluye algunas mejoras de estabilidad, por lo que es muy conveniente instalarlo de esta forma.

Nos dirigimos a la página oficial del producto y descargamos el paquete de la zona de descargas.

<http://www.partimage.org/Download>

Descomprimos el paquete con el comando:

```
tar -xjvf partimage-0.6.9.tar.bz2
```

Creamos un directorio donde realizar la instalación.

```
mkdir /custom-partimage
```

Antes de compilar hay que preparar el sistema para ello:

```
apt-get install gcc g++ make
```

Ahora nos colocamos en el directorio donde se hizo la descompresión del paquete y compilamos:

```
./configure --prefix=/custom-partimage --disable-ssl  
make  
make install
```

Esta operación instala a la vez el cliente y el servidor de partimage en las siguientes ubicaciones:

```
/custom-partimage/sbin/partimage  
/custom-partimage/sbin/partimaged (demonio partimage)
```

Al incluir la opción `--disable-ssl` hemos deshabilitado las conexiones mediante el protocolo SSL al servidor, al contrario de lo que podría parecer, esto es una medida de seguridad.

Partimage funciona de manera que las versiones del cliente y el servidor deben coincidir exactamente para que sean compatibles, deshabilitando el ssl hemos conseguido que ningún cliente se conecte al servidor, a menos que haya sido compilado con estas opciones. No se podrán subir imágenes al servidor, por ejemplo, desde un live CD, o desde cualquier otro cliente que haya sido instalado mediante repositorios.

Así mismo, el servidor de partimage incluye un sistema de autenticación bastante seguro, pues se basa en permisos otorgados a un usuario del sistema operativo.

Debemos crear un usuario para la autenticación:

```
useradd -d /var/lib/partimaged -s /bin/false -c "Partimaged user" partimag
```

Este usuario no tendrá permiso para loguearse en el sistema operativo, podemos comprobarlo buscando la línea correspondiente en el fichero de usuarios `/etc/passwd`, donde debe aparecer al final la shell por defecto como `/bin/false`:

```
cat /etc/passwd | grep partimag  
partimag:x:1001:1001:Partimaged user:/var/lib/partimaged:/bin/false
```

El directorio `/var/lib/partimaged` será el repositorio de imágenes del servidor y es el único directorio del sistema donde el usuario partimag debe tener permisos de escritura.

```
mkdir /var/lib/partimaged  
chown partimag:partimag /var/lib/partimaged
```

Y le asignamos una contraseña al usuario que será la que nos pedirá el servidor partimage cuando subamos imágenes:

```
passwd partimag
```

Ahora tenemos que incluir al nuevo usuario en la lista de usuarios de partimage. Esta configuración se hace en el fichero `/custom-partimage/etc/partimagedusers`, el cual debe tener permisos de lectura para todos.

```
chmod +r /custom-partimage/etc/partimaged/partimagedusers  
nano /custom-partimage/etc/partimaged/partimagedusers
```

```
#add only users allowed to connect partimaged  
# (only one login per line)  
#sample # user 'sample' is allowed to connect partimaged  
partimag
```



## Configurar partimaged como servicio de sistema:

Vamos a crear un script de inicio para arrancar y parar el servidor fácilmente y además arranque automáticamente si reiniciamos la máquina:

```
nano /etc/init.d/partimaged
```

```
#!/bin/sh
# Script de arranque del servicio partimaged en /etc/init.d/partimaged
case "$1" in
start)
echo "Iniciando servicio... "
#comando a ejecutar para iniciar el servicio
/custom-partimage/sbin/partimaged -n -D -d /var/lib/partimaged/
    if [ $? -eq 0 ];then
        echo "OK"
    else
        echo "ERROR"
    fi
;;
stop)
echo "Deteniendo servicio..."
# Aqui comando a ejecutar para detener el servicio
killall partimaged
;;
*)
echo "Modo de empleo: /etc/init.d/partimaged {start|stop}"
exit 1
;;
esac
exit 0
```

Damos permisos de ejecución al script:

```
chmod +x /etc/init.d/partimaged
```

Y por último creamos el servicio para los distintos modos de ejecución:

```
update-rc.d partimaged start 99 2 3 4 5 . stop 01 0 1 6 .
```

A partir de este momento se puede arrancar y parar el servidor con los comandos:

```
/etc/init.d/partimaged start
/etc/init.d/partimaged stop
```

## 5. El menú de arranque PXELINUX

A continuación se describe el proceso para crear un menú de arranque a medida.

Para conseguir los ficheros necesarios nos dirigimos a la página oficial del proyecto Syslinux:  
<http://www.syslinux.org>

En la zona de descargas buscamos la última versión disponible y descargamos el paquete completo que tendrá un aspecto como este:  
syslinux-4.04.tar.bz2

Lo descomprimos:

```
tar -xjvf syslinux-4.04.tar.bz2
```

Una vez descomprimido buscamos los ficheros que nos interesan:

- pxelinux.0
- chain.c32
- vesamenu.c32

Y los colocamos en las siguientes rutas (creando previamente los directorios necesarios):

```
mkdir -p /srv/tftp/virtuendo-installer/i386/boot-screens  
mv pxelinux.0 /srv/tftp/  
mv chain.c32 /srv/tftp/virtuendo-installer/i386/  
mv vesamenu.c32 /srv/tftp/virtuendo-installer/i386/boot-screens/
```

Ahora debemos crear una imagen para el fondo del menú:

Con un editor de imágenes, por ejemplo, Gimp, creamos una imagen de nombre **splash.png** con un tamaño de 640x480 píxeles.

Movemos la imagen a su ubicación en el directorio de datos del servidor TFTP:

```
mv splash.png /srv/tftp/virtuendo-installer/i386/boot-screens/
```

Lo siguiente es confeccionar el menú de opciones.

Creamos el fichero por defecto con el siguiente contenido:

```
nano /srv/tftp/pxelinux.cfg/default
```

```
include virtuendo-installer/i386/boot-screens/menu.cfg  
default virtuendo-installer/i386/boot-screens/vesamenu.c32  
prompt 0  
timeout 0
```

En este punto hemos definido el fichero **menu.cfg** para colocar las opciones del menú, aunque estas pueden estar distribuidas en varios ficheros distintos, esto se consigue poniendo una directiva `include`:

```
include virtuendo-installer/i386/boot-screens/txt.cfg
```

Esta sería un ejemplo de configuración para el fichero `menu.cfg`:

```
menu hshift 2
menu width 70
menu margin 6
menu title Menu de Arranque^
include virtuendo-installer/i386/boot-screens/stdmenu.cfg
include virtuendo-installer/i386/boot-screens/txt.cfg
menu begin advanced
    menu title Opciones Avanzadas
    include virtuendo-installer/i386/boot-screens/stdmenu.cfg
    label mainmenu
        menu label ^Atras...
        menu exit
    include virtuendo-installer/i386/boot-screens/adtxt.cfg
menu end
```

En el ejemplo anterior se ha creado un menú desplegable para las opciones avanzadas. En este caso el fichero `txt.cfg` contiene las opciones principales y el fichero `adtxt.cfg` las opciones avanzadas.

Ahora habría que editar los ficheros **txt.cfg** y **adtxt.cfg** con las opciones del menú, este sería un ejemplo:

```
default menu
label 1
    menu label ^Título de la primera opción del menú
    kernel virtuendo-installer/i386/linux
    append vga=788 initrd=virtuendo-installer/i386/imagen-initrd1.gz loglevel=4
    text help
    esto es un texto de ayuda para mostrar cuando se resalta esta opcion
    endtext
label 2
    menu label ^Título de la segunda opción
    kernel virtuendo-installer/i386/linux
    append vga=788 initrd=virtuendo-installer/i386/imagen-initrd2.gz loglevel=4
    text help
    este es el texto de ayuda para la segunda opcion
    endtext
```

Podemos poner tantas etiquetas label como sean necesarias, una por cada opción del menú.

El fichero **stdmenu.cfg** es de configuración, aquí podemos definir los colores, la posición de los títulos, el tamaño de letra, etc.

Lo incluimos en el fichero menu.cfg anteriormente mediante una directiva include.

Este sería un ejemplo de configuración:

```
menu background virtuendo-installer/i386/boot-screens/splash.png
menu color title      * #000000 #00000000 none
menu color border     * #00000000 #00000000 none
menu color sel        * #000000 #550000dd *
menu color unsel      * #cc333355 #00000000 none
menu color hotssel    1;7;37;40 #000000 #550000dd *
menu color hotkey     * #cc333355 #00000000 none
menu color tabmsg     * #000000 #00000000 none
menu color help       * #ee333388 #00000000 none
menu vshift 13
menu rows 12
menu helpmsgrow 11
# The command line must be at least one line from the bottom.
menu cmdlinerow 16
menu timeoutrow 16
menu tabmsgrow 18
menu tabmsg Press ENTER to boot or TAB to edit a menu entry
```

Aquí se ha descrito brevemente como diseñar un menú básico, pero si queremos conocer con más detalle para que sirve cada directiva podemos ver el manual de pxelinux en la siguiente dirección:

[http://www.syslinux.org/wiki/index.php/PXELINUX#Custom\\_Menu\\_Example\\_with\\_sub-menus](http://www.syslinux.org/wiki/index.php/PXELINUX#Custom_Menu_Example_with_sub-menus)

## 6. Las imágenes INITRD

### 6.1 Creación de imágenes INITRD

Cuando el usuario selecciona una opción del menú de arranque, el servidor TFTP descarga y ejecuta en la memoria RAM del cliente dos cosas:

- Un kernel Linux.
- Una imagen initrd.

En ese momento disponemos de un espacio de usuario disponible para ejecutar scripts y software en la máquina, éste es el objetivo que buscamos creando imágenes INITRD personalizadas.

Lo primero que necesitamos es una imagen del tipo initramfs genérica. Lo más fácil es utilizar la que incluye nuestro servidor Debian 6 para su propio arranque.

Vamos a crear un directorio de trabajo y copiamos la imagen allí.

```
mkdir /workbench  
cp /boot/initrd.img-2.6.32-5-686 /workbench/initrd.gz
```

La imagen es un fichero comprimido con gzip y empaquetado con una utilidad llamada cpio. Necesitamos descomprimirlo para trabajar sobre el sistema de ficheros.

```
cd /workbench  
zcat initrd.gz | cpio -i  
rm initrd.gz
```

Ahora deberíamos tener en el directorio de trabajo un sistema de ficheros similar a la raíz de cualquier S.O. GNU/Linux:

```
.  
├── bin  
├── conf  
├── etc  
├── init  
├── lib  
├── sbin  
└── scripts
```

Este es un sistema de ficheros básico que sólo sirve para arrancar el sistema operativo, por lo tanto hay que llenarlo con el software que necesitamos.

Para ello hay que tener en cuenta también las librerías que utiliza cada programa e incluirlas en la imagen. Lo podemos comprobar con la utilidad **ldd**.

```
ldd /custom-partimage/sbin/partimage

linux-gate.so.1 => (0xb77a9000)
libslang.so.2 => /lib/libslang.so.2 (0xb76a8000)
libcrypt.so.1 => /lib/libcrypt.so.1 (0xb7676000)
libz.so.1 => /usr/lib/libz.so.1 (0xb7661000)
libnewt.so.0.52 => /usr/lib/libnewt.so.0.52 (0xb764e000)
libbz2.so.1.0 => /lib/libbz2.so.1.0 (0xb763d000)
libpthread.so.0 => /lib/libpthread.so.0 (0xb7624000)
libstdc++.so.6 => /usr/lib/libstdc++.so.6 (0xb752f000)
libm.so.6 => /lib/libm.so.6 (0xb7509000)
libgcc_s.so.1 => /lib/libgcc_s.so.1 (0xb74ea000)
libc.so.6 => /lib/libc.so.6 (0xb73a5000)
libdl.so.2 => /lib/libdl.so.2 (0xb73a1000)
/lib/ld-linux.so.2 (0xb77aa000)
```

Este es un trabajo arduo, pues tenemos que copiar una a una las librerías de cada programa, así como el binario del programa en sí.

Los programas los copiaremos al directorio **bin** de nuestra imagen.

```
cp /custom-partimage/sbin/partimage /workbench/bin
```

Y las librerías al directorio **lib**.

```
cp /usr/lib/libz.so.1 /workbench/lib
```

Una vez tenemos los programas, debemos crear un script de inicio que haga los trabajos que queremos en la máquina según el caso.

El script de inicio por defecto de la imagen initrd se encuentra en la raíz con el nombre **init**.

Es un programa escrito en lenguaje **Shell Script** que se encarga de montar los sistemas de ficheros especiales como `/sys` `/proc` y `/dev`, por lo que sólo introduciremos modificaciones después de que el script haya hecho este trabajo.

De esa forma no nos tendremos que preocupar por el acceso al hardware, como la tarjeta de red o los discos duros.

Podemos hacer una llamada a nuestro script personalizado dentro de init justo antes de que el script intente montar el sistema de ficheros raíz en el disco duro de la máquina (recordemos que este es un script de arranque de Debian).

```
maybe_break mount
log_begin_msg "Mounting root file system"
. /scripts/local
parse_numeric ${ROOT}
maybe_break mountroot
mountroot
log_end_msg

maybe_break bottom
#[ "$quiet" != "y" ] &&
log_begin_msg "Running /scripts/init-bottom"
run_scripts /scripts/init-bottom
#[ "$quiet" != "y" ] &&
log_end_msg
. /cargamodulos

#inicio del script personalizado
/customscript
```

Una vez hecho este paso, editamos el fichero `/workbench/customscript` que será un script de bash donde escribiremos nuestro programa, por ejemplo, podría ser uno que haga el particionado del disco automáticamente.

Más adelante se describen cada uno de los programas creados para este proyecto y su forma de uso.

La personalización de la imagen `initrd`, es decir, incluir los programas y librerías, sólo se hace una vez, luego para cada programa nuevo podemos modificar solamente el fichero `/customscript` y comprimir la imagen con otro nombre.

Vamos a ver como comprimir una imagen `initrd`.

Para usarla con nuestro sistema tenemos que colocarla en el directorio `/srv/tftp/virtuendo-installer/i386/`

```
cd /workbench
find . | cpio -o -H newc | gzip > /srv/tftp/virtuendo-installer/i386/nombreimagen.gz
```

No debemos olvidar que además de la imagen, necesitamos un kernel linux en el mismo directorio, al igual que con la imagen, vamos a utilizar el kernel del servidor Debian 6.

```
cp /boot/vmlinuz-2.6.32-5-686 /srv/tftp/virtuendo-installer/i386/linux
```

## 6.2 Administración de las imágenes INITRD

Dentro del directorio **/imágenes descomprimidas** hay un subdirectorio por cada imagen initrd, donde se despliega su contenido descomprimido:

```
.  
├── formater.gz  
├── guardar.gz  
├── installer.gz  
└── partimage.gz
```

En estos directorios es donde se harán las modificaciones pertinentes de cada programa, recordemos que el script a editar siempre se llama **customscript**

También se puede incluir aquí cualquier programa o librería que necesitemos tener dentro de la imagen, copiándolo a su correspondiente directorio, por ejemplo, **bin**

Siempre que se haga una modificación, tenemos que comprimir la imagen initrd y colocarla en el directorio correspondiente de TFTP para utilizarla. De esta forma se conserva siempre la integridad y coherencia entre la imagen descomprimida y comprimida.

*Nota: deberíamos tener un backup de las imágenes originales por seguridad.*

Para facilitar esta tarea he creado un pequeño shell script de nombre **creaimagen** que hace el trabajo, el cual se puede llamar desde cualquier posición del árbol de directorios.

```
#!/bin/bash  
find . | cpio -o -H newc | gzip > /srv/tftp/virtuendo-installer/i386/$1
```

La forma de usarlo es la siguiente:

1. Nos colocamos en la raíz de la imagen descomprimida.

```
cd /imagenes descomprimidas/installer.gz
```

2. Editamos la imagen.

```
nano customscript
```

3. Lanzamos el script pasándole como parámetro el nombre de la imagen.

*Nota: En cada directorio hay un fichero vacío con el mismo nombre que la imagen, de forma que se puede rellenar con el tabulador y así no nos equivocamos al escribirlo.*

```
creaimagen installer.gz
```



## 7. Opciones del menú de arranque

### 7.1 Opciones con llamadas a los programas desarrollados

Cada imagen **INITRD** que hemos creado se corresponde con una opción del menú:

Opción del menú	Imagen Initrtd
Preparar el disco para una nueva instalación	formater.gz
Subir una imagen al servidor	guardar.gz
Instalar un sistema operativo.	installer.gz
Iniciar el cliente de partimage	partimage.gz

➤ Opción **“Preparar el disco para una nueva instalación”**:

Este programa sirve para hacer un particionado del disco y dar formato a las particiones de forma automática, permitiendo seleccionar entre NTFS y EXT3 para la partición /dev/sda1. El esquema de particiones está previamente definido en el programa y se crea de forma automática.

Este es un paso previo a la creación de imágenes de sistemas operativos, esto se explica en el apartado correspondiente: *9 Creación de imágenes de disco.*

➤ Opción **“Subir una imagen al servidor”**:

Con este programa se lanza automáticamente el cliente de partimage para subir imágenes al servidor con una configuración predefinida que evita la mayoría de los problemas que podrían surgir. Por lo tanto, esta es la opción que se debe usar siempre para crear y subir nuevas imágenes de disco en el servidor.

En el programa se incluye un sistema de autenticación con usuario y contraseña que sólo permite 3 intentos de conexión cada vez que se lanza la aplicación.

Una vez que accedamos se abre el cliente de partimage.

El servidor de imágenes partimaged tiene su propio sistema de autenticación, por lo que hay un doble nivel de seguridad.

Además el servidor rechazará cualquier intento de conexión que no sea el de nuestro sistema, la razón se describe en el apartado: *4.6 Instalación del servidor de Imágenes PARTIMAGE*

➤ Opción **“Instalar un sistema Operativo”**:

Este es el programa más importante del sistema, pues se encarga de hacer la instalación desatendida de Sistemas Operativos.

En primer lugar el usuario debe decidir si necesita particionar el disco de la máquina, para facilitar la tarea, se le muestra un esquema con el estado actual del disco.

Después el usuario sólo debe seleccionar de una lista desplegable, el sistema operativo que quiere instalar, así como el grupo al que pertenece su ordenador (cada máquina tendrá una etiqueta pegada en la torre para ver esta información).

Automáticamente se inicia el cliente partimage y se restaura en la máquina objetivo la imagen seleccionada.

Una vez restaurada la imagen, se debe reiniciar el ordenador.

Tras este paso, el sistema operativo ya está disponible para su uso.

➤ Opción “**Iniciar el cliente de partimage**”:

Inicia el cliente de partimage sin opciones preestablecidas.

Se puede usar para crear imágenes de disco, copiarlas, subirlas o restaurarlas, tanto en local como en remoto.

Incluye autenticación.

## 7.2 Opciones de arranque de sistemas operativos

En el menú de arranque hay dos opciones que nos permiten arrancar un sistema operativo de la máquina:

➤ Arrancar desde el primer disco duro:

Dirige la secuencia de arranque hacia el MBR del primer disco duro de la máquina `/dev/sda`

➤ Arrancar el sistema operativo:

Dirige el arranque hacia la primera partición del primer disco duro (`/dev/sda1`), que es donde el instalador automático hace las restauraciones de imágenes del Sistema Operativo.

Su funcionamiento es muy sencillo, y se basa en un procedimiento llamado **chainloader**.

Este es un ejemplo de configuración del menú para arrancar desde la primera partición del primer disco duro:

```
label 1
    menu label ^Arrancar desde el primer disco duro
    menu default
    kernel virtuendo-installer/i386/chain.c32
    append hd0 1
```

El menú carga el fichero **chain.c32** y luego se le añade la ruta del disco o partición que queremos arrancar.

Los discos se nombran a partir del 0 y las particiones a partir del 1.

Ejemplos:

Arrancar desde el MBR del primer disco.	append hd0
Arrancar desde la segunda partición del primer disco.	append hd0 2
Arrancar desde la primera partición del segundo disco.	append hd1 1

***Nota:** Teniendo una opción del menú resaltada, y pulsando la tecla **Tab**, podemos editar las opciones de arranque de la selección. Así se puede arrancar desde cualquier disco o partición escribiendo su número.*

## 7.3 Otras opciones

➤ Opción “**Ayuda**”:

Simplemente muestra un pequeño manual de uso rápido del sistema.

➤ Opción “**Iniciar Sysreccd**”:

Inicia el arranque de una instancia de Sysreccd.

**SystemRescueCd** es un disco de rescate Linux, que normalmente se arranca desde un Live CD. Una vez arrancado se dispone de un **sistema operativo completo**, con entorno gráfico disponible, que se puede utilizar para reparar el sistema, o para cualquier otra tarea, como por ejemplo, navegar por Internet.

El sistema descargará desde un servidor HTTP el disco completo en la memoria RAM de la máquina y lo arrancará.

En el siguiente apartado se describe el proceso de instalación y configuración de esta característica.

## 8. Arranque en red de SystemRescueCd

Lo primero que necesitamos, es una instancia “viva” de Sysreccd en el servidor. Podemos descargar el CD Live de la página oficial del proyecto:

<http://www.sysresccd.org/Download>

Grabamos la imagen iso en un CD virgen y lo introducimos en la unidad lectora. Una vez sysreccd arrancado, abrimos una terminal y copiamos el contenido completo del directorio **/tftpboot** a la ruta de nuestro servidor TFTP.

Previamente debemos montar en el sistema la partición del disco donde está instalado el servidor, así como crear un directorio para alojar los ficheros.

```
mount /dev/sda1 /mnt
mkdir /mnt/srv/tftp/sysreccd
cp -r /tftpboot /mnt/srv/tftp/sysreccd
```

Tras este paso, reiniciamos el servidor, retiramos el disco de Sysreccd y arrancamos normalmente con Debian 6.

Ahora hay que modificar el menú para que cargue el kernel y la imagen Initramfs de sysreccd:

```
label menuavanzado4
  menu label ^Iniciar Sysreccd.
  kernel sysreccd/rescue32
  append initrd=/sysreccd/initram.igz boothttp=http://192.168.50.45/sysrcd.dat
  text help
  Arrancar una instancia de sysreccd en esta maquina
  endtext
```

Los ficheros **sysrcd.dat** y **sysrcd.md5** corresponden a la imagen del disco sysreccd que ocupa unos 300 MB, por lo tanto deben ser descargados con el protocolo HTTP, ya que si lo descargamos desde el propio servidor TFTP, la descarga es lenta e insegura, pues usa el protocolo UDP para la transferencia.

Necesitamos un servidor web para la descarga HTTP, instalamos **Apache Web Server**.

```
apt-get update
apt-get install apache2
```

Una vez instalado, movemos los ficheros de datos de Sysreccd al directorio de datos de Apache.

```
cd /var/www  
mv /srv/tftp/sysreccd/sysrcd.dat /var/www  
mv /srv/tftp/sysreccd/sysrcd.md5 /var/www
```

Con estos pasos ya tenemos disponible el arranque de SystemRescueCd desde el menú.



## 10. Creación de imágenes de disco

Para este proyecto se han creado imágenes de disco funcionales para todos los equipos de las aulas de informática del I.E.S Miguel Romero Esteo.

Los sistemas operativos elegidos para las instalaciones son:

- Windows XP Profesional Edition SP3
- Ubuntu 13.04 Raring Ringtail.

De estos dos sistemas se han creado **imágenes maestras** que puedan ser replicadas en los demás equipos de la red.

Para que una imagen de disco pueda ser restaurada en un equipo, el hardware del mismo debe ser compatible con el de la máquina en la cual se creó la imagen maestra.

Para lograr el objetivo se han agrupado las máquinas según su arquitectura de procesador y modelo de placa base.

Cada ordenador tendrá un etiqueta identificativa de su grupo pegada en la torre, de manera que sea perfectamente visible.

El sistema es escalable en cuanto a las imágenes disponibles, si se desea incluir un nuevo Sistema Operativo, habrá que crear una imagen maestra distinta para cada grupo de máquinas y modificar la lista del instalador.

Para añadir o modificar grupos y sistemas disponibles del instalador, debemos editar dos arrays que se encuentran en el script de arranque de la imagen initrd **installer.gz**

```
declare -a sistemas=(WindowsXP Ubuntu-13.04)
declare -a grupos=(grupo-A grupo-B grupo-C grupo-D grupo-E)
```

La edición de imágenes initrd se detalla en el apartado: *6.2 Administración de las imágenes INITRD*

### 10.1 Particionado del disco

El esquema de particiones del disco duro de las máquinas cliente debe tener un formato establecido, para que el proceso de restauración de imágenes sea correcto.

Cada máquina tendrá dos particiones al principio del disco, una de 10 GB destinada a alojar el sistema operativo y otra de 1 GB reservada para el swap o área de intercambio, necesaria para los sistemas operativos GNU/Linux.

La primera partición **/dev/sda1** tendrá formato NTFS o EXT3, aunque el instalador rectificará el tipo de sistema de ficheros automáticamente cuando restauremos una partición, por lo que este aspecto no debe preocuparnos.

La segunda partición **/dev/sda2** tendrá formato SWAP.

Este esquema se deberá respetar siempre para que el sistema funcione correctamente, el espacio de disco restante se podrá particionar libremente.

En el espacio de disco libre el usuario también puede instalar tantos sistemas operativos como necesite, respetando siempre las dos primeras particiones para nuestro sistema.

Mediante el instalador tendremos la opción de particionar el disco de forma automática, en caso de que el esquema de particionado no sea el correcto, de esta forma creamos las particiones necesarias para la restauración.

Este es el aspecto que debería presentar el disco, tanto para la creación de nuevas imágenes maestras, como para la restauración de las mismas:

```
parted -l
```

Number	Start	End	Size	Type	File system
1	1000KB	10.0GB	10.0GB	primary	(ntfs o ext3)
2	10.0GB	11.0GB	1000MB	primary	linux-swap

Para facilitar la creación de imágenes, se ha desarrollado un script que hace el particionado automático del disco duro, sólo tenemos que seleccionar en el menú la opción: **Preparar el disco para una nueva instalación.**

A continuación se describe el proceso de creación de imágenes maestras para sistemas operativos de Microsoft y GNU/Linux.

En todas las instalaciones, tanto Windows, como Ubuntu, se ha configurado el usuario por defecto:

- nombre: “operador”
- contraseña: “entrada”.

## 10.2 Creación de imagen maestra de Windows XP SP3

Tal como se describe en el apartado anterior, el primer paso antes de crear una imagen maestra, es el particionado automático del disco, para ello arrancamos la máquina y seleccionamos en el menú de arranque en red la opción: Preparar el disco para una nueva instalación.

Siempre que instalemos un sistema operativo de Microsoft, seleccionaremos del menú desplegable la opción NTFS como sistema de ficheros para la partición /dev/sda1.

Ahora reiniciamos la máquina e introducimos el CD de instalación de Windows XP SP3.

Cuando comienza la instalación de Windows, nos pregunta donde queremos instalarlo, marcamos la primera partición del disco “C:” y seleccionamos la opción: **dejar intacto el sistema de archivos actual (sin cambios).**



Este punto es muy importante, ya que el instalador de Windows cambia el tamaño de las particiones al dar formato al disco, provocando errores en la restauración de las imágenes, por lo que tenemos que instalarlo sobre el sistema de ficheros NTFS creado anteriormente por nuestro programa automático.

Ahora continuamos la instalación normalmente y esperamos hasta que arranque el sistema operativo por primera vez.

Lo siguiente es instalar los drivers necesarios para que funcione todo el hardware correctamente.

Para esta tarea he recurrido a la utilidad **Driver Pack Solution**.

Se trata de un CD que incluye una colección de drivers para todo tipo de hardware, simplemente lo introducimos en la máquina y dejamos que busque e instale los drivers automáticamente.

Una vez reiniciado el sistema y comprobado que tenemos todos los drivers instalados, procedemos a la instalación del software adicional que queramos incluir en nuestra **imagen maestra**.

Estas son las aplicaciones que se han instalado sobre Windows XP, que se pueden descargar gratuitamente de sus sitios oficiales en Internet:

- ✓ Avira free antivirus
- ✓ Openoffice
- ✓ Mozilla Firefox
- ✓ Adobe Reader
- ✓ Adobe Flash Player 11
- ✓ PeaZip

Una vez instalado el software, debemos eliminar todos los ficheros que se hayan quedado en el disco derivados del proceso, por ejemplo, las descargas de Internet, con el fin de tener una imagen lo más limpia posible.

También se han aplicado las siguientes configuraciones en el sistema:

- Se ha configurado la pantalla a 1024 x 768 píxeles.
- Se ha cambiado el fondo de pantalla por uno personalizado.
- Se han eliminado imágenes y música de muestra.
- En el centro de seguridad de Windows, se ha definido:
  - Protección antivirus – Activado.
  - Actualizaciones automáticas - Desactivado, posteriormente el usuario puede modificar esta directiva si lo considera necesario.
  - Firewall – Activado.

Cuando estemos seguros que no necesitamos hacer **ningún cambio más** en el sistema, procedemos a la desfragmentación de la partición C:\

Abrimos Panel de Control, Herramientas administrativas, Administración de equipos, Almacenamiento, Desfragmentador de disco y hacemos click sobre la unidad C:\. Seleccionamos **Desfragmentar**.

Esperamos que finalice la desfragmentación y pasamos a chequear el disco. Este proceso es importante por que evita una serie de errores durante la restauración con partimage.

Abrimos una consola de comandos:  
Inicio, Ejecutar, cmd.

```
chkdsk /f
```

El sistema avisará que hará la comprobación en el siguiente reinicio.  
Reiniciamos y esperamos que pase el chequeo de disco y se reinicie de nuevo automáticamente.  
Ahora esperamos que aparezca el menú de arranque en red.

En este momento tenemos una **imagen maestra** totalmente limpia lista para subirse al servidor de partimage.

Seleccionamos la opción del menú: **Subir una imagen al servidor**

El programa nos pedirá un nombre para la imagen, debemos tener en cuenta que los nombres de las imágenes disponibles en el repositorio deben coincidir con los de la lista que tenemos en el programa instalador **installer.gz**

El formato de nombre de las imágenes debería ser este:  
<grupo>-<SistemaOperativo>.gz

Un ejemplo podría ser:  
**grupo-A-WindowsXP.gz**

Partimage convierte el nombre al formato grupo-A-WindowsXP.gz.000, que es el que se buscará posteriormente en el momento de la restauración.

*Nota: Siempre que se creen imágenes de disco de cualquier versión de Microsoft Windows, su nombre tiene que comenzar por **Win**, ya que el instalador lo comprueba para elegir el formato correcto de la partición, en este caso NTFS.*

Después de escribir el nombre, se iniciará el cliente de partimage y se subirá la imagen al servidor, quedando disponible para su instalación en las demás máquinas del grupo correspondiente.

## 10.3 Creación de imagen maestra de Ubuntu 13.04

El primer paso, como siempre, es seleccionar en el menú de arranque la opción: Preparar el disco para una nueva instalación.

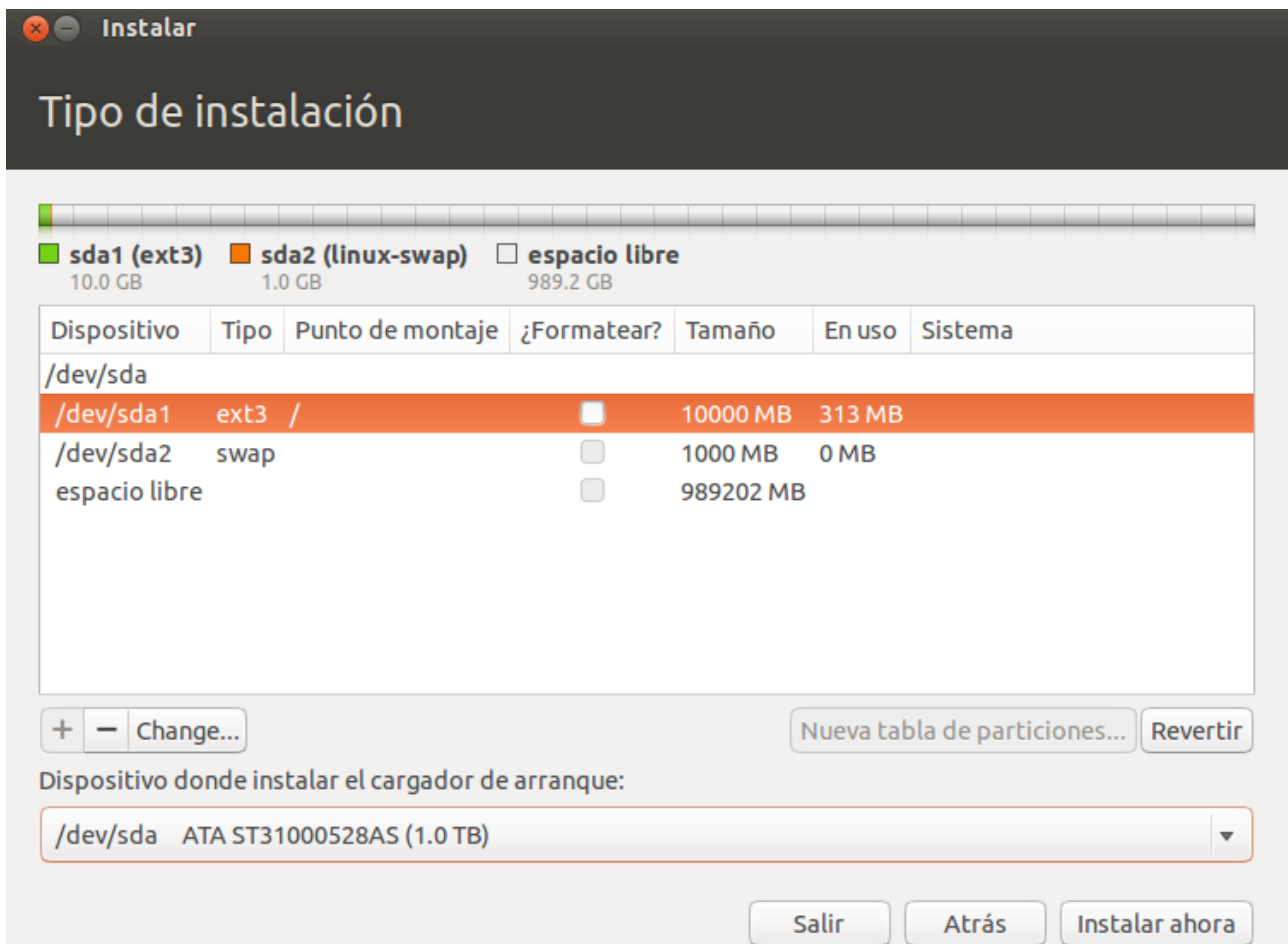
Elegimos el sistema de ficheros EXT3 para dar formato a /dev/sda1.

Introducimos el CD de instalación de Ubuntu y avanzamos normalmente hasta llegar al particionado del disco, donde elegiremos la opción que nos permita modificar el particionado de forma personalizada.

Seleccionamos la partición /dev/sda1 y ponemos como punto de montaje la raíz del sistema /. Como la partición ya tiene formato ext3, no marcamos la opción para formatear, al igual que en Windows, es mejor dejar intacto el sistema de ficheros para evitar errores.

Aunque el propio instalador permite la opción de instalar el cargador de arranque en la partición /dev/sda1, haremos esta tarea después manualmente, ya que es un procedimiento delicado y no siempre funciona.

Dejaremos la opción por defecto para la instalación de grub: /dev/sda



**Instalar**

### Tipo de instalación

**sda1 (ext3)** 10.0 GB
  **sda2 (linux-swap)** 1.0 GB
  **espacio libre** 989.2 GB

Dispositivo	Tipo	Punto de montaje	¿Formatear?	Tamaño	En uso	Sistema
/dev/sda						
/dev/sda1	ext3	/	<input type="checkbox"/>	10000 MB	313 MB	
/dev/sda2	swap		<input type="checkbox"/>	1000 MB	0 MB	
espacio libre			<input type="checkbox"/>	989202 MB		

Dispositivo donde instalar el cargador de arranque:

/dev/sda ATA ST31000528AS (1.0 TB) ▼

Continuamos la instalación normalmente y esperamos que arranque el sistema operativo.

Ubuntu 13.04 viene por defecto con la mayoría del software necesario para trabajar con el equipo, por lo que no he incluido ningún software adicional en las imágenes.

- ✓ No necesita antivirus.
- ✓ Incluye Openoffice
- ✓ Incluye Mozilla Firefox
- ✓ Incluye soporte para lectura de documentos en pdf.
- ✓ Incluye soporte para contenidos web Flash.
- ✓ Incluye herramientas de compresión de ficheros comunes como rar y zip.

Estas son las únicas modificaciones de configuración del sistema:

- Se ha configurado la pantalla a 1024 x 768 píxeles.
- Se ha cambiado el fondo de pantalla por uno personalizado.
- Se ha quitado el bloqueo de pantalla y la petición de contraseña.

Una vez tengamos todo el software instalado y configurado pasamos a la instalación de grub en la partición /dev/sda1.

Abrimos una consola y ejecutamos como root:

```
grub-install --root-directory=/boot2 /dev/sda1 --force  
grub-mkconfig -o /boot2/boot/grub/grub.cfg
```

El último paso es hacer un chequeo de la partición antes de subirla, este trabajo debe hacerse con la partición desmontada.

Al igual que en Windows, este chequeo evitará errores en la restauración con partimage.

Reiniciamos y arrancamos con **Sysreccd**, podemos usar el menú de arranque en red para ello.

Una vez arranque ejecutamos el comando:

```
e2fsck /dev/sda1
```

En ese momento tenemos una **imagen maestra** lista para subirse con partimage.

Volvemos a reiniciar y seleccionamos en el menú de arranque: **Subir una imagen al servidor.**

El nombre de la imagen tendrá este formato: **grupo-A-Ubuntu-13.04.gz**



## 12. Marketing

### Diseño de logo corporativo

Para dar personalidad propia al proyecto, he diseñado un logotipo con una tipografía específica, además he elegido un color de fondo que permita identificar el producto para su promoción.

### Promoción web

Complementando al desarrollo del proyecto, he comprado un dominio de Internet y he creado un portal web sencillo, para la futura promoción del producto.

El portal web pretende servir tanto para promocionar el sistema, como para tener una vía de contacto con cualquier persona interesada en su desarrollo, o incluso, para captar posibles clientes que demanden una implantación del producto.

[www.virtuendo.es](http://www.virtuendo.es)





- ✓ Menú de Arranque en Red.
- ✓ Instalación automática de Sistemas Operativos
- ✓ Arranque de sistemas Live CD en Red
- ✓ Gestión de Sistemas Operativos

Qué es Virtuendo   Cómo funciona   Presentación   Contacto   Acerca de mi

### Sistema de Administración centralizada

Virtuendo, es un proyecto integrado que surge de la necesidad de solucionar una serie de problemas derivados del uso continuado de los equipos informáticos en las aulas del ciclo de A.S.I.R. del I.E.S Miguel Romero Esteo, aunque su implantación es viable en cualquier aula o institución que disponga de una red de ordenadores.

En un ambiente de trabajo, en el que los ordenadores del aula se utilizan para realizar todo tipo de prácticas de clase, a menudo nos encontramos con que las máquinas no disponen de un sistema operativo instalado para su uso inmediato.

La instalación de cualquier sistema operativo, supone una pérdida de tiempo considerable, además de su posterior configuración y la instalación del software necesario para empezar a trabajar.

Con este proyecto se pretende resolver este problema, además de aportar otras funcionalidades añadidas.

*"Virtuendo es un sistema capaz de proporcionar al usuario que se sienta delante de un ordenador, un mecanismo de instalación de sistemas operativos ágil, rápido y simple."*

Administración de Sistemas Informáticos | José Felipe González Postigo | 2012/2013

## 13. Bibliografía y Recursos

Estos son sólo algunos de los recursos utilizados durante el desarrollo del proyecto.

### Sitios Web:

Protocolo PXE

[http://es.wikipedia.org/wiki/Preboot\\_Execution\\_Environment](http://es.wikipedia.org/wiki/Preboot_Execution_Environment)

Proyecto SYSLINUX

[http://www.syslinux.org/wiki/index.php/The\\_Syslinux\\_Project](http://www.syslinux.org/wiki/index.php/The_Syslinux_Project)

Menú de arranque en red PXELINUX

<http://www.syslinux.org/wiki/index.php/PXELINUX>

Desarrollo de imágenes INITRD

<http://www.thewireframecommunity.com/node/14>

Instalar grub en una partición:

<http://www.ubuntu-es.org/node/131346#.Ua9enxWLe1F>

Resolución de problemas con partimage:

<http://doc.ubuntu-es.org/Partimage>

<http://clona.pbworks.com/w/page/7807253/Problemas%20que%20pueden%20surgir>

<http://timwise.blogspot.com.es/2007/04/running-partimage-in-batch-mode.html>

Proyecto SystemRescueCD

[http://www.sysresccd.org/SystemRescueCd\\_Homepage](http://www.sysresccd.org/SystemRescueCd_Homepage)

### Páginas de manual de Linux :

*fdisk:* <http://linux.die.net/man/8/fdisk>

*parted:* <http://linux.die.net/man/8/parted>

*dhclient:* <http://linux.die.net/man/8/dhclient>

*partimage:* <http://pwet.fr/man/linux/commandes/partimage>