Servidor de Arranque Remoto REMBO

Marco Antonio Álvarez Iglesias

Índice de contenido

Introducción a Rembo

Para llevar a cabo este proyecto vamos a utilizar una herramienta llamada **Rembo** un potente software, su funcionalidad principal es dar servicio de arranque remoto, como su propio nombre índica, **Remote Boot**.

Para ello montaremos un servidor **GNU/Linux** basado en **Ubuntu:** ***Guadalinex,*** distribución actualmente promocionada por la **Junta de Andalucía** y que usaremos por su implantación en las aulas de institutos y en los centros de trabajo propios de la Junta de Andalucía.

Rembo podremos encontrarlo en la página oficial, http://www.rembo.com/ es un software propietario, **de pago**, y puede descargarse una versión de prueba de 30 días, cómo la que usaremos en este manual, de la misma página.

Configuración Previa

Una vez instalado **Guadalinex** como nuestro sistema operativo en función de **Servidor,** preparamos los componentes necesarios para la instalación de **Rembo**.

Configuración de Red

Primeramente tendremos que configurar nuestra tarjeta de red de modo estático, por defecto está en modo ***dhcp***, ya que al instalar el S**ervidor** D**hcp** tendremos que tener una **IP** fija ya que la máquina servidor repartirá las direcciones IP a los distintos equipos. La configuración de red en Linux se encuentra en el siguiente archivo: /etc/network/interfaces y podremos modificarla de la siguiente forma abriendo una **Terminal**.

- Configuramos /etc/network/interfaces con el siguiente comando.

```
$ sudo gedit /etc/network/interfaces
```

- Y tendremos que cambiar el contenido por lo siguiente.

```
# Used by ifup(8) and ifdown(8). See the interfaces(5) manpage or
# /usr/share/doc/ifupdown/examples for more information.
auto lo
iface lo inet loopback

mapping hotplug
        script grep
        map eth0

auto eth0
iface eth0 inet static
address 192.168.1.2
netmask 255.255.255.0
gateway 192.168.1.1
```

Nota: Las direcciones tienen que ser modificadas según la configuración de la red local.

- Una vez cambiada la configuración guardamos los cambios y reseteamos el servicio de red.

```
$ sudo /etc/init.d/networking restart
```

- Si todo ha ido bien nos aparecerá lo siguiente.

```
* Reconfiguring network interfaces...                                    [ ok ]
```

Instalación del DHCP

Una vez con la red debidamente configurada pasaremos a instalar el S**ervicior Dhcp** con lo que nuestro servidor podrá dar IP a aquellas máquinas que se conecten a éste. El servidor dhcp que usaremos será el más actual, **dhcp3-server**.

- Para instalar el dhcp3-server en Guadalinex basta con abrir una **Terminal** y utilizar el comando gestor de descargas **Apt** de la siguiente forma.

```
$ sudo apt-get install dhcp3-server
```

- Una vez instalado nos dará el siguiente error.

```
* Starting DHCP server...                                                [ fail ]
```

Esto se debe a que debemos de configurar el servidor de dhcp y esto se hace en nuestro archivo **dhcpd.conf**. Más adelante hablaremos de este archivo ya que no terminaremos de configurarlo del todo.

- Para configurar este archivo tendremos que ejecutar el siguiente comando.

```
$ sudo gedit /etc/dhcp3/dhcpd.conf
```

- Y tendremos que cambiar todo el contenido por lo siguiente.

```
subnet 192.168.1.0 netmask 255.255.255.0 {
 range 192.168.1.1 192.168.1.254;
        option routers                  192.168.1.1;
        option vendor-class-identifier  "PXEClient";
        option broadcast-address        192.168.1.255;
        default-lease-time 600;
        max-lease-time 7200;
}
```

Con esa configuración estamos describiendo muchas cosas. Para empezar nuestra red y subred, a continuación el rango de IP. Esas son las posibles IP que el servidor de dhcp va a repartir a sus clientes. Después marcamos la dirección del router, activamos la **opción 60** del dhcp 'PXEClient', la dirección broadcast, y el tiempo de concepción de las direcciones IP.

Más adelante en este mismo archivo tendremos que configurar las MAC de los distintos clientes para asignar una IP fija a cada máquina, es quizás, el trabajo más cansado, lo veremos más adelante.

- Una vez bien configurado y guardado el archivo **dhcpd.conf** tenemos que resetear el servicio de dhcp para su correcto funcionamiento.

```
$ sudo /etc/init.d/dhcp3-server restart
```

- Nos notificará que ha sido correctamente parado e iniciado.

```
* Stopping DHCP server...                                    [ ok ]
* Starting DHCP server...                                    [ ok ]
```

Con esto ya tendremos nuestro servidor dhcp perfectamente configurado y listo para instalar la herramienta REMBO. Vayamos a ello.

Instalación de REMBO

Proseguimos con la instalación de REMBO, para ello tendremos que descomprimir nuestro archivo **Rembo-Toolkit-Install.tar.gz** al siguiente directorio /usr/local/rembo/

- Para descomprimir el archivo y moverlo a otro directorio usamos los siguientes comandos.

```
$ sudo tar -xvf Rembo-Toolkit-Install.tar.gz
$ sudo mv rembo-2.0/ /usr/local/rembo
```

- También debemos de copiar el archivo ***rembo.key*** en el mismo directorio /usr/local/rembo/

```
$ sudo cp rembo.key /usr/local/rembo/
```

Nota: Para todos los comandos nos situamos en el directorio donde estén los archivos.

Configuración de Rembo.conf

Ahora entramos en la configuración del programa y para ello tendremos que tocar el archivo **/usr/local/rembo/rembo.conf** y cambiar unos parámetros mínimos.

- Abrimos el archivo correspondiente de la siguiente forma.

```
$ sudo gedit /usr/local/rembo/rembo.conf
```

- Y cambiamos sólo las siguientes lineas.

```
BaseDir "/usr/local/rembo"
NetPassword "tesidos"
Interfaces 192.168.1.2
```

Como vemos hemos configurado el directorio del programa, nuestra clave de acceso y la IP del servidor de arranque (nuestra IP). En este archivo vemos que la seguridad de la contraseña se encuentra en texto plano, por lo que es un riesgo que otros usuarios puedan leer este archivo. Por defecto, cualquier usuario que use el servidor podrá leer este archivo, por lo que es conveniente cambiar los permisos.

- Para cambiar los permisos del archivo lo haremos de la siguiente manera.

```
$ sudo chmod 600 rembo.conf
```

De esta forma el archivo sólo podrá verlo y configurarlo el **root**. y con esto terminamos por ahora la configuración de este archivo. Más adelante tendremos que volver a configurarlo.

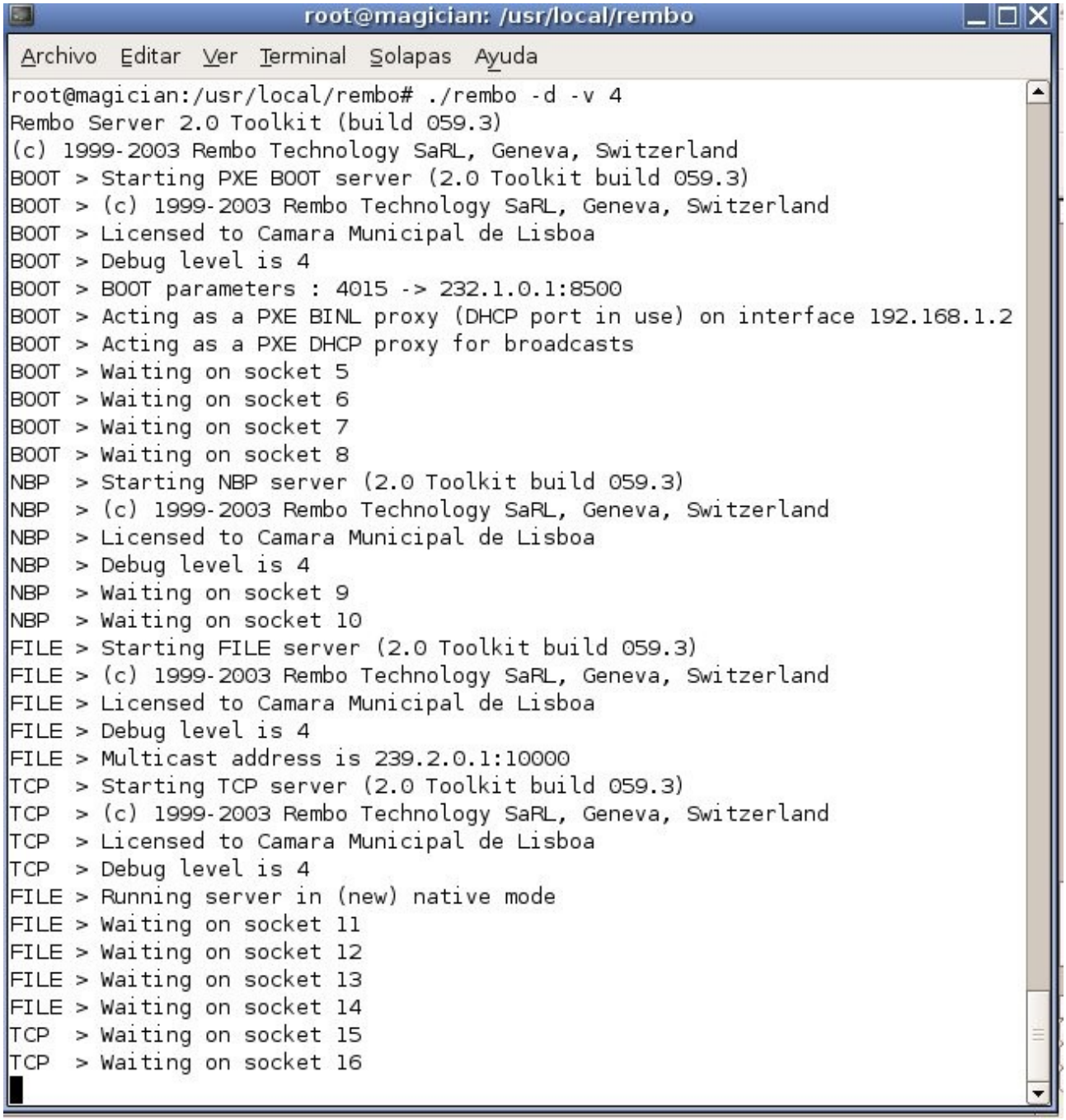
Arrancando Rembo por Primera Vez

Una vez realizado todas estas configuraciones estamos listos para arrancar Rembo en modo test para ver si funciona correctamente.

- Para iniciar rembo en modo test usamos el siguiente comando.

```
$ sudo su
# cd /usr/local/rembo
# ./rembo -d -v 4
```

- Y vemos que rembo funciona!



Configuración de Srvfiles.nc

Vemos que Rembo funciona correctamente y dejamos esa Terminal abierta, ahora está ocupada por Rembo. Sin tocar la Terminal, abrimos una **nueva Terminal** donde configuraremos el archivo ***srvfiles.nc*** donde tendremos que cambiar tan sólo una palabra.

- Abrimos el archivo.

```
$ sudo gedit /usr/local/rembo/srvfiles.nc
```

- Y cambiar la palabra ***install*** por nuestra contraseña ***tesidos.***

```
open 127.0.0.1 tesidos
```

- Después de hacer este cambio tenemos que transferir los archivos del servidor.

```
$ sudo su
# cd /usr/local/rembo/
# ./misc/netclnt srvfiles.nc
```

Procesará la transferencia de los datos que visualizaremos por la Terminal.

- Por motivos de seguridad, vamos a repetir el paso para cambiar permisos a los archivos.

```
$ sudo chmod 600 srvfiles.nc
```

Listo, ya tenemos este archivo configurado.

Arrancar Rembo en modo Nativo

- Ahora podemos parar Rembo en el primer Terminal que abrimos con las teclas **Control + C** y ejecutar Rembo en modo **nativo** para comprobar la configuración de esta forma.

```
$ sudo su
# cd /usr/local/rembo
# ./rembo -d -v 2
```

Visualizaremos de nuevo en la Terminal que el Servidor Rembo está corriendo como vemos en la última línea.

```
FILE > Running server in (new) native mode
```

Ya tenemos nuestro Servidor de Arranque Remoto funcionando, todo aquel equipo de nuestra red que se conecte con Boot de Arranque por Red estará bajo nuestro Servidor.

Arrancar Rembo Automáticamente

Vamos a depurar un poco el programa instalando Rembo como ***demonio*** en nuestro sistema para que cada vez que iniciemos Guadalinex se cargue automáticamente Rembo y no tengamos que ejecutarlos nosotros.

Configuración de Init.d

Este procedimiento es un poco más engorroso de lo que parece. Ya que el archivo que se encarga de convertir en demonio a Rembo viene preparado para una versión de Linux Red Hat, por lo que he tenido que modificar el archivo para adaptarlo a Guadalinex.

El archivo que ejecuta Rembo como demonio es **/usr/local/rembo/misc/initd** tendremos que modificar mucho este archivo.

- El archivo original /usr/local/rembo/misc/initd tiene este aspecto.

```
#!/bin/sh
# Sample init script for starting REMBO automatically on boot-up
# Created on a Linux RedHat distribution
# chkconfig: 2345 96 35
# description: REMote BOot Server

# This is the path where REMBO is installed
REMBODIR=/usr/local/rembo

# Source function library.
. /etc/rc.d/init.d/functions

# See how we were called.
case "$1" in
  start)
        echo -n "Starting REMBO services: "
        daemon $REMBODIR/rembo -c $REMBODIR/rembo.conf
        echo
        ;;
  stop)
        echo -n "Shutting down REMBO services: "
        killproc rembo
        echo
        ;;
  reload)
        echo -n "Reloading REMBO configuration: "
        killproc rembo -HUP
        echo
        ;;
  restart)
            $0 stop
        sleep 2
            $0 start
        ;;
  *)
        echo "Usage: rembo {start|stop|reload|restart}"
        exit 1
esac
```

- Podemos modificar este archivo con el comando.

```
$ sudo gedit /usr/local/rembo/misc/initd
```

- Y tenemos que cambiar el contenido por esto.

```
#!/bin/sh
#
# Este es un script para que REMBO se ejecute al inicio
#
# Modificado para un GNU/Linux Guadalinex
#
# chkconfig: 2345 96 35
# description: Remote Boot Server

# Este es el directorio donde REMBO está instalado.
REMBODIR=/usr/local/rembo

# Source function library.
#. /etc/rc.d/init.d/functions
. /lib/lsb/init-functions

# Nombres de ejecución.
case "$1" in
  start)
        echo -n "Starting REMBO services: "
        $REMBODIR/rembo -c $REMBODIR/rembo.conf
        echo
        ;;
  stop)
        echo -n "Shutting down REMBO services: "
        #pkill rembo
        start-stop-daemon --quiet --stop --exec $REMBODIR/rembo
        echo
        ;;
  reload)
        echo -n "Reloading REMBO configuration: "
        #killproc rembo -HUP
        echo
        ;;
  restart)
            $0 stop
        sleep 2
            $0 start
        ;;
  *)
        echo "Usage: rembo {start|stop|reload|restart}"
        exit 1
esac
```

- Una vez realizado los cambios y guardados, tenemos que copiar el archivo /usr/local/rembo/misc/initd en /etc/init.d/ y guardarlo con el nombre de rembo.

```
$ sudo cp /usr/local/rembo/misc/initd /etc/init.d/rembo
```

- Ahora ya podemos manejar mejor Rembo, podemos parar (stop), iniciar (start) o reiniciar (restart) el servicio sin tener que poner lineas demasiado largas. Tan sólo basta con hacer alguno de estos comandos.

```
$ sudo su
# cd /etc/init.d/
# ./rembo stop
# ./rembo start
# ./rembo restart
```

Creación de Enlaces Simbólicos

El último paso para terminar de configurar la instalación de Rembo es crear dos enlace simbólico uno en modo texto y otro en modo gráfico de la siguiente forma.

```
$ sudo ln -s /etc/init.d/rembo /etc/rc3.d/S95rembo
$ sudo ln -s /etc/init.d/rembo /etc/rc6.d/S95rembo
```

Configuración para los Clientes

Teniendo instalado REMBO en nuestra máquina pasaremos a configurar los archivos necesarios para los clientes a los que asignaremos una IP estática y entraremos en el profundo mundo de los scripts de rembo.

Preparación del "Conejillo de Indias"

La idea básica es la de tomar un equipo como conejillo de indias, que será en el que preparemos el equipo tal y como queremos que se instale en todos los demás, que deberán de ser exactamente con la misma configuración hardware. En este caso dispondremos de un equipo con un disco duro de 40 GB de disco duro y 512 de RAM y lo particionaremos de la siguiente forma.

20 GB	512 MB	9 GB	5,5 GB
Partición NTFS	Área de Intercambio	Partición EXT3	S/P
Windows XP	Linux / SWAP	GNU/Linux	Caché

- Partición Primaria NTFS Windows XP 20 GB
- Partición Primaria Área de Intercambio 512 MB (**¡Ojo! el tamaño de esta partición debe ser siempre el doble de la memoria RAM que tengamos instalada en el equipo, hasta 512 MB.**)
- Partición Primaria EXT3 GNU/Linux 9 GB
- Sin Partición para la caché 5.5 GB mínimo.

Cuando tengamos las particiones hechas con los programas como el OpenOffice, etc... instalados en nuestro ordenador tanto en Windows como en Linux estaremos preparados para hacer la imagen del disco duro de éste ordenador y mandársela a nuestro servidor de arranque remoto. Esto se hace a través de rembo y ya lo explicaremos a continuación.

Nota: Tendremos que ser equitativos a la hora de instalar programas para la creación de la imagen puesto que sólo contamos con 5.5 GB de caché en el disco duro del cliente que será donde posteriormente se guarde la imagen tanto como de Windows como de Linux. Así que procuremos que la imagen de cada sistema operativo no supere los 2,5 GB. Ésto es sólo para ahorrar tiempo a la hora de restaurar los Sistemas Operativos, por lo que es solo un comentario de recomendación.

Scripts de Creación de Imagen Windows y Linux

Antes de seguir vamos a crear dos scripts, uno para Windows xp y otro para Guadalinex. La función de estos scripts va a ser la de crear las imágenes de los sistemas operativos que tenemos en nuestro cliente y enviárselas al servidor. Ahora trabajaremos sobre el servidor.

- Primero creamos los directorios en nuestro servidor donde guardaremos las imágenes que mandaremos al servidor

```
$ sudo mkdir /usr/local/rembo/files/global/hdimages/
$ sudo mkdir /usr/local/rembo/files/global/hdimages/winnt/
$ sudo mkdir /usr/local/rembo/files/global/hdimages/linux
```

- Realizamos el script de imagen de Windows XP.

```
$ sudo gedit /usr/local/rembo/files/global/scripts/imagenxp.rbc
```

- Nos generará un archivo vacío donde tendremos que copiar lo siguiente.

```
CreateVirtualImage("nt","disk://0:1");
RemoveFile("link://nt/pagefile.sys");

if(FileExists("cache://global/hdimages/winnt/winxpbase.img"))
RemoveFile("cache://global/hdimages/winnt/winxpbase.img");
Synchronize("link://nt","cache://global/hdimages/winnt/winxpbase.img","");

FreeVirtualImage("nt");
```

- Realizamos el script de imagen de GNU/Linux.

```
$ sudo vi /usr/local/remo/files/global/scripts/imagenlinux.rbc
```

- Nos generará un archivo vacío donde tendremos que copiar lo siguiente.

```
//Restauramos imagen
BuildDiskImage(0,3,"cache://global/hdimages/linux/linuxbase.base");

//Copiamos fstab
CopyFile("disk://0:3/etc/fstab","cache://global/hdimages/linux/linuxbase.fstab");

//Copiamos Kernel
CopyFile("disk://0:3/boot/vmlinuz-2.6.12-9-386","cache://global/hdimages/linux/basekernel.krn");
```

Nota: Tendremos que poner la partición donde vaya a ser instalado Linux.
Nota: en VMLINUZ tendremos que escribir la versión que tenga cada uno en su máquina.

- Para ver nuestra versión de VMLINUZ.

```
$ ls /boot/ |grep vmlinuz
```

Conexión del "Conejillos de Indias" al Servidor de Arranque Remoto

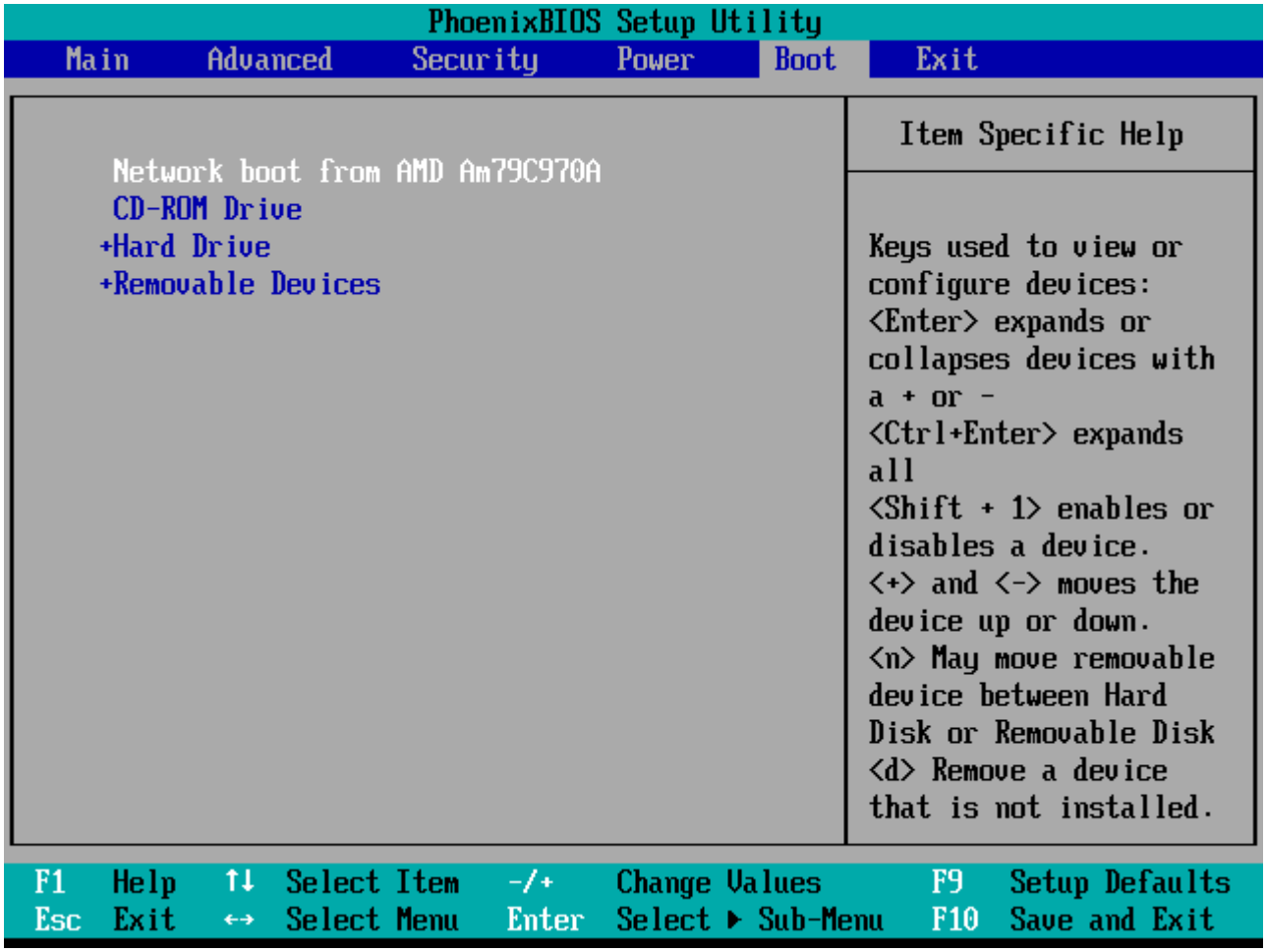
Una vez tengamos los scripts y el equipo listo para ser clonado a los demás ordenadores, tendremos que conectar el ordenador a nuestro servidor y pasarle las imágenes de nuestros sistemas operativos mediante los scripts realizados.

- Para llevar a cabo esta conexión tendremos que tener en cuenta los siguientes apartados.
  - Los dos equipos están conectados a la red
  - Rembo está ejecutado en el Servidor

```
$ sudo /etc/init.d/rembo start
```

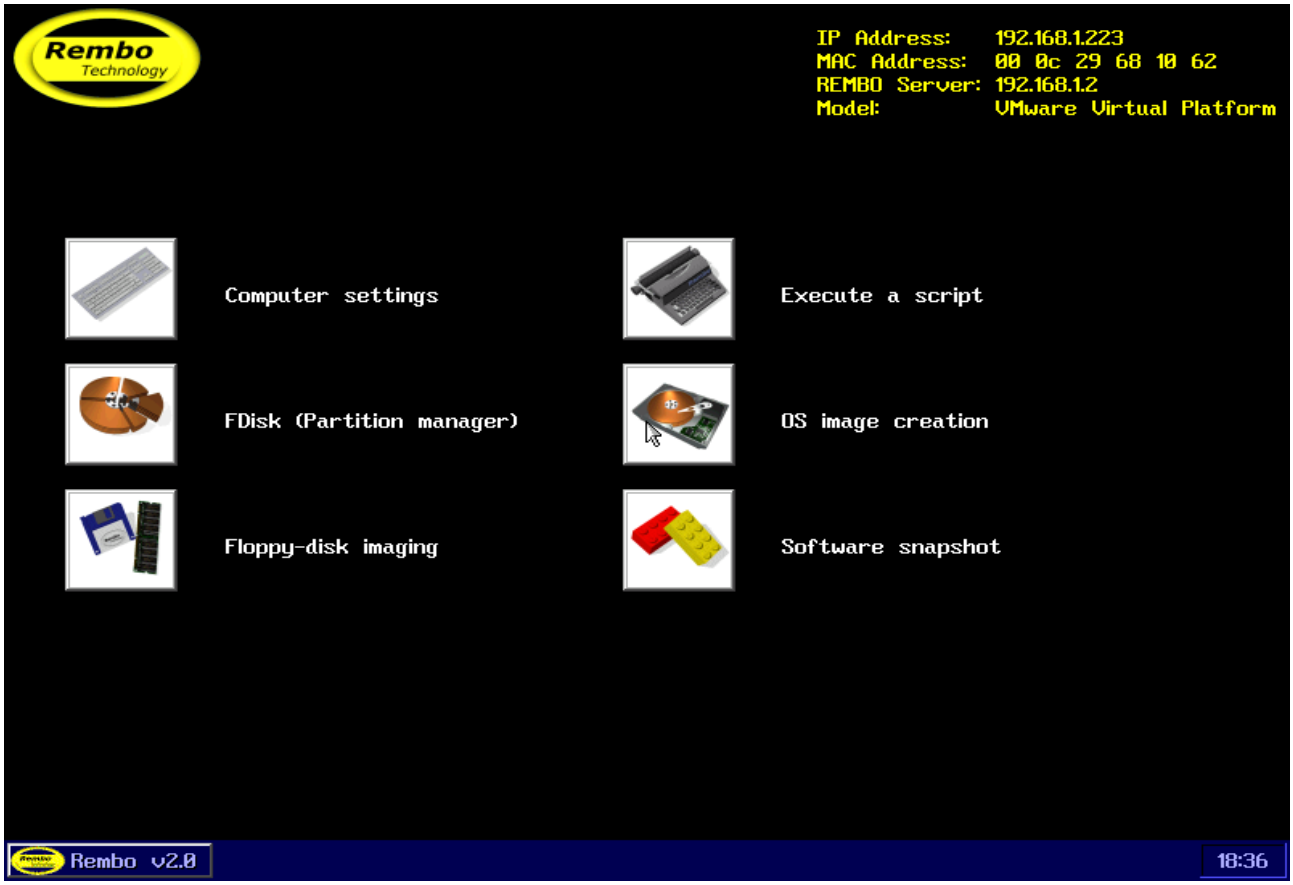
  - El equipo cliente tiene en la BIOS configurado el **boot de arranque principal** por LAN

Nota: Cuidado al configurar la BIOS, es muy peligroso si no tenemos destreza utilizándolo.



Ejemplo de Boot de Incio Principal LAN

Ahora sólo hace falta reiniciar el equipo cliente y automáticamente se conectará a nuestro servidor de arranque remoto rembo y nos mostrará la siguiente pantalla.

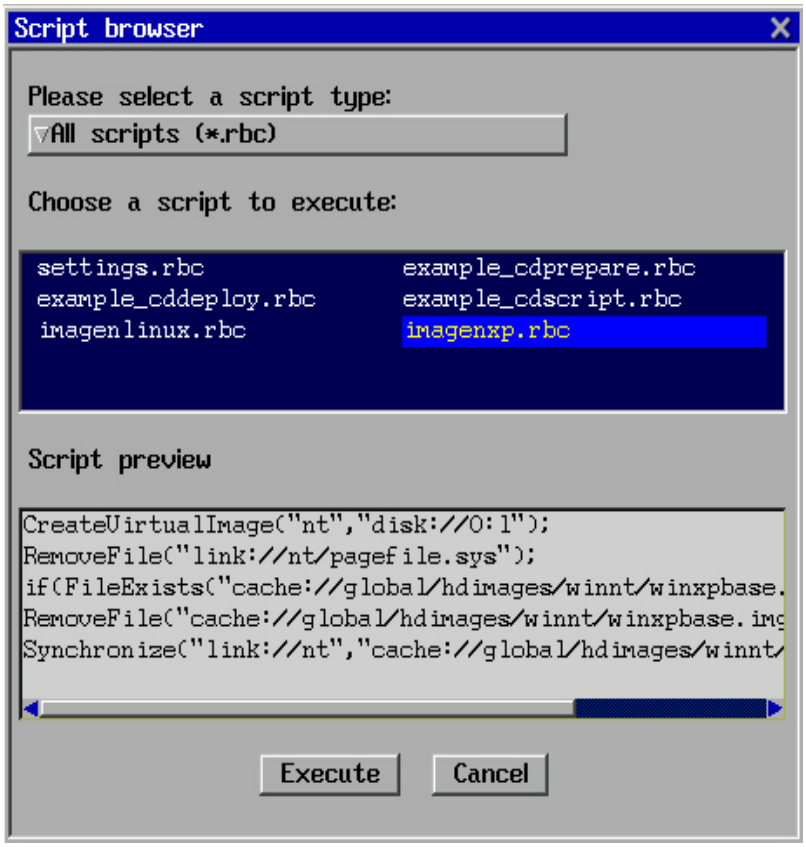


Esta es la pantalla por defecto de rembo, que más adelante configuraremos pues tiene unos privilegios un tanto peligrosos para un usuario normal y tendremos que realizar nuestro propio menú principal.

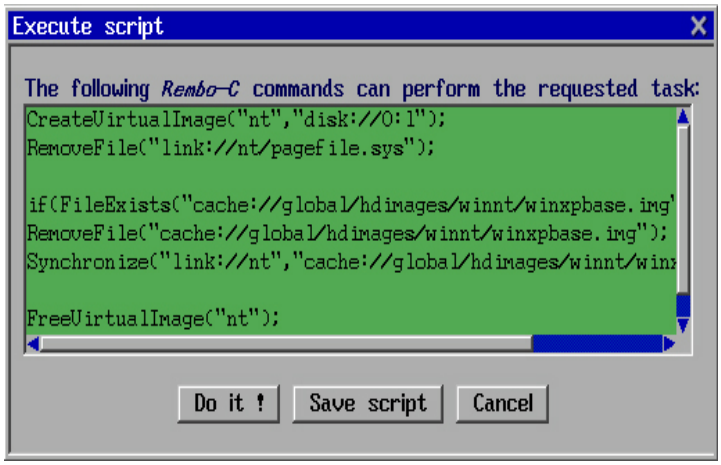
Cómo se puede observar en la parte superior nos da información de la IP que el servidor le ha dado a nuestro cliente, la dirección MAC de nuestra máquina, que más adelante tendremos que apuntar, y la IP del servidor de arranque remoto Rembo.

Ejecución de Scripts

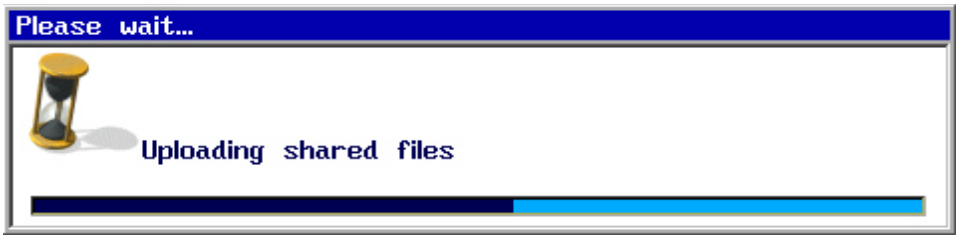
Llegados a este punto, pasaremos las imágenes de los sistemas operativos a nuestro servidor para poder difundirlo después a los demás ordenadores por red. Para esto, tendremos que usar los scripts que antes realizamos. Pulsamos sobre el botón **Execute Script** y seleccionamos el script de creación **imagenxp.rbc**



Y pulsamos sobre **Execute**, que nos avisará de nuevo con la siguiente pantalla:

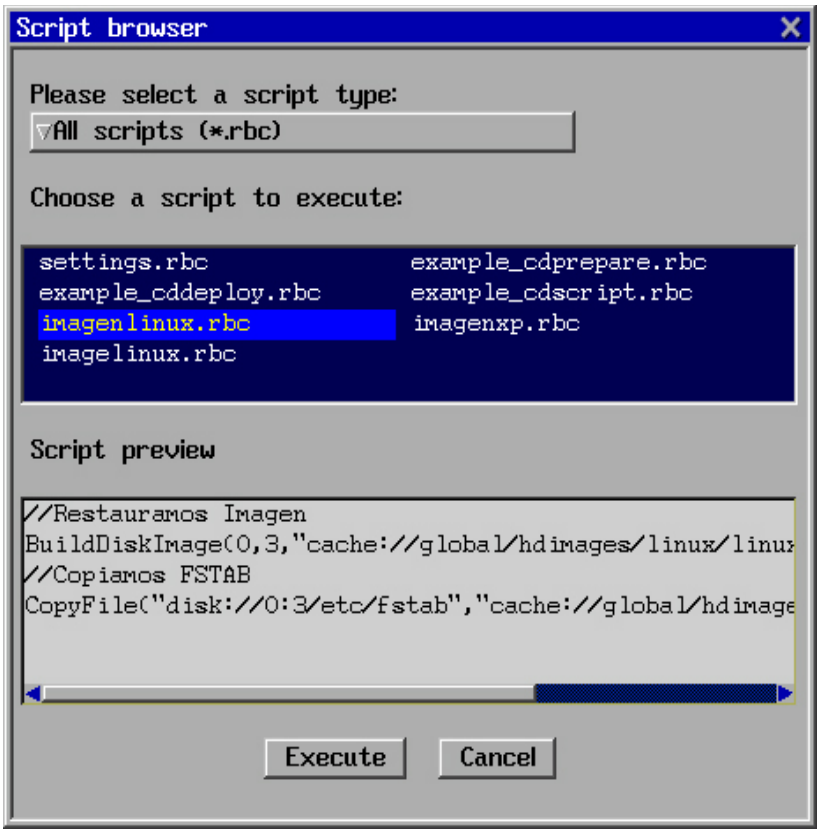


Pulsamos sobre **Do It !** y automáticamente rembo pasará la imagen de nuestro XP al servidor con la siguiente pantalla:

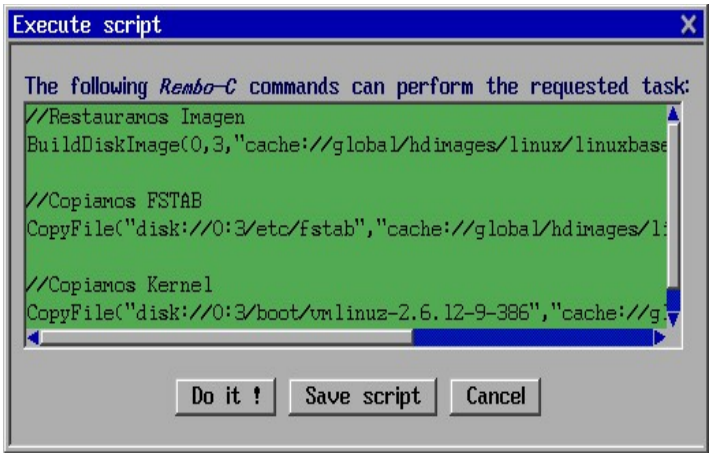


Este proceso suele tardar un poco pero tengamos en cuenta que sólo se hará una vez pues estamos copiando toda la imagen a nuestro Servidor. Cuando haya terminado ya tendremos en nuestro la imagen copiada en /usr/local/rembo/files/global/hdimages/winnt/winxpbase.img

Ahora pasaremos a crear la imagen de Guadalinex de la misma manera, tan sólo seleccionamos el script de **imagenlinux.rbc**.



Y pulsamos de nuevo sobre **Execute**, que nos avisará con la siguiente pantalla:



Pulsamos sobre **Do It !** al igual que en el paso anterior esperaremos y tendremos la imagen de linux en nuestro servidor. /usr/local/rembo/files/global/hdimages/linux/

Configuración de Dhcpd.conf

Cómo prometí anteriormente vamos a volver a configurar este archivo para asignar las Ips a nuestros clientes fijos. Esto nos dará más control sobre ellos, asegurándonos que las máquinas que están conectadas a nuestro servidor de arranque remoto son las que nosotros queremos que estén.

Lo que haremos será asignar una Ip para cada dirección **MAC**. La MAC o dirección física es una dirección que posee cada adaptador de red que la distingue entre todas las demás porque es única.

- Para ver la MAC de un ordenador podremos hacerlo de varias formas, si posee de un sistema operativo podremos verla de la siguiente forma.

```
En la Consola de Windows (Inicio > Ejecutar > cmd)
                                        > ipconfig /all
En la Terminal de Linux
                                          $ ifconfig
```

- Si no posee de sistema operativo podemos conectarlo por Red a nuestro servidor rembo que le asignará una Ip aleatoria y podremos ver la MAC en la parte superior.

Una vez con la MAC apuntada en un papel, tendremos que insertar una IP asociada a esa MAC para que cada vez que esa dirección MAC realice una petición a nuestro servidor de arranque remoto le asignemos la misma IP siempre.

- Para ello, insertaremos la MAC en el archivo ya comentado antes, dhcpd.conf.

```
$ sudo gedit /etc/dhcp3/dhcpd.conf
```

- Debiendo tener un aspecto parecido al siguiente.

```
subnet 192.168.1.0 netmask 255.255.255.0 {
  range 192.168.1.1 192.168.1.254;
        option routers                  192.168.1.1;
        option vendor-class-identifier    "PXEClient";
        option broadcast-address        192.168.1.255;
        default-lease-time 600;
        max-lease-time 7200;
}

# Clientes de prueba
        host PC01 {
                hardware ethernet 00:0C:29:C7:D6:B4;
                fixed-address 192.168.1.101;
        }
        host PC02 {
                hardware ethernet 00:0B:21:D3:D3:P0;
                fixed-address 192.168.1.102;
        }
```

Aquí hemos configurado dos ordenadores distintos que se conectarán a su IP correspondiente cuando nos haga una petición. Cuidado al poder las IP, pues debemos saber que la IP del Router y la del propio servidor Rembo ya están reservadas.

En el siguiente paso veremos como conectar un equipo a nuestro Servidor.

Configuración de Rembo.conf

Como comenté anteriormente, tenemos que volver a meter mano a este archivo para configurar adecuadamente nuestro servidor Rembo. Este archivo contiene la información para que un cliente a través de su MAC, se conecte a un grupo determinado con una página de inicio totalmente distinta que es la que crearemos para un grupo que también crearemos. Empezamos creando el grupo de nuestra clase, TESI.

- Abrimos el archivo.

```
$ sudo gedit /usr/local/rembo/rembo.conf
```

- Tendremos que añadir lo siguiente.

```
GROUP Tesi {
   StartPage "cache://global/tesi.shtml"

   # Example of host entry:
   Host 00:0C:29:F1:FD:84
}
```

Con esto creamos un grupo llamado TESI, con la página de inicio TESI.SHTML que después crearemos. Aquí tendremos que introducir cuantas MAC haya por ordenador que tengamos en nuestra clase.

- Y por seguridad, vamos a comentar todos los grupos que no sea el de TESI, incluido el grupo **Default**, que es donde se conectan aquellos Pcs que no tienen una MAC relacionada a una IP. Quedará con el siguiente aspecto.

```
#GROUP Default {
#    Options admin
#    StartPage "cache://global/rembo.shtml"
#}

#
# Admin group. Add hosts in this group if you want these hosts to see the
# admin page
#
#GROUP Admin {
#    StartPage "cache://global/admin.shtml"

   # Example of host entry:
   # Host 00:01:02:03:04:05
#}
```

Ahora ningún equipo que no tengamos especificado con su MAC en Rembo.Conf y Dhcpd.Conf no podrá conectarse a nuestro servidor.

Creación de nuestra Página de Inicio

Vamos a crear una página de inicio para nuestro grupo TESI con las siguientes opciones:

- Arrancar Windows XP
- Arrancar Guadalinex
- Limpiar Disco Duro
- Restaurar Windows XP
- Restaurar Guadalinex

Para ello tendremos que crear un archivo .shtml que es con el que trabaja rembo.

- Creamos el archivo tesi.shtml.

```
$ sudo gedit /usr/local/rembo/files/global/tesi.shtml
```

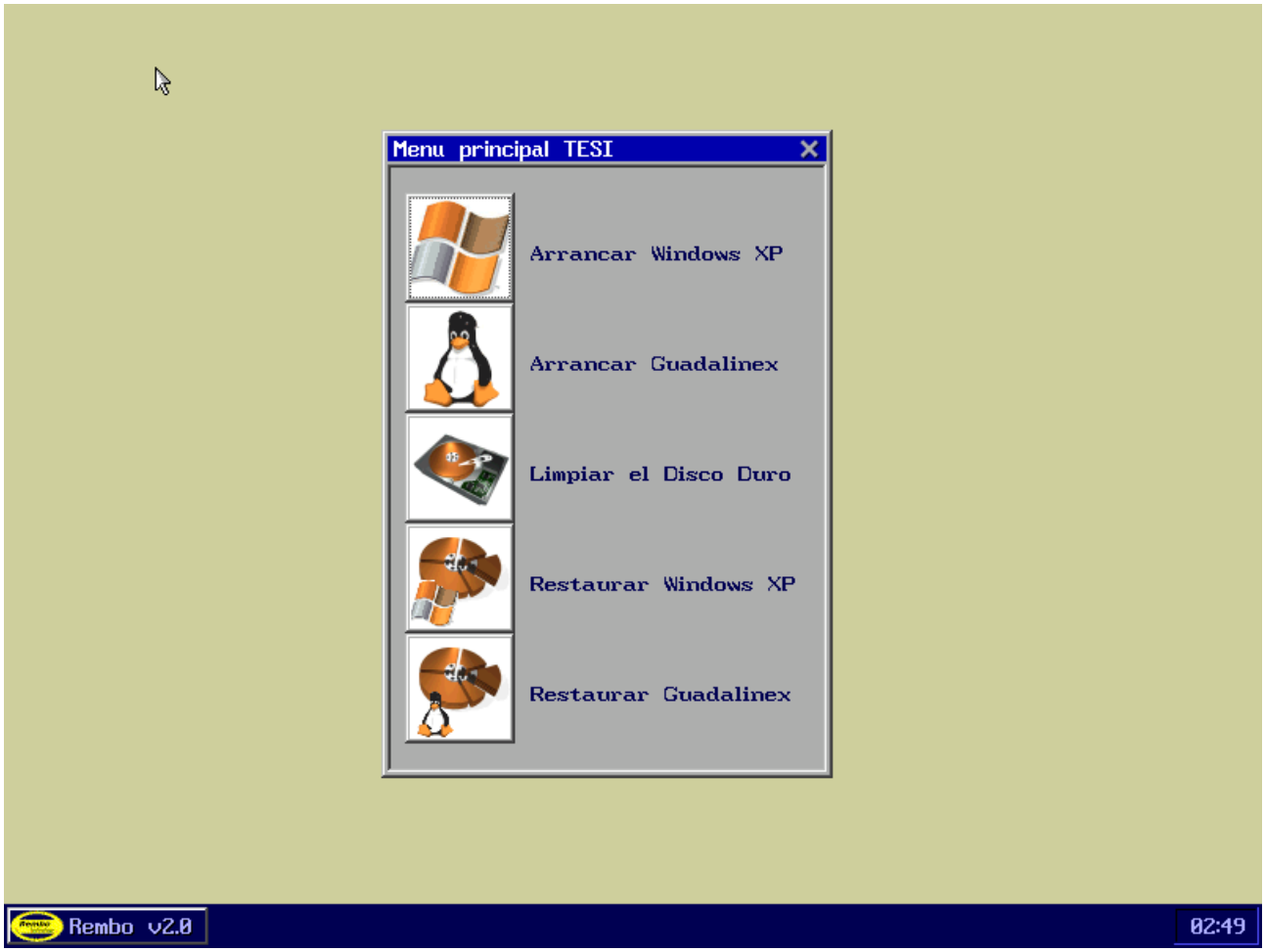
- Y editamos el contenido, quedando de la siguiente forma.

```
<!-- tesi.shtml - Probando Scripts de Rembo... -- -->

<script type="text/rembo-c">
        // Compatibilidad con versiones anteriores a la 0.99. Creo que no es necesario.
        if(FileExists("cache://global/plugins/rembo099.rbx"))
        {
                join(Exec("plugins/rembo099.rbx"));
        }
        // Cargando utils.rbx
        if(FileExists("cache://global/plugins/utils.rbx"))
        {
                join(Exec("plugins/utils.rbx"));
        }

        // Para declarar los modos por defecto
        str DefVideoMode;
        str DefKeyMap;
        str DefCodeMap;

    var BasicErrorHandler(var exc) { return exc; }
        with(BasicErrorHandler) try
        {
           // Obligamos a que se ejecute en modo 800x600 y teclado en español
       Settings.VideoMode = "800x600";
           Keyb("es");
           if(DefCodeMap != "")
                CodePage((int)DefCodeMap);
        }
        void LimpiarHD (void)
        {
                /* Las particiones que vamos a utilizar son las siguientes :
                        * Particion 1:  NTFS                20 Gb
                        * Particion 2:  LINUX-SWAP          512 MB
                        * Particion 3:  EXT3                9 Gb
                        * Libres para cache:                5,5 Gb
                */
      HideWindow("Menuppal");

                OpenMessage("limphd","Limpiando HD ...");
                delay(200);
                CloseWindow("limphd");
                SetPrimaryPartitions(0, "NTFS:20000000 LINUX_SWAP:512000
EXT3:9000000");
                HDClean(0,1);
                HDClean(0,2);
                HDClean(0,3);
                // Lo siguiente es para impedir que un listillo se salte el arranque remoto
                // dandole al escape
                OpenMessage("limphd","Limpiando MBR ...");
                delay(200);
                CloseWindow("limphd");
```

```
            SaveText("Solo se puede arrancar en red, pillin :D","disk://0:0/BootMessage");
            HDClean(0,0);
            SetBootablePartition(0,0);

            OpenMessage("finlimphd","Fin de limpieza");
            delay(300);
            CloseWindow("finlimphd");

ShowWindow("Menuppal");
   }

   void ArrancarWXP (void)
   {
CloseWindow("Menuppal");

            OpenMessage("awxp","Arrancando Windows XP...");
            delay(300);
            CloseWindow("awxp");
            HDBoot(0,1);

   }
   void ArrancarLinux (void)
   {

CloseWindow("Menuppal");

            OpenMessage("alinux","Arrancando Guadalinex...");
            delay(300);
            CloseWindow("alinux");
            LXBoot("cache://global/hdimages/linux/linuxbase.krn","","root=/dev/hda3");

   }
   void RestaurarWXP (void)
   {
      HideWindow("Menuppal");

            //LockKeyboard(true);
            //LockMouse(true);

            OpenMessage("rwxp","Restaurando Windows XP...");
            delay(300);
            CloseWindow("rwxp");

            HDClean(0,1);
            RestoreDiskImage(0,1,"cache://global/hdimages/winnt/winxpbase.img");
      //Synchronize("cache://global/hdimages/winnt/winxpbase.img","disk://0:1","b");
            var ip=StrParse(NetInfo.IPAddress,".");
            str nombre="TESI-"+ip[3];
            NTChangeName(nombre);
            OpenMessage("finrwxp","Fin de la restauracion");
            delay(300);
```

```
                CloseWindow("finrwxp");
                //LockMouse(false);
                //LockKeyboard(false);
      ShowWindow("Menuppal");
        }
        void RestaurarLinux (void)
        {
          HideWindow("Menuppal");
                //LockKeyboard(true);
                //LockMouse(true);
                OpenMessage("rlinux","Restaurando Guadalinex...");
                delay(300);
                CloseWindow("rlinux");
                HDClean(0,3);
                RestoreDiskImage(0,3,"cache://global/hdimages/linux/linuxbase.base");
          CopyFile("net://global/hdimages/linux/linuxbase.fstab","disk://0:3/etc/fstab");
                OpenMessage("finrlinux","Fin de la restauracion");
                delay(300);
                CloseWindow("finrlinux");
                //LockMouse(false);
                //LockKeyboard(false);
          ShowWindow("Menuppal");
        }
        var w = Window("root");
        // Para eliminar el menu del Rembo que sale como el de inicio de Windows.
        // Parece que no puede eliminarse debido al copyright (es su identificacion)
        // Pero si que se puede dejar sin entradas :D.
        /*SaveText(<html>
                        <select menu style="color: navy;">
                        </select>
                        </html>,"display://root/menu/SELF"); */
      // Para indicar el color de fondo
        w.widgets.color="#CCCC99";
        //w.widgets.image = "net://global/zergface.pcx";
        // nos sale ya el menu de arranque
        OpenMenu("Menuppal", 30, 30,
                        "<title>Menu principal TESI</title>"
                        "<style>B {font-weight: normal; color: red}</style>"
                        "<base href='cache://global/images'>",
                        {       {"<br><br>Limpiar el Disco
Duro","cache://global/images/lphd.pcx","LimpiarHD();"},
                                {"<br><br>Arrancar Windows
XP","cache://global/images/awxp.pcx","ArrancarWXP();"},
                                {"<br><br>Arrancar
Guadalinex","cache://global/images/alinux.pcx","ArrancarLinux();"},
                                {"<br><br>Restaurar Windows
XP","cache://global/images/rwxp.pcx","RestaurarWXP();"},
                                {"<br><br>Restaurar
Guadalinex","cache://global/images/rlinux.pcx","RestaurarLinux();"}
                        });
</script>  <body>  </body>
```

Es un poco largo, pero con esto tendremos un menú totalmente personalizado dónde tendremos un control total sobre las opciones a las que el usuario puede tener acceso. Este menú es totalmente personalizable a nuestras necesidades. Esta página es un ejemplo básico, en los Manuales Oficiales del Producto podréis informaros más acerca de esto. De momento, nosotros hemos creado esta pagina de inicio que será la que verán nuestros clientes al encender el Ordenador.



Las imágenes para los iconos las hemos cambiado para el script de rembo. Tienen un formato y debe ser del mismo tipo, con tamaño de 64x64 .PCX y que guardaremos en **/usr/local/rembo/files/global/images/**

- Las distintas opciones del menú que hemos creado van relacionadas a unas funciones en el archivo **tesi.shtml,** donde podremos arrancar los distintos sistemas operativos, limpiar y preparar el disco duro o restaurarlos.
- Lo primero que haremos al arrancar un equipo cliente por primera vez por red. Será L**impiar** el disco duro. Una vez realizado esto, instalaremos los Sistemas Operativos con las opciones de: **Restaurar** Windows y Linux.
- Después sólo tendremos que pulsar sobre el botón. **Arrancar**, para inicializar los distintos Sistemas Operativos.

Pues con esto, ya tenemos puesto en marcha nuestro Serviror de Arranque Remoto REMBO perfectamente con una configuración básica. Este programa tiene muchas aplicaciones interesantes a niveles más altos de administración de Sistemas. Os aconsejo leer el manual oficial y experimentar con las distintas opciones de imágenes incrementales con las que REMBO es capaz de trabajar. Espero que esta Guía os haya ayudado y os sirva.

Agradecimiento

En este apartado quiero hacer especial mención a **Jorge Torres Chacón** por su paciencia y especial apoyo en este tema.

Licencia