

Cómo Configurar Tarjetas Wi-Fi en LINUX usando controladores de Windows

Como ya se dijo anteriormente, configurar tarjetas WiFi con **chipsets** “modernos” o cuyos fabricantes no dan soporte alguno para LINUX es una gran odisea.

Si no existen *drivers* nativos de LINUX para vuestra tarjeta y/o no se han desarrollado todavía, nos queda una última opción... Probar si usando los controladores suministrados para NDIS de Windows pueden usarse en nuestro LINUX.

Dos maneras: **Driveloader** y/o **ndiswrapper**.

Comenzamos la odisea... Bienvenido a *linuxant*

Linuxant es un proyecto mediante el cual podemos usar **drivers** de **Windows** suministrados para plataformas **x86** en nuestro **LINUX**.

Driveloader es “la envoltura” que lo permite, en realidad es el controlador... el *driver* que “conecta” los controladores específicos de Windows para usarlos en LINUX, como si fuese una emulación... pero totalmente operativa... bueno, no del todo... puesto que **las herramientas de análisis, escaneo, auditoría, ataques, etc...** **NO FUNCIONARAN** con tarjetas configuradas con **driveloader**.

Al menos lograremos comunicación... al menos conseguiremos que la tarjeta funcione y podamos conectarnos a la red sin cables... pero **sólo si corremos LINUX en arquitecturas x86, AMD, Athlon y disponemos de los drivers para Windows**.

O sea, que los Mac y/o distribuciones LINUX para Power PC se quedarán compuestos y sin novia... uno de vosotros me lo comentó hace unas horas... lo siento... tu maravillosa Broadcom BCM4603 integrada en el portátil iBook se quedará sin funcionar... la opción que te queda es comprar algún dispositivo USB que tenga soporte.

Vamos al caso práctico... voy a configurar dos tarjetas *cardbus/pcmcia*, una con **driveloader** y otra mediante **ndiswrapper**, estas son:

US Robotics modelo USR5410 802.11g con **chipset** Texas Instrument **acx111** (*tiacx111*)
Sitecom modelo WL-011v2 802.11b/g con **chipset** **AMD** (no es ADM, es AMD)

La **USRobotics** con **driveloader**, la **sitecom** con **ndiswrapper**, sería perfectamente posible configurar ambas con cualquiera de ellos... pero mejor así... aprendamos las dos cosas.

Linuxant y **driveloader** cuestan o costaban unos 20 dólares por tarjeta y licencia... digo costaban porque la versión *trial* que se supone es para un mes, al menos a mí, no me “caducó”... posiblemente se hayan relajado... o que ya no hay que pagar nada... bien, el caso es que podremos usarlo, aunque sea por un mes.

También vamos a necesitar recompilar el *kernel*, o a lo mejor no,,, todo dependerá de la versión que tengas instalada en esto momentos y de la *distro* que se use.

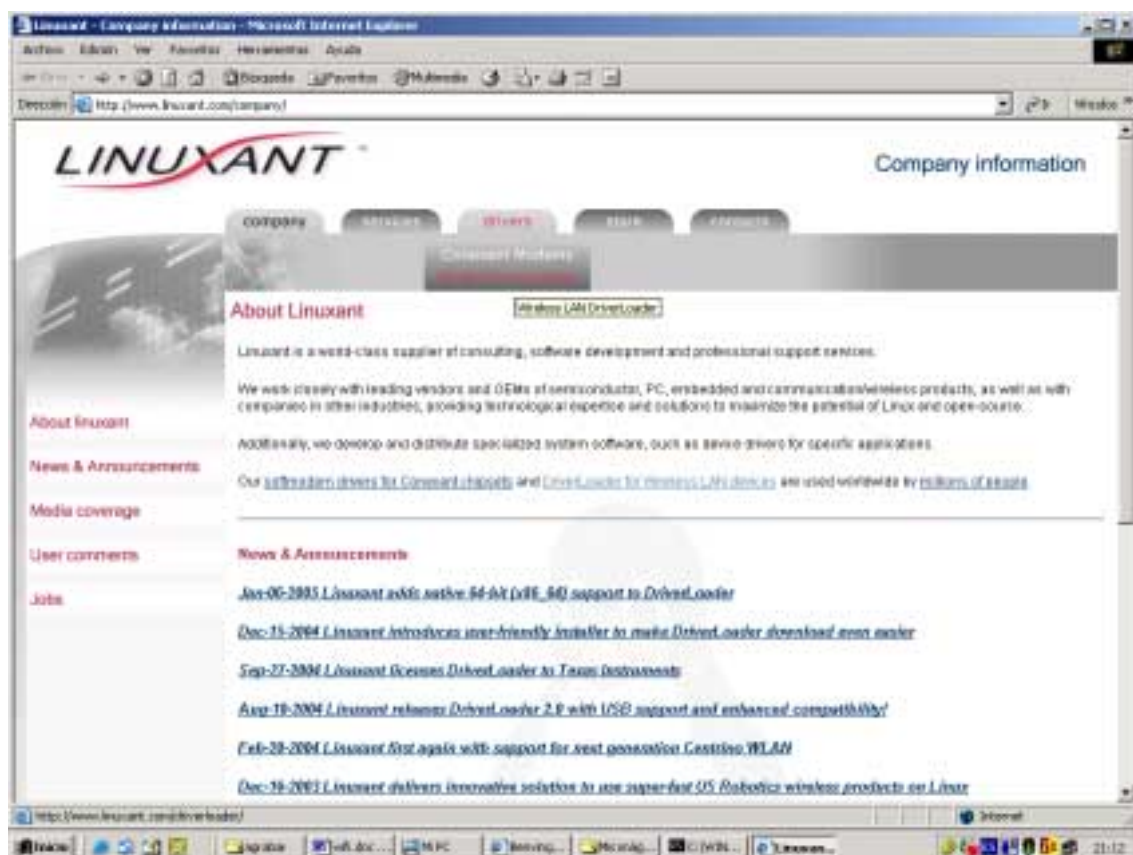
Para esta ocasión, **elegí Fedora Core 3 con núcleo 2.6.9** que parece buena elección... parece y nada mas... porque a pesar de lo actual y de un *kernel* tan moderno, tocará recompilar....

La forma de compilar el núcleo de LINUX y sus opciones se escapan de este texto, te recomiendo que te pases por la zona de LINUX si tienes dudas, problemas o no sabes cómo hacerlo... aquí se explicará “lo mínimo” de ello.

La tarjeta es esta:

Lo primero que haremos es **visitar la página de Linuxant**:

<http://www.linuxant.com/company>



y en ella **accedemos a la Ficha Drivers** y seleccionamos **WLAN DriverLoader**



Encontramos que da soporte a las siguientes **tarjetas pcmcia/cardbus**

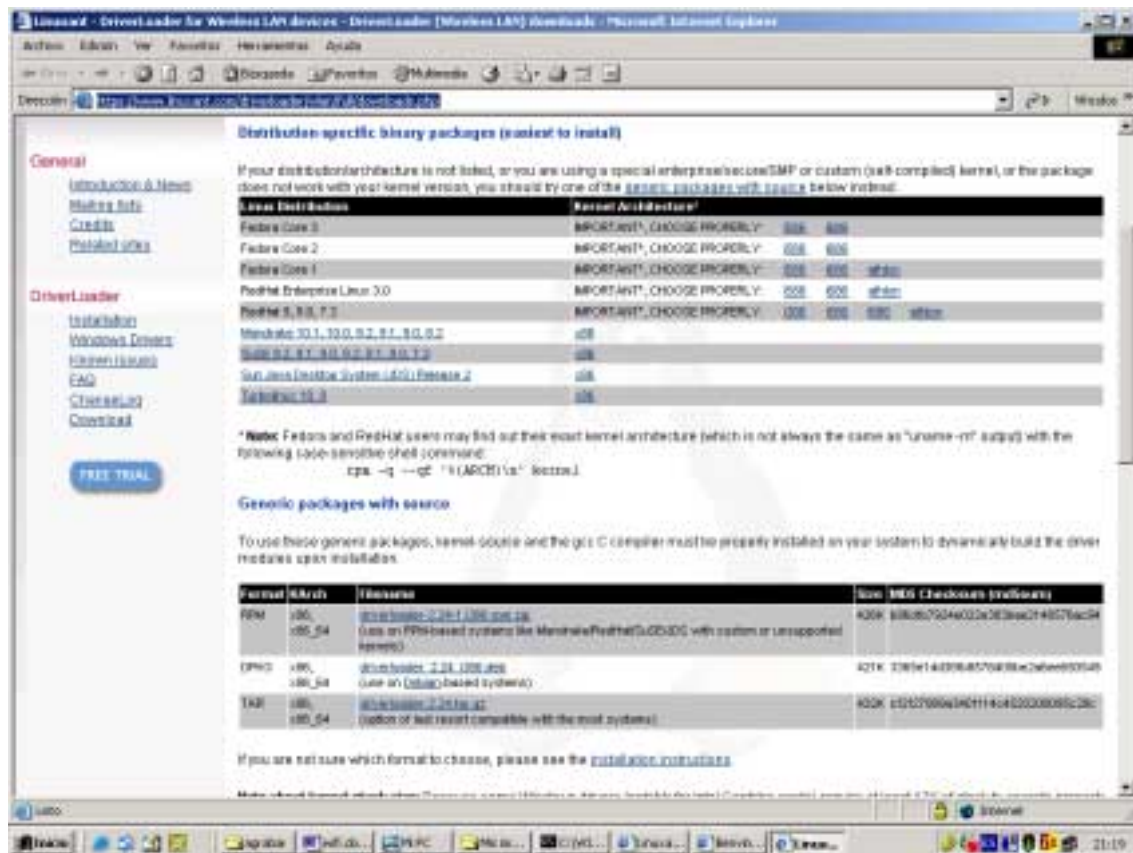
- Atheros
- Broadcom (AirForce 54g, on both x86 and 64-bit [x86_64] systems)
- Cisco (Aironet)
- INPROCOMM
- Intel (PRO/Wireless 2100, 2100A, 2200BG - Centrino)
- Conexant/GlobespanVirata/Intersil (Prism GT/Duette/Indigo)
- Marvell (Libertas)
- Realtek (RTL8180L)
- Texas Instruments (ACX100, ACX111/TNETW1130)
- ZyDAS (ZD1211)

Y otros dispositivos USB como:

- Belkin F5D7050 Wireless G USB Network Adapter
- D-Link DWL-G120 802.11g Adapter (PRISM)
- Linksys WUSB54G Wireless-G Network Adapter (PRISM)*
- U.S. Robotics USB5450 802.11g Wireless USB Adapter (BCM94333U)
- Blitz BWU723 Adapter (ZD1211)

Tras **verificar que nuestro chipset** está soportado tendremos que descargarnos la versión de **driverloader** que se corresponda con nuestra versión del núcleo de **LINUX** y su distribución:

<https://www.linuxant.com/driverloader/wlan/full/downloads.php>



Como yo estoy usando **Fedora Core 3** en un **Pentium iV**, utilizaré **i686**, ya sabéis... los usuarios de **RedHat**, **Fedora** y de otras distros también...la nomenclatura es:

- **i386**: Genérico para procesadores x386
- **i486**: Para procesadores 486
- **i586**: Normalmente AMD
- **i686**: Pentium

También dispones de versiones para más concretas para **AMD Athlon** y en formato **rpm** o paquetes para **debian**.. vamos lo de siempre...

Si eres usuario de **RedHat o Fedora**, puedes averiguarlo exactamente si haces esto:

```
rpm -q --qf '%{ARCH}\n' kernel
```

o bien,

```
uname -r
uname -m
```



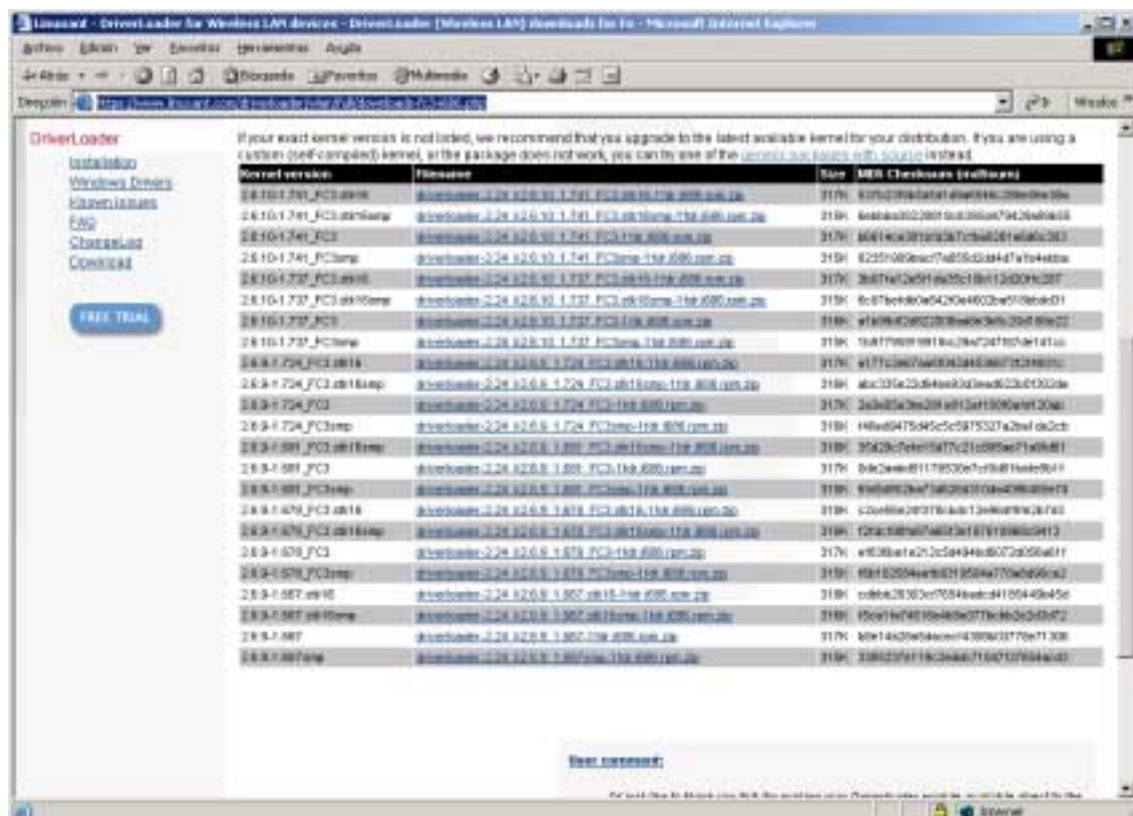
```

root@Vic-Fc:~
Archivo Editar Ver Terminal Solapas Ayuda

[root@Vic-Fc ~]# uname -m
i686
[root@Vic-Fc ~]# uname -r
2.6.9-1.667
[root@Vic-Fc ~]# rpm -q --qf '%{ARCH}\n' kernel
i686
[root@Vic-Fc ~]# █

```

<https://www.linuxant.com/driverloader/wlan/full/downloads-fc3-i686.php>



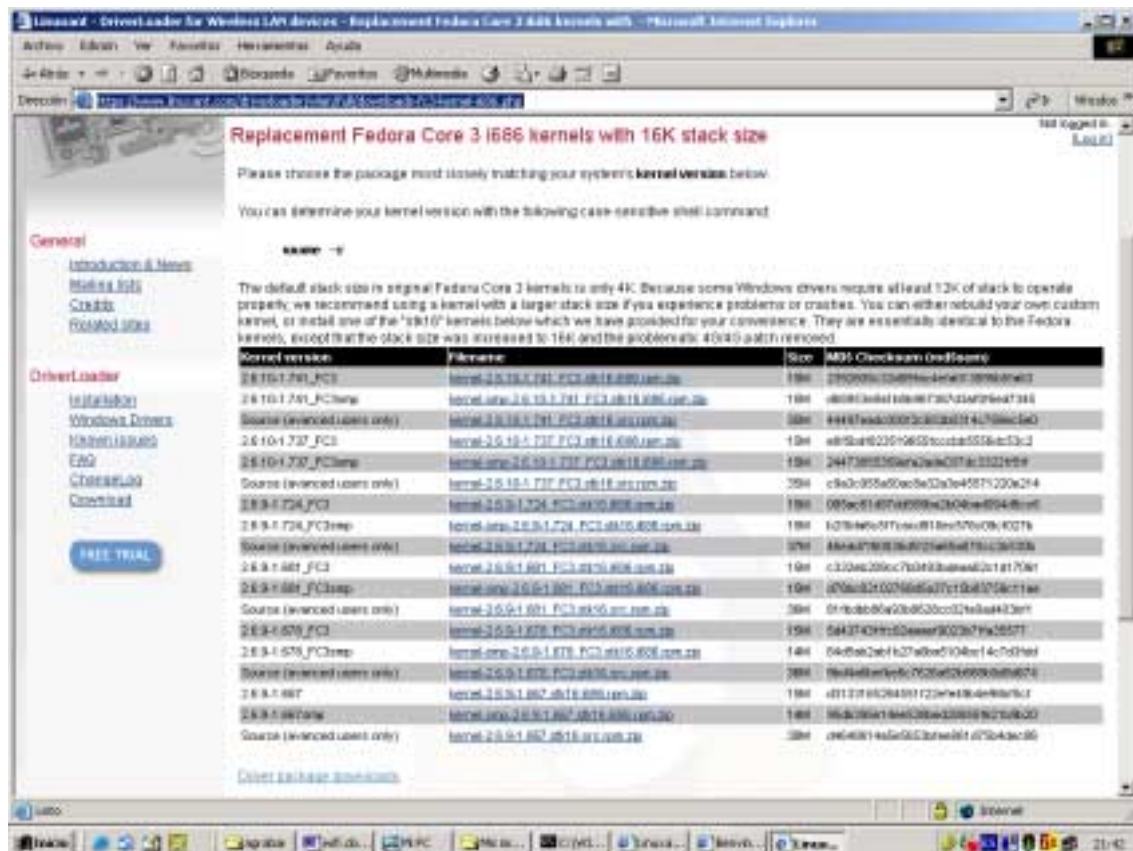
De toda esa lista, tengo que escoger la que corresponde a mi *kernel* concreto .. pero hay un problema...

Resulta que **Fedora** utiliza un **tamaño de 4K para la pila**, otras distribuciones usan un mínimo de 8K por defecto para el **Stack Size**... pero **Fedora** no... el problema es que tanto **direveloader** como **ndiswrapper**, muchos de los controladores de **Windows** **precisarán de al menos 12 K reservados en el stack size**... así que tendremos que recompilar el núcleo o parchearlo para que se pueda usar más espacio.... y ya puestos.. pues actualizaré al 2.6.10 ;)

La gente de Linuxant lo tienen todo previsto, puedes bajarte los núcleos ya compilados o sus fuentes con el parche para que los que usamos Fedora sólo tengamos que hacer un clic.... buenos que son ellos...

Así que, visitaré esta página, me descargaré el nuevo *kernel*, lo compilaré y seguiremos...

<https://www.linuxant.com/driverloader/wlan/full/downloads-fc3-kernel-i686.php>



Los *kernel .src.rpm* contienen las fuentes mientras que los *.rpm* sólo el paquete precompilado.. esto es como siempre, *stk16* simboliza que esos núcleos ya están preparados para modificar el **stack size a 16K (stk16)**

Por tanto, puedo escoger un paquete precompilado y listo para instalar

https://www.linuxant.com/driverloader/wlan/full/archive/fc3/kernel-2.6.10-1.741_FC3.stk16.i686.rpm.zip

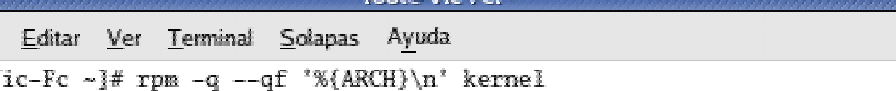
O el *rpm* con los fuentes:

https://www.linuxant.com/driverloader/wlan/full/archive/fc3/kernel-2.6.10-1.741_FC3.stk16.src.rpm.zip

Ahh.. también hay "otros" que pone *smp*... esos son para el soporte a multiprocesador, (*simetrical multiprocesor, smp*) yo no soy de los afortunados...

Un momento... compilando e instalando el nuevo núcleo... (calma, eh!!)

Bien... ya está ;) Repetiremos los pasos de antes...

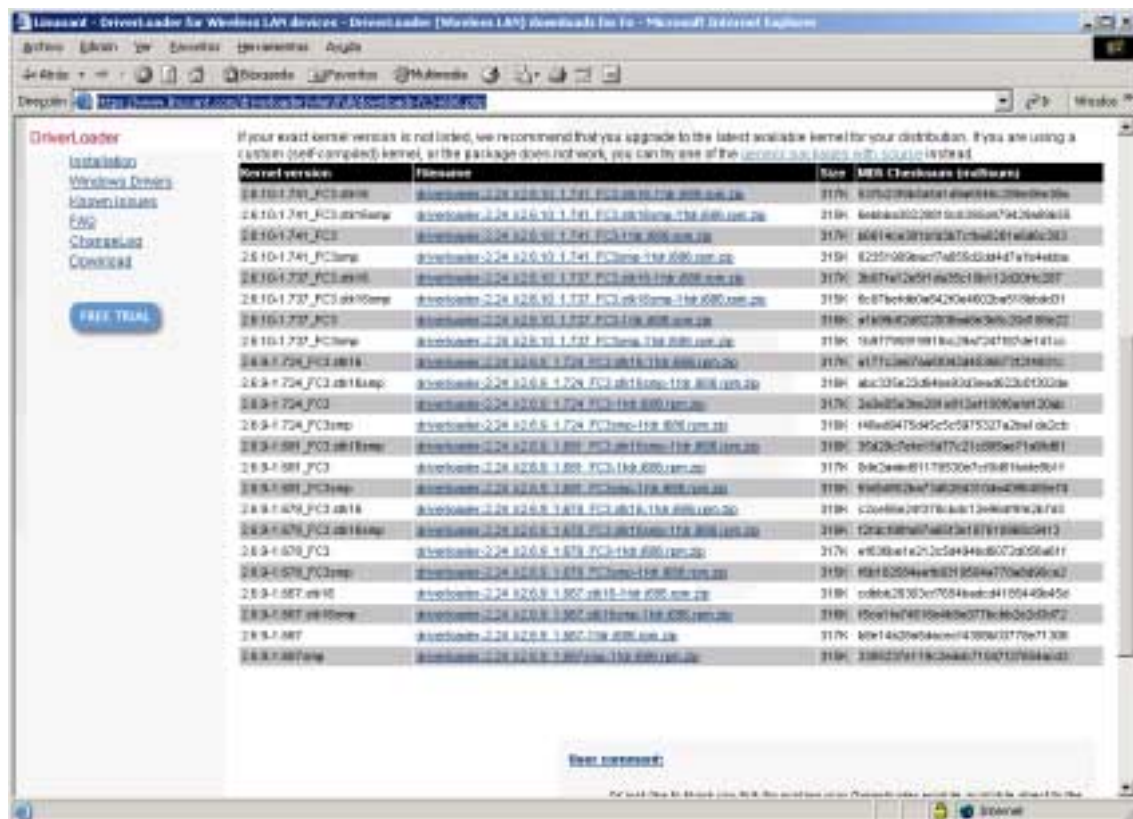


The screenshot shows a terminal window titled "root@Vic-Fc:~". The menu bar includes "Archivo", "Editar", "Ver", "Terminal", "Solapas", and "Ayuda". The terminal content is as follows:

```
[root@Vic-Fc ~]# rpm -q --qf '%{ARCH}\n' kernel
i686
[root@Vic-Fc ~]# uname -r
2.6.10-1.741_FC3.stk16
[root@Vic-Fc ~]# uname -m
i686
[root@Vic-Fc ~]#
```

Volvemos A navegar a **linuxant** directamente al link de antes:

<https://www.linuxant.com/driverloader/wlan/full/downloads-fc3-i686.php>



Y como actualicé el núcleo a la última versión disponible junto con el parche para poder usar más de 4k en el **size stack**, tengo que elegir el primero de la lista, que es el que se corresponde con la información que obtuve antes de **uname -r** y **uname -m**

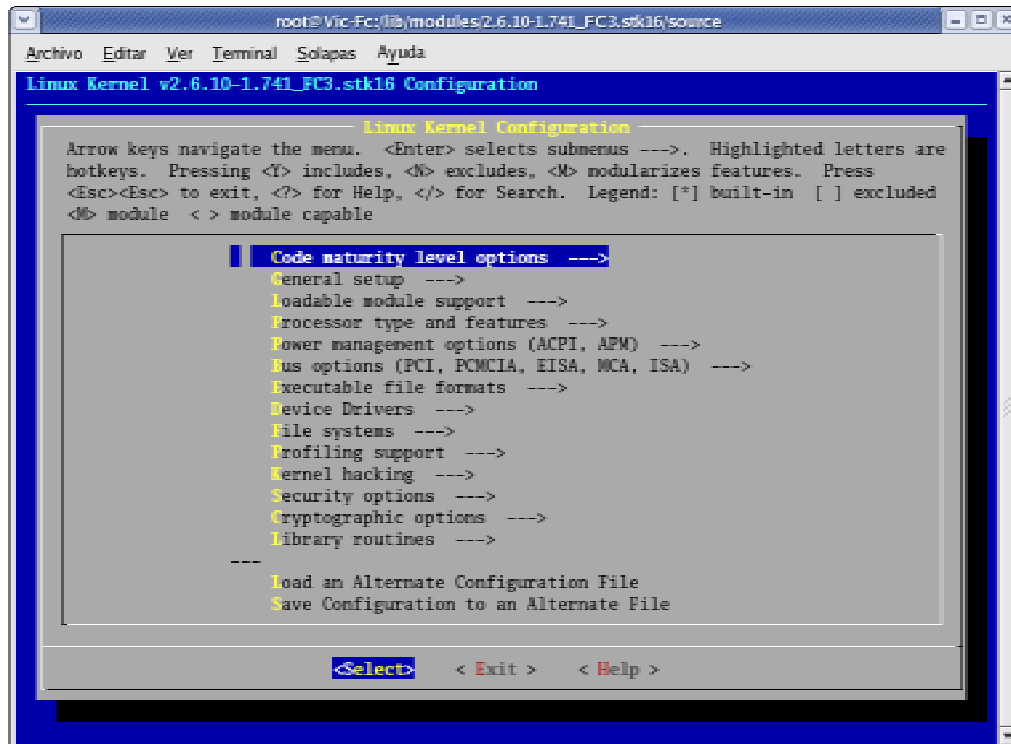
https://www.linuxant.com/driverloader/wlan/full/archive/driverloader-2.24/driverloader-2.24_k2.6.10_1.741_FC3.stk16-1fdr.i686.rpm.zip

Mientras se descarga, te cuento las opciones del *kernel* que son importantes para que todo esto funcione:

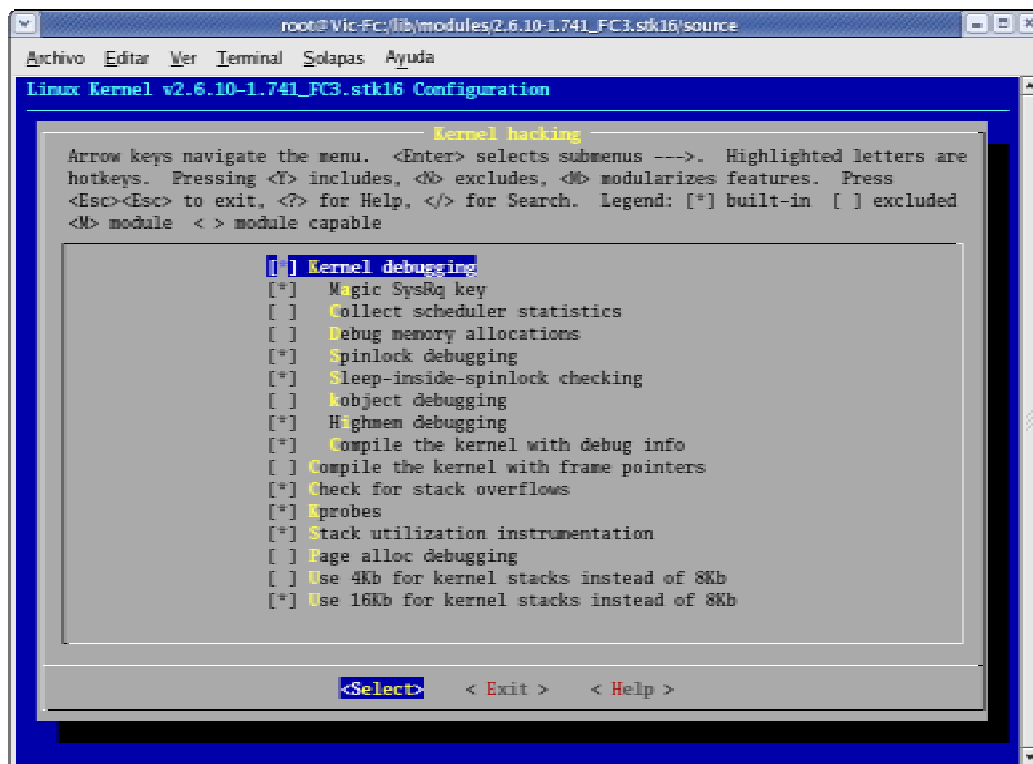
Hacemos **make menuconfig** desde el directorio en donde esté las fuentes del *kernel*... eso dependerá de la *distro*, por lo general en **/usr/src/linux-2.X** (verifica la ubicación correcta)

Bueno, también puede que anden por `/lib/modules/2.6.10-aquí-la-revisión/source` si seguiste los pasos para *Fedora Core 3* en la instalación `.rpm...` o los buscas... :P

El caso es que tras la orden ***make menuconfig*** aparecerá algo así:



La opción necesaria para aumentar el tamaño del ***stacksize*** está dentro de ***Kernel hacking***...

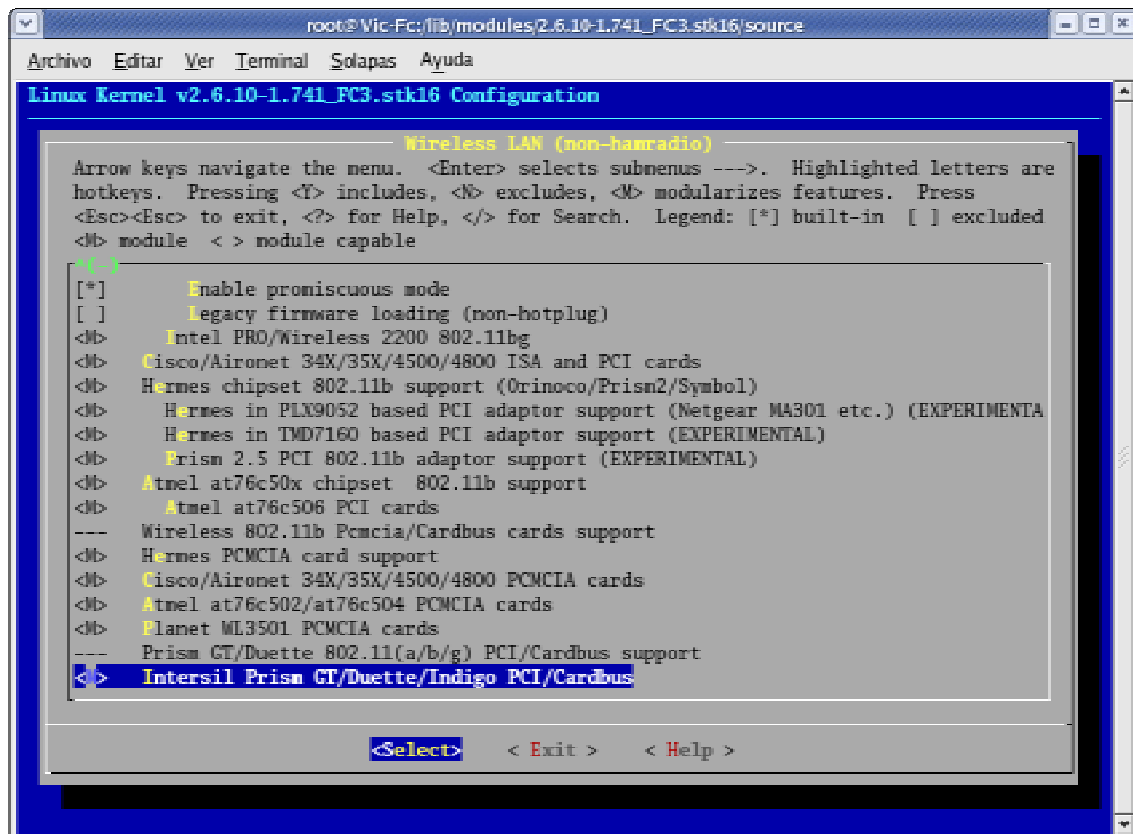


Como ves, las últimas opciones permiten seleccionar el tamaño del *stack*, se eligió **16K**

Bueno, **no se te olvide seleccionar los módulos necesarios para dispositivos pcmcia, cardbus, usb, bluetooth**, etc... no estaría de más tenerlos ya cargados por si acaso... la mayoría de los *kernels* modernos ya lo hacen, ya tienen las opciones seleccionadas por defecto, pero es bueno que se compruebe:

Desde la pantalla principal de configuración:

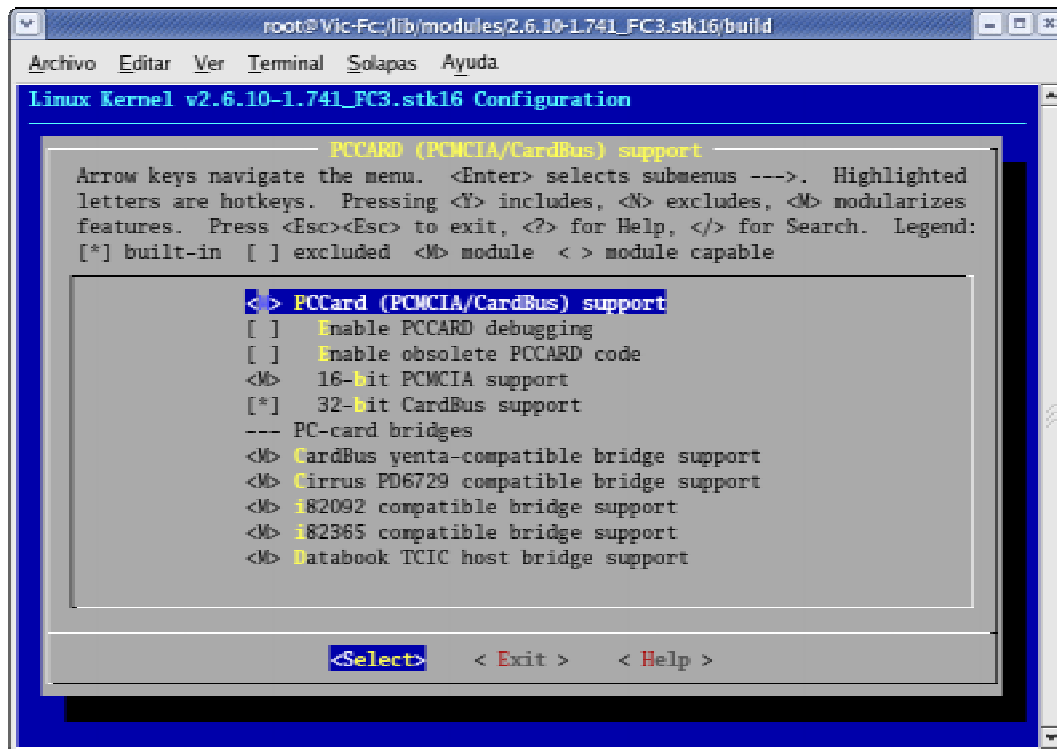
Device Drivers – NetWorking support – Wireless LAN (non-hamradio)



Si te fijas, para tarjetas *Prism* (y muchas otras) el núcleo de *LiNux* ya “*viene preparado*”, cualquiera de esas tarjetas con esos *chipsets* funcionarán “*sin más*”.

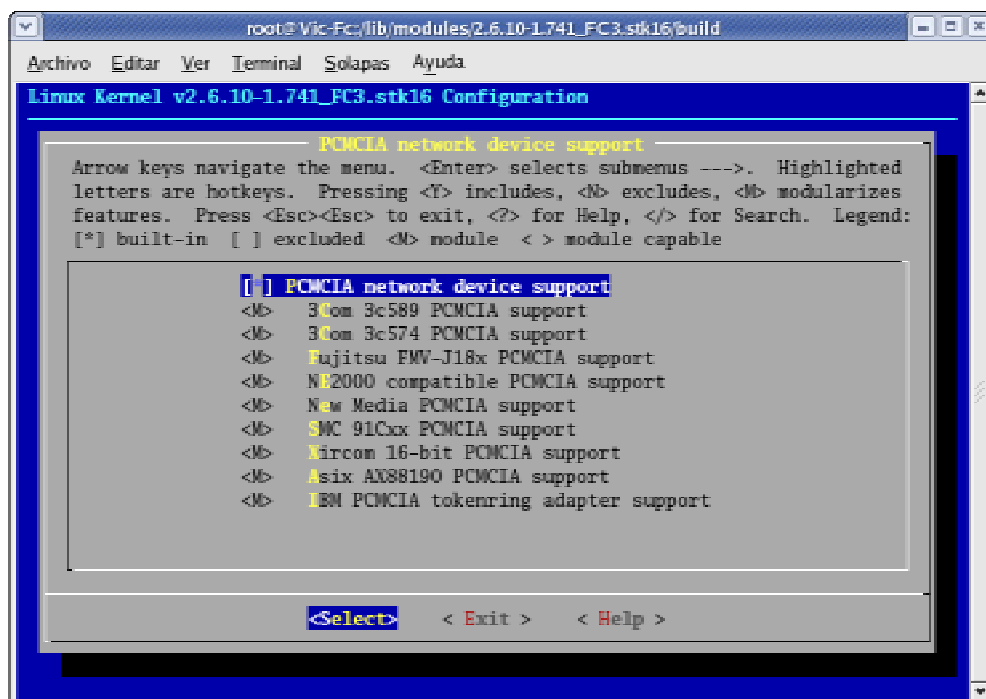
También es interesante (si se trata de un portátil) dar un vistazo a la configuración de los dispositivos *pcmcia* y *cardbus*, desde la pantalla principal de configuración:

Bus Options (PCI, PCMCIA....) – PCCARD (PCMCIA/CARDBUS)



Y también hay que mirar en:

Device Drivers – Networking Supports – PCMCIA network device support



Ninguno de nuestras opciones da soporte a la tarjeta *USRobotics* que queremos configurar, pero el *kernel* ha de soportar un *stack size* de 16K, los módulos para usar *pcmcia/cardbus* habilitados (o en su caso, soporte para *usb*)

Si modificaste el núcleo, tendrás que compilarlo... con la versión 2.6 se simplifica mucho las cosas, bastará que hagas:

```
make  
make modules_install  
make clean  
make install
```

Luego verifica que en el gestor de arranque existan las entradas al antiguo *kernel* y al nuevo (no vaya a ser que no se haya hecho bien) y reinicia...

Bueno, hay mas cosas que hablar del *kernel* y su compilación... para ello, ya sabes, a nuestra **zona de LINUX del foro**.

Seguimos con linuxant...

Tras haber descargado la versión apropiada de **driverloader** que más se ajusta a nuestra distribución de *LINUX* y versión del núcleo, descomprimos el archivo y obtendremos un *.rpm* para instalar:

No estaría de más que des un paseo por la documentación de la página oficial para ver los métodos de instalación: <https://www.linuxant.com/driverloader/wlan/install.php>

Aquí me voy a referir a una instalación desde un *.rpm*, en el link anterior tienes los pasos para hacer lo propio desde un *tarball* o para paquetes *debian*.

Si ya se habían instalado cualquier otra versión de **driverloader**, antes de empezar hay que eliminarlas:

```
rpm -e driverloader si se instalaron mediante rpm
```

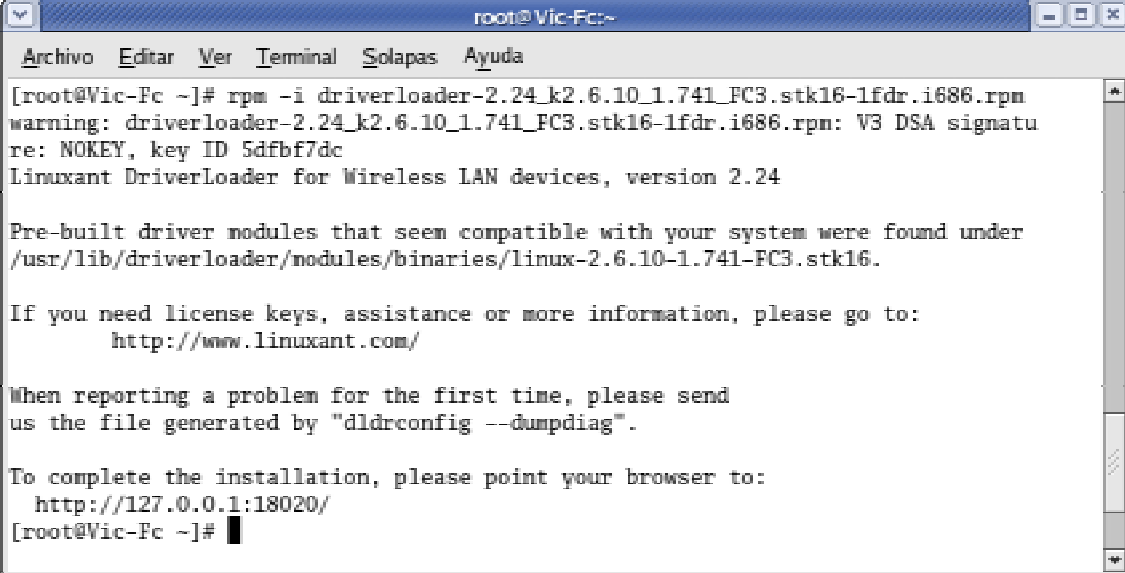
```
make uninstall si se usó la compilación des de las fuentes
```

Como "se supone" que es nuestra primera vez... no será necesario nada de lo anterior.

Para instalarlo desde el .rpm:

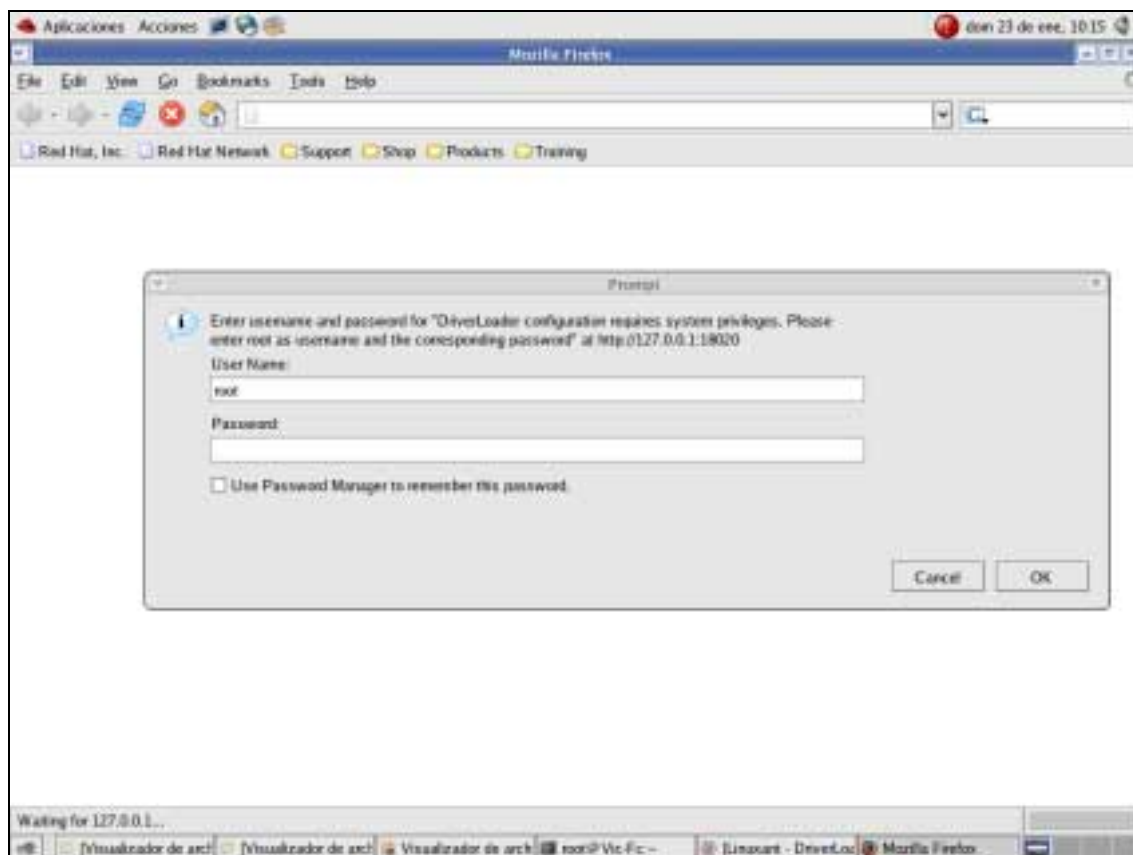
```
rpm -i driverloader-2.24_k2.6.10_1.741_FC3.stk16-1fdr.i686.rpm
```

Y verás esto:

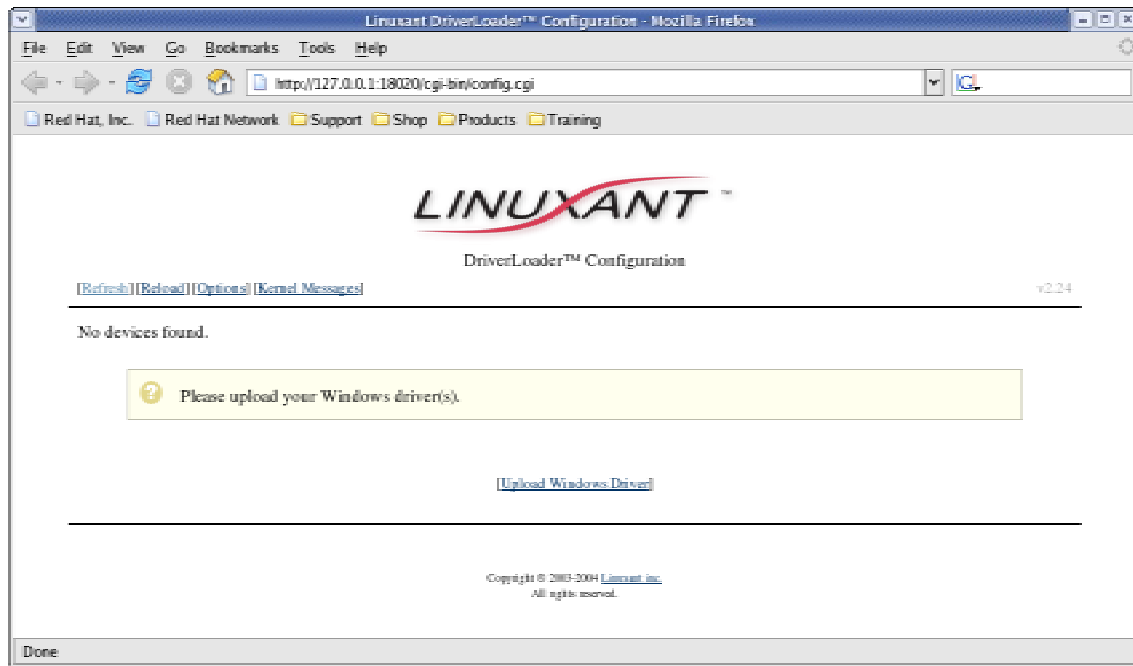


```
root@Vic-Fc:~  
Archivo Editar Ver Terminal Solapas Ayuda  
[root@Vic-Fc ~]# rpm -i driverloader-2.24_k2.6.10_1.741_FC3.stk16-1fdr.i686.rpm  
warning: driverloader-2.24_k2.6.10_1.741_FC3.stk16-1fdr.i686.rpm: V3 DSA signatu  
re: NOKEY, key ID 5dfbf7dc  
Linuxant DriverLoader for Wireless LAN devices, version 2.24  
  
Pre-built driver modules that seem compatible with your system were found under  
/usr/lib/driverloader/modules/binaries/linux-2.6.10-1.741-FC3.stk16.  
  
If you need license keys, assistance or more information, please go to:  
http://www.linuxant.com/  
  
When reporting a problem for the first time, please send  
us the file generated by "dldrconfig --dumpdiag".  
  
To complete the installation, please point your browser to:  
http://127.0.0.1:18020/  
[root@Vic-Fc ~]#
```

Ahora sólo nos queda abrir el navegador y escribir: <http://127.0.0.1:18020> y usuario **root** y su **contraseña**



Como es lógico, no tenemos ningún dispositivo instalado y nos “*recomienda*” (obliga) a “*subir*” un *driver* para *Windows*...



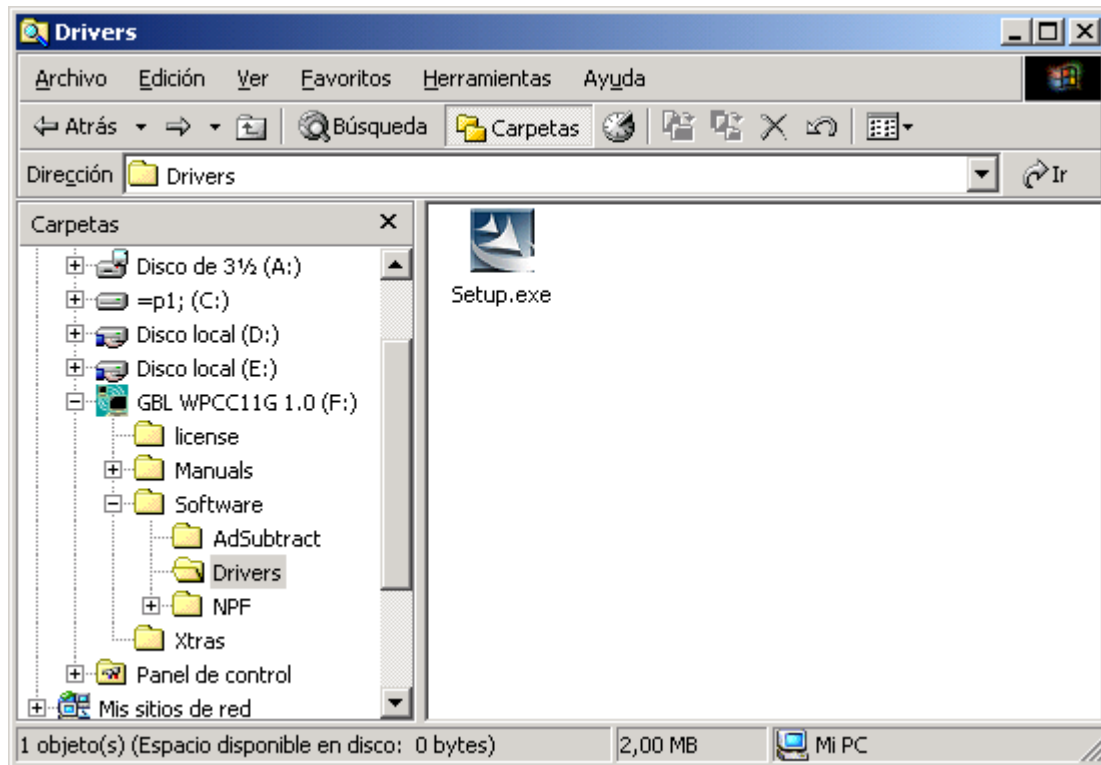
¿Dónde lo consigo?

Pues normalmente **en el CDRom** que nos suministran cuando compramos la tarjeta, **en la web del fabricante**, en la propia **página de linuxant**, vamos donde sea... si no disponemos de un *driver* válido *Windows* será inútil seguir.

Puede ocurrir que esos *drivers* para *Windows* estén en un ejecutable, si es así no quedará otro remedio que instalarlos en un *Windows* “*de verdad*” y luego copiarnos los archivos necesarios.

Es mi caso... (aunque en **linuxant** dispongo de ellos, voy a usar los que me dieron con el CD junto con la tarjeta cuando la compré)

Inicio un equipo *Windows* y lo primero que hago es ver qué hay en el CD



Lo que me esperaba... es un instalador...

Bueno pues lo ejecuto según las instrucciones de rigor, *dígase, siguiente-siguiente...finalizar :P*

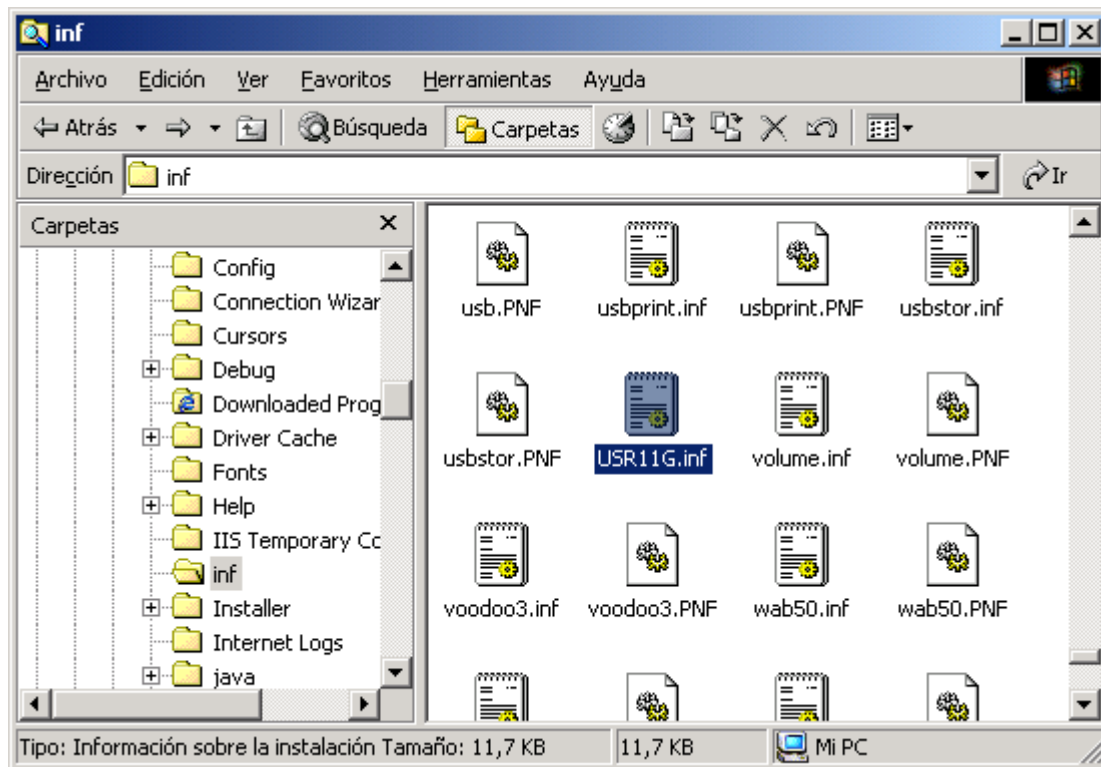
¿Qué archivos son necesarios para linuxant en LINUX?

Pues el **.inf** y alguno más....

Si pero .inf hay muchos y los otros ¿cuáles son? Seguro que estarán diseminados por el disco de Windows...

Bueno, el **.inf** , estará en **c:\wiint\inf** o en **c:\windows\inf** dependiendo de la carpeta que elegiste para instalar *Windows* en su momento...

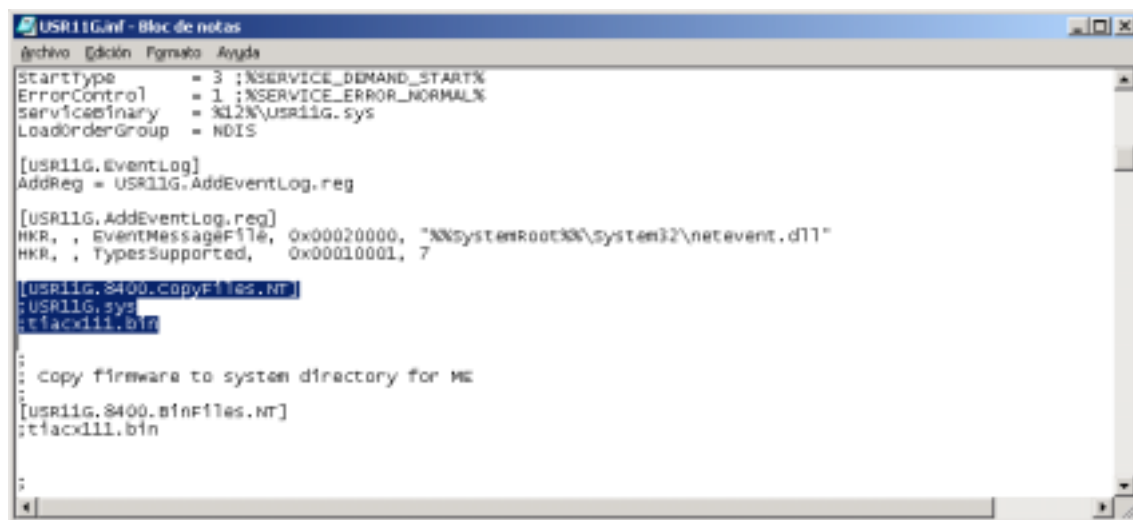
Y el archivo.... pues seguro que empieza por usr (de US Robotics, ¿no?)



Exacto!!!! Ahí está....

Bueno, **pues ya tenemos uno...** lo copiamos a un disquete o a un recurso compartido desde el que podamos acceder desde *LiNux*.

Ahora **abrimos el archivo usr11g.inf** con el bloc de notas o cualquier otro editor de texto plano y buscamos la sección que habla de los archivos necesarios para instalar el controlador, concretamente esta:



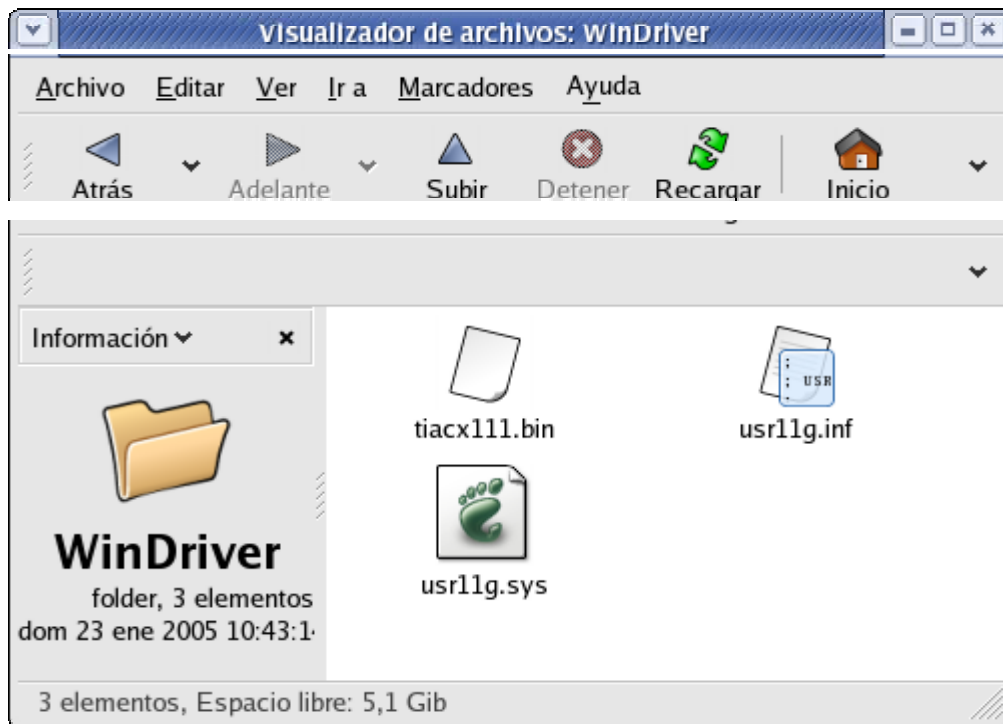
Por si no se ve claramente en la captura de pantalla... es:

```
[USR11G.8400.CopyFiles.NT]
;USR11G.sys
;tiacx111.bin
```

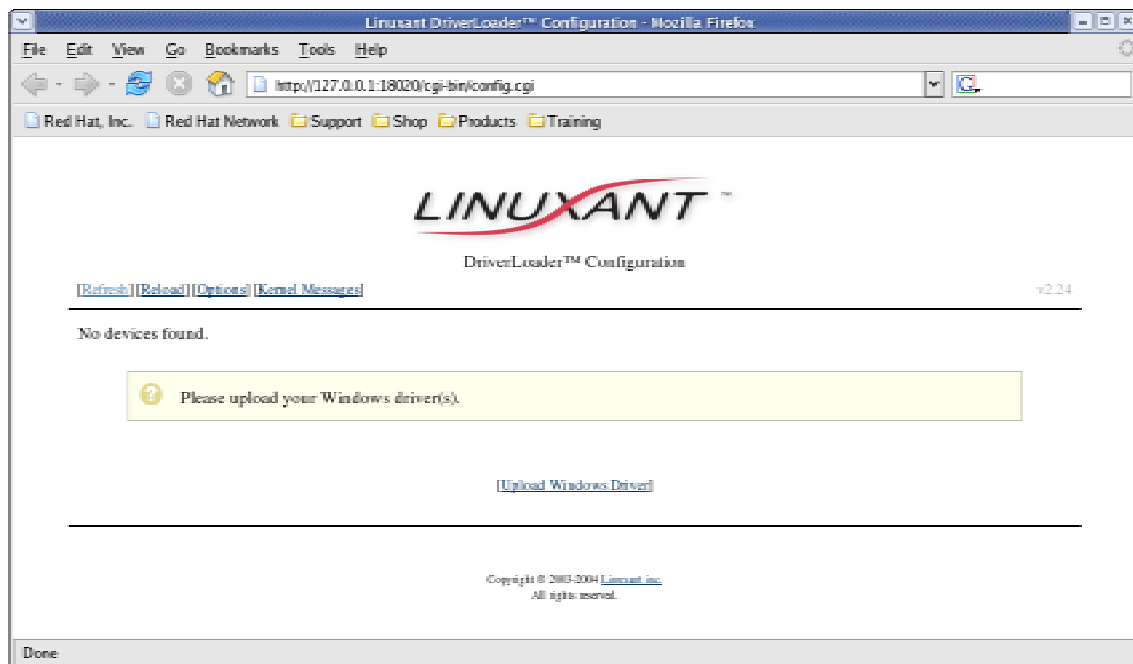
Es decir... **también necesitaremos los archivos *usr11g.sys* y *tiacx111.bin***

Los buscamos (**que seguro que están en *C:\winnt\system32\drivers***) y los copiamos al mismo lugar donde lo hicimos anteriormente con el ****.inf***

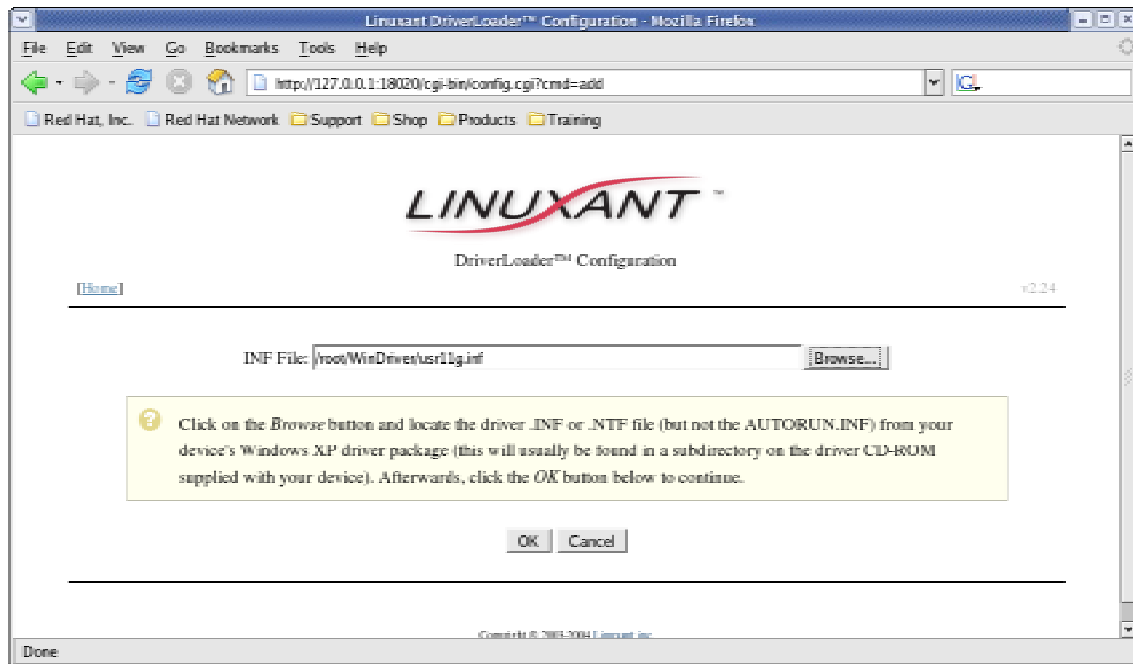
Ahora volvemos a *LiNux*.... y abrimos el medio donde colocamos estos archivos y los copiamos a cualquier directorio (yo elegí ***/root/WinDriver***)



Regresamos a la configuración de ***linuxant*** y ***driveloader***... nos habíamos quedado aquí:

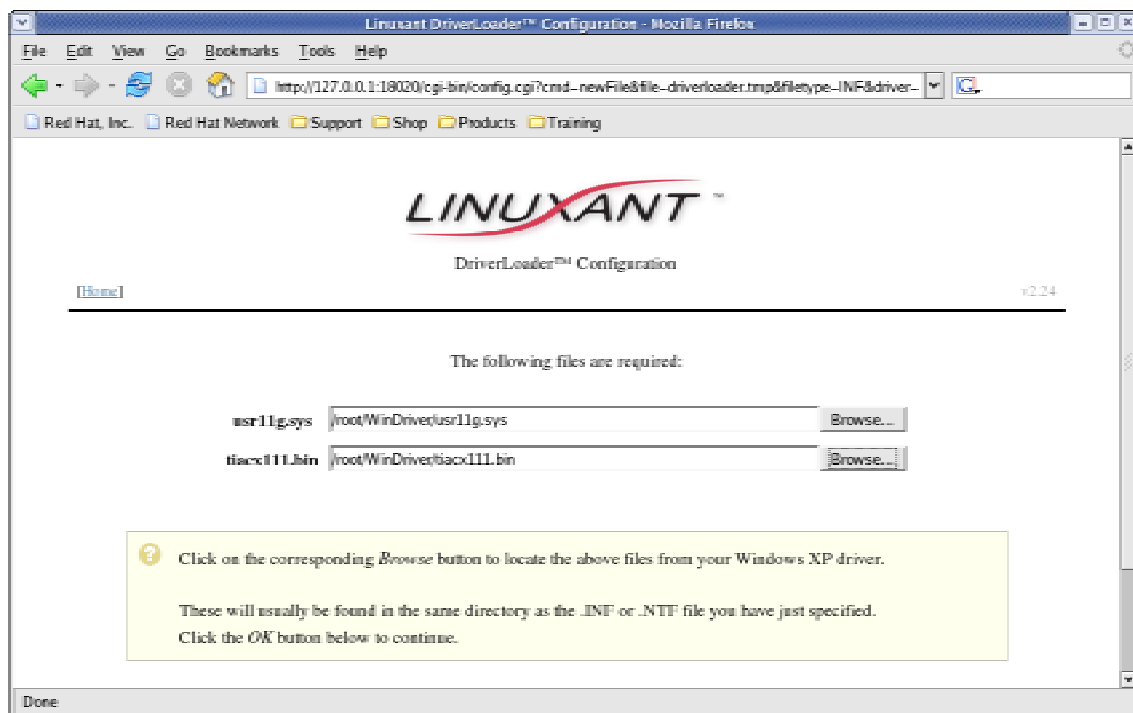


Ahora es cuando pinchamos en **Upload Windows Driver**, y nos pedirá la ruta para encontrar el archivo **.inf**, escribiré **/root/WinDriver/usr11g.inf** o mediante el botón **Browse** se accede a dicho archivo



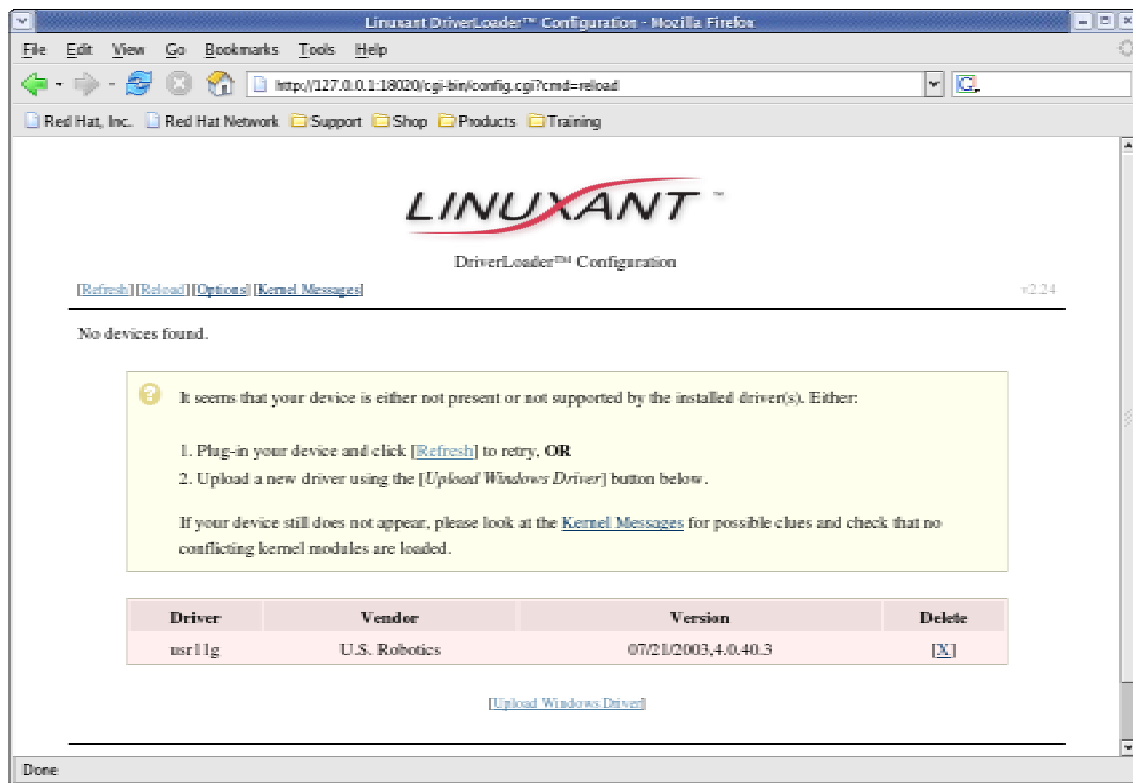
Y pulsamos en **OK**

Como ya esperábamos y tras leer el contenido del archivo **.inf**, nos pide el **.sys** y **.bin** que se necesitan... hacemos lo mismo de antes y le ponemos la ruta correcta y los nombres de archivo que necesita:



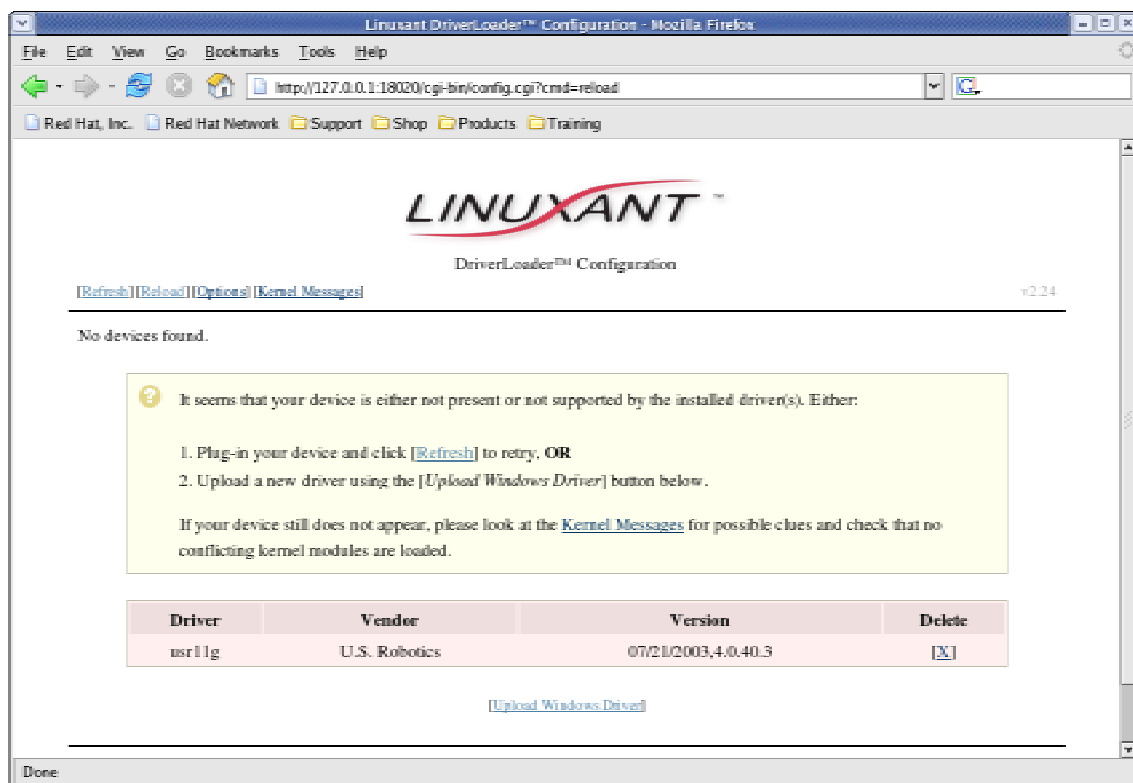
Después... de nuevo **OK**

Parece que fue todo bien.....



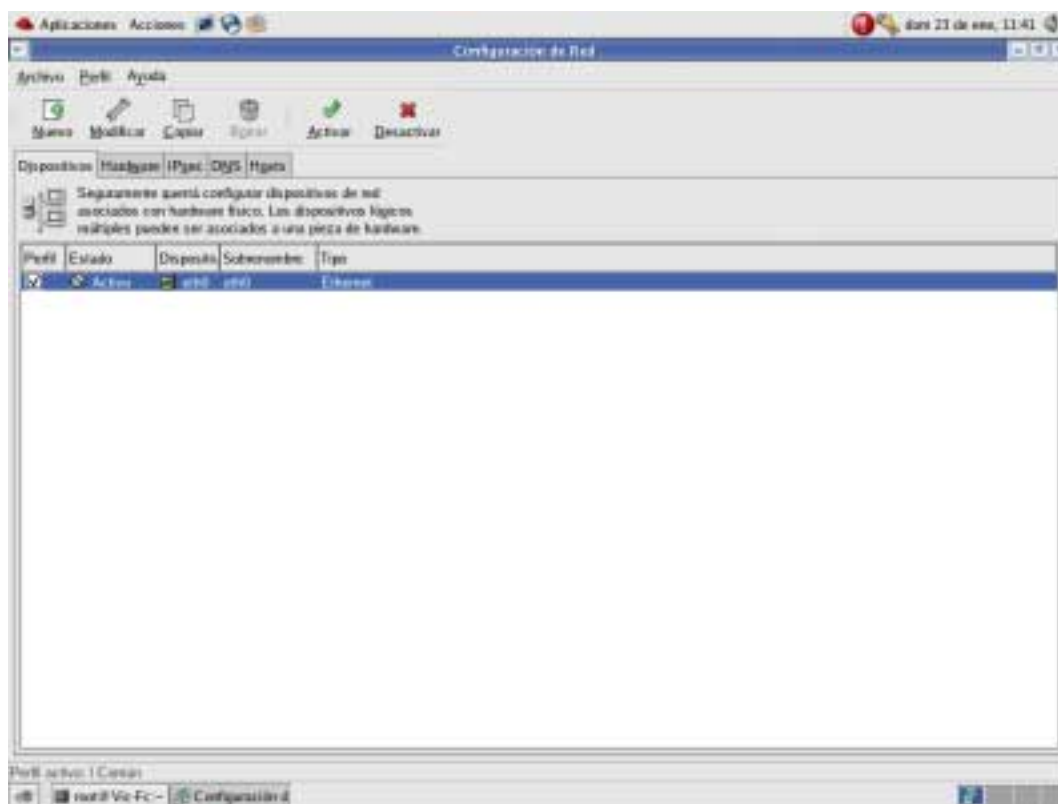
Ahora vamos a comprobar... (si se trata de una tarjeta *pcmcia* no se te olvide pincharla :P)

Ejecutamos **iwconfig** (esto lo veremos mejor cuando nos toque explicar las **linux-extension-wireless**)

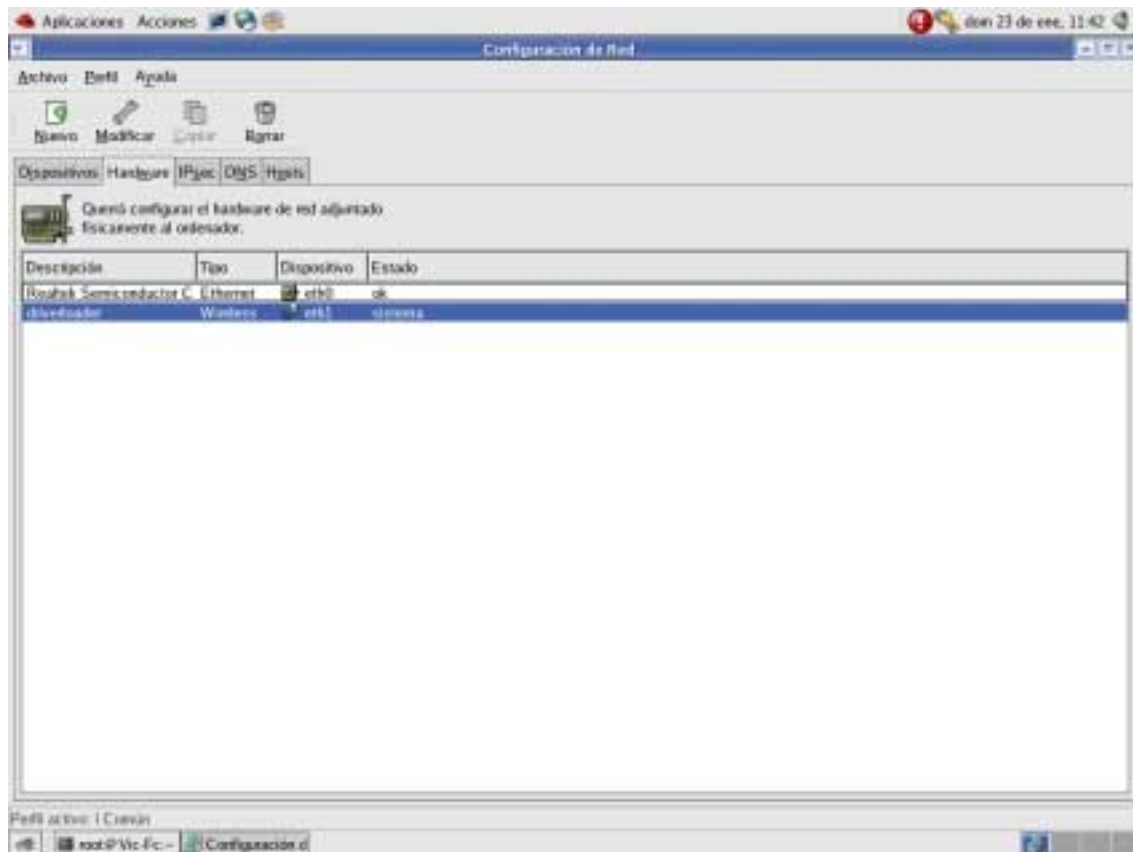


Ya vemos que existe una... **eth1** :P

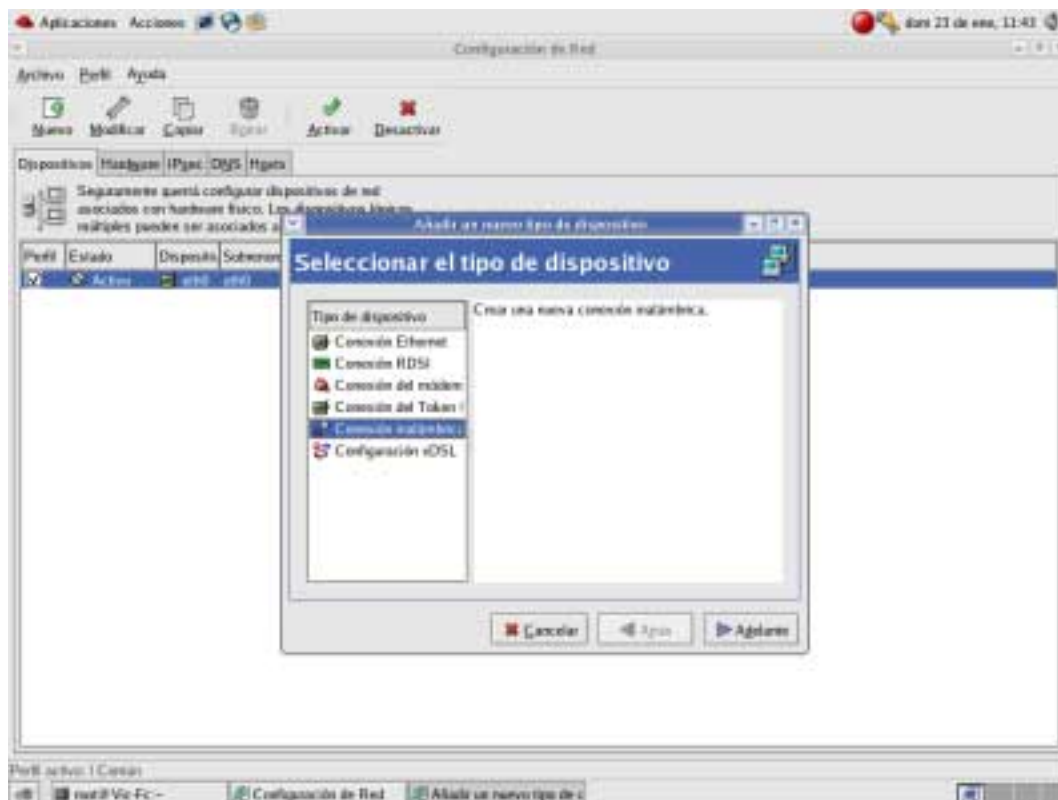
Pero antes de poderla usar... vamos a la configuración de red:



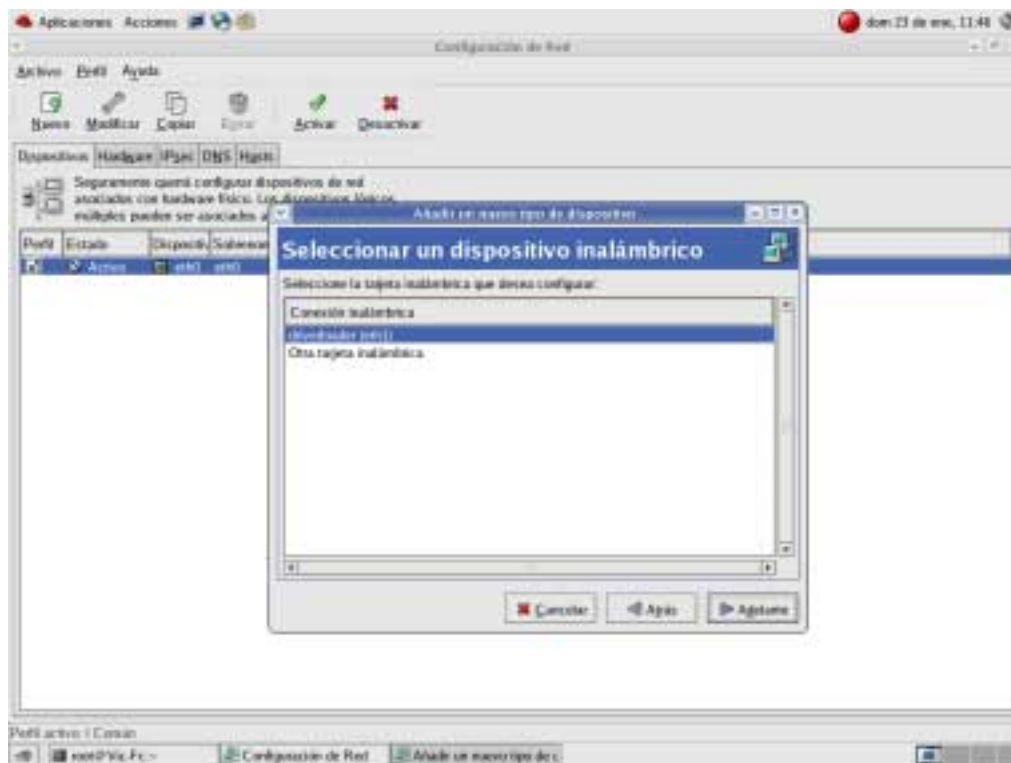
Todavía no está... comprobemos si en la **ficha de Hardware** existe:



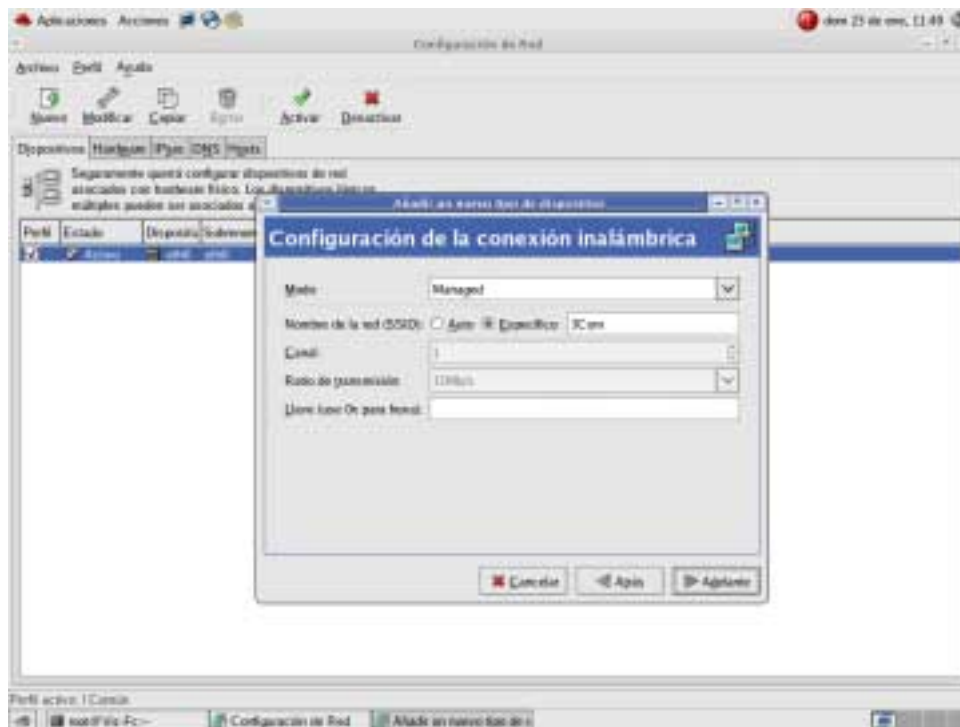
Vale, pues volvemos a la **ficha de Dispositivos** y Pinchamos en **Nuevo**, seleccionamos **Conexión inalámbrica** y **Adelante...**



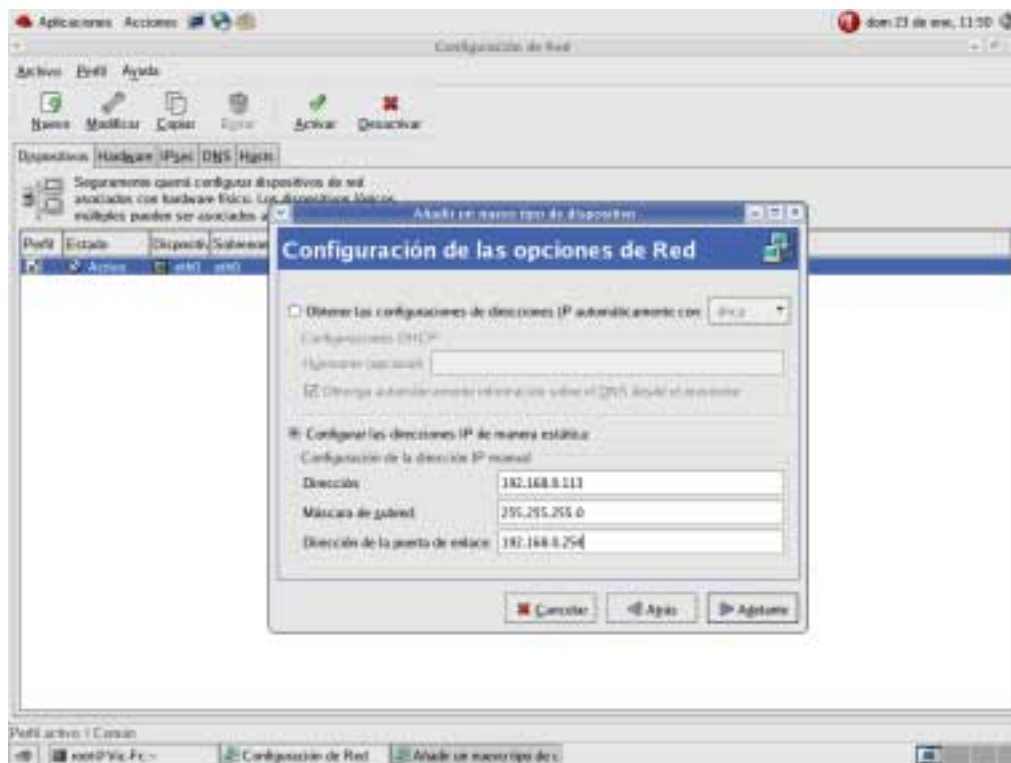
En dispositivo inalámbrico seleccionamos **driveloader (eh1)** y **Adelante**



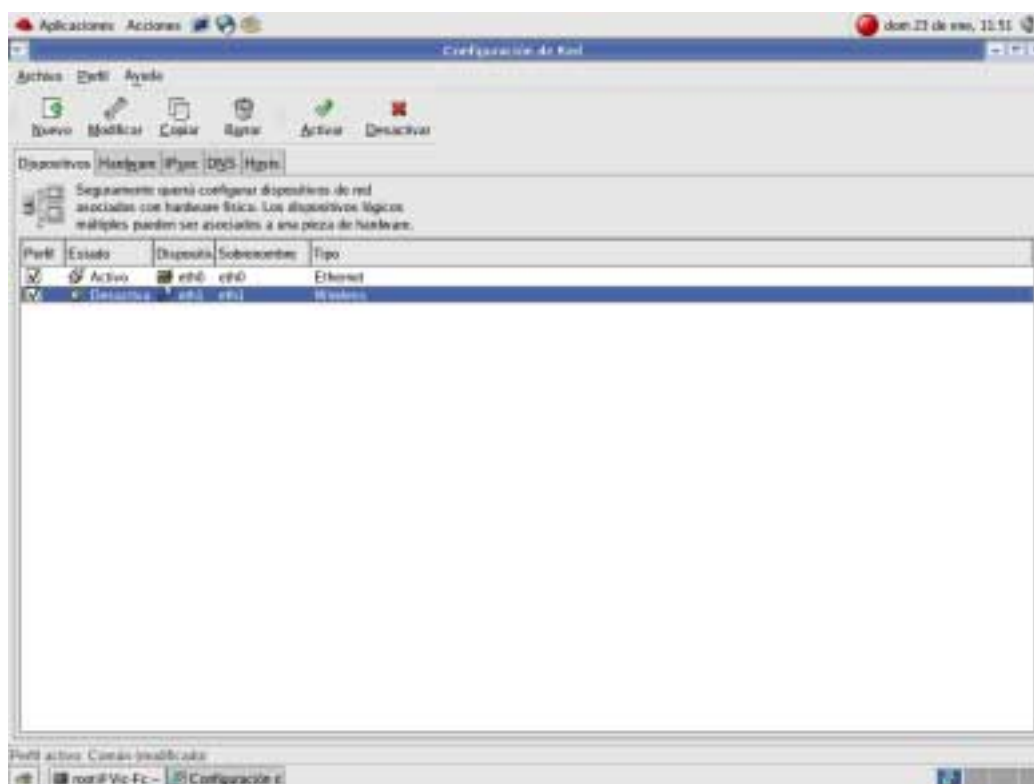
En la próxima pantalla podremos seleccionar el modo de tarjeta (**managed** para unirse a la red, **ad-hoc** para conexiones uno-a-uno, el nombre de la red, etc...) le damos los valores que correspondan o los dejamos "**tal cual**", luego se podrá cambiar...



Ahora le configuramos los valores de IP necesarios, como cualquier otra...



Aplicamos y finalizamos... ahora veremos esto:



Activamos, aplicamos, guardamos y cerramos la configuración de red.

Probamos un **ping** a la puerta de enlace desde el dispositivo eth1 (que es la *wireless*)

ping 192.168.0.254 -I eth1

```
root@Vic-Fc:~  
[root@Vic-Fc ~]# ping 192.168.0.254 -I eth1  
PING 192.168.0.254 (192.168.0.254) from 192.168.0.113 eth1: 56(84) bytes of data  
.  
64 bytes from 192.168.0.254: icmp_seq=0 ttl=64 time=2.16 ms  
64 bytes from 192.168.0.254: icmp_seq=1 ttl=64 time=2.08 ms  
64 bytes from 192.168.0.254: icmp_seq=2 ttl=64 time=2.03 ms  
64 bytes from 192.168.0.254: icmp_seq=3 ttl=64 time=2.12 ms  
  
--- 192.168.0.254 ping statistics ---  
4 packets transmitted, 4 received, 0% packet loss, time 3002ms  
rtt min/avg/max/mdev = 2.036/2.103/2.166/0.047 ms, pipe 2  
[root@Vic-Fc ~]#
```

OK!!!! Ya tenemos red ;)

Si le damos un vistazo a **ifconfig**... veremos nuestras tarjetas de red... y entre ellas la "insoportada" por **LINUX**:P

```
root@Vic-Fc:~  
[root@Vic-Fc ~]# ifconfig  
eth0      Link encap:Ethernet  HWaddr 00:00:39:C1:6A:C2  
          inet addr:192.168.0.222 Bcast:192.168.0.255 Mask:255.255.255.0  
          inet6 addr: fe80::200:39ff:fecl:6ac2/64 Scope:Link  
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1  
          RX packets:23 errors:0 dropped:0 overruns:0 frame:0  
          TX packets:54 errors:0 dropped:0 overruns:0 carrier:0  
          collisions:0 txqueuelen:1000  
          RX bytes:2473 (2.4 KiB)  TX bytes:4647 (4.5 KiB)  
          Interrupt:11 Base address:0xed00  
  
eth1      Link encap:Ethernet  HWaddr 00:C0:49:52:DF:87  
          inet addr:192.168.0.113 Bcast:192.168.0.255 Mask:255.255.255.0  
          inet6 addr: fe80::2c0:49ff:fe52:df87/64 Scope:Link  
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1  
          RX packets:21 errors:0 dropped:0 overruns:0 frame:0  
          TX packets:23 errors:0 dropped:0 overruns:0 carrier:0  
          collisions:0 txqueuelen:1000  
          RX bytes:1612 (1.5 KiB)  TX bytes:1718 (1.6 KiB)  
  
lo        Link encap:Local Loopback  
          inet addr:127.0.0.1 Mask:255.0.0.0  
          inet6 addr: ::1/128 Scope:Host  
          UP LOOPBACK RUNNING  MTU:16436  Metric:1  
          RX packets:1856 errors:0 dropped:0 overruns:0 frame:0  
          TX packets:1856 errors:0 dropped:0 overruns:0 carrier:0  
          collisions:0 txqueuelen:0  
          RX bytes:3261776 (3.1 MiB)  TX bytes:3261776 (3.1 MiB)  
  
[root@Vic-Fc ~]#
```


Si deseamos **desactivar el interface web** de configuración de *driverloader*, bastará que hagamos esto:

```
dldrconfig --webconf=127.0.0.1:18020
```

Si queremos **desactivar el driver completo**:

```
dldrstop
```

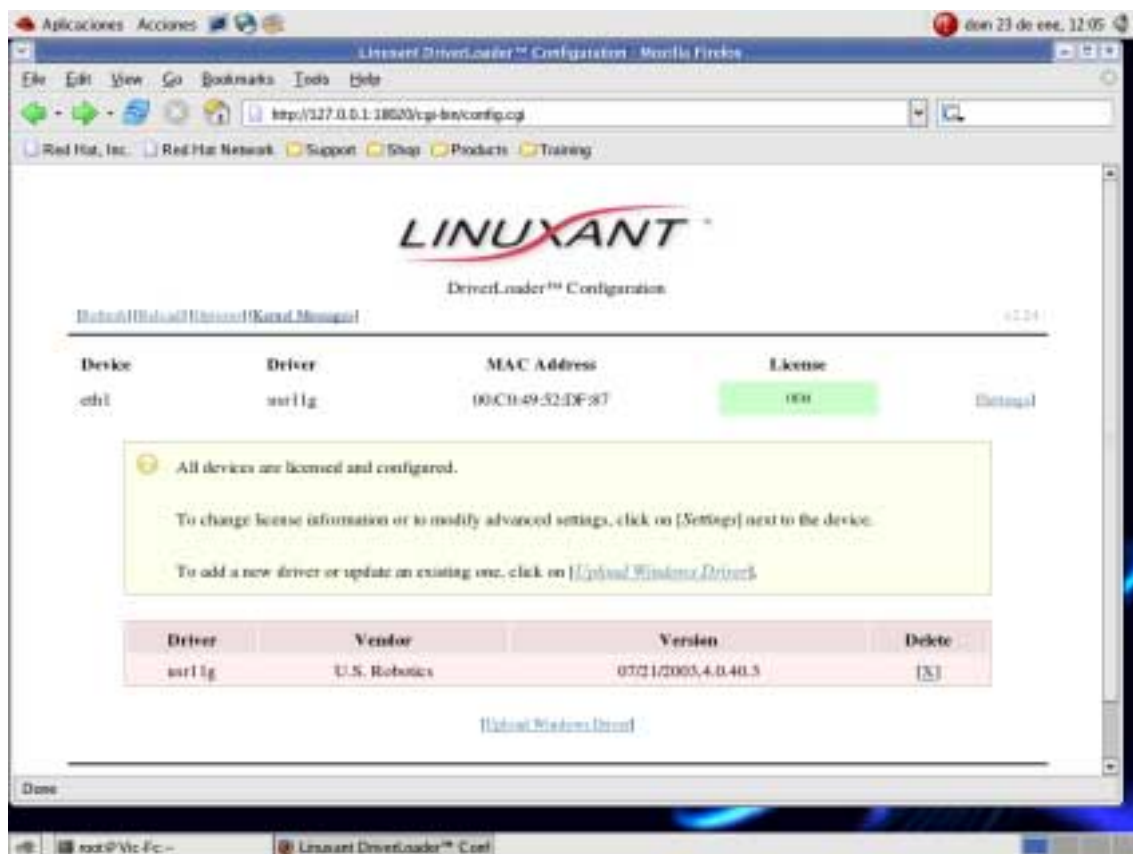
O si queremos **eliminarlo todo....**

```
rpm -e driverloader
```

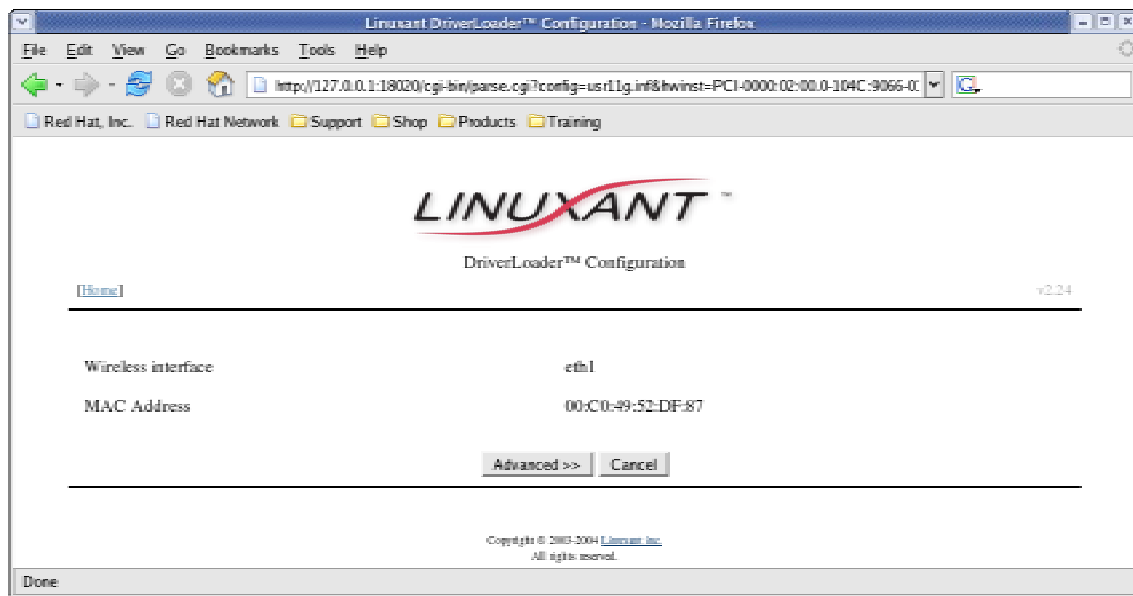
También podemos reconfigurar el *driver* y el dispositivo mediante el **interface web** que instalamos:

<http://127.0.0.1:18020>

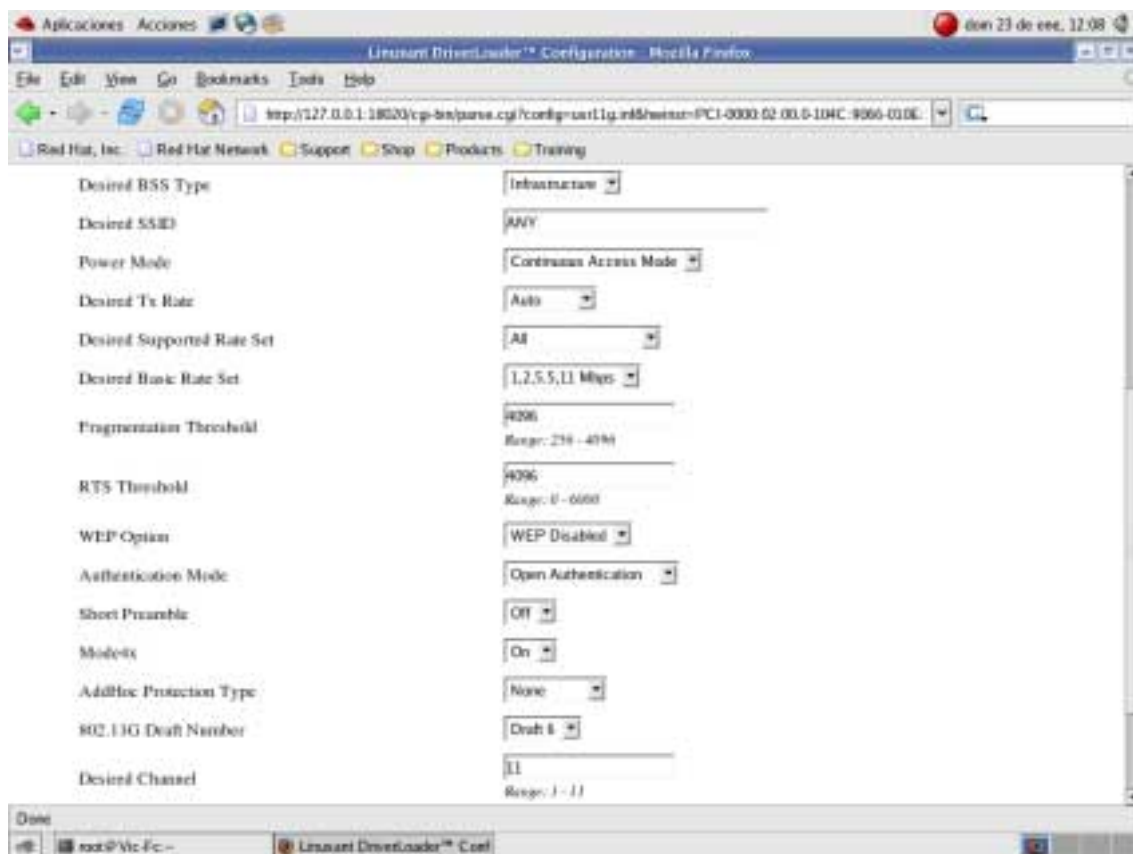
De nuevo introducimos **root** y la **contraseña...** y aparecerá esto:



Observa que aparece un nuevo enlace que pone [\[Settings\]](#) a la derecha del dispositivo *eth1*, pinchamos:



Nos informa de la MAC que usa... pinchamos en **Advanced....**



Desde aquí **podemos reconfigurar los parámetros del controlador....** si has seguido el resto de post, no tendrás problemas en entender la mayor parte de las opciones:

Desired BSS Type: *Infraestructura* ó *Ad-Hoc* para asociarnos a la red o para comunicación uno a uno

Dessired SSID: *ANY* para asociarse con cualquier dispositivo o le escribimos el *SSID* de nuestra red...

Power Mode si nuestra tarjeta soporta el modo de administración de energía.

Desired TX Rate: Velocidad de transmisión, entre 1 y 54 Mbps al ser una 802.11g

Desired Supported Rate Set: Lo dejamos en *all* (todos) o elegimos entre las opciones la velocidad a la que podemos conectarnos y participar de la red.

Desired Basic Rate Set: lo mejor es dejarlo en 1,2,5.5,11 o si la señal es buena y la tarjeta lo soporta, podemos seleccionar velocidades por encima de.... (las opciones que aparecen)

Fragmentation Threshold: Es el umbral de fragmentación... recuerda lo que dijimos del *MTU* y las tramas 802.11, son mayores que las *Ethernet* cableadas, puedes dejarlo a 4096 que es el máximo o poner reducirla a la *MTU* del cable.

RTS Threshold: Lo mismo pero para las tramas de control y administración

WEP Option: Eso, si activamos el cifrado *WEP* o no....

Authentication Mode: Modos de autenticación, abierta, mediante claves compartidas o la posibilidad de configurarla automáticamente.

Short Preamble: nos permite "*ahorrar*" ancho de banda en redes *wifi* con mucho tráfico, de ese modo las tramas serán más pequeñas

Mode-4x: Aunque está en *on*, mejor ponerlo en *off* si la tarjeta no funciona puesto que es el modo de tasa doblada y no compatible con el estándar 802.11

AddHoc Protection Type: Por si utilizamos algún tipo de protección en las comunicaciones uno a uno

Desired Channel: El número de canal que usará ...

De **la opción 802.11g Draft Number...** ni idea :P, lo buscaré pero ahora no sé que quiere decir con eso.

*Bien, hasta aquí la forma de configurar esta tarjeta con **driveloader**, te repito... hemos conseguido hacerla funcionar, pero no podremos usarla con casi ninguna de las herramientas de monitorización, los controladores NDIS para Windows no lo permitirán. **Pero funciona :P***

Ndiswrapper

La instalación de *drivers* de *Windows* bajo *Linux* con **ndiswrapper** es mucho más sencilla, entre otras cosas porque ya hemos explicado con detalle la mayor parte de las cosas que hay que hacer.

El proceso es muy similar:

- Consultar si nuestra tarjeta será soportada
- Bajarse ndiswrapper,
- Disponer de los drivers de windows
- Verificar que se disponen de 16K para el stacks size,
- Configurar la red
- Y a disfrutar :P

Es más o menos lo mismo, por eso no voy a repetir de nuevo las pantallas, sólo incluiré las líneas de comando necesarias y los enlaces para que todo funcione

Comprobar que nuestra tarjeta y chipset están soportados:

<http://ndiswrapper.sourceforge.net/phpwiki/index.php/List>

Bajamos la herramienta.

Por ejemplo de aquí:

http://prdownloads.sourceforge.net/ndiswrapper/ndiswrapper-0.12.tar.gz?use_mirror=unc

Descomprimirlo e instalarlo en el directorio que queramos.

Y de paso pegamos un vistazo al método de instalación:

<http://ndiswrapper.sourceforge.net/phpwiki/index.php?Installation>

Como suponemos que ya tenemos el núcleo compilado con soporte de los módulos necesarios y para el caso especial de *Fedora* el parche para aumentar el tamaño del *stack size* a 16K, podemos iniciar sin problemas la compilación.

Recuerda que también debes disponer de los *drivers* de *Windows* para la tarjeta en cuestión.

Nos situamos en el directorio donde descomprimos **ndiswrapper**.

```
cd /root/ndiswrapper-0.12
```

```
make && make install
```

Si todo fue bien (no hubo errores) accedemos al lugar donde están los *drivers* de *Windows* y ejecutamos:

```
ndiswrapper -i archivo.inf
```

Donde *archivo.inf* es el que nos proporciona *Windows* como vimos antes, recuerda que también serán necesarios otros, para ello **revisa la sección correspondiente el archivo .inf** como hicimos con *linuxant*

Comprobamos el driver:

```
ndiswrapper -l
```

Y debe mostrar algo así como:

```
wlannic driver present, hardware is present
```

Donde **wlannic** es el nombre del controlador usado, el que proporcionaba Windows mediante su archivo .inf

Probamos con iwconfig y verificamos que existe el dispositivo

```
iwconfig
```

Accedemos la configuración de red

Repetimos el proceso descrito para *linuxant*:

En la ficha de hardware deberá aparecer el nuevo controlador, que ahora pondrá **ndiswrapper**

En la ficha de Dispositivos tendremos que **añadirlo, configurar sus opciones y activarlo**, exactamente igual que antes.

Una vez hecho esto y tras verificar que tenemos conexión a la red, etc... podemos usar:

```
ndiswrapper -m
```

De ese modo el controlador se activará y/o desactivará cuando se conecte o desconecte la tarjeta de red.

Para eliminar el/los controladores instalados,

Se usa la **opción -e** seguida del nombre del dispositivo configurado con **ndiswrapper**:

```
ndiswrapper -e wlannic
```

Bueno... fijo que te estarás haciendo muchas otras preguntas... ya las irás posteando, seguro... en la próxima entrega explicaré qué es eso de **iwconfig**, que no es otra cosa que una parte de las **Wireless Extensions** y también veremos otros controladores y "accesorios" del sistema para controlar los dispositivos wifi... hasta entonces... descansa y prueba :P

Saludos.