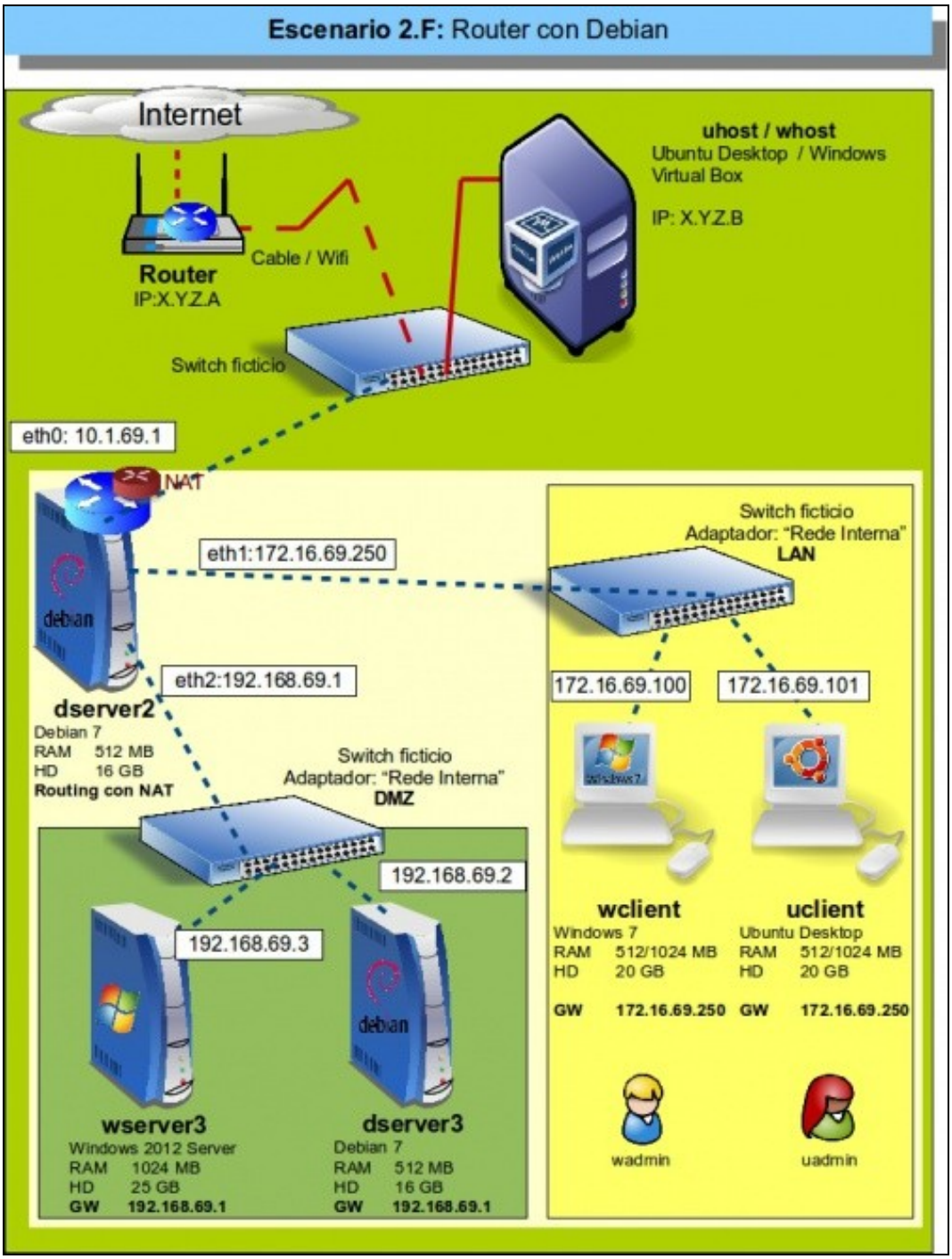
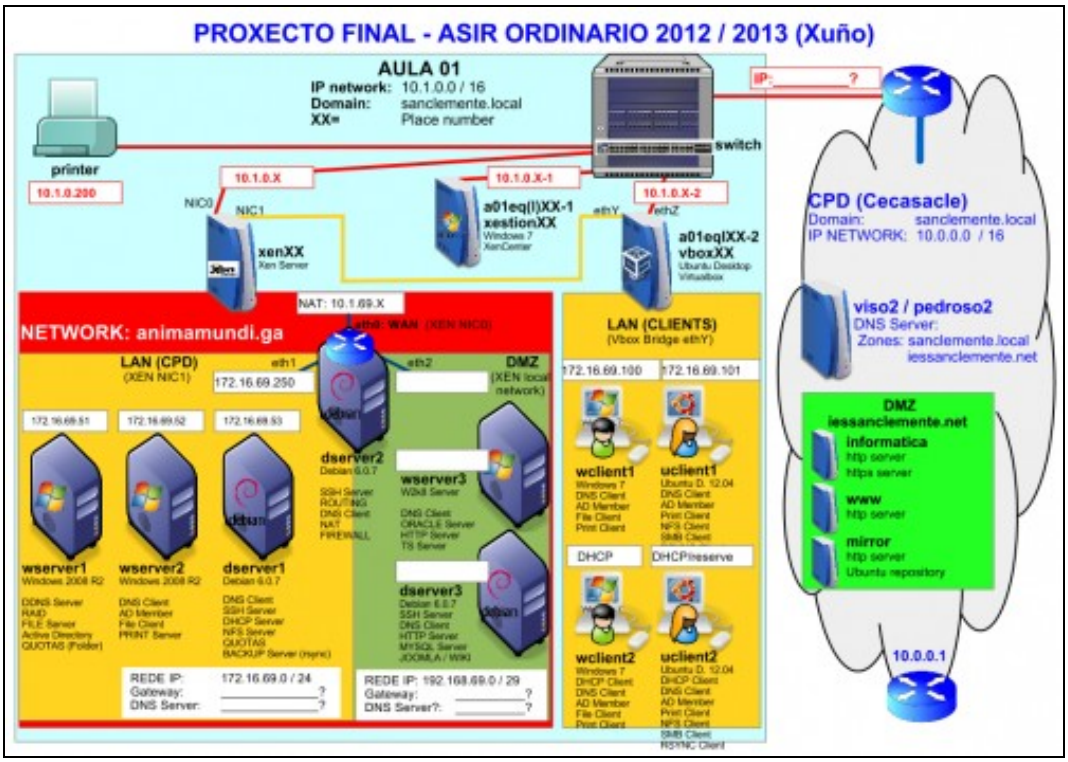
Escenario 2.F: Configuración dun router virtualizado con Debian



O que imos facer neste escenario é virtualizar o servidor *dserver2*, que realiza as funcións de router. Isto vainos permitir aplicar nun caso práctico e entender mellor o funcionamento dos modos de conexión en VirtualBox, xa que este servidor fai unha función similar á que realiza o propio VirtualBox cando nunha máquina conectamos unha tarxeta de rede en modo NAT ou rede NAT.

Este escenario está extraído do seguinte esquema de rede, no que se virtualiza este mesmo servidor sobre Xen Server:



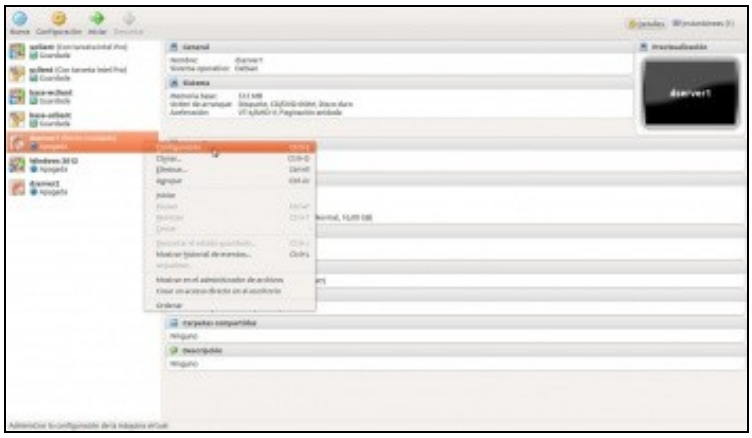
Sumario

- 1 Renomear e agrupar as máquinas do escenario
- 2 Configurar as tarxetas de rede das máquinas
- 3 Instalación de webmin e shorewall en *dserver2*
- 4 Configuración das interfaces de rede
- 5 Activación servizo de ruteo
- 6 Configuración da devasa e activación de NAT
- 7 Reenvío de portos

Renomear e agrupar as máquinas do escenario

Facendo uso da funcionalidade de VirtualBox de crear grupos de máquinas, imos agrupar todas as máquinas que van intervir neste escenario para facilitar o seu manexo.

- Renomear e agrupar as máquinas do escenario

- 

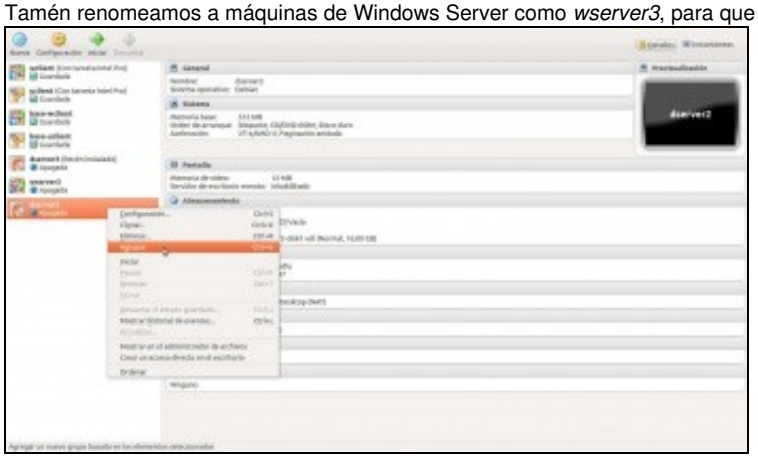
Imos á configuración de *dserver1*...

- 

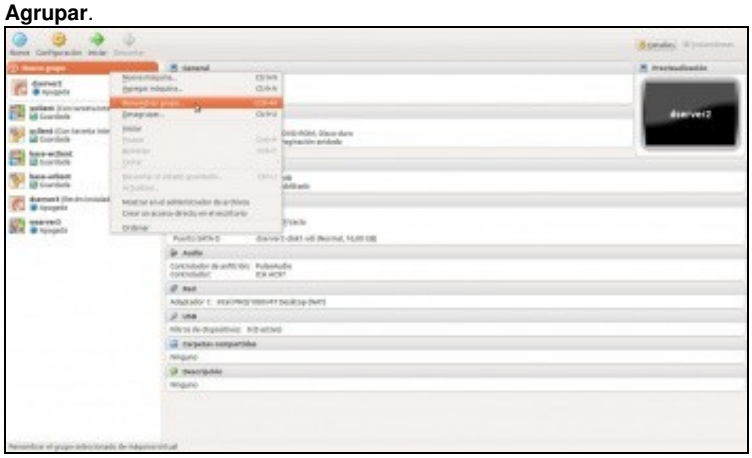
para cambiarlle o nome por *dserver3*.

- 

Tamén renomeamos a máquinas de Windows Server como *wserver3*, para que tamén coincida o seu nome co do escenario.

- 

Agora imos agrupar as máquinas que van intervir no escenario. Picamos co botón dereito sobre unha delas e seleccionamos a opción de **Agrupar**.

- 

Renomeamos o grupo, para chamalo **Escenario 2.F**.

- 

Arrastramos ao grupo todas as máquinas que imos utilizar.

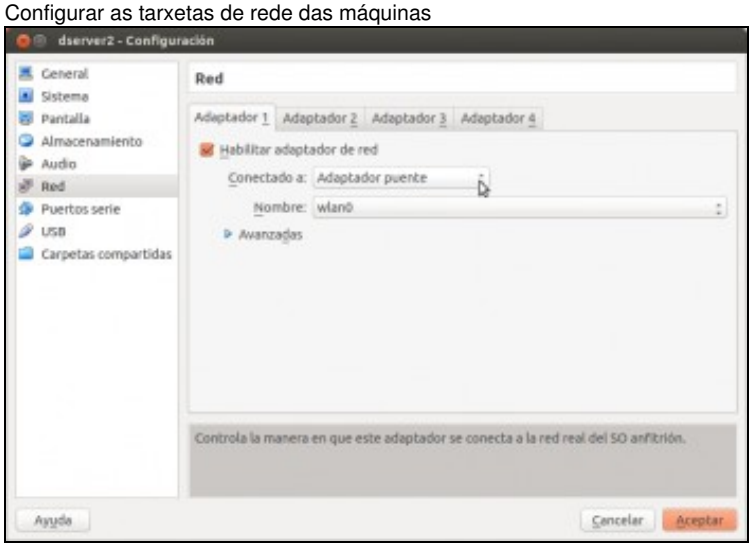
- 

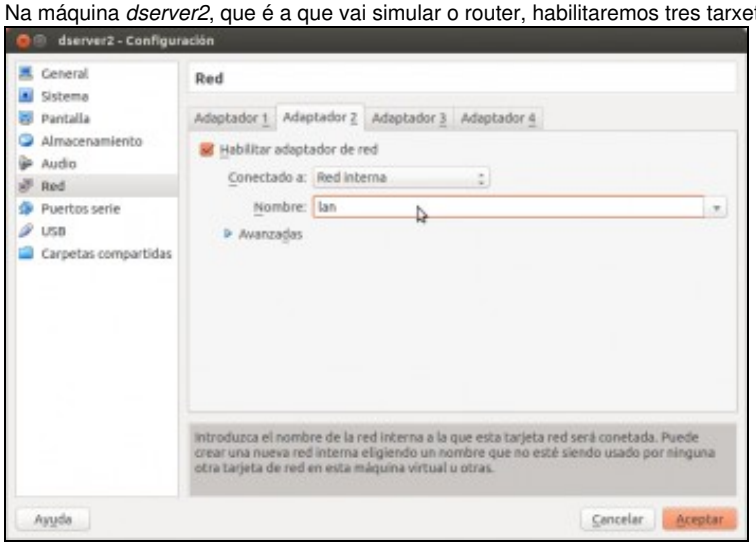
Vista do grupo coas máquinas xa incluídas.

Configurar as tarxetas de rede das máquinas

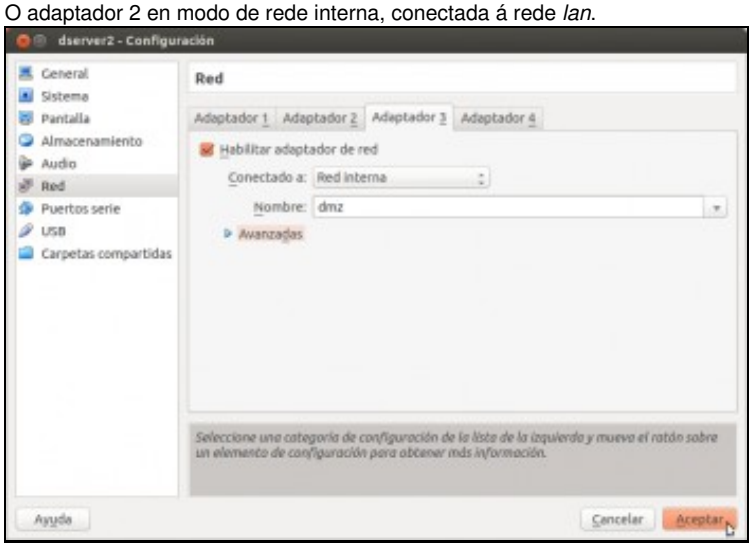
Como segundo paso, imos engadir nas máquinas os adaptadores necesarios e os modos de conexión de cada unha. Se revisamos o escenario, veremos que todos os adaptadores estarán en modo de rede interna excepto o adaptador 1 da máquina *dserver2* que estará en modo ponte.

Agora ben, non todos os adaptadores estarán na mesma rede interna. Dado que queremos simular dúas LANs distintas (a que leva por nome **LAN** e a que leva de nome **DMZ**), imos definir dúas redes internas diferentes, ás que lle poremos ese nome. Desa forma, os adaptadores que están conectados a unha rede interna teñen conexión entre si, pero non terán conexión cos que están conectados a outra rede interna.

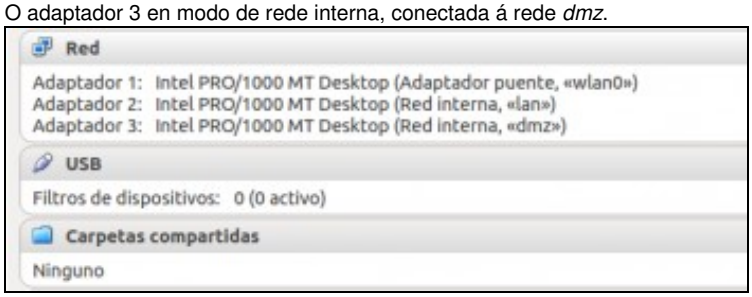
- Configurar as tarxetas de rede das máquinas
- 

Na máquina *dserver2*, que é a que vai simular o router, habilitaremos tres tarxetas de rede. O adaptador 1 en modo ponte.
- 

O adaptador 2 en modo de rede interna, conectada á rede *lan*.

- 

O adaptador 3 en modo de rede interna, conectada á rede *dmz*.

- 

Vista das propiedades de rede da máquina *dserver2* cos tres adaptadores.

- 

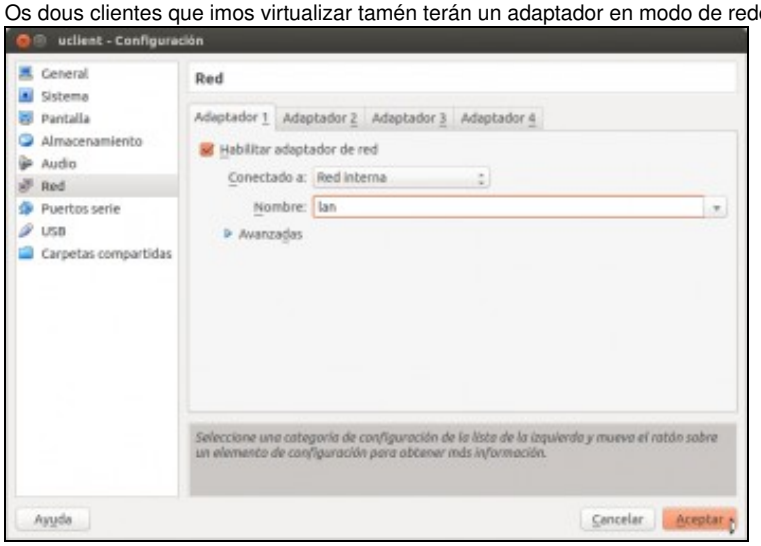
Os dous servidores que virtualizaremos, *dserver3* e *wserver3* terán os dous un adaptador en modo de rede interna, conectados á rede *dmz*.

- 

wserver3 igual que *dserver3*.

- 

Os dous clientes que imos virtualizar tamén terán un adaptador en modo de rede interna, pero neste caso conectados á rede *lan*.

- 

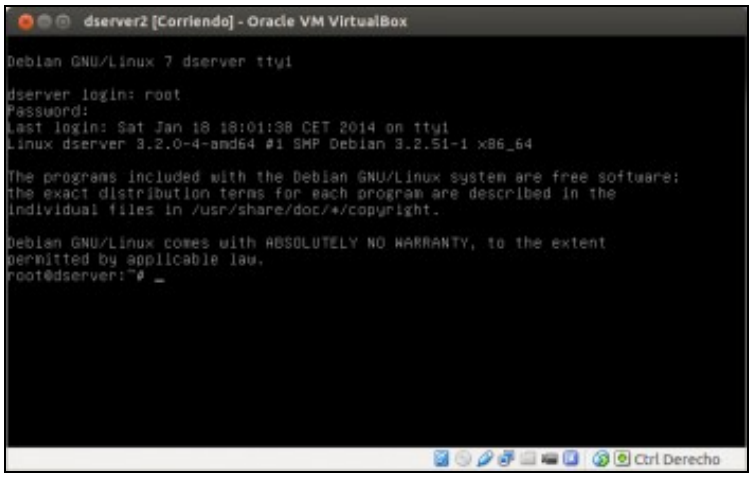
uclient igual que *wclient*.

Instalación de webmin e shorewall en *dserver2*

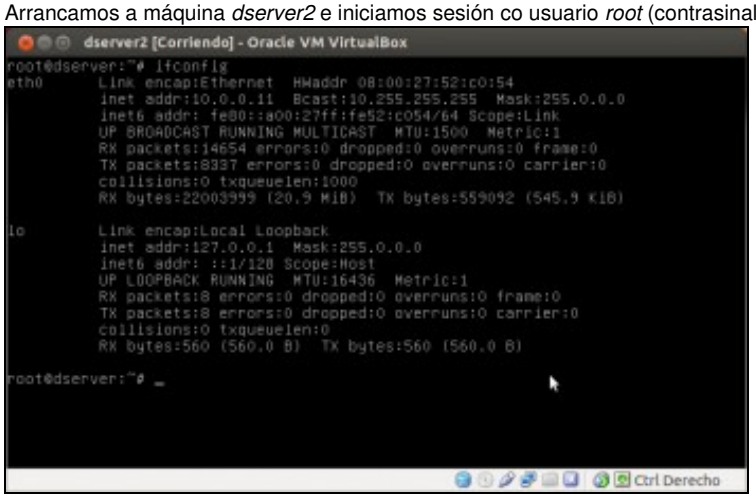
Imos ver os pasos a seguir para configurar a máquina *dserver2* para que realice as funcións que se reflicten no escenario. O obxectivo deste curso non é afondar na configuración de servizos de rede en Debian, así que intentaremos propoñer unha configuración o máis sinxela posible. Utilizaremos a ferramenta de administración de sistemas GNU/Linux webmin, que nos permitirá configurar o servizo de ruteo e devasa da máquina sen ter que manipular directamente os ficheiros de configuración.

Por iso imos instalar en primeiro lugar esta ferramenta na máquina *dserver2*, xunto co módulo **shorewall** que nos permitirá configurar as regras da devasa de forma máis accesible.

- Instalación de webmin e shorewall en *dserver2*

- 

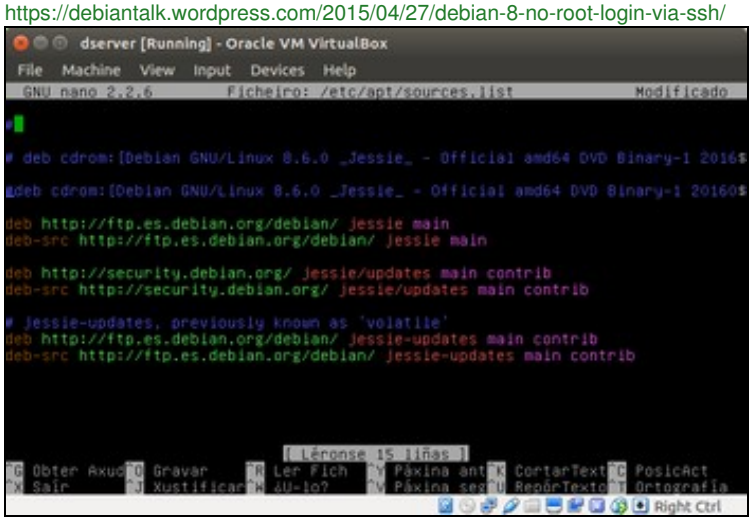
Arrancamos a máquina *dserver2* e iniciamos sesión co usuario *root* (contrasinal *abc123.*)

- 

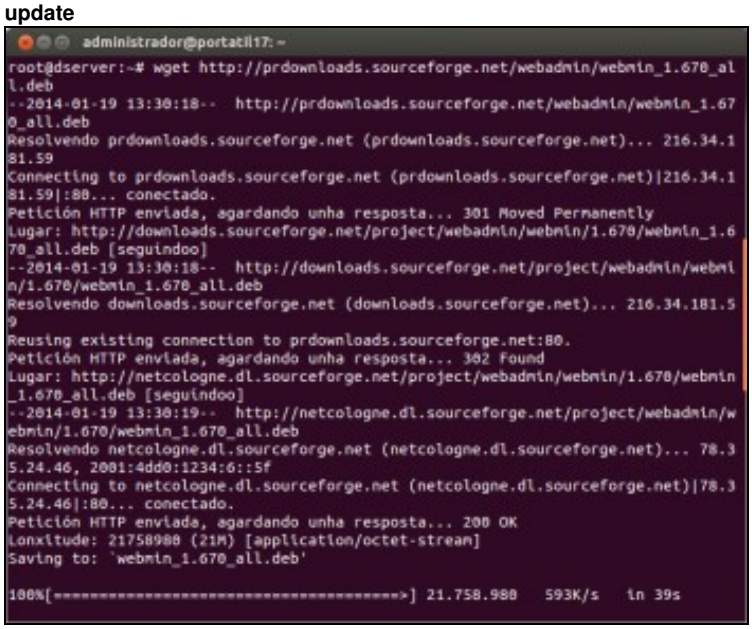
Co comando **ifconfig** podemos ver a dirección IP que a máquina tomou automaticamente por DHCP na interface que ten conectada en modo ponte. Se non houbese un servidor DHCP na rede, habería que configurar a dirección IP de forma manual. Nun apartado posterior no que se explica a configuración das interfaces de rede das distintas máquinas do escenario pódense ver os pasos da configuración das interfaces en *dserver3* para ver como facelo.

- 

Dado que a máquina *dserver2* ten instalado o servidor ssh, agora que sabemos a súa dirección IP podemos conectarnos con un cliente ssh dende o *host* ou outro equipo da rede, xa que isto nos facilitará copiar e pegar os comandos que vaimos introducindo (se o *host* é un equipo Windows podemos utilizar o programa putty como cliente ssh).
Ollo que nas novas versións de ssh non deixa, por defecto, iniciar sesión co usuario root. Por tanto podemos iniciar sesión co usuario dadmin (abc123.) e unha vez no servidor pasarse a root con **su -**. Tamén se pode editar o ficheiro /etc/ssh/sshd-config tal como se indica en: https://debiantalk.wordpress.com/2015/04/27/debian-8-no-root-login-via-ssh/

- 

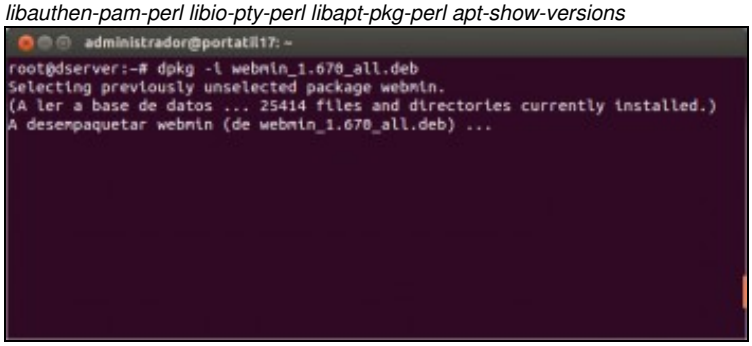
Asegurarse de que están comentadas as sources do CD-ROM (**nano /etc/apt/sources.list**). E actualizar a lista dos paquetes: **apt-get update**

- 

Descargamos o paquete do webmin para debian, co comando *wget http://prdownloads.sourceforge.net/webadmin/webmin_1.820_all.deb* (Descargaremos e instalaremos a última versión, independentemente da versión que aparece na imaxe)

- 

Instalamos unha serie de paquetes necesarios para poder instalar o webmin. Introducimos o comando: *apt-get install libnet-ssleay-perl libauthen-pam-perl libio-pty-perl libapt-pkg-perl apt-show-versions*

- 

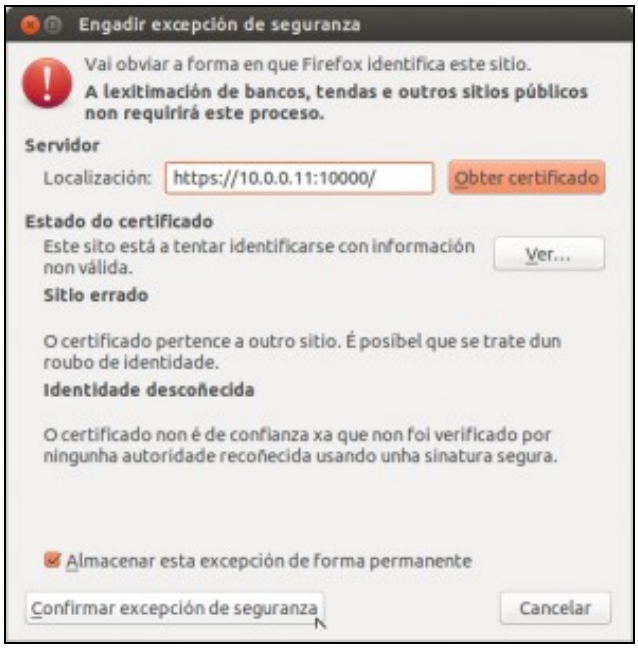
Instalamos o webmin, introducindo o comando *dpkg -i webmin_1.820_all.deb*

- 

E por último, instalamos o shorewall, introducindo o comando *apt-get install shorewall*

- 

Xa podemos conectarnos ao webmin instalado en *dserver2*. Webmin é un servizo de administración remota que *corre* no porto 10000 e ao que pode accederse con un navegador usando unha conexión segura (*https*). Así que no *host* ou en calquera equipo da rede abrimos un navegador e introducimos como dirección https://IP_dserver2:10000. Aparecerá o aviso do navegador debido a que o certificado de seguridade non é fiable, cousa totalmente normal. Engadimos unha excepción...

- 

Confirmamos a excepción...

- 

e xa podemos ver a páxina de inicio de sesión de Webmin, na que nos *loguearemos* co usuario *root* e contrasinal *abc123*.

- 

Xa estamos na páxina de inicio de Webmin, e o que podemos facer para deixar a ferramenta personalizada é cambiar o idioma, xa que por defecto ven en Inglés. Picamos dentro da categoría **Webmin** en **Change Language and Theme**.

- 

En **Webmin UI language** activamos **Personal choice** e seleccionamos **Spanish** (desafortunadamente, non contamos con tradución ao galego).

- 

Recargamos a páxina e xa temos dispoñible toda a interface en castelán.

Configuración das interfaces de rede

Neste apartado imos abordar a configuración IP de todas as máquinas virtuais que forman o escenario. Cada unha delas será diferente xa que contamos con unha máquina Windows 7 (*wclient*), unha máquina Ubuntu (*uclient*), unha máquina Windows 2012 Server (*wserver3*) e dúas máquinas debian pero unha delas configurarémola co webmin (*dserver2*) e a outra manipulando directamente os ficheiros de configuración (*dserver3*).

- Configuración das interfaces de rede en *dserver2*
- 

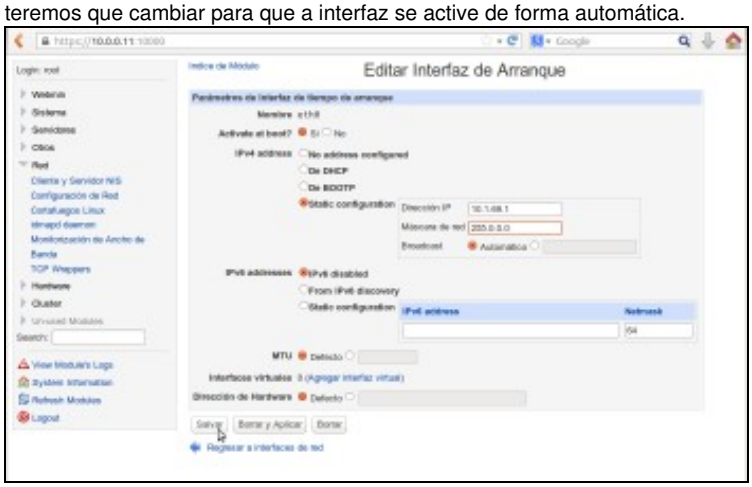
Neste caso imos facer a configuración IP deste equipo mediante o webmin. Dentro do apartado de **Rede**, picamos en **Configuración de Rede**.

- 

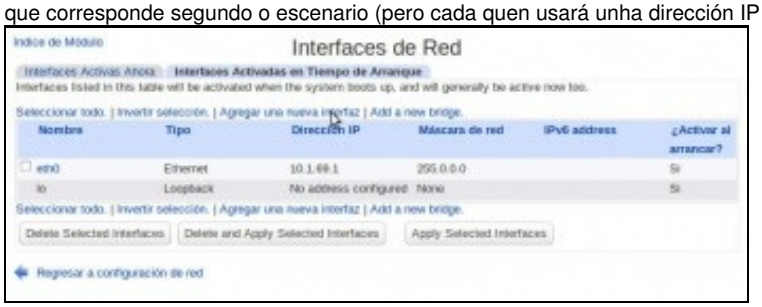
Entramos no apartado de **Interfaces de Rede**

- 

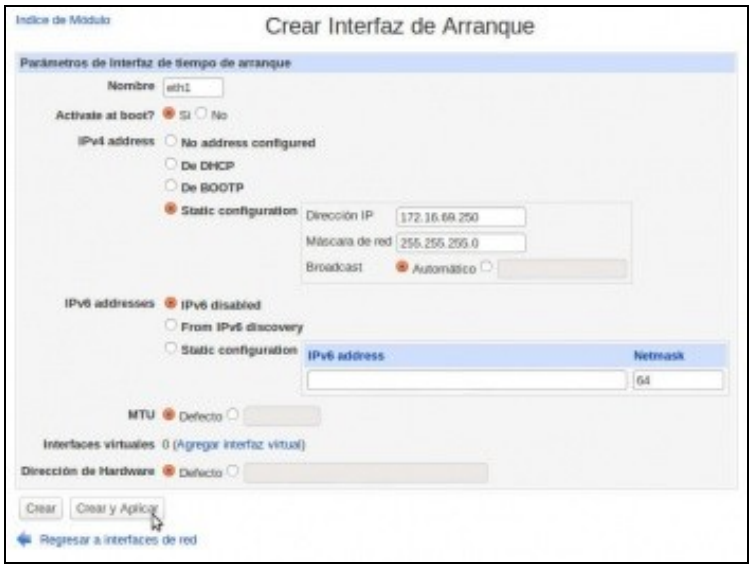
É moi importante prestar atención a que esta páxina se divide en dúas pestanas: **Interfaces Activas Agora** e **Interfaces Activadas en Tempo de Arranque**, e sempre teremos que facer os cambios nesta última (que é a que vemos por defecto), xa que senón os cambios non perdurarán cando se reinicie a máquina virtual. Veremos que só hai unha interfaz configurada, *eth0*, por DHCP. Picamos sobre ela para poñerlle a dirección que lle corresponde no escenario. Fixarse antes en que na columna de **Activar ao inicio** pon que non, parámetro que teremos que cambiar para que a interfaz se active de forma automática.

- 

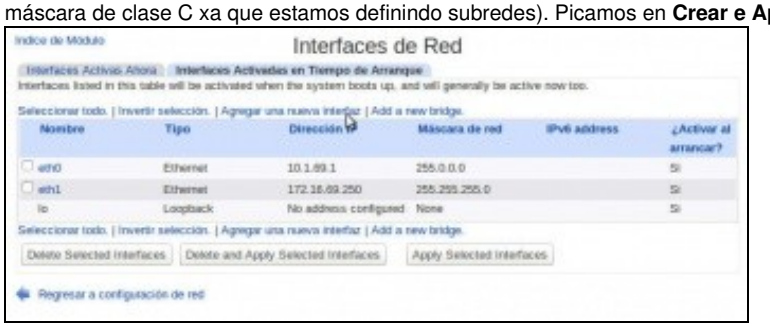
No parámetro **Activate at boot** marcamos que **Si**, e en **IPv4 addess** marcamos **Static configuration**. Introducimos a dirección IP e máscara que corresponde segundo o escenario (pero cada quen usará unha dirección IP que pertenza a rede que englobe ao *host*) e salvamos.

- 

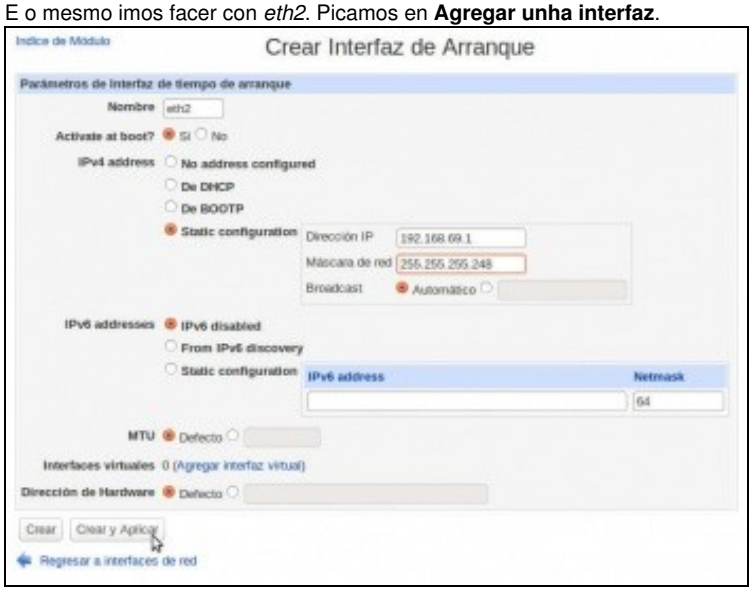
Agora imos engadir outra interfaz de rede, xa que no equipo xa existen dúas interfaces máis (pódense ver co comando **ifconfig -a**) que son *eth1* e *eth2*, pero non están configurados. Picamos en **Agregar unha nova interfaz**.

- 

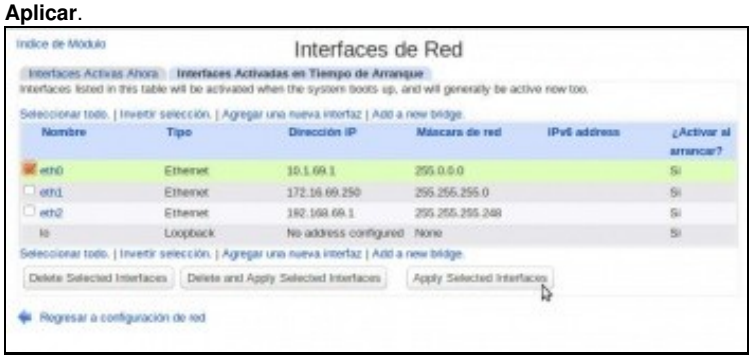
Poñemos como nome da interfaz *eth1*, marcamos que se active no inicio (*Activate at boot->Si*), e marcamos **Static configuration** en **IPv4 address** para introducir a dirección IP e máscara que se indican no escenario (fixarse que estamos usando unha dirección IP de clase B con máscara de clase C xa que estamos definindo subredes). Picamos en **Crear e Aplicar**.

- 

E o mesmo imos facer con *eth2*. Picamos en **Agregar unha interfaz**.

- 

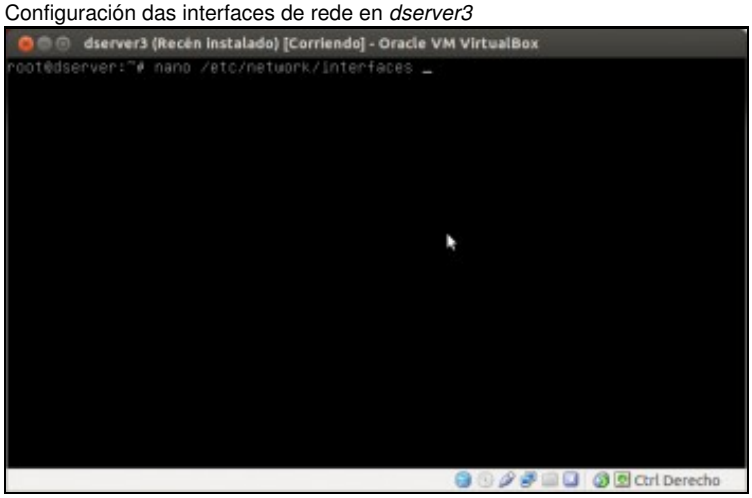
Introducimos os datos da interfaz segundo o escenario (de novo estamos facendo subredes na rede de clase C). Picamos en **Crear y Aplicar**.

- 

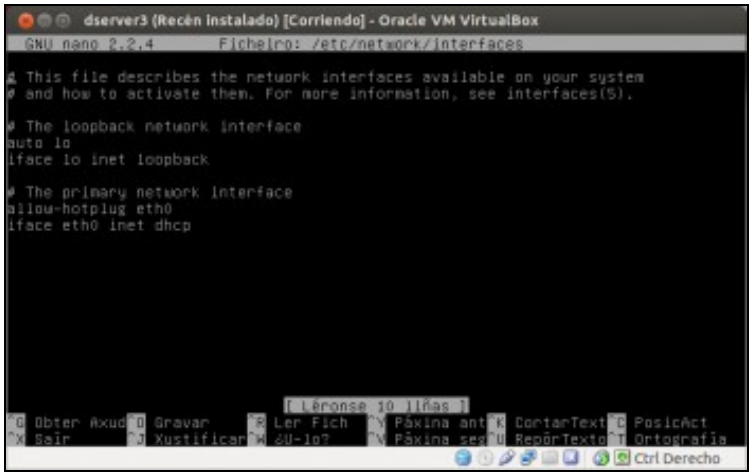
Seleccionamos a interfaz *eth0*, que é o único cambio que aínda non está aplicado, e picamos no botón de **Apply Selected Interfaces**. Neste momento o webmin deixará de responder, xa que acabamos de cambiar a dirección IP da interfaz pola que nós nos estabamos conectado co navegador (era 10.0.0.11 e pasa a ser 10.1.69.1). Así que teremos que cambiar a dirección do navegador para poñer https://10.1.69.1:10000

- 

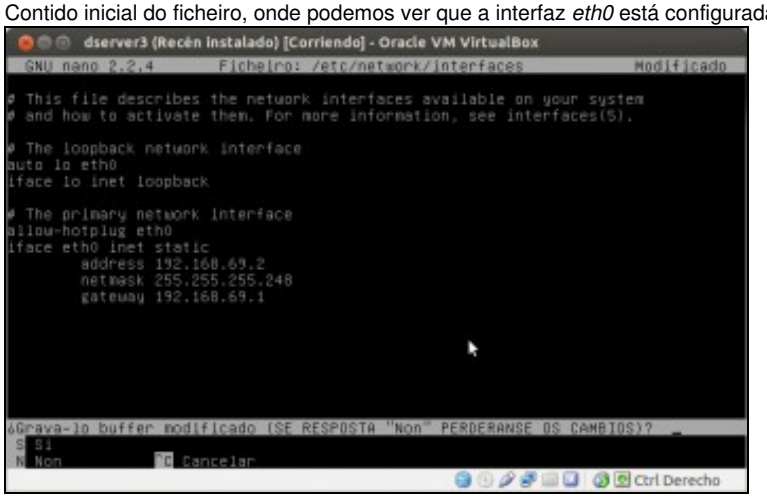
E xa debemos ter acceso ao webmin.

- Configuración das interfaces de rede en *dserver3*
- 

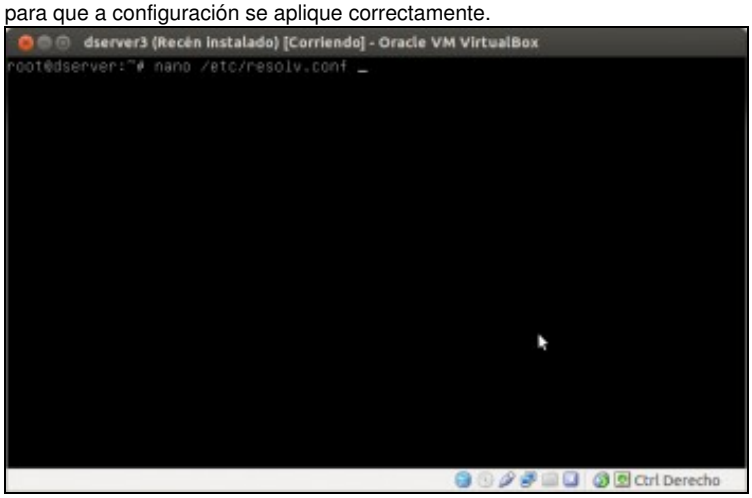
Neste caso, na máquina non temos o webmin instalado, así que imos facer a configuración IP da súa interfaz directamente nos ficheiros de configuración. O ficheiro de configuración básico das interfaces de rede en Debian é **/etc/network/interfaces**, así que imos editar este ficheiro co editor *nano*.

- 

Contido inicial do ficheiro, onde podemos ver que a interfaz *eth0* está configurada por DHCP.

- 

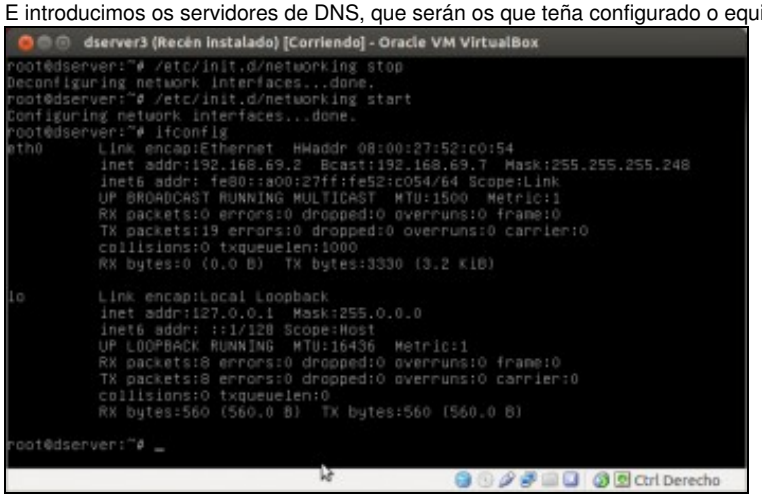
Na imaxe vese o contido que imos deixar no ficheiro, engadindo a interfaz na liña **auto...** para que se active automticamente e establecendo unha configuración IP estática, cos datos de dirección e máscara indicados no escenario. A porta de enlace predeterminada (*gateway*) para este equipo será a dirección IP da interfaz *eth2* de *dserver2*. É moi importante revisar ben a sintaxe de todo o que se introduciu no ficheiro para que a configuración se aplique correctamente.

- 

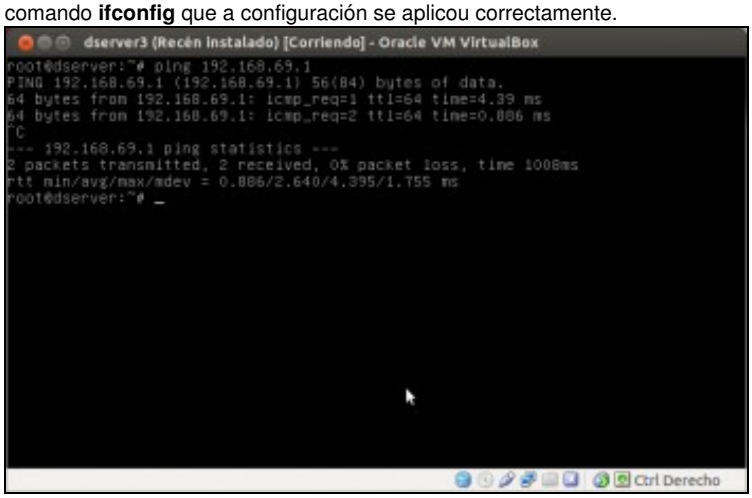
Só nos falta indicar os servidores de DNS, que se introducen no ficheiro **/etc/resolv.conf**. Editamos este ficheiro....

- 

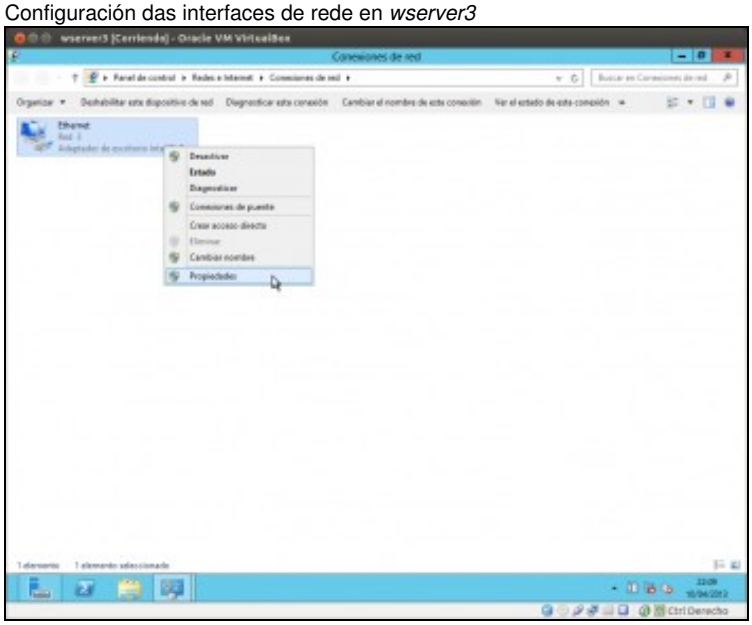
E introducimos os servidores de DNS, que serán os que teña configurado o equipo *host*. Cada quen debe introducir os do seu equipo.

- 

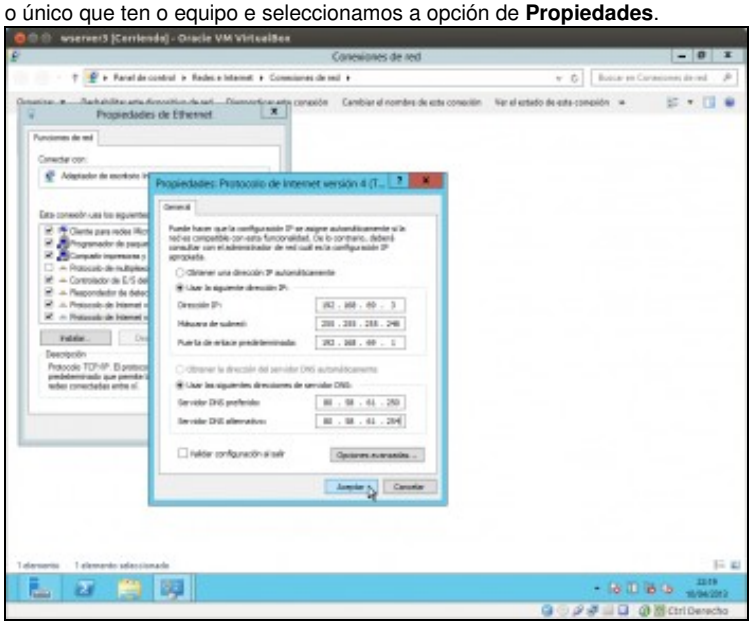
Detemos e iniciamos o servizo de rede cos comandos **/etc/init.d/networking stop** e **/etc/init.d/networking start**. Convén comprobar co comando **ifconfig** que a configuración se aplicou correctamente.

- 

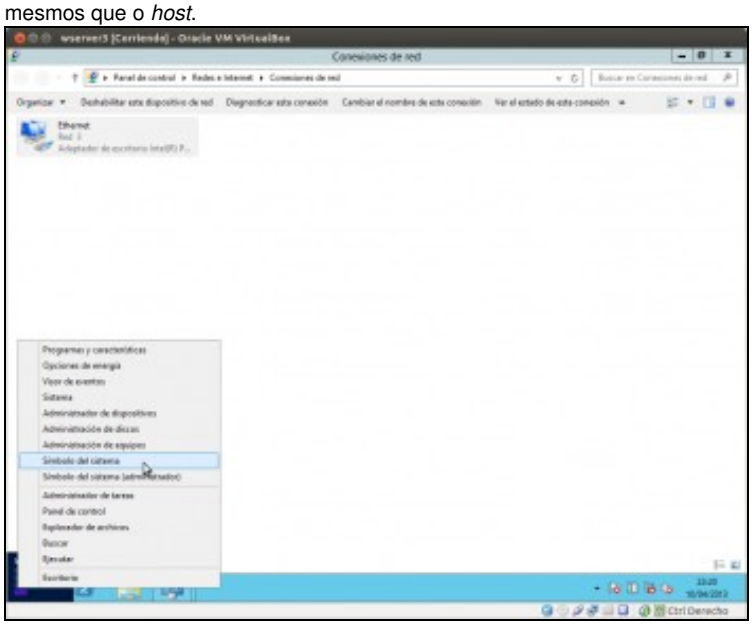
Tamén imos comprobar que temos conectividade co equipo *dserver2*. É normal que non teñamos conexión co *host* nin a Internet, xa que *dserver2* non está facendo as funcións de ruteo e NAT.

- Configuración das interfaces de rede en *wserver3*
- 

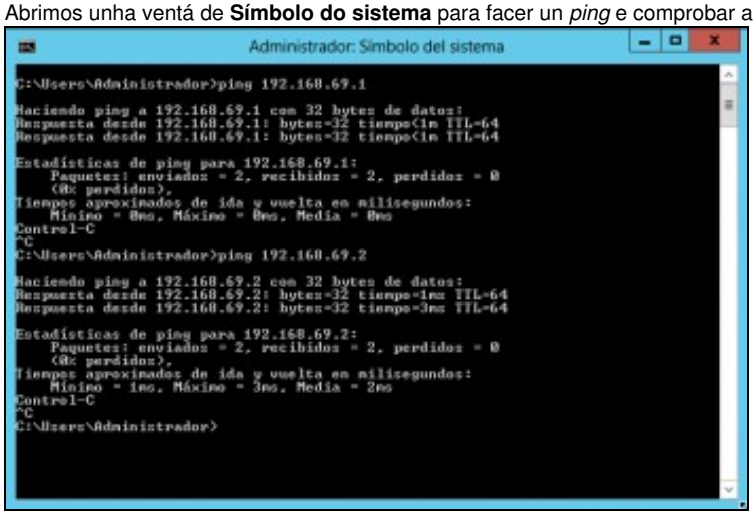
Neste caso a configuración é similar a que temos feito en apartados anteriores sobre *wclient*. Na lista de adaptadores de rede, picamos sobre o único que ten o equipo e seleccionamos a opción de **Propiedades**.

- 

Facemos dobre clic sobre o **Procolo de Internet (TCP/IP) versión 4** e introducimos na ventá os datos de dirección IP e máscara que se indican no escenario. A porta de enlace predeterminada será de novo a dirección IP da interfaz *eth2* de *dserver2*, e os servidores DNS os mesmos que o *host*.

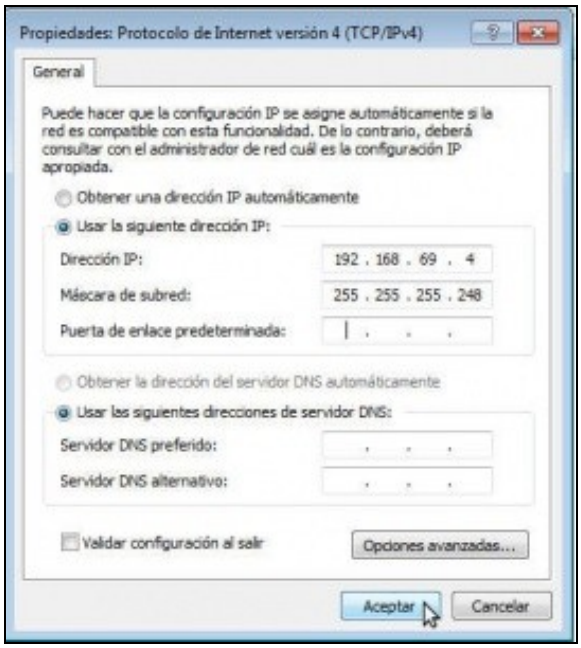
- 

Abrimos unha ventá de **Símbolo do sistema** para facer un *ping* e comprobar a conectividade con *dserver2* e *dserver3*

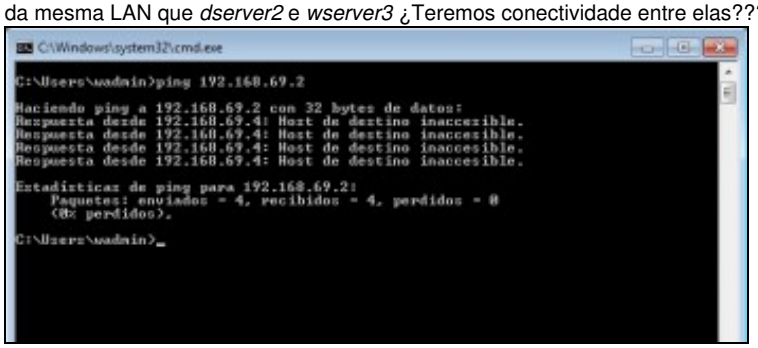
- 

Perfecto!!

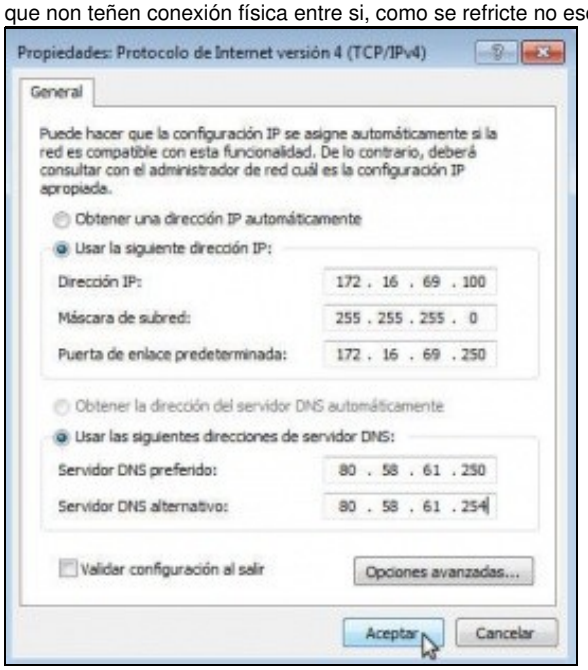
- Configuración das interfaces de rede en *wclient*

- 

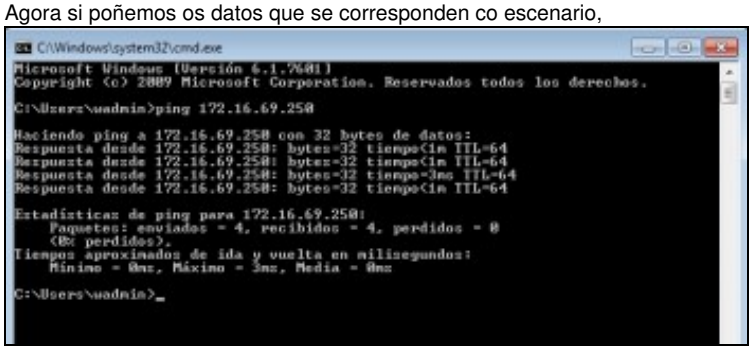
Neste caso, para comprender mellor o funcionamento da rede interna en VirtualBox, imos poñerlle á interfaz antes unha dirección IP dentro da mesma LAN que *dserver2* e *wserver3* ¿Teremos conectividade entre elas???

- 

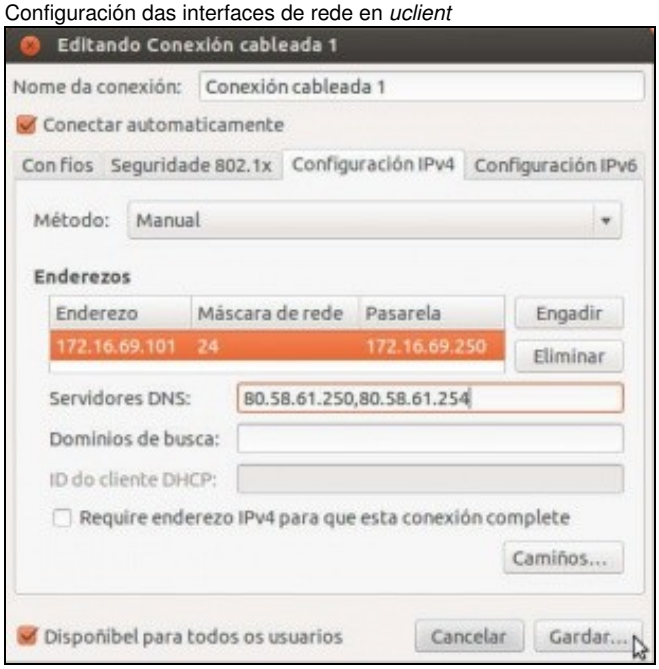
A resposta é que non, porque aínda que todas as interfaces destas máquinas están en modo de rede interna, están conectadas a redes internas diferentes (a de esta máquina á rede *lan* e as das outras dúas á rede *dmz*), así que están conectados e *switchs ficticios* distintos e que non teñen conexión física entre si, como se refricte no escenario.

- 

Agora si poñemos os datos que se corresponden co escenario,

- 

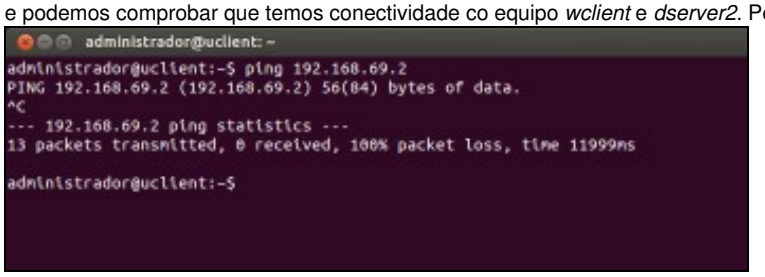
e temos conexión coa interfaz *eth1* de *dserver2*, xa que esa interfaz si que está en modo de rede interna e conectada á rede *lan*.

- Configuración das interfaces de rede en *uclient*
- 

E por últimoa introducimos a configuración IP para este equipo segundo os datos do escenario. Gardamos os cambios,

- 

e podemos comprobar que temos conectividade co equipo *wclient* e *dserver2*. Perfecto!!

- 

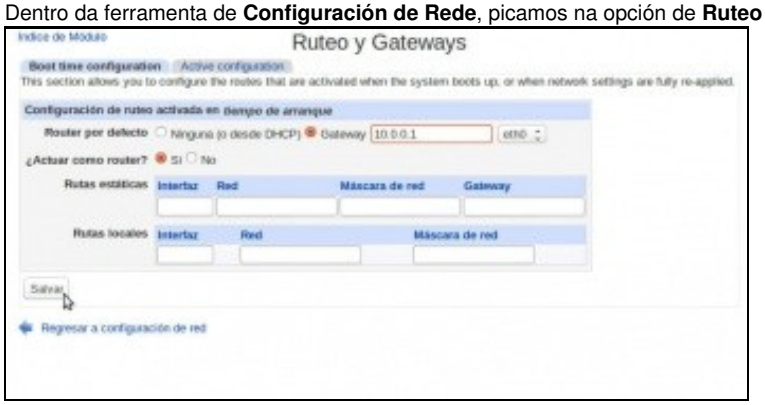
Pero non temos conectividade cos equipos que están na rede *dmz*, xa que *dserver2* non ten activado o servizo de ruteo e polo tanto non reenvía os paquetes que esta máquina lle está mandando ao seu destino. Imos ver como podemos solucionalo...

Activación servizo de ruteo

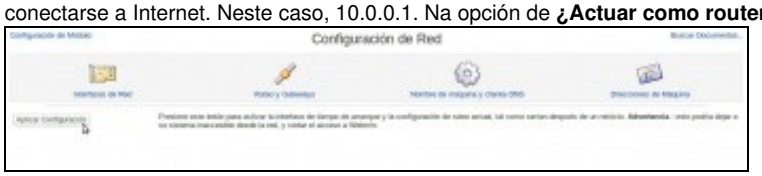
Utilizando o webmin, imos activar o servizo de enrutamento na máquina *dserver2* para poder ter conexión entre as máquinas que están nas dúas redes (*lan* e *dmz*):

- Activar o servizo de ruteo na MV Debian
- 

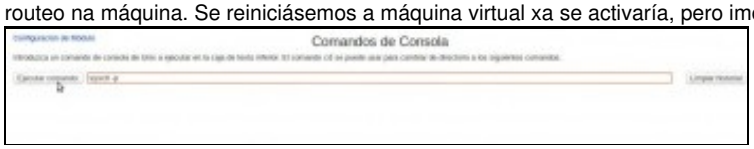
Dentro da ferramenta de **Configuración de Rede**, picamos na opción de **Ruteo e gateways**.

- 

Establecemos como porta de enlace predeterminada (*Router por defecto*) para este equipo a mesma que estea usando o equipo *host* para conectarse a Internet. Neste caso, 10.0.0.1. Na opción de **¿Actuar como router?** indicamos que si.

- 

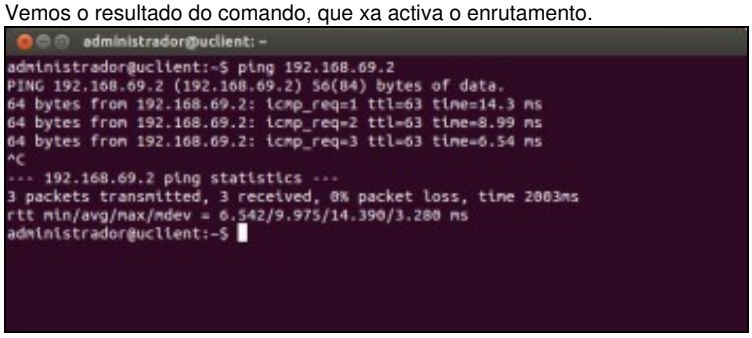
Picamos no botón de **Aplicar configuración**, pero neste caso (bug do webmin) con isto non conseguimos realmente activar xa o servizo de routeo na máquina. Se reiniciásemos a máquina virtual xa se activaría, pero imos ver como podemos activar o cambio sen ter que reiniciar.

- 

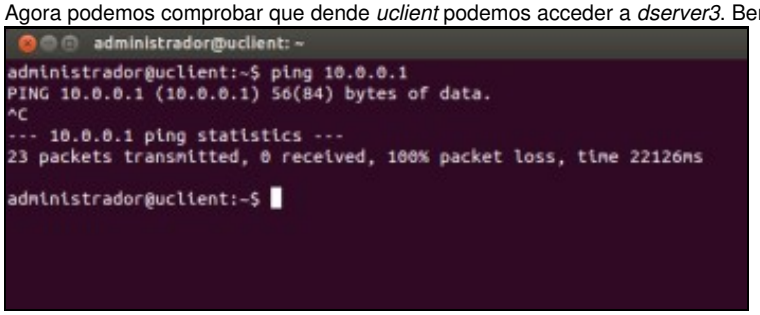
No propio webmin, imos á ferramenta de **Comandos de consola** (dentro do apartado de **Otros**) e introducimos o comando: **sysctl -p**. Picamos no botón de **Executar comando** para executar este comando no sistema.

- 

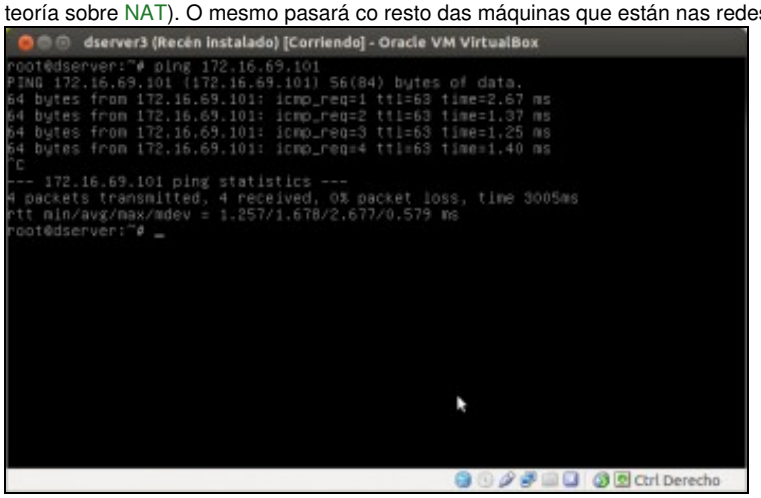
Vemos o resultado do comando, que xa activa o enrutamento.

- 

Agora podemos comprobar que dende *uclient* podemos acceder a *dserver3*. Ben!!

- 

Pero... ¿podemos acceder a un equipo da rede real (o router de saída a Internet, por exemplo)?... Non... ¿Por que? Porque *dserver2* non está facendo a función de NAT, e polo tanto un equipo como *uclient* que ten unha dirección IP privada non pode acceder a unha rede pública (aínda que a rede 10 sexa unha rede privada, neste caso para os equipos que están nas redes *lan* e *dmz*, é como se fose pública. Revisar a teoría sobre NAT). O mesmo pasará co resto das máquinas que están nas redes *lan* e *dmz*.

- 

E aínda temos outro problema... Dende o equipo *dserver3* podemos acceder ao equipo *uclient* ¿Debe ser así? Seguindo as regras da devasa, suponse que non se deberían permitir conexións que intenten entrar na rede interna, xa que é a rede que queremos protexer do exterior. A única rede accesible dende o exterior debería ser en todo caso a zona desmilitarizada ou dmz, que é na que se atopan equipos que prestan servizos accesibles dende Internet (chamados comunmente bastións). Imos agora a resolver todo isto...

Configuración da devasa e activación de NAT

Para resolver as dúas problemáticas que acabamos de detectar no apartado anterior, imos configurar a devasa no equipo *dserver2* e activar a función de NAT. Utilizaremos para iso o módulo de *shorewall* de webmin, que nos facilitará a configuración de iptables, que é o módulo de Linux que xestiona as regras da devasa.

- 

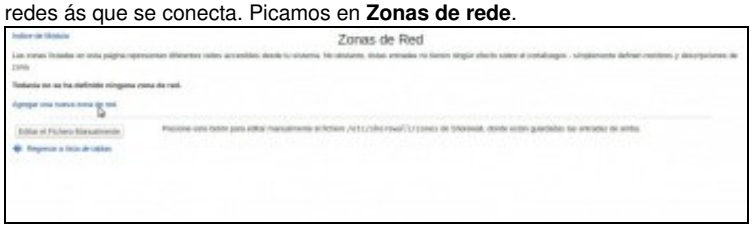
De momento o módulo de *Shorewall* non aparece en ningunha das categorías do webmin, xa que foi instalado despois. Picamos na opción de **Refresh Modules** para que webmin busque se ten novos módulos instalados e os engada na categoría correspondente.

- 

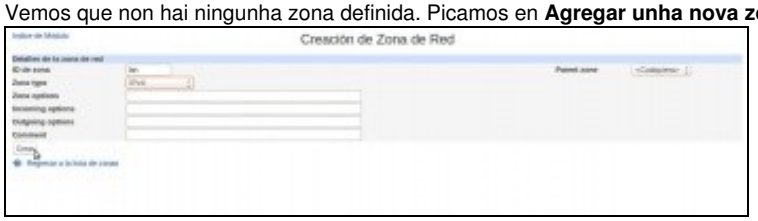
Atoparemos agora dentro da categoría de *Rede* o módulo de **Cortafuegos Shoreline**. Entramos nel. De momento o módulo está parado (fixarse en que temos o botón de *Iniciar el Cortafuegos*), pero se intentamos inicialo veremos que nos da un erro, debido a que é necesario establecer unha configuración básica para podelo facer.

- 

En primeiro lugar, teremos que definir as zonas de rede que vai xestionar a devasa, que non son máis que nomes lóxicos para as distintas redes ás que se conecta. Picamos en **Zonas de rede**.

- 

Vemos que non hai ningunha zona definida. Picamos en **Agregar unha nova zona de rede**.

- 

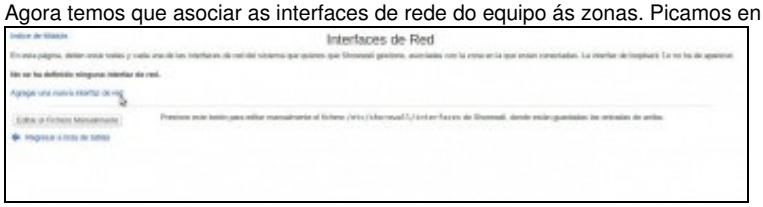
Poñemos un ID da zona (*lan*), seleccionamos como tipo *IPv4* e picamos no botón de **Crear**. Esta zona representa a conexión coa rede *lan* do escenario.

- 

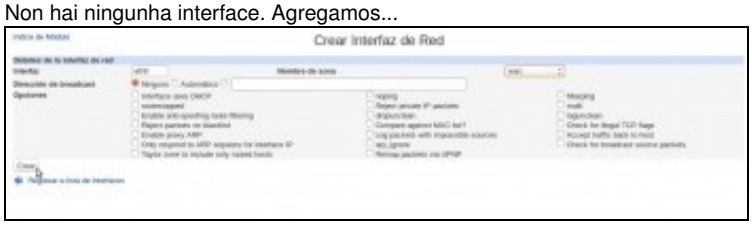
Así teremos que definir as zonas *wan* (que representa a rede pública) e *dmz* (que representa a rede *dmz* do escenario). Tamén teremos que definir unha zona de tipo *Firewall system*, que representa ao propio equipo. Neste caso chamámoslle ds2. Recoméndase seguir estes nomes xa que teñen que ser nomes curtos e hai moitos caracteres inválidos.

- 

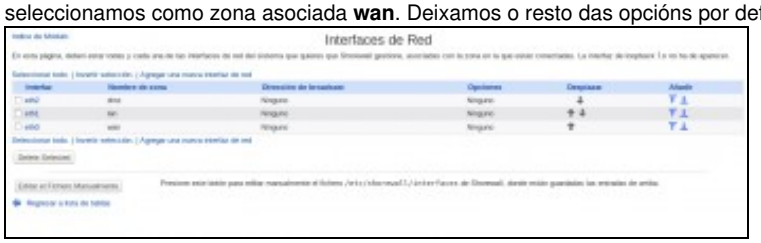
Agora temos que asociar as interfaces de rede do equipo ás zonas. Picamos en **Interfaces de rede**.

- 

Non hai ningunha interface. Agregamos...

- 

Introducimos o nome dunha interface de rede do equipo, neste caso *eth0*. Esta interface é a que se conecta á rede pública, así que seleccionamos como zona asociada **wan**. Deixamos o resto das opcións por defecto e picamos en **Crear**.

- 

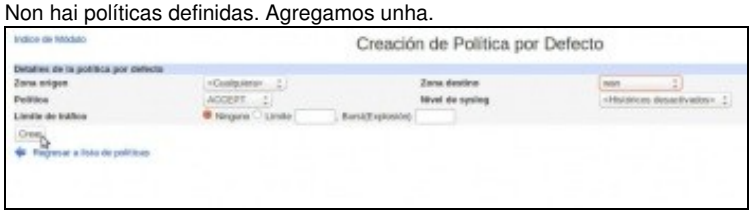
Ao final temos que ter asociadas as tres interfaces *eth0*, *eth1* e *eth2* ás zonas *wan*, *lan* e *dmz* respectivamente.

- 

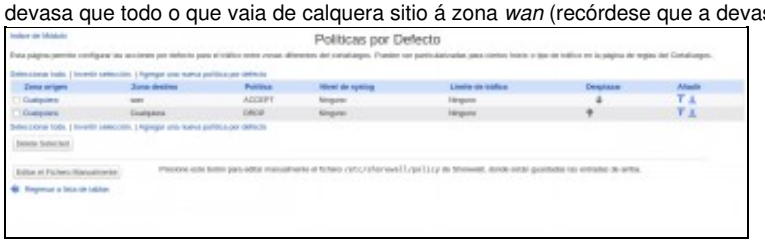
O terceiro paso é definir a política por defecto da devasa; é dicir, como se vai comportar a nivel xeral. Aquí haberá que ter en conta dúas cuestións: primeiro que todo posible tráfico que poida manexar o equipo (da rede *lan* á rede *wan*, da rede *wan* á rede *dmz*, etc.) debe estar incluído nalgunha política, xa que se non a devasa non sabería que facer con el. E segundo que as políticas son as pautas básicas que rexen o comportamento da devasa, que logo afinaremos máis concretamente coas regras. As regras teñen prioridade sobre as políticas, así que a devasa mirará primeiro se ao paquete se lle pode aplicar unha regra, e se non é así, aplicaralle unha das políticas por defecto que teña definidas.

- 

Non hai políticas definidas. Agregamos unha.

- 

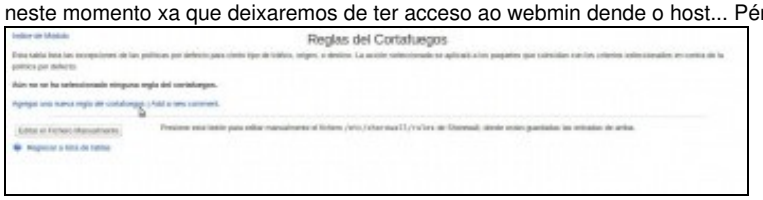
As políticas simplemente teñen unha zona orixe, unha zona de destino e unha acción a tomar. Por exemplo, neste caso estamos dicíndolle á devasa que todo o que vaia de calquera sitio á zona *wan* (recórdese que a devasa ten asociada a zona *wan* á interface *eth0*) o acepte.

- 

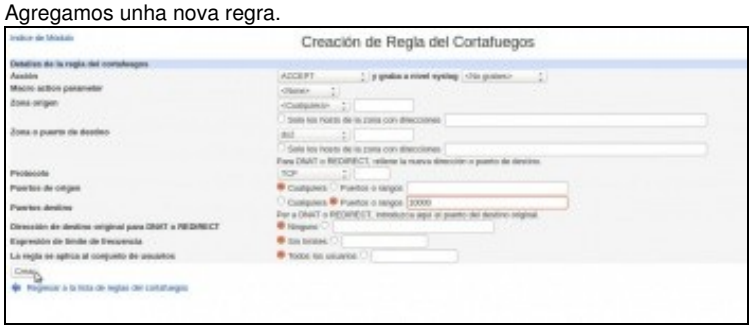
A configuración da devasa é unha decisión moi delicada e dependerá moito do nivel de seguridade e restricións que queiramos aplicar na nosa rede, pero unha opción para este caso podería ser deixar estas dúas políticas. Acéptase todo o que vaia á *wan*, e o resto rexéitase. Nótese que a orde na que se definen as políticas son moi importantes (por iso hai botóns para subilas e baixalas), xa que se aplican se arriba a abaixo. Por exemplo, se neste caso situásemos a segunda regra como primeira, executaríase sempre, xa que encaixa con calquera tráfico.

- 

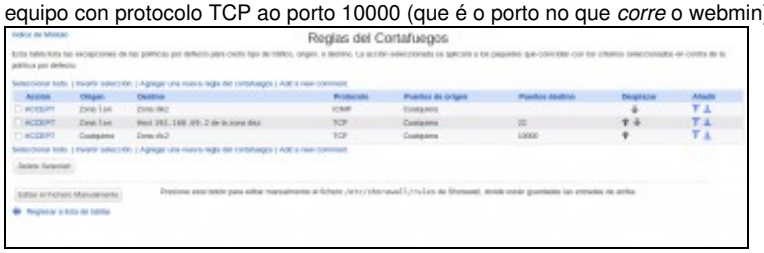
Por último, imos introducir algunhas regras para afinar o comportamento da devasa antes de iniciala (Ollo!! Non se debe iniciar a devasa neste momento xa que deixaremos de ter acceso ao webmin dende o host... Pénsese por que).

- 

Agregamos unha nova regra.

- 

Como podemos ver, as regras ofrecen criterios moito máis específicos para poder indicar á devasa que tipo de tráfico se debe aceptar e cal non (protocolo, IP de orixe e destino, porto de orixe e destino, etc.). Con este regra indicámoslle que acepte o tráfico que vaia ao propio equipo con protocolo TCP ao porto 10000 (que é o porto no que *corre* o webmin).

- 

Podemos deixar estas regras: Permitimos o acceso ao webmin, os *pings* dende a *lan* á *dmz* e acceder dende a *lan* ao porto 22 da máquina *dserver3*. En función dos servizos que quixésemos ter accesibles nos servizos da *DMZ* dende a *lan* ou dende Internet iríamos engadindo máis regras. A orde nas regras tamén é moi importante, aínda que neste caso non se solapan unhas coas outras.

- 

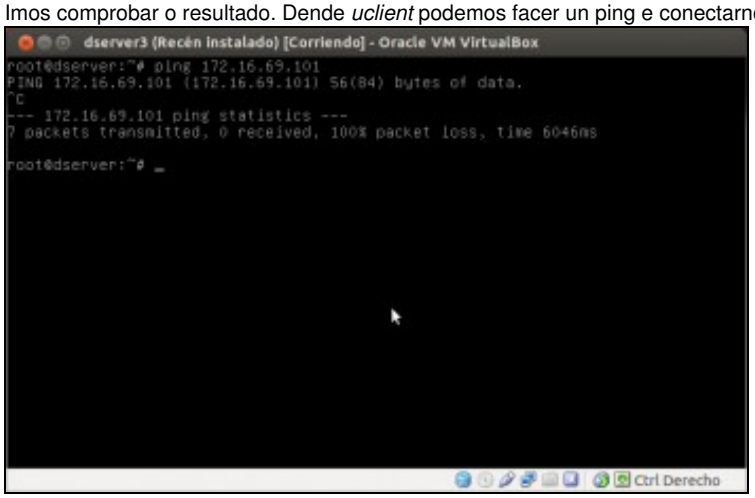
Por fin!! Xa podemos iniciar a devasa (Agora se se cometeu algún erro na configuración a devasa non arrancará, neste caso revísese as zonas, interfaces, políticas e regras definidas).

- 

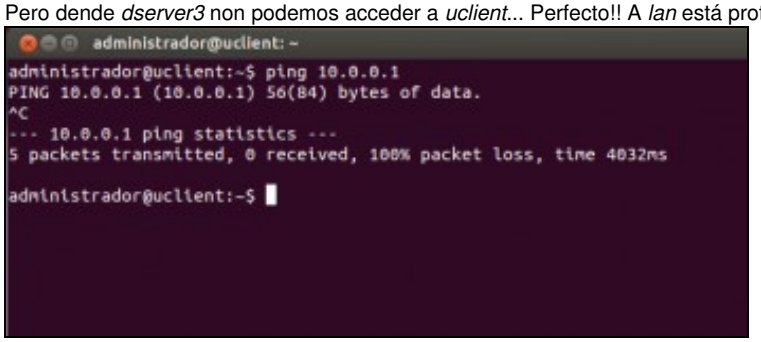
As opcións que aparecen agora de manexo da devasa indican que está iniciado.

- 

Imos comprobar o resultado. Dende *uclient* podemos facer un ping e conectarnos por ssh a *dserver3*.

- 

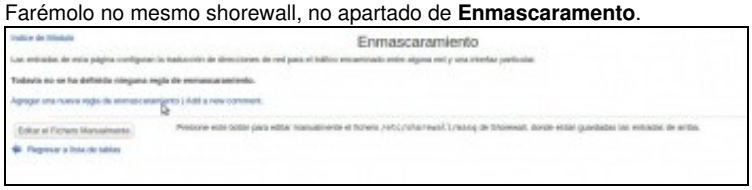
Pero dende *dserver3* non podemos acceder a *uclient*... Perfecto!! A *lan* está protexida pola devasa.

- 

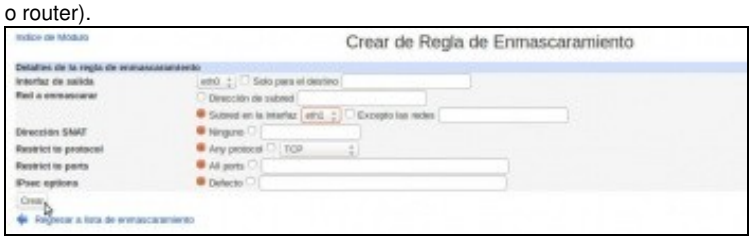
Aínda temos un problema por solucionar, xa que *uclient* non pode conectarse á rede pública; falta activar NAT no router *dserver2*.

- 

Farémolo no mesmo shorewall, no apartado de **Enmascaramento**.

- 

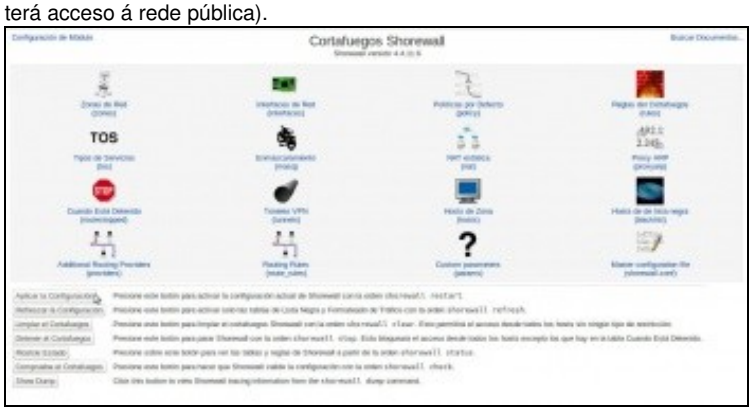
Agregamos unha nova regra de enmascaramento (xa que o imos facer é *enmascarar* unha rede privada sobre unha interface pública que ten o router).

- 

Como interface de saída seleccionamos *eth0*, que é a interface pública do router. Como rede a enmascarar, seleccionamos a subrede na interface *eth1*, que é a rede *lan*. O resto das opcións deixámolas como están.

- 

Engadimos outra regra igual pero para enmascarar a subrede na interface *eth2*, que é a rede *dmz* (esta rede tamén é privada e se non non terá acceso á rede pública).

- 

Aplicamos a configuración, e....

- 

Agora si!! Xa temos os dous problemas resoltos.

- Pero.... ¿todo isto non se sae un chisco do obxecto do curso? Ademais de xogar de forma máis profunda cos modos das interfaces de rede en VirtulBox, o router que acabamos de simular é precisamente o que VirtualBox implementa cando configuramos unha interfaz de rede dunha máquina virtual por NAT ou rede NAT. Nese caso, é VirtualBox o que fai de router, con NAT, pero ademais tamén implementa o servidor DHCP e DNS (nós ímonos quedar aquí). Con isto preténdese así que quede máis claro todo este proceso, e entender a virtualización da rede que implementa VirtualBox.

Reenvío de portos

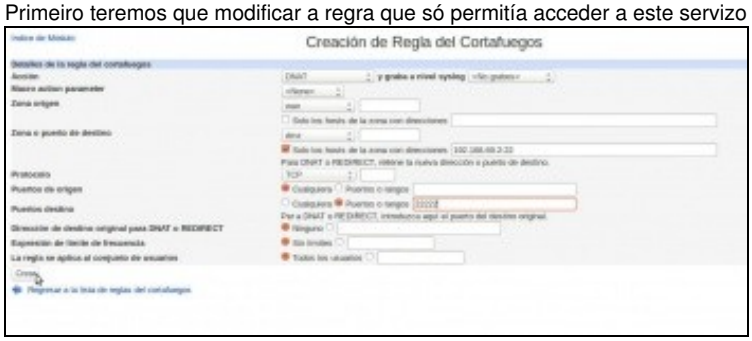
Para rematar, imos facer co noso router virtualizado a mesma función que VirtualBox permite coas tarxetas en modo NAT e rede NAT co reenvío de portos. Recórdese que isto permite acceder a un porto da máquina virtual mediante o reenvío dun porto da máquina *host*.

O equivalente neste caso sería redirixir un porto libre de *dserver2* a un servizo por exemplo duha máquina da *DMZ*. Imos coller o servizo de ssh de *dserver3*:

- Reenvío de portos coa MV Debian

- 

Primeiro teremos que modificar a regra que só permitía acceder a este servizo dende a zona *lan*, para poñer como zona orixe *calquera*.

- 

Creamos unha regra de tipo *DNAT* (Destination NAT), que redirixe o tráfico que veña da zona *wan* e vaia á zona *dmz*, concretamente ao porto 22 do equipo 192.168.69.2, co protocolo TCP os paquetes que reciba para o porto 22222 (este será o porto de *dserver2* que será redirixido ao servidor ssh de *dserver3*).

- 

Vista de como quedan as dúas regras. Aplicamos os cambios no shorewall.

- 

E xa podemos acceder dende un equipo da rede pública (por exemplo dende o *host*) por ssh ao porto 22222 da máquina *dserver2* usando a súa dirección IP pública. Unha vez dentro executamos o comando **ifconfig** para saber en que máquina estamos realmente; a dirección IP indica que estamos en *dserver3*.

-- Antonio de Andrés Lema e Carlos Carrión Álvarez --