

# **SERVIDOR WEB VIRTUAL PORTABLE**

**Ismael Luque Jiménez**

**Ángel Valera Motos**



## **DESCRIPCIÓN**

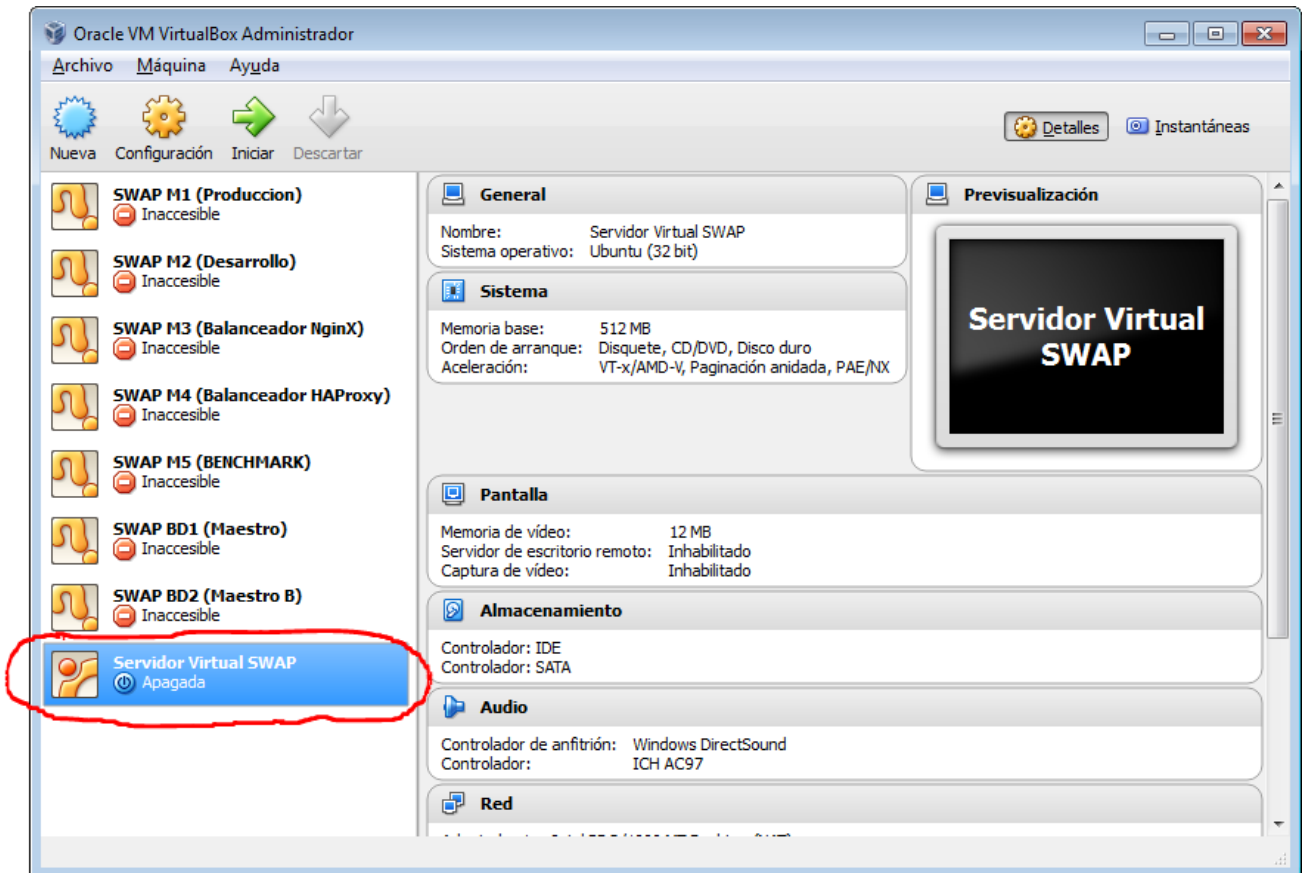
Mostrar cómo hacer una máquina virtual preparada para ponerla en cualquier hardware anfitrión y que comience a hacer de servidor web. La idea es que en cualquier máquina con conexión a internet y el software de virtualización se lanza el servidor virtualizado (un clon de nuestro servidor en producción) y se podría sustituir a nuestra máquina en pocos minutos. Para ello, contaremos cómo configurar dicha estructura usando diferentes tipos de software de virtualización, mostrando como demo el funcionamiento de uno de ellos.

## **ÍNDICE**

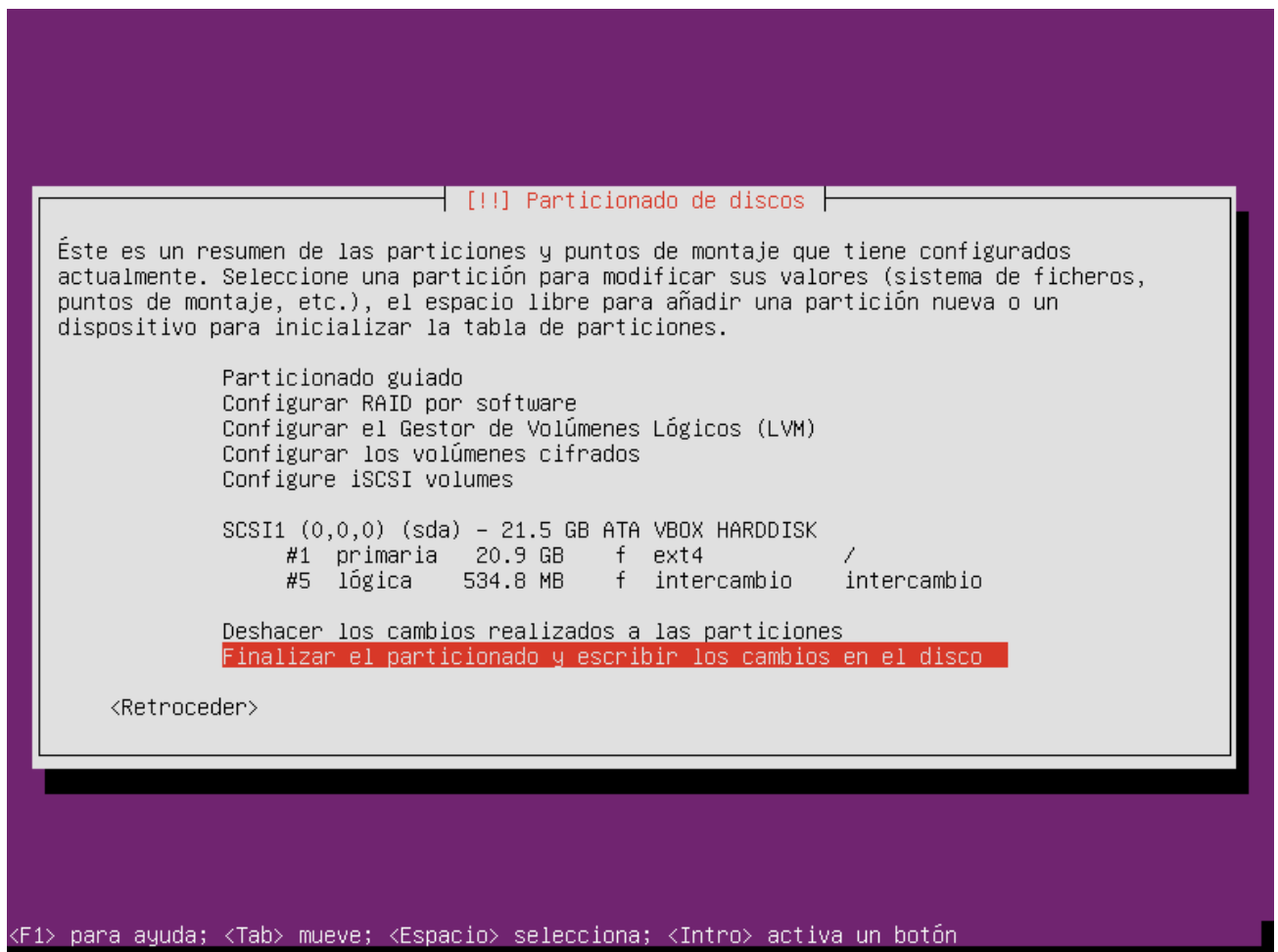
1. [Oracle VM VirtualBox](#)
2. [Windows Virtual PC](#)
3. [QEMU](#)
4. [VMware Player](#)
5. [REDIRECCIÓN DE PETICIONES](#)
  - a. [Windows](#)
  - b. [Linux](#)

## Oracle VM VirtualBox

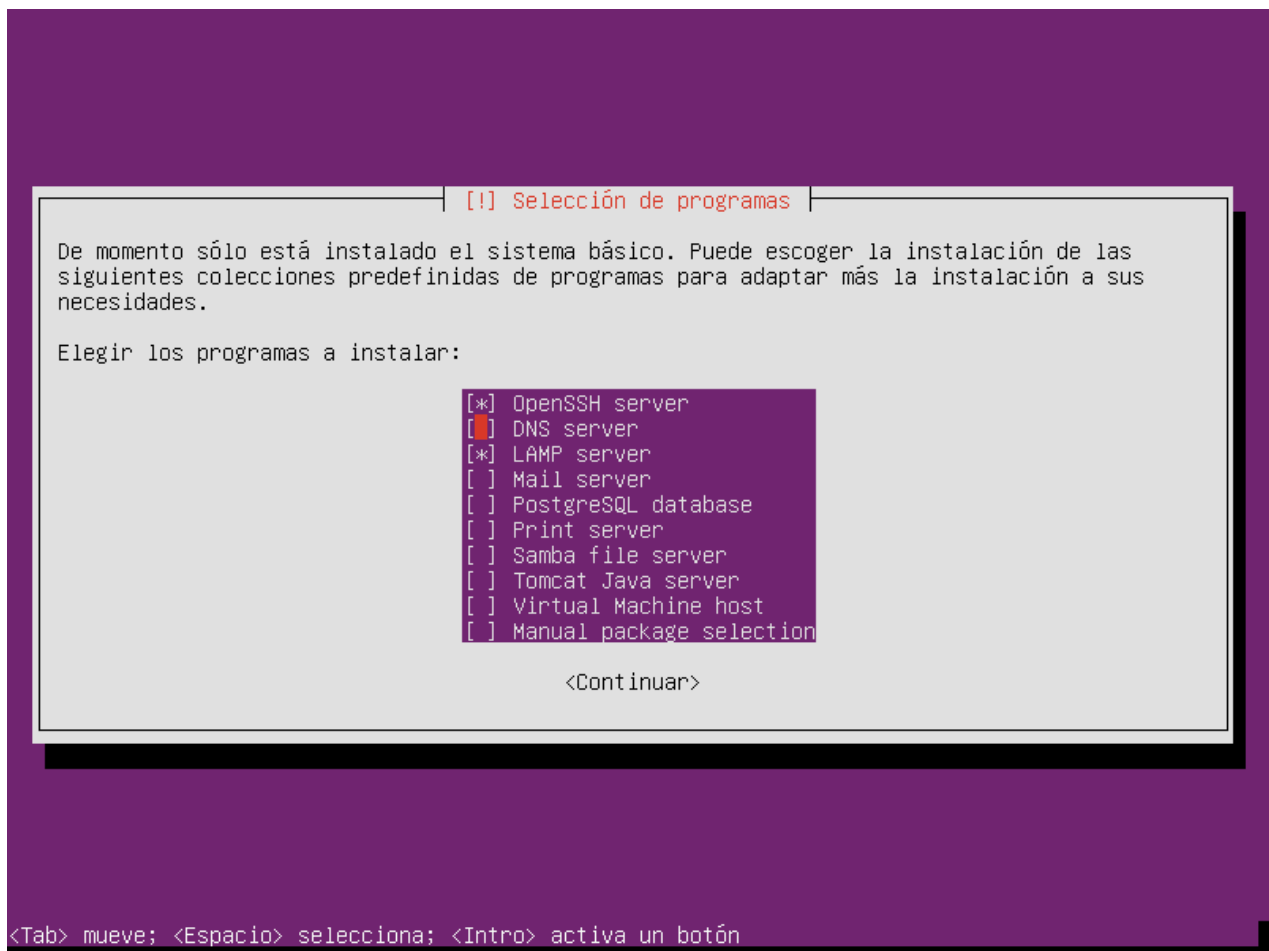
1. Creamos una nueva máquina virtual en VirtualBox seleccionando Linux => Ubuntu (32 bit). Después, elegimos la cantidad de memoria que creamos necesaria (en este caso con 512MB tenemos suficiente). A continuación, creamos un disco duro VDI del tamaño que queramos (es recomendable crear un disco bastante grande pero con almacenamiento reservado dinámicamente para prevenir posibles ampliaciones). Una vez hayamos hecho esto, tendremos una máquina virtual lista para instalar el sistema operativo. Así nos debe aparecer:



2. Procedemos a instalar el sistema operativo. Para ello, arrancamos la máquina virtual, elegimos la ISO del sistema operativo que queremos instalar (en este caso, Ubuntu Server 12 x86) y seguimos el proceso de instalación. Podemos utilizar el particionamiento guiado para crear una partición con el disco completo, o crear varias particiones para los distintos directorios:

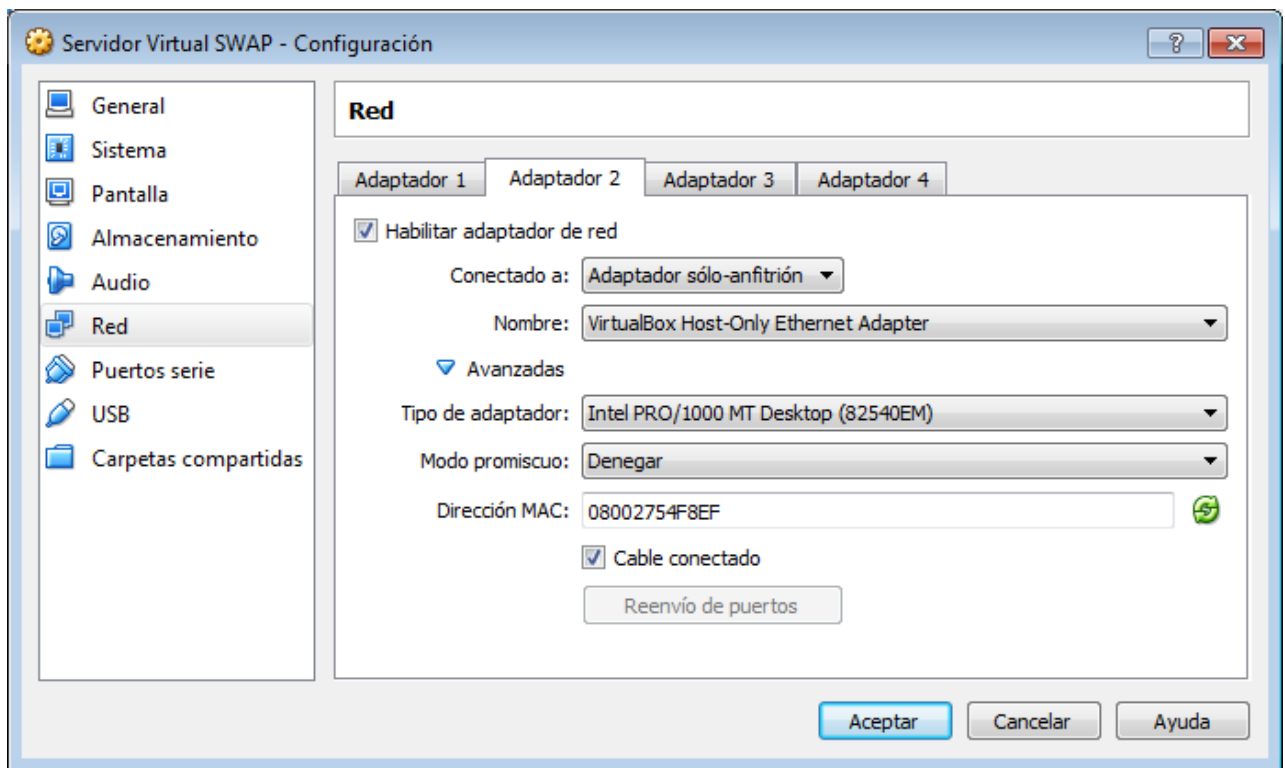


3. Seleccionamos actualizaciones automáticas para estar siempre a la última en rendimiento y seguridad. Posteriormente, nos aparecerá un menú para elegir los servicios a instalar. Instalamos sólo aquellos que necesitemos para el servidor web, en pos de reducir al mínimo la carga del sistema (en este caso, instalamos el paquete LAMP, que incluye todo lo necesario para funcionar como servidor web):



Adicionalmente, instalamos también el servicio OpenSSH para conectar con el servidor desde fuera. A continuación, tendremos que configurar el servicio MySQL seleccionando una contraseña para el usuario “root”. Finalmente, aceptamos la instalación de GRUB por si en un futuro necesitamos tener varios sistemas operativos instalados.

4. Antes de comenzar a utilizar el servidor, debemos añadir un adaptador de red para poder conectar la máquina virtual con la máquina anfitrión o con otras máquinas huéspedes, de forma que podamos acceder al servidor mediante SSH o transferir archivos mediante SFTP. Con la máquina apagada, añadimos un adaptador Host-Only en el menú de configuración de la máquina:



Esto añade el adaptador hardware virtual, pero VirtualBox no crea la interfaz de red necesaria para que la máquina virtual forme parte de la red. Esto debemos hacerlo nosotros modificando el archivo de configuración de las interfaces de red (/etc/network/interfaces) para definir la interfaz Host-Only y asignarle una IP:

```
GNU nano 2.2.6      Archivo: /etc/network/interfaces      Modificado
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# NAT network interface
auto eth0
iface eth0 inet dhcp

# Host-only network interface
auto eth1_
iface eth1 inet static
    address 192.168.56.100
    netmask 255.255.255.0
    network 192.168.56.0
    broadcast 192.168.56.255
```

5. Para comprobar que el servidor web funciona correctamente, creamos una página web de prueba para que se muestre al realizar peticiones:

```
<!DOCTYPE html>

<html>
  <head>
    <meta charset="UTF-8"/>
    <title>Plan de Estudios GII TI (ETSIIT)</title>
  </head>
  <body>
    <header>
      <h1>Plan de Estudios GII TI (ETSIIT)</h1>
      

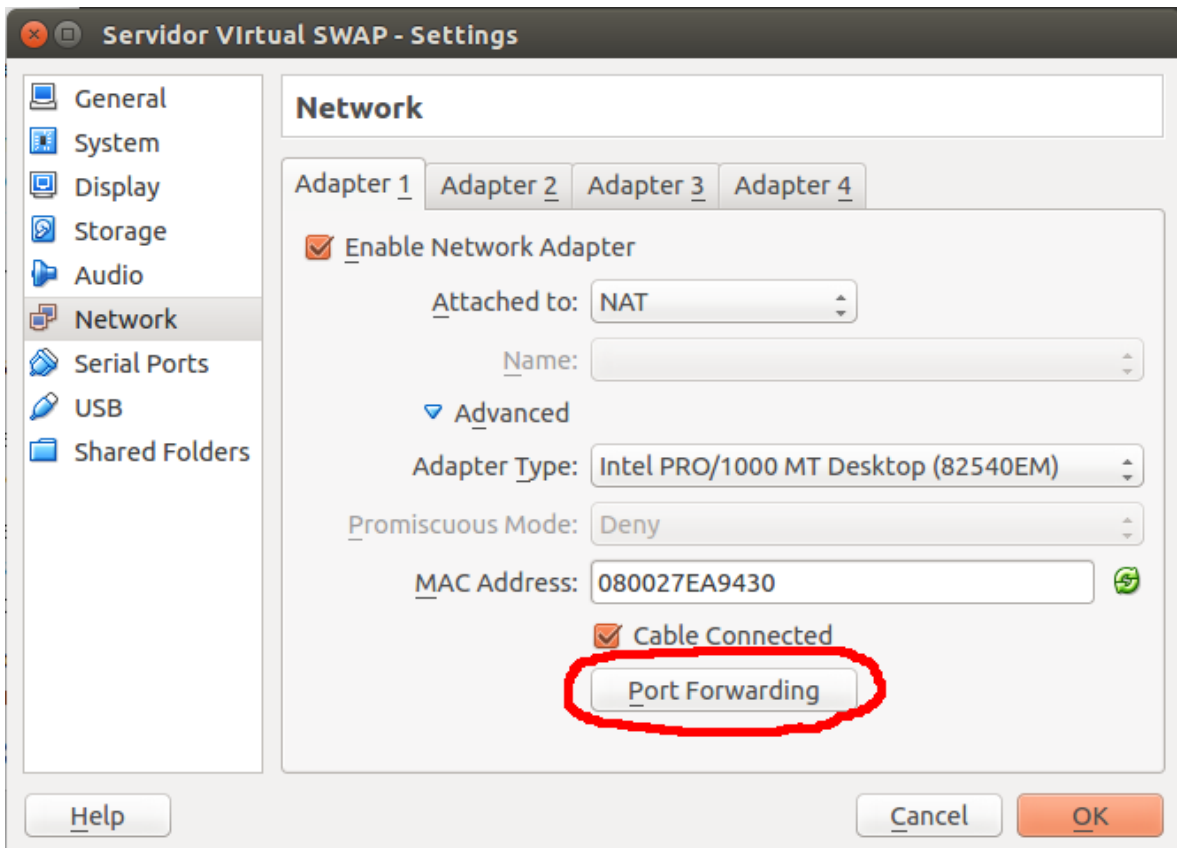
      <nav>
        <li><a href="#1">Primer Curso</a></li>
        <li><a href="#2">Segundo Curso</a></li>
        <li><a href="#3">Tercer Curso</a></li>
        <li><a href="#4">Cuarto Curso</a></li>
      </nav>
    </header>

    <dl>
      <dt>GII</dt>
      <dd>Grado en Ingeniería Informática</dd>
    </dl>
  </body>
</html>
```

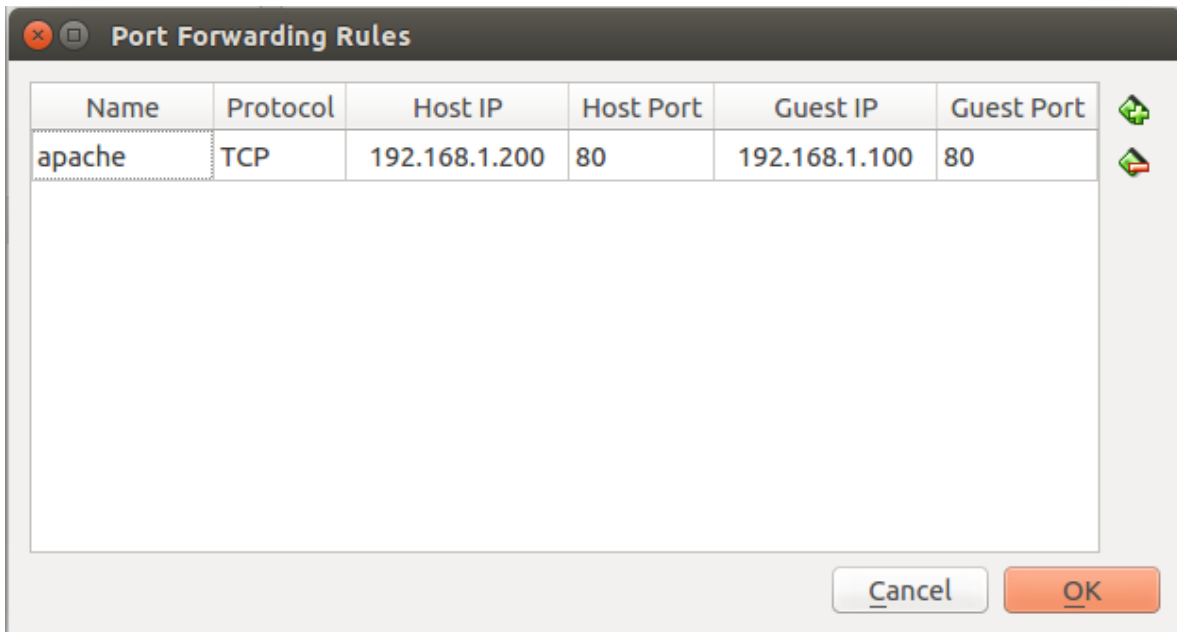
Comprobamos que el servidor está sirviendo correctamente la página realizando peticiones a su IP desde el navegador del host:



6. Finalmente, establecemos la conexión NAT. La idea es redirigir las peticiones que vayan a la IP de la máquina para que sean atendidas por el servidor virtual instalado en la misma. Podemos realizar esta configuración directamente desde VirtualBox. Dentro de la configuración de la máquina virtual, entramos en el apartado Red y accedemos a la opción Port Forwarding del adaptador NAT:



Dentro de esta tabla, introducimos una entrada con el servicio que queremos redirigir, los puertos del host y el guest y sus IP:



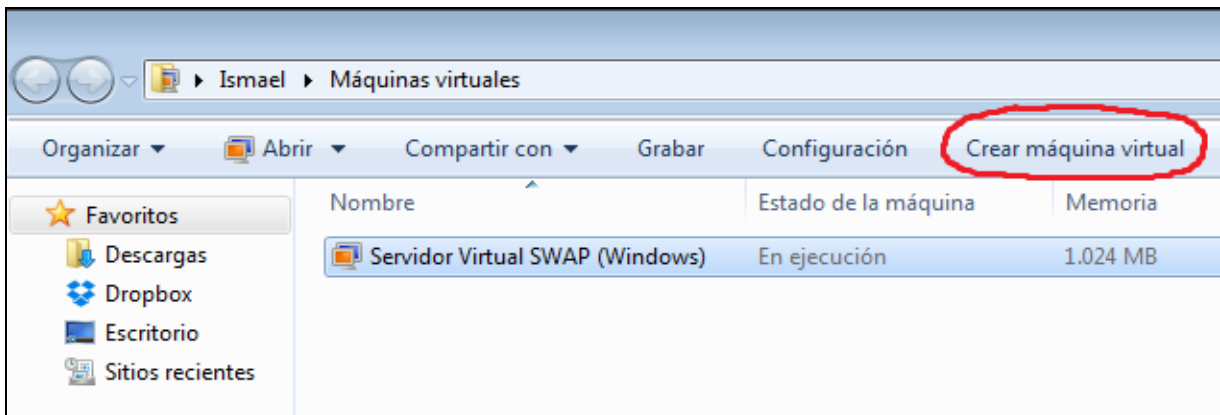
6. En caso de que esto no nos funcione, podemos utilizar una utilidad de línea de comandos de VirtualBox denominada VBoxManage. Introduciendo los siguientes comandos en una terminal con la máquina apagada (y escogiendo un adaptador PC Net), conseguiremos dicha redirección:

```
VBoxManage setextradata NOMBRE_MAQUINA  
"VBoxInternal/Devices/pcnet/0/LUN#0/Config/PROCESO/GuestPort" PUERTO_GUEST  
  
VBoxManage setextradata NOMBRE_MAQUINA  
"VBoxInternal/Devices/pcnet/0/LUN#0/Config/PROCESO/HostPort" PUERTO_HOST  
  
VBoxManage setextradata NOMBRE_MAQUINA  
"VBoxInternal/Devices/pcnet/0/LUN#0/Config/PROCESO/Protocol" PROTOCOLO
```

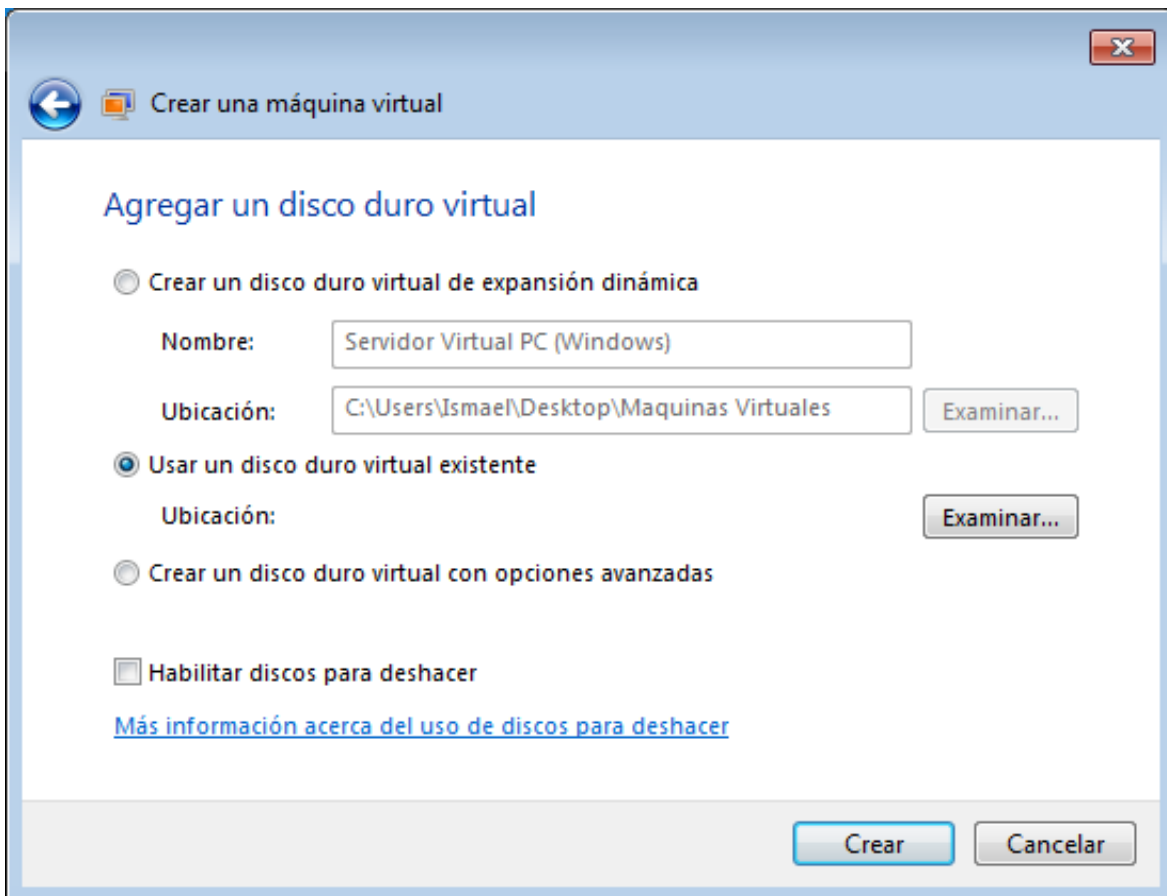


## Windows Virtual PC

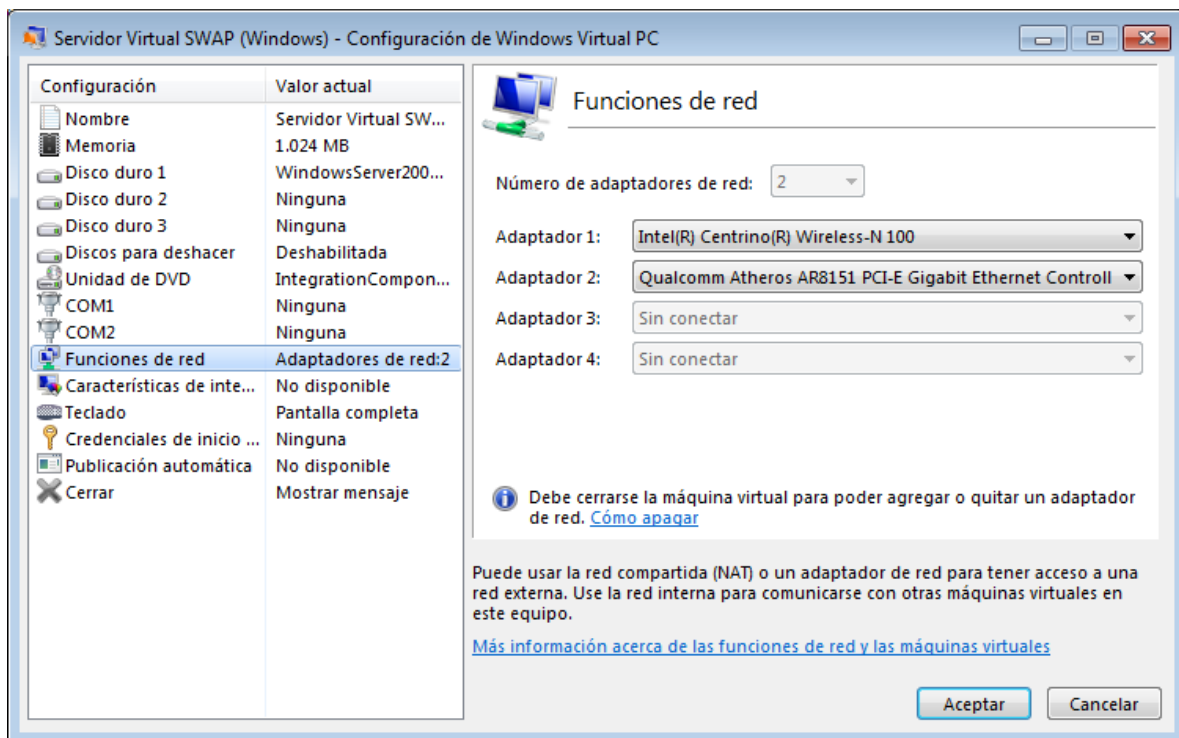
1. Instalamos Windows Virtual PC (denominado Microsoft Virtual PC en su versión anterior Windows XP e Hyper-V en su última versión para Windows 8) descargándolo de la página oficial de Microsoft. Viene en forma de actualización de Windows (KB958559), por lo que tras instalarlo debemos reiniciar la máquina para poder utilizarlo.
2. Ejecutamos Virtual PC y nos aparecerá una ventana del explorador de Windows situada en **C:\Users\USUARIO\Virtual Machines**, en la cual nos aparecerán la nueva funcionalidad Crear Máquina Virtual en la barra de herramientas. Seleccionamos dicha opción:



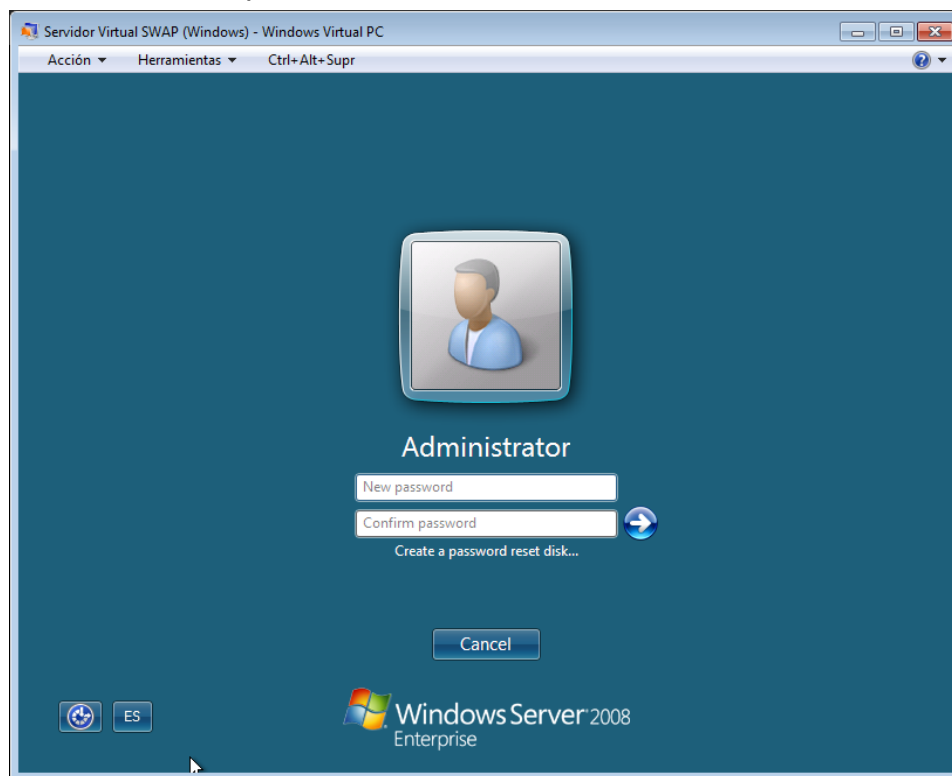
3. La creación de la máquina virtual es similar a los demás programas de virtualización, pero en este caso, asignamos el doble de memoria RAM puesto que Windows viene con entorno gráfico de serie, lo que consume más recursos. Otra diferencia es que la versión de Windows Server 2008 de 32 bits (la cual usamos porque no nos dejaba crear máquinas con la de 64 bits) viene en formato de disco duro virtual VHD, por lo que para cargarlo debemos seleccionar dicho disco duro a la hora de crear la máquina en lugar de cargar la ISO para instalar el SO:



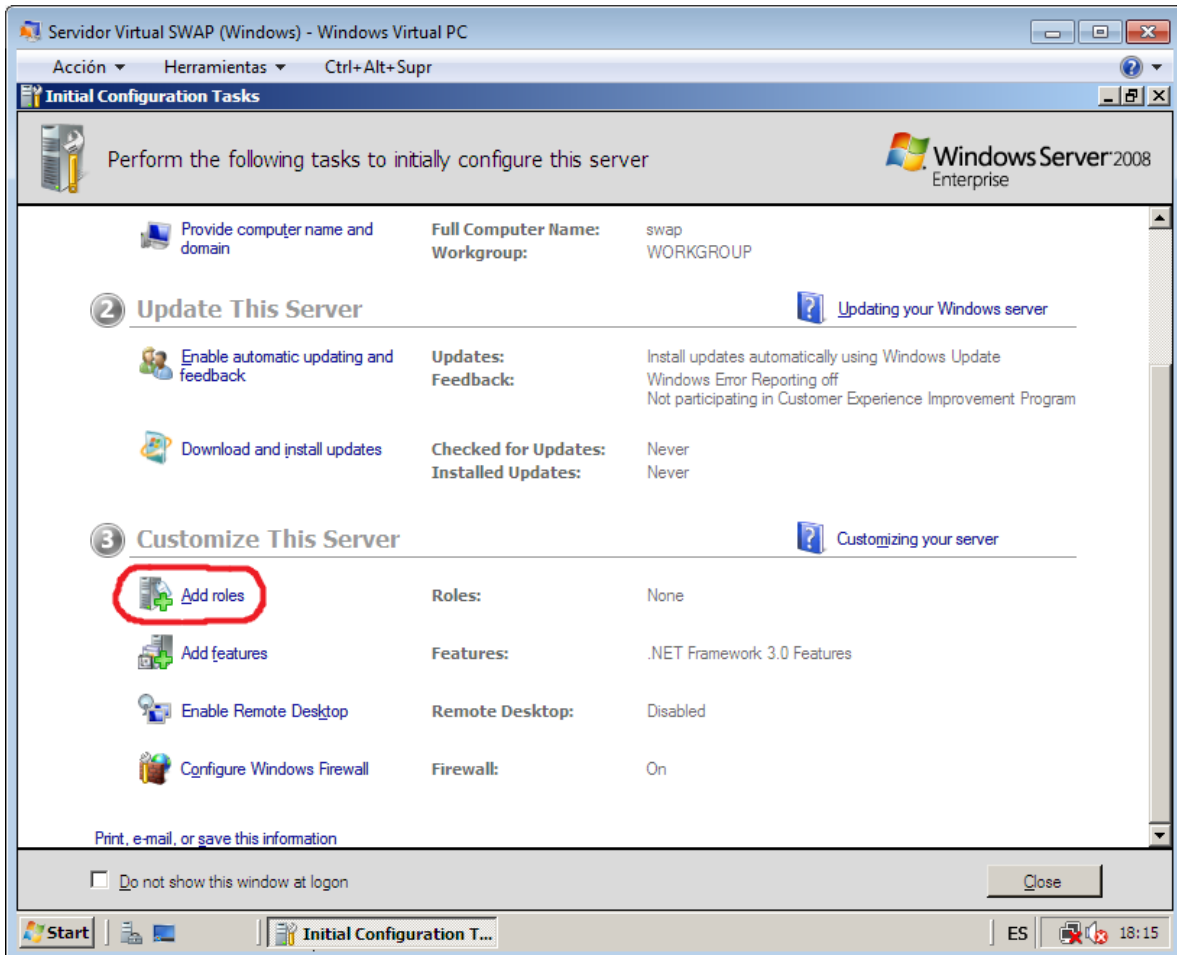
4. Antes de ejecutar la máquina virtual por primera vez, vamos a realizar la configuración de red. Sólo necesitamos que se conecte con la máquina anfitrión, puesto que ella se encarga de recibir las peticiones de Internet. Para ello, añadimos el adaptador físico que esté conectado a la red en la que queramos que se vea el servidor virtual. Esto se conoce como “bridge”:

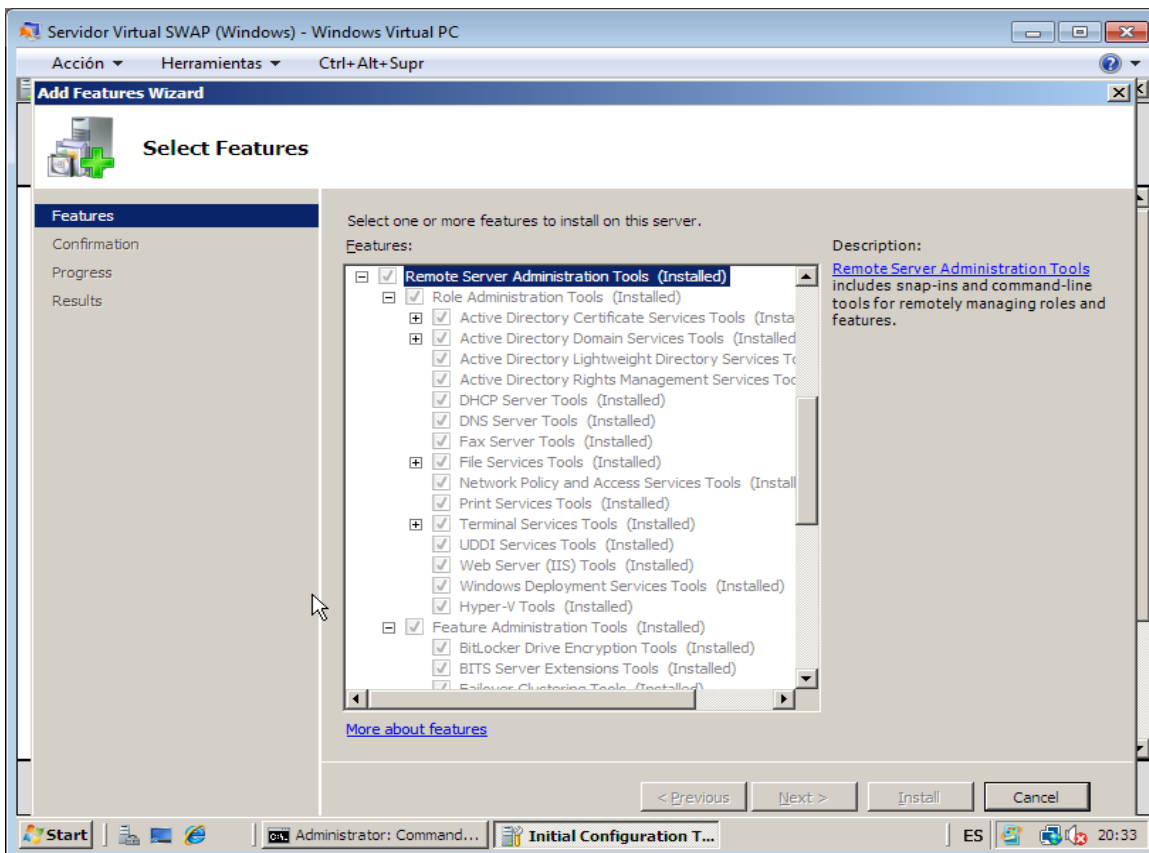
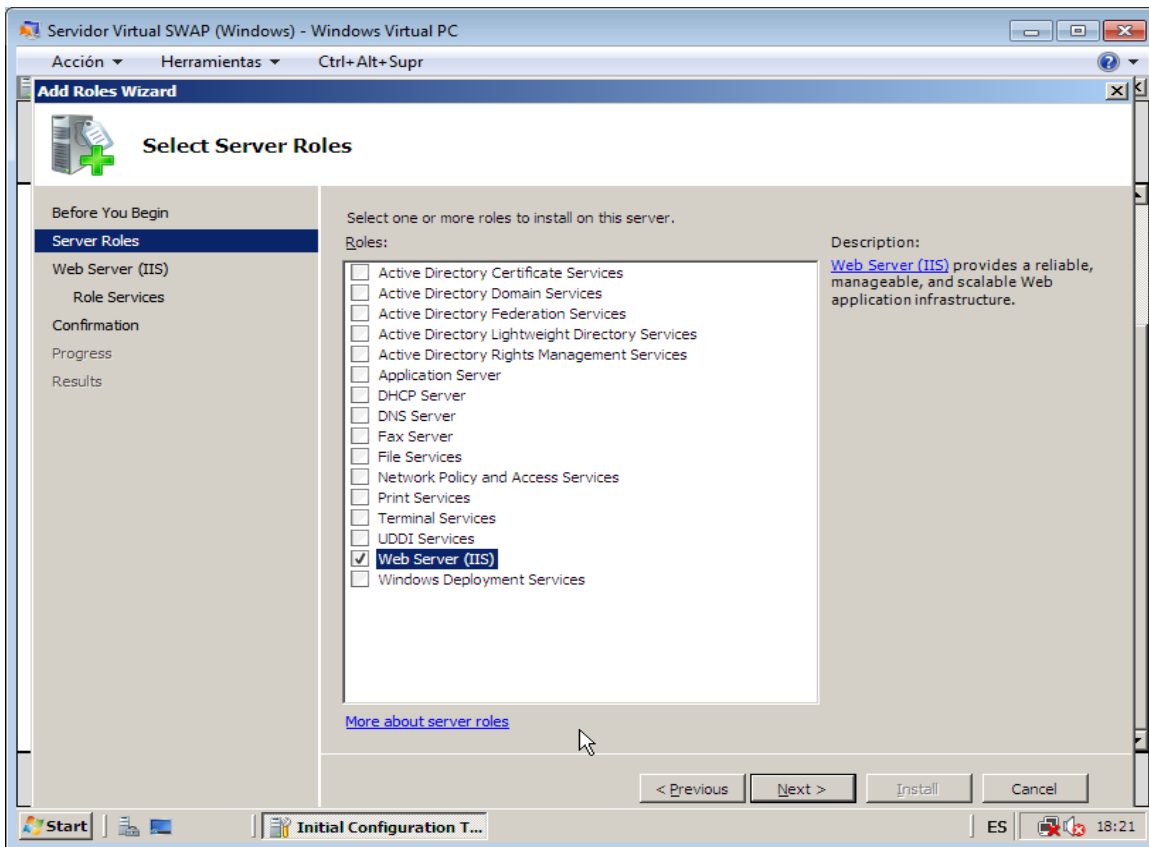


5. Arrancamos la máquina virtual y seguimos los pasos para completar la instalación. Nos aparecerá nuestra ventana de inicio de sesión:



6. Una vez iniciada la sesión, debemos instalar el software de servidor web IIS junto con todas las funcionalidades que necesitemos para nuestro servicio. Para ello, dentro de las Initial Configuration Tasks que nos aparecen al arrancar el sistema, seleccionamos la opción “Add Roles” y elegimos “Web Server (IIS)”, instalando las herramientas que queramos:





7. Una vez instalado IIS, comprobamos su funcionamiento accediendo a la IP de la máquina virtual y obteniendo la página inicial de IIS:



Al final del documento veremos cómo conseguir que las peticiones que vayan a la máquina física sean atendidas por el servidor de Virtual PC.

## QEMU

1. La creación de una máquina en QEMU es similar a VirtualBox. Creamos una nueva máquina virtual seleccionando Linux Distribution como SO:

**New Virtual Machine Wizard**

Welcome to the new virtual machine wizard. This wizard will help you create a new virtual machine. First please specify a virtual machine name and the type of virtual machine you wish to create.

**General**

Please enter a name for your virtual machine

Servidor Virtual SWAP

Please specify where you wish to store your virtual machine configuration and disk images:

Default VM Store

Please select the platform for the virtual machine.

Standard x86/x64 PC

Operating System

Linux Distribution

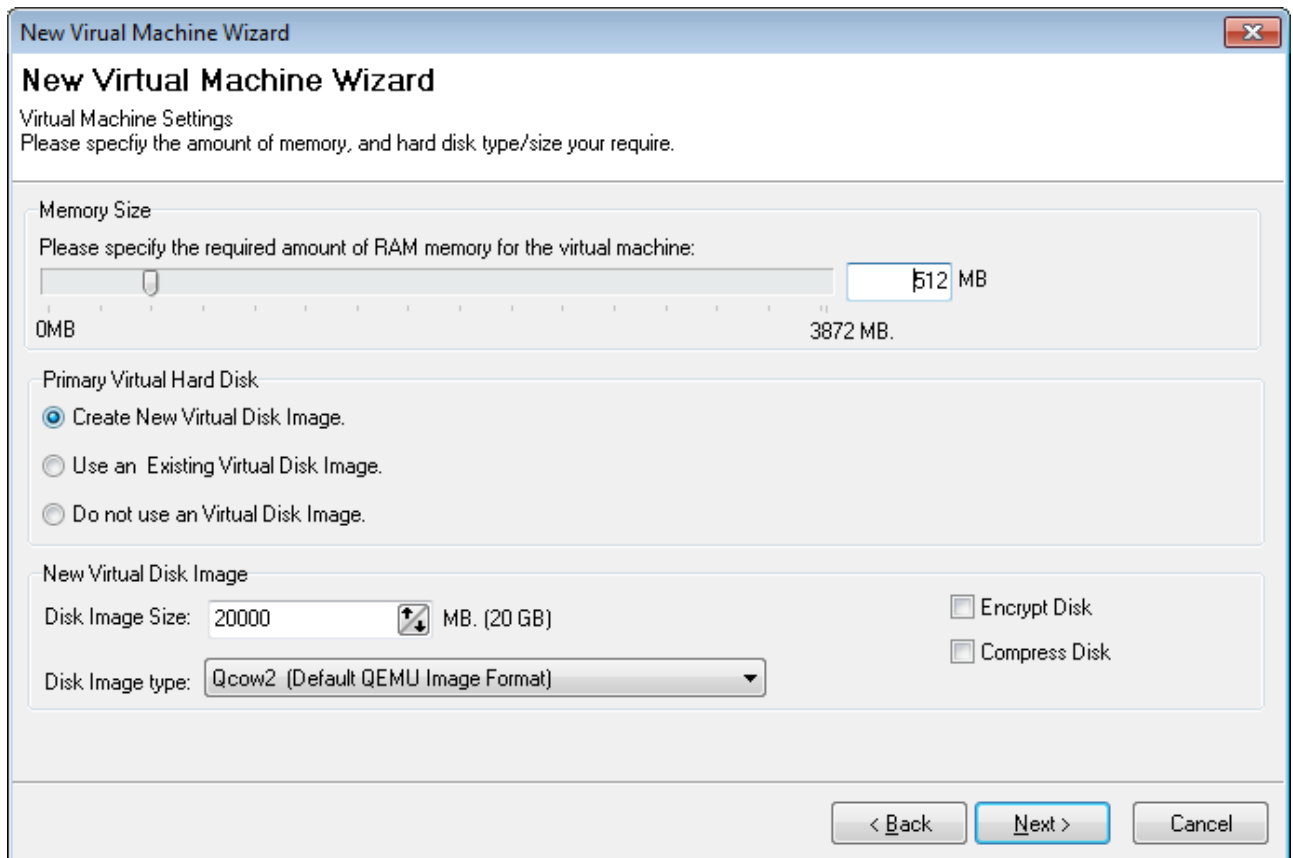
Qemu Manager supports multiple versions of QEMU, depending on the version you select will lock / unlock certain features of Qemu Manager. If unsure do not change this setting.

QEMU Version:

Qemu Manager QEMU 0.11.1

< Back Next > Cancel

Finalmente, elegimos la cantidad de memoria que creamos necesaria (en este caso con 512MB tenemos suficiente) y creamos un disco duro virtual del tamaño que queramos (por defecto utiliza el formato Qcow2 con un tamaño de 20GB, suficiente en este caso):



2. Hemos estado utilizando una interfaz gráfica para Windows denominada QemuManager. También existen GUIs de QEMU para Linux como Qemulator o AQEMU. Sin embargo, QEMU es una aplicación diseñada para línea de comandos y la mayoría de la documentación que se puede encontrar por Internet utiliza CLI, puesto que ofrece mayores opciones.

Para crear una máquina virtual como la anterior en Linux, debemos instalar los paquetes **qemu** y **kqemu-common**:

```
sudo apt-get install qemu kqemu-common qemu qemu-system-x86_64
```

A continuación, creamos un disco duro virtual de tipo qcow con 4 GB:

```
qemu-img create -f qcow swap.qcow 4G
```

Finalmente, arrancamos una máquina con dicho disco duro virtual y 512 MB de memoria RAM, cargando la ISO de Ubuntu Server para su instalación:

```
qemu -cdrom Downloads/ubuntu-15.04-server-i386.iso -hda swap.qcow -m 512 -boot d
```



3. La instalación de Ubuntu Server es igual que en los otros casos. Seguimos los pasos hasta que se complete la instalación y se reinicie el sistema. En esta ocasión, vamos a instalar el servidor web Lighttpd, que funciona de forma similar a Apache pero es mucho más ligero en consumo de recursos. Para instalarlo, introducimos en una terminal el siguiente comando:

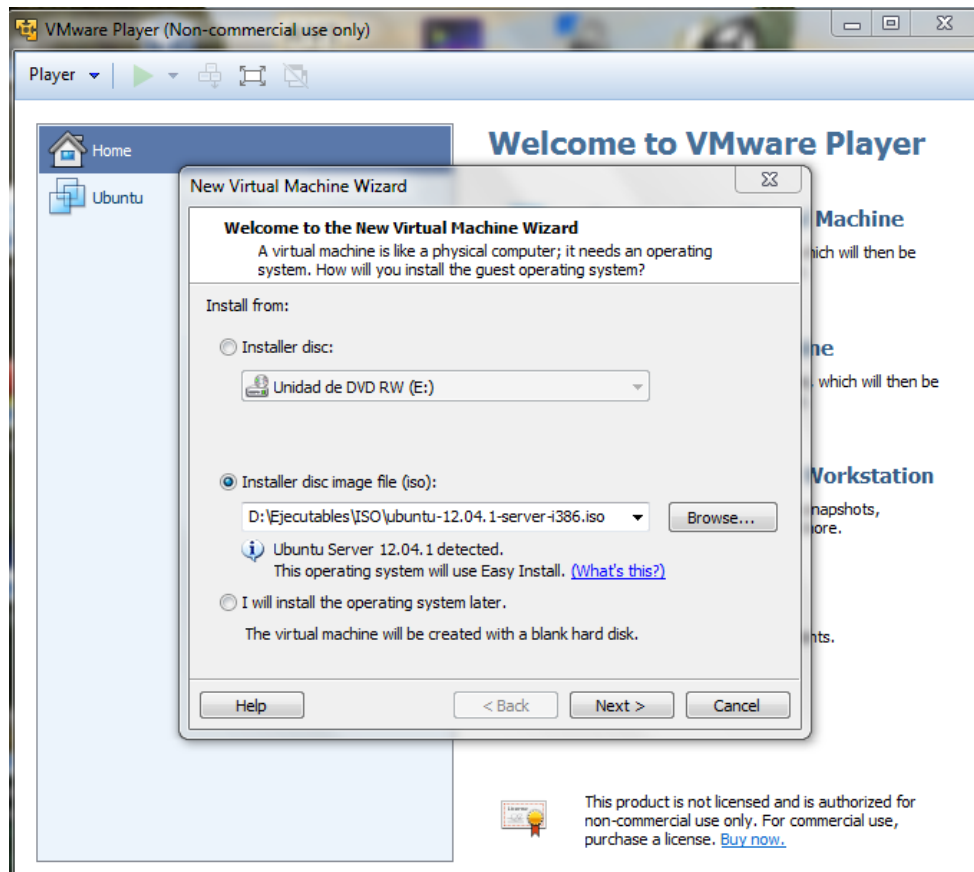
```
sudo apt-get install lighttpd
```

Al final del documento veremos cómo conseguir que las peticiones que vayan a la máquina física sean atendidas por el servidor virtual QEMU.

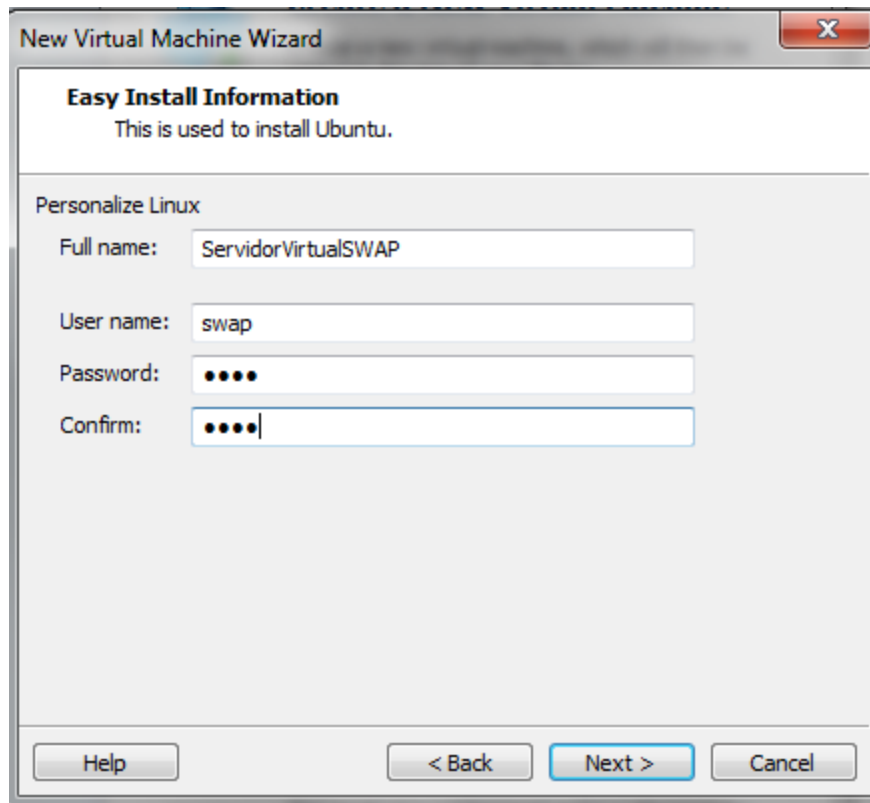
# VMware Player

Al instalar una máquina virtual en VMware debemos seguir los siguientes pasos:

1. Elegimos la iso del sistema operativo que vamos a instalar, en este caso un Ubuntu 12.04.

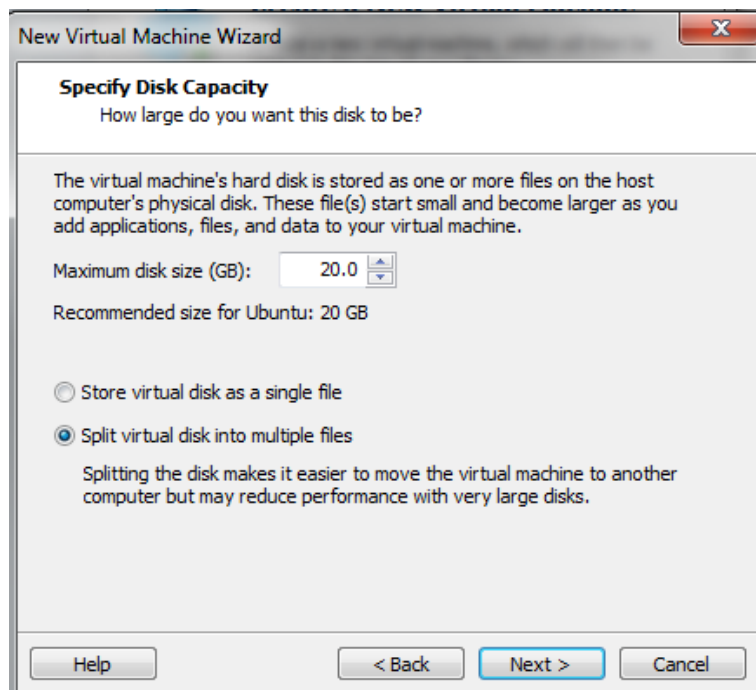


2. Ahora tenemos que elegir el nombre de la máquina, así como el nombre de usuario y contraseña.



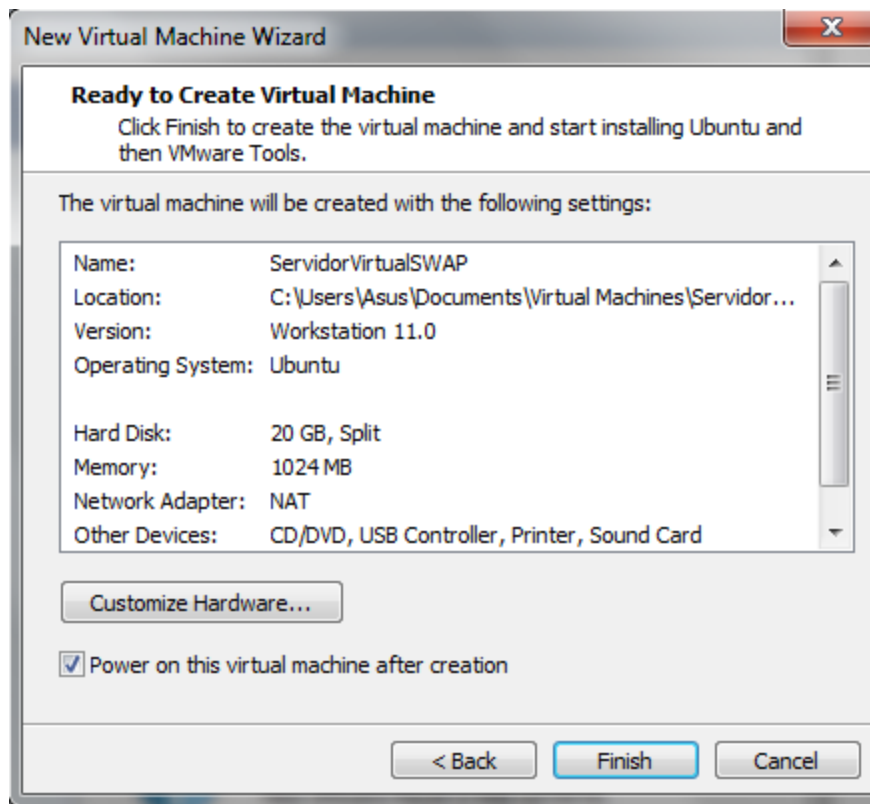
## PASO 2

3. Después de eso elegimos el tamaño del disco duro y si queremos que esté en un archivo solo o en varios.



## PASO 3

4. El siguiente paso consiste en personalizar el hardware que queremos que tenga nuestra máquina virtual.



#### PASO 4

5. Seguimos los pasos de instalación del sistema operativo.



#### PASO 5

```
Ubuntu 12.04.1 LTS ubuntu tty1
ubuntu login: _
```

FINAL

### **Cambiar configuración de teclado**

Al instalar Ubuntu como no podemos configurar nada durante el proceso de instalación, nos configuró el teclado al estilo americano, para cambiarlo tuvimos que cambiarlo manualmente una vez instalada la máquina, para ello entramos en el archivo:

```
sudo nano /etc/default/keyboard
```

y cambiamos el valor de "XKBLAYOUT" al código del idioma en nuestro caso "es". Guardamos el archivo e introducimos el siguiente comando.

```
sudo dpkg-reconfigure console-setup
```

Se nos abrirán una serie de menús de configuración, una vez terminado ya tendremos el teclado configurado como español.

Ahora pasamos a instalar el servidor Web en este caso hemos instalado nginx, para ello seguimos los siguientes pasos:

Lo primero que debemos hacer es importar la clave del repositorio de software:

```
cd /tmp/
wget http://nginx.org/keys/nginx_signing.key apt-key add
/tmp/nginx_signing.key
rm -f /tmp/nginx_signing.key
```

A continuación, debemos añadir el repositorio al fichero `/etc/apt/sources.list`. Para ello, podemos ejecutar las siguientes órdenes en el terminal:

```
echo "deb http://nginx.org/packages/ubuntu/ lucid nginx" >>
/etc/apt/sources.list

echo "deb-src http://nginx.org/packages/ubuntu/ lucid nginx" >>
/etc/apt/sources.list
```

Ahora ya podemos instalar el paquete del nginx:

```
apt-get update

apt-get install nginx
```

Hecho esto ya solo tenemos que colocar nuestras páginas webs en el siguiente directorio para empezar a servirlas:

```
/usr/share/nginx/html/
```

En cuanto a la conexión host-only, VMware ya la configura directamente sin necesidad de tener que hacerlo nosotros manualmente.

Comprobamos que efectivamente está sirviendo nuestras páginas webs.



Al final del documento veremos cómo conseguir que las peticiones que vayan a la máquina física sean atendidas por el servidor virtual VMware.

# REDIRECCIÓN DE PETICIONES

Una vez tengamos una máquina virtual (con cualquier software de virtualización o sistema operativo) que tenga conexión con la máquina física en la que se ejecuta, tenemos que redireccionar las peticiones web que lleguen a la máquina para que sean atendidas por el servidor virtual. Para ello, dicha máquina virtual debe estar conectada con un adaptador host-only o un adaptador bridge, de forma que comparta alguna red con el host y puedan verse entre ellos.

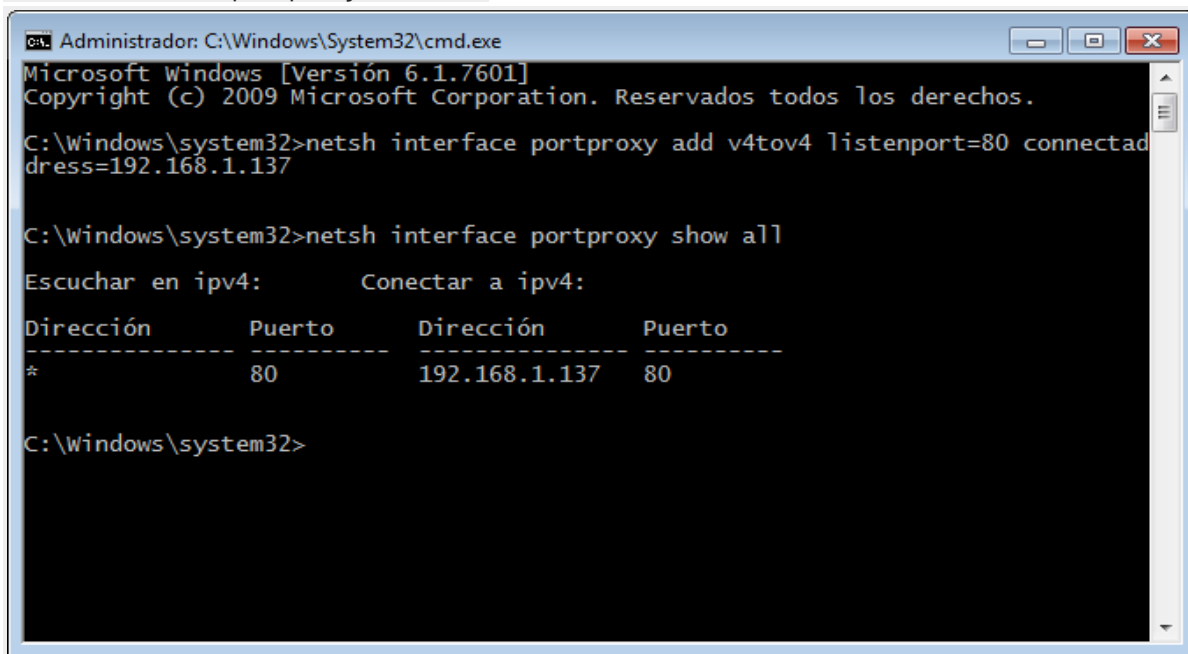
## Windows

Para realizar esta traducción de direcciones, utilizamos la herramienta Netsh, que ofrece varias opciones para la configuración de la red. Abrimos un CMD como administrador e introducimos el siguiente comando:

```
netsh interface portproxy add v4tov4 listenport=80 connectaddress=IP_SERV_VIRTUAL
```

Sustituyendo en cada caso la IP por la de la máquina virtual en cuestión. Para comprobar que hemos introducido la regla correctamente usamos este comando:

```
netsh interface portproxy show all
```



```
Administrator: C:\Windows\System32\cmd.exe
Microsoft Windows [Versión 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Reservados todos los derechos.

C:\Windows\system32>netsh interface portproxy add v4tov4 listenport=80 connectaddress=192.168.1.137

C:\Windows\system32>netsh interface portproxy show all

Escuchar en ipv4:      Conectar a ipv4:
Dirección      Puerto      Dirección      Puerto
-----
*              80          192.168.1.137  80

C:\Windows\system32>
```

## Linux

Para realizar este direccionamiento en Linux, una forma sencilla es utilizar un balanceador de carga que redirija todas las peticiones al puerto 80 hacia una misma IP (la de la máquina virtual). En este caso, hemos usado HAProxy puesto que es algo más sencillo de configurar que otras alternativas como NginX.

1. Primero, instalamos el servicio HAProxy en el host con este comando:

```
sudo apt-get install haproxy
```

2. Para que se lleve a cabo el reenvío de peticiones desde el host hacia el guest, debemos modificar el archivo de config. /etc/haproxy/haproxy.cfg para añadir las siguientes líneas al final:

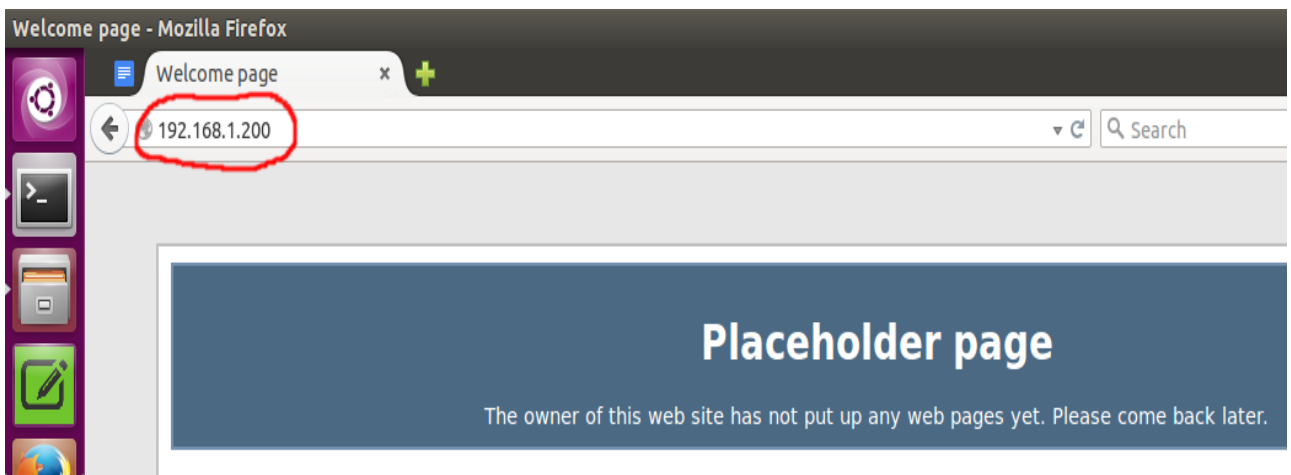
```
frontend http-in
    bind *:80
    default_backend servers

backend servers
    server swap 192.168.1.100:80 maxconn 32
```

3. Finalmente, activamos el servicio HAProxy para que comience a funcionar:

```
/usr/sbin/haproxy -f /etc/haproxy/haproxy.cfg
```

4. Para comprobar que todo funciona correctamente, lanzamos una petición HTTP a la IP de nuestra máquina física y observamos cómo aparece la página de inicio del servidor virtual:







# Referencias



- <https://carloscamposfuentes.wordpress.com/2014/03/12/configurar-red-interna-virtualbox>
- <http://es.fregly.com/question/982/c%C3%B3mo-cambiar-la-distribuci%C3%B3n-del-teclado-en-ubuntu-server>
- [http://foro.elhacker.net/tutoriales\\_documentacion/manual\\_instalar\\_una\\_maquina\\_virtual\\_sobre\\_qemu\\_en\\_windows-t125308.0.html](http://foro.elhacker.net/tutoriales_documentacion/manual_instalar_una_maquina_virtual_sobre_qemu_en_windows-t125308.0.html)
- [https://github.com/isma94/SWAP2015/blob/master/P2\\_ISE\\_Guion.pdf](https://github.com/isma94/SWAP2015/blob/master/P2_ISE_Guion.pdf)
- <https://github.com/isma94/SWAP2015/blob/master/Practica3/GuionPractica3.pdf>
- [https://github.com/AngelValera/swap1415/blob/master/practicas/practica3/P3\\_Angel-Valera-Motos.md](https://github.com/AngelValera/swap1415/blob/master/practicas/practica3/P3_Angel-Valera-Motos.md)
- [http://aztcs.org/meeting\\_notes/winhardsig/virtualmachines/WindowsVirtualPC/Virtual\\_Networks\\_in\\_WinVirtualPC.pdf](http://aztcs.org/meeting_notes/winhardsig/virtualmachines/WindowsVirtualPC/Virtual_Networks_in_WinVirtualPC.pdf)
- <http://www.archivoslog.es/2010/07/servidor-virtualizado-con-conexion-nat-virtualbox-y-windows/>
- <http://stackoverflow.com/questions/21365681/redirecting-my-local-address-to-my-virtual-machine>
- <http://www.howtogeek.com/122641/how-to-forward-ports-to-a-virtual-machine-and-use-it-as-a-server/>
- <https://www.linode.com/docs/websites/lighttpd/lighttpd-web-server-on-ubuntu-12-04-precise>