

---

# Servidor Web 2010

---

## Linux Ubuntu Server

---



**project2010**  
the web server project by forat.info

---

Manual original, creado por **David Sanchez Cantero**.  
Mas conocido en la red por **Forat**.

FORAT@GMAIL.COM

---

Ya tenemos nuestro servidor web bajo Linux Ubuntu Server preparado para servir contenidos hacia Internet o hacia vuestra red local. Para los que esperaban que el manual se completara para comenzar a montarlo ya pueden hacerlo porque con este ultimo capitulo terminamos. A continuación los contenidos del Project 2010 ...

---

## Introducción

- Vol 1 ( [Como instalar Linux Ubuntu Server 9.10](#) )
  - Vol 2 ( [Configuración de Red y manejo remoto vía OpenSSH con SSH y SFTP](#) )
  - Vol 3 ( [Como instalar LAMP + PhpMyAdmin](#) )
  - Vol 4 ( [Abrir y redirigir puertos desde nuestro Router](#) )
  - Vol 5 ( [Encontrando nuestro servidor desde Internet con No-Ip](#) )
  - Vol 6 ( [Servidor web Apache y su VirtualHost con NoIp](#) )
  - Vol 7 ( [Dominios comerciales + VirtuaHost en Apache](#) )
  - Vol 8 ( [Estadísticas web Open Source con Piwik](#) )
  - Vol 9 ( [Estadísticas sobre nuestro Hardware con PhpSysInfo](#) )
- 

Recordemos que este servidor web se maneja remotamente desde una Terminal y podemos ver desde la web los contenidos que carguemos en el ya sean archivos como paginas web, [blogs](#) y demás contenidos que puedan interesarnos. Además nos ofrece unas cuantas direcciones web. Se trata de las estadísticas ya sean por parte de nuestro [Hardware con Phpsysinfo](#) o por el trafico que generen nuestras webs con [Piwik](#). Además la creación de las bases de datos MySQL también tienen acceso desde la web con [phpMyAdmin](#).

## Sipnosis

En un principio pensé en incluir un panel de control gráfico para poder manejar el **servidor web** pero yo soy de los que piensa que es preferible instalar lo justo y malgastar los menos recursos posibles a la hora de instalar programas. Quizás en el [servidor web con Linux Ubuntu Server](#) que estamos montando no sea necesario ahorrar muchos [recursos de la maquina](#) ya que yo mismo tengo 2 Gb de ram y no es un problema porque no me cuestan dinero cada mes. Cuando tenemos un **servidor remoto** tanto **dedicado** como un **VPS** es muy importante ahorrarnos los recursos posibles ya que pagaremos más cada mes a medida que los vayamos necesitando.

Es mejor acostumbrarse a hacer esto en una maquina casera cuando comenzamos a trabajar con servidores simplemente porque cuando nuestras webs crezcan y necesitemos mas ancho de banda en nuestra ADSL es posible que tengamos que alquilar al mes un servidor remoto en alguna de las empresas que existen. Cuando nuestra experiencia llegue a ese punto tendremos por la mano la administración y el **manejo del servidor desde la Terminal** prescindiendo así del alquiler y licencias de programas como el **Cpanel** y otros administradores gráficos.

Espero que os sea de utilidad y lo aprovechéis para aprender y pasar un rato divertido frente a vuestras maquinas haciendo lo que nos gusta.

## Introducción

Bueno chicos se que hace tiempo que estáis esperando un manual de los grandes para **montar un servidor Web** algo mas completo que los proyectos sobre **servidores Web** fabricados anteriormente tanto en [Linux Debian](#) como en [Linux Ubuntu Server](#). Para montar estos servidores no es que necesite quince días para tenerlo acabado pero como comprenderéis después del [punto y coma que tuvimos en forat.info](#) y del [por fin llego el gran día](#) tendremos que ir leyendo poco a poco los capítulos hasta llegar al manual final donde encontraremos un índice con todo el proyecto al completo mas que nada por el tiempo que requiere ir fabricando los artículos ...



Los que soléis leer mis artículos seguramente os disteis cuenta que la web sufrió una especie de bache en el artículo [Defcon 2 entre bastidores](#) respecto a la publicación de artículos y caídas de la misma. El motivo de este atasco no fue otro que la migración de todos mis sitios hacia otro servidor. Yo mismo alquile un servidor en una empresa y me lleve todos mis contenidos hacia un nuevo servidor una vez acabé de montarlo. Ahí es cuando sufrimos un atasco en el servidor por falta de recursos de la maquina y una configuración por defecto nada tuneada para que funcionase bien y aguantara como un campeón todo el tráfico que generan mis sitios en internet.

A día de hoy todo funciona como la seda y es por este mismo motivo por el que he decidido volver a montar un nuevo **servidor web** actualizado y en forma de capítulos para que todo el que lo necesite pueda montarse un servidor modesto y con lo necesario para poder tener varios sitios web alojados con sus respectivas bases de datos **MySQL** y dominios .com .net .org, etc ... Durante este tiempo muchos de vosotros me habéis pedido muchas cosas que a cualquiera de los proyectos que tengo en [Project.forat.info](#) para **servidores web** le faltaban como el **servidor DNS** o la **configuración** de los **VirtualHost** para poder alojar varias webs.

Antes de que comencéis a montarlo me gustaría que supieseis que vamos a montar un **servidor web** modesto pero cañero ... ¿ Que quiere decir el loco de [Forat](#) con esto ? Pues simplemente quiero decir que vamos a montar todo lo que necesitamos pero no vamos a explicarlo todo, mas que nada porque si tuviera que explicar todo lo que le he hecho al servidor web que yo tengo contratado no acabaríamos nunca y yo no me iría a dormir nunca a la cama y a mi mujer eso no le gustaría y de [mi hijo](#) no digo nada porque el pobre todavía no entiende de nada mas que de leche y de pañales 😊

En fin, este nuevo **servidor web** que vamos a montar podrá ofrecer los siguientes servicios hacia o

desde internet para que podáis montarlo tanto en vuestra casa como en un servidor remoto contratado en alguna empresa ...



## ¿ Que será capaz de hacer nuestro Servidor Web ?

### 1

**Servidor Web** el cual alojará nuestros sitios web y estará preparado para servirlos al exterior. Este servidor Web estará preparado para servir un sitio web o los que queramos ya que vamos a montar un servidor multidominio al cual podremos apuntar subdominios gratuitos como pueden ser los de **No-IP** o los de **DynDNS** o de pago como pueden ser los **.com** , **.net** , **.org** , **.es** , etc ...

### 2

**Servidor OpenSSH** con el que podremos **controlar nuestro servidor remotamente** como si de tu **Terminal** se tratase. Además con **OpenSSH** podremos subir y bajar archivos como si de una carpeta mas de nuestro sistema operativo de escritorio **Linux Ubuntu** se tratase facilitando la subida y bajada de archivos de una manera totalmente segura y sencilla prescindiendo así de [instalar un servidor FTP](#).

### 3

Podremos **encontrar nuestro servidor desde Internet** gracias al servicio gratuito que nos ofrece [no-ip.com](#). Gracias a un subdominio que en esta empresa registraremos teniendo así una dirección Web desde que apuntará hacia nuestro servidor tengamos **IP dinámica** o no. Este podremos usarlo para conectarnos remotamente vía **SSH**, subir y bajar archivos vía **SFTP** y por supuesto **alojar** nuestra primera web. Todo desde tu dirección web unica despreocupándonos de si nos cambia la **IP del Router** o no.

## 4

**Sistema de estadísticas** con el que descubriremos todo sobre el funcionamiento y recursos de nuestra maquina en cuanto al [Hardware](#) se refiere además de un completo **sistema de estadísticas** con el que veremos al detalle todo sobre el **tráfico** de la web que queramos controlar así sabremos cosas como cual es la palabra con la que mas acceden desde los buscadores o simplemente cuantas visitas hemos tenido en una o en todas las webs en general que tengamos controladas.

Detrás de todos estos servicios hay un conjunto de programas y configuraciones necesarias como pueden ser el **servidor DNS** o el **VirtualHost del servidor web**.

## Situación

En un principio este **servidor Web** bajo **Linux Ubuntu Server** que voy a fabricar es algo parecido al que he montado en una empresa remota con [Linux Debian](#) la cual se ocupa de proveer mi servidor de ancho de banda y energía suficiente para aguantar todo el tráfico de mis sitios web.

## ¿ IP Fija o Dinámica ?

En esta ocasión voy a montarlo en mi casa y para hacer esto en casa con nuestras **ADSL** hay que contar con algunos aspectos con los que no contamos cuando contratamos un servidor externo. Si contratamos un servidor externo y montamos allí nuestras webs no necesitamos configurar nuestro **Router** para que sepa hacia que ordenador de nuestra red debe apuntar cuando un usuario desde **Internet** quiere acceder nuestro puerto **80**, ni tampoco contamos con que nuestras **ADSL** tienen una **IP dinámica**. En mi caso con el servidor de pago externo tengo dos IP's fijas y una vez montado el **servidor DNS** no tengo ningún problema al redireccionar todos mis dominios hacia mi servidor. A diferencia en nuestras casas con las **ADSL** que tenemos no solemos tener ni una ni dos **IPs fijas** si no que la tenemos solemos tener **dinámica** y tan solo una. Esto es un factor a tener en cuenta a la hora de poder tener varios **dominios alojados en un servidor** con una **ADSL** casera sin **IP fija**.

Próximamente comenzaremos con el montaje de este **servidor web** que nos servirá para alojar nuestras propias webs, nuestras **DNS**, nuestros archivos remotos y sobre todo a encontrar nuestra maquina desde Internet.

## Requisitos mínimos

Para montar este **servidor web** vamos a usar la distribución **Linux Ubuntu Server** en su versión **9.10 ( Karmic Koala )**. Esta requiere como recursos mínimos del ordenador donde vayamos a instalarla que tenga un procesador con velocidad mínima de 100 Mhz, 128 Mb de memoria ram y unas 500 Mb de disco duro para la instalación base del sistema operativo. Una vez montado el servidor al completo necesitaremos mas memoria ram, mas procesador y mas disco duro que los requisitos mínimos que el sistema operativo requiere ya que instalaremos varios programas y recibiremos tráfico con lo que el consumo de recursos aumentará. Si queréis una recomendación mía yo instalaría este servidor web en una maquina con al menos un procesador de 800 o 1000 Mhz con 512 Mb o 1 Gb de ram y disco duro de al menos 5 Gb ahora podéis probar con maquinas mas pequeñas pero si algo no funciona como debería siempre podremos acusar alguno de nuestros componentes informáticos.

En mi caso voy a usar el ordenador que usaba como ordenador de escritorio antes de montarme el [ordenador del siglo 21](#) que ya todos conocéis. El ordenador que voy a usar es un Pentium 4 a 3.0 Ghz con 2 Gb de ram y 500 Gb de disco duro.

Como podéis ver tengo maquina suficiente como para cumplir los requisitos mínimos de **Linux Ubuntu Server**. Recordad que siempre podemos **montar un servidor web** como el que vamos a

montar en un ordenador con los requisitos mínimos pero siempre es aconsejable usar lo máximo en hardware que tengamos a mano ya que cuando mas rápidos y mas amplios sean nuestros recursos mejor y mas rápido será capaz de funcionar nuestra maquina.

Básicamente en lo que refiere al **Hardware** un **servidor web** si hay algo que necesita mas que otras cosas son las siguientes que he enumerado por importancia empezando por la memoria ram como recurso de mas importancia ...

- 1 – [Memoria Ram](#)
- 2 – [Procesador](#)
- 3 – [Disco Duro](#)
- 4 – [Tarjeta de Red](#)
- 5 – [Fuente de alimentación](#)
- 6 – [Placa base](#)
- 7 – [Tarjeta gráfica](#)

En la ultima posición de importancia de componentes nos encontramos con la *Tarjeta gráfica*. Esta no es importante simplemente porque tan solo vamos a usarla para la instalación del sistema operativo **Linux Ubuntu Server**. Una vez completado el siguiente manual ya podremos desinstalarla y controlar el servidor por red desde otro ordenador que tengamos conectado a nuestra red de área local vía **OpenSSH**. Esto lo explicaremos durante el proyecto así que si os interesa montaros una maquina de estas y aprender muchas cosas sobre **servidores web** y **GNU/Linux** os invito a que os montéis uno con los manuales que iré publicando hasta completar el servidor.

## El ancho de banda

La finalidad de este proyecto es la de que podamos montar nuestro propio **servidor web en nuestra casa** detrás de nuestro **Router ADSL**, este es un factor con el que debemos contar ya que si alojamos paginas que tengan mucho trafico seguramente quedaremos saturados. Dependiendo del ordenador que tengamos seguramente quedaremos saturados antes por el poco ancho de banda de subida que tenemos que por sobrecarga en la maquina.

Mi **ADSL** de casa es una ADSL de 3 Mb o por lo menos así la venden, en realidad esta me ofrece una velocidad de descarga de 300 kb/s y una subida de entre 25 y 30 Kb/s mas o menos reales. Este cuello de botella casi seguro que nos afecta si queremos ofrecer servicios a Internet. Por el contrario si este **servidor Web** vamos a usarlo para una red local interconectada con las tarjetas de red típicas de 10/100 Mbit/s obtendremos una velocidad suficiente como para satisfacer muchas mas peticiones al servidor y he aquí entonces donde entra el factor **Hardware** y los recursos de la maquina serán los que manden ya que no contaremos con nada que obstruya la salida y entrada de datos a nuestro **servidor web**.

Una vez finalizado el proyecto veremos el articulo final donde serán incluidos en forma de índice todos los capítulos de este nuevo **servidor web** bajo **Linux Ubuntu Server** que puede haceros aprender y practicar para cuando tengáis mas trafico en vuestros sitios web y tengáis que montaros uno remoto en un empresa externa.

Si os apetece aprender a montar una maquina como esta con la que podréis aprender a **fabricar vuestro propio Hosting** preparad vuestros ordenadores porque próximamente comenzamos este nuevo proyecto que espero sea de vuestro agrado.

## Conclusión

La finalidad de este proyecto no es tan solo el **montar un servidor web** donde podremos alojar las webs que queramos si no que aprenderemos sobre el funcionamiento del sistema operativo **Linux Ubuntu Server** y de los programas que instalemos e incluso del funcionamiento en general desde

que un usuario quiere leer una de las webs que tengamos alojada desde su ordenador llegando a nuestro Router y pasando finalmente hacia nuestro servidor web que estará preparado para satisfacer dicha petición devolviendo la web deseada.

*Jugar, Aprender, Disfrutar, GNU/Linux, Informática, Mi Hijo ....*

# Vol 1 – Como instalar Linux Ubuntu Server 9.10

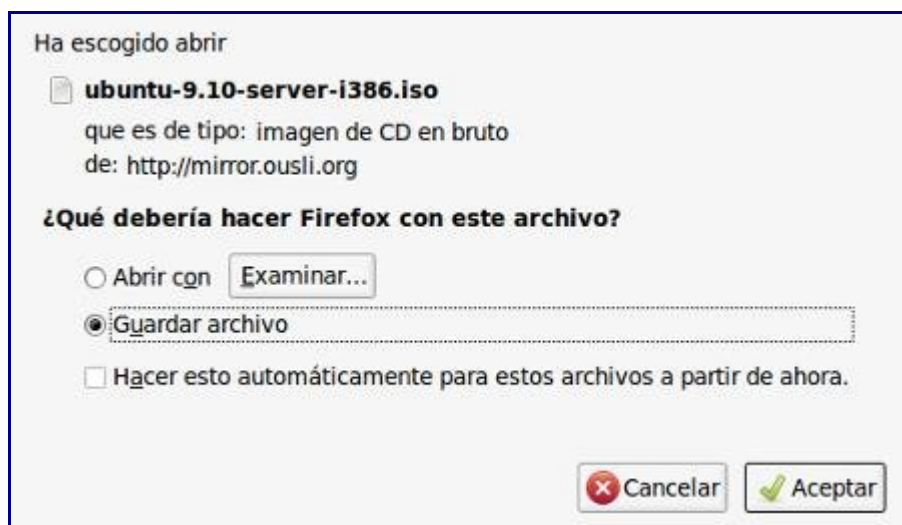


Con este tutorial comenzaremos el montaje de nuestro **servidor Web** en **Linux Ubuntu Server**. Una vez lo tengamos instalado tendremos una instalación mínima y casi limpia del sistema operativo **Linux Ubuntu Server 9.10 Karmic Koala** así que si vas a seguir este proyecto ya puedes comenzar, si no es así y tan solo te interesa la instalación de este **sistema operativo GNU/Linux** también puedes hacerlo ya que tan solo instalamos con este tutorial el sistema operativo limpio, sin entorno gráfico y prácticamente sin paquetes extras. Tan solo se instala el software **OpenSSH** para poder manejar el servidor remotamente ...

Para comenzar accederemos al apartado de descargas de la web oficial de [Ubuntu](http://www.ubuntu.com) en el apartado ( **Server** ) [desde aquí](#) ...



En el apartado ( *Download location* ) seleccionamos la zona mas cercana a donde vivimos para que la descarga sea mas rápida. En mi caso he elegido ( *Spain* ). Una vez seleccionada la zona pulsamos sobre la opción ( *Alternative download options* ) y seleccionamos si queremos la versión para 32 o 64 Bits. Una vez seleccionada la versión que queramos pulsaremos sobre el botón verde ( *Begin download* ) y nos descargamos la imagen ISO ...

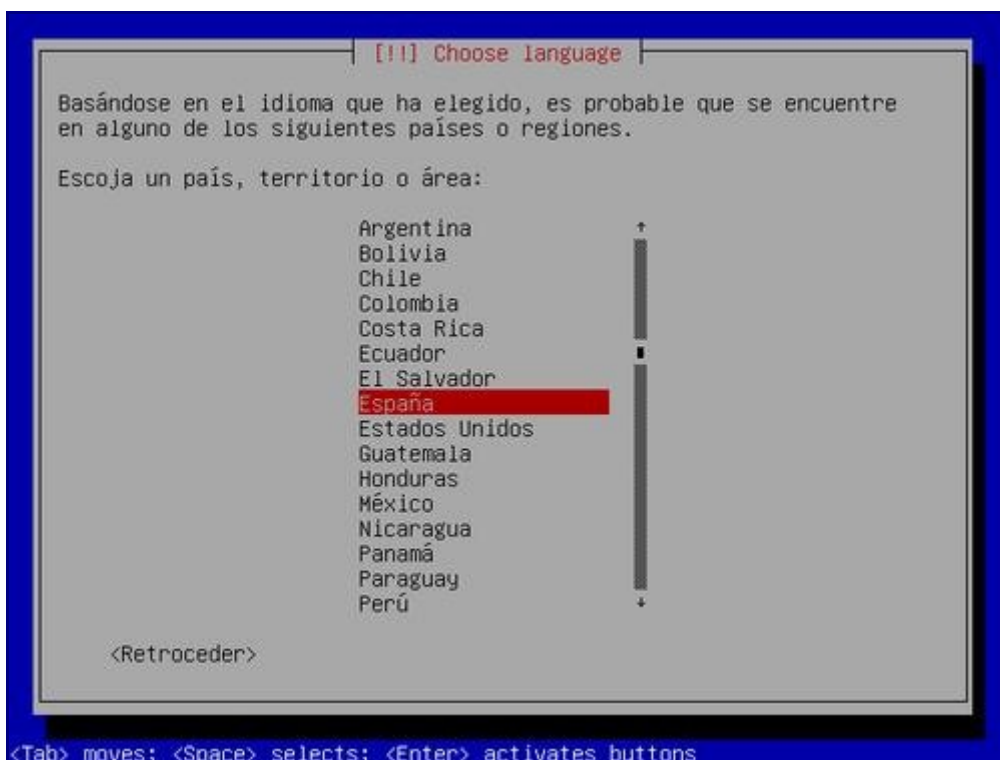


Una vez tengamos la imagen de CD ISO la grabamos en un CD de la forma habitual con el programa que mas os guste. Una vez tengamos el CD con **Linux Ubuntu Server 9.10** correctamente grabado en un CD arrancamos nuestro servidor y veremos lo siguiente ...

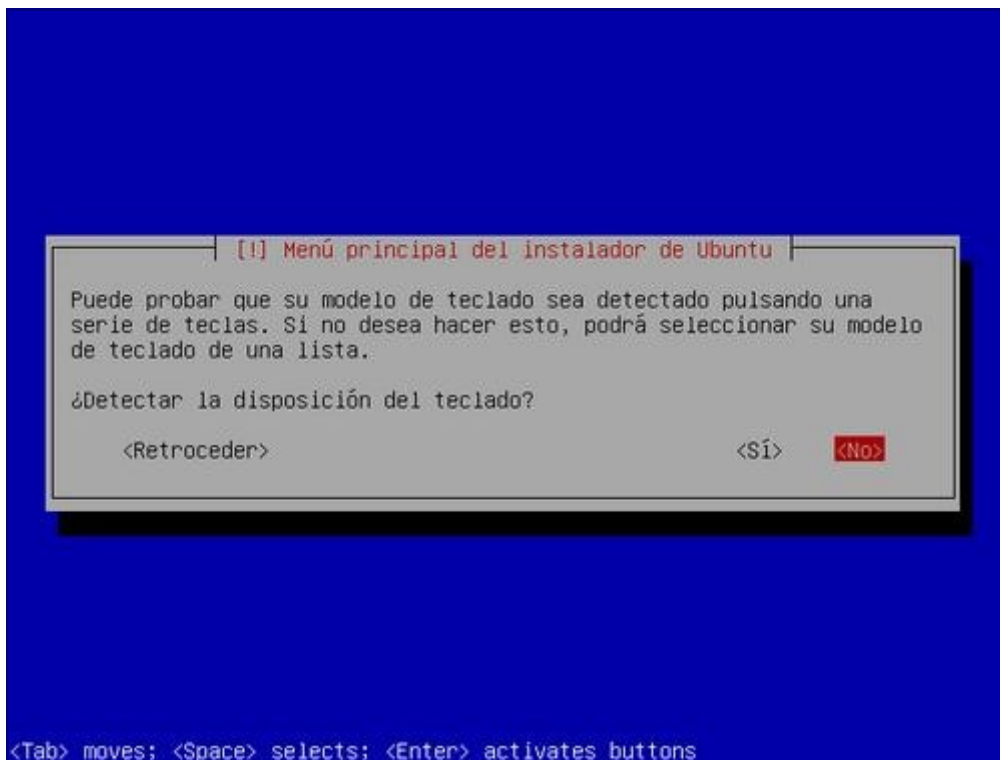
Seleccionamos nuestro idioma y pulsamos la tecla ( *Enter* ) ...



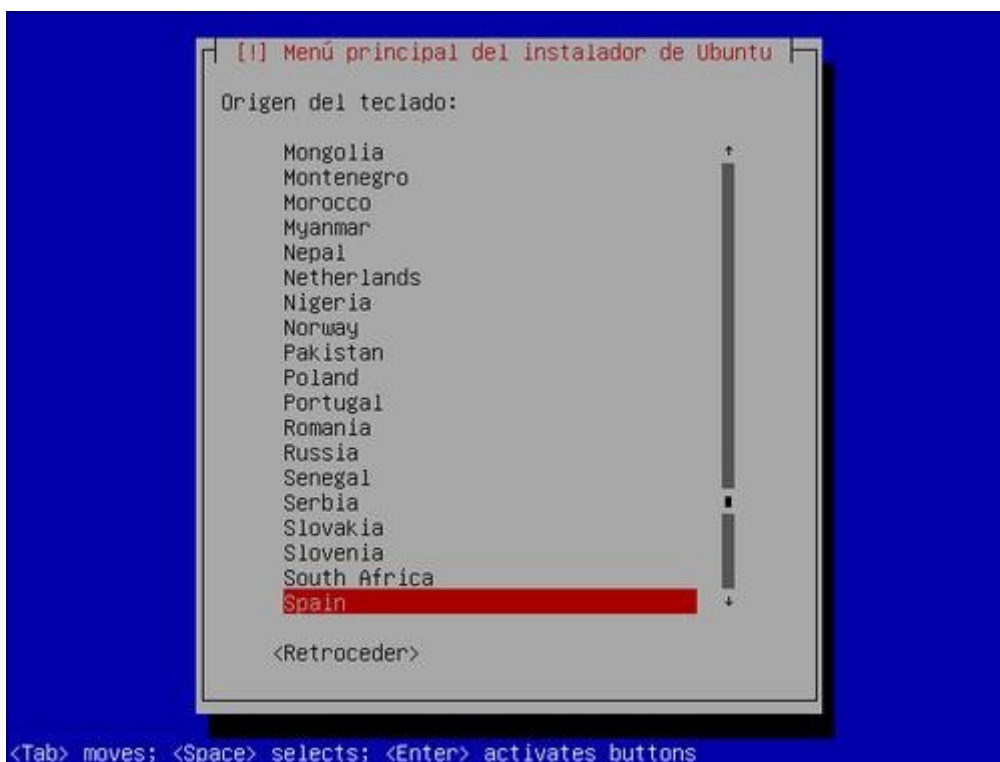
Seleccionamos la primera opción ( **Instalar Ubuntu Server** ) y pulsamos la tecla ( *Enter* ) ...



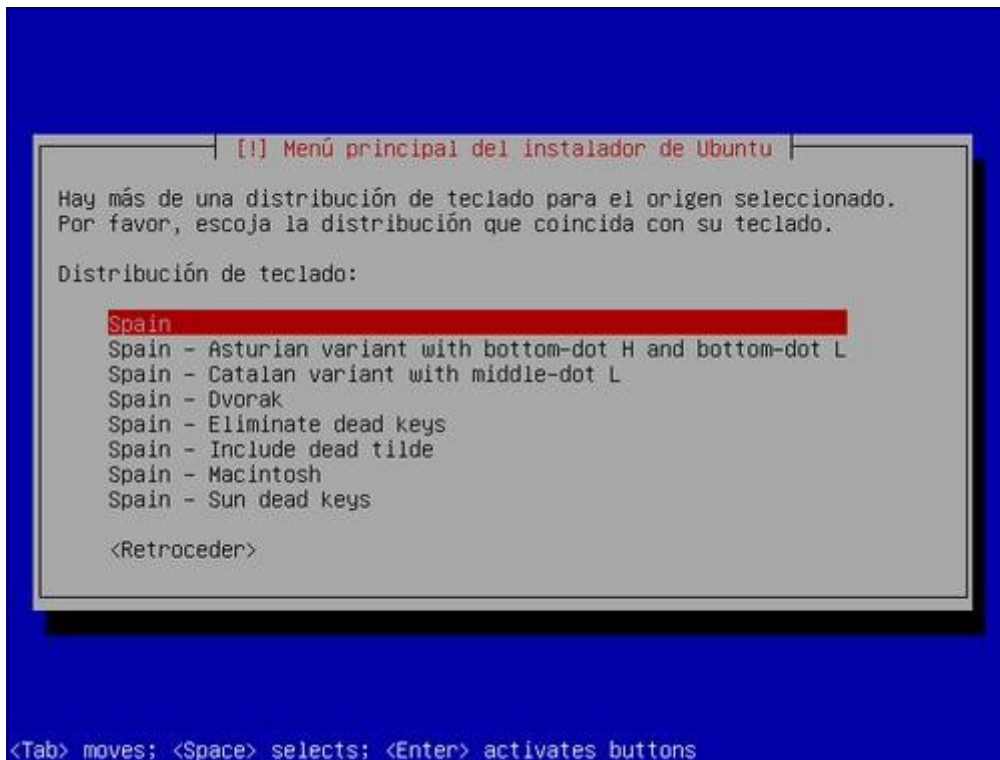
Seleccionamos nuestro idioma y pulsamos la tecla ( *Enter* ) ...



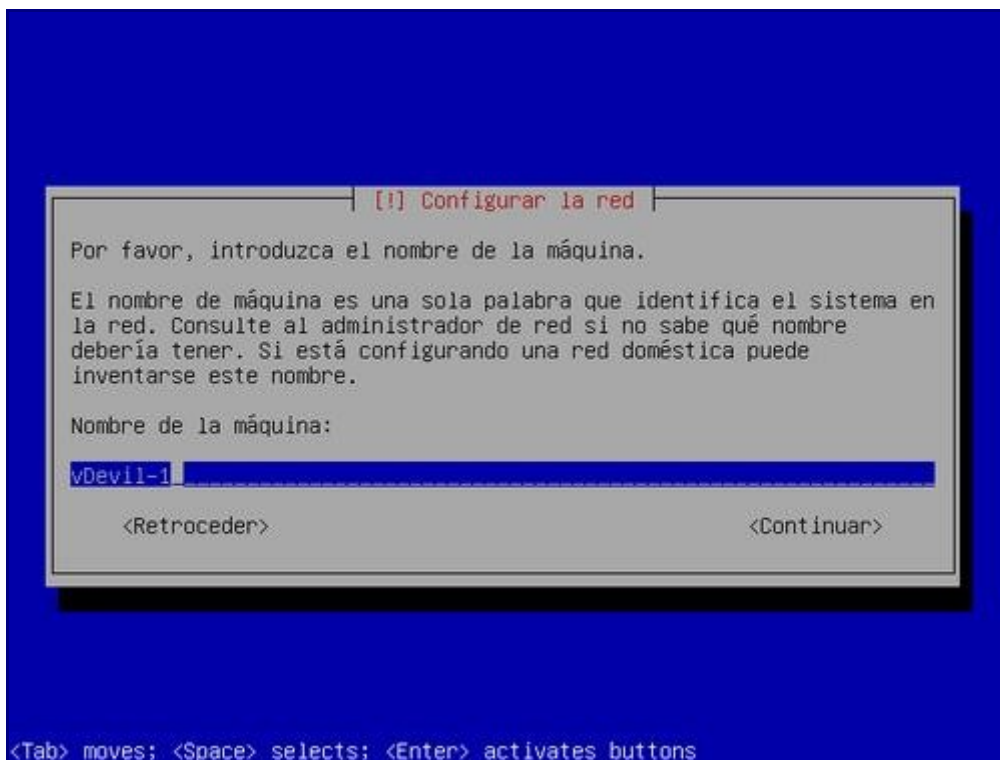
En esta pantalla podemos probar nuestro teclado, en principio no deberíamos tener problemas y mas si ya hemos usado algunas veces este teclado con otra distribución **Ubuntu** y nos ha funcionado bien. Si tienes dudas pulsa ( *Si* ) y haz la prueba y si crees que te funcionará bien como es mi caso pulsamos la opción ( *No* ) y seguidamente la tecla ( *Enter* ) ...



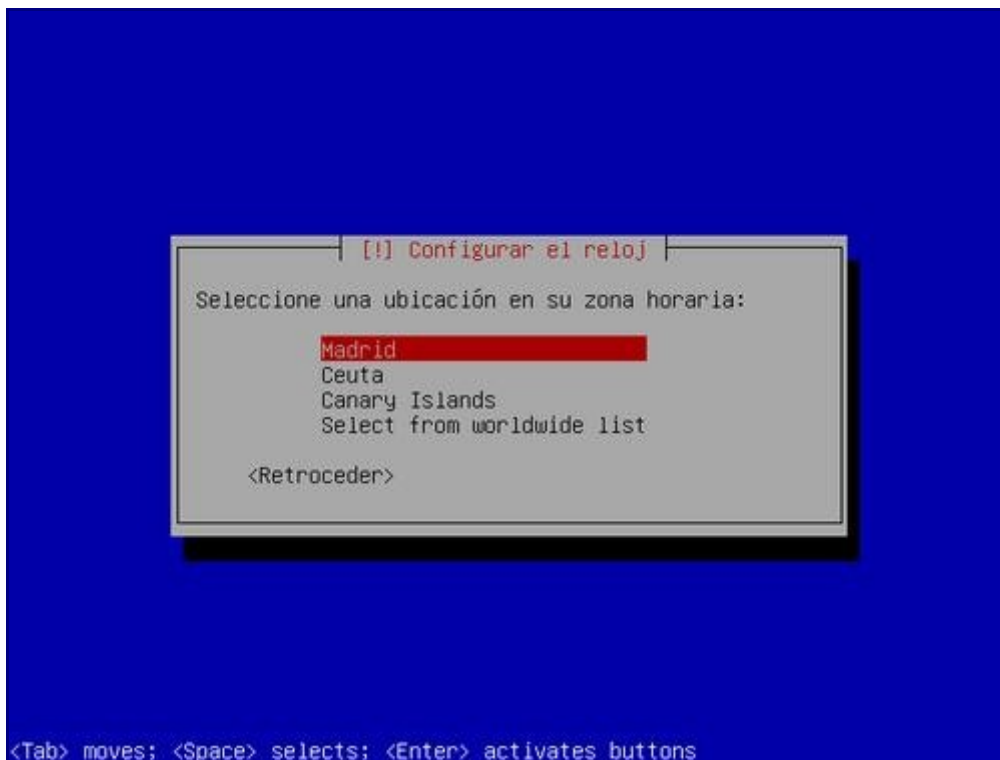
Seleccionamos nuestro idioma nuevamente y continuamos pulsando la tecla ( *Enter* ) ...



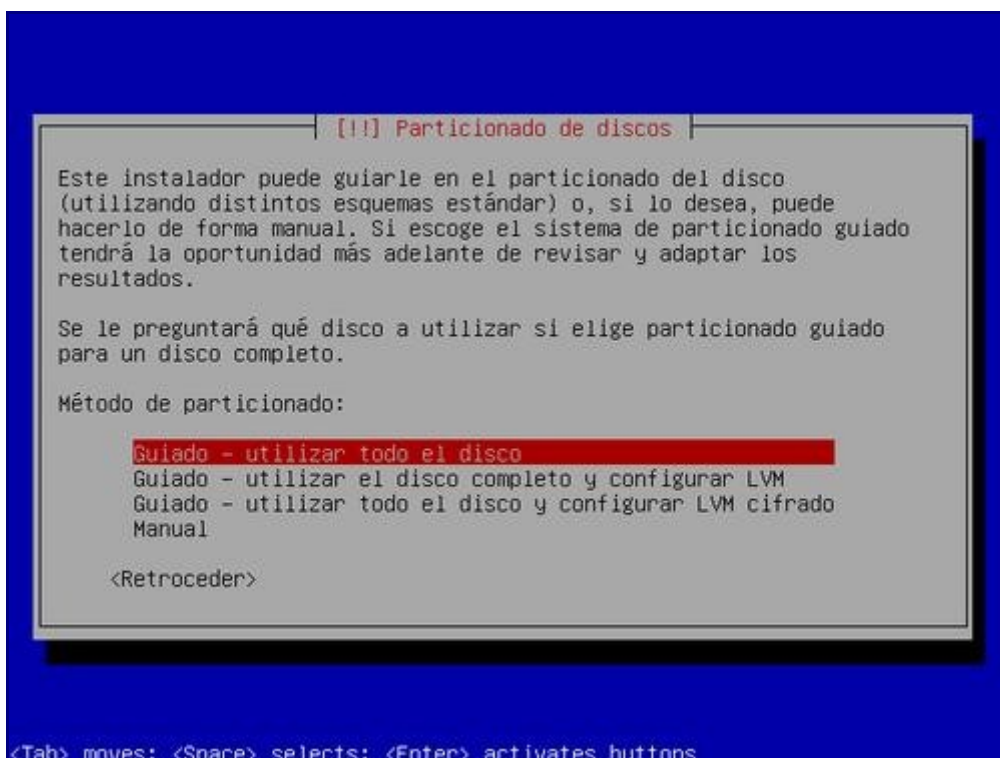
Seleccionamos la distribución de nuestro teclado y pulsamos la tecla ( *Enter* ) ...



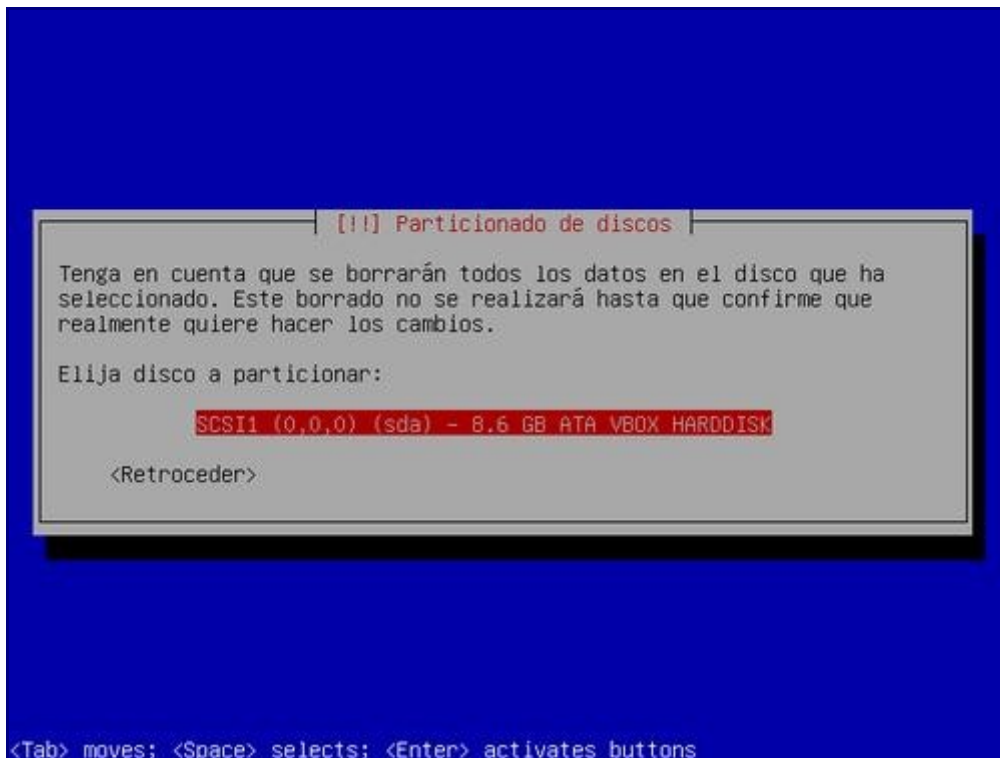
Introducimos el nombre que queramos ponerle a nuestra maquina, en mi caso ( **vDevil-1** ), le damos el nombre que mas nos guste y pulsamos la tecla ( *Enter* ) ...



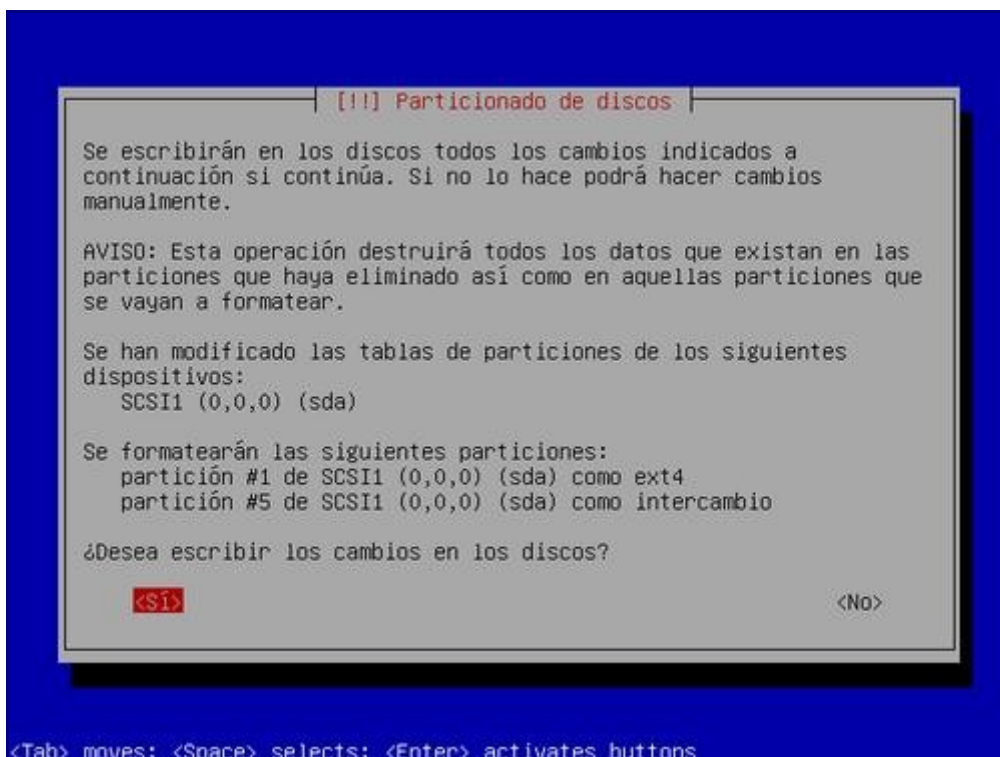
Seleccionamos nuestra ubicación o la mas cercana a nosotros y pulsamos la tecla ( *Enter* ) ...



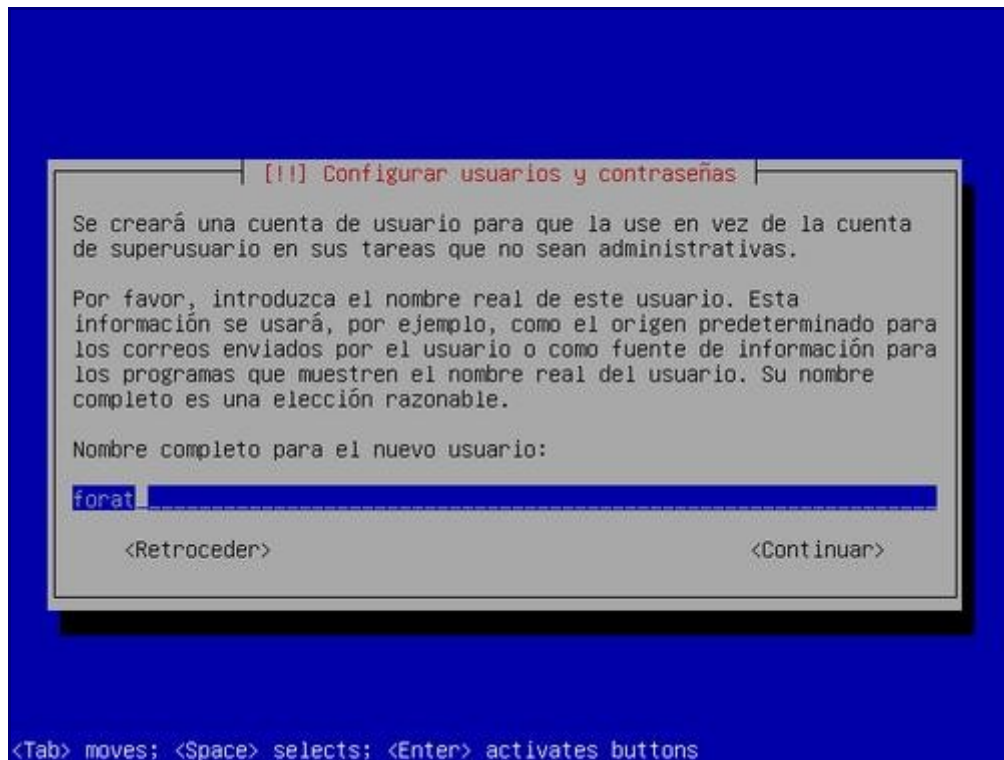
Este ordenador que estamos usando lo vamos a utilizar solo como servidor así que ocuparemos el espacio al completo del disco seleccionando la primera opción ( *Guiado – utilizar todo el disco* ) y seguidamente pulsamos la tecla ( *Enter* ) ...



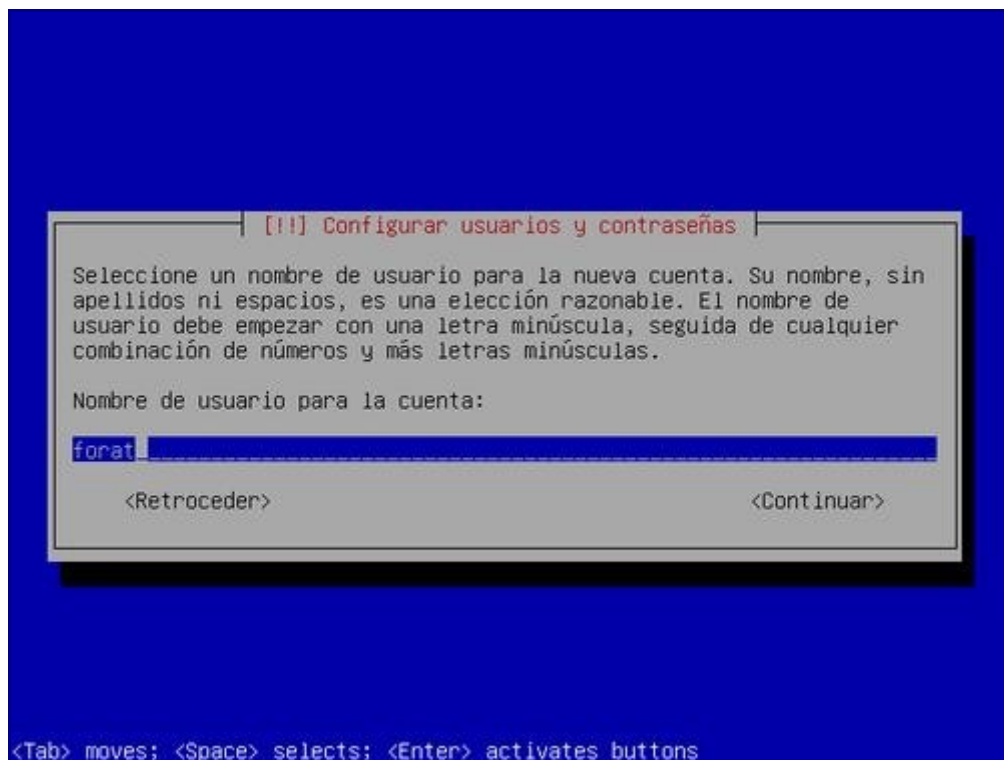
Si tenemos mas de un disco duro instalado aquí podemos verlo. Seleccionamos el disco donde vayamos a **instalar Linux Ubuntu Server** y pulsamos la tecla ( *Enter* ) ...



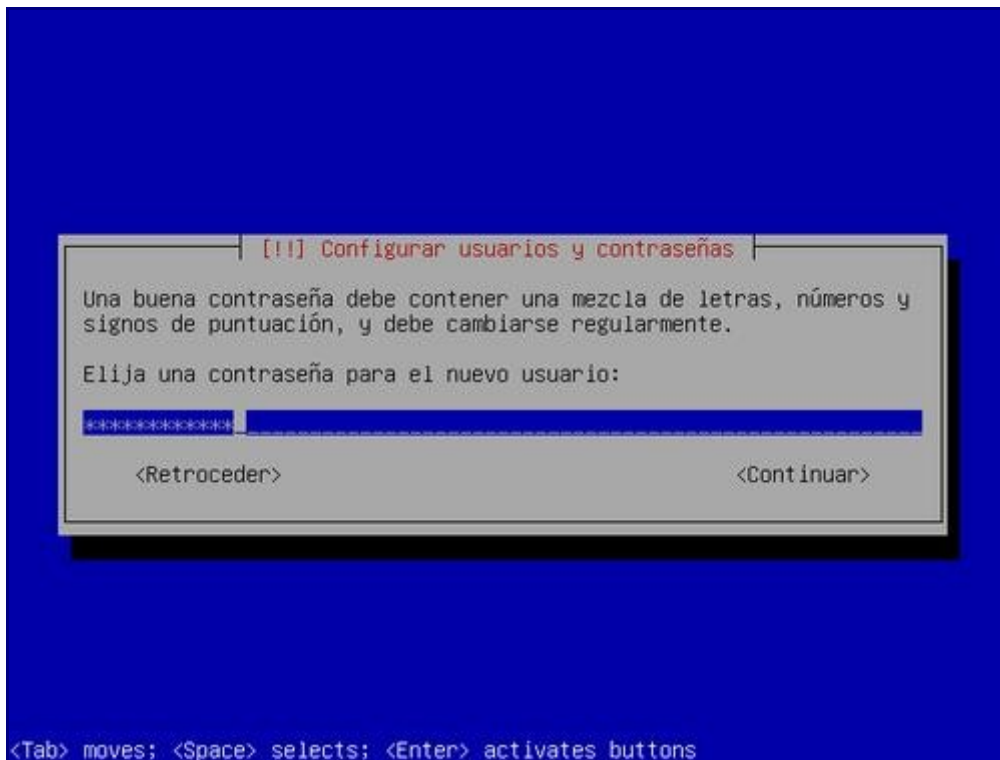
En esta pantalla veremos una lista sobre todo lo que estamos a punto de hacerle a nuestro disco duro. Si todo está correcto seleccionamos la opción ( *Si* ) y continuamos con la tecla ( *Enter* ) ...



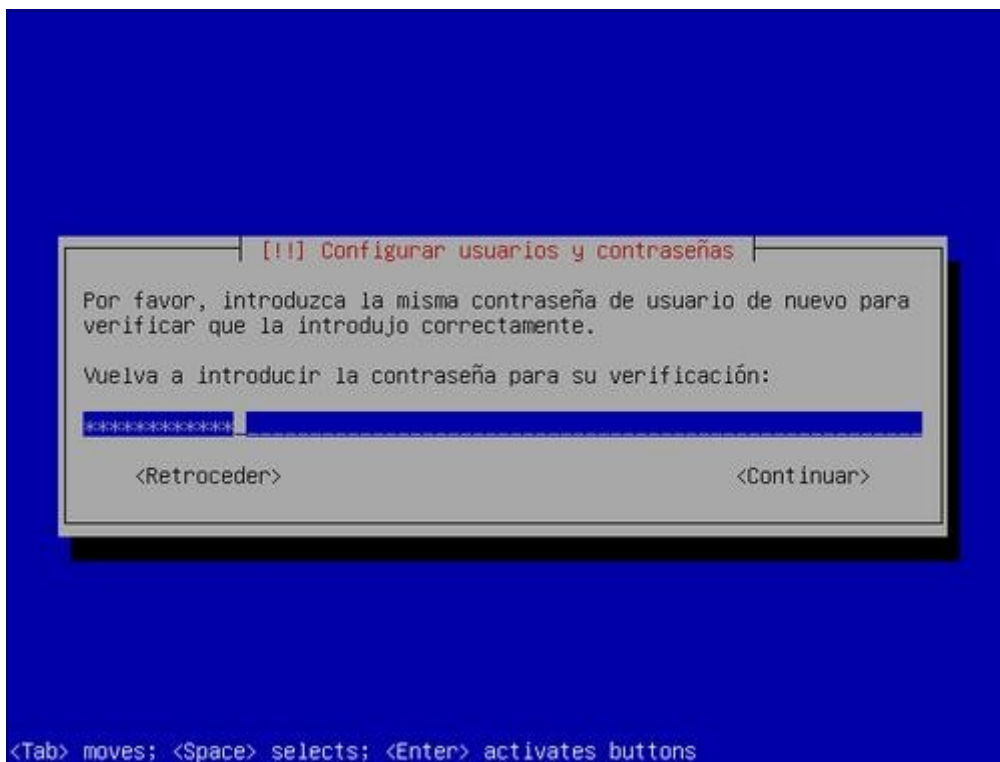
Introducimos el nombre del usuario que usará la maquina y pulsamos la tecla ( *Enter* ) ...



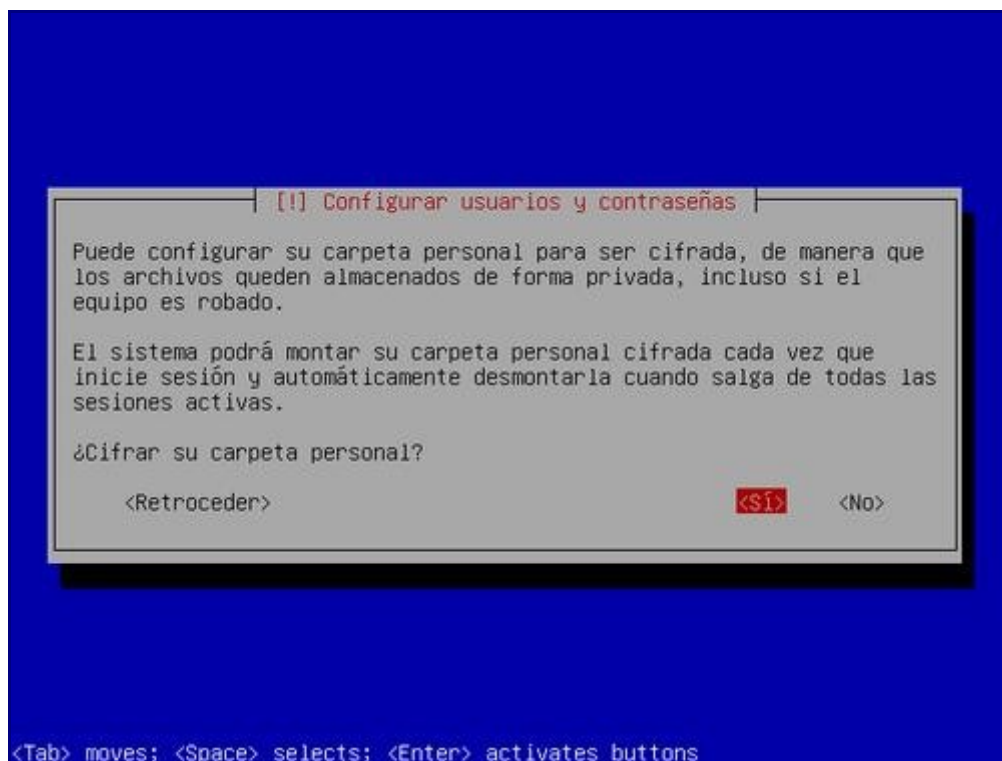
Seguidamente introducimos el nombre de usuario que usaremos para acceder al servidor y pulsamos la tecla ( *Enter* ) ...



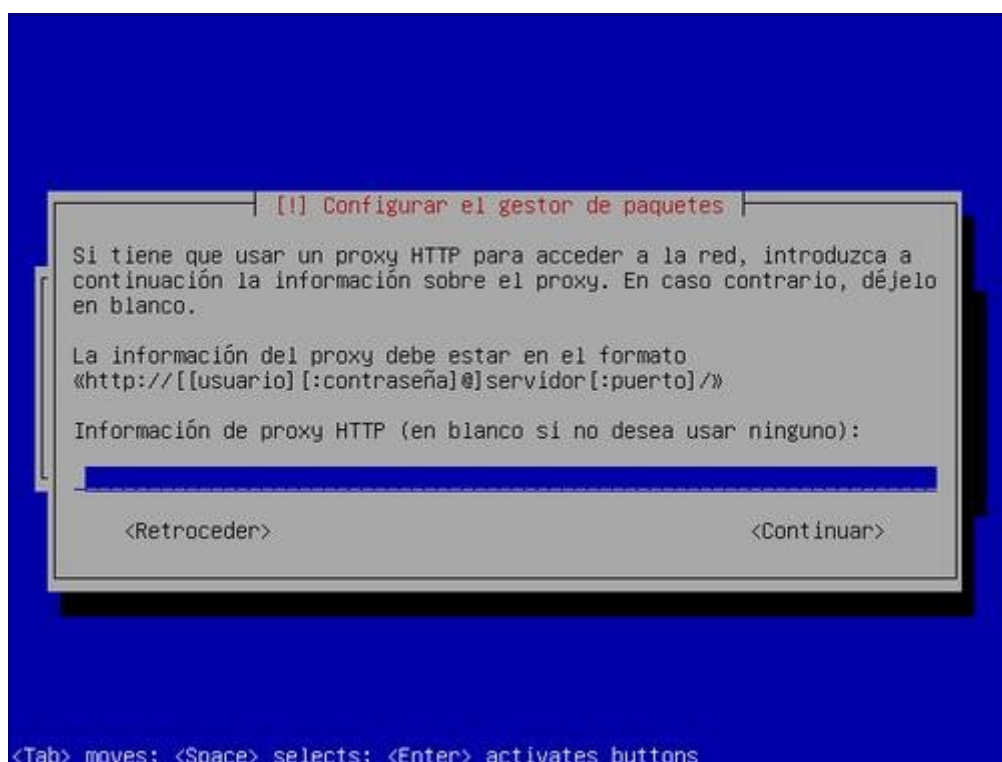
Introducimos la contraseña para nuestro usuario y pulsamos la tecla ( *Enter* ) ...



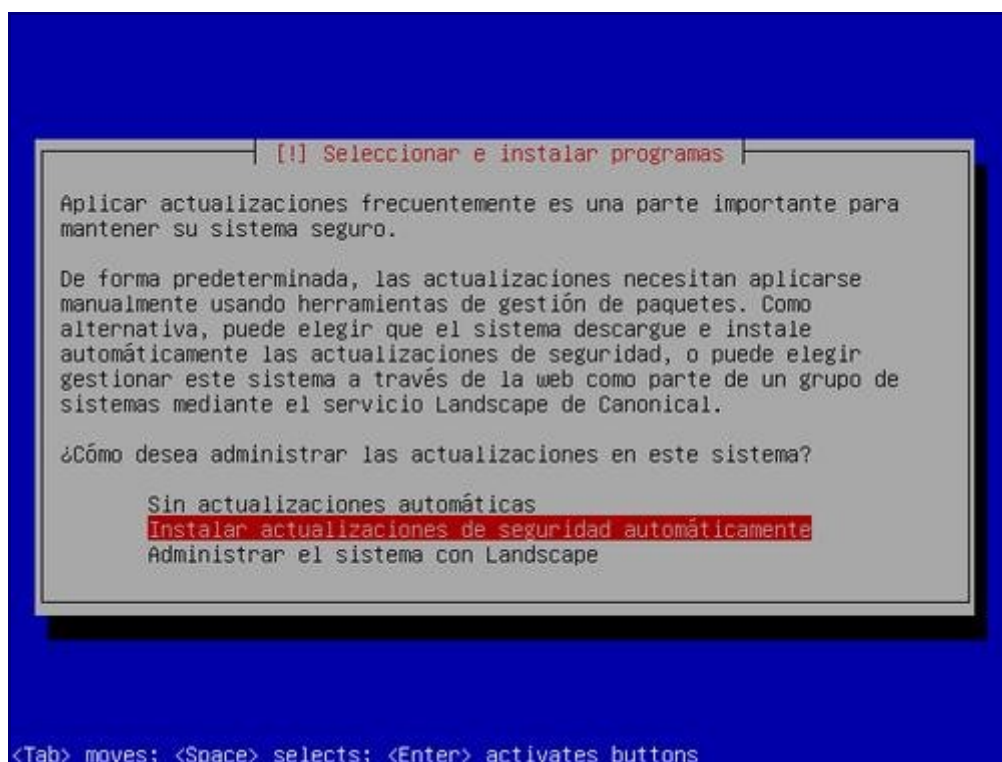
Introducimos de nuevo la misma contraseña y pulsamos la tecla ( *Enter* ) ...



En esta opción como en la instalación de [Linux Ubuntu Desktop 9.10](#) podemos **cifrar nuestra carpeta personal** para que otro usuario de la maquina no pueda verla e incluso si nos roban el servidor no puedan acceder a nuestra carpeta personal de usuario. En mi caso voy a seleccionar ( *No* ) porque hace poco tiempo que se ha implementado y no me fío de su estabilidad, aunque esto os lo dejo a vuestra elección continuaremos pulsando la tecla ( *Enter* ) ...

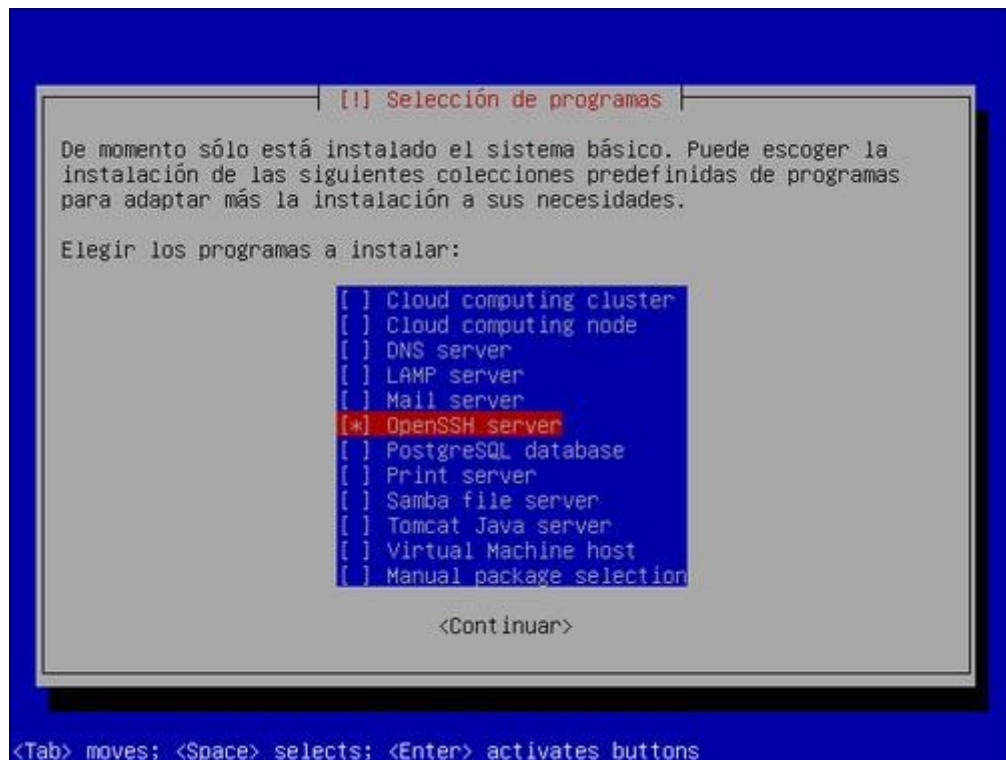


Si disponemos de un servidor proxy que nos da acceso a la red introducimos aquí sus datos. Si no es así y en la mayoría de los casos de **ADSL** normales no lo es, dejamos este espacio en blanco y continuamos pulsando la tecla ( *Enter* ) ...



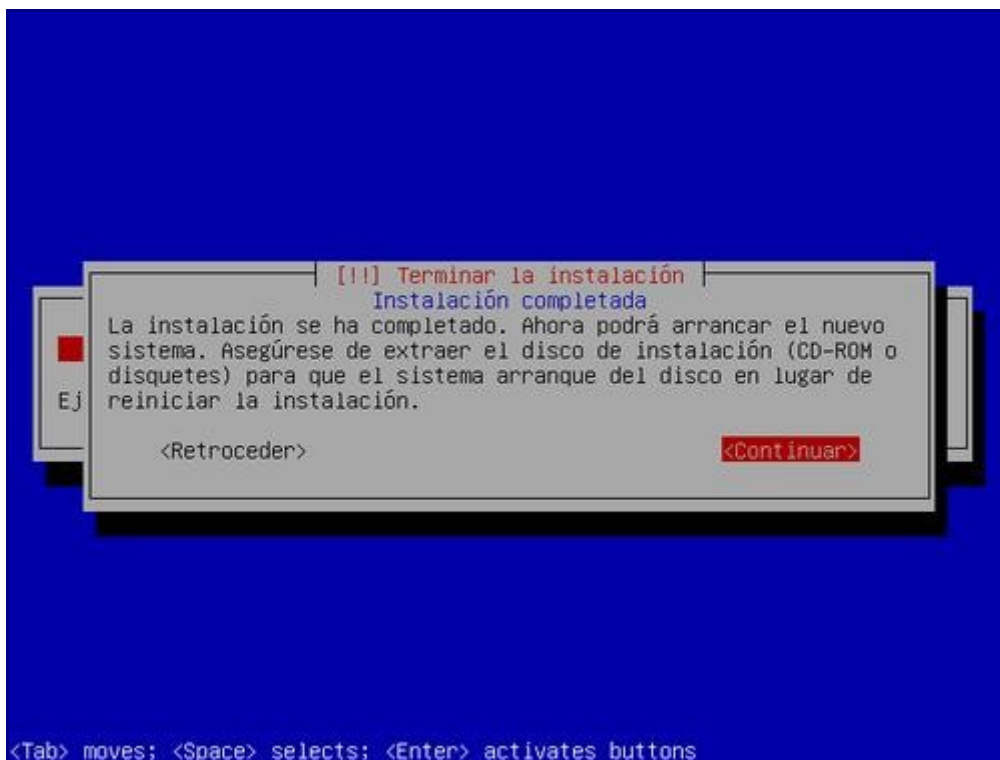
En este apartado vamos a seleccionar la segunda opción ( *Instalar actualizaciones de seguridad automáticamente* ) por si algún despistado una vez funcionando el servidor se olvida por completo de su mantenimiento y tan solo se dedica a usarlo.

Se que muchos de vosotros no estaréis de acuerdo con esta técnica de actualizaciones automáticas de seguridad así que si sois usuarios avanzados en el mundo **GNU/Linux** seguramente vais a preferir el modo ( *Sin actualizaciones automáticas* ) y actualizar vuestro sistema vosotros mismos revisando todos y cada uno de los paquetes que vamos a actualizar. Así que ya sabéis para novatos la segunda opción y para " pro's " la segunda, seguidamente continuamos con la tecla ( *Enter* ) ...



En esta sección podemos seleccionar que tipo de paquetería necesita nuestro **servidor web**. En este caso lo único que vamos a instalar será el servidor seguro ( **OpenSSH server** ) situándonos sobre el y pulsando la barra espaciadora. Como podréis apreciar desde aquí podemos instalar varios servicios como el **servidor DNS**, **LAMP**, **Mail** y demás. Ahora no vamos a seleccionar otro que no sea el **servidor seguro SSH** porque mas adelante en otros capítulos iremos instalando todo lo que necesitaremos en su respectivo orden. En estos momentos tan solo necesitamos **instalar Linux Ubuntu Server** de forma mínima y a partir de aquí iremos instalando paso a paso mas programas y terminaremos con un **servidor web completo**.

Además de ir aprendiendo un poco mas en cada capitulo sobre que instalamos para que funcione una cosa u otra aprenderemos a solucionar problemas una vez los tengamos. Si instalamos varios servidores en este menú no aprenderemos ni que es lo que se ha instalado ni para que sirve y como la finalidad de estos tutoriales no es mas que la de aprender yo personalmente opto por aceptar tan solo la opción ( **OpenSSH Server** ) y pulsar la tecla ( *Enter* ) para no perder la costumbre ...



Llegados a este paso ya tenemos **Linux Ubuntu Server 9.10** correctamente **instalado** con lo mínimo para empezar a trabajar. Pulsamos la tecla ( *Enter* ) y continuamos ...

```
fsck from util-linux-ng 2.16
/dev/sda1: clean, 28913/498736 files, 209700/1994060 blocks
[ 13.424283] piix4_smbus 0000:00:07.0: SMBus base address uninitialized - upgr
ade BIOS or use force_addr=0xaddr
* Setting preliminary keymap...
* Starting init crypto disks...
* Starting early crypto disks... * cryptswap1 (starting)

One or more of the mounts listed in /etc/fstab cannot yet be mounted:
(ESC for recovery shell)
swap: waiting for /dev/mapper/cryptswap1
* Starting remaining crypto disks... [ OK ]

* Starting AppArmor profiles
* Setting up console font and keymap...
* Starting OpenBSD Secure Shell server sshd
Ubuntu 9.10 vDevil-1 tty1
vDevil-1 login: _
```

Como podemos ver una vez reiniciada la maquina nos pide que le introduzcamos el nombre de usuario. Usaremos el que creamos durante la instalación y pulsamos la tecla ( *Enter* ) ...

```

fsck from util-linux-ng 2.16
/dev/sda1: clean, 28913/498736 files, 209700/1994060 blocks
[ 13.424283] piix4_smbus 0000:00:07.0: SMBus base address uninitialized - upgr
ade BIOS or use force_addr=0xaddr
* Setting preliminary keymap...
* Starting init crypto disks...
* Starting early crypto disks...      * cryptswap1 (starting)

One or more of the mounts listed in /etc/fstab cannot yet be mounted:
(ESC for recovery shell)
swap: waiting for /dev/mapper/cryptswap1
* Starting remaining crypto disks... [ OK ]

* Starting AppArmor profiles
* Setting up console font and keymap...
* Starting OpenBSD Secure Shell server sshd
Ubuntu 9.10 vDevil-1 tty1

vDevil-1 login: forat
Password: _

```

Seguidamente nuestra contraseña y pulsamos la tecla ( *Enter* ) ...

```

To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/

System information as of Tue Dec 22 02:04:14 CET 2009

System load: 0.04          Memory usage: 2%    Processes:      65
Usage of /:  9.1% of 7.49GB Swap usage:   0%    Users logged in: 0

Graph this data and manage this system at https://landscape.canonical.com/

42 packages can be updated.
14 updates are security updates.

The programs included with the Ubuntu system are free software:
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

forat@vDevil-1:~$ _

```

Una vez identificados nuestro servidor nos presenta unos porcentajes sobre su estado y se actualiza automáticamente. Ahora tenemos el sistema preparado para empezar a instalarle software y configurarlo correctamente. Esto será en los siguientes capítulos donde aprenderemos a **montar un servidor web completo** con todos los servicios que podamos necesitar.

Antes de terminar vamos a actualizar nuestros repositorios con ...

~ código ~

```
sudo apt-get update
```

Y actualizaremos el sistema ...

~ código ~

```
sudo apt-get upgrade
```

Si quieres trastear ahí tienes la terminal pura y dura, puedes entretenerte con el manual que explica [como está organizado GNU/Linux por dentro.](#)

## Vol 2 – Configuración de Red y manejo remoto vía OpenSSH con SSH y SFTP



Si hay una cosa que hay que tener clara antes de montar un servidor web en casa es como tenemos configurada la red. Con **configuración de la red** no me refiero solo a la **IP** de la tarjeta de red de nuestro **servidor** sino a como tenemos configurado el **Router** para que pueda aceptar peticiones desde el exterior. Imaginemos que un usuario desde Internet se dispone a visitar un sitio web que tenemos en nuestro **servidor local**. Si lo hace mediante el navegador el mismo enviará una petición al *puerto 80 del Router*, el *Router* debe de saber hacia que maquina redirigir el **puerto 80** para satisfacer dicha petición como podemos ver en la siguiente imagen ...



Para poder redireccionar un puerto desde nuestro Router hacia nuestro servidor vamos a necesitar que nuestro servidor disponga de una dirección IP fija. Si no tenemos una **IP fija** no podremos decirle a nuestro Router hacia donde apuntaran las peticiones que vengan desde el exterior.

Este [servidor web con Linux Ubuntu Server](#) que estamos montando necesita una **IP fija** interna que esté dentro del rango de IP de nuestra red. Quiero decir con esto que si nuestro **Router** tiene una **IP interna** 192.168.0.1, nuestro servidor debería tener una **IP** dentro del mismo rango por ejemplo la 192.168.0.10

Si sabemos cual es la **IP** de nuestro **Router** mejor y si no tampoco va a sernos un problema averiguar cual es ya que [Linux Ubuntu Server](#) viene con la asignación de *IP dinámica* por *DHCP* y nuestro **Router** nos asigna una diferente cada vez que arrancamos el sistema.

Arrancaremos nuestro **servidor web**, nos identificamos con nuestro nombre de usuario, password y nos dispondremos a ejecutar el comando **route** el cual nos dará la configuración actual de nuestro ruteado. Tecleamos lo siguiente ...

- codigo -

```
route -n
```

Y veremos algo parecido a esto ...

```
forat@vDevil-1:~$ route -n
Tabla de rutas IP del núcleo
Destino      Pasarela      Genmask      Indic Métric Ref      Uso
10.0.2.0     0.0.0.0       255.255.255.0 U      0      0      0
0.0.0.0      10.0.2.2      0.0.0.0      UG     100    0      0
forat@vDevil-1:~$
```

En mi caso y como podemos ver en la imagen la dirección de mi **Router** es la 10.0.2.2 con lo que tendré que asignarle una **IP** dentro del rango 10.0.2.X. Si vuestra **IP** no es 10.X.X.X no es un

problema ya que también pueden estar dentro de otros rangos privados de **IP** como pueden ser las *192.X.X.X* o las *172.X.X.X*. Estos tres rangos *10*, *192* y *172* son los reservados para redes locales en todos los casos.

Una vez sepamos la **IP** de nuestro **Router**, en mi caso la *10.0.2.2*, configuraremos la tarjeta de red de nuestro servidor para que sea una **IP fija** la que tengamos siempre. Para ello editaremos el archivo ( **interfaces** ) con el siguiente comando ...

- código -

```
sudo nano /etc/network/interfaces
```

Y veremos lo siguiente ...

```
# 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

# The primary network interface
auto eth0
iface eth0 inet dhcp
```

[ Leer 10 líneas ]

Ver ayuda Guardar Leer Fich Pág Ant Cortar Tex Pos a  
Salir Justificar Dónde Está Pág Sig PegarTxt Ortogr

Esta configuración por defecto en **Linux Ubuntu Server** hace que nuestra tarjeta de red ( *eth0* ) recoja una **IP dinámica** vía **DHCP** que nuestro **Router** le asigna. Para asignarle una **IP fija** vamos a borrar la línea referente a nuestra tarjeta de red ( *eth0* ) y agregamos la siguiente configuración quedándonos así ...

```
# 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

# The primary network interface
auto eth0
iface eth0 inet static
-
    address 10.0.2.10
    netmask 255.255.255.0
    gateway 10.0.2.2
```

Ver ayuda Guardar Leer Fich Pág Ant Cortar Tex Pos a  
Salir Justificar Dónde Está Pág Sig PegarTxt Ortogr

Como podemos ver en el apartado ( *address* ) he introducido la **IP fija** que tendrá a partir de ahora mi **servidor web** y en el apartado ( *gateway* ) introducimos la **IP de nuestro Router**. Una vez hechos los cambios guardaremos el archivo y saldremos del editor con la tecla ( *F2* ), seguidamente

con la tecla ( *S* ) y por ultimo la tecla ( *Enter* ).

Ahora para que los cambios surjan efecto debemos reiniciar la red con el siguiente comando ...

~ codigo ~

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

Ahora si hacemos la prueba con el comando *ifconfig* veremos la nueva configuración de nuestra tarjeta de red ...

~ codigo ~

```
ifconfig
```

En mi caso ...

```
forat@vDevil-1:~$ ifconfig
eth0      Link encap:Ethernet  direcciónHW 08:00:27:31:65:96
          Direc. inet:10.0.2.10  Difus.:10.0.2.255  Másc:255.255.255.0
          Dirección inet6: fe80::a00:27ff:fe31:6596/64 Alcance:Enlace
          ACTIVO DIFUSIÓN FUNCIONANDO MULTICAST  MTU:1500  Métrica:1
          Paquetes RX:33 errores:0 perdidos:0 overruns:0 frame:0
          Paquetes TX:44 errores:0 perdidos:0 overruns:0 carrier:0
          colisiones:0 long.colaTX:1000
          Bytes RX:4114 (4.1 KB)  TX bytes:3730 (3.7 KB)
          Interrupción:11 Dirección base: 0xd020

lo        Link encap:Bucle local
          Direc. inet:127.0.0.1  Másc:255.0.0.0
          Dirección inet6: ::1/128 Alcance:Anfitrión
          ACTIVO LOOPBACK FUNCIONANDO  MTU:16436  Métrica:1
          Paquetes RX:0 errores:0 perdidos:0 overruns:0 frame:0
          Paquetes TX:0 errores:0 perdidos:0 overruns:0 carrier:0
          colisiones:0 long.colaTX:0
          Bytes RX:0 (0.0 B)  TX bytes:0 (0.0 B)

forat@vDevil-1:~$ _
```

En estos momentos sabemos cual es la **IP de nuestro Router** y cual será la **IP fija de nuestro servidor web**. Ahora ya no necesitamos estrictamente tener monitor si disponemos de otro ordenador conectado al **Router** ya que podremos acceder por **SSH** desde otro ordenador por su **Terminal**.

## Conexión remota vía SSH

Para conectarnos desde otro ordenador conectado a nuestra red interna deberemos abrir una **Terminal** desde el ordenador con el que queramos conectarnos y ejecutar lo siguiente ...

~ codigo ~

```
ssh usuario@ipdelservidor
```

En mi caso seria ...

```
ssh forat@10.0.2.10
```

El sistema nos preguntará el password y listo. Ya estamos dentro de nuestro servidor como si tuviéramos el monitor conectado pero en una ventana dentro de nuestro sistema operativo [GNU/Linux](#) o [Mac](#). Esto también puede hacerse en otros sistemas como **Windows** gracias al cliente de **Terminal** llamado [Putty](#).

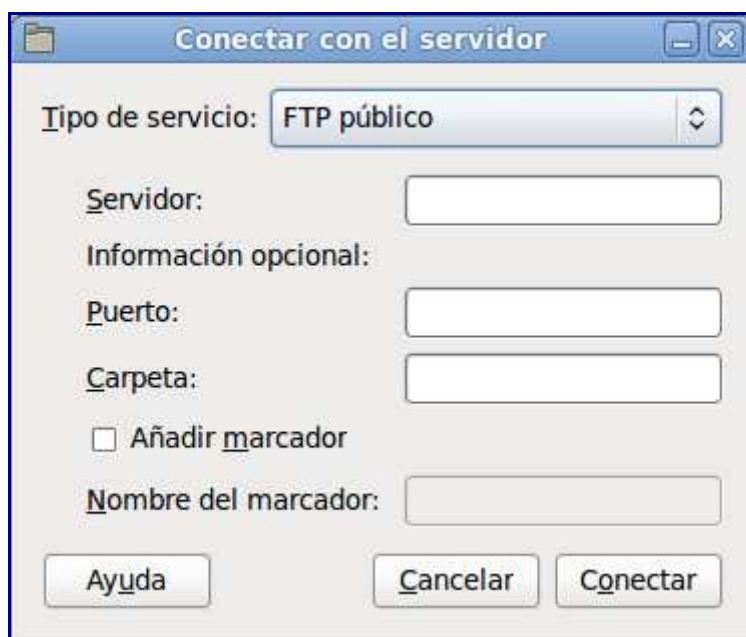
Durante la instalación en el paso donde nos preguntaba que paquetes queríamos instalar instalamos el servidor [OpenSSH](#) precisamente para poder **conectarnos a nuestro servidor remotamente** sin que necesitemos monitor. Esta muy bien acostumbrarse a no tener monitor en el servidor así cuando tengamos tanto trafico que no podamos navegar y nos decidamos a **alquilar un servidor** en alguna

empresa estaremos totalmente acostumbrados a trabajar remotamente desde la **Terminal** y no tendremos que pagar mas por no saber usar la **Terminal** y necesitar panel de control.

## Transferencia remota de archivos vía sFTP

**OpenSSH** nos ofrece la posibilidad de conectarnos a nuestro servidor remotamente desde la **Terminal** pero esto no es lo único que puede hacer ya que también puede hacer que nos conectemos en Red gracias a **sFTP**. Este es el **FTP** seguro de **SSH** que nos ofrece la posibilidad de conectarnos por una vía segura como es la de **SSH** desde un explorador de archivos como puede ser el que viene por defecto en **Gnome** llamado **Nautilus**.

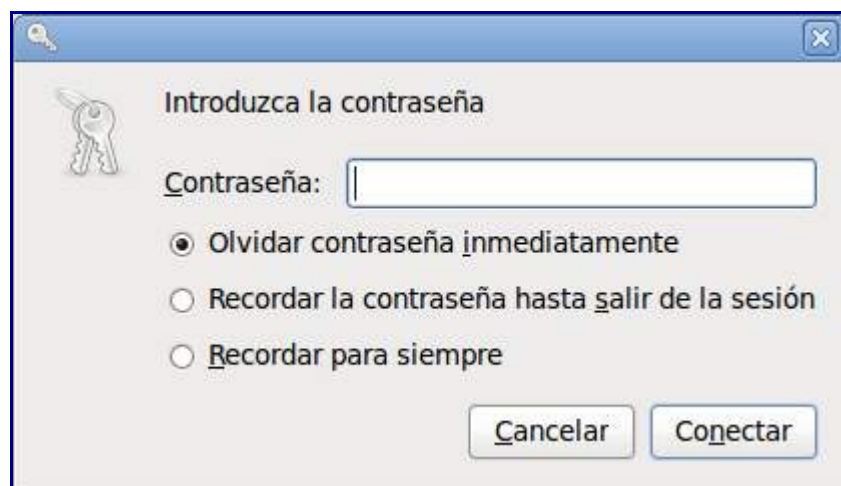
Desde un ordenador cliente con **Linux Ubuntu** conectado al **Router** tan solo tenemos que ir a la pestaña ( *Lugares* ), después a ( *Conectar con el servidor* ) y veremos lo siguiente ...



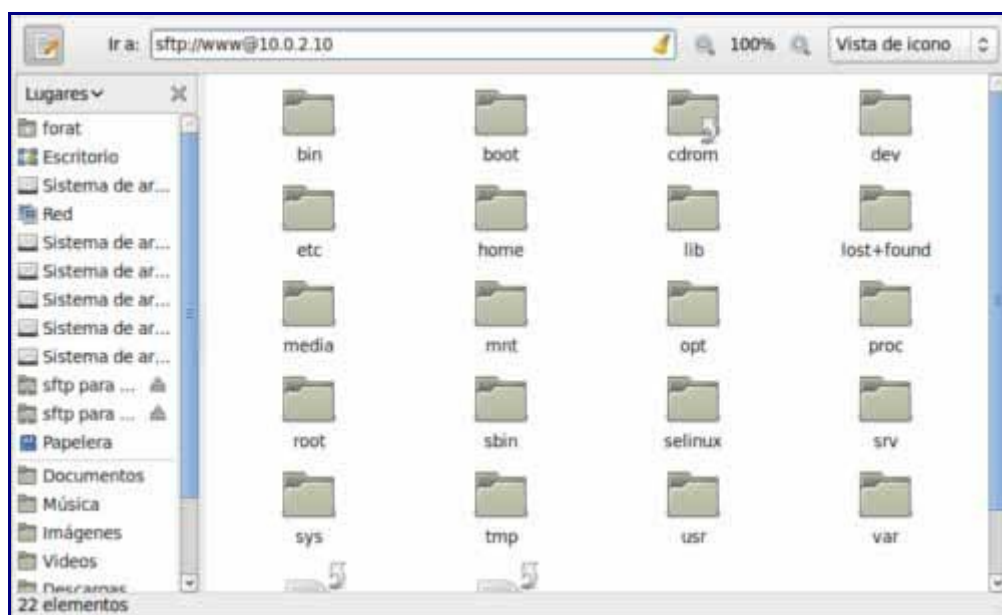
Con este **Frontend** para **Linux Ubuntu Desktop** podemos conectarnos a diferentes servicios remotos. Para conectarnos vía ( **sFTP** ) seleccionamos como tipo de servicio ( **SSH** ), seguidamente en el apartado servidor introducimos la dirección **IP fija** de nuestro **servidor web** y en el nombre de usuario el usuario que usamos en el servidor. Pulsamos sobre la casilla ( *Añadir marcador* ) y le damos un nombre al marcador ...



Pulsamos sobre el botón ( *Conectar* ) y nos pedirá nuestra contraseña de usuario ...



Se la introducimos y accederemos a todo el disco duro ...



Desde aquí y muy cómodamente podremos subir y bajar ficheros a nuestro servidor web como si de un disco duro mas se tratase. Esta es la forma de conectarse desde un sistema operativo [Linux Ubuntu](#) Desktop aunque también podreis conectaros desde **Windows** y **Mac** con cualquier **cliente FTP** compatible con **sFTP** o **SSH** segun el cliente.

En este **servidor Web** no vamos a [instalar un Servidor FTP](#) mas que nada porque podemos usar **sFTP** que es mas seguro que el **FTP** y viene de serie con **OpenSSH**. El lado oscuro de no instalar un **Servidor FTP** es que si queremos que otros usuarios usen cuentas **FTP** veran directorios y archivos de dentro del disco duro. Esto puede preocuparnos o no dependiendo quien sea quien va a usar el servidor y de si activamos el cifrado de disco durante la **instalación de Linux Ubuntu Server**. Al [activar el cifrado](#) protejemos todos y cada uno de los datos que esten dentro de la */home* de cada usuario así que aunque se conecte otro usuario y sea capaz de llegar al directorio */home* nunca verá su contenido al igual que todo lo que vera por todo el disco duro, al intentar ver algo que no sea del usuario que acaba de conectarse no podrá verlo gracias a la maravillosa jerarquia de usuarios y permisos que ofrecen los sistemas operativos **GNU/Linux** y al cifrado de disco que también aporta su parte importante en la seguridad del servidor.

De momento voy a dejar la **configuración del Router** para mas adelante ya que vamos instalar en nuestro servidor web diferentes servicios que podremos incluir a nuestro gusto y abriremos solo los que vayamos a utilizar realmente.

## Apuntes ...

Ya tenemos nuestro servidor con [Linux Ubuntu Server](#) instalado, **OpenSSH** funcionando, **Ip fija de la tarjeta de red** configurada y podemos **conectarnos remotamente desde una Terminal via ssh**, además de poder **navegar, subir y bajar archivos** por nuestro disco duro **remotamente** desde otro ordenador con un cliente compatible **sFTP** o **SSH**.

## Vol 3 – Como instalar LAMP + PhpMyAdmin



**Linux Ubuntu Server** viene con una utilidad para instalar grupos de paquetes llamada ( **tasksel** ) esta hace el mismo efecto que el seleccionar un grupo de paquetes durante la instalación del sistema operativo. **Tasksel** puede mostrarnos una lista de grupos de paquetes disponibles para instalar de serie. Entre ellos se encuentra el paquete **LAMP** que no es ni mas ni menos que un conjunto de paquetes que equiparan a nuestro servidor del **servidor web Apache2**, **servidor de bases de datos MySQL 5.1** y **sopORTE para PHP5,Perl y Python**. Vamos a instalarlo, *veamos como ...*

Como os decía **Linux Ubuntu Server** viene equipada con la utilidad para instalar grupos de paquetes llamada **Tasksel**. Para saber que grupos de paquetes podemos instalar tan solo tendremos que arrancar nuestro servidor, identificarnos y ejecutar lo siguiente ...

~ codigo ~

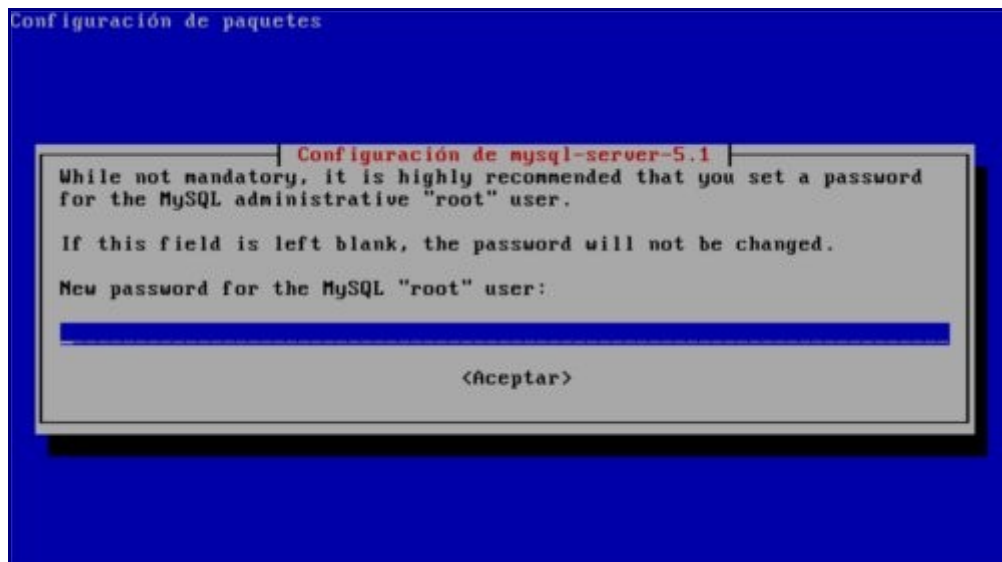
```
tasksel --list-tasks
```

En la lista veremos la letra ( *u* ) como grupo disponible y la letra ( *i* ) como grupo instalado. Si nos fijamos existe un grupo llamado ( **lamp-server** ) y será este el que utilicemos para la instalación del servidor web de la siguiente forma ...

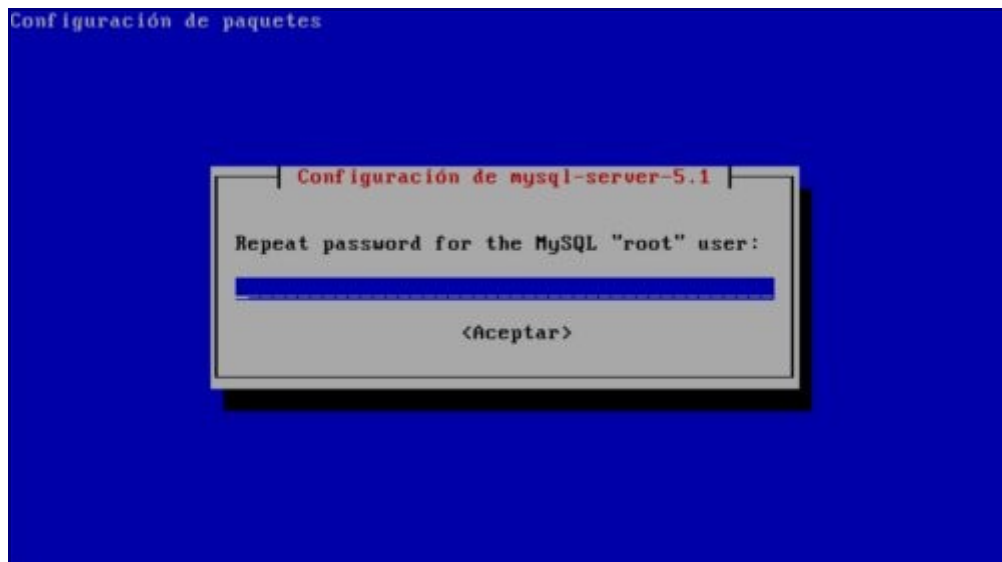
~ codigo ~

```
sudo tasksel install lamp-server
```

El sistema comenzará a descargar los archivos necesarios desde **Internet** hasta llegar al siguiente paso ...



Aquí vamos a introducir la contraseña del súper usuario ( **root** ) para acceder al **servidor de bases de datos MySQL**. Recordemos apuntar bien esta contraseña ya que la vamos a necesitar cada vez que queramos una base de datos nueva para cualquiera de los Blogs que alojemos. Una vez introducida pulsamos la tecla ( **Enter** ) ...



Introducimos de nuevo la misma contraseña para verificar que no nos hemos equivocado, pulsamos la tecla ( **Enter** ) y finalizará aquí mismo la instalación del conjunto de paquetes que forman **LAMP**.

Ahora desde otro ordenador que tengamos conectado a nuestra red de área local vamos a comprobar que nuestro servidor web funciona ejecutando un navegador e introduciendo la **IP fija** de nuestro servidor Web, en mi caso *http://10.0.2.10* ...

## It works!

This is the default web page for this server.

The web server software is running but no content has been added, yet.

Como podemos ver esto es la pagina por defecto que nos ofrece el **servidor web Apache** para cerciorarnos de que todo ha ido bien. En nuestro disco duro esta pagina por defecto se ubica en el directorio */var/www* con lo cual quiere decir que todo lo que este dentro de ese directorio puede ser

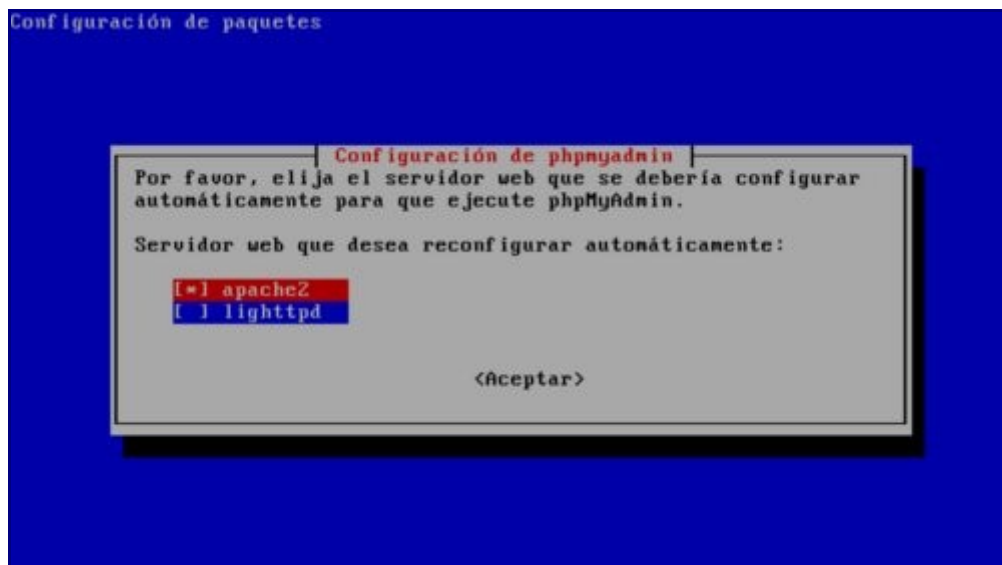
visto vía web con tan solo poner nuestra *IP fija* ” de momento “.

Ahora que ya sabemos donde está y por qué se muestran nuestros contenidos en un navegador web vamos a instalar el **administrador de bases de datos MySQL** llamado **PhpMyAdmin**. Todos los paquetes que necesitamos para poder administrar nuestras bases de datos han sido instaladas junto al grupo de paquetes **lamp-server** pero para administrarlas de una forma mas fácil e intuitiva vamos a instalar el **interface web phpmyadmin** con el que podremos hacer de todo con nuestras bases de datos pero con un entorno web y no desde la terminal simplificando aun mas nuestra tarea. Para instalarlo ejecutaremos lo siguiente ...

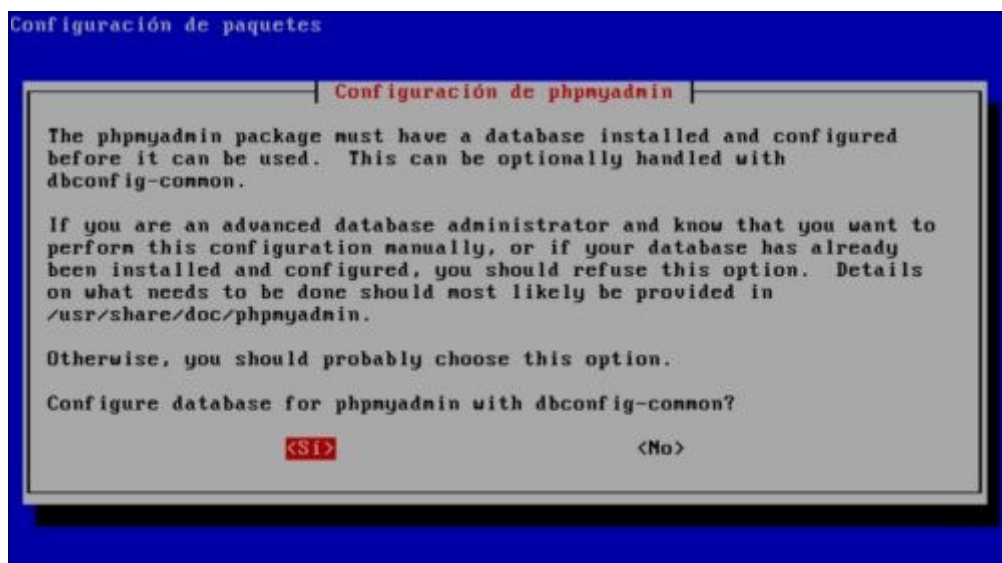
~ codigo ~

```
sudo aptitude install phpmyadmin
```

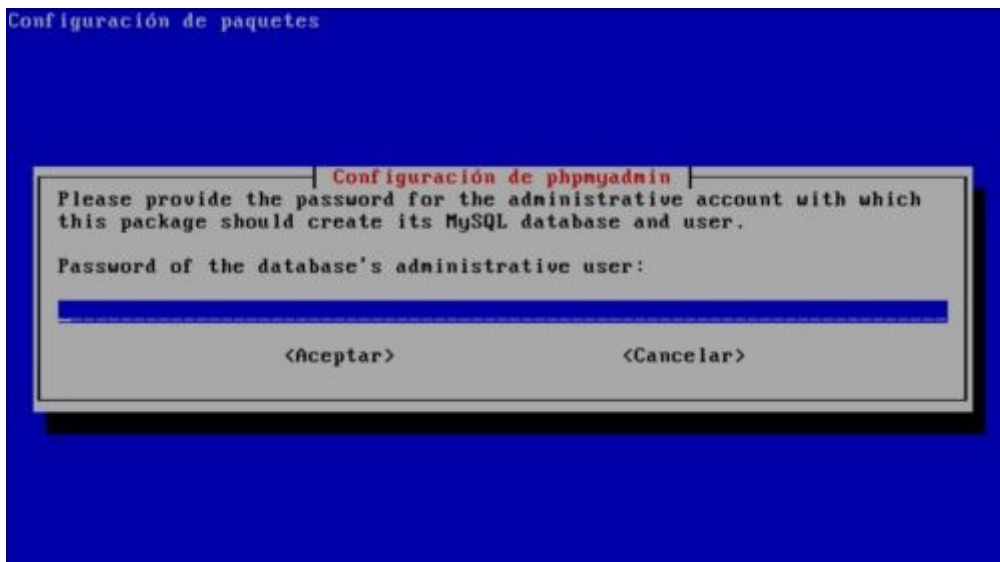
Durante la instalación llegaremos a este punto donde tendremos que indicarle para que tipo de **servidor Web** queremos que adapte la instalación ...



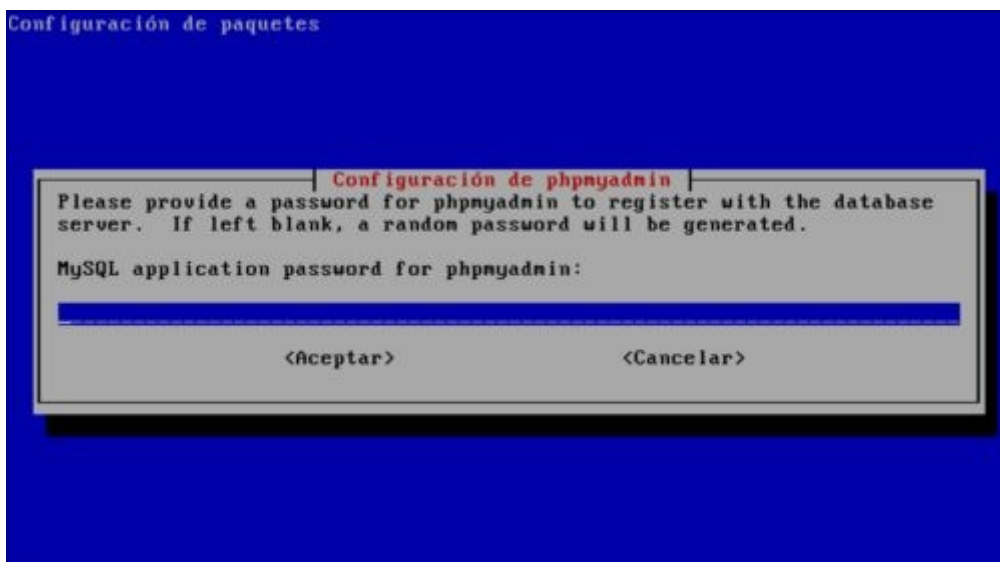
En nuestro caso marcamos la opción ( **Apache2** ) con la barra espaciadora y pulsamos la tecla ( *Enter* ) ...



**PhpMyAdmin** necesita de una base de datos para funcionar. Esta la crearemos ahora mismo seleccionando la opción ( *Si* ) y pulsando la tecla ( *Enter* ) ...



Para crear la base de datos el sistema nos pide la contraseña que antes le indicamos para el súper usuario ( **root** ). Se la introducimos y pulsamos la tecla ( *Enter* ) ...

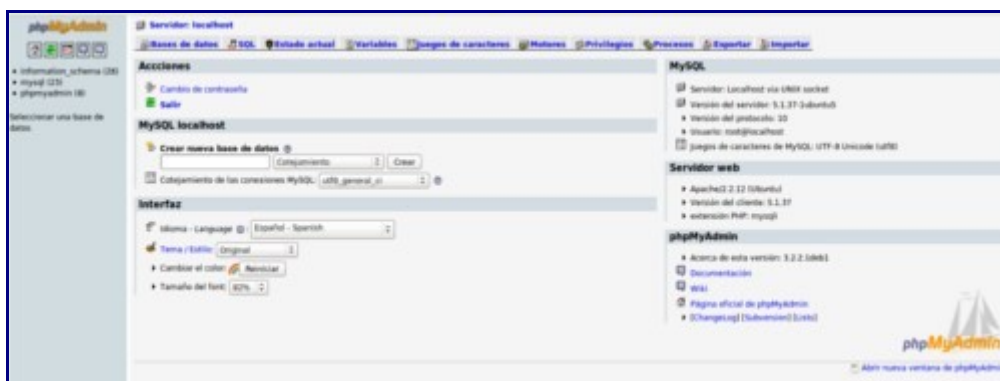


Ahora la aplicación **PhpMyAdmin** necesita que le indiquemos una *contraseña* con la que podremos acceder a el. Introducimos la que mas nos guste y nos acordamos de apuntárnosla para que no se nos olvide y pulsamos la tecla ( *Enter* ) ...



Le volvemos a introducir la nueva contraseña, pulsamos de nuevo la tecla ( *Enter* ) y con esto terminaremos la **instalación de PhpMyAdmin**. Ahora nos vamos a otro ordenador que tengamos conectado a la misma red local y ejecutamos un navegador con el que podremos ver el entorno de administración desde la dirección *http://tuipfija/phpmyadmin*, en mi caso *http://10.0.2.10/phpmyadmin* ...

Introducimos como usuario a ( *root* ) y como contraseña la que le indicamos anteriormente durante el manual. A continuación veremos lo siguiente ...



## Apuntes ...

Una vez llegados a este apartado ya sabemos como acceder a interface web de **phpmyadmin** desde otro ordenador conectado a nuestra red local para crear, borrar, modificar, entre otros nuestras bases de datos **MySQL**.

Ej : *http://ipdetuservidorweb/phpmyadmin*

Por otro lado sabemos cual es el directorio raiz de nuestro servidor Web ...

Ej : */var/www*

Todo lo que copiamos en ese directorio será visible desde otro ordenador que acceda a nuestro *puerto 80* como sitio raíz de nuestro **servidor web**.

Mas adelante configuraremos otras cosas del **servidor web Apache** así que la configuración de los **Virtual Host** entre otras cosas pasaran a formar parte de otro capitulo.

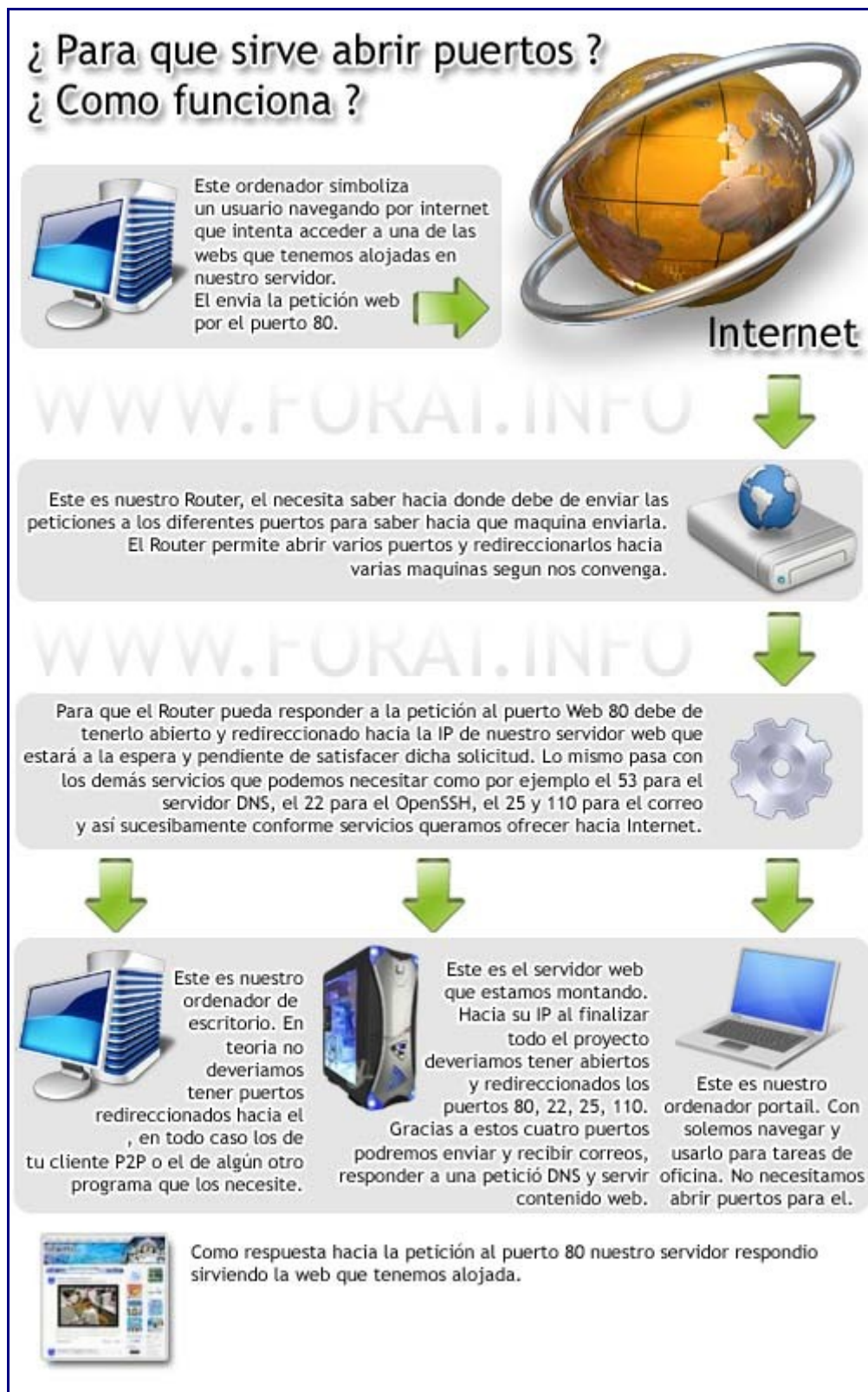
## Vol 4 – Abrir y redirigir puertos desde nuestro Router



Cuando contratamos un **servidor** en un alojamiento externo no tenemos que pensar en abrir puertos y redireccionarlos hacia una maquina en si porque de eso se ocupa la empresa que nos provee de dicho alojamiento, pero cuando el **hosting** lo tenemos en casa la cosa cambia. Cada uno de los servicios que usamos a diario en Internet necesitan un **puerto** para poder comunicarse, cuando navegamos por la red, enviamos y recibimos un correo, nos conectamos remotamente, etc ...

Para aprender bien como funciona esto de los **puertos** y los **redireccionamientos** he pensado en crear una escena como esta basada tan solo en el sistema de puertos. En los próximos **tutoriales** explicaremos el funcionamiento total desde que un usuario introduce un sitio web en su navegador hasta que nuestro servidor llegue a responder, aunque de momento hablaremos tan solo nuestro **Router** y como actúa la **apertura de puertos** y el **redireccionamiento** de los mismos que es lo que necesitamos en este momento.

En el siguiente esquema podemos ver como funciona una petición web al puerto 80 ...



Cada uno de nuestros Routers son diferentes pero prácticamente todos tienen las mismas funciones. Para acceder al **Router** tan solo tendréis que **abrir** un navegador e introducir la **IP interna del Router**. En el capítulo donde hablábamos de la [configuración de red](#) aprendimos a saber la IP de nuestro **Router**.

Básicamente para casi todos los Router suele haber algún alma caritativa que ha creado un artículo sobre como **abrir los puertos** de un tipo de Router en especial. Si buscáis en [Google](#) y tan solo ponéis ( *abrir puertos marca y modelo de tu router* ) suelen salir muchísimos tutoriales sobre como hacerlo, así que si no os atrevéis vosotros mismos siempre tendréis esta opción.

Por cierto, los puertos que tendremos que abrir y redireccionar serán el **80** para el **servidor Web** y el **22** para **conexión remota por SSH o SFTP** hacia la IP de nuestro [servidor web en Linux Ubuntu Server](#). Una vez redireccionados estos puertos podremos aceptar peticiones a los mismos desde Internet. Cada **Router** tiene un diferente fabricante y cada fabricante incluye un software diferente para configurar su modelo de **Router**, así que me es imposible crear un tutorial generalizado donde pueda tratar este tema pero lo que sí puedo hacer es terminar el siguiente manual que hablará sobre **como encontrar nuestro servidor desde Internet** y así podremos probar con certeza si tenemos bien abiertos y redireccionados los puertos desde el **Router** hacia nuestro servidor web.

## Vol 5 – Encontrando nuestro servidor desde Internet con No-Ip



Cuando contratamos una línea **ADSL** todo el mundo se fija en cuanto ofrecen de descarga sin prestar atención a cuanto nos dan de subida. La descarga todo el mundo sabe para que la necesitamos pero la subida por si no lo sabéis será algo que en este caso también necesitaremos tanto o mas que la descarga ya que lo que vamos a tener en casa es un [Servidor Web](#) y necesitaremos enviar paquetes hacia aquellos que requieran de ellos desde sus ordenadores nada mas y nada menos que desde nuestra **ADSL** y con nuestra subida.

Otro factor que deberíamos tener en cuenta es si necesitamos tener **IP fija** o no. La mayoría por no decir prácticamente todas las **ADSL** que podemos encontrar en el mercado ofrecen una **IP dinámica** por defecto que va cambiando cada cierto tiempo o simplemente cuando reiniciemos el **Router**. La **IP dinámica** no es que sea una IP mala ya que no influye en ningún aspecto en el rendimiento de la navegación, descarga, subida de archivos, etc ... Para un usuario normal no es un problema pero si vamos mas allá y nos proponemos el montarnos nuestro propio servidor en casa seria lo suyo tener **IP fija** ya que nos evitaríamos algún que otro problemilla.

En mi caso y como en el casi todos vosotros tengo una **ADSL** con **IP Dinámica** y si la tuviera fija con aprendérmela seria suficiente para que pudiera encontrar mi servidor desde internet pero como no la tengo necesito un **redireccionamiento de DNS dinámico** como el que ofrece la empresa **No-IP** entre otras. Para los que soléis leer mis tutoriales el servicio de **No-IP** os será familiar ya que lo hemos usado en otros proyectos como por ejemplo en el anterior [Servidor Web con Linux Debian](#). Por mas que pruebe otros servicios siempre acabo trabajando con **No-IP**. Hace meses que vigilo el funcionamiento de este servicio y funcionan al *99% de uptime*.

## ¿ Como funciona el redireccionamiento DNS dinamico ?



El funcionamiento gratuito que nos ofrece la empresa **No-IP** no es otro que el ir relacionando nuestra IP con un **subdominio** que nosotros mismos elegiremos durante el registro. Dicho subdominio siempre estará relacionado con nuestra **IP** aunque sea dinámica ya que para ello instalaremos un cliente en nuestro servidor que se encargará de enviarle cada cierto tiempo un aviso con la IP que tenemos actualmente así el subdominio siempre que lo introduzcamos en un navegador será capaz de encontrar nuestro servidor desde internet

Si ya estáis registrados en **No-IP** y ya disponéis de un dominio ya sea el mítico *sytes.net* o el que sea podéis saltaros el proceso de registro y actuar directamente sobre la máquina con la instalación del cliente. Si no es así a continuación vamos a registrarnos desde la [web oficial de No-Ip desde aquí](#) y veremos lo siguiente ...

## Registro ...



Para registrarnos gratuitamente tendremos que pulsar sobre el enlace llamado ( *Create Account* ) que está situado a la izquierda del botón en naranja llamado ( *Login* ). Pulsamos sobre el enlace y veremos lo siguiente ...

The image shows the No-IP website's registration page. At the top, there is a dark blue header with the No-IP logo (a green circle with 'ip' and a slash) and the text 'no-ip The DNS Service Provider'. Navigation links for 'Home', 'Download', 'Services', 'Support', and 'Company' are in the center, while 'Home', 'Contact Us', and 'Login' are on the right. A large blue banner below the header says 'No-IP is Free, Sign up Now!'. Below this, a breadcrumb trail shows 'Home > Free SignUp'. The main heading is 'Create Your No-IP Account'. A link '(sign in here)' is provided for existing users. The form is divided into three light blue sections: 'Account Information' with fields for Email, Password, and Confirm Password; 'About You' with fields for First Name, Last Name, a dropdown for 'How did you hear about us?', and a Zip/Postal Code field; and 'Account Access' with a dropdown for 'Security Question', a text field for 'Your Answer', and a 'Birthday' field with month, day, and year sub-fields.

ip no-ip  
The DNS Service Provider

Home Download Services Support Company

Home Contact Us Login

No-IP is Free, Sign up Now!

Home > Free SignUp

Create Your No-IP Account

If you already have an account then you can [\(sign in here\)](#)

**Account Information:**

Email:

Password:

Confirm Password:

**About You:**

First Name:

Last Name:

How did you hear about us?:

Zip/Postal Code:

**Account Access:**

Security Question:

Your Answer:

Birthday:

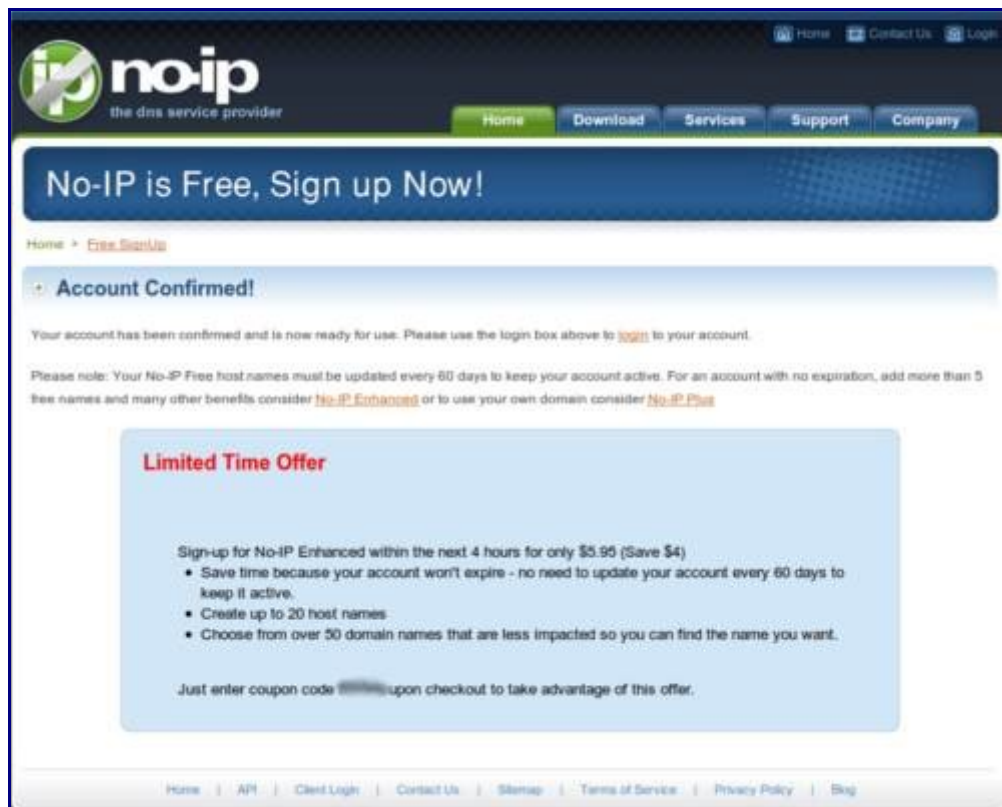
Rellenamos todo el formulario con nuestros datos reales y aceptamos los términos de uso marcando la ultima casilla. Una vez correctamente rellenado pulsamos sobre el botón inferior llamado ( *I Accept, Create my Account* ) y veremos lo siguiente ...



Ahora necesitamos acceder a nuestro correo electrónico para confirmar la activación del servicio ...

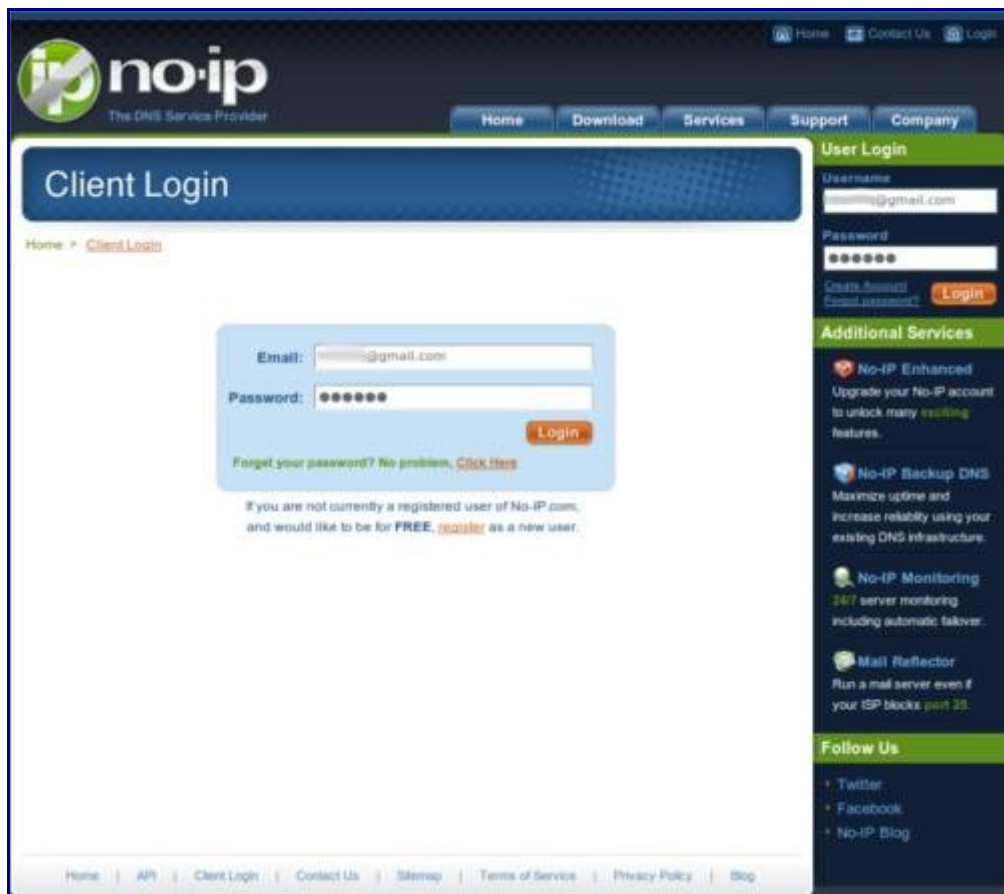


Pulsamos sobre el enlace de activación y veremos lo siguiente ...



Como podemos ver el registro ha sido satisfactorio. En este ultimo paso nos informan de que el subdominio que registremos a nuestro nombre gratuitamente caducará cada 60 días. Esto no es un problema ya que cada 60 días recibiremos un email dándonos la opción a que lo podamos reactivar por 60 días mas y así podemos ir haciendo infinitamente.

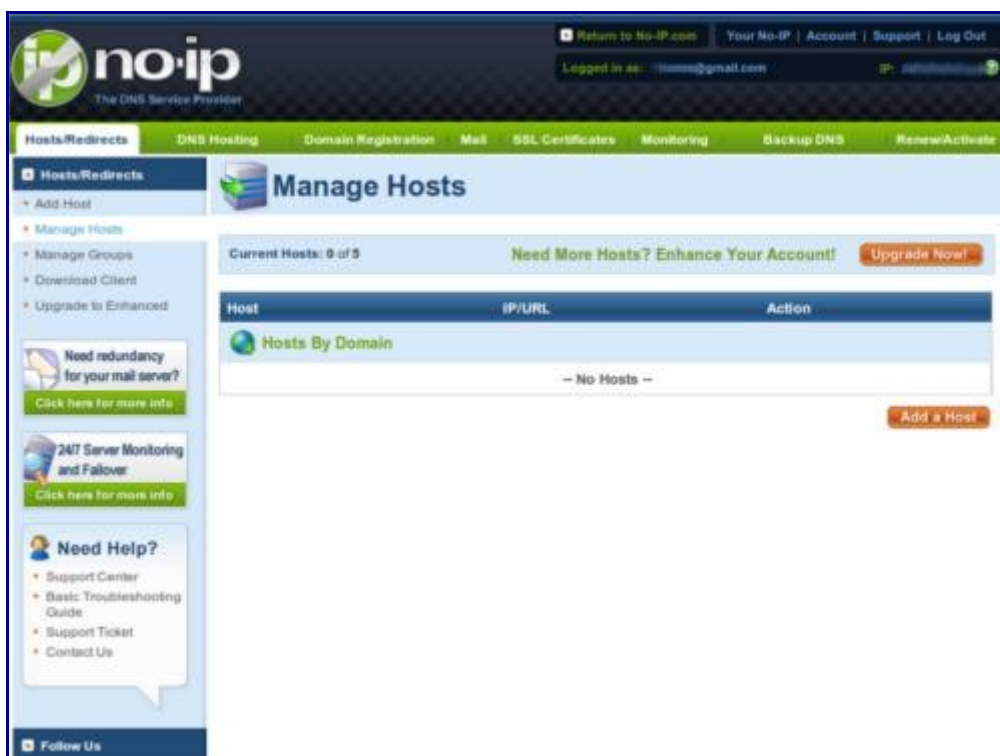
Ahora vamos a registrar un subdominio. Para hacer esto necesitaremos identificarnos en la web de **No-Ip**, podemos identificarnos desde la portada de la web o desde el enlace ( *Login* ) de esta ultima pantalla ...



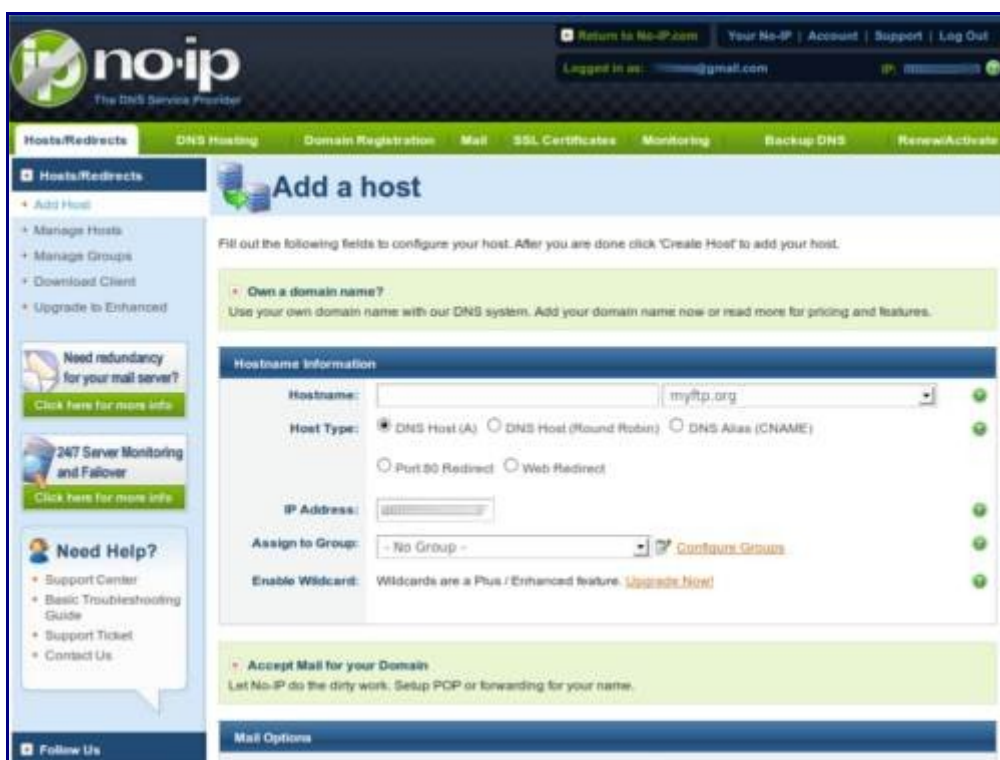
Pulsamos sobre el botón ( *Login* ) y continuamos ...



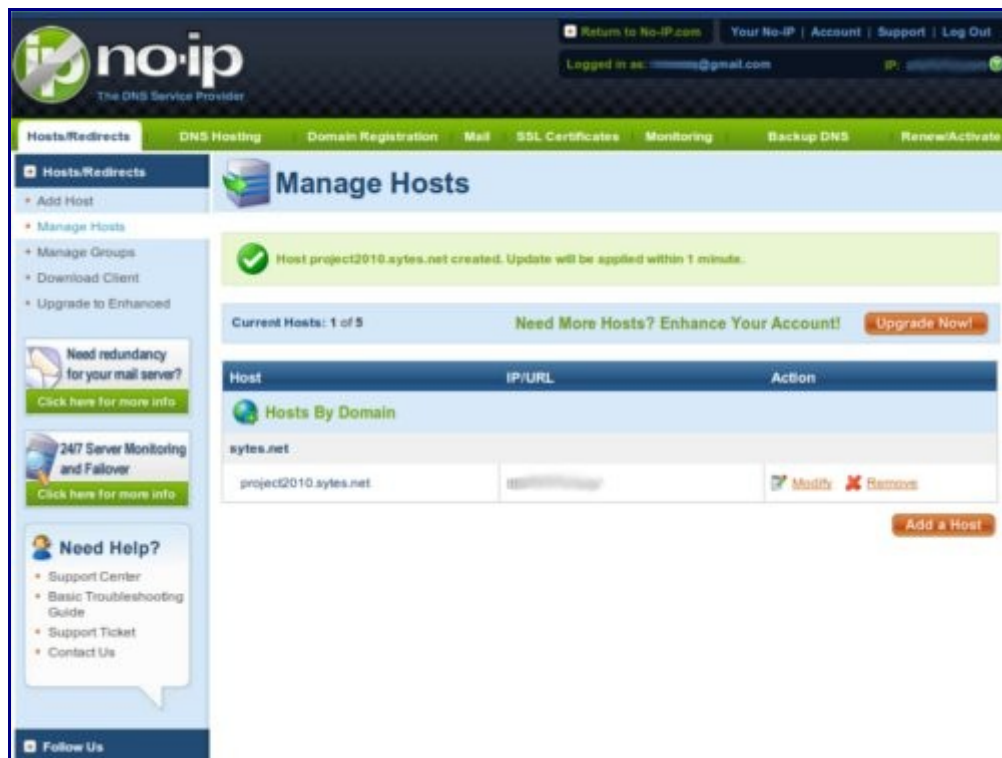
Ahora vamos a **crear el subdominio** que mas nos guste para encontrar nuestro servidor desde Internet. Pulsamos sobre el enlace situado a la izquierda llamado ( *Hosts/Redirects* ) y veremos lo siguiente ...



A continuación pulsamos sobre el botón ( *Add Host* ) ...



Para comenzar vamos a introducir en la casilla ( *Hostname* ) el nombre que queramos, yo pondré el nombre ( *Project2010* ). Seguidamente tenemos que seleccionar el dominio que no-ip nos deja para poder redireccionarnos. Ojo seleccionar el que mas os guste pero fijaos de que esté debajo de **No-IP Free Domains** para que os salga gratis. Yo voy a seleccionar ( *sytes.net* ) así quedará mi dirección como **project2010.sytes.net**. El resto lo dejamos tal cual y pulsamos sobre el botón ( *Create Host* ) ...



Bien !! Ya tenemos la dirección web con la que encontraremos nuestro servidor desde Internet. Ahora necesitamos configurar el cliente con **Linux Ubuntu Server** para que envíe el aviso con nuestra **IP externa** cada X tiempo que le indiquemos.

Su configuración en según que **Router** puede aparecer sin problemas y seguramente allí mismo podéis configurar el cliente y así prescindir de configurar el cliente en el servidor. Si vuestro **Router** no dispone de este servicio no os preocupéis ya que estamos montando un [Servidor Web en Linux Ubuntu Server](#) y vamos a configurar el cliente sin problemas, así que arrancar vuestro servidor, identificaros y comenzamos con la instalación ...

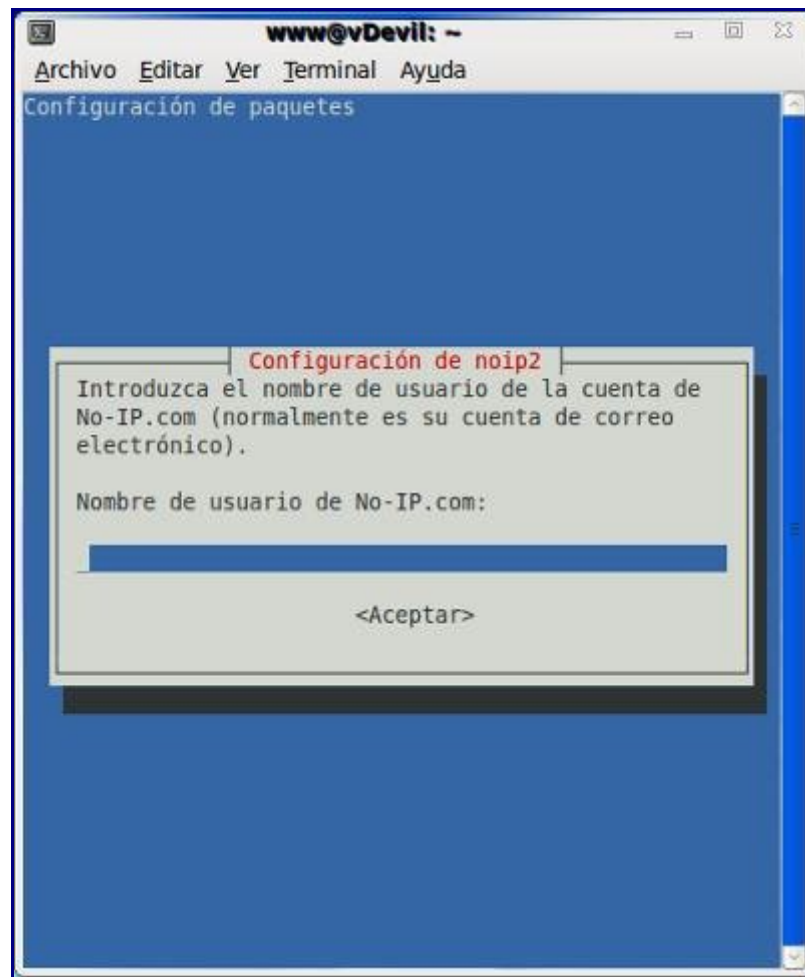
## Instalación y configuración del cliente No-IP

Para comenzar vamos a instalar el software cliente de no IP con los siguiente comandos ...

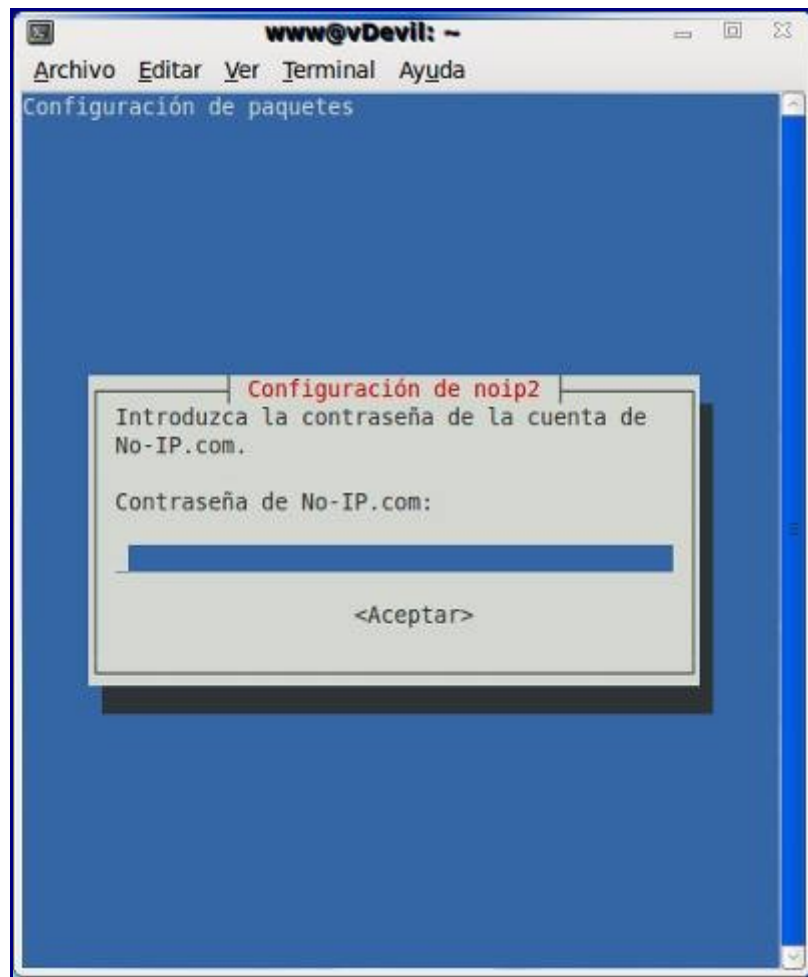
```
~ codigo ~
```

```
sudo aptitude install no-ip
```

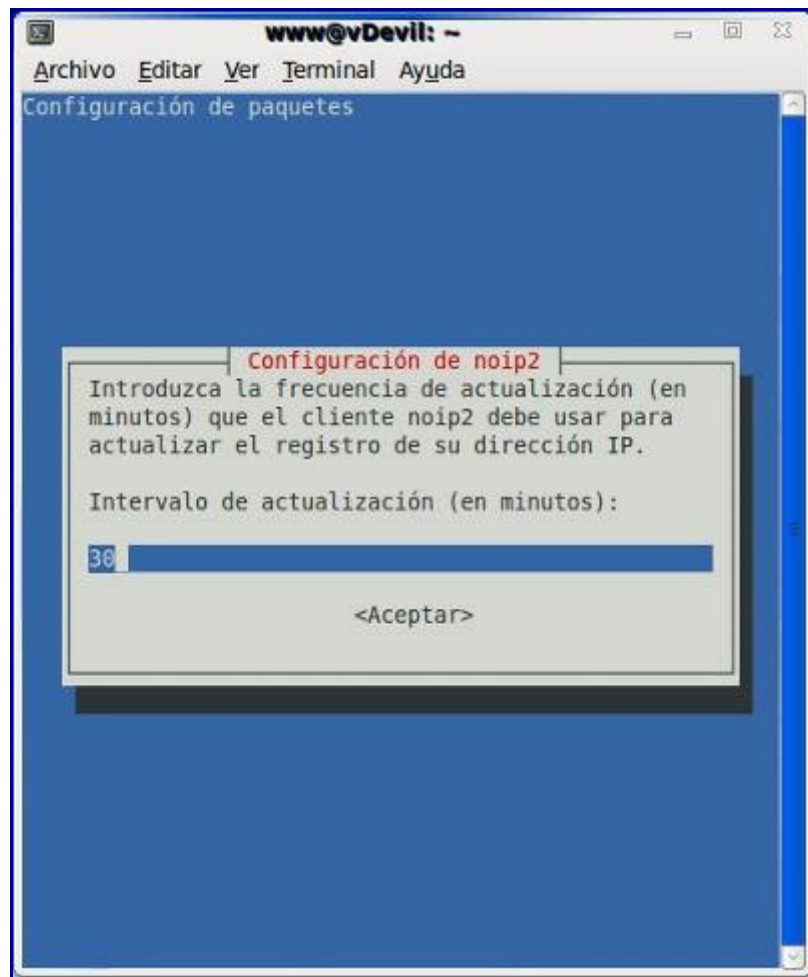
Durante la instalación vamos a configurarlo ...



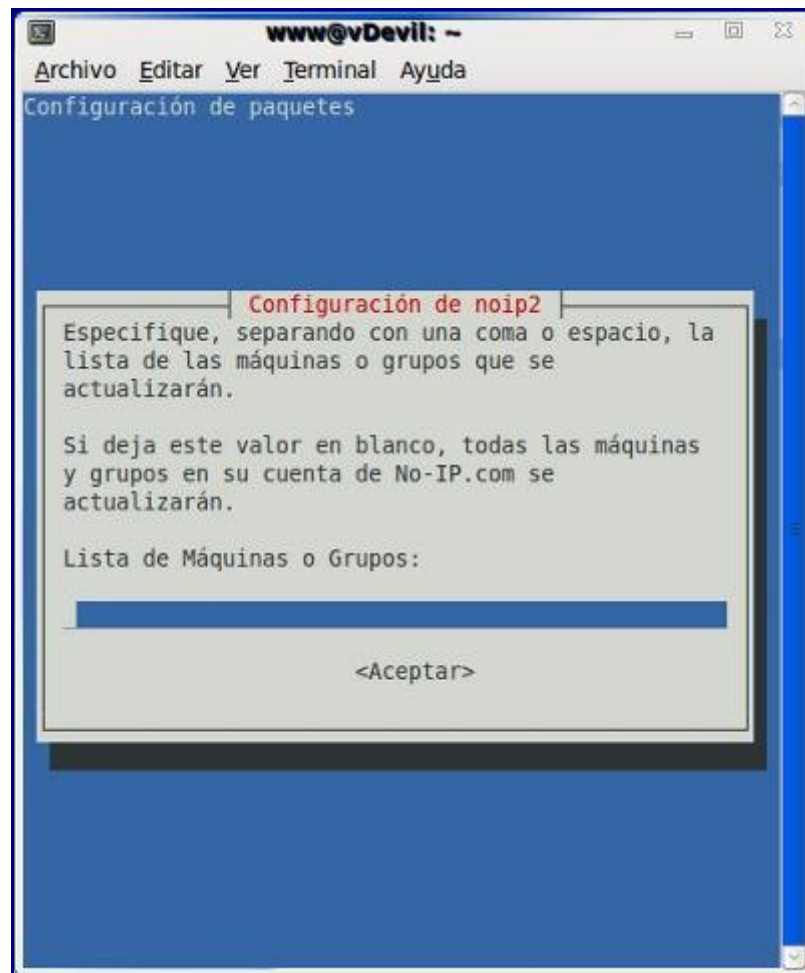
Introducimos la cuenta de correo con la que nos registramos y pulsamos la tecla ( *Enter* ) ...



Introducimos la contraseña con la que nos registramos en No-Ip y pulsamos la tecla ( *Enter* ) ...

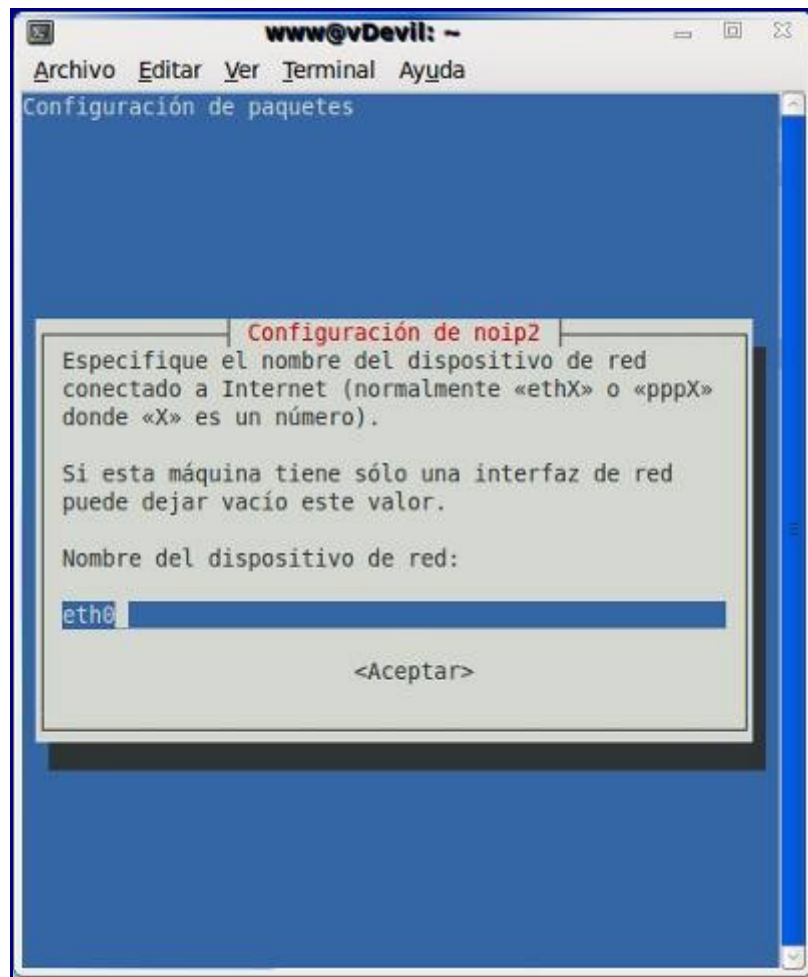


En este paso le diremos al cliente cada cuanto tiempo tiene que enviar el aviso a no-ip sobre nuestra IP. Por defecto tenemos unos 30 minutos, yo voy a introducirle 15, no me gusta perder el tiempo. Una vez ajustado continuamos pulsando la tecla ( *Enter* ) ...



Como no creamos ningún grupo en nuestra área personal pulsamos la tecla ( *Enter* ) y continuamos

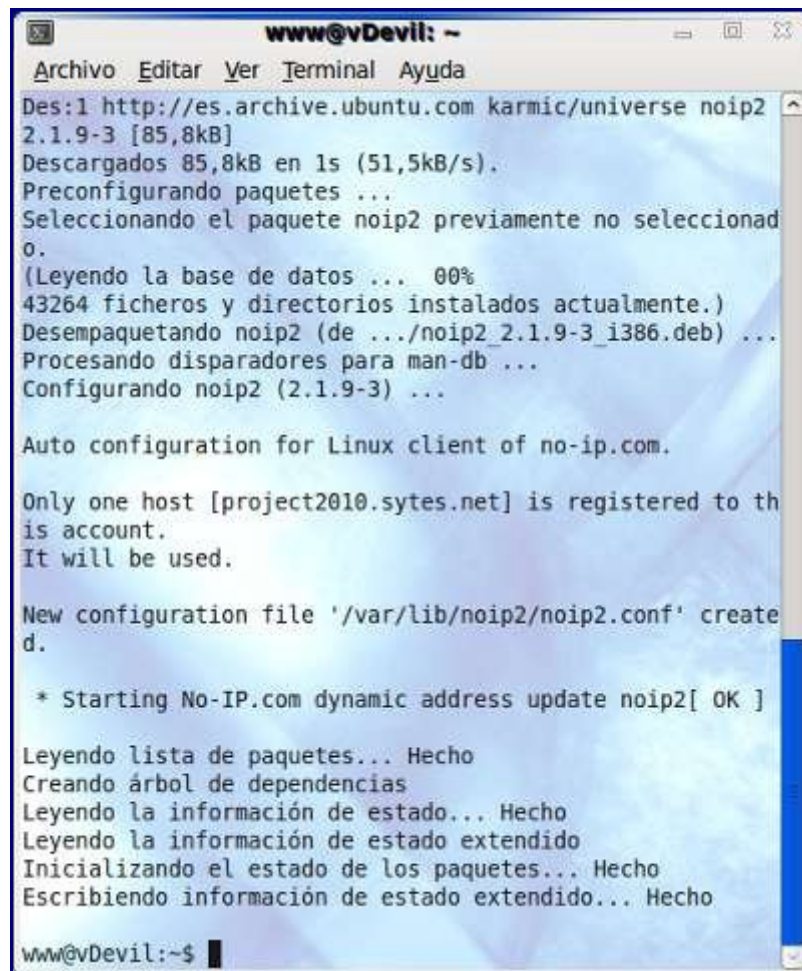
...



Introducimos el identificador de nuestra tarjeta de red, en mi caso y en la mayoría de tarjetas de red internas es ( *eth0* ), continuamos pulsando la tecla ( *Enter* ) ...



Dejamos esta opción en ( *No* ) y pulsamos la tecla ( *Enter* ) ...



```
www@vDevil: ~
Archivo Editar Ver Terminal Ayuda
Des:1 http://es.archive.ubuntu.com karmic/universe noip2
2.1.9-3 [85,8kB]
Descargados 85,8kB en 1s (51,5kB/s).
Preconfigurando paquetes ...
Seleccionando el paquete noip2 previamente no seleccionado.
(Leyendo la base de datos ... 00%
43264 ficheros y directorios instalados actualmente.)
Desempaquetando noip2 (de .../noip2_2.1.9-3_i386.deb) ...
Procesando disparadores para man-db ...
Configurando noip2 (2.1.9-3) ...

Auto configuration for Linux client of no-ip.com.

Only one host [project2010.sytes.net] is registered to this
account.
It will be used.

New configuration file '/var/lib/noip2/noip2.conf' created.

* Starting No-IP.com dynamic address update noip2[ OK ]

Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Leyendo la información de estado extendido
Inicializando el estado de los paquetes... Hecho
Escribiendo información de estado extendido... Hecho

www@vDevil:~$
```

Si nos fijamos en la **Terminal** veremos que automáticamente se ha registrado el subdominio [project2010.sytes.net](http://project2010.sytes.net). Ahora necesitamos saber si el **subdominio** resuelve correctamente nuestra IP. Para hacer esto vamos a usar la herramienta **nslookup** la cual nos traduce nombres a IP's del siguiente modo ...

- código -

```
nslookup subdominio.sytes.net
```

En el apartado *Address* podéis ver vuestra IP. Si ahora abrimos un navegador en otra maquina y ponemos esa *IP* como **dirección web** veremos nuestro servidor y si ponemos *subdominio.sytes.net* también.

Ahora si vuestra **IP** cambia no importa ya que gracias a vuestro **dominio** *subdominio.sytes.net* siempre podréis encontrar vuestro **servidor** os cambien la IP las veces que os las cambien.

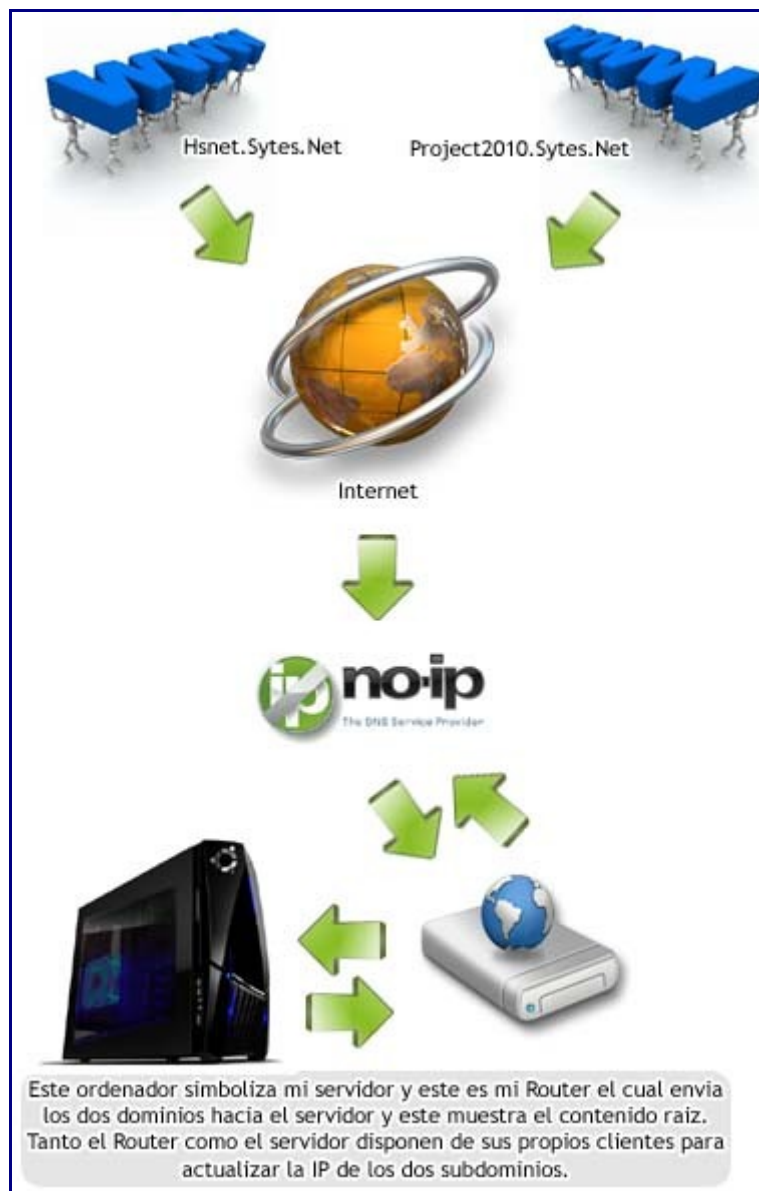
## Vol 6 – Servidor web Apache y su VirtualHost con NoIp



Antes de explicar como montar los **VirtualHost** para poder apuntar cuantos dominios nos venga en gana incluidos los comerciales, he pensado que no estaría de mas aprender un poco el como funcionan los **VirtualHost** del **Apache** y crear uno para la dirección web que registramos en el capitulo anterior con **No-IP.com** ...

Desde hace unos años tengo registrado el subdominio [hsnet.sytes.net](http://hsnet.sytes.net) que apunta hacia el **servidor web** que tengo en mi casa. Ahí tengo montada una pequeña web donde tengo algunas cosillas más. En el articulo anterior y para este [servidor web 2010 bajo Linux Ubuntu Server](#) he registrado el subdominio [project2010.sytes.net](http://project2010.sytes.net). En principio al introducir en el navegador la dirección [hsnet.sytes.net](http://hsnet.sytes.net) y [project2010.sytes.net](http://project2010.sytes.net) apuntan hacia la **IP** de mi **Router** con lo que el **servidor web Apache** responde a dicha petición con lo que hay en la raíz del servidor web, osea en el directorio `/var/www`

No quiero dejar de usar el antiguo subdominio [hsnet.sytes.net](http://hsnet.sytes.net) pero por otro lado quiero usar el nuevo que he registrado llamado [project2010.sytes.net](http://project2010.sytes.net). Por los clientes que envían la **IP** a la empresa **No-IP** no tengo problemas ya que la de [hsnet.sytes.net](http://hsnet.sytes.net) la envía el cliente que tengo en el **Router** y la de [project2010.sytes.net](http://project2010.sytes.net) la envía el cliente que tengo instalado en mi servidor. Con lo que las dos direcciones siempre apuntan hacia la raíz de mi **servidor Web** ...



En principio los dos subdominios apuntan hacia mi **Router** y el mismo hacia el directorio raíz /var/www así que para que cada uno de los dos **subdominios** apunten hacia un directorio diferente en nuestro disco duro necesitamos usar los **VirtualHost** del **Apache**.

Si tenemos estos dos **subdominios** apuntando hacia nuestro **servidor** en mi caso voy a dejar que **hsnet.sytes.net** apunte hacia la raíz y ahí dejaré la **pequeña web** que tengo. En este caso y para que podáis practicar todos vamos a crear un **VirtualHost** para que **project2010.sytes.net** apunte hacia otro directorio que no sea el raíz del servidor web sino hacia otro directorio donde alojaremos otros contenidos.

Para comenzar arrancaremos nuestro **servidor web con Linux Ubuntu Server**, nos identificamos y vamos a crear un directorio dentro de nuestro /home donde alojaremos nuestro contenido web como por ejemplo ...

~ código ~

```
sudo mkdir /home/usuario/web
```

Ahora vamos a crear un enlace simbólico desde el directorio Raíz de nuestro servidor web hacia el directorio que acabamos de crear dentro de nuestro directorio personal dentro de la */home* del siguiente modo ...

~ código ~

```
sudo ln -s /home/usuario/web /var/www/web
```

Con esto lo que conseguimos es que todo lo que esté dentro del directorio */home/usuario/web* se vea reflejado en un directorio llamado **/web** dentro de nuestro directorio raíz real o sea */var/www* como si fuera un directorio quedando en la URL **http://usuario.sytes.net/web**

Ahora vamos al directorio de configuraciones del **servidor web Apache** ...

~ código ~

```
cd /etc/apache2
```

Para crear los **VirtualHost** en **Apache** existe un directorio llamado **sites-available** y allí es donde crearemos los **VirtualHost**. Accedemos al directorio ...

```
cd sites-available
```

Copiaremos el archivo por defecto default para crear el primer VirtualHost en este paso para la dirección de nuestro server ...

~ código ~

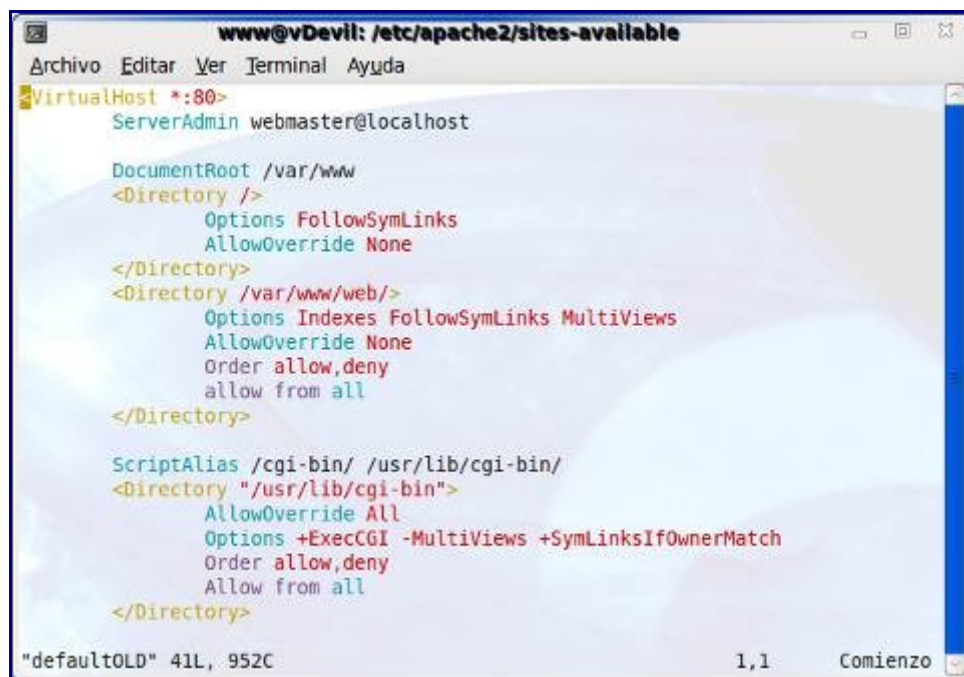
```
sudo cp default nuestrodominio.sytes.net
```

A continuación editaremos el archivo **nuestrodominio.sytes.net** del siguiente modo ...

~ código ~

```
sudo vi nuestrodominio.sytes.net
```

Este archivo *default* es el que usa *Apache* para decir cual es la raíz de nuestro servidor Web. En este caso y como podemos ver tenemos el archivo por defecto ...



```
www@vDevil: /etc/apache2/sites-available
Archivo Editar Ver Terminal Ayuda
VirtualHost *:80>
    ServerAdmin webmaster@localhost

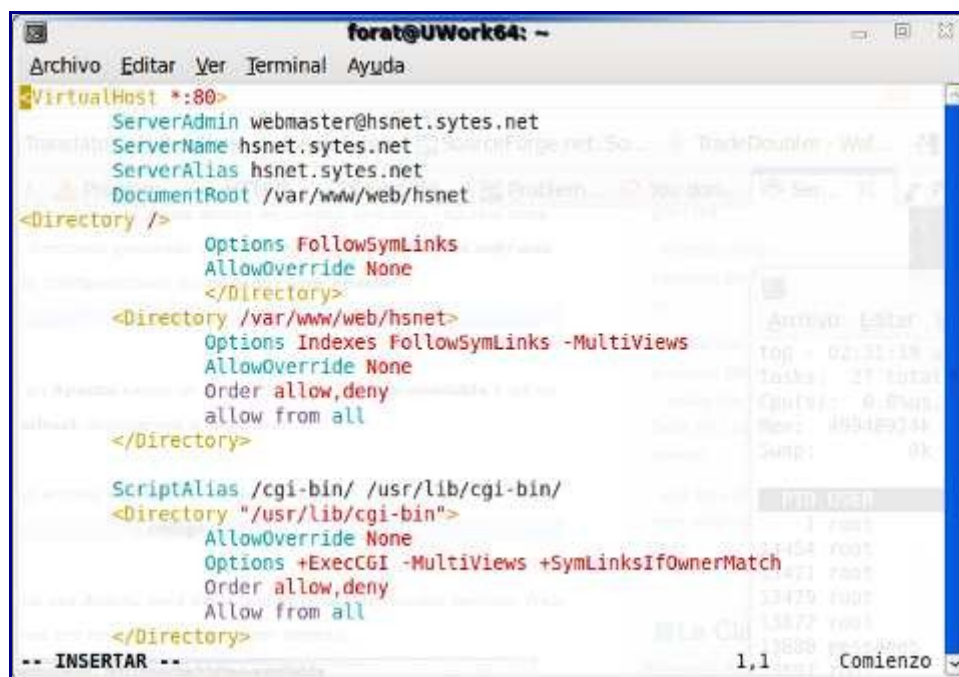
    DocumentRoot /var/www
    <Directory />
        Options FollowSymLinks
        AllowOverride None
    </Directory>
    <Directory /var/www/web/>
        Options Indexes FollowSymLinks MultiViews
        AllowOverride None
        Order allow,deny
        allow from all
    </Directory>

    ScriptAlias /cgi-bin/ /usr/lib/cgi-bin/
    <Directory "/usr/lib/cgi-bin">
        AllowOverride All
        Options +ExecCGI -MultiViews +SymLinksIfOwnerMatch
        Order allow,deny
        Allow from all
    </Directory>

"default0LD" 41L, 952C                               1,1 Comienzo
```

Además de modificar algunas líneas tendremos que agregar otras que no están en el archivo de configuración por defecto. En mi caso dispongo de dos **subdominios** que quiero **alojar**. El de

**Hsnet.Sytes.Net** para el directorio `/hsnet` que estará dentro del directorio que creamos anteriormente `"/web"` así que agregaré este a la nueva configuración quedándose así ...



```
forat@UWork64: ~  
Archivo Editar Ver Terminal Ayuda  
VirtualHost *:80  
    ServerAdmin webmaster@hsnet.sytes.net  
    ServerName hsnet.sytes.net  
    ServerAlias hsnet.sytes.net  
    DocumentRoot /var/www/web/hsnet  
    <Directory />  
        Options FollowSymLinks  
        AllowOverride None  
    </Directory>  
    <Directory /var/www/web/hsnet>  
        Options Indexes FollowSymLinks -MultiViews  
        AllowOverride None  
        Order allow,deny  
        allow from all  
    </Directory>  
    ScriptAlias /cgi-bin/ /usr/lib/cgi-bin/  
    <Directory "/usr/lib/cgi-bin">  
        AllowOverride None  
        Options +ExecCGI -MultiViews +SymLinksIfOwnerMatch  
        Order allow,deny  
        Allow from all  
    </Directory>  
-- INSERTAR --  
1,1 Comienzo
```

Si comparáis vuestro archivo *default* con el mio veréis que he agregado algunas nuevas opciones como por ejemplo **ServerName** y **ServerAlias**, también he agregado la ruta `/var/www/web/hsnet` en el **DocumentRoot** y en la línea, etc ...

Comparadlo con el vuestro y rellenarlo con vuestros datos, nombre de subdominio y directorios. Una vez rellenado correctamente salimos del editor con la tecla **"ESC"** y **":wq!"**.

Antes de continuar creamos el directorio al cual apuntará *nuestrodominio.sytes.net*, en mi caso ...

```
~ codigo ~  
sudo mkdir /home/usuario/web/hsnet
```

Ahora hay que hacer que coja nuestras modificaciones, para esto vamos a usar el comando **a2ensite** del siguiente modo ...

```
~ codigo ~  
sudo a2ensite nuestrodominio.sytes.net
```

Reiniciamos el **servidor Apache** con ...

```
~ codigo ~  
sudo /etc/init.d/apache2 restart
```

Llegados a este punto si abrimos un navegador e introducimos nuestra dirección web, en mi caso **hsnet.sytes.net** veremos en realidad lo que hay dentro de `/var/www/web/hsnet`.

Esta es la manera exacta de **crear un VirtualHost en Apache bajo Linux Ubuntu Server**. Especifico la distribución porque no en todos los **sistemas operativos GNU/Linux** se configura de la misma forma.

Ahora vamos a crear otro **VirtualHost** para la dirección que registré en el artículo anterior **project2010.sytes.net**. Si dispones de otro subdominio así es como tienes que hacerlo y si no cuando tengas ganas de tener otro sabrás como hacerlo.

Para comenzar copiamos el archivo **default** por el que queramos darle al nuestro nuevo

**VirtualHost**, en mi caso **project2010** ...

~ codigo ~

**sudo cp default project2010**

Creamos un directorio dentro del directorio /web que se llame **project2010** por ejemplo, vosotros llamadle como queráis ...

~ codigo ~

**sudo mkdir /home/usuario/web/project2010**

Editamos el archivo **project2010** ...

**sudo vi project2010**

Y modificaremos las líneas que apuntan hacia nuestro **dominio** y hacia el **directorio** que acabamos de crear quedándonos así ...



```
VirtualHost *:80>
ServerAdmin webmaster@project2010.sytes.net
ServerName project2010.sytes.net
ServerAlias project.sytes.net
DocumentRoot /var/www/web/project2010

<Directory />
    Options FollowSymLinks
    AllowOverride All
</Directory>
<Directory /var/www/web/project2010/>
    Options Indexes FollowSymLinks MultiViews
    AllowOverride All
    Order allow,deny
    allow from all
</Directory>

ScriptAlias /cgi-bin/ /usr/lib/cgi-bin/
<Directory "/usr/lib/cgi-bin">
    AllowOverride None
    Options +ExecCGI -MultiViews +SymLinksIfOwnerMatch
    Order allow,deny
    Allow from all
</Directory>
```

Una vez modificado con nuestros datos salimos y guardamos pulsando la tecla " **ESC** " y seguidamente " **:wq!** ". Ahora usaremos de nuevo el comando **a2ensite** para que coja el nuevo **VirtualHost** ...

~ codigo ~

**sudo a2ensite project2010**

Reiniciamos el **servidor Apache** con ...

~ codigo ~

**sudo /etc/init.d/apache2 restart**

Y ahora tan solo nos queda introducir la URL en el navegador para ver que apunta hacia el directorio **project2010** que tenemos dentro de nuestro **/home/usuario/web/project2010** y así podríamos hacer con todos los que nos viniera en gana.

Recordamos para terminar que no necesitamos tener un **servidor DNS** instalado simplemente porque tanto **hsnet.sytes.net** como **project2010.sytes.net** ya apuntan hacia la **IP externa** que tiene mi ADSL. Yo tan solo con los **VirtualHost** les digo hacia donde tienen que ir una vez llegan a mi servidor con sus peticiones web.

Finalmente agregaremos el modulo Rewrite que nos hará falta mas adelante ...

~ codigo ~

```
cd /etc/apache2/mods-available  
sudo a2enmod rewrite
```

Cuando necesitamos **alojar dominio .com, .net, .org**, etc ... la forma sobre como hacerlo es diferente. Antes de terminar con este proyecto del **servidor web con Linux Ubuntu Server** publicaré un artículo sobre como **alojar** este tipo de **dominios**. De momento ya sabemos como hacerlo con estos gratuitos. 😊

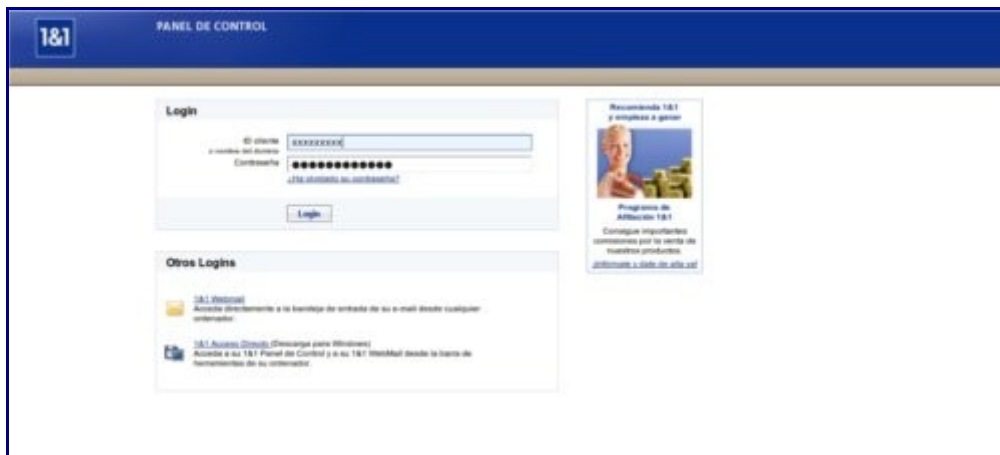
## Vol 7 – Dominios comerciales + VirtualHost en Apache



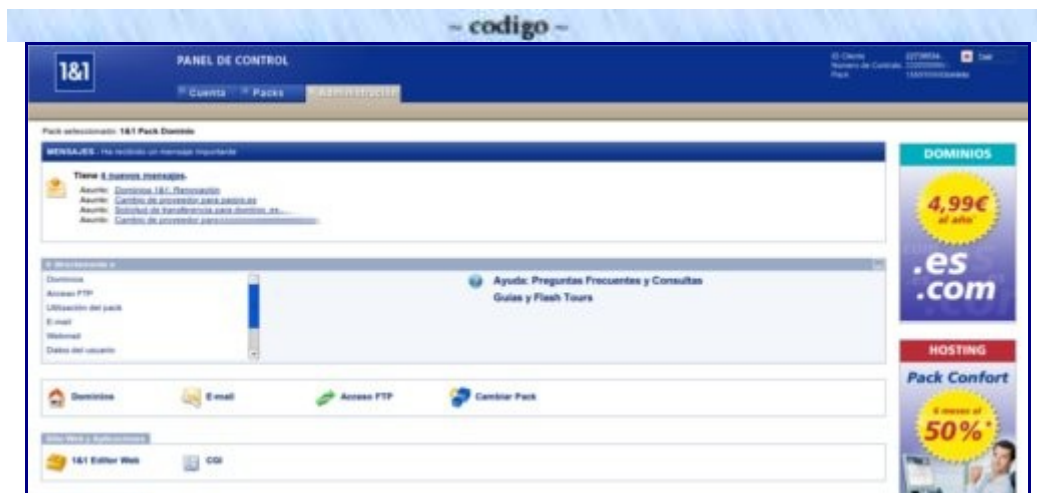
En este capítulo vamos a poder comprar un dominio comercial del estilo .com, .net, .es, .org o lo que quieras y lo redireccionaremos hacia nuestro [servidor web en Linux Ubuntu Server](#) para poder alojar nuestras webs o blogs con nuestros propios dominios. Si tenéis **dominios** comprados podéis basaros en lo aquí explicado para configurarlo aunque no todos los registradores de dominio poseen esta opción o quizás no este reseñada del mismo modo. En este caso y para que sirva de ejemplo voy a explicar mi experiencia con el registrador de dominios que yo uso. Uso [land1](#) desde hace ya bastante tiempo y nunca hasta el día de hoy he tenido ningún problema con ellos. Si quieres comprar un dominio al mejor precio que al menos yo he encontrado ( 4,99 € + IVA / año ) podéis comprar vuestros dominios [aquí mismo](#) y seguir con el manual tal y como lo voy a explicar a continuación ...

Una vez comprado el **dominio** vamos a necesitar que este apunte hacia nuestro **servidor** donde le estará esperando la web que le preparemos. Para hacer esto vamos a usar la opción para nombres canónicos. El **dominio** que **registramos** en el capítulo anterior con la empresa **No-IP** siempre apuntará hacia nuestra IP ya que el cliente que tenemos instalado en nuestro servidor se la envía cada cierto tiempo con lo que si usamos nuestra dirección *dominio.sytes.net* como nombre canónico cuando un usuario acceda a nuestro **dominio comercial** este apuntará hacia el dominio de **No-IP** y este hacia nuestro servidor.

Para comenzar accederemos al panel de control de **1and1** ...



Y veremos directamente la pestaña ( **Packs** ) en el panel de control. En ella veremos nuestro **dominio** del estilo ( **1&1 Pack Dominio** ) pulsamos sobre el enlace y veremos lo siguiente ...



Ahora pulsamos sobre el enlace ( **Dominios** ) situado en la casilla ( **Ir directamente a** ) ...



Ahora marcamos la casilla que está junto a nuestro dominio y pulsamos sobre el desplegable ( **DNS** ), como veréis nos aparecen tres opciones y tendremos que seleccionar ( **Modificar Configuración DNS** ) ...

**Datos del Dominio**

Nombre de dominio reactivox.com

Configuración DNS [Comprobar DNS](#)

Configuración general ☐ DNS ☒ Nombre canónico

---

**Configuración CNAME**

Alias \*

☐ Conozco y acepto que toda c

Aquí tenemos que seleccionar la casilla ( **Nombre canónico** ) y mas abajo nos aparecerá una casilla llamada ( **Configuración CNAME** ) en ella tenemos que introducir el [dominio que registramos en el capítulo anterior](#). Una vez hecho esto marcamos la casilla ( **Conozco y acepto que toda configuración DNS, direcciones de e-mail y configuraciones para el reenvío actuales se desactivarán.** ) y pulsamos sobre el botón ( **Enviar** ) ...

**Operación realizada**

Se ha guardado su configuración DNS. Su dominio se actualizará con la nueva configuración DNS.

No tengo muy claro cuanto tiempo se tarda en hacer la actualización para que apunte el dominio comercial hacia el nombre canónico mas que nada porque con cada uno de los que lo he hecho me ha tardado algo diferente, así que si no te apunta el dominio inmediatamente no te alarmes porque en menos de 24 horas lo tienes funcionando seguro.

Bien, ya tenemos nuestro dominio comercial configurado para que apunte hacia nuestro servidor web, ahora tendremos que configurar un **VirtualHost** en nuestro [servidor web apache](#) para que cuando alguien quiera ver que hay detrás de ese dominio que acabamos de comprar nuestro servidor sepa que ofrecerle.

Arrancamos nuestro [Project 2010](#), nos identificamos como de costumbre y comenzamos ...

## VirtualHost en Apache Web Server

Para comenzar vamos a usar el dominio que compré para hacer este manual llamado [www.reactivox.com](http://www.reactivox.com). A este también como a los gratuitos que registramos en el capítulo anterior le montaremos un **VirtualHost en nuestro Servidor Web Apache** para que sepa devolver lo que nosotros queramos y no escupa el directorio raíz de nuestro servidor web por defecto que es como esta ahora. Recordemos que en el **panel de control de 1and1** le asignamos como nombre canónico el dominio de redireccionamiento con No-IP que registramos en el capítulo anterior. Con lo que este nuevo dominio comercial en realidad apunta hacia el dominio `xxxx.sytes.net` y este mismo hacia nuestro Router y desde nuestro [Router hacia nuestro servidor](#).

A continuación vamos a crear un directorio para el nuevo dominio que queremos alojar. Para que veamos un ejemplo voy a usar **reactivox.com** y será en este en el que nos basaremos. En tu caso con cambiar el **usuario** por **tu usuario** y **reactivox.com** por tu **dominio.com** sera suficiente ...

```
- codigo -
```

```
sudo mkdir /home/usuario/web/reactivox
```

Vamos al directorio donde se encuentran nuestras rutas por defecto con ...

```
- codigo -
```

```
cd /etc/apache2/sites-available
```

Copiamos la que hay por defecto para crear la nueva ruta para nuestro nuevo dominio ...

```
- codigo -
```

```
sudo cp default reactivox.com
```

Editamos el nuevo fichero ...

```
sudo vi reactivox.com
```

Introducimos lo siguiente modificando lo necesario para que se adapte a nuestro dominio y a la ruta donde queremos situarlo ...

```
<VirtualHost *:80>
    ServerAdmin webmaster@reactivox.com
    ServerName www.reactivox.com
    ServerAlias reactivox.com
    DocumentRoot /var/www/web/reactivox

    <Directory />
        Options FollowSymLinks
        AllowOverride All
    </Directory>
    <Directory /var/www/web/reactivox/>
        Options Indexes FollowSymLinks MultiViews
        AllowOverride All
        Order allow,deny
        allow from all
    </Directory>

    ScriptAlias /cgi-bin/ /usr/lib/cgi-bin/
    <Directory "/usr/lib/cgi-bin">
        AllowOverride None
        Options +ExecCGI -MultiViews +SymLinksIfOwnerMatch
        Order allow,deny
        Allow from all
    </Directory>

    ErrorLog /var/log/apache2/error.log

    # Possible values include: debug, info, notice, warn, error, crit,
    # alert, emerg.
    LogLevel warn

    CustomLog /var/log/apache2/access.log combined

    Alias /doc/ "/usr/share/doc/"
    <Directory "/usr/share/doc/">
        Options Indexes MultiViews FollowSymLinks
```

```
AllowOverride None
Order deny,allow
Deny from all
Allow from 127.0.0.0/255.0.0.0 ::1/128
</Directory>
```

```
</VirtualHost>
```

Una vez modificadas estas líneas el servidor ya sabe hacia donde debe apuntar cuando llegue una petición desde dicho dominio. Salimos del editor ( **vi** ) y guardamos los cambios pulsando la tecla ( **ESC** ) y seguidamente ( **:wq!** ). Ahora damos de alta el nuevo sitio con ...

```
- codigo -
```

```
sudo a2ensite reactivex.com
```

Para terminal y para asegurarnos de que todo funciona bien reiniciamos el **servidor web Apache** con ...

```
- codigo -
```

```
sudo /etc/init.d/apache2 restart
```

Y listo !! Ahora tenemos el dominio que compramos apuntando no solo hacia nuestro servidor web con Linux Ubuntu Server sino que también hacia el directorio que queremos con lo que ahora tan solo necesitamos montar la web, blog o lo que te apetezca. 😊

Este proceso tendréis que hacerlo cada vez que compréis un dominio y queráis alojarlo en vuestro servidor, como podéis ver en mi caso [hsnet.sytes.net](http://hsnet.sytes.net), [project2010.sytes.net](http://project2010.sytes.net) y [reactivox.com](http://reactivox.com) son tres sitios diferentes y ofrecen diferentes contenidos aunque todos están alojados en el mismo servidor. Bonito, bonito 😊

# Vol 8 – Estadísticas web Open Source con Piwik



Para este proyecto de [Servidor Web en Linux Ubuntu Server](#) he probado varios programas de estadísticas. Casi siempre que montaba **Servidores web** usaba [Awstats](#) o [WebAlizer](#). Los dos son muy buenos programas para ver que es de tu tráfico web en cuanto a estadísticas de red en general. He pasado por varios **lectores de logs gráficos** y no tan gráficos y para este **Project 2010** había pensado incluir algo tan bueno como [Piwik](#). ¿Queréis verlo ?

Si habéis usado anteriormente **Google Analytics** cuando instaléis **Piwik** todo os va a parecer igual de familiar pero un poco mas a mano y personalizable. Si queréis ver una demo mirad la de la [web oficial](#) y si queréis ver este gran programa de **estadísticas** en directo controlando tus sitios web te invito a que lo instales del siguiente modo ...

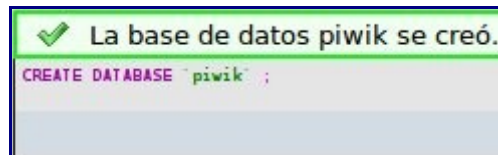
Para comenzar arrancamos nuestro [Servidor Web en Linux Ubuntu Server](#) y nos conectaremos a el desde un ordenador conectado a la red local. Accederemos a [phpmyadmin](#) para crear una base de datos desde nuestro navegador con la dirección .... <http://ipdetuserver/phpmyadmin> y veremos lo siguiente ....

The image is a screenshot of the phpMyAdmin web interface. At the top, there is a logo of a sailboat with the text "phpMyAdmin" below it. Below the logo, the text "Bienvenido a phpMyAdmin" is displayed. Underneath, there is a section for language selection labeled "Idioma - Language" with a dropdown menu showing "Español - Spanish". Below that is a login section labeled "Iniciar sesión ?" with fields for "Usuario:" (containing "root") and "Contraseña:" (masked with dots). A "Continuar" button is located at the bottom right of the login section. At the very bottom, there is a yellow banner with an information icon and the text "Las 'cookies' deben estar habilitadas."

Nos identificamos con nuestro súper usuario root de la base de datos **MySQL** y pulsamos sobre el botón ( *Continuar* ) ...



Introducimos el nombre de la base de datos nueva en la casilla ( *Crear nueva base de datos* ) y pulsamos sobre el botón ( *Crear* ) ...



Ahora volveremos a la **Terminal** de nuestro **servidor web** e instalaremos el programa de compresión **unzip** con ...

```
~ codigo ~
```

```
sudo apt-get install unzip
```

Si recordamos el artículo anterior donde hablábamos de los [subdominios con No-IP](#), hicimos que cada **subdominio** apuntara hacia un directorio diferente dentro de nuestro **servidor apache** con sus **VirtualHost**.

En mi caso el **subdominio** [project2010.sytes.net](http://project2010.sytes.net) apunta hacia la ruta `/web/project2010` de nuestro servidor web.

Yo quiero que Piwik sea un directorio mas para el Project2010 así que voy a instalarlo dentro de ese mismo directorio.

Vamos hacia la ruta donde queramos crear el directorio, en mi caso ....

```
~ codigo ~
```

```
cd /home/usuario/web/project2010
```

Una vez dentro vamos a descargarnos la última versión de Piwik del siguiente modo ...

```
~ codigo ~
```

```
sudo wget http://piwik.org/latest.zip
```

Ahora vamos a descomprimirlo con ...

```
~ codigo ~
```

```
sudo unzip latest.zip
```

Ahora vamos a instalarlo. Para hacer esto Piwik necesitará que le demos permisos de escritura y lectura para el directorio que acabamos de crear al descomprimir con ...

```
~ codigo ~
```

```
sudo chmod 777 piwik
```

Y al directorio de configuración con ...

```
~ codigo ~
```

```
sudo chmod 777 piwik/config/
```

Y al temporal ...

```
sudo chmod -R 777 piwik/tmp
```

Ahora como dijimos anteriormente **Piwik** se ha creado como un directorio con lo que desde el exterior lo veremos como <http://tusubdominiodominio.sytes.net/piwik> o <http://ipdetuservidor/web/project2010/piwik> desde vuestro navegador y veremos lo siguiente ...



Pulsamos sobre el botón ( *Siguiente* ) ...



Como podemos comprobar lo tenemos todo en verde siempre y cuando estéis siguiendo el proyecto sobre [como montar un Servidor web en Linux Ubuntu Server](#).

Ahora bajamos al final de la pagina y pulsamos sobre el enlace ( *Siguiente* ) ...

**Piwik** # Análisis Web de Código Abierto

1. ¡Bienvenido!  
2. Comprobación del sistema  
**3. Database setup**  
4. Database check  
5. Creando las tablas  
6. Configuración general  
7. Configurar una página web  
8. Etiqueta JavaScript  
9. Enhorabuena

**Database setup**

|                 |           |
|-----------------|-----------|
| database server | localhost |
| login           |           |
| password        |           |
| database name   |           |
| table prefix    | piwik_    |
| adapter         | PDO_MYSQL |

**¡Ir!**

**Estado de la instalación**

En esta parte tenemos que agregarle los datos para la base de datos que creamos anteriormente con **phpmyadmin**. En el primer apartado ( Database server ) lo dejamos tal en ( *localhost* ). En ( *Login* ) y ( *Password* ) insertaremos los datos con los que accedemos a **phpmyadmin** aunque siempre podemos crear un usuario nuevo para esta base de datos ...

**Piwik** # Análisis Web de Código Abierto

1. ¡Bienvenido!  
2. Comprobación del sistema  
**3. Database setup**  
4. Database check  
5. Creando las tablas  
6. Configuración general  
7. Configurar una página web  
8. Etiqueta JavaScript  
9. Enhorabuena

**Database setup**

|                 |                  |
|-----------------|------------------|
| database server | localhost        |
| login           | root             |
| password        | ●●●●●●●●●●●●●●●● |
| database name   | piwik            |
| table prefix    | piwik_           |
| adapter         | PDO_MYSQL        |

**¡Ir!**

**Estado de la instalación**

En ( *database name* ) insertaremos el nombre de la base de datos que creamos anteriormente con **phpmyadmin**. En resto lo dejamos tal cual. Continuaremos la instalación pulsando el botón ( **Ir** )  
....



Piwik # Análisis Web de Código Abierto

1. ¡Bienvenido!  
2. Comprobación del sistema  
3. Database setup  
4. Database check  
5. Creando las tablas  
**6. Configuración general**  
7. Configurar una página web  
8. Etiqueta JavaScript  
9. ¡Félicitations!

### Configuración general

|                         |                          |
|-------------------------|--------------------------|
| Nombre de super usuario | <input type="text"/>     |
| Contraseña              | <input type="password"/> |
| contraseña (repetir)    | <input type="password"/> |
| email                   | <input type="text"/>     |

☒ envíeme un email con las principales actualizaciones y alertas de seguridad de Piwik

☒ envíeme un email con las actualizaciones de la comunidad (nuevos plugins, nuevas características, etc.)

**Piwik** es capaz de ofrecer sus contenidos como programa de estadísticas a diferentes usuarios. Ahora vamos a crear la cuenta de administrador rellenando los cuatro campos de contacto y continuaremos con el botón ( *Ir* ) ...

Piwik # Análisis Web de Código Abierto

1. ¡Bienvenido!  
2. Comprobación del sistema  
3. Database setup  
4. Database check  
5. Creando las tablas  
6. Configuración general  
**7. Configurar una página web**  
8. Etiqueta JavaScript  
9. ¡Félicitations!

### Configurar una página web

|              |                                                 |
|--------------|-------------------------------------------------|
| website name | <input type="text"/>                            |
| website URL  | <input type="text" value="http://example.org"/> |

**Estado de la instalación**

75 % Hecho

Ahora vamos a introducir los datos de la primera web que vamos a vigilar con **Piwik**. En mi caso voy a controlar la web que me he montado para explicar [como redireccionar dominios comerciales hacia nuestro servidor](#) llamado [www.reactivox.com](http://www.reactivox.com), ahí he montado un [portal de videojuegos online](#) donde poder pasar algunos ratos libres.

Puedes controlar una web que tengas alojada en tu **servidor web** o en una que tengas en un **servidor externo**.

Rellenamos las casillas con el título de la web y url así ...

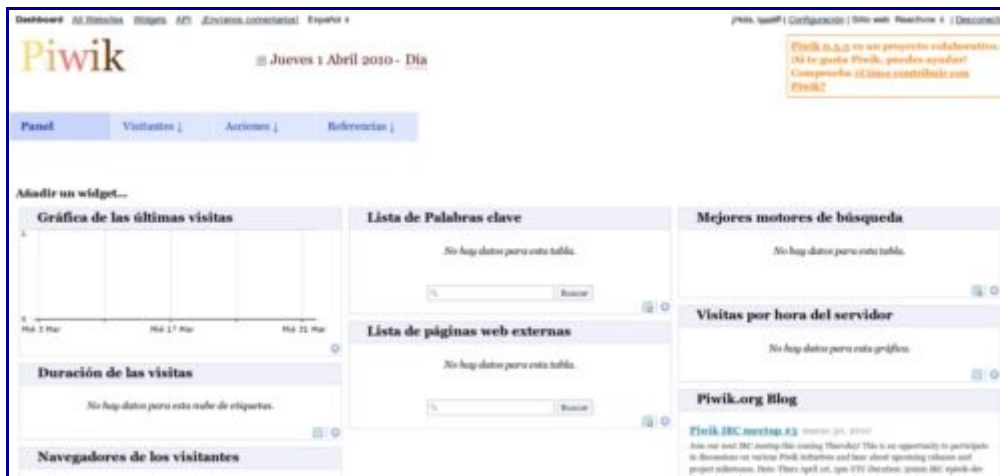




Hemos llegado al final, ahora vamos accedemos a **Piwik** pulsando el enlace ( **Ir a Piwik** ) ...



Introducimos el nombre y usuario y password que creamos durante la instalación y pulsamos sobre el botón ( *Conectar* ) ...



Como podemos ver aparece toda el área personal sin datos y es que ahora hay que tener visitas en la web que acabamos de inscribir y comenzaremos a ver todo tipo de datos.

Puedes agregar usuarios para que puedan ver las estadísticas, activar y desactivar sitios, Plugins desde el botón ( *Configuración* ) que se encuentra situado arriba a la derecha.

Bajo mi punto de vista es totalmente valido frente al gran **Google Analytics** aunque siempre podrás controlar con los dos programas la misma web y comparar sus estadísticas 😊

## Vol 9 – Estadísticas sobre nuestro Hardware con PhpSysInfo



Para **monitorizar** nuestro **Hardware** y desde una web sin duda vuelvo a elegir [phpSysInfo](#). Este proyecto se basa en montar un [servidor web en linux ubuntu server](#) que cumpla todo tipo de funciones como **servidor web**, basándonos en esto además de querer ver todo respecto al tráfico que generarán los sitios web que alojemos también quiero saber que tal está mi servidor en cuanto a sus constantes vitales de [Hardware](#) en plan *consumo de memoria*, *cpu*, espacio en nuestras *particiones*, *carga* del mismo, etc ...

### En el artículo anterior ...

Todo buen administrador es adicto a visitar sus **estadísticas** y ficheros **log** para ver que se cuece dentro de su maquina, esto es algo que visitandolo frecuentemente se aprende mucho sobre el sistema y la reacción del usuario con tus contenidos. Recursos en Internet que monitoreen nuestro **tráfico** podemos encontrar varios pero si lo que queremos es tener el control absoluto y que solo tu puedas ver cosas como de donde vienen nuestros visitantes, navegador, sistema operativo, cuantas visitas tuviste hoy, en esta semana o incluso en el ultimo año son tan solo una pizca de lo que [Piwik](#) puede hacer por nosotros y en nuestro idioma. El no solo es valido para monitorear todo el tráfico entrante y saliente de las webs que tengamos en nuestro servidor y si tenemos mas webs en otros servidores externos también podremos hacerle un seguimiento.

Si quieres saber que se cuece dentro de tu servidor en todo momento desde un entorno web hablabamos de [Piwik](#) para monitorizar el tráfico web y **PhpSysInfo** con el que monitorizaremos el **Hardware** del servidor así a golpe de web.

### Constantes Vitales con PhpSysInfo

Para **monitorizar** nuestro **servidor** en cuanto a su **hardware** he buscado otras opciones y después

de probar otros creo que el mejor es sin duda **PhpSysInfo**. Con el podremos ver estadísticas en cuanto a la carga de nuestro servidor, usuarios conectados, tiempo uptime, versión del sistema, versión de kernel, dirección IP, procesadores, modelo, velocidad de CPU, memoria usada, swap, discos duros, particiones, espacio libre entre otras mas cosas que podemos tener en una sola pagina web construida en **PHP** para que podamos ver que tal se encuentra nuestro [servidor web con Linux Ubuntu Server](http://project2010.sytes.net/phpsysinfo) desde donde queramos.

Para su instalación teclearemos lo siguiente ...

```
~ codigo ~
```

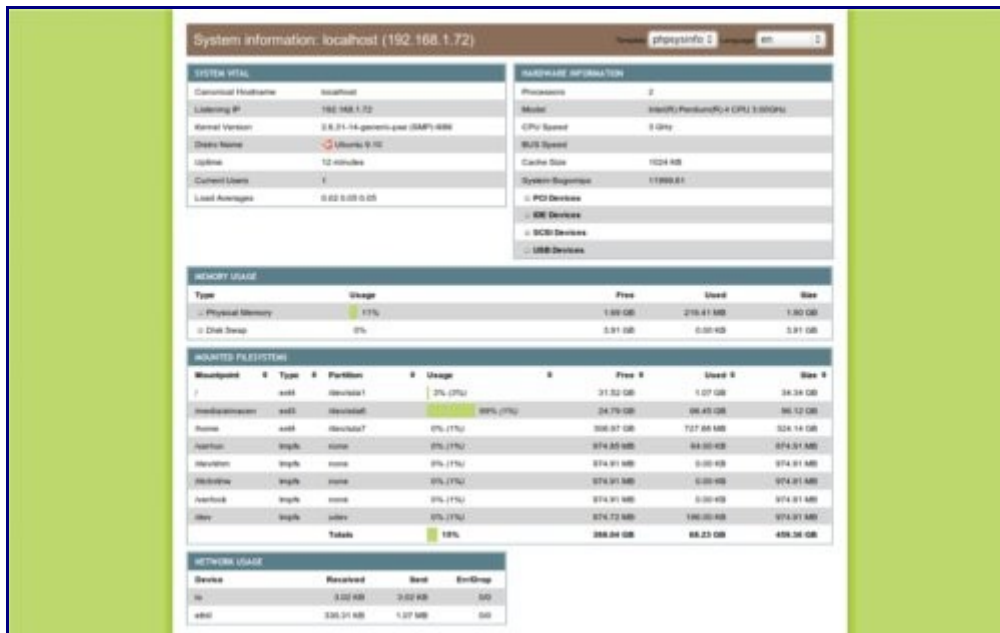
```
sudo apt-get install phpsysinfo
```

Una vez instalado vamos a crear un enlace simbólico dentro de nuestro directorio raíz donde están todas nuestras webs. En mi caso el directorio raíz de mi servidor web se encuentra bajo la ruta /var/www/web substituye esta si no es la tuya ...

```
~ codigo ~
```

```
sudo ln -s /usr/share/phpsysinfo /var/www/web/phpsysinfo
```

Ahora tan solo tendremos que abrir un navegador y visitar la dirección <http://laipdetuserver/phpsysinfo> ...



The screenshot displays the PhpSysInfo web interface. At the top, it shows 'System information: localhost (192.168.1.72)' and a 'phpsysinfo 2' version indicator. The interface is divided into several sections: 'SYSTEM INFO', 'HARDWARE INFORMATION', 'MEMORY USAGE', 'MOUNTED FILESYSTEMS', and 'NETWORK USAGE'. The 'SYSTEM INFO' section lists details like hostname, listening IP, kernel version, distro name, uptime, current users, and load averages. The 'HARDWARE INFORMATION' section lists processor, model, CPU speed, BUS speed, cache size, and system bus speed. The 'MEMORY USAGE' section shows a table with columns for Type, Usage, Free, Used, and Size. The 'MOUNTED FILESYSTEMS' section shows a table with columns for Mountpoint, Type, Partition, Usage, Free, Used, and Size. The 'NETWORK USAGE' section shows a table with columns for Device, Received, Sent, and Error/Group.

| Mountpoint | Type | Partition | Usage    | Free     | Used      | Size      |
|------------|------|-----------|----------|----------|-----------|-----------|
| /          | ext4 | /dev/sda1 | 2% (75G) | 31.52 GB | 1.07 GB   | 32.59 GB  |
| /home      | ext4 | /dev/sda7 | 0% (17G) | 34.79 GB | 96.43 GB  | 131.22 GB |
| /boot      | ext4 | /dev/sda1 | 0% (17G) | 34.79 GB | 96.43 GB  | 131.22 GB |
| /usr       | ext4 | /dev/sda1 | 0% (17G) | 34.79 GB | 96.43 GB  | 131.22 GB |
| /var       | ext4 | /dev/sda1 | 0% (17G) | 34.79 GB | 96.43 GB  | 131.22 GB |
| /tmp       | ext4 | /dev/sda1 | 0% (17G) | 34.79 GB | 96.43 GB  | 131.22 GB |
| /etc       | ext4 | /dev/sda1 | 0% (17G) | 34.79 GB | 96.43 GB  | 131.22 GB |
| Totals     |      |           | 100%     | 66.31 GB | 192.86 GB | 259.17 GB |

Si queremos podemos dejarlo así puesto que funciona perfectamente para verlo tanto desde modo local como desde internet con la dirección que registramos con no-ip, seria ...

<http://project2010.sytes.net/phpsysinfo> . Ahora si somos conformistas nos podemos quedar así pero hay varias cositas que podríamos hacer antes de terminar con este programa.

Podemos configurar mínimamente el idioma, el theme y algunas cosas mas, si quieres configurarte el tuyo y dejarlo a tu estilo edita lo siguiente ...

```
~ codigo ~
```

```
sudo vi /etc/phpsysinfo/config.php
```

Ahora vamos a configurar algunas cosas. Para comenzar buscamos la linea ...

```
define('lang', 'en');
```

Y la modificamos con la sigla de nuestro idioma ...

```
define('lang', 'es');
```

Por otro lado una vez visitemos **PhpSysInfo** en la parte del disco duro podemos ver el sistema de montado de particiones. Si queremos ocultarlo podemos editar esta linea ...

```
define('showMountPoint', true);
```

y modificarla con ...

```
define('showMountPoint', false);
```

Si queréis daros un paseo por este archivo de configuración veréis que podemos hacerle prácticamente de todo aunque yo personalmente creo que esa es mínimamente la configuración que cada uno debería tener.

Salimos y guardamos el archivo de configuración pulsando la tecla ( ESC ) y seguidamente ( :wq! ).

Ahora si hemos instalado [Piwik](#) y **PhpSysInfo** tendremos controlado tanto nuestro trafico como nuestra maquina con un par de clicks.

---

Sois libres de copiar, modificar o incluso usar este manual para lo que queráis incluidos los estudiantes universitarios como proyecto para presentarlo.

Este manual ha sido creado para el aprendizaje de todo el que le interese aprender y puede hacer con el lo que quiera. Tan solo se pide por parte del autor que se le reconozca con un link en tu web

---