

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA DE TELECOMUNICACIÓN
UNIVERSIDAD POLITÉCNICA DE CARTAGENA



Proyecto Fin de Carrera

**Plataforma móvil para Aula Virtual basada en wrt160nl-openwrt-usb
Wrt160nl-openwrt-usb based eLearning Mobile Platform**



AUTOR: D. Grari Hicham

DIRECTOR: Dr. Francesc Burrull i Mestres
Noviembre / 2012



Autor	Grari Hicham
E-mail del Autor	Hicham26_7@hotmail.com
Director(es)	Dr. Francesc Burrull i Mestres
E-mail del Director	francesc.burrull@upct.es
Título del PFC	Plataforma móvil para Aula Virtual basada en wrt160nl-openwrt-USB
Descriptorios	Linksys WRT160NL, OpenWRT, USB
<p>Resumen</p> <p>Como consecuencia del desarrollo de las TIC la educación no presencial crece en importancia día a día a nivel mundial. En concreto, en el entorno universitario la educación a distancia es cada vez más utilizada. Si nos centramos en el caso de la UPCT, disponemos de la plataforma "Aula Virtual", basada en la tecnología Moodle, plataforma de eLearning de código abierto.</p> <p>Por lo que respecta a los países subdesarrollados, básicamente el problema es el acceso a Internet, que suele ser prohibitivo para la gran mayoría de la población.</p> <p>Para este segmento de población existen iniciativas para dotar de ordenadores portátiles con wifi y algunos contenidos educativos, pero el problema reside en que no tienen acceso a internet.</p> <p>El objetivo de éste PFC es que los docentes pueden subir su material a un blog y los alumnos acceder a éste de manera local en institutos o universidades sin conexión a internet.</p>	
Titulación	Ingeniero Técnico Telecomunicación, Especialidad en Telemática
Departamento	Tecnologías de la Información y las Comunicaciones
Fecha de Presentación	

ÍNDICE DE CONTENIDOS

1. Objetivos.....	8
2. Introducción.....	9
3. Router linksys WRT160NL.....	10
3.1 Presentación.....	11
3.2 La potencia de Wireless-N.....	11
3.3 Puerto de Almacenamiento.....	12
3.4 Puerto de Ethernet.....	12
3.5 Especificaciones Técnicas.....	12
3.6 Resumen.....	13
4. OpenWRT.....	14
4.1 Selección del Sistema Operativo del Router.....	15
4.2 Eligiendo el Firmware Adecuado.....	17
4.3 ¿Qué versión de OpenWRT Instalo?.....	17
4.4 Instalación de Openwrt Versión Attitude Adjustment.....	18
4.5 Instalación de OpenWRT mediante Adaptador USB.....	20
4.6 Configuración Básica.....	26
4.7 Uso de Disco Externo USB para Instalar Paquetes y Partición Swap.....	28
4.8 Gestores de Paquetes.....	28
4.8.1 Introducción.....	28
4.8.2 Instalar el Paquete para el Funcionamiento del Pendrive (USB).....	29
4.8.3 Instalación de los Paquetes de Almacenamiento Externo.....	29
4.8.4 Instalar el Paquete de Sntp.....	36
4.9 Habilitando la Memoria Extra (SWAP).....	38
4.10 Configuración de Wireless (Wifi).....	39
5. Servidor Web Lighttpd.....	41
5.1 Instalación y Configuración del Web Server Lighttpd.....	42
5.1.1 Introducción.....	42
5.1.2 Instalación de Lighttpd y sus Módulos sobre OpenWRT.....	42
5.1.3 Configuración Básica del Web Server.....	43
5.1.4 Configurando las Directivas Básicas del Web Server Lighttpd.....	48
5.2 Comprobación del Servicio Web.....	51

6. PHP5.4.5.....	53
6.1 Instalación de PHP 5.4.5.....	54
6.1.1 Introducción a PHP.....	54
6.1.2 Instalación y Configuración de PHP.....	55
7. Base de Datos PostgreSQL.....	66
7.1 Instalación de Bases de Datos.....	67
7.1.1 Introducción.....	67
7.1.2 ¿Qué es PostgreSQL?.....	67
7.1.3 Instalación de PostgreSQL en OpenWRT.....	68
7.2 Configuración de PostgreSQL.....	68
7.2.1 Usuario PostgreSQL.....	69
7.2.2 Preparando la Instalación del Servidor PostgreSQL.....	70
7.2.3 Creación de Base de Datos Junto con un Usuario y Password para Moodle.....	72
7.2.4 Permisos.....	73
8. Virtual E-learning Moodle.....	75
8.1 Entornos de Aprendizajes Virtuales.....	76
8.2 Eligiendo el software adecuado.....	76
8.2.1 Moodle.....	76
8.2.1.1 ¿Por qué se eligió Moodle en este PFC?	76
8.3 Instalación del Entorno de Aprendizaje Virtual Moodle.....	78
8.3.1 Requerimientos.....	78
8.3.2 Descarga e instalación de Moodle.....	79
8.3.2.1 Descargar Moodle.....	79
8.3.2.2 Instalar Moodle.....	80
8.3.3 Creación/registro de usuarios.....	86
8.3.4 Creación de Cursos.....	87
9. Conclusión.....	90
10. Bibliografía.....	91

ÍNDICE DE FIGURAS

Figura 1: Router Linksys WRT160NL.....	11
Figura 2: Logo DD-WRT.....	15
Figura 3: Logo de Openwrt.....	16
Figura 4: 4 firmwares de Openwrt compatibles con wrt160nl.....	17
Figura 5: Pagina web del firmware original.....	18
Figura 6: Adaptador USB – Serie con chipset CP2102.....	19
Figura 7: La placa base del router wrt160nl.....	20
Figura 8: Configuración de pines del puerto serie.....	21
Figura 9: Pantalla de bienvenida de ATTITUDE ADJUSTMENT.....	26
Figura 10: Pantalla de inicio del interfaz web LuCi.....	27
Figura 11: Pantalla de configuración de los parámetros de red local.....	27
Figura 12: Actualizar la lista de software.....	29
Figura 13: Ejecución de comando mount.....	35
Figura 14: Creación de un directorio en el disco externo para almacenar o instalar paquetes.....	35
Figura 15: Captura de pantalla al ejecutar el comando df.....	36
Figura 16: Habilitar el swap en el fichero fstab.....	39
Figura 17: Captura de pantalla al ejecutar el comando free.....	39
Figura 18: Pantalla de configuración de la red Wifi.....	40
Figura 19: Instalación de Lighttpd y sus extensiones.....	42
Figura 20: Instalación de PHP y sus extensiones.....	55
Figura 21: Captura de pantalla al ejecutar el comando php-fcgi.....	64
Figura 22: Parte de la página cargada phpinfo.php.....	65
Figura 23: Instalar el paquete Postgresql-Server.....	68
Figura 24: Lista de base de datos.....	71
Figura 25: Insertar la Base de Datos moodle2 en lista.....	73
Figura 26: Descargar la versión 1.9.19+.....	80
Figura 27: Configuración de idioma.....	81

Figura 28: Comprobación de ajustes de PHP.....	81
Figura 29: Página de condiciones de uso.....	83
Figura 30: Creando tablas para almacenar datos.....	84
Figura 31: Rellenando el formulario del administrador.....	84
Figura 32: Ajustes de portada.....	85
Figura 33: Portada principal de la cuenta del administrador.....	85
Figura 34: Inscribir a un estudiante.....	86
Figura 35: Lista de usuarios inscritos.....	86
Figura 36: Captura de pantalla de cómo crear cursos.....	87
Figura 37: Asignar curso a un alumno.....	88
Figura 38: Agregar Archivos en Sección Temáticos.....	89
Figura 39: Pantalla de login.....	89

Agradecimientos

A mis padres, hermanos por su paciencia y por confiar en mí todo estos años.

A mi mujer Ilham que siempre me apoyo en este largo camino.

A mi director del proyecto, Francesc Burrull i Mestres por su ayuda y predisposición.

A todos vosotros.

1. Objetivos

El objetivo de este proyecto es incorporar en un router móvil una plataforma de enseñanza que permite a la mayoría de los alumnos que pertenecen a países subdesarrollados de recoger toda información y documentación que el docente lo imparta durante el curso mediante el uso de un ordenador con conexión wifi pero sin acceso a internet. Para ello necesitamos integrar en el router wrt160nl varias aplicaciones permitidas para el funcionamiento de dicha aula de enseñanza:

- Una llave de almacenamiento USB, en nuestro caso hemos usado de 4 GB.
- Instalar un sistema operativo de código abierto para poder personaliza el router y ajustarlo a las necesidades del proyecto
- Instalar PHP versión 4.1.0 o mayor y cualquier servidor web ligero.
- Instalar una base de datos para dar soporte a los cursos virtuales
- Instalar una aplicación web (aula virtual) a la que se accede por medio de un navegador web.

2. Introducción

El avance tecnológico ha logrado que la humanidad alcance niveles de desarrollo muy considerables, lo que ha permitido que la educación igualmente se alimente de tecnologías que ayudan a que está llegue muy rápidamente y que el conocimiento sea asimilado a un ritmo muy acelerado; sin embargo en los países subdesarrollados se observa que estas tecnologías no están siendo aplicadas en ciertas áreas o sistemas educativos, el problema suele estar no en disponer de ordenador sino en acceder a la información mediante el acceso a internet que suele ser prohibitivo para la gran mayoría de la población, por esto nos hemos planteado en resolver quizás el problema mediante una plataforma móvil (Moodle) integrada en un router para que el estudiante pueda descargar cursos que el profesor los imparta durante el curso.

Materiales utilizados:

Parte de hardware:

- Router WRT160NL
- Llave de almacenamiento USB de 4 GB.
- Adaptador USB TTL – Serie con chipset CP2102.

Parte de software:

- Sistema Operativo OpenWRT.
- Servidor web Lighttpd y sus módulos
- PHP 5.4.5 y sus módulos.
- Base de datos PostgreSQL.
- Aplicación Web de sistema educativo llamado Moodle

3.Router Linksys WRT160NL

3.1 Presentación



Figura 1: Router Linksys WRT160NL

Después de que algunos de los routers de linksys, como el WRT54GL marcaran un antes y un después por funcionar con una distribución de Linux en su interior se presenta esta nueva gama de routers bajo el nombre de linksys WRT160NL con un aspecto renovado, nuevas características y dicho sistema operativo libre.

Una de las grandes características del linksys WRT160NL junto con el funcionamiento basado en Linux es su conectividad 802.11N que asegura grandes velocidades de transferencia así como su capacidad para hacer las funciones de servidor de contenidos a través de su puerto USB y soporte para DLNA.

Si nos metemos en las entrañas del aparato descubriremos un procesador que funciona a 400 Mhz , 32MB de memoria RAM y una memoria flash 8 MB en la que no cabrá mucho más que el sistema operativo.

3.2 La potencia de Wireless-N

La velocidad y el ancho de banda de Wireless-N, la tecnología inalámbrica actual más avanzada, facilitan que varios usuarios compartan el acceso a Internet, archivos y sistemas de impresión, también actividades multimedia, como juegos, vídeo y llamadas telefónicas de voz sobre IP, todo ello al mismo tiempo, con menos esperas y sin reducir

el rendimiento. Alcanza velocidades más altas si se usa con dispositivos compatibles con Wireless-N, pero es totalmente compatible con equipos Wireless-G y -B.

3.3 Puerto de almacenamiento

La función de puerto de almacenamiento le permite conectar un disco duro o dispositivo de almacenamiento USB tipo flash directamente a la red. De esta forma, podrá aumentar los gigabytes de almacenamiento de forma sencilla. Acceda a sus archivos de datos, vídeo o música desde cualquier PC de la red o de forma segura a través de Internet.

3.4 Puerto de Ethernet

Los cuatro puertos Ethernet 10/100 le permiten conectar impresoras, unidades de almacenamiento de red y otros dispositivos compatibles con Ethernet a la red, para que tenga acceso a ellos y pueda compartirlos en toda la casa.

3.5 Especificaciones técnicas

Modelo:	WRT160NL
Estándares:	802.3, 802.3u, 802.11b, 802.11g, versión 802.11n
Puertos:	Internet, Ethernet [1-4], USB, Alimentación
Botones:	Configuración Wi-Fi protegida, Reinicio
Luces:	LAN [1-4], W-Fi Protected Setup™ (Configuración Wi-Fi protegida), Conexión inalámbrica, Internet, Alimentación
Tipo de cableado:	CAT 5e
Número de antenas:	2
Tipo de conector:	R-SMA

Desmontable (s/n): Sí

Potencia de radiofrecuencia (EIRP) en dBm: Versión 11n: HT20: Habitualmente 17 +/-1,5 dBm a temperatura normal (2 cadenas)

HT40: Habitualmente 15 +/-1,5 dBm a temperatura normal (2 cadenas)

802.11g: Habitualmente: 15 +/- 1,5 dBm a temperatura normal

802.11b: Habitualmente: 19 +/- 1,5 dBm a temperatura normal

Sensibilidad de recepción:

- 802.11n HT40/MCS15 270 Mbps: -67 dBm 10% PER
- 802.11n HT40/MCS0 13,5 Mbps: -75 dBm 10% PER
- 802.11n HT20/MCS15 130 Mbps: -69 dBm 10% PER
- 802.11n HT20/MCS0 6,5 Mbps: -79 dBm 10% PER
- 802.11g 54 Mbps: -74 dBm 10% PER
- 802.11g 6 Mbps: -84 dBm 10% PER
- 802.11b 11 Mbps: -86 dBm 10% PER
- 802.11b 1 Mbps: -92 dBm 10% PER

Ganancia de la antena en dBi: 802.11g: 2,4 GHz <= 1,8 dBi
Versión 11n: 2,4 GHz <= 1,8 dBi

Cert./compat. UPnP: Compatible

Compatibilidad con .Net: No

Funciones de seguridad: WEP, WPA, WPA2, RADIUS, Firewall SPI

Bits de clave de seguridad: Encriptación de hasta 128 bits

Sistemas de archivos compatibles para el dispositivo de almacenamiento: FAT16, FAT32, NTFS

3.6 Resumen

- El enlace de almacenamiento incorporado permite transformar cualquier dispositivo de almacenamiento USB en un sistema de almacenamiento de red (NAS)
- El servidor de medios incorporado permite compartir archivos multimedia con cualquier cliente conectado que tenga activada la función uPnP
- La tecnología MIMO utiliza varios radios para crear una señal potente que alcance una mayor distancia y reduzca los puntos muertos
- Velocidad inalámbrica rápida para aplicaciones que consumen un gran ancho de banda, como transmisión de vídeos o intercambio de archivos
- Compatible con la encriptación de hasta 128 bits
- Sistemas de archivos compatibles (lectura y escritura) para dispositivo de almacenamiento USB: FAT16, FAT32 y NTFS

4. OpenWRT

4.1 Selección del Sistema Operativo del Router

Una vez seleccionado el router que se utilizará, se procederá a seleccionar el SO basado en Linux que le instalaremos al router, en este apartado disponemos de numerosas opciones de las cuales solo nos centraremos en las que dispongan de licencia GNU GENERAL PUBLIC LICENSE y que tengan una comunidad de desarrolladores que garanticen la continuidad del proyecto.

A continuación se pasa a enumerar las distintas opciones que podemos utilizar:

-DD-WRT: Es un firmware libre para diversos routers inalámbricos o Wifi, es muy común observarlo en equipos Linksys Ejecuta un reducido sistema operativo basado en linux. Está licenciado bajo la GNU GENERAL PUBLIC LICENSE.

El firmware es mantenido por Brain Slayer y está alojado en www.dd-wrt.com. Las primeras versiones de DD-WRT se basaron en el firmware Alchemy de Sveasoft Inc., que a su vez estaba basado en el firmware original de Linksys con licencia GPL y en una serie de proyectos de código abierto. DD-WRT fue creado directamente desde el software de Sveasoft, pero la decisión de comenzar a cobrar por su firmware, cerró la puerta a opensource.



Figura 2: Logo DD-WRT

- **OpenWrt:** En lugar de crear un firmware único y estático, OpenWrt proporciona un sistema de archivos totalmente modificable con la gestión de paquetes. Esto le libera de la selección de aplicaciones y la configuración proporcionada por el vendedor y le permite personalizar el dispositivo incorporando aplicaciones a través del uso de paquetes. Para los desarrolladores, OpenWrt es el marco para construir una aplicación sin tener que construir un firmware completo a través de ella. Para los usuarios, esto significa una capacidad de personalización completa, para utilizar un dispositivo embebido en formas que nunca imaginó.



Figura 3: Logo de OpenWRT

Algunas de las características de OpenWrt:

- ❖ **Libre y de código abierto:** El proyecto es gratuito y de código abierto (GPL). El proyecto tiene la meta de estar alojado en un sitio accesible a todo el mundo, con el código fuente al completo para que se pueda desarrollar fácilmente.
- ❖ **Fácil y de libre acceso:** El proyecto está abierto a nuevas contribuciones, cualquier persona puede ser capaz de contribuir con OpenWrt. Los desarrolladores conceden acceso a cualquier persona que lo solicite, ya que piensan que las personas son responsables cuando se les da responsabilidad.
- ❖ **Impulsado por la comunidad:** No se trata de un proyecto en el que los desarrolladores ofrecen algo a los usuarios, se trata de que todos trabajen juntos para lograr una meta común.

OpenWrt ha demostrado durante mucho tiempo que es la mejor solución de firmware de su clase. Es muy superior a otras soluciones en rendimiento, estabilidad, extensibilidad, robustez y diseño. La meta de los desarrolladores es continuar y ampliar OpenWrt para garantizar que es el sistema favorito para el desarrollo y la innovación.

4.2 Eligiendo el Firmware Adecuado

El firmware que utilizaremos será OpenWrt, que está basado en el kernel de Linux, ofrece herramientas de desarrollo y tiene una gran documentación en la web del proyecto mientras el firmware dd-wrt no tiene suficiente documentación que necesitamos para instalar la aula virtual en el router wrt160nl y también tiene una desventaja que suele ocupar un espacio determinado de memoria Rom en el router, por lo que necesitan entre 1,5mb hasta 3mb de almacenamiento.

4.3 ¿Qué Versión de Openwrt Instalo?

Para descargar la versión adecuada de Openwrt, hay que ir al repositorio de la página siguiente <http://downloads.openwrt.org> luego elegir la versión que vamos a instalar en el router WRT160NL, en nuestro caso sería ubicada en el directorio /attitude-adjustmnet/12.09-beta/ar71xx/generic.

Si accedemos a dicho directorio nos encontramos una larga lista de dispositivos soportados, nosotros nos fijamos en estos:



openwrt-ar71xx-generic-wndrmacv2-jffs2-sysupgra..>	04-Sep-2012 19:11	3866628
openwrt-ar71xx-generic-wndrmacv2-squashfs-facto..>	04-Sep-2012 19:11	2883717
openwrt-ar71xx-generic-wndrmacv2-squashfs-svsup..>	04-Sep-2012 19:11	2883588
openwrt-ar71xx-generic-wrt160nl-jffs2-factory.bin	04-Sep-2012 19:11	4260864
openwrt-ar71xx-generic-wrt160nl-jffs2-sysupgrad..>	04-Sep-2012 19:11	4259872
openwrt-ar71xx-generic-wrt160nl-squashfs-factor..>	04-Sep-2012 19:11	3277824
openwrt-ar71xx-generic-wrt160nl-squashfs-sysupg..>	04-Sep-2012 19:11	3276832
openwrt-ar71xx-generic-wrt400n-jffs2-factory.bin	04-Sep-2012 19:11	3834380
openwrt-ar71xx-generic-wrt400n-jffs2-sysupgrade..>	04-Sep-2012 19:11	4194308
openwrt-ar71xx-generic-wrt400n-squashfs-factory..>	04-Sep-2012 19:11	2851340

Figura 4: 4 firmwares de Openwrt compatibles con WRT160NL

Hay que tener cuenta que los 4 firmwares seleccionados en rojo de la figura 4 se diferencian entre sí:

- ❖ **Jffs2-factory:** Son para flashear por primera vez el router desde un firmware diferente del tipo WRT y tienen un sistema de archivos de lectura/escritura.
- ❖ **Jffs2-sysupgrade:** Son para hacer una actualización del firmware si ya estamos en OpenWRT y tienen sistema de archivos de lectura/escritura.

- ❖ **Squashfs-factory:** Son para flashear el router desde un firmware diferente del tipo WRT y tienen un sistema de archivos de solo lectura.
- ❖ **Squashfs-sysupgrade:** Son para hacer una actualización del firmware si ya estamos en OpenWRT y tienen sistema de archivos de SOLO lectura.

En mi caso como quiero modificar muchos parámetros del firmware y no voy a actualizar desde otra versión de Openwrt, instalaré jffs2-factory.

4.4 Instalación de Openwrt Versión Attitude Adjustment

Hay muchas formas de instalar el firmware, se explicará cada método a continuación. Se puede usar cualquier método, al final el resultado será el mismo. Tras instalar el firmware, el dispositivo se reiniciará automáticamente con el nuevo firmware. Si no estamos contentos con OpenWrt, siempre se puede reinstalar el firmware original.

- ✓ **Por medio de la página web del firmware original:** Este es el método más sencillo. Abriendo el navegador web y usando la página de actualización de firmware del dispositivo para instalar el firmware de OpenWrt, pero en este modelo de router WRT160NL se quedó parado es decir se dejó de responder la única manera para instalar el firmware es con vía consola serie, lo explicaré más adelante.

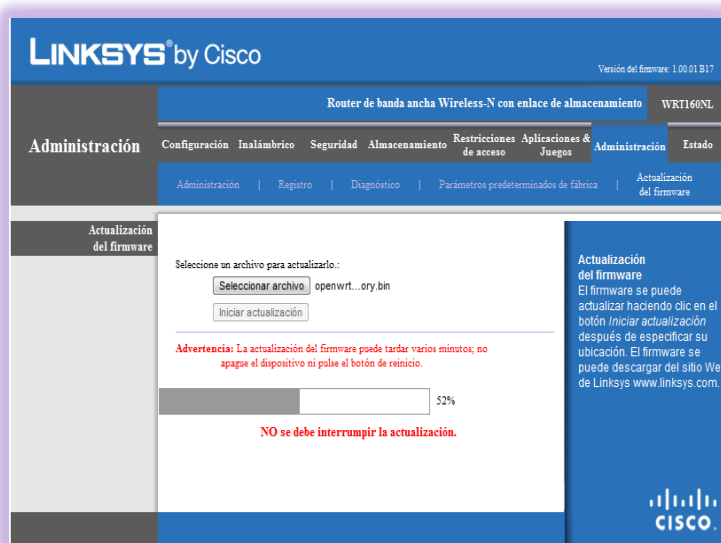


Figura 5: Pagina web del firmware original

- ✓ **A través de TFTP:** Si se hace con mucha prudencia o si estamos intentando reinstalar tras una actualización fallida, se puede usar tftp para instalar el firmware. Este método está explicado a continuación. Los pasos para instalar el firmware por medio de un cliente tftp es prácticamente similar en cualquier sistema operativo.
- ✓ **A través de USB 2 TTL Adapter:** Si ya se tiene un cable de conexión por puerto serie, se puede utilizar para poder cargar de nuevo un firmware funcional y recuperar (unbrick) el router.

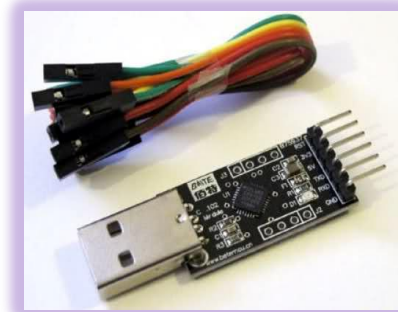


Figura 6: Adaptador USB – Serie con chipset CP2102

- ✓ **A través de JTAG:** No se recomienda instalar la imagen del kernel por medio de JTAG, ya que tardará alrededor de 2 horas, pero es posible instalarlo.

4.5 Instalación de OpenWRT mediante Adaptador USB.

En primer lugar antes de instalar el firmware en el router mediante un adaptador TTL, intenté instalarlo desde el menú de Administración y actualización del firmware de la interfaz web de Linksys que este último no fue bien es decir que dejó el WRT160NL brickeado, entonces recurrí a otro método bastante sencillo que es el puerto de serie para poder cargar de nuevo un firmware funcional y recuperar (unbrick) el router.

Tenemos dos opciones para acceder al puerto serie: abriendo la carcasa y localizando el conector J3 en la placa, o aprovechar los contactos metálicos que podemos encontrar mirando dentro de los conectores hembra RJ-45, puertos WAN y LAN 4. Esta última opción nos permitirá acceso a la consola serie sin tener que abrir el aparato y por tanto sin perder la garantía, si apañamos una manera de usar esos contactos. Por mi parte, he optado por la primera opción por comodidad, ya que los cables de los que dispongo son mucho más fáciles de conectar a los pines del conector interno.



Figura 7: La placa base del router WRT160NL

Así que una vez abierto, el orden y función de cada pin del conector son los siguientes:

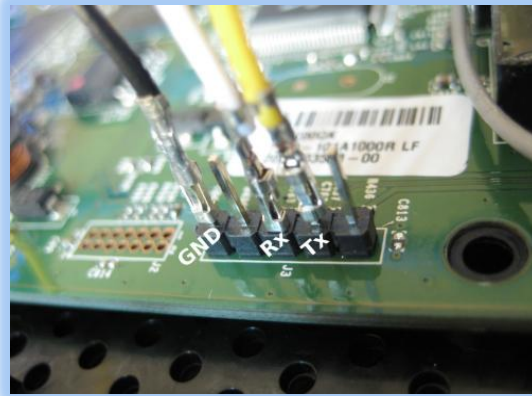


Figura 8: Configuración de pines del puerto serie

Ya tenemos el puerto serie del router perfectamente localizado, ahora solo necesitamos un adaptador USB 2 TTL que usa el chipset CP2102 a puerto serie, la ventaja es que estos adaptadores ya convierten el nivel de la señal, si es necesario. Por lo que he leído, si se usa directamente el puerto serie del PC hay que tener cuidado con esto y convertir la señal a 3.3V que es con la que trabaja el puerto del WRT160NL.

Así que ya tenemos un puerto serie en ambos extremos. Solo necesitamos unos cables que no sean excesivamente largos (yo usé los que me venían con el propio adaptador USB) y conectarlos de la siguiente manera:

- (PC) GND ↔ GND (Router)
- (PC) Rx ↔ Tx (Router)
- (PC) Tx ↔ Rx (Router)

Por último: conecto el PC y el router por Ethernet, usando el puerto LAN1, para más adelante poder subir el firmware por TFTP. Esto es importante: tiene que ser obligatoriamente al puerto número 1, ya que si no la conexión TFTP no se puede realizar, acabará dando un timeout.

Con el WRT160NL apagado, ya tenemos la parte hardware lista. Ahora vamos a necesitar un emulador de terminal y un cliente TFTP, además del firmware que queremos instalar en el router para recuperarlo (por ejemplo, el oficial). Los que yo he usado son PuTTY y TFTP32. En PuTTY se debe crear una sesión de tipo Serial, y las características del puerto son:

- Baud rate =115200
- Data bits 8
- Parity N o None
- Stop bits 1

El puerto COM exacto tenemos que verlo en las propiedades del dispositivo USB en Windows. Ahora daríamos a Open para abrir la sesión, y encenderíamos el router. Si todo está correcto deberíamos ver una serie de mensajes y, en un momento dado, nos pedirá que pulsemos una tecla para detener el arranque del sistema y entrar en la consola del cargador U-Boot. ¡Ojo! El tiempo que nos da para ello es de solamente un segundo, así que tenemos que estar atento.

Cuando realizamos este paso obtenemos en la ventana del Putty un montón de caracteres extraños en vez del texto que nuestro debajo, luego revisamos bien este par de cosas:

- Lo primero es asegurarse de que la conexión entre el puerto serie en la placa base, y el adaptador USB es correcta según. Comprobamos también que no hay contacto entre ninguno de los pines con los adyacentes, etc.
- En segundo lugar, es frecuente que se produzcan interferencias entre el cable Ethernet que conecta al PC con el router, y los cables que van del puerto serie al adaptador USB. Entonces debemos alejar ambos lo máximo posible y asegurarse de que no se cruzan o están cerca. Posiblemente.

Teniendo en cuenta estos dos puntos y si todo va bien, deberíamos ver en el Putty algo como esto:

```
U-Boot 1.1.6 (sep 14 2012 - 14:02:36)

DRAM:  ar7100_dds_initial_config(237) enter!
ar7100_dds_initial_config(269) exit!

U-Boot 1.1.6 (sep 14 2012 - 14:02:36)

AP81 (ar7100) U-boot
sri
32 MB
WRT160NL u-boot version: 1.0.0
Top of RAM usable for U-Boot at: 82000000
Reserving 279k for U-Boot at: 81fb8000
Reserving 192k for malloc() at: 81f88000
Reserving 44 Bytes for Board Info at: 81f87fd4
Reserving 36 Bytes for Global Data at: 81f87fb0
Reserving 128k for boot params() at: 81f67fb0
Stack Pointer at: 81f67f98
Now running in RAM - U-Boot at: 81fb8000
id read 0x100000ff
flash size 8MB, sector count = 128
Flash:  8 MB
*** Warning - bad CRC, using default environment

In:     serial
Out:    serial
Err:    serial
Net:    ag7100_enet_initialize...
ag7100 get ethaddr for device eth0
Fetching MAC Address from 0x81feble0

-----***** Get the RTL8306SD Manafactory ID=379c *****-----
Reg6: speed=0 nway=1 duplex=0
Reg5: speed=0 nway=0 duplex=0
Reg1: a1=7fd9 a2=30e0 a3=15ac a4=30e0 a5=0
Reg1: a1=7fd9 a2=30e0 a3=15ac a4=30e0
Reg1: a1=7fd9 a2=30e0 a3=15ac a4=30e0
Reg1: a1=7fd9 a2=30e0 a3=15ac a4=30e0
Reg1: a1=7fd9 a2=30e0 a3=15ac a4=30e0
eth0: c0:c1:c0:78:f7:56
eth0 up
eth0
### main_loop entered: bootdelay=1

Hit any key to stop autoboot:  0

ar7100>
```

Ahora podemos empezar a teclear comandos: empezando por help, nos mostrará la lista de los disponibles. Uno de ellos es el comando upgrade que nos permite cargar el firmware que hemos descargado en el router. Ejecutamos:

```
ar7100> upgrade code.bin
```

En origen, el fichero puede tener cualquier nombre, pero en el router ha de llamarse code.bin. Si el cliente no nos da la opción de especificar el nombre de destino, lo renombramos primero en el PC antes de subirlo. Este comando abrirá un servidor TFTP, y será en este momento cuando podremos cargar el fichero BIN a través del programa TFTP32 o similar.

La salida que genera el comando upgrade se muestra a continuación. Los mensajes de checksum bad no tienen muy buena pinta, pero... el comando funcionó. Se queda esperando por el archivo justo después del último de estos mensajes, y una vez terminada la transferencia TFTP, continúa automáticamente con el flasheo.

```
ar7100> upgrade code.bin
  check link duplex:Full/speed:100
Tftpd start listening on port[69]!
Load address: 0x80060000
checksum bad
checksum bad
checksum bad
checksum bad
checksum bad
checksum bad
checksum bad
checksum bad
checksum bad
checksum bad
checksum bad
checksum bad
checksum bad
checksum bad
Receiving firmware

[/code] from [169.254.90.160] Write File : CODE.BIN # Current Code
Pattern:NL16 , Upgrade Code Pattern:NL16 Code Pattern is correct!
#####
#####
#####
#####
#####
#####
#####
#####
```


4.6 Configuración Básica

Recién instalado el firmware por puerto serie, el router tiene como IP local 192.168.1.1. Desde el PC con el que estemos conectados vía Ethernet al router, lo primero es abrir una sesión telnet para establecer la contraseña de root, lo que nos ofrece de forma automática. Una vez lo hayamos hecho, OpenWRT desactivará el acceso por telnet y activará el SSH.

```
Using username "root".
root@192.168.1.2's password:

BusyBox v1.19.4 (2012-08-26 12:49:54 UTC) built-in shell (ash)
Enter 'help' for a list of built-in commands.

|-----| .-----|-----| | | | .-----| | | | | |
| - | | _ | - | | | | | | _ | |
|-----| | _ | | | | | | | | | |
|_| W I R E L E S S F R E E D O M

-----
ATTITUDE ADJUSTMENT (12.09-beta, r33312)
-----
* 1/4 oz Vodka      Pour all ingredients into mixing
* 1/4 oz Gin        tin with ice, strain into glass.
* 1/4 oz Amaretto
* 1/4 oz Triple sec
* 1/4 oz Peach schnapps
* 1/4 oz Sour mix
* 1 splash Cranberry juice
-----
root@OpenWrt:~# █
```

Figura 9: pantalla de bienvenida de ATTITUDE ADJUSTMENT

A continuación, podemos seguir por la configuración de red. En mi caso, el WRT160NL no gestiona el acceso a Internet, eso lo hace el equipo que me suministró mi proveedor y que es el que tiene asignada la IP 192.168.1.1. Así que antes de conectar el Linksys a mi ordenador, le cambio su IP local a 192.168.1.2 (por ejemplo) a través del interfaz web LuCi incluido en el firmware (también podríamos hacerlo por línea de comandos).

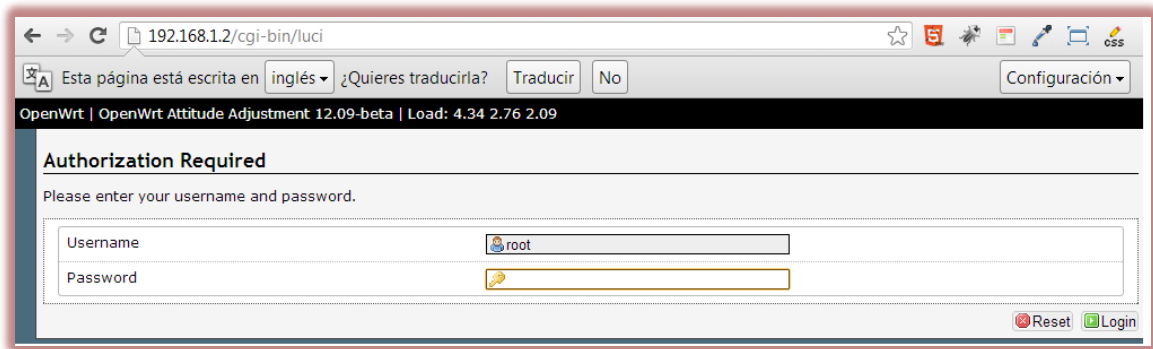


Figura 10: Pantalla de inicio del interfaz web LuCi

Una vez hecho login con la contraseña establecida anteriormente, pinchamos en Administration para pasar a modo avanzado y tener todas las opciones de menú disponibles. Después, en el menú **Network > Interfaces > LAN** podemos configurar los parámetros de interfaz de red local del router, entre ellas la IP local, la IP de la pasarela de acceso (192.168.1.1) y los servidores DNS.

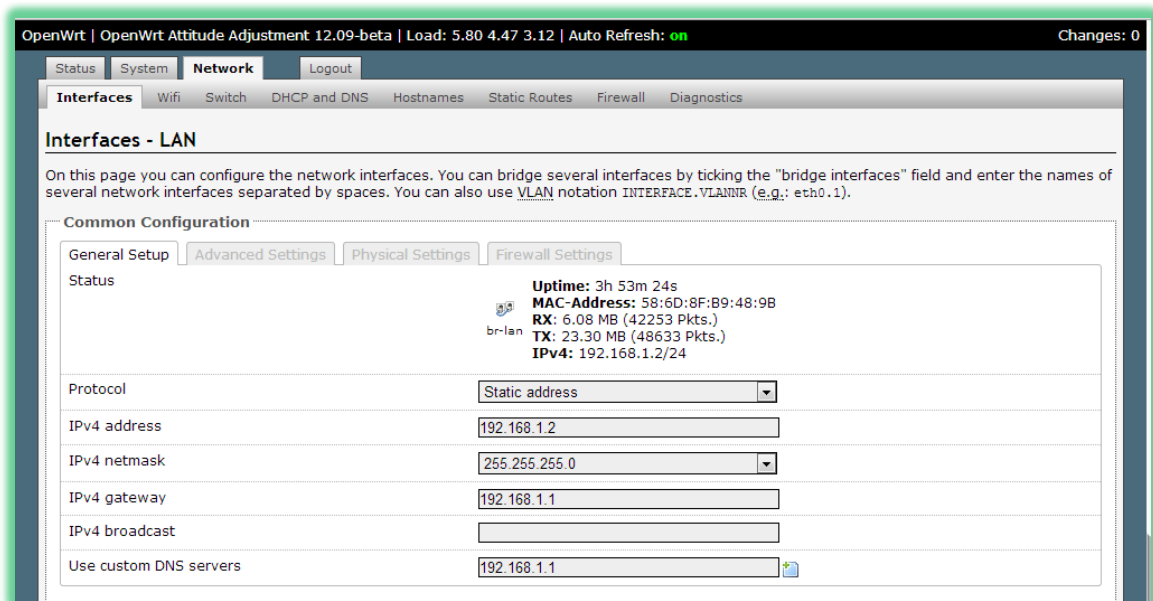


Figura 11: Pantalla de configuración de los parámetros de red local

Una vez hecho esto y guardados los cambios ya podemos reiniciar el router. Accedemos por SSH al router, con el usuario root y la contraseña que definimos anteriormente (si estamos en Windows podemos usar PuTTY como cliente de consola).

Escribiendo: **ssh root@192.168.1.2**

4.7 Uso de Disco Externo USB para Instalar Paquetes y Partición Swap

Llegados a este punto de tener casi la configuración más básica realizada, vamos a conectarle al puerto USB de nuestro router WRT160NL un disco USB para instalar software (paquetes) desde la página oficial de OpenWRT y luego poder subir aplicaciones (en nuestro caso “Moodle”).

Lo primero sería particionar la llave de almacenamiento desde Linux o bien con un programa específico desde Windows antes de empezar a instalar paquetes. El disco USB se divide en dos particiones una para almacenar datos y otra para dar mejor rendimiento al router, la primera partición de 3.5 GB la reservé al sistema de archivo EXT4 (he elegido este tipo de partición por la velocidad de acceso a los datos, pues EXT4 es superior a cualquier otra) y finalmente una partición swap de 546 MB.

La partición nos deberá quedar:

```
/dev/sda1    EXT4-----> 3453530 KB  
/dev/sda2    SWAP -----> 546204 KB
```

Una vez tenemos particionado nuestro USB, es hora de instalar una serie de paquetes para que el sistema reconozca los distintos tipos de particiones.

4.8 Gestores de Paquetes

4.8.1 Introducción

La utilidad `opkg` es un gestor de paquetes ligero usado para descargar e instalar paquetes de OpenWrt a través de internet. (A los usuarios de GNU/Linux les parecerá similar al comando `apt-get`). Este se puede administrar por línea de comandos o interface web.

Se pueden encontrar más opciones a través del comando `opkg --help`

El archivo de configuración de `opkg` es: `/etc/opkg.conf`.

4.8.2 Instalar el Paquete para el Funcionamiento del Pendrive (USB)

Una vez que OpenWRT utiliza un gestor de paquetes llamado `opkg`, cuyo uso es muy sencillo. Lo primero que debemos hacer es actualizar la lista de software disponible.

```
root@OpenWrt:~# opkg update
Downloading http://downloads.openwrt.org/attitude_adjustment/12.09-beta/ar71xx/generic/packages/Packages.gz.
Inflating http://downloads.openwrt.org/attitude_adjustment/12.09-beta/ar71xx/generic/packages/Packages.gz.
Updated list of available packages in /var/opkg-lists/attitude_adjustment.
root@OpenWrt:~# █
```

Figura 12: Actualizar la lista de software

Esto hay que hacerlo cada vez que hayamos reiniciado el router, ya que no se mantiene en memoria, supongo que por temas de ahorro de almacenamiento.

Ahora podemos comenzar a instalar los paquetes necesarios para el soporte USB aunque en la última versión que hemos instalado (ATTITUDE ADJUSTEMENT VERSION 12.09 BETA), lleva integrado el paquete para dar soporte al pendrive, sin embargo para lo que tienen la versión inferior a la nuestra, pueden usar el siguiente comando:

`opkg install kmod-usb2`

Que instalará el paquete `kmod-usb2` y las dependencias necesarias que conforman el software mínimo para poder utilizar el puerto USB del WRT160NL.

4.8.3 Instalación de los paquetes de almacenamiento externo

Una vez tenemos el puerto USB listo, los paquetes a instalar para soporte del pendrive es:

`opkg install kmod-usb-storage block-mount`

Hecho esto, ahora necesitamos el driver correcto para el tipo de sistema de archivos que tenga el disco que queremos compartir desde nuestro router. Lo ideal sería tener un

filesystem Linux como ext3 o ext4: más fiables y robustos y con soporte nativo y completo.

Hay que tener en cuenta que el pendrive ha de estar formateado en ext2 o superior, y en la línea device poner el identificador de unidad que corresponda en cada sistema. Para incorporar el soporte para filesystem tipo EXT:

opkg install kmod-fs-ext4 e2fsprogs

Una vez que tengamos instalados todos los paquetes necesarios para acceder al disco USB, vamos a comprobar que efectivamente se detectan las unidades externas examinando la salida del comando **dmesg**. Debería haber una sección similar a esta:

```
root@OpenWrt:~# dmesg
```

```
[ 0.000000] Linux version 3.3.8 (blogic@Debian-60-squeeze-64-minimal) (gcc version 4.6.3 20120201 (prerelease) (Linaro GCC 4.6-2012.02) ) #2 Mon Sep 3 17:51:18 UTC 2012
[ 0.000000] MyLoader: syp=aaaa5554, boardp=aaaa5554, parts=aaaa5554
[ 0.000000] bootconsole [early0] enabled
[ 0.000000] CPU revision is: 00019374 (MIPS 24Kc)
[ 0.000000] SoC: Atheros AR9130 rev 2
[ 0.000000] Clocks: CPU:400.000MHz, DDR:400.000MHz, AHB:200.000MHz, Ref:5.000MHz
[ 0.630000] IP route cache hash table entries: 1024 (order: 0, 4096 bytes)
[ 0.630000] TCP established hash table entries: 1024 (order: 1, 8192 bytes)
[ 0.640000] TCP bind hash table entries: 1024 (order: 0, 4096 bytes)
[ 0.640000] TCP: Hash tables configured (established 1024 bind 1024)
[ 0.650000] TCP reno registered
[ 0.650000] UDP hash table entries: 256 (order: 0, 4096 bytes)
[ 0.660000] UDP-Lite hash table entries: 256 (order: 0, 4096 bytes)
[ 0.660000] NET: Registered protocol family 1
[ 0.670000] PCI: CLS 0 bytes, default 32
[ 0.690000] squashfs: version 4.0 (2009/01/31) Phillip Lougher
[ 0.690000] JFFS2 version 2.2 (NAND) (SUMMARY) (LZMA) (RTIME) (CMODE_PRIORITY) (c) 2001-2006 Red Hat, Inc.
```

```

[ 0.720000] Serial: 8250/16550 driver, 16 ports, IRQ sharing enabled

[ 0.760000] serial8250.0: ttyS0 at MMIO 0x18020000 (irq = 11) is a 16550A

[ 1.200000] ag71xx ag71xx.0: eth0: connected to PHY at ag71xx-mdio.0:00 [uid=
00008306, driver=Realtek RTL8306S]

[ 1.210000] eth1: Atheros AG71xx at 0xba000000, irq 5

[ 1.510000] ag71xx ag71xx.1: eth1: connected to PHY at ag71xx-mdio.0:04 [uid=
00008306, driver=Realtek RTL8306S]

[ 1.520000] TCP cubic registered

5.200000] eth0: link up (100Mbps/Full duplex)

[ 6.780000] JFFS2 notice: (465) jffs2_build_xattr_subsystem: complete buildin
g xattr subsystem, 1 of xdatum (0 unchecked, 0 orphan) and 12 of xref (0 dead, 3
orphan) found.

[ 7.570000] usbcore: registered new interface driver usbfs

[ 7.580000] usbcore: registered new interface driver hub

[ 7.590000] usbcore: registered new device driver usb

[ 8.130000] ehci_hcd: USB 2.0 'Enhanced' Host Controller (EHCI) Driver

[ 8.130000] ehci-platform ehci-platform: Generic Platform EHCI Controller

[ 8.140000] ehci-platform ehci-platform: new USB bus registered, assigned bus
number 1

[ 8.180000] ehci-platform ehci-platform: irq 3, io mem 0x1b000000

[ 8.200000] ehci-platform ehci-platform: USB 2.0 started, EHCI 1.00

[ 8.200000] hub 1-0:1.0: USB hub found // -----> un Puerto USB encontrado

[ 8.210000] hub 1-0:1.0: 1 port detected

[ 8.220000] ohci_hcd: USB 1.1 'Open' Host Controller (OHCI) Driver

[ 8.280000] Initializing USB Mass Storage driver...-----> los drivers de initialization del
almacenamiento de USB están instalados

[ 8.290000] usbcore: registered new interface driver usb-storage

[ 8.290000] USB Mass Storage support registered.

[ 8.530000] usb 1-1: new high-speed USB device number 2 using ehci-platform

[ 8.770000] scsi0 : usb-storage 1-1:1.0

[ 9.780000] scsi 0:0:0:0: Direct-Access Generic Flash Disk 8.07 PQ :
0 ANSI: 2

[ 9.790000] sd 0:0:0:0: [sda] 8216576 512-byte logical blocks: (4.20 GB/3.91
GiB)

```

```

[ 9.800000] sd 0:0:0:0: [sda] Write Protect is off
[ 9.800000] sd 0:0:0:0: [sda] Mode Sense: 03 00 00 00
[ 9.810000] sd 0:0:0:0: [sda] No Caching mode page present
[ 9.810000] sd 0:0:0:0: [sda] Assuming drive cache: write through
[ 9.820000] sd 0:0:0:0: [sda] No Caching mode page present
[ 9.830000] sd 0:0:0:0: [sda] Assuming drive cache: write through

[ 10.240000] sda: sda1 sda2 -----> dos particiones: uno para almacenar datos en ext4 y otro
para el swap

[ 10.240000] sd 0:0:0:0: [sda] No Caching mode page present
[ 10.250000] sd 0:0:0:0: [sda] Assuming drive cache: write through
[ 10.260000] sd 0:0:0:0: [sda] Attached SCSI removable disk

[ 28.790000] eth0: link down

[ 30.220000] Compat-wireless backport release: compat-wireless-2012-07-13
[ 30.230000] Backport based on wireless-testing.git master-2012-07-16
[ 30.230000] compat.git: wireless-testing.git
[ 30.260000] cfg80211: Calling CRDA to update world regulatory domain
[ 30.270000] cfg80211: World regulatory domain updated:

[ 30.270000] cfg80211: (start_freq - end_freq @ bandwidth), (max_antenna_gai
n, max_eirp)

[ 30.880000] ath: EEPROM regdomain: 0x0
[ 30.880000] ath: EEPROM indicates default country code should be used
[ 30.880000] ath: doing EEPROM country->regdmn map search
[ 30.880000] ath: country maps to regdmn code: 0x3a
[ 30.880000] ath: Country alpha2 being used: US
[ 30.880000] ath: Regpair used: 0x3a
[ 30.880000] ieee80211 phy0: Selected rate control algorithm 'minstrel_ht'
[ 30.880000] ieee80211 phy0: Atheros AR9100 MAC/BB Rev:7 AR2122 RF Rev:a2 mem=
0xb80c0000, irq=2

[ 36.360000] eth0: link up (100Mbps/Full duplex)

```



```
[ 36.390000] br-lan: port 1(eth0) entered forwarding state
[ 36.390000] br-lan: port 1(eth0) entered forwarding state
[ 37.490000] device wlan0 entered promiscuous mode
[ 38.050000] br-lan: port 2(wlan0) entered forwarding state
[ 38.060000] br-lan: port 2(wlan0) entered forwarding state
[ 38.390000] br-lan: port 1(eth0) entered forwarding state
[ 38.550000] ar71xx: pll_reg 0xb8050018: 0x13000a44
[ 38.550000] eth1: link up (100Mbps/Full duplex)
[ 40.060000] br-lan: port 2(wlan0) entered forwarding state
[ 49.750000] EXT4-fs (sda1): warning: maximal mount count reached, running e2fsck is
recommended
[ 49.760000] EXT4-fs (sda1): recovery complete
[ 49.760000] EXT4-fs (sda1): mounted filesystem with ordered data mode. Opts:
(null)
[ 50.170000] Adding 546204k swap on /dev/sda2. Priority:-1 extents:1 across:5
46204k-----> 546204 KB de swap (memoria ram) en sd2
root@OpenWrt:~#
```

Ya tenemos todos los datos necesarios para montar el disco externo USB como una unidad más accesible en nuestro router. Comenzamos creando un punto de montaje, por ejemplo en /mnt/hicham

mkdir -p /mnt/hicham

El siguiente paso es crear una entrada para nuestra unidad USB en el fichero de configuración /etc/config/fstab. Pero antes debemos comprobar que no exista ya creada por defecto una entrada de ejemplo, como ocurría en mi caso, que además utilizaba el mismo identificador de dispositivo /dev/sda1 que el correspondiente a mi unidad. Comprueba el fichero fstab

vi /etc/config/fstab

Y si existe alguna entrada config 'mount' ya creada, la eliminamos editando el fichero /etc/config/fstab o bien mediante la siguiente secuencia:

```
/etc/init.d/fstab stop
uci delete fstab.@mount[0]
uci commit fstab
/etc/init.d/fstab start
```

Ahora ya podemos configurar nuestro disco USB usando UCI, sería una secuencia de comandos como esta:

```
/etc/init.d/fstab stop
uci add fstab mount
uci set fstab.@mount[-1].device=/dev/sda1
uci set fstab.@mount[-1].options=rw,sync
uci set fstab.@mount[-1].enabled_fsck=0
uci set fstab.@mount[-1].enabled=1
uci set fstab.@mount[-1].target=/mnt/hicham
uci commit fstab
/etc/init.d/fstab start
```

¿Qué es UCI?

UCI es el Interfaz de Configuración Unificada (Unified Configuration Interface en el idioma de Shakespeare), una utilidad de línea de comando pensada para centralizar y facilitar las tareas de configuración del sistema. De todas formas, si no queremos usar la interfaz UCI podemos editar directamente dentro de los ficheros de configuración.

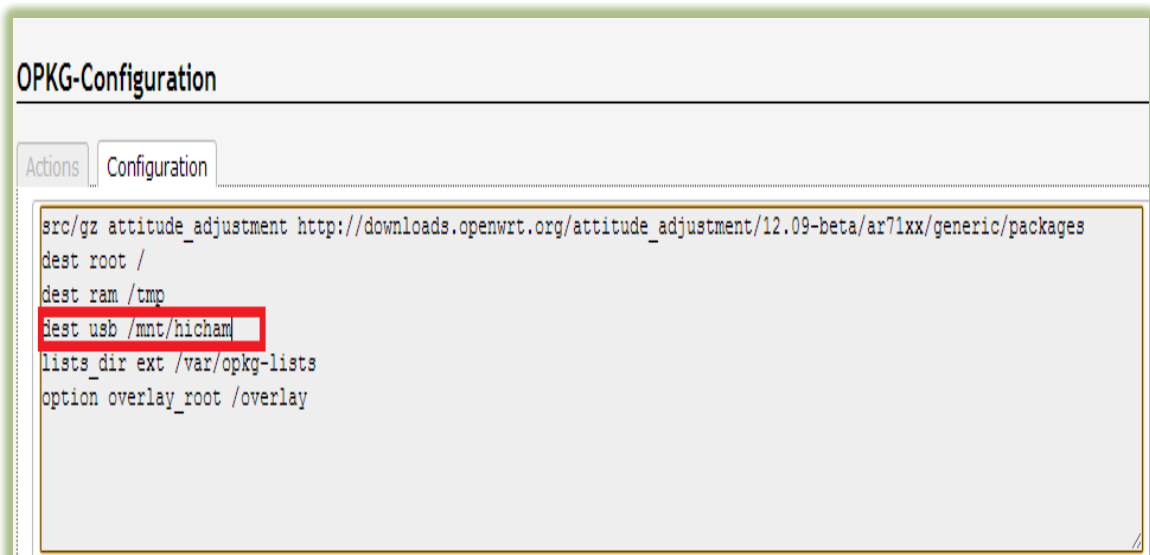
Para comprobar si todo ha ido bien ejecutamos el comando: **mount**

```
root@OpenWrt:~#  
root@OpenWrt:~# mount  
rootfs on / type rootfs (rw)  
/dev/root on /rom type squashfs (ro,relatime)  
proc on /proc type proc (rw,noatime)  
sysfs on /sys type sysfs (rw,noatime)  
tmpfs on /tmp type tmpfs (rw,nosuid,nodev,noatime)  
tmpfs on /dev type tmpfs (rw,noatime,size=512k,mode=755)  
devpts on /dev/pts type devpts (rw,noatime,mode=600)  
/dev/mtdblock3 on /overlay type jffs2 (rw,noatime)  
overlayfs:/overlay on / type overlayfs (rw,relatime,lowerdir=/,upperdir=/overlay)  
debugfs on /sys/kernel/debug type debugfs (rw,relatime)  
none on /proc/bus/usb type usbfs (rw,relatime)  
/dev/sda1 on /mnt/hicham type ext4 (rw,sync,relatime,user_xattr,barrier=1,data=0)  
root@OpenWrt:~# █
```

Figura 13: Ejecución de comando mount

Después de verificar con el comando mount que la unidad USB está montada, podemos instalar cualquier paquete en dicha unidad de destino. Solo tenemos que especificar una línea de configuración para cada posible unidad en el archivo /etc/opkg.conf similar a esta:

dest usb /mnt/hicham



```
OPKG-Configuration  
Actions Configuration  
src/gz attitude_adjustment http://downloads.openwrt.org/attitude_adjustment/12.09-beta/ar71xx/generic/packages  
dest root /  
dest ram /tmp  
dest usb /mnt/hicham  
lists_dir ext /var/opkg-lists  
option overlay_root /overlay
```

Figura 14: creación de un directorio en el disco externo para almacenar o instalar paquetes

Donde USB es el alias que le damos al destino, y que utilizaremos luego como parámetro del comando opkg. Y /mnt/hicham es el punto de montaje que acabamos de configurar para el pendrive.

Creamos el siguiente enlace:

```
ln -s /mnt/hicham /opt
```

Y editamos el fichero /etc/profile para añadir las siguientes dos líneas al final:

- ✓ Export PATH=\$PATH:/opt/bin:/opt/sbin:/opt/usr/bin:/opt/usr/sbin
- ✓ Export LD_LIBRARY_PATH=\$LD_LIBRARY_PATH:/opt/lib:/opt/usr/lib

Ahora, cuando queramos añadir uno o varios paquetes nuevos, podemos especificar que se instalen en el pendrive mediante el parámetro -d, por ejemplo:

```
opkg -d usb install php5
```

Por último si queremos saber al espacio disponible tanto en el router como en el USB ejecutamos el siguiente comando: df o bien df -h

```
root@OpenWrt:~# df -h
Filesystem      Size      Used Available Use% Mounted on
rootfs          4.5M      2.5M      2.0M   55% /
/dev/root        2.0M      2.0M        0 100% /rom
tmpfs           14.3M     148.0K     14.2M    1% /tmp
tmpfs            512.0K        0     512.0K    0% /dev
/dev/mtdblock3  4.5M      2.5M      2.0M   55% /overlay
overlayfs:/overlay 4.5M      2.5M      2.0M   55% /
/dev/sda1       3.3G     985.0M     2.2G   31% /mnt/hicham
```

Figura 15: captura de pantalla al ejecutar el comando df

4.8.4 instalar el paquete Sntp

Ssmtp es una pequeña herramienta que permite al profesor enviar emails desde Moodle a las personas matriculadas a sus asignaturas (por ejemplo mandarles la contraseña de acceso a Moodle, información del curso, la fecha de entrega de prácticas....etc).

Ahora podemos comenzar a instalar el paquete necesario para el envío de correos desde cualquier software instalado en el servidor OpenWRT

```
opkg update
```

```
opkg -d usb install smtptrap
```

Y luego modificamos el archivo ssmtp.conf

```
# /etc/ssmtp.conf -- a config file for sSMTP sendmail.
#
# The person who gets all mail for userids < 1000
# Make this empty to disable rewriting.
root=hicham26_7@hotmail.com
# The place where the mail goes. The actual machine name is required
# no MX records are consulted. Commonly mailhosts are named
mail.domain.com
# The example will fit if you are in domain.com and your mailhub is so
named.
mailhub=mail
# Example for SMTP port number 2525
mailhub=smtp.hotmail.com:2525
# Example for SMTP port number 25 (Standard/RFC)
# mailhub=mail.your.domain
# Example for SSL encrypted connection
mailhub=smtp.hotmail.com:465
# Where will the mail seem to come from?
rewriteDomain=hotmail.com
# The full hostname
hostname=hotmail.com
# Set this to never rewrite the "From:" line (unless not given) and to
# use that address in the "from line" of the envelope.
FromLineOverride=YES
# Use SSL/TLS to send secure messages to server.
UseTLS=YES
# Use SSL/TLS certificate to authenticate against smtp host.
#UseTLSCert=YES
# Use this RSA certificate.
```

```
#TLSCert=/etc/ssl/certs/ssmtp.pem

# Get enhanced (*really* enhanced) debugging information in the logs

# If you want to have debugging of the config file parsing, move this
option

# to the top of the config file and uncomment

#Debug=YES
```

4.9 Habilitando la Memoria Extra (SWAP)

El espacio Swap, o de "intercambio", es lo que se conoce como memoria virtual. La diferencia entre la memoria del router WRT160NL (32MB) y la de virtual USB (546MB) es que está última utiliza espacio en el USB en lugar de un módulo de memoria. Cuando la memoria de router se agota, el sistema copia parte del contenido de esta directamente en este espacio Swap a fin de poder realizar otras tareas.

A continuación habilitamos el swap entrando en el fichero de configuración `/etc/config/fstab` mediante un editor de texto vi o bien con el comando `uci`, una vez dentro del fichero cambiamos el '0' por '1' en la option de swap de `config swap`, luego guardamos y tras un reinicio con el comando `/etc/init.d/fstab restart` se habilita la memoria swap

```
config global automount
    option from_fstab 1
    option anon_mount 1

config global autoswap
    option from_fstab 1
    option anon_swap 0

config mount
    option target /mnt/hicham
    option device /dev/sda1
    option fstype ext4
    option options rw, sync
    option enabled 1
    option enabled_fsck 0

config swap
    option device /dev/sda2
    option enabled 1
```

→ se cambia de '0' a '1' para que se habilite el swap

Figura 16: Habilitar el Swap en el fichero fstab

De esta manera quedaría habilitada la memoria SWAP, usando el comando free.

```
root@OpenWrt:~# free
      total        used         free       shared    buffers
Mem:    29344       25736         3608           0        1596
-/+ buffers:
Swap:   546204       19268       526936
```

Figura 17: Captura de pantalla al ejecutar el comando free

4.10 Configuración de Wireless (Wifi)

Para detectar Wifi en el hardware WRT160NL configuramos desde el terminal root el siguiente comando:

```
wifi detect > /etc/config/wireless
```

Donde el comando wifi detect genera un fichero de configuración del sistema Wifi a partir de las propiedades de nuestro hardware.

Ahora podemos configurar los parámetros de nuestra red Wifi a través de la interfaz web Luci, entrando en **Network >Wifi**

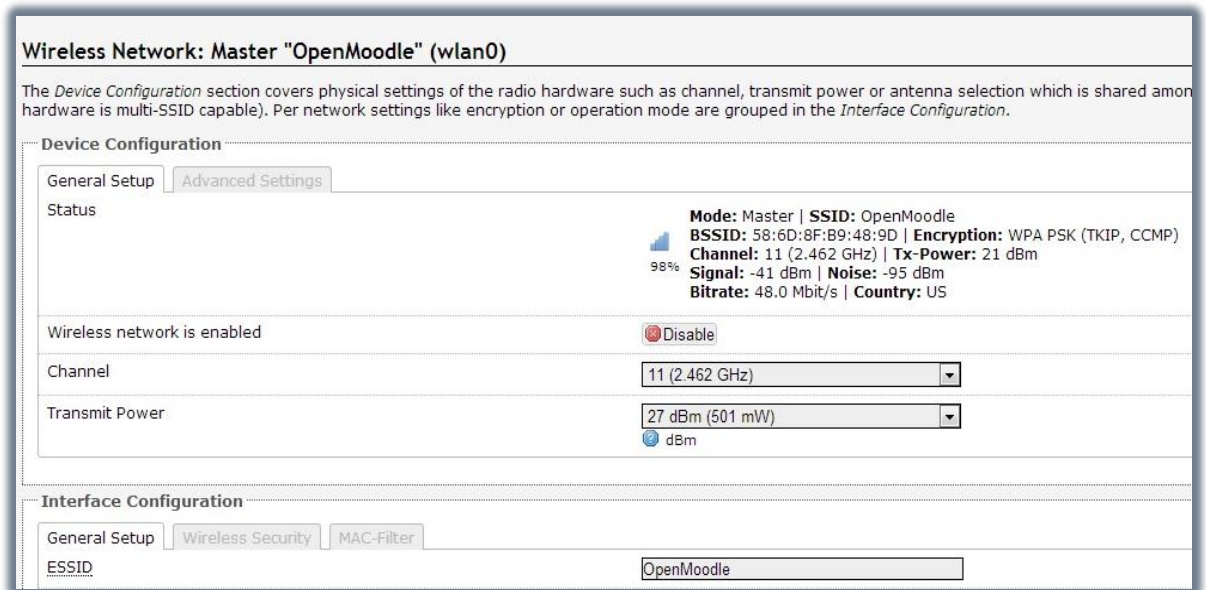


Figura 18: Pantalla de configuración de la red Wifi

En la sección de configuración de interfaz, he puesto un nombre a la red (OpenMoodle) en la casilla de ESSID, Luego activé el modo de encriptación a WPA-PSK e introduje la contraseña correspondiente. El resto de valores lo que viene por defecto. Después de guardar los ajustes podemos conectarnos al router a través del Wifi.

5. Servidor web

Lighttpd

5.1 Instalación y configuración del web server lighttpd

5.1.1 Introducción

Lighttpd es un servidor web para los sistemas operativos Unix, Linux y Microsoft Windows. La instalación de este servidor también conocido como Lighty, es una alternativa muy flexible para el servidor de páginas web Apache o Internet Information Services.

Este web server diseñado para ser seguro, rápido (muy rápido a decir verdad), compatible con los estándares y flexible a la vez esta optimizado para entornos en los cuales la velocidad es crítica. Su huella de memoria es muy pequeña (en comparación a otros servidores web; como apache o IIS), una ligera carga en el CPU y su enfoque en velocidad hacen de lighttpd perfecto para servidores con demasiada carga.

5.1.2 Instalación de Lighttpd y sus Módulos sobre OpenWRT

La instalación del web server y los módulos en OpenWRT es muy sencilla. Para instalar Lighttpd en Attitude Adjustment, tecleamos como root:

- 1) Lo primero que debemos hacer es actualizar la lista de software disponible, esto hay que hacerlo cada vez que queramos instalar un paquete.
- 2) Instalar el software de Lighttpd en el directorio /opt o bien en /mnt/hicham.
- 3) Instalar los módulos del servidor web que estarán ubicados en la librería del directorio /opt del USB.

```
root@OpenWrt:~# opkg update
Downloading http://downloads.openwrt.org/attitude_adjustment/12.09-beta/ar71xx/g
eneric/packages/Packages.gz.
Inflating http://downloads.openwrt.org/attitude_adjustment/12.09-beta/ar71xx/gen
eric/packages/Packages.gz.
Updated list of available packages in /var/opkg-lists/attitude adjustment.
root@OpenWrt:~# opkg -d usb install lighttpd lighttpd-mod-fastcgi lighttpd-mod-auth
lighttpd-mod-status lighttpd-mod-acceslog lighttpd-mod-access lighttpd-mod-rewrite
```

Figura 19: Instalación de Lighttpd y sus extensiones

5.1.3 Configuración Básica del web server

Ahora debemos modificar el fichero Lighttpd.conf que está ubicado en la dirección /mnt/hicham/etc/lighttpd para añadir al servidor la posibilidad de servir PHP. El resultado es el siguiente:

```
# lighttpd configuration file
#
## modules to load
# all other module should only be loaded if really necessary
# - saves some time
# - saves memory
server.modules = (
    "mod_rewrite",
#   "mod_redirect",
#   "mod_alias",
    "mod_auth",
    "mod_status",
#   "mod_setenv",
    "mod_fastcgi",
#   "mod_proxy",
#   "mod_simple_vhost",
#   "mod_cgi",
#   "mod_ssi",
#   "mod_usertrack",
    "mod_expire",
#   "mod_webdav",
    "mod_compress",
    "mod_access",
    "mod_accesslog"
)

# force use of the "write" backend (closes: #2401)
server.network-backend = "write"
server.event-handler = "poll"
## a static document-root, for virtual-hosting take look at the
## server.virtual-* options
server.document-root = "/mnt/hicham/mimie/www/"

## where to send error-messages to
server.errorlog = "/mnt/hicham/var/log/lighttpd/error.log"

## files to check for if .../ is requested
index-file.names = ( "index.php", "index.html", "default.html", "index.htm", "default.htm" )
```

```

## mime type mapping
mime type.assign = (
    ".pdf" => "application/pdf",
    ".class" => "application/octet-stream",
    ".pac" => "application/x-ns-proxy-autoconfig",
    ".swf" => "application/x-shockwave-flash",
    ".wav" => "audio/x-wav",
    ".gif" => "image/gif",
    ".jpg" => "image/jpeg",
    ".jpeg" => "image/jpeg",
    ".png" => "image/png",
    ".svg" => "image/svg+xml",
    ".css" => "text/css",
    ".html" => "text/html",
    ".htm" => "text/html",
    ".js" => "text/javascript",
    ".txt" => "text/plain",
    ".dtd" => "text/xml",
    ".xml" => "text/xml",
    ".conf" => "text/plain",
    ".log" => "text plain",
    ".zip" => "application/zip",
    ".tar.gz" => "application/x-tgz",
    ".tgz" => "application/x-tgz",
    ".tar" => "application/x-tar"
)

## Use the "Content-Type" extended attribute to obtain mime type if possible
#mimetypes.use-xattr = "enable"

## send a different Server: header
## be nice and keep it at lighttpd
#server.tag = "lighttpd"
##### accesslog module
accesslog.filename = "/mnt/hicham/var/log/lighttpd/access.log"

$HTTP["url"] =~ "\.pdf$" {
    server.range-requests = "disable"
}
#####añadido para moodle
$HTTP["url"] =~ "\.(gif|GIF|jpg|JPG|png|PNG|swf|SWF|ico|ICO)$" {
    expire.url = ( "" => "access 2 days" )
}

$HTTP["url"] =~ "\.(css|CSS|js|JS)$" {
    expire.url = ( "" => "access 1 days" )
}
#some versions of lihttpd may require the following directives for full functionality

```

```

#etag.use-inode = "enable"
#etag.use-mtime = "enable"
#etag.use-size = "enable"
#static-file.etags = "enable"
##
# which extensions should not be handle via static-file transfer
#
# .php, .pl, .fcgi are most often handled by mod_fastcgi or mod_cgi
static-file.exclude-extensions = ( ".php", ".pl", ".fcgi" )

##### Options that are good to be but not necessary to be changed
#####

## bind to port (default: 80)
server.port = 8080
#server.socket = "[::]:8080"
## bind to localhost (default: all interfaces)
#server.bind = "localhost"

## error-handler for status 404
#server.error-handler-404 = "/error-handler.html"
#server.error-handler-404 = "/error-handler.php"

## to help the rc.scripts
server.pid-file = "/mnt/hicham/var/run/lighttpd.pid"

##### virtual hosts
##
## If you want name-based virtual hosting add the next three settings and load
## mod_simple_vhost
##
## document-root =
## virtual-server-root + virtual-server-default-host + virtual-server-docroot or
## virtual-server-root + http-host + virtual-server-docroot
##
#simple-vhost.server-root = "/home/weigon/wwwroot/servers/"
#simple-vhost.default-host = "grisu.home.kneschke.de"
#simple-vhost.document-root = "/pages/"

##
## Format: <errorfile-prefix><status>.html
## -> .../status-404.html for 'File not found'
#server.errorfile-prefix = "/www/error-"

## virtual directory listings
#server.dir-listing = "enable"

```

```

## send unhandled HTTP-header headers to error-log
#debug.dump-unknown-headers = "enable"

### only root can use these options
#
# chroot() to directory (default: no chroot() )
#server.chroot = "/"

## change uid to <uid> (default: don't care)
#server.username = "nobody"
#
server.upload-dirs = ( "/mnt/hicham/tmp" )

## change uid to <uid> (default: don't care)
#server.groupname = "nobody"

#### compress module añadido para que el funcionamiento de moodle
compress.cache-dir = "/mnt/hicham/tmp/compress.tmp"
compress.filetype = ("text/plain", "text/html", "text/javascript",
"text/css", "text/xml")

#### proxy module
## read proxy.txt for more info
#proxy.server = (
#   ".php" => (
#       "localhost" => (
#           "host" => "192.168.0.101",
#           "port" => 80
#       )
#   )
#)

#### fastcgi module
## read fastcgi.txt for more info
fastcgi.debug = 1
fastcgi.server = (
    ".php" => (
        "localhost" => (
            "socket" => "/mnt/hicham/tmp/php-fastcgi.socket",
            "bin-path" => "/mnt/hicham/usr/bin/php-fcgi"
        )
    )
)

#### CGI module
#cgi.assign = ( ".php" => "/opt/usr/bin/php-cgi" )
#cgi.assign = ( ".pl" => "/usr/bin/perl", ".cgi" => "/usr/bin/perl" )

#### SSL engine

```

```

#ssl.engine = "enable"
#ssl.pemfile = "server.pem"

#### status module
status.status-url = "/server-status"
status.config-url = "/server-config"

#### auth module
## read authentication.txt for more info
#auth.backend = "plain"
#auth.backend.plain.userfile = "lighttpd.user"
#auth.backend.plain.groupfile = "lighttpd.group"
#auth.require = (
#   "/server-status" => (
#       "method" => "digest",
#       "realm" => "download archiv",
#       "require" => "group=www|user=jan|host=192.168.2.10"
#   ),
#   "/server-info" => (
#       "method" => "digest",
#       "realm" => "download archiv",
#       "require" => "group=www|user=jan|host=192.168.2.10"
#   )
#)

#### url handling modules (rewrite, redirect, access)
#url.rewrite = ( "^/$" => "/server-status" )
#url.redirect = ( "^/wishlist/(.*)" => "http://www.123.org/$1" )

#### both rewrite/redirect support back reference to regex conditional using %n
#$HTTP["host"] =~ "^www\.(*)" {
#   url.redirect = ( "^/(.*)" => "http://%1/$1" )
#}

#### expire module
#expire.url = ( "/buggy/" => "access 2 hours", "/asdhas/" => "access plus 1
seconds 2 minutes" )

#### ssi
#ssi.extension = ( ".shtml" )

#### setenv
#setenv.add-request-header = ( "TRAV_ENV" => "mysql://user@host/db" )
#setenv.add-response-header = ( "X-Secret-Message" => "42" )

#### variable usage:
## variable name without "." is auto prefixed by "var." and becomes "var.bar"
#bar = 1
#var.mystring = "foo"

```

```

## integer add
#bar += 1
## string concat, with integer cast as string, result: "www.foo1.com"
#server.name = "www." + mystring + var.bar + ".com"
## array merge
#index-file.names = (foo + ".php") + index-file.names
#index-file.names += (foo + ".php")

#### include
#include /etc/lighttpd/lighttpd-inc.conf
## same as above if you run: "lighttpd -f /etc/lighttpd/lighttpd.conf"
#include "lighttpd-inc.conf"

#### include_shell
#include_shell "echo var.a=1"
## the above is same as:
#var.a=1

#### webdav
#$HTTP["url"] =~ "^/webdav($|/)" {
# webdav.is-readonly = "enable"
# webdav.sqlite-db-name = "/var/run/lighttpd-webdav-lock.db"
#}

```

5.1.4 Configurando las Directivas básicas del web server lighttpd

server.document-root = "/mnt/hicham/mimie/www/" Especifica el directorio por default para los sitios.

server.port = 8080 Especifica el puerto del servidor web lighttpd, y la interfaz de Luci utiliza por defecto el puerto 80. Por ejemplo si escribimos en el navegador 192.168.1.2:80 accederíamos a la interfaz Luci mientras que si escribimos 192.168.1.2:8080 en la misma dirección pero distinto puerto, accederíamos al servidor lighttpd

server.pid-file = "/mnt/hicham/var/run/lighttpd.pid", donde aparece el numero de proceso

server.username = www-data Usuario de Linux, ya sea Debian, Ubuntu, OpenWRT derivada sobre el cual corre lighttpd.

server.groupname = www-data Especifica el nombre de usuario y grupo con los cuales arranca/detiene el demonio del servidor lighttpd. Esta es una característica de seguridad, así de esta manera no corre con los privilegios de root.

server.bind = server-ip-address Especifica la dirección ip del servidor. Así mismo también es posible ingresar un hostname o localhost.

server.tag =lighttpd Se utiliza para asignar el nombre y el numero de versión de lighttpd (default). Esta es una característica de seguridad. Se puede configurar de esta manera:

```
server.tag = "Mi servidor web v1.0"
```

server.errorlog = /opt/var/log/php-error.log Especifica el archivo de bitacora de error.

accesslog.filename = /var/log/lighttpd

Especifica el archivo de bitácora de accesos, el cual puede ser utilizado por una aplicación de estadísticas de visitantes como webalizer.

index-file.names = ("index.php", "index.html", "default.html", "index.htm", "default.htm") . Un listado de los archivos por omisión en cada directorio.

```
server.modules = (  
mod_access,  
mod_accesslog,  
mod_fastcgi,  
mod_rewrite,  
mod_auth  
mod_compress,  
mod_expire  
);
```

Los módulos anteriormente descritos serán los únicos que cargara el servidor web. Claro que se pueden agregar más. De aquí el secreto de la rapidez y bajo consumo de memoria del servidor, veamos una descripción más detallada:

mod_access: El modulo de acceso es utilizado para denegar acceso a ciertos archivos.

mod_accesslog: Utilizado para escribir la bitácora CLF, flexible como apache.

mod_fastcgi: Soporte FastCGI para perl/PHP etc, que proporciona un mejor rendimiento. En lugar de crear un nuevo proceso (el programa CGI) para cada solicitud, FastCGI utiliza un único proceso persistente que se ocupa de muchas peticiones durante su vida útil.

mod_rewrite: Necesario para re-escribir Url's SEO compatibles.

mod_auth: Autenticación, usualmente para directorios protegidos por contraseña.

mod_compress: Salida de la compresión reduce la carga de la red y puede mejorar el rendimiento general y por supuesto del servidor web

mod_expire: Es útil para configurarlo para archivos estáticos que deben ser cacheados agresivamente como imágenes, hojas de estilo o similar.

```
mimetype.assign = (  
.pdf => application/pdf,  
.sig => application/pgp-signature,  
.mpg => video/mpeg,  
.mov => video/quicktime,  
.qt => video/quicktime,  
.avi => video/x-msvide,  
.asf => video/x-ms-asf,  
.asx => video/x-ms-asf,  
.wmv => video/x-ms-wmv,  
.bz2 => application/x-bzip,  
.tbz => application/x-bzip-compressed-tar,  
.tar.bz2 => application/x-bzip-compressed-tar  
)
```

Se utilizan para asignar el mapeo del mimetype (extensiones multipropósito de correo de internet") son una serie de convenciones o especificaciones dirigidas al intercambio a través de Internet de todo tipo de archivos (texto, audio, vídeo, etc.) de forma transparente para el usuario. Una parte importante del MIME está dedicada a mejorar las posibilidades de transferencia de texto en distintos idiomas y alfabetos.

Por último, hay que agregar unas cuantas líneas para que el servidor web Lighttpd pueda manejar las peticiones fastcgi de php.

```
fastcgi.server = ( ".php" =>  
    ( "localhost" =>  
        "socket" => "/tmp/php-fastcgi.socket",  
        "bin-path" => "/opt/bin/php-fcgi"  
    )  
)
```

5.2 Comprobación del Servicio Web

Ya que todo está configurado a nuestras necesidades, grabamos el archivo y reiniciamos el servidor web lighttpd para que los cambios tengan efectos:

```
root@OpenWrt:~# /opt/etc/init.d/lighttpd restart
```

Pero en mi caso el servidor web no se iniciaba en el arranque, creo que debido a la versión de Lighttpd, gracias a los mensajes de errores que están ubicados en este directorio `/opt/var/log/php-error.log` podré localizar y resolver el problema en la ruta de librerías y módulos que deberían estar en `/usr/lib` pero están en `/mnt/hicham/usr/lib` y `/op/usr/lib/lighttpd`:

```
libpcre.so.0
libssl.so.0.9.8
libcrypto.so.0.9.8
libz.so.1
mod_status.so
mod_fastcgi.so

mod_auth.so
mod_compress.so
mod_expire.so
mod_access.so
```

Para cada uno hacer un enlace:

```
Ln -s /opt/usr/lib/php/mod_compress.so /usr/lib
```

Ahora, probaremos nuestra configuración para asegurarnos que funciona todo correctamente. Creamos por ejemplo cualquier fichero `hola.html` para probar nuestra instalación y aseguramos de que el servidor web funciona bien.

```
✓ Vi /opt/www/hola.html
```

Y le agregamos las siguientes líneas de código:

```
<html>
<title>hola a todos</title>
<body>
<h3>hola a todos</h3>
</body>
</html>
```

Luego, abrimos nuestro explorador internet e ingresamos

✓ <http://192.168.1.2:8080/hola.html>

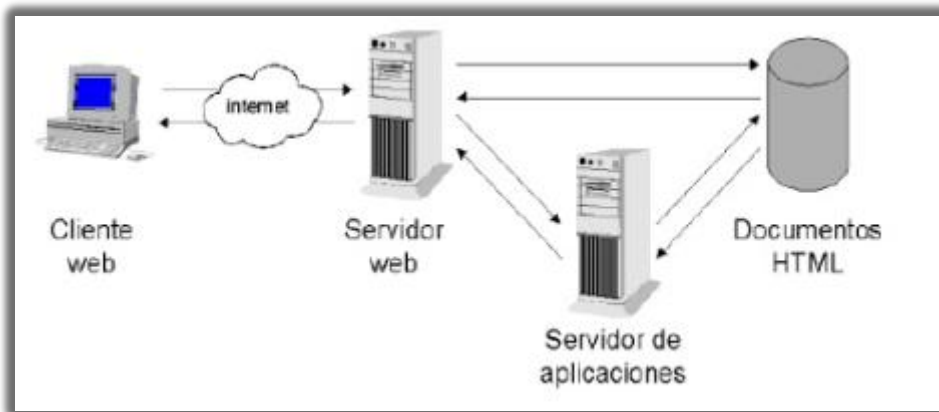
6. PHP 5.4.5

6.1 Instalación de PHP 5.4.5

6.1.1 Introducción a PHP

Lenguaje de script

- PHP es un lenguaje de script del lado del servidor.
- Los scripts PHP están incrustados en los documentos HTML y el servidor los interpreta y ejecuta antes de servir las páginas al cliente.
- El cliente no ve el código PHP sino los resultados que produce.



Breve historia de PHP

- ❖ Creado por Rasmus Lerdorf para uso personal en 1994.
- ❖ PHP = Personal Hypertext Processor.
- ❖ Es un módulo que se añade al servidor web.

¿Por qué PHP?

- ✓ Por sus ventajas: es potente, fácil de aprender, de libre distribución, permite el acceso a bases de datos (en este caso PostgreSQL, detallado en el punto 4.) y otras funcionalidades orientadas a la red.
- ✓ Dispone de abundante soporte en la web.

Requisitos

- Servidor web en nuestro caso Lighttpd
- Módulos de PHP
- Base de datos PostgreSQL para crear paginas dinámicas

6.1.2 Instalación y Configuración de PHP

Lo primero que hay que hacer, es instalar PHP versión 5.4.5 y sus extensiones en el directorio del USB /mnt/hicham, necesarias para que pueda funcionar la base de datos PostgreSQL y algunos módulos para el funcionamiento de Moodle. Por tanto, ejecutamos los siguientes comandos en el programa PuTTY vía ssh:

```
root@OpenWrt:~# opkg update
Downloading http://downloads.openwrt.org/attitude_adjustment/12.09-beta/ar71xx/g
eneric/packages/Packages.gz.
Inflating http://downloads.openwrt.org/attitude_adjustment/12.09-beta/ar71xx/gen
eric/packages/Packages.gz.
root@OpenWrt:~# opkg -d usb install php5 php5-fastcgi php5-mod-pgsql php5-mod-ic
onv php5-mod-mbstring php5-mod-session php5-mod-soap php5-mod-gd php5-mod-xml ph
p5-mod-zip php5-mod-sockets php5-mod-json php5-mod-hash php5-mod-curl php5-mod-c
type php5-mod-openssl php5-mod-dom php5-mod-tokenizer
```

Figura 20: Instalación de PHP y sus extensiones

Una vez instalado los paquetes deberemos modificar el archivo de configuración de PHP, llamado php.ini, situado en la carpeta /opt/etc/php.ini, para que se ajuste a las necesidades de este PFC.

A continuación, se muestra los ajustes de PHP necesarios para el funcionamiento de la aplicación web MOODLE, luego se detalla el código fuente del archivo php.ini con algunas modificaciones marcadas en negrita.

Ajustar PHP para el funcionamiento de Moodle:

- register_globals debe estar a OFF
- safe_mode debe estar a OFF.
- memory_limit debe ser por lo menos 40M (aunque algunas funciones pueden no funcionar si esta bajo). Se recomienda 128M.
- session.save_handler debe estar a FILES.
- magic_quotes_gpc debe estar a OFF.
- magic_quotes_runtime debe estar a OFF.
- file_uploads debe estar a ON.
- session.auto_start debe estar a OFF.
- session.bug_compat_warn debe estar a OFF.
- post_max_size y upload_max_filesize: es el tamaño máximo de archivo que se pueden cargar en la carpeta moodledata de moodle

El código script del archivo php.ini:

```
[PHP]
```

```
zend.ze1_compatibility_mode = Off
```

```
; Language Options
```

```
engine = On
```

```
;short_open_tag = Off
```

```
precision = 12
```

```
y2k_compliance = On
```

```
output_buffering = Off
```

```
;output_handler =
```

```
zlib.output_compression = Off
```

```
;zlib.output_compression_level = -1
```

```
;zlib.output_handler =
```

```
implicit_flush = Off
```

```
unserialize_callback_func =
```

```
serialize_precision = 100
```

```
;open_basedir =
```

```
disable_functions =
```

```
disable_classes =
```

```
; Colors for Syntax Highlighting mode. Anything that's acceptable in
```

```
; <span style="color: ???????"> would work.
```

```
;highlight.string = #DD0000
```



```

;highlight.comment = #FF9900
;highlight.keyword = #007700
;highlight.bg      = #FFFFFF
;highlight.default = #0000BB
;highlight.html    = #000000

;ignore_user_abort = On
;realpath_cache_size = 16k
;realpath_cache_ttl = 120

; Miscellaneous

expose_php = On

; Resource Limits

max_execution_time = 30 ; Maximum execution time of each script, in seconds.
max_input_time = 60   ; Maximum amount of time each script may spend parsing
request data.
;max_input_nesting_level = 64
memory_limit = 128M // debe ser por lo menos 40M (aunque algunas
funciones pueden no funcionar si esta bajo). Se recomienda 128M.

; Error handling and logging

; Error Level Constants:
; E_ALL          - All errors and warnings (includes E_STRICT as of PHP 6.0.0)
; E_ERROR        - fatal run-time errors
; E_RECOVERABLE_ERROR - almost fatal run-time errors
; E_WARNING      - run-time warnings (non-fatal errors)
; E_PARSE        - compile-time parse errors
; E_NOTICE       - run-time notices (these are warnings which often result
;                from a bug in your code, but it's possible that it was
;                intentional (e.g., using an uninitialized variable and
;                relying on the fact it's automatically initialized to an
;                empty string)
; E_STRICT       - run-time notices, enable to have PHP suggest changes
;                to your code which will ensure the best interoperability
;                and forward compatibility of your code
; E_CORE_ERROR   - fatal errors that occur during PHP's initial startup
; E_CORE_WARNING - warnings (non-fatal errors) that occur during PHP's
;                initial startup
; E_COMPILE_ERROR - fatal compile-time errors
; E_COMPILE_WARNING - compile-time warnings (non-fatal errors)
; E_USER_ERROR   - user-generated error message
; E_USER_WARNING - user-generated warning message
; E_USER_NOTICE  - user-generated notice message
; E_DEPRECATED   - warn about code that will not work in future versions
;                of PHP

```

```
; E_USER_DEPRECATED - user-generated deprecation warnings
;
; Common Values:
; E_ALL & ~E_NOTICE (Show all errors, except for notices and coding standards
warnings.)
; E_ALL & ~E_NOTICE | E_STRICT (Show all errors, except for notices)
; E_COMPILE_ERROR|E_RECOVERABLE_ERROR|E_ERROR|E_CORE_ERROR (Show
only errors)
; E_ALL | E_STRICT (Show all errors, warnings and notices including coding
standards.)
; Default Value: E_ALL & ~E_NOTICE
```

```
error_reporting = E_ALL & ~E_NOTICE & ~E_STRICT // Para activar y
mostrar los tipos de errores
```

```
display_errors = On //Habilitar a ON para imprimir por pantalla los errores.
display_startup_errors = On // Imprimir por pantalla los errores a la
arrancar PHP
```

```
log_errors = On //Para registrar los mensajes de errores en un registro
```

```
log_errors_max_len = 1024
ignore_repeated_errors = Off
ignore_repeated_source = Off
report_memleaks = On
;report zend_debug = 0
track_errors = Off
;html_errors = Off
;docref_root = "/phpmanual/"
;docref_ext = .html
;error_prepend_string = "<font color=#ff0000>"
;error_append_string = "</font>"
; Log errors to specified file.
```

```
error_log = /opt/var/log/php_errors.log // para guardar los mensajes de
errores en el archivo llamado php_errors.log
```

```
; Log errors to syslog.
;error_log = syslog
```

```
; Data Handling
```

```
;arg_separator.output = "&"
;arg_separator.input = "&"
variables_order = "EGPCS"
request_order = "GP"
register_globals = Off
register_long_arrays = Off
register_argc_argv = On
auto_globals_jit = On
post_max_size = 32M // el tamaño máximo de archivo que se pueden cargar.
magic_quotes_gpc = Off
magic_quotes_runtime = Off
magic_quotes_sybase = Off
```

```

auto_prepend_file =
auto_append_file =
default_mimetype = "text/html"
;default_charset = "iso-8859-1"
;always_populate_raw_post_data = On

; Paths and Directories

; UNIX: "/path1:/path2"
;include_path = "./php/includes"
include_path = ".: /mnt/hicham/usr/lib:/mnt/hicham/usr/lib/php"
doc_root = "/opt/mimie/www" // Directorio raíz de PHP en el servidor
user_dir =
extension_dir = "/opt/usr/lib/php" // Para cargar dinámicamente desde
este directorio las extensiones .so
enable_dl = On
;cgi.force_redirect = 1
;cgi.nph = 1
;cgi.redirect_status_env = ;
cgi.fix_pathinfo=1
;fastcgi.impersonate = 1;
;fastcgi.logging = 0
;cgi.rfc2616_headers = 0

; File Uploads

file_uploads = On
upload_tmp_dir = "/opt/tmp"
upload_max_filesize = 10M //es el tamaño de archivo subidos a la carpeta de
moodledata de Moodle
max_file_uploads = 20

; Fopen wrappers

allow_url_fopen = On
allow_url_include = Off
;from="john@doe.com"
;user_agent="PHP"
default_socket_timeout = 60
;auto_detect_line_endings = Off

; Dynamic Extensions

//Habilitamos algunas extensiones para el funcionamiento de Moodle
y PostgreSQL

extension=ctype.so
extension=curl.so

```

```
extension=dom.so
extension=exif.so
extension=ftp.so
extension=gd.so
extension=gmp.so
extension=hash.so
extension=iconv.so
extension=json.so
;extension=ldap.so
extension=mbstring.so
;extension=mcrypt.so
;extension=mysql.so
;extension=pdo-pgsql.so
extension=pdo_sqlite.so
extension=pgsql.so
extension=session.so
extension=soap.so
extension=sockets.so
;extension=sqlite.so
extension=sqlite3.so
extension=tokenizer.so
extension=xml.so
;extension=xmlreader.so
;extension=xmlwriter.so
extension=zip.so
;extension=simplexml.so
;extension=fileinfo.so
;extension=shmop.so
;extension=dio.so
;extension=pcntl.so
extension=apc.so
;extension=http.so
;extension=libevent.so
;extension=sysvsem.so
;extension=sysvshm.so
; Module Settings
```

[APC]

```
apc.enabled = 1
```

```
apc.shm_segments = 1 ;The number of shared memory segments to allocate for
the compiler cache.
```

```
;apc.shm_size = 4M ;The size of each shared memory segment.
```

//Hay que quitar los corchetes de Date para el funcionamiento de cualquier página dinámica sino saldrá un mensaje de error diciendo que no se puede abrir la pagina o bien aparece en blanco

Date

date.timezone = "Europe/Madrid" // Tiene que ser igual que La zona horaria predeterminada usada en el fichero TZ del sistema operativo OpenWRT

```
;date.default_latitude = 31.7667  
;date.default_longitude = 35.2333  
;date.sunrise_zenith = 90.583333  
;date.sunset_zenith = 90.583333
```

```
[filter]  
;filter.default = unsafe_raw  
;filter.default_flags =
```

```
[iconv]  
;iconv.input_encoding = ISO-8859-1  
;iconv.internal_encoding = ISO-8859-1  
;iconv.output_encoding = ISO-8859-1
```

```
[sqlite]  
;sqlite.assoc_case = 0
```

```
sqlite3  
;sqlite3.extension_dir = "/opt/usr/lib"
```

```
[Pdo_mysql]  
;pdo_mysql.cache_size = 2000  
;pdo_mysql.default_socket=
```

```
[MySQL]  
;mysql.allow_local_infile = On  
;mysql.allow_persistent = On  
;mysql.cache_size = 2000  
;mysql.max_persistent = -1  
;mysql.max_links = -1  
;mysql.default_port =  
;mysql.default_socket =  
;mysql.default_host =  
;mysql.default_user =  
;mysql.default_password =  
;mysql.connect_timeout = 60  
mysql.trace_mode = Off
```

```
[PostgreSQL]  
;pgsql.allow_persistent = On  
;pgsql.auto_reset_persistent = Off  
;pgsql.max_persistent = -1  
;pgsql.max_links = -1  
;pgsql.ignore_notice = 0  
;pgsql.log_notice = 0
```

```

[Session]
session.save_handler = files
session.save_path = "/opt/tmp" //directorio para guardar las cookies
session.use_cookies = 1
;session.cookie_secure =
session.use_only_cookies = 1
session.name = MoodleSession // nombre de la sesión que se usa como nombre de cookie
session.auto_start = 0 // para que Moodle no se conecta automáticamente
session.cookie_lifetime = 1600 // la cookie tiene un tiempo de vida de 1600 segundos
session.cookie_path = /
session.cookie_domain =
session.cookie_httponly =
session.serialize_handler = php
session.gc_probability = 1
session.gc_divisor = 100
session.gc_maxlifetime = 7200 // Las sesiones que no se han actualizado desde los últimos 7200 segundos serán eliminadas.
session.bug_compat_42 = On
session.bug_compat_warn = Off // solo se habilita a versiones anteriores de PHP que tienen error no documentada que permite inicializar una variable de sesión en el ámbito global, aunque register_globals esté deshabilitado.
session.referer_check =
session.entropy_length = 0
;session.entropy_file = /dev/urandom
session.entropy_file =
;session.entropy_length = 16
session.entropoy_length = 0
session.cache_limiter = nocache
session.cache_expire = 180 // la session se caduca cada 180 segundos
session.use_trans_sid = 0
session.hash_function = 0
session.hash_bits_per_character = 4
url_rewriter.tags = "a=href,area=href,frame=src,input=src,form=,fieldset="

```

```

[mbstring]
;mbstring.language = Japanese
;mbstring.internal_encoding = EUC-JP
;mbstring.http_input = auto
;mbstring.http_output = SJIS
;mbstring.encoding_translation = Off
;mbstring.detect_order = auto
;mbstring.substitute_character = none;
;mbstring.func_overload = 0
;mbstring.strict_detection = Off
;mbstring.http_output_conv_mimetype=
;mbstring.script_encoding=

```

```
[gd]
;gd.jpeg_ignore_warning = 0
```

```
[exif]
;exif.encode_unicode = ISO-8859-15
;exif.decode_unicode_motorola = UCS-2BE
;exif.decode_unicode_intel = UCS-2LE
;exif.encode_jis =
;exif.decode_jis_motorola = JIS
;exif.decode_jis_intel = JIS
```

```
[soap]
soap.wSDL_cache_enabled=1
soap.wSDL_cache_dir="/tmp"
soap.wSDL_cache_ttl=86400
soap.wSDL_cache_limit = 5
```

```
[sysvshm]
;sysvshm.init_mem = 10000
```

```
[ldap]
ldap.max_links = -1
```

```
[mcrypt]
;mcrypt.algorithms_dir=
;mcrypt.modes_dir=
```

Una vez configurado el archivo php.ini, pasamos a comprobar si carga los módulos correctamente mediante el comando:

```
root@OpenWrt:~# php-fcgi -m
[PHP Modules]
apc
cgi-fcgi
Core
ctype
curl
date
dom
ereg
exif
ftp
gd
gmp
hash
iconv
json
libxml
mbstring
openssl
pcre
PDO
pdo_sqlite
pgsql
posix
Reflection
session
soap
sockets
SPL
sqlite3
standard
tokenizer
xml
zip
zlib
[Zend Modules]
```

Figura 21: Captura de pantalla al ejecutar el comando php-fcgi

Esto no debería lanzar ningún error y debería mostrar los módulos inicializados como en la figura 19

Ahora, probaremos nuestra configuración para asegurarnos que funciona todo correctamente. Creamos un archivo phpinfo.php para probar nuestra instalación y aseguramos de que el servidor web funciona bien.

✓ Vi /opt/mimie/www/phpinfo.php

Y le agregamos las siguientes líneas de código:

```
<?php
    Phpinfo();
?>
```


Una vez creado y con el servidor web Lighttpd arrancado, escribimos en la barra de direcciones del navegador:

- ✓ <http://192.168.1.2:8080/phpinfo.php>

Y nos cargara la pagina .php con toda la información que hemos configurado en el php.ini (ver figura 20)

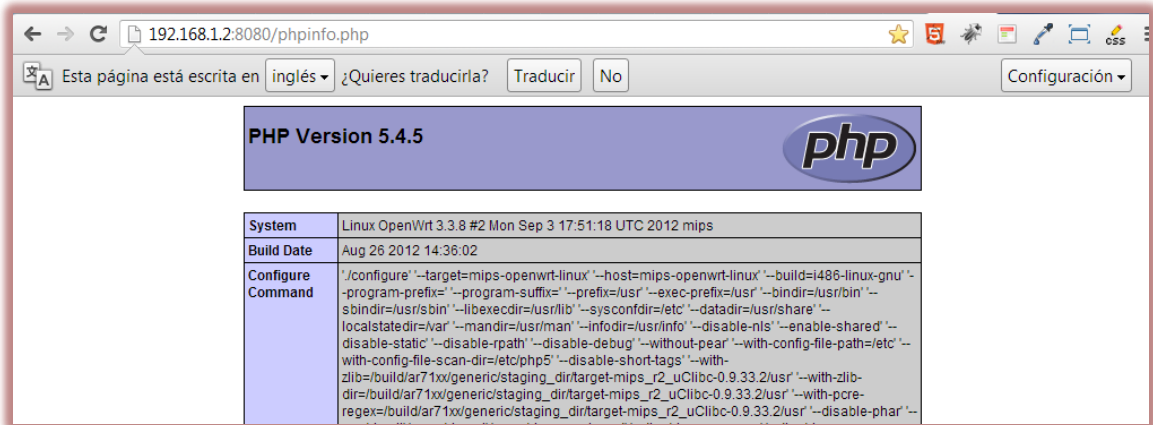


Figura 22: Parte de la página cargada phpinfo.php

En este punto nuestro servidor ya puede leer páginas web con contenido dinámico.

7. Base de Datos PostgreSQL

7.1 Instalación de bases de datos

7.1.1 Introducción

Las bases de datos son el método preferido para el almacenamiento estructurado de datos. Desde las grandes aplicaciones multiusuario, hasta los teléfonos móviles y las agendas electrónicas utilizan tecnología de bases de datos para asegurar la integridad de los datos y facilitar la labor tanto de usuarios como de los programadores que las desarrollaron.

En este proyecto trataremos de utilizar una aplicación web de tipo educativo virtual es decir un sistema de gestión de cursos y datos que estos últimos se almacenan en una base de datos SQL.

Los sistemas de gestión de base de datos soportadas por Moodle son: MySQL, PostgreSQL, Oracle, SQLite3.

Se intento en principio de instalar tanto MySQL como SQLite3 pero las páginas de Moodle tardaban mucho en abrirse casi de 3 minutos hasta 5 minutos por sencilla razón que los dos servidores consumían demasiada cantidad de memoria. Finalmente se utilizo el PostgreSQL que su ventaja:

- consume menos memoria es decir, no necesita muchos requerimientos de sistema.
- la velocidad de respuesta de una página es de 30segudos hasta 1 minutos.

7.1.2 ¿Qué es PostgreSQL?

Es un sistema de base de datos relacional orientada a objetos, de la universidad de Berkeley. El director de este proyecto es el profesor Michael Stonebraker, y fue patrocinado por Defence Advanced Research Projects Agency (DARPA), el Army Research Office (ARO), el NationalScience Foundation (NSF).

PostgreSQL esta publicado bajo una licencia BSD. Es un proyecto de código libre debido a estas características sus mejoras han sido un poco más rápidas en comparación con otros sistemas de base de datos. Originalmente este programa se llamaba postgre posteriormente cambio su nombre postgres95 hasta llegar a ser llamado PostgreSQL como es conocido en la actualidad.

7.1.3 Instalación de PostgreSQL en OpenWRT

Para utilizar el servidor de bases de datos PostgreSQL en OpenWRT es necesario instalar el paquete `postgresql-server` ejecutando los siguientes comandos en el programa PuTTY vía ssh:

```

|_| W I R E L E S S F R E E D O M
-----
ATTITUDE ADJUSTMENT (12.09-beta, r33312)
-----
* 1/4 oz Vodka      Pour all ingredients into mixing
* 1/4 oz Gin        tin with ice, strain into glass.
* 1/4 oz Amaretto
* 1/4 oz Triple sec
* 1/4 oz Peach schnapps
* 1/4 oz Sour mix
* 1 splash Cranberry juice
-----
root@OpenWrt:~# opkg update 1)actualizar los paquetes
Downloading http://downloads.openwrt.org/attitude_adjustment/12.09-beta/ar71xx/g
eneric/packages/Packages.gz.
Inflating http://downloads.openwrt.org/attitude_adjustment/12.09-beta/ar71xx/gen
eric/packages/Packages.gz.
Updated list of available packages in /var/opkg-lists/attitude_adjustment.
root@OpenWrt:~# opkg -d usb install postgresql-server 2)instalar postgresql-server en USB
```

Figura 23: Instalar el paquete Postgresql-Server

7.2 Configuración de PostgreSQL

Una vez instalado PostgreSQL, ahora tendremos que configurar un par de cosas para poder ponerlo a funcionar. En este apartado veremos cómo iniciar el directorio de base de datos `/opt/www/postgresql/data` que es donde el gestor de bases de datos guardara la información, cursos, contenidos general de la aula virtual así como crear el usuario `postgres` que desde luego se encargara de utilizar el sistema

Antes de empezar hay que modificar la ruta `/opt/etc/config/postgresql` donde se van a guardar los datos y también cambiar la ruta donde se guardara los mensajes de errores:

```
config postgresql config
option PGUSER postgres
option PGDATA /mnt/hicham/www/postgresql/data
option PGLOG /mnt/hicham/www/postgresql/data/postgresql.log
option PG_CTL /mnt/hicham/usr/bin/pg_ctl
```

7.2.1 Usuario PostgreSQL

Lo primero que necesitamos es un usuario que se va a encargar de lanzar La instalación de PostgreSQL, y será el dueño de los ficheros del sistema gestor de bases de datos.

Se le puede llamar como uno quiera, yo voy a llamarlo postgres.

Antes de crear al usuario, tenemos que crear el grupo y un password, para ello nos conectamos como root en el terminal y dentro del router escribimos:

```
# echo 'postgres:x:1000:1000:postgres:/home/postgres:/bin/sh' >>/etc/passwd
# echo 'postgres:x:1000:' >>/etc/group
# passwd postgres
# mkdir /home/postgres
# chown postgres:postgres /home/postgres
```

Con esto hemos creado el usuario postgres, en /home/postgres, perteneciente al grupo postgres, y con password postgres que va a ser el encargado, por lo menos de administrar, lanzar, detener el sistema gestor de bases de datos.

7.2.2 Preparando la instalación del servidor PostgreSQL

Para la instalación y configuración, conectaremos al router, mediante el programa PuTTY vía ssh si estamos en Windows y escribimos el siguiente comando:

```
Ssh postgres@192.168.1.2.
```

Ahora tendremos 2 terminales abiertos, uno con el usuario root donde vamos a crear una base de datos con un usuario y un password para el sistema educativo virtual Moodle, y en el segundo terminal postgres donde se almacenaran los contenidos de dicha base de datos, entonces para empezar a funcionar y a guardar información, tenemos que habilitar primero un lugar donde se almacenaran estos contenidos de la Aula Virtual, para hacer esto, tenemos el comando initdb, que nos permite crear este directorio /opt/www/postgresql/data

```
postgresql@OpenWrt:~$ LC_COLLATE="C" initdb --pwprompt -D
/mnt/hicham/www/postgresql
The files belonging to this database system will be owned by user "postgresql".
This user must also own the server process.
```

```
The database cluster will be initialized with locale C.
The default database encoding has accordingly been set to SQL_ASCII.
The default text search configuration will be set to "english".
```

```
fixing permissions on existing directory /mnt/Hicham/www/postgresql ... ok
creating subdirectories ... ok
selecting default max_connections ... 100
selecting default shared_buffers ... 28MB
creating configuration files ... ok
creating template1 database in /mnt/Hicham/www/postgresql/base/1 ... ok
initializing pg_authid ... ok
Enter new superuser password:
Enter it again:
setting password ... ok
initializing dependencies ... ok
creating system views ... ok
loading system objects' descriptions ... ok
creating conversions ... ok
creating dictionaries ... ok
setting privileges on built-in objects ... ok
creating information schema ... ok
loading PL/pgSQL server-side language ... ok
vacuuming database template1 ... ok
copying template1 to template0 ... ok
copying template1 to postgres ... ok
```

WARNING: enabling "trust" authentication for local connections
You can change this by editing pg_hba.conf or using the -A option the next time you run initdb.

Success. You can now start the database server using:

```
postgres -D /mnt/hicham/www/postgresql  
or  
pg_ctl -D /mnt/hicham/www/postgresql -l logfile start
```

Postgres es el daemon en sí, por lo que al lanzarlo directamente, hará un fork y se pondrá en background a escuchar las peticiones de los clientes y usuarios que se conecten a las bases de datos.

Para lanzar simplemente postgres, podemos hacerlo de la siguiente manera:

\$ postgres -D /opt/www/postgresql/data

-D: Este el parámetro le dice al postgres que lo que sigue es el directorio donde se almacenaran las operaciones realizadas en cliente.

Una vez lanzado el daemon postgres seguida de la ruta donde se almacenaran la información, nos conectamos por terminal al root para ejecutar un comando psql que se conecte al servidor postgres para que nos liste las bases de datos que existen (ver figura21).

```
root@OpenWrt:/mnt/Hicham/www/postgresql# psql -d postgres -U postgres  
psql (9.0.1)  
Type "help" for help.  
  
postgres=# help  
You are using psql, the command-line interface to PostgreSQL.  
Type: \copyright for distribution terms  
       \h for help with SQL commands  
       \? for help with psql commands  
       \g or terminate with semicolon to execute query  
       \q to quit  
postgres=# \l  
  
                List of databases  
 Name | Owner  | Encoding | Collation | Ctype | Access privileges  
-----+-----+-----+-----+-----+-----  
 postgres | postgres | SQL_ASCII | C         | C     |  
 template0 | postgres | SQL_ASCII | C         | C     | =c/postgres      +  
           |          |          |          |          | postgres=Ctc/postgres  
 template1 | postgres | SQL_ASCII | C         | C     | =c/postgres      +  
           |          |          |          |          | postgres=Ctc/postgres  
(3 rows)
```

Figura 24: Lista de base de datos

- **Psql:** es la utilidad de gestión de base de datos en línea de comandos de PostgreSQL, es aquí donde vamos a ejecutar las consultas SQL que queramos, etc..., y creamos al usuario.

7.2.3 Creación de base de datos junto con un usuario y password para Moodle

A la hora de utilizar el sistema gestor de bases de datos, lo mejor es crear un usuario moodleuser2 que controlará la creación de la base de datos Moodle2

- ✓ postgres=# create user moodleuser2 createdb;
- ✓ postgres=# create database moodle2 **WITH ENCODING = 'UTF8'**;
- ✓ postgres=# alter user moodleuser2 norestricteddb;
- ✓ postgres=# alter user moodleuser2 with encrypted password 'moodle';

Donde:

- **Createdb:** Nos indica que el usuario moodleuser2 va a poder crear un base de datos que la hemos llamado moodle2. Lo ideal es que tan sólo este usuario pueda crear bases de datos, una vez creadas estas, se les puede cambiar el dueño, así como asignar permisos a quien queramos, lo que es más que aconsejable.
- **Encrypted:** Hace que el password se guarde en la tabla de usuarios de PostgreSQL en formato md5, en lugar de en texto plano.
- **UTF-8:** Es un método de codificación de ASCII para Unicode (ISO-10646), el Conjunto de Caracteres Universal. Éste codifica la mayoría de los sistemas de escritura del mundo en un solo conjunto de caracteres, permitiendo la mezcla de lenguajes y guiones en un mismo documento sin la necesidad de ajustes para realizar los cambios de conjuntos de caracteres.

List of databases					
Name	Owner	Encoding	Collation	Ctype	Access privileges
moodle2	moodleuser2	UTF8	C	C	
postgres	postgres	SQL_ASCII	C	C	
template0	postgres	SQL_ASCII	C	C	=c/postgres
					postgres=C/c/postgre
template1	postgres	UTF8	C	C	

(4 rows)

Figura 25: Insertar la Base de Datos moodle2 en lista

De esta manera tenemos la nueva base de datos Moodle2, con codificación UTF-8, usuario moodleuser2, contraseña ‘moodle’, datos que tendremos que introducir en la página de instalación de Moodle o bien manualmente en el archivo de configuración config.php.

7.2.4 Permisos

Una vez creado el usuario y las base de datos, el siguiente paso es echarle un vistazo al fichero de control de acceso de Postgresql, el pg_hba.conf, donde establecemos quien se puede conectar y a que. El fichero pg_hba.conf tiene el siguiente formato, como podemos ver si le echamos un ojo al propio fichero:

```
# TYPE DATABASE USER CIDR-ADDRESS METHOD
# "local" is for Unix domain socket connections only
local all all trust
local all all password
local moodle2 moodleuser2 md5
```

- **Local:** Conexiones de usuarios dentro del propio host, por ejemplo un usuario conectado por ssh, que ejecuta un psql a una base de datos.
- **Trust:** Con este método confiamos plenamente en el usuario y ni siquiera le autenticamos, con que se identifique como ese nombre de usuario, se podrá conectar.
- **Password:** Método usado si hemos establecido un password en el momento de crear al usuario con create user, pero sin usar la opción encrypted.

- **md5:** Esta si hemos utilizado la opción encrypted.

8. Virtual E-learning

Moodle

8.1 Entornos de Aprendizajes Virtuales

Un entorno de aprendizaje virtual (Virtual Learning Environment) es un sistema de software diseñado para apoyar el proceso de enseñanza y aprendizaje en un entorno educativo. Se puede definir como un espacio o comunidad organizada que tiene el propósito de aprender, donde los contenidos de aprendizaje están disponibles en línea y cuenta con herramientas interactivas que permiten la retroalimentación automática a los estudiantes.

Actualmente hay más de 45 programas de software libre (Open Source Software) que ofrecen entornos de aprendizaje virtuales de manera gratuita, entre los más conocidos están: Moodle, Ilias, Claroline, SAKAI, etc.

8.2 Eligiendo el software adecuado

8.2.1 Moodle

Moodle (Modular Object-Oriented Dynamic Learning Environment) fue creado por Martín Dougiamas y es un sistema de gestión de cursos de código abierto de distribución libre basado en una pedagogía de aprendizaje social. Este entorno de aprendizaje virtual (Virtual Learning Environment) permite la creación de comunidades de aprendizaje donde tanto los educadores, administradores y usuarios contribuyen en conocimiento de dicha comunidad.

8.2.1.1 ¿Por qué se eligió Moodle en este PFC?

Las razones por las que se eligió Moodle como sistema de gestión de aprendizaje en este PFC son:

- **Software de distribución gratuita y de software libre:** Este es un factor sumamente importante, dado que como se mencionó en principio del PFC (1. objetivos), los países subdesarrollados son países pobres carentes de recursos económicos, factor que imposibilita a diferentes centros educativos la compra de licencias altamente costosas de plataformas privadas que desempeñan los mismos roles y funcionalidades que Moodle (por ejemplo blackboard learning system)

- **Comunidad:** Nuestra comunidad OpenMoodle (nombre de la red del router WRT160NL) cuenta casi de 100 alumnos por curso, formada de un docente que gestiona cursos.
- **Documentación disponible de Moodle (Moodle Docs):** Donde se cuenta con más de 1858 artículos y cualquier usuario puede contribuir con esta documentación.
 - ✓ **Preguntas frecuentes (Frequent Asked Questions):** Cuenta con una lista de temas de preguntas frecuentes.
 - ✓ **Foros (forums):** Espacio donde todos los usuarios que pertenecen a la comunidad OpenMoodle (nombre de la red del router WRT160NL) pueden debatir, preguntar, proporcionar o brindar soporte e información.
- **Características:** Este Entorno Modular de Aprendizaje Dinámico Orientado a Objetos (Moodle) promueve una pedagogía constructivista social mediante interacción de los estudiantes y profesores en la construcción del conocimiento. Los principales Módulos con los que cuenta Moodle son los siguientes:
 - ✓ **Módulo base de datos:** Esta actividad permite solo al dueño del router WRT160NL de registrar o inscribir a sus alumnos en formularios que estarán guardadas durante el año en una base de datos (PostgreSQL) hasta el fin del curso.
 - ✓ **Módulo tareas:** Permite la entrega de tareas cuando el alumno este conectado al router WRT160NL del docente, donde el alumno pueda subir archivos para que puedan ser revisadas por el profesor.
 - ✓ **Módulo recurso:** Da la posibilidad a los docentes de facilitar contenidos en formato digital a los alumnos inscritos al cursos, como por ejemplo presentaciones en Power Point, PDF, Word, Videos, tutoriales y todo tipo de contenido multimedia.
 - ✓ **Módulo cuestionario:** Permite la realización y evaluación de preguntas o exámenes on-line con diferentes tipos de consultas como opción múltiple, verdadero/falso, respuestas abiertas/cerradas etc.

- ✓ **Módulo foro:** Esta actividad tiene como finalidad fomentar la discusión y debate de los diferentes temas del curso.
 - ✓ **Módulo lección:** Ofrece la posibilidad al profesor de poder dividir un tema en pequeños partes, mostrando así a los estudiantes parte por parte. Generalmente para acceder a la siguiente parte, el estudiante debe responder correctamente el cuestionario de cada sub tema.
 - ✓ **Módulo wiki:** Esta actividad permite a los usuarios de Moodle crear contenidos web que pueden ser editadas por ellos mismos. Por ejemplo el docente puede crear una wiki con la finalidad de que un grupo de alumnos trabaje en un mismo documento.
 - ✓ **Módulo glosario:** Todos los participantes pueden crear y mantener una lista de definiciones de palabras como un diccionario.
 - ✓ **Otros módulos:** Moodle al ser modular permite la integración de más de 627 módulos desarrollados por terceras partes.
- **Popularidad:** Actualmente Moodle se ha convertido en la plataforma de e-learning de uso más extendido a nivel mundial encontrándose en más de 195 países

8.3 Instalación del Entorno de Aprendizaje Virtual Moodle

Para poder concretar el despliegue de Moodle es conveniente considerar los requerimientos que son necesarios para contar con este entorno virtual de aprendizaje.

8.3.1 Requerimientos

Nuestro objetivo en este proyecto es desarrollar que Moodle este principalmente instalado y funcionando en OpenWRT usando Lighttpd, php 5.4.5 y PostgreSQL

Requerimientos adicionales:

- Librería GD para poder construir los gráficos de los registros de Moodle.
- mbstring - es requerido para manipular cadenas de caracteres multi-byte (iconv también es recomendable para la versión Moodle 1.9 que hemos instalado en este proyecto).
- La extensión pgsql para utilizar una base de datos PostgreSQL.
- La extensión zlib es necesaria si vamos a utilizar las funcionalidades zip/unzip.

- otras extensiones PHP podrían ser necesarias dependiendo de las funcionalidades opcionales de Moodle por ejemplo tokenizer, openssl, sockets....etc.

8.3.2 Descarga e instalación de Moodle

8.3.2.1 Descargar Moodle

Para descargar la plataforma, accedemos a la zona de descargas de la página de Moodle <http://download.moodle.org/>.

En la página principal de descargas hay varias secciones:

- **Standard packages:** en esta sección se encuentran todas las versiones disponibles para descargar (en formato ZIP o TGZ).
- **Moodle for Windows / MacOS X:** Estas secciones contienen versiones de la aplicación empaquetadas con un servidor web / intérprete de PHP (XAMPP para Windows y MAMP para MacOS).
- **Themes:** Galería de temas visuales para la plataforma. Pueden descargarse e instalarse pero, como ya se verá más adelante, la propia plataforma incluye unos cuantos temas para elegir, por lo que antes de descargar nada, puede ser útil ver si alguno de los predeterminados nos puede valer.
- **Language packs:** paquetes de idiomas para la aplicación. Se pueden descargar e instalar directamente desde la plataforma, aunque requiere acceso a Internet.

Yo me he decantado por descargar la versión 1.9.19+ en formato .tgz desde los "Standard packages". Esta es la mejor opción para nuestro router, ya que probé con versiones muy recientes, que al cargar las páginas tardaban mucho en abrirse, creo por falta de Memoria RAM.



Figura 26: Descargar la versión 1.9.19+

8.3.2.2 Instalar Moodle

Para poder instalar Moodle en el router WRT160NL se deben tomar en cuenta los requerimientos mencionados anteriormente. Una vez descargada nuestra versión de Moodle en Windows, el siguiente paso es abrir desde PuTTY un terminal para mandar por scp a nuestro router el paquete descargado:

```
Scp /escrtorio /moodle1.9.19_lastest.tgz root@192.168.1.2:8080:/opt/mimie/www
```

Una vez descomprimido el paquete dentro de la carpeta /opt/mimie/www con el comando:

```
tar -xvzf moodle1.9.19_lastest.tgz
```

El siguiente paso es crear en el mismo directorio /opt/mimie, una nueva carpeta llamada moodledata con permisos de lectura, escritura y ejecución al dueño, grupo y otros usuarios (chmod 777). En esta carpeta se almacenaran todos los datos de los usuarios registrados en Moodle, como nombre de usuarios, contraseña, apuntes, idiomas, correo electrónico, notas, y otro tipo de información.

Proceso de instalación: Abrimos en el navegador la siguiente dirección <http://192.168.1.2:8080/moodle/install.php> y continuamos con el proceso de instalación, hasta que se crea automáticamente un script llamado config.php dentro del directorio Moodle.

Los pasos que debemos dar para la instalación desde el primer paso serán los siguientes:

- En principio sólo deberíamos configurar el idioma y el país



Figura 27: Configuración de idioma

- Comprobando los ajustes de PHP para que Moodle pueda funcionar correctamente.



Figura 28: Comprobación de ajustes de PHP

Una vez elegimos el idioma con el que vamos a seguir la instalación y comprobar los ajustes de PHP, procedemos a los siguientes:

- En este paso, debemos indicarle a Moodle la dirección web, así como el directorio donde hemos instalado Moodle; y por último, el directorio de datos moodledata donde Moodle creará los datos necesarios para las tablas que necesita para trabajar.
- debemos indicarle los datos que deberemos poseer sobre:
 - ✓ Servidor de la base de datos: **localhost** es por defecto.
 - ✓ Nombre de la base de datos: **moodle2** nombre con el que hemos denominado nuestra base de datos.
 - ✓ Usuario de la base de datos: **moodleuser2**
 - ✓ Contraseña de la base de datos: **moodle**
- En este paso, Moodle lo que hace es comprobar que nuestro servidor web Lighttpd cumple con los requerimientos mínimos para poder arrancar el Moodle.
- Una vez comprobado que cumplimos con los requerimientos mínimos para su funcionamiento, procedemos a abrir el archivo config.php para comprobar si el script si ha sido correctamente creado.

Este es el código fuente config.php que ha sido correctamente creado:

```
<?php  /// Moodle Configuration File ^M
^M
unset($CFG);^M
^M
$CFG = new stdClass();^M
$CFG->dbtype      = 'postgres7';^M
$CFG->dbhost      = 'user=\'moodleuser2\' password=\'moodle\'
dbname=\'moodle2\'';^M
$CFG->dbpersist   = false;^M
$CFG->prefix      = 'mdl_';^M
^M
$CFG->wwwroot     = 'http://192.168.1.2:8080/moodle';^M
$CFG->dirroot     = '/mnt/hicham/mimie/www/moodle';^M
$CFG->dataroot    = '/mnt/hicham/mimie/moodledata';^M
$CFG->admin       = 'admin';^M
^M
$CFG->directorypermissions = 00777; // try 02777 on a server in Safe
Mode^M
^M
$CFG->passwordsaltmain = 'Ng;o, DLo`XMfbMd2iQ/M:UX/MQX';^M
^M
require_once("$CFG->dirroot/lib/setup.php");^M
// MAKE SURE WHEN YOU EDIT THIS FILE THAT THERE ARE NO SPACES, BLANK
LINES,^M
// RETURNS, OR ANYTHING ELSE AFTER THE TWO CHARACTERS ON THE NEXT
LINE.^M
?>
```

Una vez que el archivo `config.php` ha sido correctamente creado en los pasos anteriores, le damos a siguiente para seguir la instalación en este directorio de `.../moodle/install.php`, pero nos encontramos con otra página llamada "administración", que por lo visto es para continuar con el resto de la configuración. Cuando accedemos por la primera vez a la página de administración, nos mostrara un acuerdo de licencia.



Figura 29: Página de condiciones de uso

Con el cual debemos estar de acuerdo antes de continuar con la instalación. En este momento, Moodle empezará configurando la base de datos y creando las tablas para guardar los datos. En primer lugar, se crean las principales tablas de la base de datos. Debería ver una serie de sentencias SQL seguidas por mensajes de color verde parecidas a éstas.

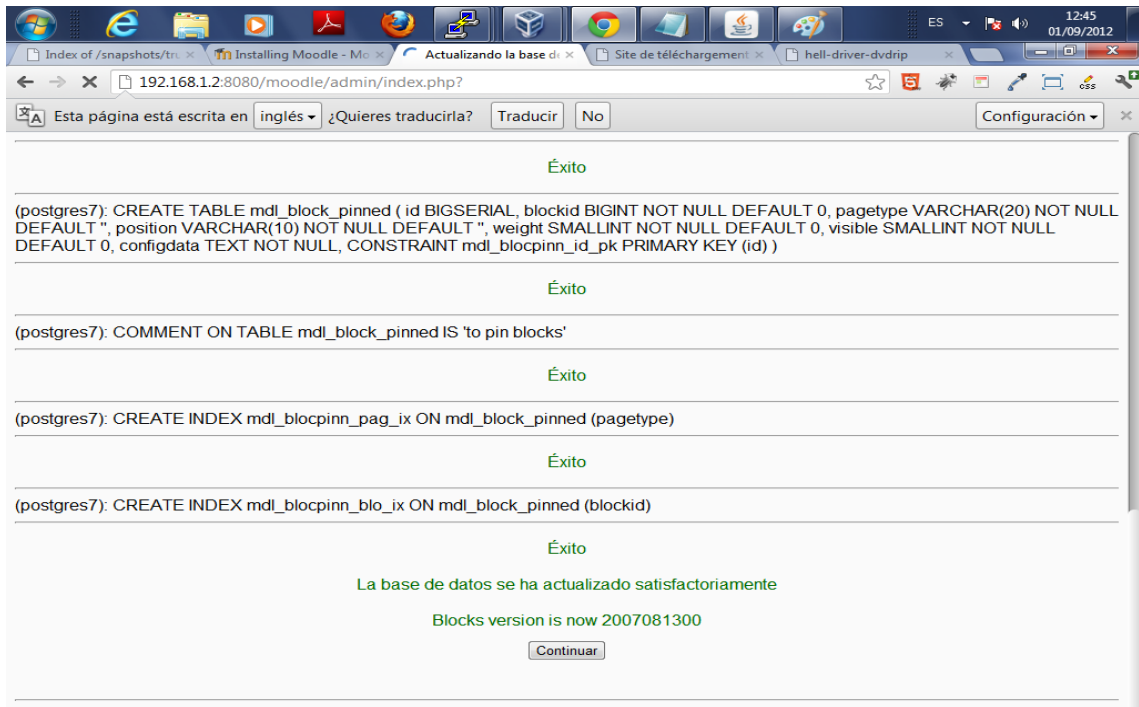


Figura 30: Creando tablas para almacenar datos

Cuando el programa de instalación acaba de crear las tablas, llega el momento de crear el usuario administrador de la aplicación, donde necesitaremos meter, por lo menos, los datos que están señalados con un asterisco rojo, ya que son de obligado relleno.

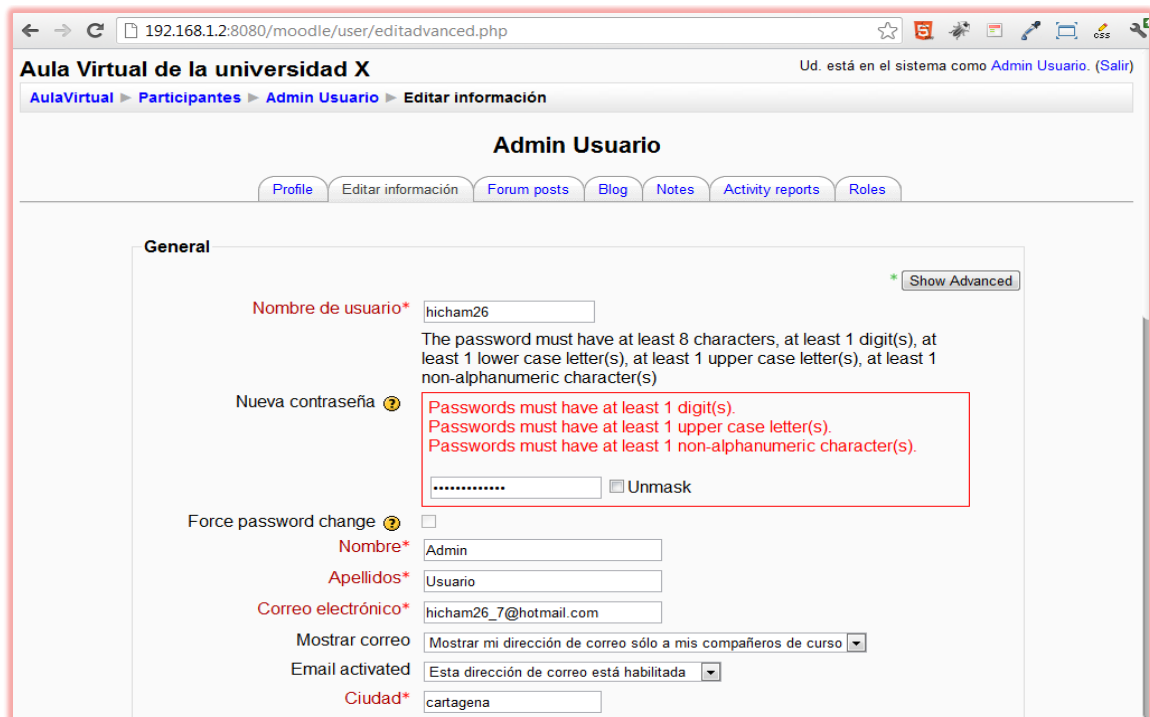


Figura 31: Rellenando el formulario del administrador

Finalmente terminamos la instalación, configurando la información general del sitio Moodle y la página principal, rellenando un formulario tal como nombre completo, corto de la página y una descripción de la página principal.

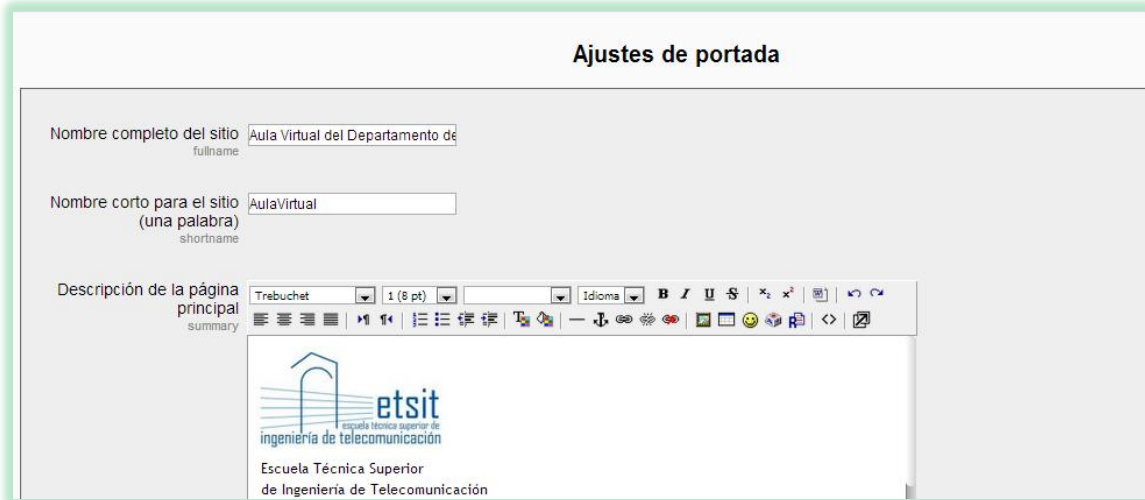


Figura 32: Ajustes de portada

Una vez guardamos la configuración anterior, podemos volver a la página principal. Esta contiene dos fotos de la UPCT una al centro y la otra con una serie de enlaces que aparece en la parte izquierda de la pagina, estos enlaces solo son visibles para el administrador que se encarga por ejemplo de crear o borrar cursos, crear cuentas del alumno.....etc.

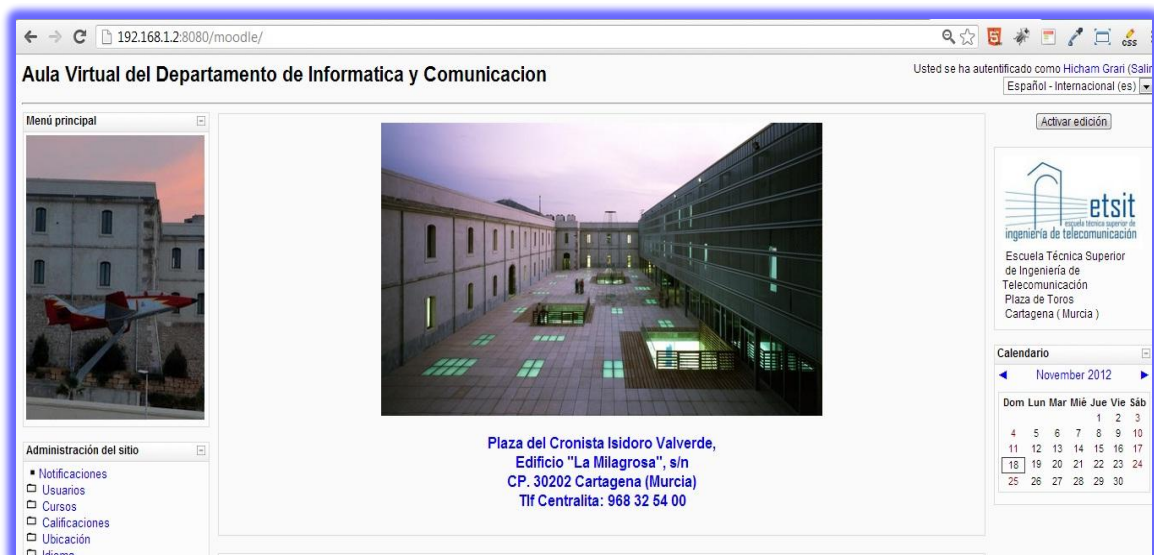
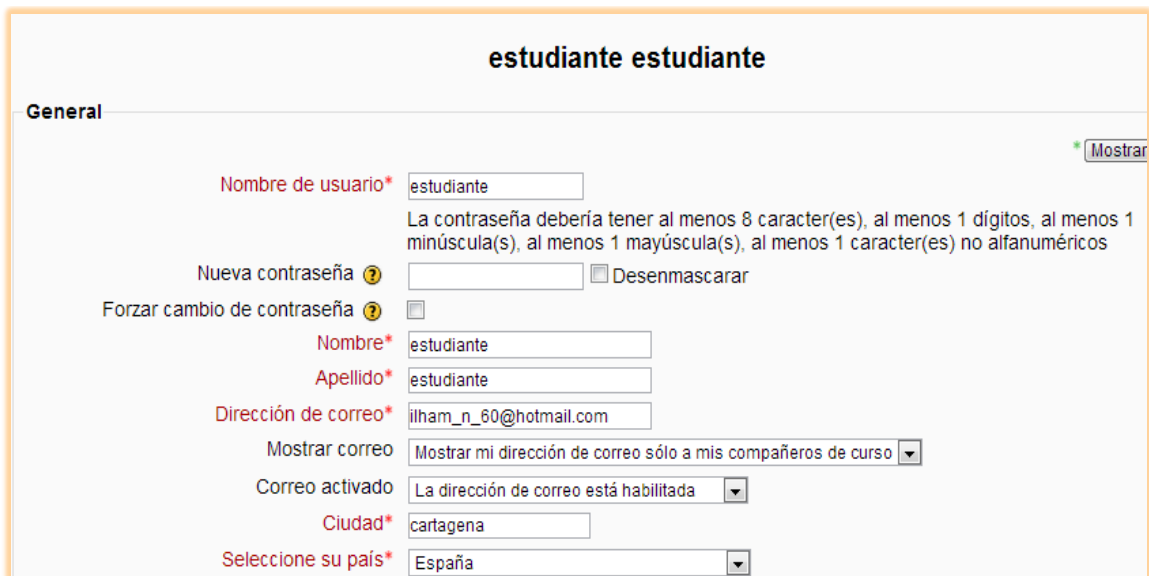


Figura 33: Portada principal de la cuenta del administrador

8.3.3 Creación/registro de usuarios

Para que el alumno que está matriculado y aún no pertenece a la red OpenMoodle, pueda inscribirse esperando que el profesor le mande la contraseña de acceso al sitio para que pueda realizar cualquier acción sobre Moodle (consultar asignaturas, descargar apuntes, escribir correos al profesor). Por esto es necesario que el administrador (en este caso es el profesor o bien el dueño de la red OpenMoodle) inscribe manualmente a su alumnos, si están matriculados dichas asignaturas que le pertenecen, y luego otorgándoles roles de estudiantes.

Para registrar a un estudiante, vamos a la opción "Usuarios" > "Cuentas" > "Agregar usuario".



The screenshot shows the Moodle user registration form titled "estudiante estudiante". The form is under the "General" tab. It includes the following fields and options:

- Nombre de usuario***: Input field with "estudiante".
- Nueva contraseña ?**: Input field with a "Desenmascarar" checkbox.
- Forzar cambio de contraseña ?**: Input field.
- Nombre***: Input field with "estudiante".
- Apellido***: Input field with "estudiante".
- Dirección de correo***: Input field with "ilham_n_60@hotmail.com".
- Mostrar correo**: Dropdown menu with "Mostrar mi dirección de correo sólo a mis compañeros de curso".
- Correo activado**: Dropdown menu with "La dirección de correo está habilitada".
- Ciudad***: Input field with "cartagena".
- Seleccione su país***: Dropdown menu with "España".

Figura 34: Inscribir a un estudiante



The screenshot shows the Moodle user list table titled "Agregar usuario". The table has the following columns: Nombre / Apellido, Dirección de correo, Ciudad, País, Último acceso, and two action buttons (Editar and Borrar).

Nombre / Apellido	Dirección de correo	Ciudad	País	Último acceso		
Hicham Grari	hicham26_7@hotmail.com	cartagena	España	54 segundos	Editar	
estudiante estudiante	ilham_n_60@hotmail.com	cartagena	España	20 horas 40 minutos	Editar	Borrar

Figura 35: Lista de usuarios inscritos

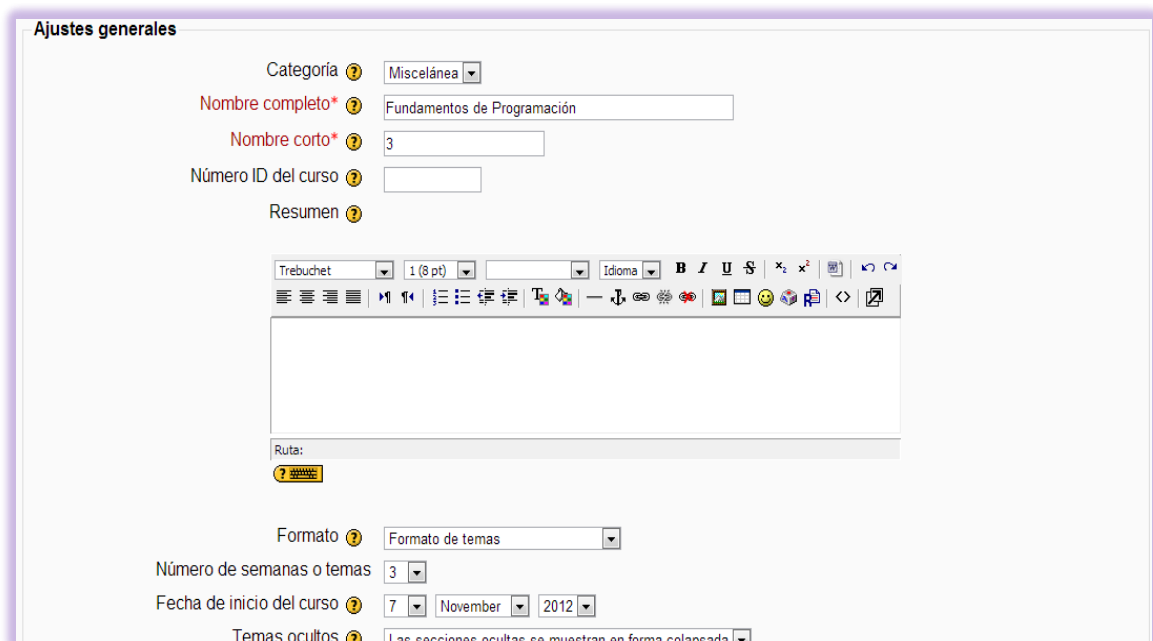
8.3.4 Creación de Cursos

Una vez registrados los estudiantes en la base de datos (PostgreSQL) de Moodle, se debe proceder a la planificación y organización de cursos a cada uno de ellos. Este es una tarea que se debe llevar a cabo el dueño del router wrt160nl (profesor) para poder satisfacer y facilitar a su alumnos metodología de enseñanza.

Antes de comenzar a repartir las signaturas a cada alumno, es necesario llevar a cabo una serie de pasos:

- ✓ Crear el curso.
- ✓ Configuración del curso.
- ✓ Luego, matricular a los estudiantes.

Crear un curso a través de la opción de menú Cursos → Agregar/editar cursos. Nos aparecerá una ventana de configuración del curso tal como la siguiente:



The screenshot shows the 'Ajustes generales' (General Settings) form for creating a course in Moodle. The form includes the following fields and options:

- Categoría**: Miscelánea (dropdown)
- Nombre completo***: Fundamentos de Programación (text input)
- Nombre corto***: 3 (text input)
- Número ID del curso**: (empty text input)
- Resumen**: (empty text area)
- Trebuchet**: (dropdown)
- 1 (8 pt)**: (dropdown)
- Idioma**: (dropdown)
- Formato**: Formato de temas (dropdown)
- Número de semanas o temas**: 3 (dropdown)
- Fecha de inicio del curso**: 7, November, 2012 (date picker)
- Temas ocultos**: Las secciones ocultas se muestran en forma colapsada (dropdown)

Figura 36: Captura de pantalla de cómo crear cursos

Una vez que guardamos la configuración del nuevo curso, nos aparecerá la ventana en la que podemos asignar a cada estudiante, las asignaturas que le corresponda.

Roles asignados localmente [Anular permisos](#)

Asignar roles en Curso: Fundamentos de la Señal ?

Roles	Descripción	Usuarios
Administrador	Normalmente los administradores pueden hacer cualquier cosa en el sitio, en todos los cursos.	0
Creador de curso	Los creadores de cursos pueden crear nuevos cursos y enseñar en ellos.	0
Profesor	Los profesores pueden realizar cualquier acción dentro de un curso, incluyendo cambiar actividades y calificar a los estudiantes.	0
Profesor sin permiso de edición	Los profesores sin permiso de edición pueden enseñar en los cursos y calificar a los estudiantes, pero no pueden modificar las actividades.	0
Estudiante	Los estudiantes tienen por lo general menos privilegios dentro de un curso.	1 estudiante estudiante
Invitado	Los invitados tienen privilegios mínimos y normalmente no están autorizados para escribir.	0

este estudiante tiene permiso para acceder al curso de fund.de señal

[Haga clic aquí para entrar a su curso](#)

Figura 37: Asignar curso a un alumno

Antes de empezar a enlazar archivos (apuntes, convocatorias, notas,.....etc), hay que elegir un formato donde se guardaran estos archivos, en nuestro caso hemos elegido el formato temático donde los contenidos serán repartidos en cada sección (teoría, practica, exámenes.....).

Una pantalla de ejemplo de este último apartado en la figura siguiente:

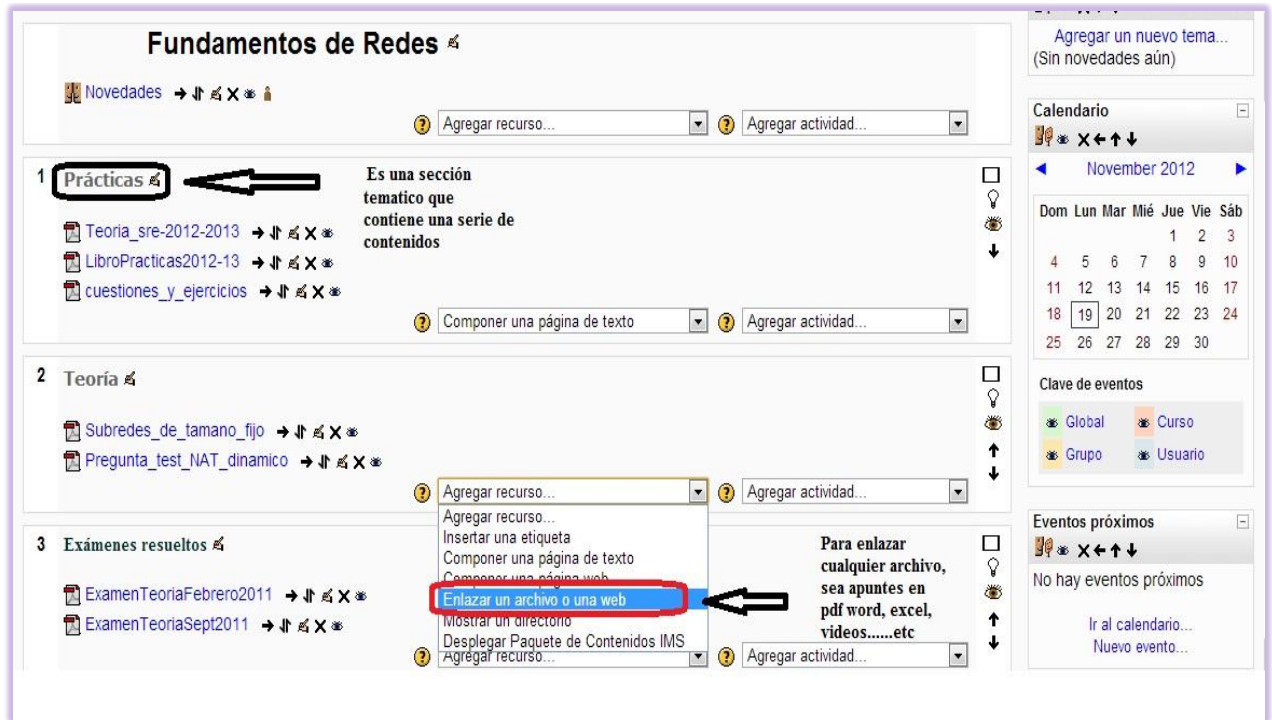


Figura 38: Agregar Archivos en Sección Temáticos

Y por último, si todo el proceso de agregar usuarios y cursos finaliza correctamente, el alumno puede ahora disfrutar de su cuenta acudiendo con su contraseña a la página Principal de Moodle.



Figura 39: Pantalla de login

9. Conclusiones

Se han alcanzado todos los objetivos de este proyecto, es decir, se ha incorporado en un router inalámbrico una plataforma de enseñanza que permite a alumnos pertenecientes a países subdesarrollados de recoger toda información y documentación que el docente imparta durante un curso mediante el uso de un ordenador con conexión wifi pero sin acceso a Internet. Para ello se ha integrado en el router wrt160nl varias aplicaciones que permiten el funcionamiento del aula de enseñanza Moodle:

1. Una llave de almacenamiento USB, en nuestro caso hemos usado de 4 GB.
2. Instalado un sistema operativo de código abierto para poder personalizar el router y ajustarlo a las necesidades del proyecto
3. Instalado PHP versión 4.1.0 y el servidor web ligero lighttpd.
4. Instalado la base de datos PostgreSQL para dar soporte a los cursos virtuales
5. Instalado la aplicación web Moodle (aula virtual) a la que se accede por medio de un navegador web.

10. Bibliografía

www.wrt160nl.org
www.es.wikipedia.org
www.openwrt.org
www.postgresql.es
www.moodle.org