



Treball fi de carrera

**ENGINYERIA TÈCNICA EN
INFORMÀTICA DE SISTEMES**

**Facultat de Matemàtiques
Universitat de Barcelona**

Puesta en marcha de un WISP

YURIY RUSNAK

Director: Sergio Escalera

Realitzat a: Departament de
Matemàtica
Aplicada i Anàlisi.
UB

Barcelona, 20 de enero de 2012

Índice general

1. Resumen	3
1.1. Resumen	3
1.2. Resum.	3
1.3. Abstract	3
2. Introducción.	4
3. Análisis	
3.1. Estudio del software	7
3.2. Estudio del hardware.	8
3.3. Estudio herramientas de monitoring y estadística.	9
3.3.1. SNMP Y CACTI.	9
3.3.2. Servidor RSysLog y SysLogAnalyzer.	10
4. Diseño y implementación.	
4.1. Diseñando nuestra red	11
4.1.1 Direccionamiento en la red.	11
4.1.2 PPPoE vs HotSpot.	13
4.2. Puesta en marcha del DataCenter	14
4.2.1 Servidor Linux.	14
4.2.2 Instalación RADIUS y conexión RADIUS-MySQL . . .	15
4.2.3 Instalación Daloradius.	17
4.2.4 Instalación del servidor RSysLog y SysLogAnalyzer. .	20
4.2.5 Instalación SMTP y Cacti	25
4.3 Instalación Router Os Mikrotik	30
4.4 Validación.	35
5. Conclusiones	38
6. Bibliografía.	39
7.1. Anexo 1 - Contenido del CD.	40
7.2. Anexo 2 - QoS(las colas de trafico)	41
7.3. Anexo 3 - Mapa de Red	44
7.4. Anexo 4 Implementación de un Punto de Acceso.	45

1.1 RESUMEN

En el presente trabajo se analizará la implementación de una red con Mikrotik Router OS. Vamos a diseñar, dimensionar y desplegar la infraestructura de un proveedor de internet inalámbrico. Se realizó un análisis de estructura de la red. Se diseñó una implementación del servidor DataCenter (Debian+ entorno LAMP). También se realizó un balanceo de carga para optimizar el tráfico que fluye desde y hacía Internet. Se implementaron políticas de colas de tráfico, autenticación de usuario mediante un servidor Radius. La interconectividad de todos estos servicios se logra por medio del uso del sistema operativo Mikrotik RouterOS, con el cual se logro tener un altísimo rendimiento, seguridad y estabilidad en la red sin comprometer altísimos costos de implementación, comparado con otro tipo de sistemas.

1.2 RESUM

En el present treball s'analitzarà la implementació d'una xarxa amb Mikrotik Router OS. Anem a dissenyar, dimensionar i desplegar la infraestructura d'un proveïdor d'internet sense fils. Es va realitzar una anàlisi d'estructura de la xarxa. Es va dissenyar una implementació del servidor DataCenter (Debian + entorn LAMP). També es va realitzar un balanceig de càrrega per optimitzar el trànsit que flueix des de i cap a Internet. Es van implementar polítiques de cues de trànsit, autenticació d'usuari mitjançant un servidor Radius. La connectivitat de tots aquests serveis s'aconsegueix mitjançant l'ús del sistema operatiu Mikrotik RouterOS, amb el qual es assoliment tenir un altíssim rendiment, seguretat i estabilitat a la xarxa sense comprometre altíssims costos d'implementació, comparat amb un altre tipus de sistemes.

1.3 ABSTRACT

In this paper we analyze the implementation of a Mikrotik Router OS network. We will design, measure and deploy the infrastructure of a wireless internet provider(WISP). We performed an analysis of network structure. Deployment was designed DataCenter Server (Debian + LAMP). We also performed a load balancing to optimize traffic flows to and from Internet. Were implemented traffic queues, user authentication by Radius server. The interconnection of all these services is accomplished through the use Mikrotik RouterOS operating system. With this Operation System we have a high performance, security and stability in the network without compromising high costs of implementation compared to other systems.

2. Introducción.

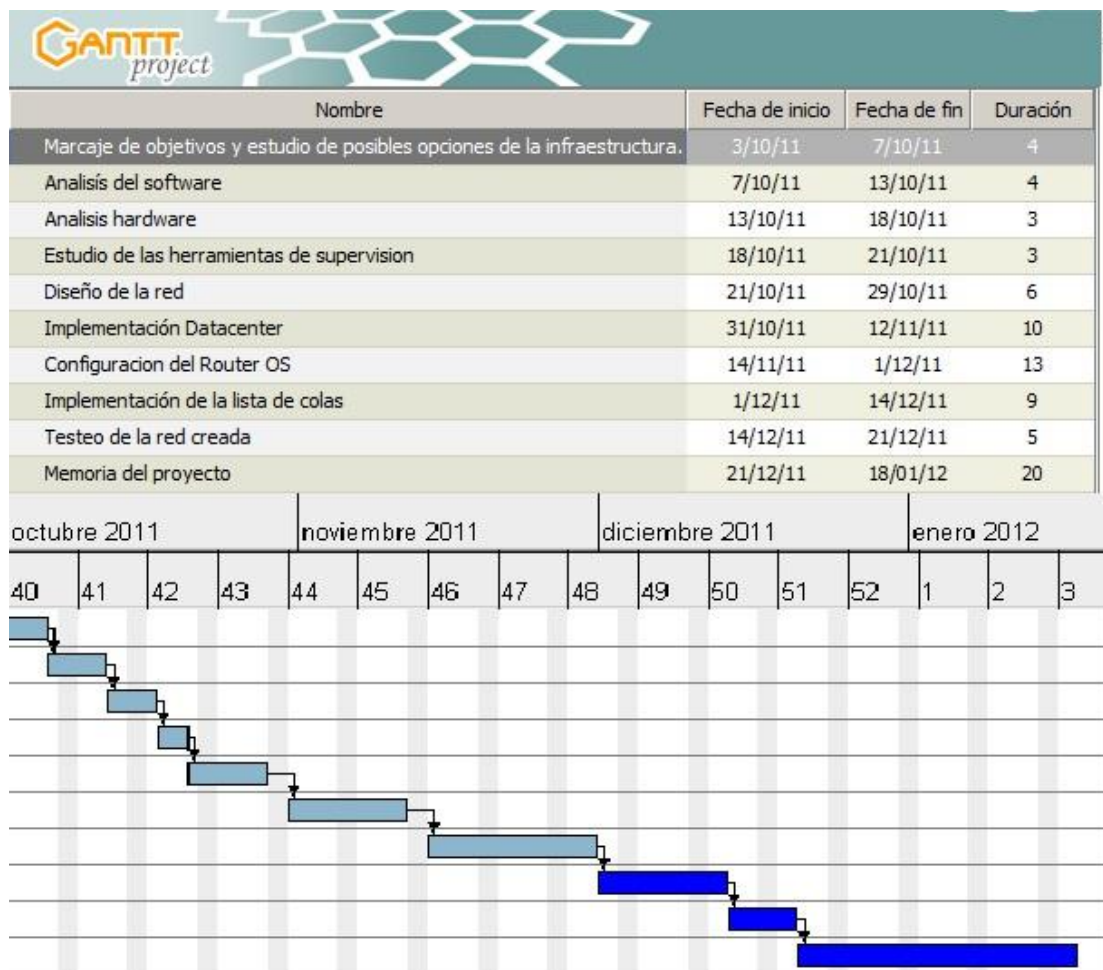
Cuando elegimos las aplicaciones a instalar se han tenido en cuenta valores como la estabilidad, flexibilidad y seguridad.

A parte de la elección de los componentes de comunicaciones, también se ha implementado un servidor AAA basado en FreeRadius 2, que aúna seguridad (autenticación, autorización y accounting), además de la flexibilidad de tener la seguridad centralizada.

FreeRadius2 es uno de los servidores RADIUS mas usados, potentes y escalables, aunque no especialmente sencillo de configurar.

Como frontend hemos escogido DaloRadius, una solución que nos permitirá administrar usuarios, cuentas, grupos, NAS, realizar estadísticas, aplicar atributos/restricciones... en fin una solución completa de las necesidad de administración de nuestro WISP a través de una aplicación web.

La planificación temporal del nuestro proyecto la representamos mediante diagrama de Gantt utilizando la aplicación llamada Gantt Project.



El coste de este proyecto tenemos que calcular a partir de dos valores.

Primero contabilizamos los gastos de material:

1. Un equipo de sobremesa para el servidor "DataCenter" :

Ordenador con Intel Core 2 Duo 2,4 GHz, 2 Gb DDR2 ,
disco duro de 200 Gb y pantalla de 17 pulgadas valorado en 600 €.

2. Cuatro puntos de acceso :

Material	cantidad	precio	total
Mikrotik Routerboard RB433AH	4	180	720
Caja Exterior Mikrotik OTU	4	80	320
Ubiquiti Tarjeta miniPCI 200mW 5Ghz	12	60	720
Antena Direccional 5Ghz	6	70	420
Antenas Sectorial 120º	6	120	720
Total			2900

3. Un router que se puede programar como balanceador de carga, por ejemplo, un Router RB750GL valorado en 120 euros.

El precio del todo **material** utilizado es alrededor de **3620 euros**.

Por otra parte es el coste del personal.

Vamos a necesitar al menos dos profesionales:

El de Gestor del proyecto y el de Técnico/Programador.

El proyecto durará 77 días laborables. Supondremos que el jefe de proyecto dedicará una media de 3 horas semanales a la dirección de este proyecto, de modo que dedicará un total de 45 horas.

El Técnico/Programador dedicará una media de 2 horas diarias, lo que suma un total de 154 horas.

Gestor del proyecto	45 horas	5,6 días	200 euros por día	1120 euros
Técnico/Programador	154 horas	19,2 días	110 euros por día	2112 euros
El coste del personal				3232 euros

Sumando todos los valores se obtienen un total de **6852 euros** para desarrollo de este proyecto. Esta claro que el presupuesto es aproximado.

3. Análisis.

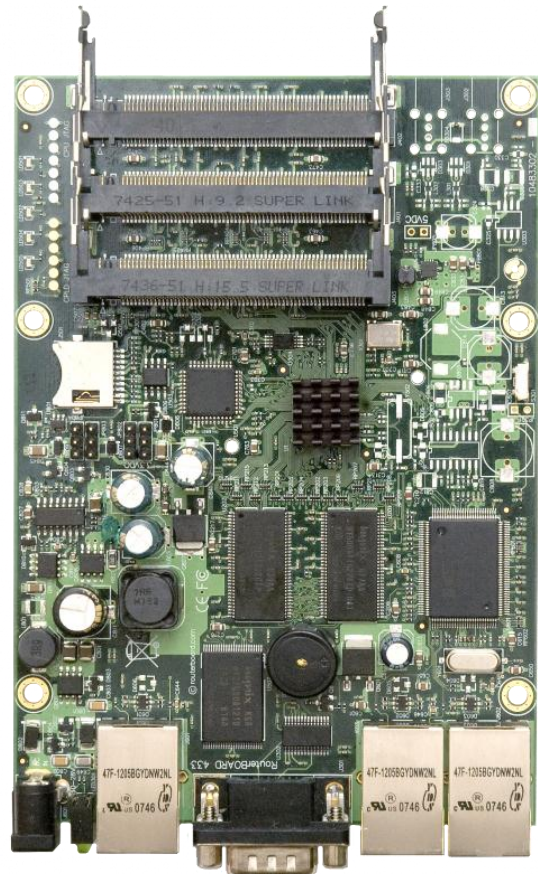
3.1 Estudio del hardware.

Comparando entre varios proveedores como **LIGOWAVE, LOBOMETRICS, MIKROTIK, MOTOROLA, ARCFLEX**, el equipamiento de red escogido para los puntos de acceso se basa en la conocida distribución RouterOS con hardware de Mikrotik actualizados a su versión 5.7, precisamente por la gran flexibilidad y potencia que confiere a la hora de establecer cualquier enlace de red y por su bajo coste.

Después de valorar varias opciones, nos decantamos por el equipo RB433AH de la firma MikroTik.

A continuación detallamos un resumen de sus características técnicas, así como una imagen del aspecto del mismo:

Velocidad CPU	680MHz
RAM	128MB
Arquitectura	MIPS-BE
Int. Fast Ethernet	3
Ranuras MiniPCI	3
Ranuras de memoria	1
Tipo de memoria	microSD
Power Jack	10-28V
Rango PoE	10-28V
Monitor de Volt.	Yes
Rango de temp.	-30C a +60C



En la parte cliente se ha escogido al reconocido fabricante Ubiquiti Networks por considerar que sus productos ofrecen calidad, potencia y flexibilidad suficiente para satisfacer las necesidades actuales y futuras que puedan surgir.

3.2 Estudio software.

Vamos a centralizar todos los servicios en un único servidor, así que escogemos la distribución Debian de Linux, por tener un buen compromiso entre fiabilidad, escalabilidad y soporte.

También se ha pensado en una gestión de las cuentas de usuario – clientes – facturación,... flexible y accesible a través de la nube.

Para la administración y monitorización de nuestro sistema se ha recurrido a una plataforma opensource: entorno LAMP basado en la distro Linux Debian, Apache2, MySQL y PHP...

RADIUS (del inglés *Remote Authentication Dial-In User Server*). Es un protocolo **AAA** (autenticación, autorización y accounting) para aplicaciones de acceso a la red. Utiliza el puerto 1812 UDP para establecer sus conexiones.

Cuando se realiza la conexión con un ISP, se envía una información que generalmente es un nombre de usuario y una contraseña. Esta información se transfiere a un dispositivo Network Access Server (NAS) sobre el protocolo PPP, quien a su vez envía la petición a un servidor RADIUS . El servidor RADIUS verifica las credenciales utilizando esquemas de autenticación como PAP, CHAP o EAP. Si es aceptado, el servidor autorizará el acceso al sistema del ISP y le asigna los recursos de red como una dirección IP, y otros parámetros como L2TP, etc.

Una de las características más importantes del protocolo RADIUS es su capacidad de “accounting” o manejar sesiones, notificando cuando comienza y termina una conexión, así que al usuario se le podrá determinar su consumo y facturar en consecuencia; los datos se pueden utilizar con propósitos estadísticos.

RADIUS fue desarrollado originalmente por Livingston Enterprises para la serie PortMaster de sus Servidores de Acceso a la Red(NAS), más tarde se publicó como RFC 2138 y RFC 2139. Actualmente existen muchos servidores RADIUS, tanto comerciales como de código abierto. Las prestaciones pueden variar, pero la mayoría pueden gestionar los usuarios en archivos de texto, servidores LDAP, bases de datos varias, etc. Los servidores Proxy RADIUS se utilizan para una administración centralizada y pueden reescribir paquetes RADIUS al vuelo (por razones de seguridad, o hacer conversiones entre dialectos de diferentes fabricantes gracias a los diccionarios)....

Daloradius es una avanzada aplicación HTTP que sirve de sistema de administración para RADIUS, y está diseñada para ser utilizada para administrar hotspots, WISP's y uso general para la autenticación de Proveedores de Acceso a Internet (ISP). Incluye gestión de usuarios, reportes gráficos, contabilidad, motor de facturación e integración con Google Maps para Geolocalización.

3.3 Análisis herramientas de monitoring y estadística.

3.3.1 SNMP y Cacti.

El Protocolo Simple de Administración de Red o SNMP (del inglés Simple Network Management Protocol) es un protocolo de la capa de aplicación que facilita el intercambio de información de administración entre dispositivos de red. Es parte de la familia de protocolos TCP/IP; utiliza los puertos UDP 161 y 162 y está recogido en los RFC's 1157 y 3410. SNMP permite a los administradores supervisar el funcionamiento de la red, buscar y resolver sus problemas, y planear su crecimiento.

Las versiones de SNMP más utilizadas son SNMP versión 1 (SNMPv1) y SNMP versión 2 (SNMPv2).

SNMP en su última versión (SNMPv3) posee cambios significativos con relación a sus predecesores, sobre todo en aspectos de seguridad, sin embargo no ha sido mayoritariamente aceptado en la industria.

Una vez que hemos decidido usar SNMP para supervisar nuestra red, decidimos instalar CACTI como herramienta.

¿Y que es Cacti? Cacti es una completa solución para la generación de gráficos en red, diseñada para aprovechar el poder de almacenamiento y la funcionalidad para hacer gráficos a las aplicaciones RRDtool. Esta herramienta, desarrollada en PHP, provee un "pooler" ágil, plantillas de gráficos avanzadas, múltiples métodos para la recopilación de datos, y manejo de usuarios. Tiene una interfaz de usuario fácil de usar, que resulta conveniente para instalaciones del tamaño de una LAN, así como también para redes complejas con cientos de dispositivos.

3.3.2 Servidor RSysLog y SysLogAnalyzer.

Rsyslog (Demonio de logs de sistema): es una herramienta que se encarga de visualizar los eventos o logs del sistema y guardar los mensajes enviados por los diferentes servicios y por el kernel, captura los mensajes que envía el sistema y los guarda en diferentes archivos generalmente en el directorio /var/log.

Puede ser configurado para que los mensajes se envíen a otro servidor, ser alertado mediante correo electrónico, etc. En su archivo de configuración /etc/rsyslog.conf se puede observar las rutas de los diferentes logs que guarda del sistema.

SysLogAnalyzer (antiguamente phpLogCon) es una aplicación desarrollada en php que permite visualizar, analizar, filtrar, realizar estadísticas, gráficas y muchas cosas más. Puede interpretar los datos directamente del log del sistema en formato de texto, aunque para explotar su potencial, lo suyo es acceder a los datos almacenados en una base de datos.

4. Diseño y Validación.

4.1 Diseñando nuestra red.

De manera muy esquemática, nuestra red está dibujada en el **Anexo 2**.

4.1.1 Direccionamiento en nuestra red.

Para las conexiones con el cliente final, se ha optado por establecer una conexión del tipo PPPoE, y aunque tiene los conocidos problemas de seguridad propios de PPPoE (“man in the middle” o suplantación de identidad), se ha considerado el canal suficientemente seguro puesto que se usamos la encriptación WPA2 en una red funcionando a 5Ghz.

En la banda de 2.4Ghz, se ha aprovechado la infraestructura de la red para desplegar una serie de “hotspots”, desde dónde podemos ofrecer un servicio libre, de pago, patrocinado con anuncios, etc... y directamente accesible desde cualquier dispositivo con tecnología Wi-Fi, al estilo de los ofrecidos por hoteles o bibliotecas.

Como hemos tenido la oportunidad de diseñar nuestra red con total libertad, usando libremente los rangos privados disponibles, hemos intentado hacer un direccionamiento lo más lógico y práctico posible, y que entre otras cosas nos permita hacer supernetting.

Así pues, en nuestro AP-Core, hemos usado 3 interfaces inalámbricas diferentes, una para cada enlace, así como 2 interfaces Ethernet, para balancear la carga entre 2 líneas VDSL2 de 30Mb.

Para el direccionamiento usado en este nivel, hemos optado por usar 3 redes diferentes con un direccionamiento “classless”. Hemos escogido una máscara /29 (255.255.255.248), porque por una parte podemos usar hasta un total de 32 redes en el rango 10.0.0.x lo que nos permitirá crecer de manera considerable si romper la lógica de nuestro diseño, y por otra parte, permitimos hasta un total de 6 hosts en cada enlace (no solo 2 como sería una punto a punto), por lo que deja la puerta abierta a posibles imprevistos. Con estas premisas, usaremos el siguiente direccionamiento:

1ª red:

dirección de red: 10.0.0.0 / 29

Rango de direcciones: 10.0.0.1-10.0.0.6 /29

dirección de broadcast: 10.0.0.7 / 29

2ª red:

dirección de red: 10.0.0.8 / 29

Rango de direcciones: 10.0.0.9-10.0.0.14 /29

dirección de broadcast: 10.0.0.15 / 29

3ª red:

dirección de red: 10.0.0.16 / 29

Rango de direcciones: 10.0.0.17-10.0.0.22 /29

dirección de broadcast: 10.0.0.23 / 29

Por último decir que hemos usado la primera dirección de cada rango para las interfaces del dispositivo AP-Core, y la segunda dirección disponible para cada uno de los AP-Slave de cada red.

Por otra parte, para hemos calculado una máscara /16 (255.255.0.0) para asignar direcciones a nuestros clientes, de manera que tenemos $(2^{16})-2$ direcciones disponibles en cada nodo para asignar y hemos usado el segundo octeto para que coincida con el número de AP, de manera que tenemos el siguiente direccionamiento:

AP Slave1:

dirección de red: 10.1.0.0 /16

Rango de direcciones: 10.1.0.1-10.1.255.254 /16

dirección de broadcast: 10.1.255.255 /16

AP Slave2:

dirección de red: 10.2.0.0 /16

Rango de direcciones: 10.2.0.1-10.2.255.254 /16

dirección de broadcast: 10.2.255.255 /16

AP Slave3:

dirección de red: 10.3.0.0 /16

Rango de direcciones: 10.3.0.1-10.3.255.254 /16

dirección de broadcast: 10.3.255.255 /16

Y hemos usado la primera dirección disponible de cada red para el AP correspondiente.

Una vez diseñado por completo el direccionamiento, vamos a ver el hardware escogido:

4.1.2 PPPoE vs HotSpot.

Hemos escogido PPPoE para proporcionar internet a nuestros clientes.

PPPoE (Point-to-Point Protocol over Ethernet o Protocolo Punto a Punto sobre Ethernet) es un protocolo de red para la encapsulación PPP sobre una capa de Ethernet. Es utilizada mayoritariamente para proveer conexión de banda ancha mediante servicios de cable módem y xDSL. Este ofrece las ventajas del protocolo PPP como son la autenticación, cifrado, mantención y compresión.

En esencia, es un protocolo túnel, que permite implementar una capa IP sobre una conexión entre dos puertos Ethernet, pero con las características de software del protocolo PPP, por lo que es utilizado para virtualmente "marcar" a otra máquina dentro de la red Ethernet, logrando una conexión "serial" con ella, con la que se pueden transferir paquetes IP, basado en las características del protocolo PPP.

Esto permite utilizar software tradicional basado en PPP para manejar una conexión que no puede usarse en líneas seriales pero con paquetes orientados a redes locales como Ethernet para proveer una conexión clásica con autenticación para cuentas de acceso a Internet. Además, las direcciones IP en el otro lado de la conexión sólo se asignan cuando la conexión PPPoE es abierta, por lo que admite la reutilización de direcciones IP (direccionamiento dinámico).

El objetivo y funcionamiento de PPPoE es análogo al protocolo PPP sobre RTC con el que a finales de los 90 y bajo un stack tcp, se establecía un enlace ip punto a punto a través de la red telefonica conmutada (RTC), permitiendo utilizar por encima una serie de protocolos de nivel de aplicación tipo http, ftp, telnet, etc.

PPPoE fue desarrollado por UUNET, Redback y RouterWare. El protocolo está publicado en la RFC 2516.

Aunque somos conocedores de su principal problema de seguridad (la suplantación de la identidad del servidor con la consiguiente obtención de las credenciales de un cliente), consideramos el canal suficientemente seguro como para menospreciar las posibilidades de que esto ocurra.

También hemos planeado la instalación de un HotSpot en el punto de acceso AP Slave3.

Un HotSpot es una modalidad de acceso a internet a través de una red WiFi. Tiene la ventaja que no necesita ninguna configuración en el ordenador o terminal del cliente. Es el tipo de conexión que puede encontrarse en aeropuertos, bibliotecas, hoteles,...

Un HotSpot se controla mediante un portal cautivo dónde el cliente se autenticará y un software que permite controlar los accesos y usuarios que se conectan a internet, ancho de banda. En nuestro caso, lo vamos a conectar a nuestro servidor de autenticación RADIUS y lo administraremos con DaloRadius.

Cabe decir que los RouterBoard RB433 incluyen entre sus funcionalidades la gestión propia de un HotSpot, lo cual nos facilitará enormemente la faena.

4.2 Puesta en marcha del DataCenter

4.2.1 Servidor Linux.

La instalación se realiza en unos sencillos pasos:

Introducimos el cd de Debian y ajustamos el sistema (BIOS) si fuera necesario para que tras el POST inicie el arranque a través de la unidad de discos ópticos cd-dvd.

En el menú de Debian, escogemos la opción **Install**. Siguiendo el proceso, escogemos **idioma** (en nuestro caso Spanish), escogemos el **país** (España), tipo de teclado, **nombre de nuestro servidor**. Después nos preguntará la contraseña para el usuario **root**, y nos dará la oportunidad de crear otro usuario con permisos de seguridad estándar. Seguiremos avanzando por sencillos menús de configuración.

Un paso siempre importante en la instalación de Linux es la configuración de las particiones de disco, siendo muchas veces recomendable separar las carpetas **/home** (datos de usuario), **/usr** y **/var**(aplicaciones) y la carpeta temporal **/tmp**.

En este caso vamos a dejar que la instalación nos ayude a crear estas particiones, usando el valor recomendado.

El siguiente apartado es la configuración de la ubicación del servidor de replicación de la paquetería de Debian. Escogeremos el más cercano a nuestro país, en este caso **España**.

La instalación descargará las versiones más actualizadas del repositorio que le hayamos especificado.

Pasados unos instantes, saldrá un menú para personalizar las funciones del equipo. Marcaremos las opciones de **servidor www**, **servidor ssh** y **base de datos sql**, ya que sinó tendremos que instalarlo luego.

Seguimos los pasos hasta finalizar la instalación. Ya tenemos instalado nuestro servidor y está listo para acoger las funcionalidades de AAA, servidor de registro remoto, control SNMP, etc...

El siguiente paso es instalar FreeRadius2.

4.2.2 Instalación RADIUS y connection RADIUS-MySQL.

Empezamos instalando la paquetería LAMP...

```
apt-get install apache2
apt-get install php5 libapache2-mod-php5
apt-get install mysql-server
```

...unas librerías necesarias

```
apt-get install debhelper libltdl3-dev libpam0g-dev
libmysqlclient15-dev build-essential libgdbm-dev libldap2-dev
libsasl2-dev libiodbc2-dev libkrb5-dev snmp autotools-dev dpkg-dev
libperl-dev libtool dpkg-dev libpq-dev libsnmp-dev libssl-dev
```

y finalmente FreeRadius2

```
apt-get install freeradius freeradius-mysql
```

Ahora vamos a configurar nuestro servidor RADIUS para que realice la consulta de las credenciales a una base de datos alojada en nuestro servidor de SQL, ya que por defecto no viene configurado así. Lo primero es modificar el fichero

```
/etc/freeradius/radiusd.conf
```

y añadir el módulo de SQL, así que toca descomentar la línea:

```
$INCLUDE sql.conf
```

Añadir la configuración de la interconexión con el siguiente fichero:

```
$INCLUDE ${confdir}/sql.conf
```

Y cargar los módulos de los que vamos a responder peticiones:

```
modules {
    pap {
        encryption_scheme = crypt
    }
    chap {
        authtype = CHAP
    }
    mschap {
        authtype = MS-CHAP
        use_mppe = no
    }
}
```

Ya hemos acabado de modificar este fichero. Dese aquí se pueden variar muchos parámetros del comportamiento de FreeRadius2, aunque de momento es suficiente.

Vamos a crear ahora la base de datos para nuestro servidor RADIUS, dónde se almacenarán las cuentas, permisos, registro de actividad,...

```
mysql -u root -p
```

mysql nos pedirá el password para `root`, que hemos especificado durante la instalación de mysql.

```
CREATE DATABASE radius;

GRANT ALL ON radius.* TO radius@localhost IDENTIFIED BY "labserver";

exit;
```

Bien, en este momento hemos creado la base de datos llamada *radius* y también un usuario del mismo nombre y con el password "*labserver*" con acceso total a la base de datos.

Vamos ahora a configurar los parámetros de conexión:

```
vi /etc/freeradius/sql.conf
```

```
# Connection info:
server = "localhost"
login = "radius"
password = "labserver"
readclients = yes
```

Editamos también el siguiente fichero

```
vi /etc/freeradius/sites-available/default
```

```
authorize {
    preprocess
    chap
    mschap
    suffix
    eap
    sql
    noresetcounter
    dailycounter
    monthlycounter
    accessperiod
}
authenticate {
    Auth-Type PAP {
        pap
    }
    Auth-Type CHAP {
        chap
    }
    Auth-Type MS-CHAP {
        mschap
    }
}preacct {
    acct_unique
}accounting {
    detail
    sql
}session {
    sql
}post-auth {
    sql
}
```


4.2.3 Instalación Daloradius.

Vamos a instalarlo.

Primero de todo descargamos la última versión disponible y descomprimos los ficheros:

```
tar xvzf daloradius-0.9-9.tar.gz
```

Luego copiamos este archivo bajo el directorio de publicación del servidor web. Como estamos trabajando con una distro Debian, el directorio de publicación por defecto es:

```
/var/www
```

Así que copiamos los ficheros y subdirectorios comprimidos

```
cp daloradius-0.9-9 /var/www/daloradius -R
```

Ahora cambiamos los permisos y propiedades del directorio de daloradius.

```
chown www-data:www-data /var/www/daloradius/ -R
```

Cambiamos los permisos del archivo:

```
chmod 644 /var/www/daloradius/library/daloradius.conf.php
```

Para que Daloradius funcione correctamente, se requiere agregar algunas tablas más a la base de datos de MySQL. Hay un script en el directorio de DaloRadius, por lo que solo es necesario ejecutar:

```
mysql -u root -p radius < /var/www/daloradius/contrib/db/fr2-mysql-  
daloradius-and-freeradius.sql
```

y comprobamos que se hayan creado correctamente todas las tablas:

```
use radius;
```

```
show tables;
```

y obtendremos algo similar a esto:

```
mysql> use radius;  
Database changed  
mysql> show tables;
```

```

+-----+
| Tables_in_radius |
+-----+
| batch_history |
| billing_history |
| billing_merchant |
| billing_notifications_settings |
| billing_paypal |
| billing_plans |
| billing_plans_profiles |
| billing_rates |
| cui |
| dictionary |
| hotspots |
| invoice |
| invoice_items |
| invoice_status |
| invoice_type |
| nas |
| node |
| nodes_settings |
| operators |
| operators_acl |
| operators_acl_files |
| payment |
| payment_type |
| proxys |
| radacct |
| radcheck |
| radgroupcheck |
| radgroupreply |
| radhuntgroup |
| radippool |
| radpostauth |
| radreply |
| radusergroup |
| realms |
| userbillinfo |
| userinfo |
| wimax |
+-----+
37 rows in set (0.01 sec)

```

```
mysql>
```

Estas son las tablas que necesita DaloRadius en su versión 0.9-9 para funcionar. Finalmente editamos el archivo `daloradius.conf.php` para poner los valores de la conexión al servidor de la base de datos:

```
.....  
CONFIG_DB_ENGINE = mysql  
CONFIG_DB_HOST = 127.0.0.1  
CONFIG_DB_USER = radius  
CONFIG_DB_PASS = labserver  
CONFIG_DB_NAME = radius  
.....
```

Una vez esto, reiniciamos el servidor apache y freeradius2:

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

Con lo que ya tenemos listo nuestro servidor AAA y un frontend para gestionarlo a través de una interfaz www y con software opensource.

4.2.4 Instalación del servidor RSysLog y SysLogAnalyzer

Vamos a aprovechar para instalar un servidor rsyslog, de manera que tendremos centralizados y clasificados los registros de sucesos de nuestros equipos, y publicados en nuestro servidor apache, para poder, de un vistazo, conocer el estado de nuestra red.

Así pues, instalamos el siguiente paquete:

```
sudo apt-get install rsyslog-mysql
```

Durante la instalación, se nos pedirá el password de root de mysql (que no el de sistema) y creará el usuario `rsyslog` (nos pedirá una contraseña). Creará también la base de datos `Syslog` y creará en ella 2 tablas:

```
SystemEvents y SystemEventsProperties.
```

Ahora editaremos el fichero `/etc/rsyslog.conf` y activaremos el módulo de compatibilidad con mysql añadiendo o descomentando en caso de que ya exista la siguiente línea:

```
$ModLoad ommysql
```

Y activaremos la recepción en el Puerto TCP 514 y UDP 514 de los mensajes provenientes de nuestros dispositivos de red:

```
$ModLoad imtcp
$InputTCPServerRun 514
$ModLoad imudp
$UDPServerRun 514
```

Opcionalmente especificaremos la dirección o rango de direcciones desde dónde están autorizados a enviar mensajes a nuestro servidor :

```
$AllowedSender UDP, 127.0.0.1, ...
$AllowedSender TCP, 127.0.0.1, ...
```

Y finalmente vamos a indicar que todos los mensajes recibidos por nuestro servidor rsyslog se almacenen en la base de datos de mysql:

```
*.* :ommysql:127.0.0.1,Syslog,rsyslog,contraseña
```

Usamos en este caso la dirección de loopback 127.0.0.1 para indicar que el servidor de mysql está en la misma máquina que el servidor de rsyslog.

La contraseña será la que hemos especificado durante la instalación del paquete `rsyslog-mysql`.

En este punto, y tras reiniciar rsyslog con:

```
service rsyslog reload
```

Todos los mensajes se almacenarán en nuestra base de datos SQL. Instalaremos ahora el software SysLogAnalyzer, de Adiscon.

Como ya hemos hecho esto en el apartado anterior, vamos a realizar el último paso.

Descargamos la última versión disponible

```
wget http://download.adiscon.com/loganalyzer/loganalyzer-3.2.3.tar.gz
```

Descomprimos...

```
tar -xvzf loganalyzer-3.2.3.tar.gz
```

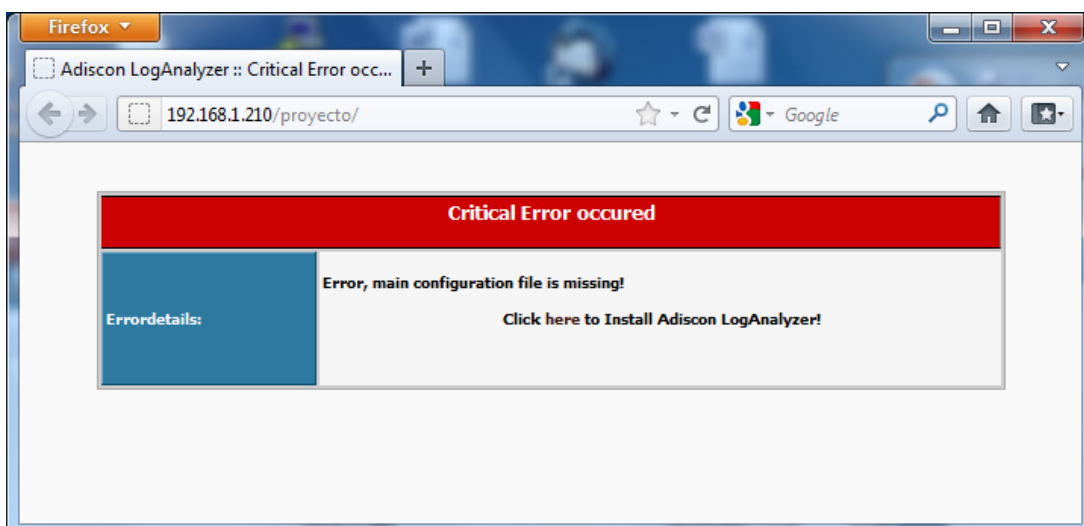
Publicamos el contenido del subdirectorio `/src` copiándolo a nuestra carpeta de publicación web. Igual que cuando instalamos DaloRadius, en Debian, la encontraremos en `/var/www`.

Ahora buscaremos un par de scripts que están ubicados en el subdirectorio `/contrib`. Llamados `configure.sh` y `secure.sh`. Los copiamos en la misma carpeta dónde hemos publicado la aplicación.

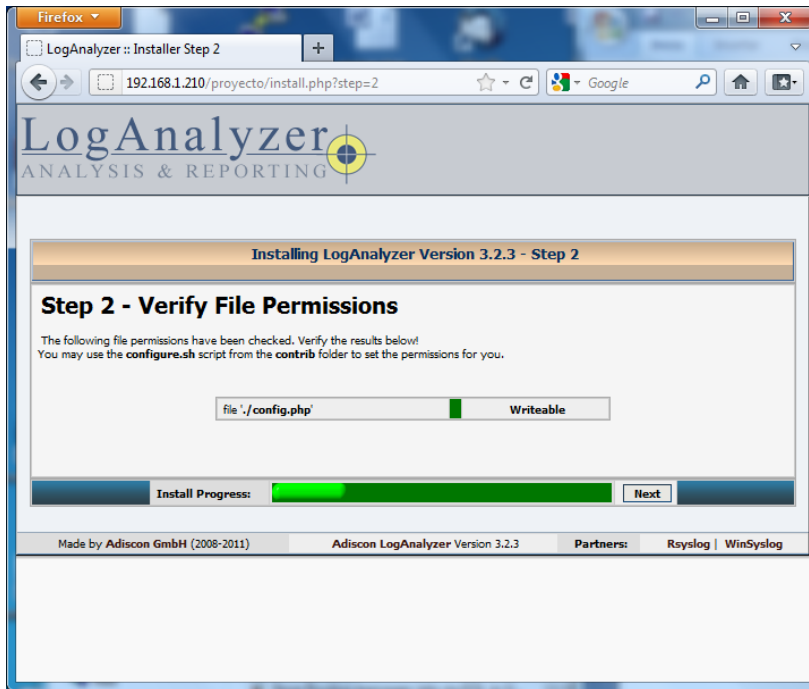
Ejecutamos desde esta ubicación `configure.sh` y creará un fichero en la misma carpeta con el nombre `config.php`.

Ya estamos listos para finalizar la instalación.

Lanzamos desde nuestro explorador `www` la aplicación y nos encontramos con la primera pantalla que nos indica que no existe ningún fichero de configuración.

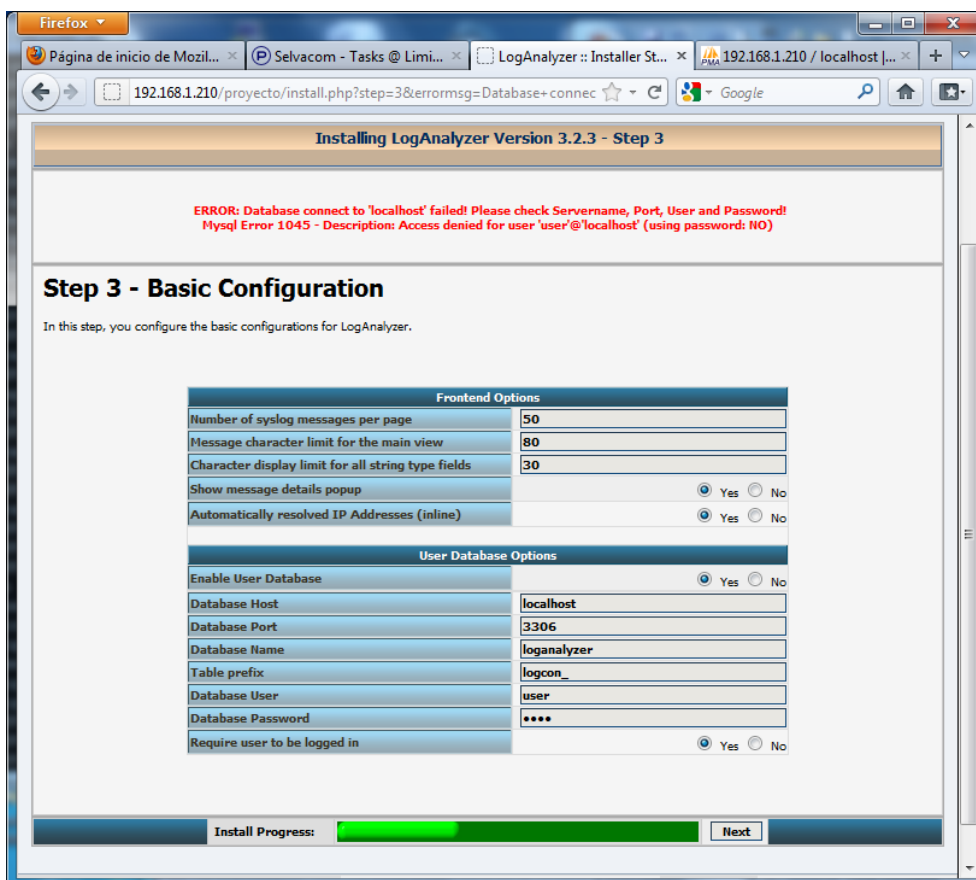


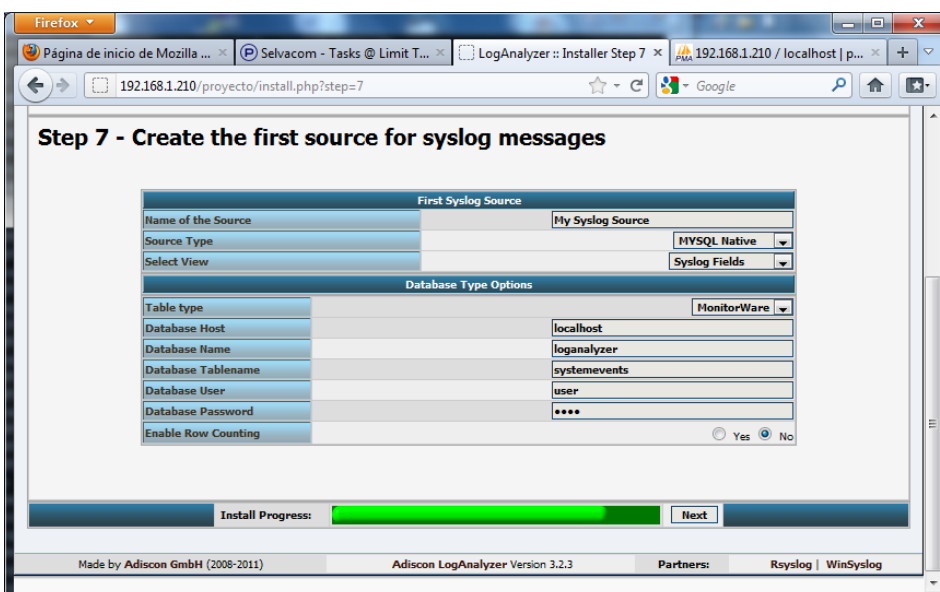
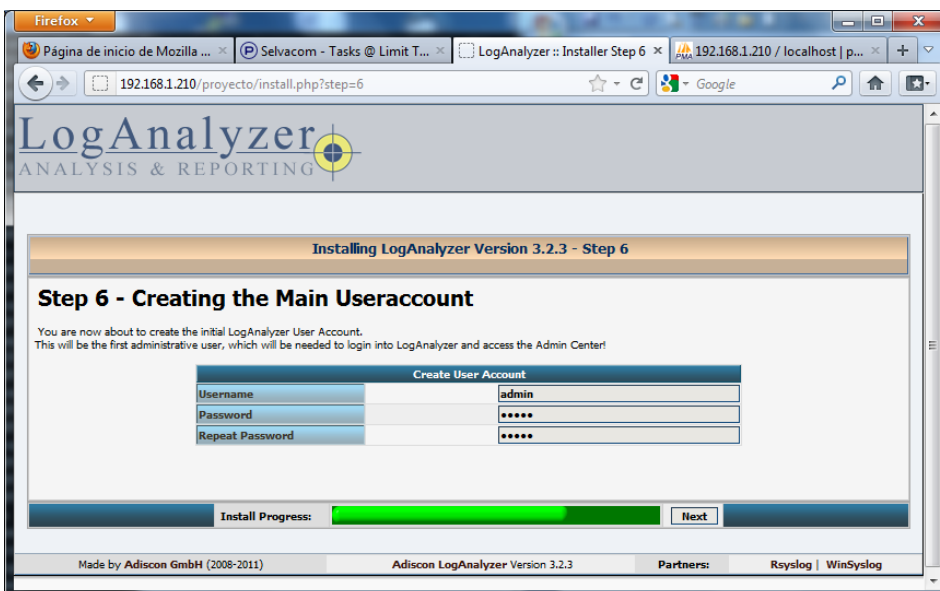
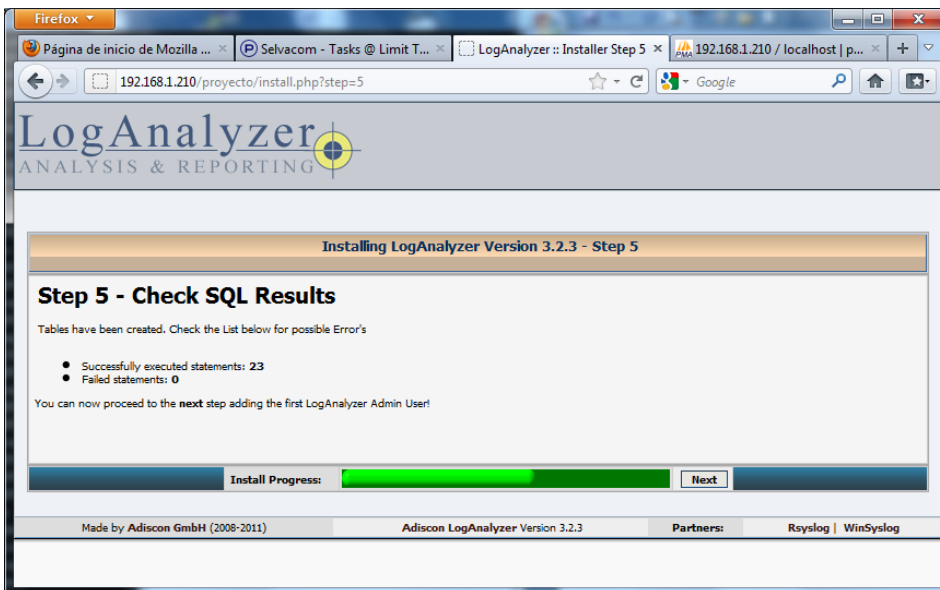
Continuamos en la siguiente pantalla:



En las siguiente pantallas iremos configurando la conexión con la base de datos, usuario dirección, etc...

Configuramos la conexión con la base datos





Como se puede ver, es una instalación extremadamente intuitiva.

Cuando accedamos y tras autenticarnos, veremos una pantalla con los eventos pintados en diferentes colores, con el siguiente aspecto:

The screenshot displays the LogAnalyzer web interface. At the top, there are navigation menus for Search, Show Events, Statistics, Help, and Admin Center. A search filter is set to 'Advanced Search' with a sample query 'facility:local0 severity:warning'. Below this is a table of 'Recent syslog messages' with 23 records found. The table has columns for Date, Facility, Severity, Host, Syslogtag, ProcessID, Messagetype, and Message. The messages are color-coded by severity: INFO (green), WARN (yellow), and ERR (red). The footer of the interface shows 'Made by Adiscon GmbH (2008-2011)', 'Adiscon LogAnalyzer Version 3.0.7', and 'Partners: Rsyslog | WinSyslog'. It also indicates 'Page rendered in: 0.0245 seconds | DB queries: 11 | GZIP enabled: yes | Script Timeout: 30 seconds'.

Date	Facility	Severity	Host	Syslogtag	ProcessID	Messagetype	Message
Today 13:33:00	CRON	INFO	040prx005	/usr/sbin/cron[30428]:		Syslog	(operator) CMD (/usr/libexec/save-entropy)
Today 13:30:00	CRON	INFO	040prx005	/usr/sbin/cron[30425]:		Syslog	(root) CMD (/usr/libexec/atrun)
Today 13:25:00	CRON	INFO	040prx005	/usr/sbin/cron[30410]:		Syslog	(root) CMD (/usr/libexec/atrun)
Today 13:22:00	CRON	INFO	040prx005	/usr/sbin/cron[30393]:		Syslog	(operator) CMD (/usr/libexec/save-entropy)
Today 13:20:00	CRON	INFO	040prx005	/usr/sbin/cron[26808]:		Syslog	(root) CMD (/usr/libexec/atrun)
Today 13:15:00	CRON	INFO	040prx005	/usr/sbin/cron[24503]:		Syslog	(root) CMD (/usr/libexec/atrun)
Today 13:11:00	CRON	INFO	040prx005	/usr/sbin/cron[24365]:		Syslog	(operator) CMD (/usr/libexec/save-entropy)
Today 13:10:00	CRON	INFO	040prx005	/usr/sbin/cron[24363]:		Syslog	(root) CMD (/usr/libexec/atrun)
Today 13:05:00	CRON	INFO	040prx005	/usr/sbin/cron[24357]:		Syslog	(root) CMD (/usr/libexec/atrun)
Today 13:04:27	SYSLOG	INFO	040prx005	rsyslogd:		Syslog	[origin software="rsyslogd" swVersion="3.22.3" x-pid="24353" x-info="http://www ...
Today 13:02:54	SYSLOG	ERR	040prx005	rsyslogd:		Syslog	Warning: backward compatibility layer added to following directive to rsyslog.co ...
Today 13:02:54	SYSLOG	ERR	040prx005	rsyslogd:		Syslog	Warning: backward compatibility layer added to following directive to rsyslog.co ...
Today 13:02:54	SYSLOG	ERR	040prx005	rsyslogd:		Syslog	Warning: backward compatibility layer added to following directive to rsyslog.co ...
Today 13:02:54	SYSLOG	ERR	040prx005	rsyslogd:		Syslog	WARNING: rsyslogd is running in compatibility mode. Automatically generated conf ...
Today 13:02:54	SYSLOG	INFO	040prx005	rsyslogd:		Syslog	[origin software="rsyslogd" swVersion="3.22.3" x-pid="24327" x-info="http://www ...
Today 13:01:25	SYSLOG	ERR	040prx005	rsyslogd:		Syslog	Warning: backward compatibility layer added to following directive to rsyslog.co ...
Today 13:01:25	SYSLOG	ERR	040prx005	rsyslogd:		Syslog	Warning: backward compatibility layer added to following directive to rsyslog.co ...
Today 13:01:25	SYSLOG	ERR	040prx005	rsyslogd:		Syslog	Warning: backward compatibility layer added to following directive to rsyslog.co ...
Today 13:01:25	SYSLOG	ERR	040prx005	rsyslogd:		Syslog	No UDP listen socket could successfully be initialized, message reception via UD ...
Today 13:01:25	SYSLOG	ERR	040prx005	rsyslogd:		Syslog	bind: Address already in use
Today 13:01:25	SYSLOG	ERR	040prx005	rsyslogd:		Syslog	bind: Address already in use
Today 13:01:25	SYSLOG	ERR	040prx005	rsyslogd:		Syslog	WARNING: rsyslogd is running in compatibility mode. Automatically generated conf ...
Today 13:01:25	SYSLOG	INFO	040prx005	rsyslogd:		Syslog	[origin software="rsyslogd" swVersion="3.22.3" x-pid="24291" x-info="http://www ...

Con lo que tendremos disponible en todo momento la información relevante para el buen funcionamiento de nuestra empresa, y con un formato mucho mas usable y práctico que el visor de sucesos por defecto.

Hemos conseguido tener todos los eventos centralizados y accesibles desde cualquier sitio. El siguiente paso es además monitorizar el uso, rendimiento, estado de equipos críticos. Aquí dónde los logs no son capaces de darnos mas información llega el momento de usar el protocolo SNMP.

4.2.6 SMTP y Cacti

Vamos a instalar Cacti:

```
sudo aptitude install cacti-spine
```

Esto iniciara la instalación de Cacti y hará unas preguntas rápidas.

Configurar la opción libphp-adodb seleccione OK para Continuar.

```
WARNING: include path for php has changed!

libphp-adodb is no longer installed in /usr/share/adodb. New
installation path is now /usr/share/php/adodb.

Please update your php.ini file. Maybe you must also change your
web-server configuraton.

<Ok>
```

Seleccionaremos el tipo de servidor web que tenemos instalado...

```
Which kind of web server should be used by cacti?

Select "None" if you would like to configure your webserver by hand.

Webserver type

Apache
Apache-SSL
Apache2
All
None

<Ok>
```

```
The cacti package must have a database installed and configured before
it can be used. This can be optionally handled with dbconfig-common.

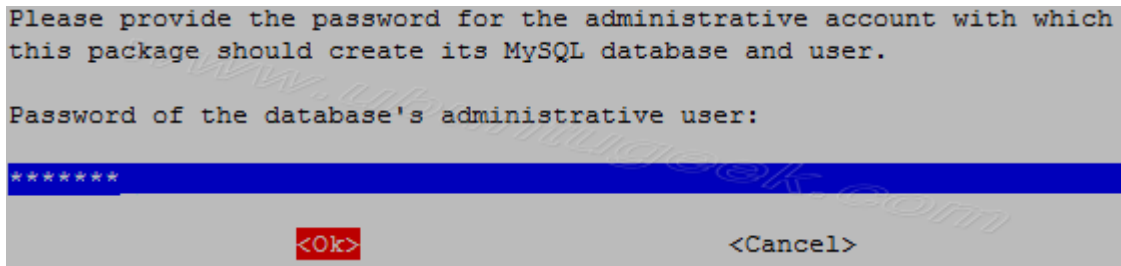
If you are an advanced database administrator and know that you want to
perform this configuration manually, or if your database has already
been installed and configured, you should refuse this option. Details
on what needs to be done should most likely be provided in
/usr/share/doc/cacti.

Otherwise, you should probably choose this option.

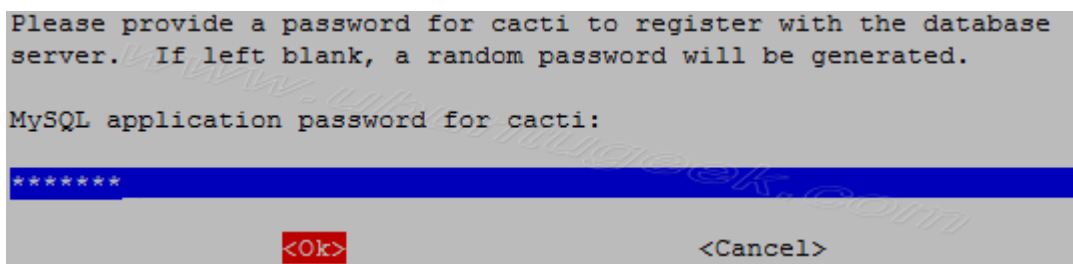
Configure database for cacti with dbconfig-common?

<Yes> <No>
```

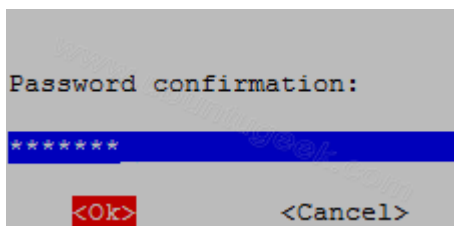
Indicamos en esta pantalla que queremos que el programa de instalación cree la base de datos,



Usuario...



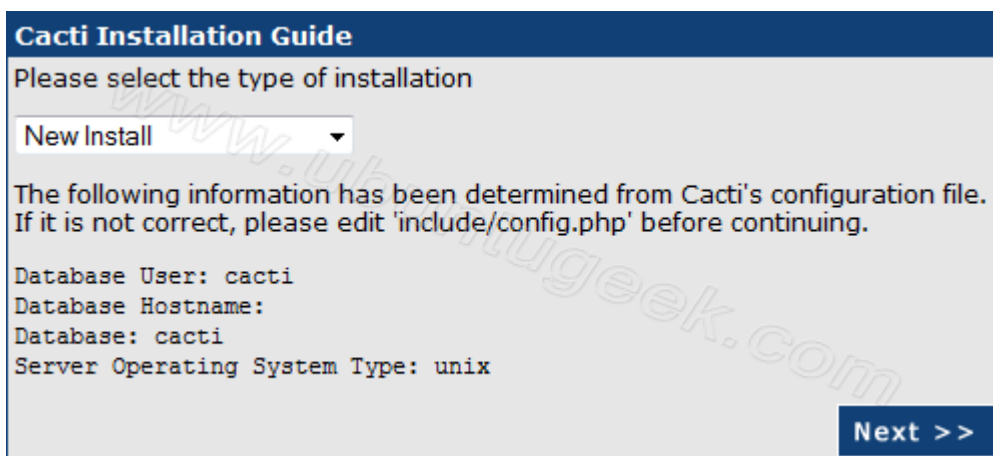
Password...



Bueno el primer paso ya está completado. Ahora, igual que hemos hecho con la instalación de SysLogAnalyzer, apuntaremos a la dirección de nuestro servidor de la siguiente manera:

<http://servername/cacti/>

Siguimos por las pantallas que quedan de instalación,



Cacti Installation Guide

Make sure all of these values are correct before continuing.

[FOUND] RRDTool Binary Path: The path to the rrdtool binary.

 [OK: FILE FOUND]

[FOUND] PHP Binary Path: The path to your PHP binary file (may require a php recompile to get this file).

 [OK: FILE FOUND]

[FOUND] snmpwalk Binary Path: The path to your snmpwalk binary.

 [OK: FILE FOUND]

[FOUND] snmpget Binary Path: The path to your snmpget binary.

 [OK: FILE FOUND]

[FOUND] snmpbulkwalk Binary Path: The path to your snmpbulkwalk binary.

 [OK: FILE FOUND]

[FOUND] snmpgetnext Binary Path: The path to your snmpgetnext binary.

 [OK: FILE FOUND]

[FOUND] Cacti Log File Path: The path to your Cacti log file.

 [OK: FILE FOUND]

SNMP Utility Version: The type of SNMP you have installed. Required if you are using SNMP v2c or don't have embedded SNMP support in PHP.

RRDTool Utility Version: The version of RRDTool that you have installed.

NOTE: Once you click "Finish", all of your settings will be saved and your database will be upgraded if this is an upgrade. You can change any of the settings on this screen at a later time by going to "Cacti Settings" from within Cacti.

Y hasta aquí la instalación. Acto seguido, aparecerá la pantalla de login



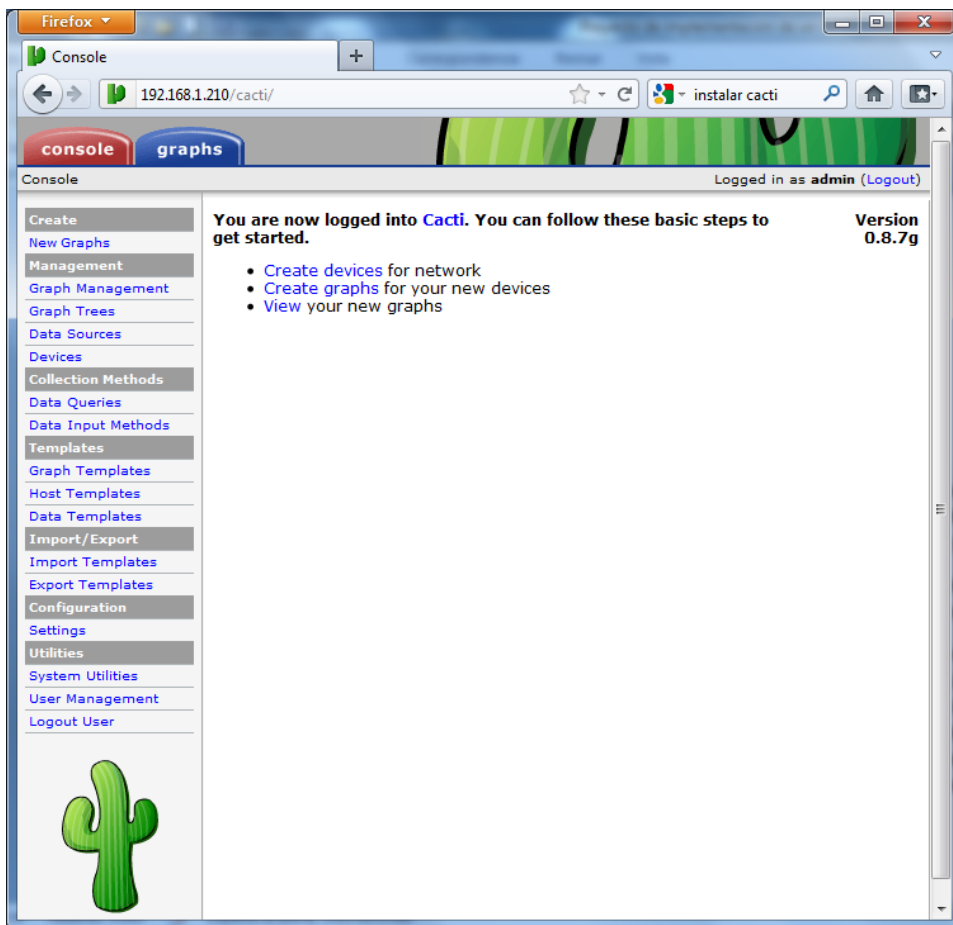
User Login

Please enter your Cacti user name and password below:

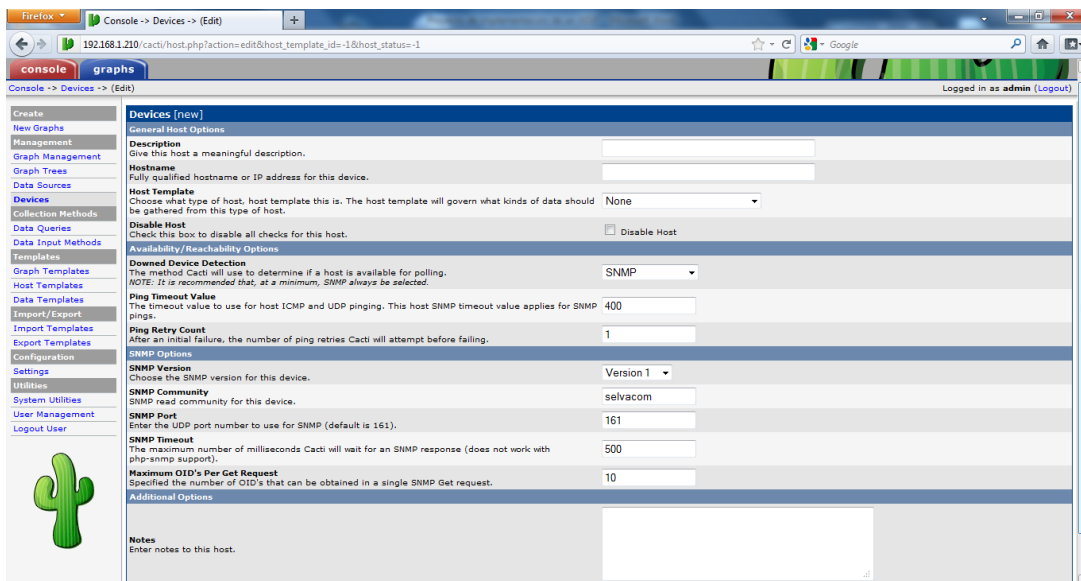
User Name:

Password:

Y el menú principal de Cacti.

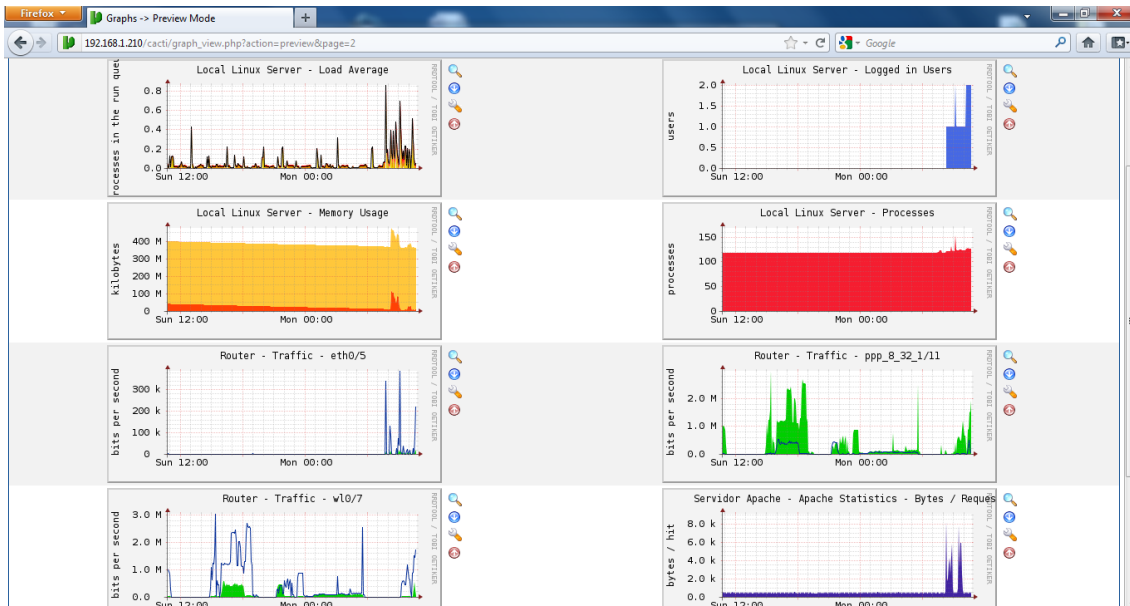


Si pinchamos encima de create device vamos a la pantalla para crear un nuevo dispositivo (al que vamos a monitorear). Aparecerá la siguiente pantalla:



Aquí pondremos el nombre o dirección IP, tipo de dispositivo, versión SNMP,... y una vez finalizado, cacti empezará a monitorear el o los dispositivos que vayamos entrando.

Desde graphs, tendremos acceso a una serie de plantillas para visualizar de manera gráfica datos acerca de muchos parámetros de funcionamiento...



Este es un ejemplo de pantalla con gráficos de diferentes parámetros simultáneamente. Haciendo click en un gráfico, lo podremos ver en detalle y con diferentes escalas de tiempo.

Existe en las comunidades de soporte gran cantidad de plantillas ya creadas para la gran mayor parte de los dispositivos de red, de manera que se puede empezar fácilmente a usar con éxito Cacti.

4.3 Router Os Mikrotik

Vamos a empezar a configurar nuestro primer RB433AH - AP Core.



```

MMM      MMM      KKK      TTTTTTTTTT      KKK
MMMM     MMMM     KKK      TTTTTTTTTT      KKK
MMM MMMM MMM III  KKK  KKK  RRRRRR   OOOOOO   TTT   III  KKK  KKK
MMM MM  MMM III  KKKKK  RRR  RRR  OOO  OOO   TTT   III  KKKKK
MMM     MMM III  KKK  KKK  RRRRRR   OOO  OOO   TTT   III  KKK  KKK
MMM     MMM III  KKK  KKK  RRR  RRR   OOOOOO   TTT   III  KKK  KKK

MikroTik RouterOS 5.7 (c) 1999-2011      http://www.mikrotik.com/

[admin@MikroTik] >

```

En primer lugar vamos a poner un nombre al dispositivo y asignar las direcciones IP a las interfaces:

```
/system identity
set name=AP-Core
```

```
/ip address
```

```
add address=10.0.0.1/29 comment="Enlace AP Slave1" disabled=no
\interface=wlan1 network=10.0.0.0
```

```
add address=10.0.0.9/29 comment="Enlace AP Slave1" disabled=no
\interface=wlan1 network=10.0.0.8
```

```
add address=10.0.0.17/29 comment="Enlace AP Slave1" disabled=no
\interface=wlan1 network=10.0.0.16
```

```
add address=192.168.3.2/24 disabled=no \interface=eth1
network=192.168.3.0
```

```
add address=192.168.4.2/24 disabled=no \interface=eth2
network=192.168.4.0
```

Aplicamos seguridad un perfil de seguridad llamado Security con encriptación WPA2 y definimos las interfaces como AP's, con su correspondiente nombre.

El código facilitamos en el **Anexo 4**.

El paso final consiste en balancear la carga entre las 2 ADSL's, para ello tomamos cada segundo paquete que establece una nueva sesión (nota connection-state=new) y marcamos con marca conexión "impar". Consecuentemente todos los sucesivos paquetes siguientes de la misma sesión llevarán la marca conexión "impar". Notar que estamos pasando esos paquetes a la segunda regla (passthrough=yes) para poner la marca de ruteo en esos paquetes añadiendo la marca conexión.

```
/ ip firewall mangle

add chain=prerouting in-interface=Local connection-state=new nth=1,1,0
\ action=mark-connection new-connection-mark=impar passthrough=yes
comment="" \ disabled=no

add chain=prerouting in-interface=Local connection-mark=impar
action=mark-routing \ new-routing-mark=odd passthrough=no comment=""
disabled=no

add chain=prerouting in-interface=Local connection-state=new nth=1,1,1
\action=mark-connection new-connection-mark=par passthrough=yes
comment="" \ disabled=no

add chain=prerouting in-interface=Local connection-mark=even
action=mark-routing \ new-routing-mark=par passthrough=no comment=""
disabled=no
```

Aplicamos NAT para cada una de las interfaces

```
/ ip firewall nat
add chain=srcnat connection-mark=impar action=src-nat to-
addresses=192.168.3.2 \ to-ports=0-65535 comment="" disabled=no

add chain=srcnat connection-mark=par action=src-nat to-
addresses=192.168.4.2 \ to-ports=0-65535 comment="" disabled=no
```

Y enrutamos todo el tráfico marcado como "par" por una interfaz, y todo el marcado como "impar" por la otra.

```
/ ip route
add dst-address=0.0.0.0/0 gateway=192.168.3.1 scope=255 target-
scope=10 routing-mark=impar \ comment="" disabled=no

add dst-address=0.0.0.0/0 gateway=192.168.4.1 scope=255 target-
scope=10 routing-mark=par \ comment="" disabled=no

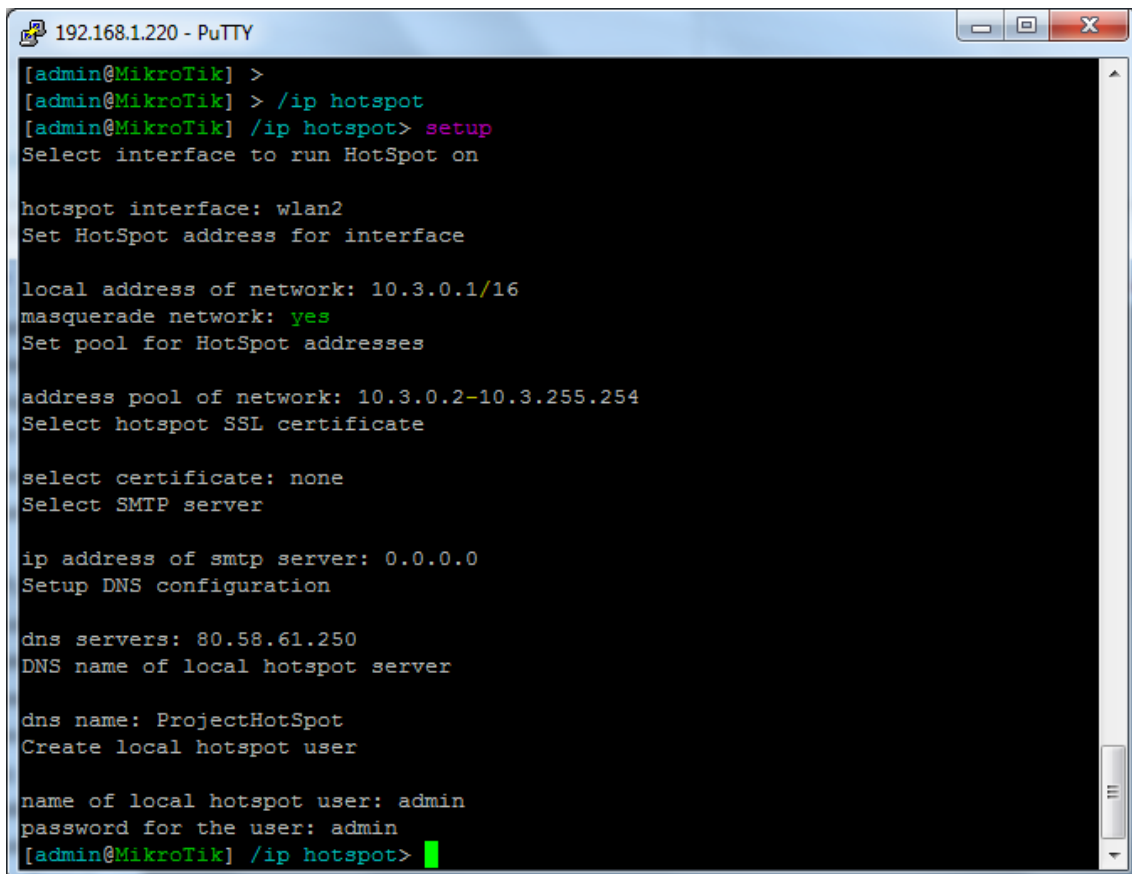
add dst-address=0.0.0.0/0 gateway=192.168.3.1 scope=255 target-
scope=10 comment="" \disabled=no comment="gateway for the router
itself"
```

Configuración de un punto de acceso con HotSpot y servidor PPPoE.

Vamos a configurar el “AP-Slave 3”, que tiene tanto un servidor HotSpot como un servidor PPPoE.

La configuración de las interfaces se hace exactamente igual que en el caso anterior, así que nos vamos a centrar en la creación de un portal cautivo para el HotSpot y de un servidor PPPoE que autentifique a los clientes, compruebe sus credenciales y le otorgue unos determinados permisos, así como auditar las sesiones de usuario.

Para crear nuestro HotSpot, podemos usar el asistente de tal manera que contestando a unas sencillas preguntas lo tendremos creado en unos minutos:



```
[admin@MikroTik] >
[admin@MikroTik] > /ip hotspot
[admin@MikroTik] /ip hotspot> setup
Select interface to run HotSpot on

hotspot interface: wlan2
Set HotSpot address for interface

local address of network: 10.3.0.1/16
masquerade network: yes
Set pool for HotSpot addresses

address pool of network: 10.3.0.2-10.3.255.254
Select hotspot SSL certificate

select certificate: none
Select SMTP server

ip address of smtp server: 0.0.0.0
Setup DNS configuration

dns servers: 80.58.61.250
DNS name of local hotspot server

dns name: ProjectHotSpot
Create local hotspot user

name of local hotspot user: admin
password for the user: admin
[admin@MikroTik] /ip hotspot>
```

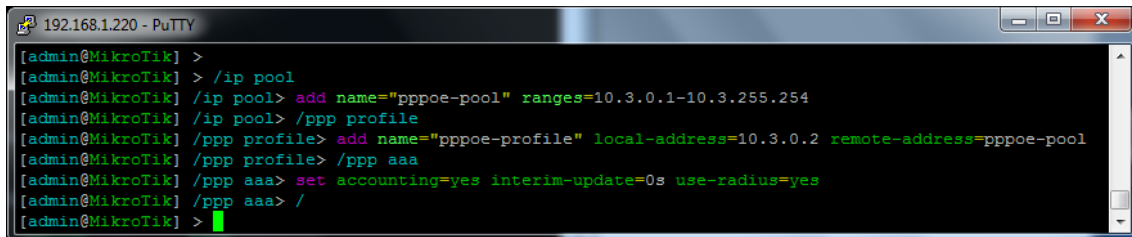
Y para indicarle que para autentificar use nuestro servidor RADIUS, escribiremos lo siguiente:

```
/ip hotspot profile
set hspofl use-radius=yes
```


Configuramos la conexión con nuestro servidor RADIUS:

```
/radius
add service=hotspot,ppp address=80.24.100.100 secret=SuperSecret
```

Y ahora vamos a crear un servidor PPPoE, servicio que será extensible a los otros 2 access points:



```
192.168.1.220 - PuTTY
[admin@MikroTik] >
[admin@MikroTik] > /ip pool
[admin@MikroTik] /ip pool> add name="pppoe-pool" ranges=10.3.0.1-10.3.255.254
[admin@MikroTik] /ip pool> /ppp profile
[admin@MikroTik] /ppp profile> add name="pppoe-profile" local-address=10.3.0.2 remote-address=pppoe-pool
[admin@MikroTik] /ppp profile> /ppp aaa
[admin@MikroTik] /ppp aaa> set accounting=yes interim-update=0s use-radius=yes
[admin@MikroTik] /ppp aaa> /
[admin@MikroTik] >
```

Ahora vamos a implementar nuestra política de QoS, que aparte de priorizar según el protocolo, también vamos a considerar el tamaño de la conexión de tal manera que las descargas grandes, van bajando en la escala de prioridad.

Primero de todo hay que identificar el tráfico y marcarlo:

```
/ip firewall mangle
add action=mark-packet chain=forward new-packet-mark=icmp
passthrough=no protocol=icmp
add action=mark-packet chain=forward dst-port=443 new-packet-mark=ssl
passthrough=no protocol=tcp
add action=mark-packet chain=forward new-packet-mark=p2p p2p=all-p2p
passthrough=no
add action=mark-packet chain=forward new-packet-mark=udp-100 packet-
size=0-100 passthrough=no protocol=udp
add action=mark-packet chain=forward new-packet-mark=udp-500 packet-
size=100-500 passthrough=no protocol=udp
add action=mark-packet chain=forward new-packet-mark=udp-other
passthrough=no protocol=udp
add action=mark-packet chain=forward dst-port=1863 new-packet-
mark=msn-messenger passthrough=no protocol=tcp
add action=mark-packet chain=forward dst-port=110 new-packet-mark=pop3
passthrough=no protocol=tcp
add action=mark-packet chain=forward dst-port=25 new-packet-mark=smtp
passthrough=no protocol=tcp
add action=mark-packet chain=forward dst-port=143 new-packet-mark=imap
passthrough=no protocol=tcp
add action=mark-packet chain=forward new-packet-mark=gre
passthrough=no protocol=gre
add action=mark-packet chain=forward new-packet-mark=ipsec-esp
passthrough=no protocol=ipsec-esp
add action=mark-packet chain=forward new-packet-mark=ipsec-ah
passthrough=no protocol=ipsec-ah
add action=mark-packet chain=forward new-packet-mark=ipencap
passthrough=no protocol=ipencap
add action=mark-packet chain=forward new-packet-mark=ipip
passthrough=no protocol=ipip
```

```

add action=mark-packet chain=forward new-packet-mark=Youtube
passthrough=no src-address-list=Youtube
add action=mark-packet chain=forward dst-port=80 new-packet-mark=http
passthrough=no protocol=tcp
add action=mark-packet chain=forward connection-bytes=1-512000 new-
packet-mark=0bytes passthrough=yes
add action=mark-packet chain=forward connection-bytes=512000-1000000
new-packet-mark=1Mbyte passthrough=yes
add action=mark-packet chain=forward connection-bytes=1000000-3000000
new-packet-mark=3Mbyte passthrough=yes
add action=mark-packet chain=forward connection-bytes=3000000-6000000
new-packet-mark=6Mbyte passthrough=yes
add action=mark-packet chain=forward connection-bytes=6000000-30000000
new-packet-mark=30Mbyte passthrough=yes
add action=mark-packet chain=forward connection-bytes=30000000-
60000000 new-packet-mark=60Mbytes passthrough=yes
add action=mark-packet chain=forward connection-bytes=60000000-0 new-
packet-mark=Infinite passthrough=yes

```

con esto conseguimos identificar y marcar los paquetes que viajan por nuestro AP.

```
/queue type
```

```

add kind=pcq name=PCQ_up_3M pcq-burst-rate=3000k pcq-burst-threshold=0
\ pcq-burst-time=50s pcq-classifier=dst-address pcq-dst-address-
mask=32 \ pcq-dst-address6-mask=128 pcq-limit=20 pcq-rate=1800k \
pcq-src-address-mask=32 pcq-src-address6-mask=128 pcq-total-limit=500

```

Ahora hemos creado una “pcq” (per client queue) de 3Mb por cliente de manera que cada cliente tenga limitado el ancho de banda a 3Mb.

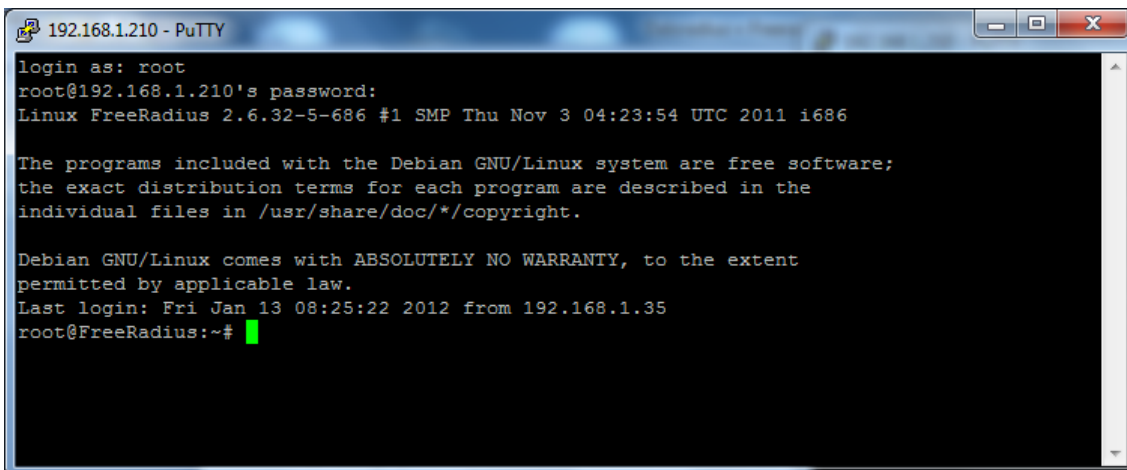
Finalmente creamos el Queue Tree o árbol de colas, lo que va a establecer las prioridades conforme al QoS que hemos creado.

El script para crear las colas de tráfico facilitamos en el **Anexo 2** .

Cabe decir que RouterOS permite usar filtrado de capa 7, aunque el creciente volumen de tráfico encriptado desaconsejan su uso en entornos de producción.

4.4 Resultados.

Arrancamos nuestra maquina.



```
192.168.1.210 - PuTTY
login as: root
root@192.168.1.210's password:
Linux FreeRadius 2.6.32-5-686 #1 SMP Thu Nov 3 04:23:54 UTC 2011 i686

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

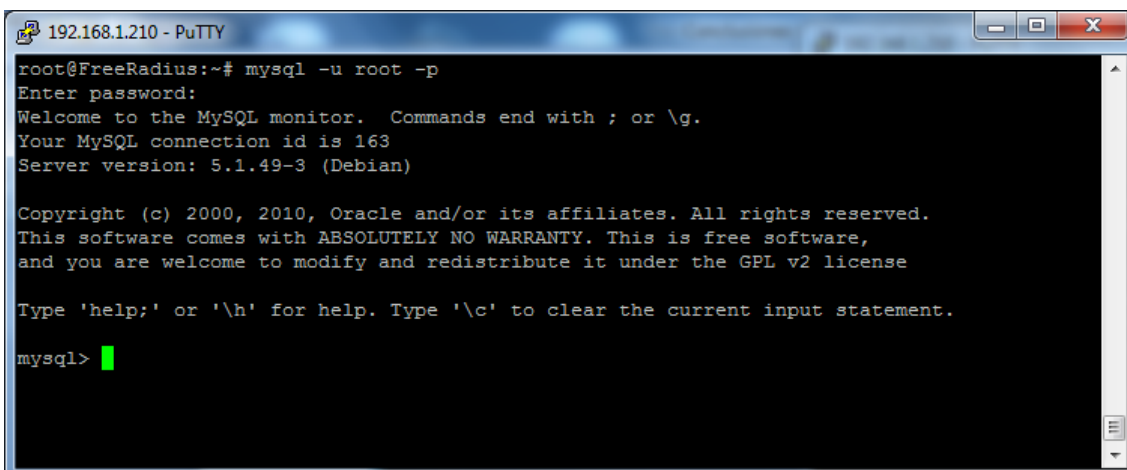
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Fri Jan 13 08:25:22 2012 from 192.168.1.35
root@FreeRadius:~#
```

Vamos a utilizar la bases de datos radius , logeamos en mysql con el comando:

```
mysql -u root -p
```

ponemos la contraseña y ejecutamos comando

```
use radius;
```



```
192.168.1.210 - PuTTY
root@FreeRadius:~# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 163
Server version: 5.1.49-3 (Debian)

Copyright (c) 2000, 2010, Oracle and/or its affiliates. All rights reserved.
This software comes with ABSOLUTELY NO WARRANTY. This is free software,
and you are welcome to modify and redistribute it under the GPL v2 license

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

```

192.168.1.210 - PuTTY
mysql> show databases
-> ;
+-----+
| Database |
+-----+
| information_schema |
| Syslog |
| cacti |
| inventario |
| mysql |
| phpllogcon |
| phpmyadmin |
| projectpier |
| radius |
+-----+
9 rows in set (0.00 sec)

mysql> use radius;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql>

```

Y vamos a probar crear un usuario:

```

INSERT INTO radcheck (UserName, Attribute, Value) ('test user',
'Password', 'passwtest');

```

```

192.168.1.210 - PuTTY
+-----+
9 rows in set (0.00 sec)

mysql> use radius;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

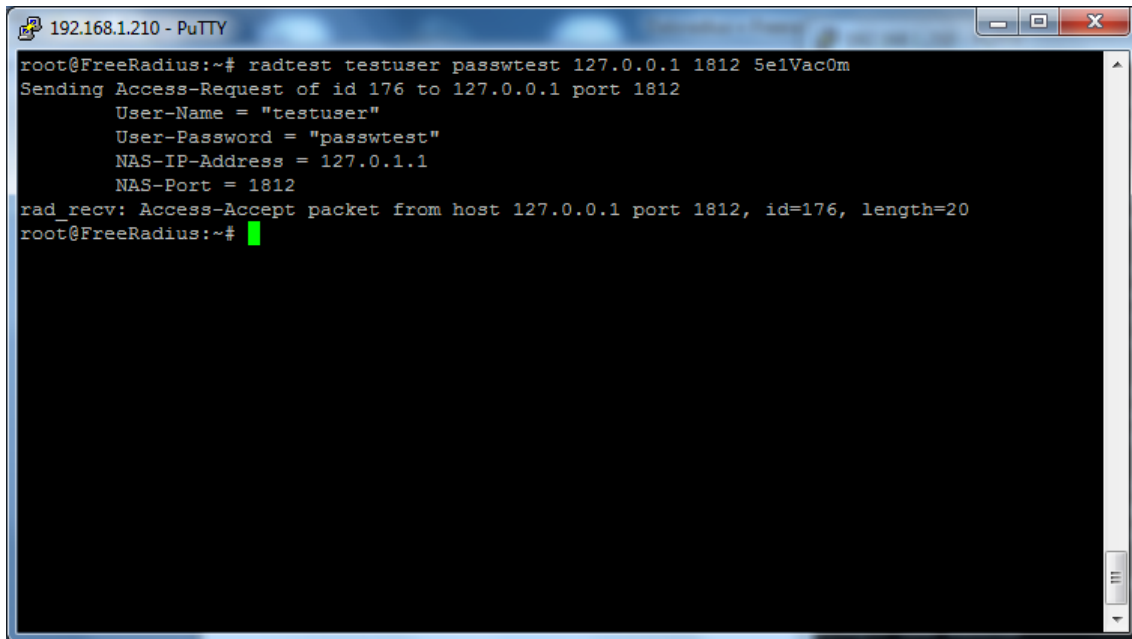
Database changed
mysql>
mysql>
mysql> INSERT INTO radcheck (UserName, Attribute, Value) VALUES ('testuser', 'Password', '
passwtest');
Query OK, 1 row affected (0.00 sec)

mysql> select * from radcheck where UserName like 'testuser'
-> ;
+-----+-----+-----+-----+-----+
| id | username | attribute | op | value |
+-----+-----+-----+-----+-----+
| 242 | testuser | Password | == | passwtest |
+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql>

```

Aseguramos que el usuario esta creado.

A screenshot of a PuTTY terminal window titled "192.168.1.210 - PuTTY". The terminal shows a root user at a machine named "FreeRadius" executing the command "radtest testuser passwtest 127.0.0.1 1812 5e1Vac0m". The output shows the details of the sent Access-Request packet, including the username "testuser", password "passwtest", and NAS IP address "127.0.1.1". The terminal then shows the receipt of an Access-Accept packet from the host "127.0.0.1" on port "1812" with ID "176" and length "20". The prompt returns to "root@FreeRadius:~#".

```
root@FreeRadius:~# radtest testuser passwtest 127.0.0.1 1812 5e1Vac0m
Sending Access-Request of id 176 to 127.0.0.1 port 1812
  User-Name = "testuser"
  User-Password = "passwtest"
  NAS-IP-Address = 127.0.1.1
  NAS-Port = 1812
rad_recv: Access-Accept packet from host 127.0.0.1 port 1812, id=176, length=20
root@FreeRadius:~#
```

Consultamos el log de FreeRadius despues de conectar con el AP por el protocolo PPPoE:

```
Dec 20 08:37:56 FreeRadius freeradius[2009]: Login OK:
[testuser/passwtest] (from client localhost port 1812) Login OK
Dec 20 08:38:08 FreeRadius freeradius[2009]: Login OK:
```

5. Conclusiones.

Con este proyecto, finalizamos la primera parte de la puesta en marcha de un WISP, sin embargo, se deja abierta la puerta a muchas mejoras.

Por ejemplo, Cacti nos permite monitorear no solo los AP, sino también los dispositivos finales, pudiendo tener una visión completa de lo que está sucediendo en la red, el uso de ancho de banda, caídas de línea, y monitorización del rendimiento de todo el hardware de red.

También FreeRADIUS tiene muchas más posibilidades de las que usamos. Nos permite, por ejemplo, autenticar por MAC-address para dispositivos que no pueden proporcionar autenticación, así como usarlo para la autenticación del cifrado WPA2, creando un entorno mucho más seguro.

La implementación de QoS responde a un planteamiento bastante básico, y aunque en la práctica resulta efectiva, se puede trabajar para optimizar su rendimiento.

El ancho de banda disponible actualmente permitiría implementar voz sobre IP, y ofrecer toda una serie de servicios asociados. También, gracias a la tecnología actual, permite ofrecer estos servicios (tanto internet como voz sobre IP) en zonas donde la tecnología ADSL no llega.

En el estudio se previó varias opciones, que finalmente se han descartado, por ejemplo, el uso de tecnología WiMax, pero el alto precio que tienen los dispositivos de esta tecnología, comparados con los de tecnología WiFi, no compensa las ventajas que podía ofrecer. También el diseño de una red tipo "mesh", pero no nos convenció la aleatoriedad en las decisiones de ruteo que utiliza.

Como en cualquier proyecto, esto solo es el principio. A partir de aquí, una buena formación y una política de mejora continua nos irá acercando a las tecnologías y soluciones más indicadas en cada momento.

6. BIBLIOGRAFIA.

1. Mallery, John; Zann, Jason; Kelly, Patrick. "Blindaje de Redes". 1ra Edicion.España. Anaya Multimedia
2. Mikrotik 2011. "Mikrotik Forum" <<http://forum.mikrotik.com>>
3. Freeradius 2011. "Wiki site" <http://wiki.freeradius.org/Main_Page>
4. Mikrotik. 2011. "Wiki Site" <<http://wiki.mikrotik.com>>
5. Mikrotik 2008. "Mikrotik Router OS Wiki" <http://www.mikrotikrouters.com/>
6. Web de daloradius "DaloRadius" <http://daloradius.sourceforge.net/>
7. "Web MySQL" <http://forums.mysql.com/>

7. Anexos

7.1 Contenido del CD.

El CD adjunto contiene los siguientes ficheros:

- La carpeta con el script de implementación de Mikrotik Router OS.
- Memoria del proyecto en pdf.

Anexo2

Script para crear el arbol de las colas y prioridades.

```

/queue tree
add burst-limit=0 burst-threshold=0 burst-time=0s disabled=no limit-
at=0 \
    max-limit=0 name=0-512 packet-mark=0bytes parent=PRIO1 priority=1
queue=\
    default
add burst-limit=0 burst-threshold=0 burst-time=0s disabled=no limit-
at=0 \
    max-limit=0 name=ICMP packet-mark=icmp parent=PRIO1 priority=1
queue=\
    default
add burst-limit=0 burst-threshold=0 burst-time=0s disabled=no limit-
at=0 \
    max-limit=0 name=POP3 packet-mark=pop3 parent=PRIO1 priority=1
queue=\
    default
add burst-limit=0 burst-threshold=0 burst-time=0s disabled=no limit-
at=0 \
    max-limit=0 name=SMTP packet-mark=smtp parent=PRIO1 priority=1
queue=\
    default
add burst-limit=0 burst-threshold=0 burst-time=0s disabled=no limit-
at=0 \
    max-limit=0 name=IMAP packet-mark=imap parent=PRIO1 priority=1
queue=\
    default
add burst-limit=0 burst-threshold=0 burst-time=0s disabled=no limit-
at=0 \
    max-limit=0 name=HTTP packet-mark=http_in parent=PRIO3 priority=1
queue=\
    default-small
add burst-limit=0 burst-threshold=0 burst-time=0s disabled=no limit-
at=0 \
    max-limit=0 name=SSL packet-mark=ssl parent=PRIO1 priority=1
queue=\
    default
add burst-limit=0 burst-threshold=0 burst-time=0s disabled=no limit-
at=0 \
    max-limit=0 name=MSN-MESSENGER packet-mark=msn-messenger
parent=PRIO1 \
    priority=1 queue=default
add burst-limit=0 burst-threshold=0 burst-time=0s disabled=no limit-
at=0 \
    max-limit=0 name=1Mbyte packet-mark=1Mbyte parent=PRIO3 priority=3
queue=\
    default
add burst-limit=0 burst-threshold=0 burst-time=0s disabled=no limit-
at=0 \
    max-limit=0 name=3Mbyte packet-mark=3Mbyte parent=PRIO4 priority=4
queue=\
    default
add burst-limit=0 burst-threshold=0 burst-time=0s disabled=no limit-
at=0 \
    max-limit=0 name=6Mbyte packet-mark=6Mbyte parent=PRIO5 priority=5
queue=\
    default
add burst-limit=0 burst-threshold=0 burst-time=0s disabled=no limit-
at=0 \

```

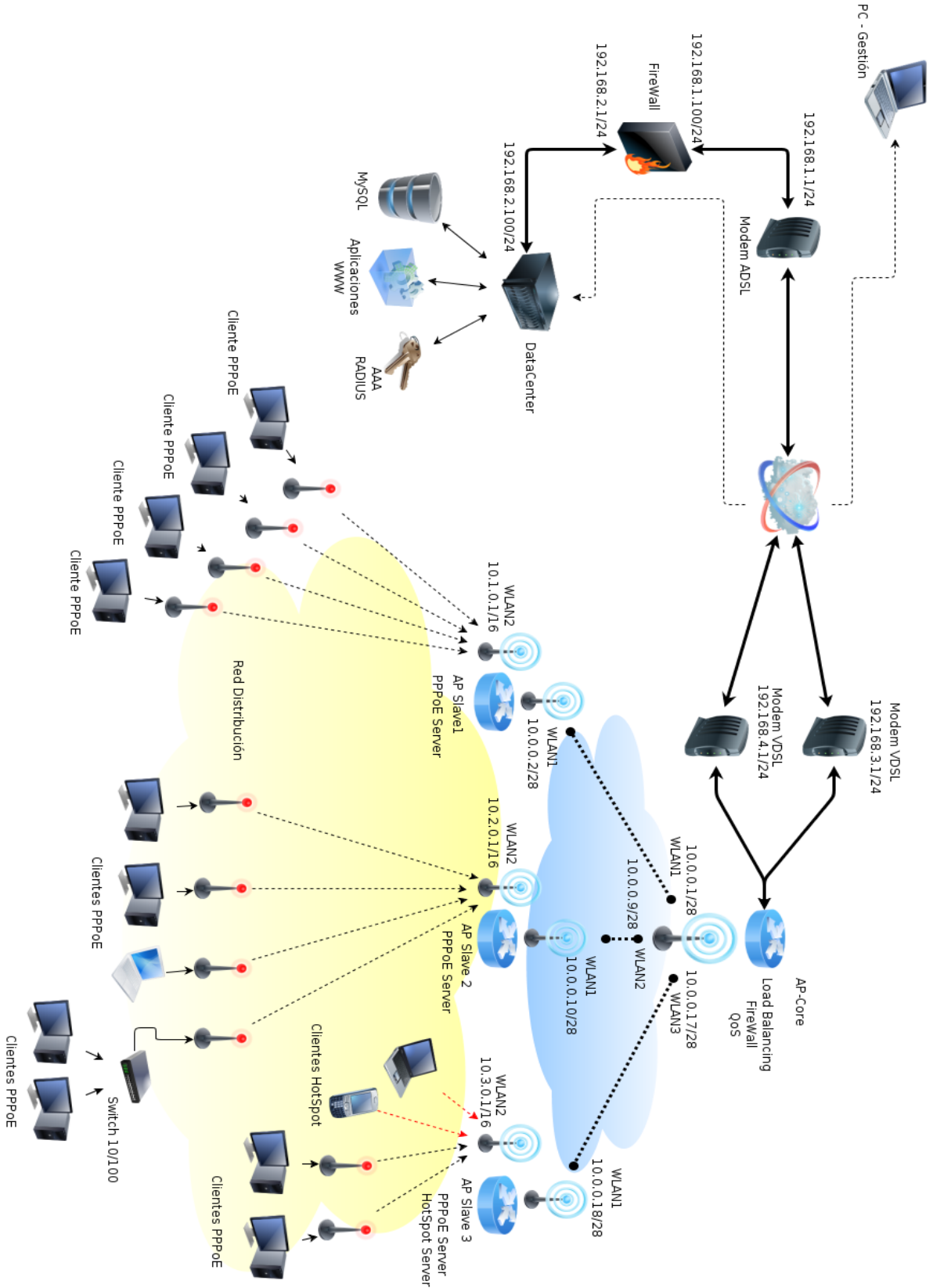
```

    max-limit=0 name=30Mbyte packet-mark=30Mbyte parent=PRIO6
priority=6 \
    queue=default
add burst-limit=0 burst-threshold=0 burst-time=0s disabled=no limit-
at=0 \
    max-limit=0 name=Youtube packet-mark=Youtube parent=PRIO7
priority=7 \
    queue=default
add burst-limit=0 burst-threshold=0 burst-time=0s disabled=no limit-
at=0 \
    max-limit=0 name=60Mbyte packet-mark=60Mbytes parent=PRIO7
priority=7 \
    queue=default
add burst-limit=0 burst-threshold=0 burst-time=0s disabled=no limit-
at=0 \
    max-limit=0 name=Infinite packet-mark=Infinite parent=PRIO8
priority=8 \
    queue=PCQ_up_300k
add burst-limit=0 burst-threshold=0 burst-time=0s disabled=no limit-
at=0 \
    max-limit=0 name=GRE packet-mark=gre parent=PRIO8 priority=8
queue=\
    default
add burst-limit=0 burst-threshold=0 burst-time=0s disabled=no limit-
at=0 \
    max-limit=0 name=IPSEC-ESP packet-mark=ipsec-esp parent=PRIO8
priority=8 \
    queue=default
add burst-limit=0 burst-threshold=0 burst-time=0s disabled=no limit-
at=0 \
    max-limit=0 name=IPSEC-AH packet-mark=ipsec-ah parent=PRIO8
priority=8 \
    queue=default
add burst-limit=0 burst-threshold=0 burst-time=0s disabled=no limit-
at=0 \
    max-limit=0 name=P2P packet-mark=p2p parent=PRIO8 priority=8
queue=\
    PCQ_up_15k
add burst-limit=0 burst-threshold=0 burst-time=0s disabled=no limit-
at=0 \
    max-limit=0 name=IPENCAP packet-mark=ipencap parent=PRIO8
priority=8 \
    queue=default
add burst-limit=0 burst-threshold=0 burst-time=0s disabled=no limit-
at=0 \
    max-limit=0 name=IPIP packet-mark=ipip parent=PRIO8 priority=8
queue=\
    default
add burst-limit=0 burst-threshold=0 burst-time=0s disabled=no limit-
at=0 \
    max-limit=0 name=UDP-100 packet-mark=udp-100 parent=UDP priority=6
queue=\
    PCQ_up_20k
add burst-limit=0 burst-threshold=0 burst-time=0s disabled=no limit-
at=0 \
    max-limit=0 name=UDP-500 packet-mark=upd-500 parent=UDP priority=8
queue=\
    PCQ_up_20k
add burst-limit=0 burst-threshold=0 burst-time=0s disabled=no limit-
at=0 \

```

```
    max-limit=0 name=UDP-Other packet-mark=upd-other parent=UDP
priority=8 \
    queue=PCQ_up_15k
add burst-limit=0 burst-threshold=0 burst-time=0s disabled=no limit-
at=0 \
    max-limit=0 name=OVERALL_IN packet-mark=all parent=global-in
priority=1 \
    queue=PCQ_download
add burst-limit=0 burst-threshold=0 burst-time=0s disabled=no limit-
at=0 \
    max-limit=0 name=Skype packet-mark=Skype parent=PRIO1 priority=1
queue=\
    default
add burst-limit=0 burst-threshold=0 burst-time=0s disabled=no limit-
at=0 \
    max-limit=0 name=DNS packet-mark=dns parent=PRIO1 priority=8
queue=\
    default
```

ANEXO 3



ANEXO 4

Script para configurar miniPCI.

```

set default authentication-types="" eap-methods=passthrough group-
ciphers="" \
  group-key-update=5m interim-update=0s management-
protection=disabled \
  management-protection-key="" mode=none name=default \
  radius-eap-accounting=no radius-mac-accounting=no \
  radius-mac-authentication=no radius-mac-caching=disabled \
  radius-mac-format=XX:XX:XX:XX:XX:XX radius-mac-mode=as-username \
  static-algo-0=none static-algo-1=none static-algo-2=none static-
algo-3=\
  none static-key-0="" static-key-1="" static-key-2="" static-key-
3="" \
  static-sta-private-algo=none static-sta-private-key="" \
  static-transmit-key=key-0 supplicant-identity=MikroTik tls-
certificate=\
  none tls-mode=no-certificates unicast-ciphers="" wpa-pre-shared-
key="" \
  wpa2-pre-shared-key="supersecret"
add authentication-types=wpa-psk,wpa2-psk group-ciphers=tkip \
  group-key-update=5m interim-update=0s management-
protection=allowed \
  management-protection-key="" mode=dynamic-keys name=Security \
  radius-eap-accounting=no radius-mac-accounting=no \
  radius-mac-authentication=no radius-mac-caching=disabled \
  radius-mac-format=XX:XX:XX:XX:XX:XX radius-mac-mode=as-username \
  static-algo-0=none static-algo-1=none static-algo-2=none static-
algo-3=\
  none static-key-0="" static-key-1="" static-key-2="" static-key-
3="" \
  static-sta-private-algo=none static-sta-private-key="" \
  static-transmit-key=key-0 supplicant-identity="" tls-
certificate=none \
  tls-mode=no-certificates unicast-ciphers=tkip wpa-pre-shared-
key=supersecret \
  wpa2-pre-shared-key=supersecret
/interface wireless
set 0 adaptive-noise-immunity=none allow-sharedkey=no antenna-gain=0 \
  antenna-mode=ant-a area="" arp=enabled band=5ghz-a basic-rates-
a/g=6Mbps \
  bridge-mode=enabled burst-time=disabled channel-width=20mhz
compression=\
  no country=no_country_set default-ap-tx-limit=0 default-
authentication=\
  yes default-client-tx-limit=0 default-forwarding=yes dfs-mode=none
\
  disable-running-check=no disabled=no disconnect-timeout=3s
distance=\
  dynamic frame-lifetime=0 frequency=5280 frequency-mode>manual-
txpower \
  frequency-offset=-2200 hide-ssid=no hw-fragmentation-
threshold=disabled \
  hw-protection-mode=none hw-protection-threshold=0 hw-retries=7
12mtu=2290 \
  mac-address=00:21:A4:31:1D:37 max-station-count=2007 mode=station
mtu=\
  1500 name=wlan1 noise-floor-threshold=default nv2-cell-radius=30 \

```

```

nv2-noise-floor-offset=default nv2-preshared-key="" nv2-
qos=default \
nv2-queue-count=2 nv2-security=disabled on-fail-retry-time=100ms \
periodic-calibration=default periodic-calibration-interval=60 \
preamble-mode=both proprietary-extensions=post-2.9.25 radio-name=\
0021A4311D37 rate-selection=legacy rate-set=default scan-
list=default \
security-profile=Security ssid=DodiLab station-bridge-clone-mac=\
00:00:00:00:00:00 supported-rates-a/g=\
6Mbps,9Mbps,12Mbps,18Mbps,24Mbps,36Mbps,48Mbps,54Mbps tdma-period-
size=2 \
tx-power-mode=default update-stats-interval=disabled wds-cost-
range=\
50-150 wds-default-bridge=none wds-default-cost=100 wds-ignore-
ssid=no \
wds-mode=disabled wireless-protocol=unspecified wmm-
support=disabled
set 1 adaptive-noise-immunity=none allow-sharedkey=no antenna-gain=0 \
antenna-mode=ant-a area="" arp=enabled band=2ghz-onlyg basic-
rates-a/g=\
6Mbps basic-rates-b=1Mbps bridge-mode=enabled burst-time=disabled
\
channel-width=20mhz compression=no country=no_country_set \
default-ap-tx-limit=0 default-authentication=yes default-client-
tx-limit=\
0 default-forwarding=yes dfs-mode=none disable-running-check=no
disabled=\
no disconnect-timeout=3s distance=dynamic frame-lifetime=0
frequency=2442 \
frequency-mode>manual-txpower frequency-offset=0 hide-ssid=no \
hw-fragmentation-threshold=disabled hw-protection-mode=none \
hw-protection-threshold=0 hw-retries=7 l2mtu=2290 mac-address=\
00:C0:CA:28:37:3C max-station-count=2007 mode=ap-bridge mtu=1500
name=\
wlan2 noise-floor-threshold=default nv2-cell-radius=30 \
nv2-noise-floor-offset=default nv2-preshared-key="" nv2-
qos=default \
nv2-queue-count=2 nv2-security=disabled on-fail-retry-time=100ms \
periodic-calibration=default periodic-calibration-interval=60 \
preamble-mode=both proprietary-extensions=post-2.9.25 radio-name=\
00C0CA28373C rate-selection=legacy rate-set=default scan-
list=default \
security-profile=Security ssid=Lab2 station-bridge-clone-mac=\
00:00:00:00:00:00 supported-rates-a/g=\
6Mbps,9Mbps,12Mbps,18Mbps,24Mbps,36Mbps,48Mbps,54Mbps supported-
rates-b=\
1Mbps,2Mbps,5.5Mbps,11Mbps tdma-period-size=2 tx-power-
mode=default \
update-stats-interval=disabled wds-cost-range=50-150 wds-default-
bridge=\
none wds-default-cost=100 wds-ignore-ssid=no wds-mode=disabled \
wireless-protocol=unspecified wmm-support=disabled

```