



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

CAMPUS D'ALCOI

MikroTik Switching Cloud Manager

Grado en Ingeniería Informática

Memoria presentada por:

Jorge Lajara Ruiz

Director:

Raúl Llinares Llopis

Codirector:

David Vaello Reig

RESUMEN

El objetivo de este proyecto de fin de grado se centra en establecer una solución y una estandarización a la hora de la configuración y creación de VLANs (Virtual Local Area Networks), concretamente para equipos y soluciones MikroTik, basándose en la solución aportada por otros fabricantes de equipos de interconexión. Para ello se ha realizado una API Web basada en PHP para el control y configuración de equipos MikroTik.

Para poder lograr este objetivo, se ha llevado a cabo una investigación tipo teórico-práctica basada en la revisión y análisis de los equipos y soluciones MikroTik para la creación y configuración de VLANs y todo lo relacionado con ellas.

El marco teórico está compuesto en varios apartados donde se elaboran las siguientes temáticas: concepto de Switching, concepto y utilidad de VLANs, implementación de VLANs a nivel genérico, implementación de VLANs en MikroTik de forma estándar e implementación de VLANs en MikroTik usando la API web desarrollada en este proyecto.

El marco práctico está compuesto sobre todo por parte de código desarrollado en HTML, PHP, CSS, JQUERY y otras tecnologías, donde será comentado sólo partes consideradas más importantes para agilizar la lectura del documento. El resto de código es libre y se encuentra accesible desde el repositorio ya indicado por el autor.

Concluye este proyecto con un conjunto de reflexiones finales, consejos y futuras mejoras.

Palabras clave: MikroTik, Cloud, VLAN, enrutado Inter-VLA, filtrado InterVLAN.

SUMMARY

The objective of this final project degree focuses on establishing a solution and a standardization to the time of the creation and configuration of VLANs (Virtual Local Area Networks), specifically for MikroTik equipment and solutions, based on the solution provided by other manufacturers of interconnection equipment. To do this a Web API based on PHP has been done to control and form MikroTik equipment.

In order to reach this objective, a research has been conducted with a practical and theoretical type based on the review and analysis of the MikroTik equipment and solutions for the creation and configuration of VLANs and everything related with them.

The theoretical framework consists on several paragraphs where the following themes are drawn: switching concept, concept and usefulness of VLANs, deployment of VLANs to generic level, implementation of VLANs in MikroTik standard and implementation of VLANs in MikroTik using the web API developed in this project.

The practical framework is mainly composed by code developed in HTML, PHP, CSS, JQUERY and other technologies, which will be discussed only the parts considered most important for speeding up the reading of the document. The rest of code is free and is accessible from the repository already indicated by the author.

This project concludes with a set of final thoughts, tips, and future improvements.

Keywords: MikroTik, Cloud, VLAN, InterVLAN routing, InterVLAN filtering

AGRADECIMIENTOS

Querría expresar mi reconocimiento y agradecimiento a todas aquellas personas que, gracias a su colaboración, han contribuido a la realización de este Trabajo Fin de Grado:

En primer lugar, mi sincero agradecimiento a Raúl Llinares y David Vaello, director y codirector de este proyecto, por su consejo, ayuda, dedicación y trabajo durante el desarrollo de este proyecto. Sin ellos y sin su apoyo este proyecto no habría sido posible.

También a mis padres por darme una educación de calidad y a todas las personas que han contribuido a ésta: profesores, educadores, hermano, familiares, compañeros...

Por último agradecer a la dirección de la Escuela y del Departamento de Informática junto con los profesores y colaboradores que, gracias a su esfuerzo, han conseguido que el grado en Ingeniería Informática fuera posible.

ÍNDICE

1. INTRODUCCIÓN	
1.1. INTRODUCCIÓN	5
1.1.1. MikroTik	6
1.1.2. Antecedentes	6
1.1.3. Plataformas Soportadas/ Compatibilidad	9
1.2. MOTIVACIÓN	10
1.2.1. ¿Por qué el proyecto?	10
1.2.2. ¿Por qué VLANS?	11
1.2.3. ¿Por qué usar la API?	11
1.3. OBJETIVOS	12
1.4. CONTENIDO DEL PROYECTO	12
2. MARCO TEÓRICO	
2.1. Switching	13
2.1.1. ¿Qué es Switching?	13
2.1.2. ¿Qué es un Switch?	13
2.1.3. ¿Cómo funciona un Switch Capa 2?	14
2.1.4. ¿Cómo funciona un Switch Capa 3?	15
2.2. VLANs	15
2.2.1. ¿Qué es una VLAN?	15
2.2.2. Protocolo IEEE 802.1 Q y Tramas Ethernet	16
2.2.3. Tipo de puertos	16
2.2.4. ¿Por qué usar VLANs?	17
3. MARCO PRÁCTICO	
3.1. Como implementar VLANs a nivel genérico	18
3.2. Como crear VLANs en MikroTik sin usar API	19
3.3. Como crear VLANs en MikroTik usando API	25
3.4. Establecimiento servidor LAMP	27
3.5. Conexiones con la API MikroTik	28
3.6. Código	29
3.6.1. Index	30
3.6.2. Status	32
3.6.3. Switch	33
3.6.4. VLANs	34
3.6.5. Routing	36
3.6.6. ACLs	37
4. CONCLUSIONES	38
4.1. Conclusiones	38
4.2. Futuras líneas de trabajo	39
5. BIBLIOGRAFÍA	40

1. INTRODUCCIÓN

1.1. INTRODUCCIÓN

El sector de las telecomunicaciones y tecnologías de la información cuentan con uno de los futuros más prometedores del tejido productivo, y en consecuencia del empleo, gracias al desarrollo continuo de nuevos productos y servicios cada vez más sofisticados. Además, estos servicios proveen herramientas fundamentales para incrementar la productividad.

Sobra decir que el mundo web ha desbancado a competidores mucho más antiguos como la televisión o los teléfonos y se ha convertido en la vía más utilizada para informarse, entretenerse o comunicarse, por lo que cualquier empresa debe asegurarse que tiene una correcta y eficiente implementación de estos servicios.

En este proyecto se pretende facilitar la implementación de VLANs de cara a un usuario con conocimientos de interconexiones de redes que no está familiarizado con la solución aportada para dispositivos MikroTik. Para ello se ha elaborado una aplicación web a un nivel tan simple como potente, y además basado en las soluciones aportadas por fabricantes con más peso en el campo de las redes de datos.

Para concluir, se ha desarrollado la aplicación de forma que pueda ser ejecutada en plataformas heterogénea (en cualquier dispositivo o sistema operativo) y accesible desde cualquier lugar o equipo con una correcta configuración.

1.1.1. MikroTik

MikroTik es una empresa con sede en Riga (Letonia), fundada en 1995 para desarrollar routers y sistemas Wireless ISP. Actualmente, MikroTik ofrece hardware y software para la conexión a Internet en la mayoría de los países de todo el mundo.

Nacieron con la creación de un sistema operativo para PC basado en ip-tables, aumentando progresivamente las funcionalidades del mismo hasta llegar a realizar a día de hoy, cualquier funcionalidad de switching, routing o Access Point, desarrollando incluso protocolos propios. Su experiencia en el uso de hardware de PC estándar de la industria y sistemas de enrutamiento completa les permitió en 1997 crear el sistema operativo RouterOS que proporciona una amplia estabilidad, control y flexibilidad para todo tipo de interfaces de datos y enrutamiento.

En el año 2002 decidieron desarrollar su propio hardware, y así nació la marca RouterBoard. Desde entonces, se ha creado un amplio canal de distribución ofreciendo estas soluciones en la mayor parte del mundo. Actualmente, disponen de un amplio abanico de hardware, cubriendo prácticamente cualquier escenario de interconexión de datos.

Poco a poco, hemos visto como MikroTik ha ido posicionándose en el mercado de los WISP (Wireless Internet Service Provider), buscando hoy en día su hueco en CPD con su nueva gama de routers CCR. Hoy en día, es uno de los fabricantes de electrónica de red con mayor proyección internacional.



1.1.2. Antecedentes

Previamente al desarrollo de este proyecto, se analizó las soluciones que aportan otros fabricantes de dispositivos que operan en la misma área en que se basa este proyecto: Cloud Switching. Y se pudo observar que:

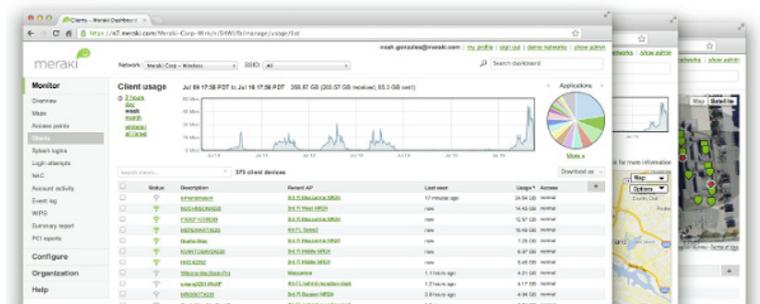
Soluciones Cloud Switching:

Para establecer una base en la creación de la API web se observó principalmente las soluciones aportadas por dos fabricantes: Cisco con la propuesta de Meraki y Ubiquiti con EdgeMax.

 Meraki: fundada en 2006 Sanjit Biswas y John Bicket y adquirida por Cisco en 2012, ofrece una familia de productos completa administrados a través de la nube, incluyendo LAN inalámbrica, switches Ethernet y dispositivos de seguridad.

En el apartado de switches ofrece dos soluciones actualmente: MS320 (Switches capa 3) y MS220 (Switches capa 2) que son configurados a través de un panel de control web alojado en los servidores de Cisco, dando de alta previamente los equipos correspondientes. Entre otras características de Meraki cabe destacar:

- Panel único de administración de recursos para la implementación distribuida de switches, puntos de acceso inalámbricos y firewalls en múltiples ubicaciones en toda la web.
- Compatibilidad con IEEE 802.1X para el control de acceso a la red basado en puertos.
- Configuración VLAN por puerto.
- Capa 3 en la serie MS320 que amplía el routing hasta el borde de la red.
- Herramientas eficaces en vivo, como pruebas de cable para aislar los problemas de capa física.



Panel de control Meraki

Switch ports for the last week

VOIP 1-5 link.*100 Mbps* [help](#) 31 switch ports

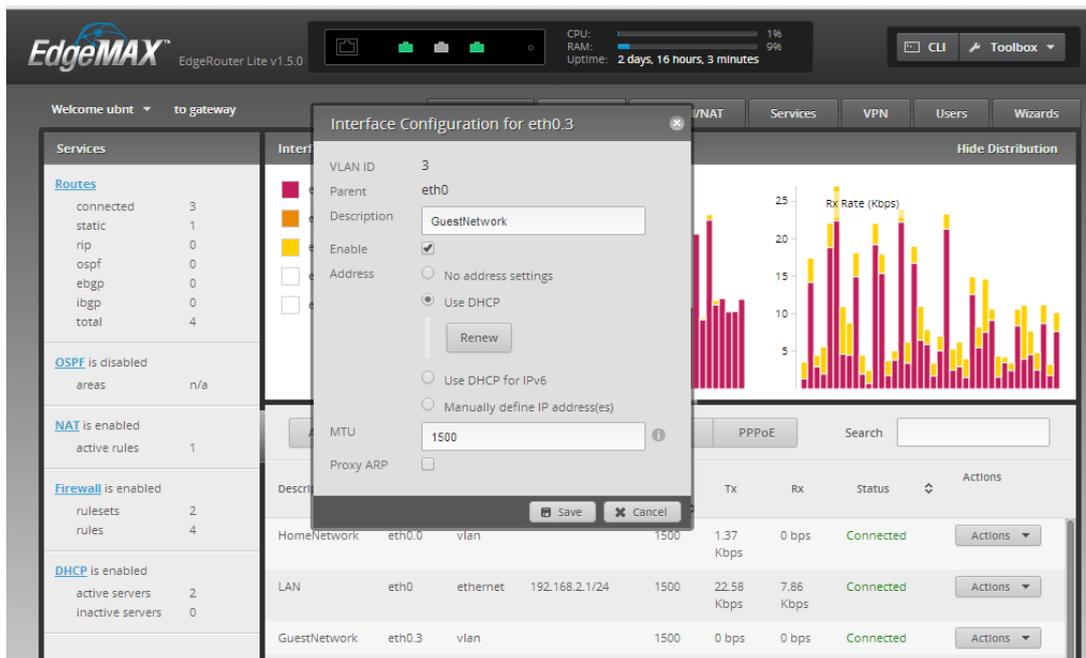
Switch / Port	Type	VLAN	Tags	Port#	Link
4 A10 Amit / 4	access	108, voice 104	VOIP	4	Auto negotiate (100 Mbps)
4 A10 Amit / 5	access	108, voice 104	VOIP	5	Auto negotiate (100 Mbps)
4 A9 Amit / 2	access	108, voice 104	VOIP	2	Auto negotiate (100 Mbps)
4 A9 Amit / 3	access	108, voice 104	VOIP	3	Auto negotiate (100 Mbps)
4 A9 Amit / 4	access	108, voice 104	VOIP	4	Auto negotiate (100 Mbps)
4 A9 Amit / 5	access	108, voice 104	VOIP	5	Auto negotiate (100 Mbps)

Configuración de puertos a través de Meraki

EdgeMAX EdgeMax: es una plataforma para configurar, controlar y llevar a cabo las funciones de enrutamiento, conmutación, seguridad y gestión necesarios para ejecutar eficientemente los equipos de Ubiquiti. Ubiquiti es una compañía estadounidense fundada en 2003 proveedora de tecnología para la creación de redes de datos.

Entre otras características de EdgeMax cabe destacar:

- Acceso al panel de control mediante un navegador web junto a la IP del equipo.
- Visibilidad de esquema del equipo junto al estado de los puertos.
- Configuración VLAN por puerto.
- Enrutamiento por interfaces VLAN.



EdgeMax y configuración de puerto para VLAN

Previamente a la solución en la nube, se encuentra la solución de configuración directamente del equipo sin pasar por un interfaz web. Para ello se analizó la metodología propuesta por varios fabricantes (Cisco y HP) y se concluyó de cada uno que:

Soluciones Switching:



Cisco: es una empresa global fundada en 1984 con sede en San José, (California), principalmente dedicada a la fabricación, venta, mantenimiento y consultoría de equipos de telecomunicaciones. Observando la solución para el establecimiento de VLANs de sus switches se observa que:

- Primero se declara la VLAN.
- Después se establece puerto a puerto si es modo acceso (Access) o modo troncal (Trunk).
- Si se desea enrutar, se determina una IP a la interfaz VLAN, una puerta de enlace por defecto y las rutas correspondientes.



Hewlett-Packard: más conocida como HP, es una de las mayores empresas de tecnologías de la información del mundo, siendo estadounidense y con sede en Palo Alto, California. Fabrica y comercializa hardware y software además de brindar servicios de asistencia relacionados con la informática. La compañía fue fundada en 1939 por William Hewlett y David Packard. Observando la solución para el establecimiento de VLANs de sus switches se observa que:

- Primero se declara la VLAN.
- Después se establece puerto a puerto si es modo acceso (Untagged) o modo troncal (Tagged).
- Si se desea enrutar, se determina una IP a la interfaz VLAN y una puerta de enlace por defecto y las rutas correspondientes.

Por lo tanto, se establece que existe una base común entre fabricantes a la hora de crear y establecer VLANs, y que tras el análisis y estudio de las soluciones MikroTik se ve que difiere a la del resto de fabricantes, ya que establece una solución propia junto a una potente capacidad de configuración.

Posteriormente será analizada la solución ofrecida por MikroTik y por qué ha sido objetivo de este proyecto.

1.1.3. Plataformas soportadas / Compatibilidad

La idea en el desarrollo de la plataforma web de este proyecto ha sido siempre el de crear una base común a otros desarrolladores y simplificar lo máximo posible de cara a un usuario final. Unificar también las diferentes configuraciones entre dispositivos de la gama MikroTik para establecer una solución única.

La elección de la API PHP se debe a que permitía establecer una plataforma común entre dispositivos con distintos sistemas operativos (Windows, Linux, Mac OS, Android, IOS...), ya que con el establecimiento de un servidor Apache con módulo PHP es suficiente para poner en marcha el servicio.

Simplemente, cualquier cliente accediendo al servidor PHP y siendo la autenticación correcta, es capaz desde su navegador de usar la plataforma web.

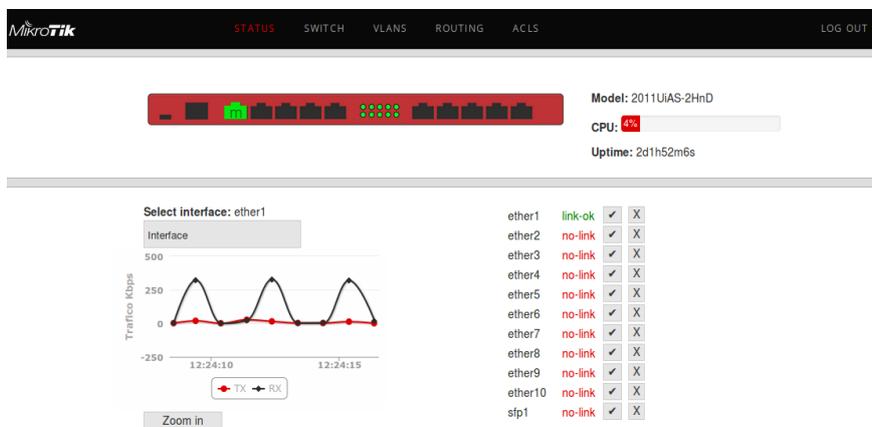
Para establecer el servidor en los distintos sistemas operativos se recomienda el uso de las siguientes herramientas para facilitar el uso. Entiéndase que es una recomendación y que cada usuario es libre de usar sus propias herramientas.

Linux: Instalar un servidor LAMP nativo (Linux, Apache, MySQL, PHP). La instalación varía según la distribución utilizada. En apartados posteriores se muestra la forma usada en este proyecto de instalación y comprobación del funcionamiento.

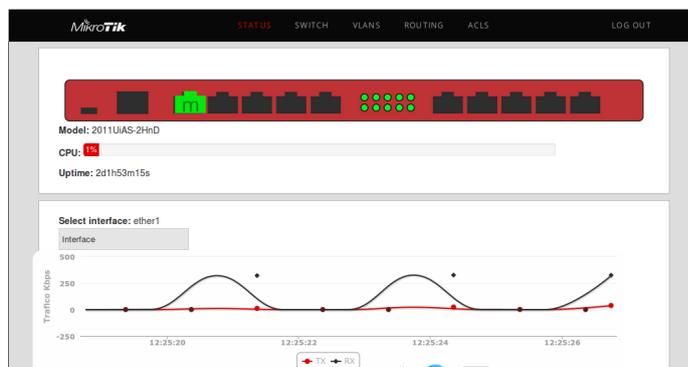
Windows: Wamp server (<http://www.wampserver.com>)

Mac OS: Mamp server (<https://www.mamp.info>)

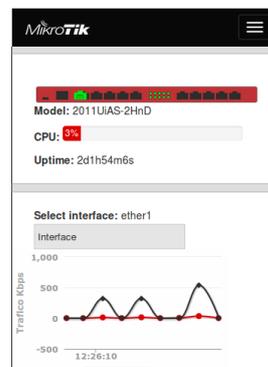
La parte de la vista se ha resuelto utilizando el framework Bootstrap para establecer una vista correcta entre la multitud de dispositivos y resoluciones posibles. Haciendo que cada cliente tenga una vista adecuada y a su vez con la misma potencia de configuración.



Resolución ordenador



Resolución tablet



Resolución móvil

1.2 MOTIVACIÓN

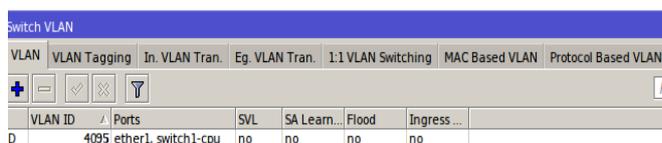
1.2.1. ¿Por qué el proyecto?

Antes de empezar con este proyecto, se observó la gran potencia de configuración que nos podía permitir el uso de equipos MikroTik, convirtiéndose a su vez esa libertad de configuración en su principal problema de cara a un usuario final.

La libertad de configuración era tal que a la hora de definir e implementar VLANs se perdía mucho tiempo simplemente en intentar traducir que significaba cada opción, incluyendo que la documentación online de MikroTik, al ser una solución prácticamente nueva, no dispone de demasiados ejemplos teórico-prácticos como podría ser de otros fabricantes.

Del mismo modo, no parte de la base de creación y definición de VLANs que otros fabricantes han establecidos. Al determinar su propio sistema operativo, la implementación de VLANs no es intuitiva ya que al disponer de tanta libertad de elección, hay tener varias cosas en cuenta que dificultan a un usuario final condicionado con las soluciones de otros fabricantes.

Además MikroTik tiene dos gamas de productos principalmente: Cloud Router Switch (configuración por defecto de Switch) y RouterBoard (configuración por defecto de Router). Por lo que se tenía que abordar el problema intentando establecer una plataforma común a ambos tipos de equipos, ya que aunque el menú de configuración pareciera similar, en el apartado de Switching hay una gran diferencia de potencial y de opciones.



Menú Cloud Router Switch



Menú RouterBoard

Paralelamente y siguiendo con la tendencia actual a servicios cloud, se aprovecha la posibilidad de tener una API web que conecta con el dispositivo y permite el envío de comandos para la configuración de los equipos. Ésto brindó una oportunidad para establecer una plataforma en la nube, permitiendo ser accedida desde cualquier lugar y proporcionando una heterogeneidad de acceso universal.

Analizando estos problema, se decidió establecer una herramienta útil y sencilla basada en las bases que otros desarrolladores de equipos de interconexión han establecido, permitiendo que un usuario final, sea capaz de utilizando esta API, de llevar a cabo una configuración completa y funcional.

1.2.2. ¿Por qué VLANs?

La solución a esta pregunta es simple: porque la solución MikroTik al menos, no sigue el patrón común de configuración de otros fabricantes en cuanto al apartado de VLANs.

Del mismo modo que MikroTik ofrece una solución compleja en el apartado de Switching, sin embargo tiene una potente solución en otros apartados tales como: Wireless, Bridging, Routing, Firewalling... Por lo tanto, entrar en otros puntos del que se aprecia una potencia y sencillez de configuración adecuada, sería una pérdida de tiempo y por lo tanto perder enfoque en establecer algún tipo de solución práctica.

Además, gracias al incremento de dispositivos informáticos y de redes por parte de las empresas, el uso de las VLANs o subredes, cada vez está más en aumento y se requiere de una correcta configuración por parte del administrador de éstas. Si el administrador no consigue establecer una configuración adecuada de forma óptima, la potencia de división de VLANs y subredes puede ser desperdiciada o incluso nunca llegar a ser correcta y por tanto tener problemas en la explotación de sus recursos.

Para concluir, la definición y ventajas del uso de VLANs serán analizadas posteriormente.

1.2.3. ¿Por qué usar la API?

La respuesta a esta pregunta es por la potencia, sencillez, homogeneidad, seguridad y efectividad de cara a un usuario final.

-Potencia, ya que con el uso de la API, se consigue que con un conocimiento básico de creación de VLANs establecer una configuración de una forma óptima y adecuada.

-Sencillez, ya que los menús han sido simplificados al máximo, dotando al usuario final una simplicidad básica, ya que se ha ocultado toda la capa de complejidad que dificultaba una configuración previa.

-Homogeneidad, ya que de cara al usuario final, se ha establecido una base de configuración igual para las distintas soluciones que plantea MikroTik (RouterBoard y Cloud Router Switch) partiendo del patrón común de configuración del resto de fabricantes.

-Seguridad, ya que eliminando la gran cantidad de menús que se mostraban con el uso de la solución MikroTik, el usuario es capaz de enfocarse en las partes básicas para establecer la configuración adecuada sin riesgo a desperdiciar recursos por una mala gestión.

-Efectividad, ya que el tiempo ahorrado usando la API, elimina estudiar todas las soluciones posibles aportadas por MikroTik, centrándose en lo mínimo para una configuración óptima.

La principal desventaja de la API es la puesta en marcha de un servidor PHP para poder ejecutarla, pero esta principal desventaja dota a la plataforma de libertad y heterogeneidad para que cada usuario establezca sus propias conexiones, permitiendo ser accedida a su gusto y desde cualquier plataforma.

1.3. OBJETIVOS

En cuanto a los objetivos de este proyecto fin de grado, se centran en dos tipos: objetivo general y objetivos específicos.

Objetivo general:

-Realizar una aplicación web para el control y configuración de equipos MikroTik, concretamente en la gestión y configuración de VLANs.

Objetivos específicos:

- Exponer la situación actual de los diversos fabricantes y las soluciones aportadas por éstos a la hora de configuración de switches y VLANs.
- Familiarizarse con el entorno MikroTik y su potencia de configuración.
- Establecer conexiones PHP mediante la API MikroTik y un servidor Apache previamente configurado.
- Realizar la estructura de la aplicación web para posteriormente dotarla de lógica de aplicación.
- Comprobar ejemplos de estudio prácticos utilizando la API y su correcto funcionamiento.
- Optimizar la metodología MikroTik para mejorar la experiencia a un usuario final.

1.4. CONTENIDO DEL PROYECTO

El contenido del proyecto se centra en dos partes claramente diferenciadas:

-Marco teórico: se estudia el concepto de Switching, concepto y utilidad de VLANs, implementación de VLANs a nivel genérico, implementación de VLANs en MikroTik de forma estándar e implementación de VLANs en MikroTik usando la API web desarrollada en este proyecto. Además de teoría a nivel de desarrollo web, de puesta a punto de un servidor Apache en diversas plataformas y de API de MikroTik, entre otros.

-Marco práctico: compuesto sobretudo por parte de código desarrollado en HTML, PHP, CSS, JQUERY y otras tecnologías. A parte, la configuración y puesta a punto de equipos MikroTik, establecimiento de servidores Apache e interconexiones de la API MikroTik, entre otros.

2. MARCO TEÓRICO

2.1. Switching

2.1.1. ¿Qué es switching?

Switching es una tecnología de redes que consiste en el uso de conmutadores (Switch), para la interconexión de redes de datos. Esta tecnología incrementa el rendimiento, disminuye la latencia y obtiene escalabilidad en el ancho de banda, entre otras cosas.

2.1.2. ¿Qué es un switch?

Un switch o conmutador, es un dispositivo de red que opera en la Capa 2 (nivel de enlace de datos) del modelo OSI, aunque puede operar en la Capa 3 (nivel de red), integrando Routing (enrutamiento). Su función es la de interconectar dos o más segmentos de red en una red local (LAN).

La gran diferencia con un Hub (conmutador), es que los datos provenientes del puerto de origen solamente son enviados al puerto de destino. Esto se debe a que los switches almacenan una tabla donde se establece un canal de comunicación exclusiva entre el origen y el destino, de esta forma, la red no queda “limitada” a un único puerto en el envío de información. Ésto aumenta el rendimiento de la red ya que la comunicación está siempre disponible, excepto cuando dos o más computadoras intentan enviar datos simultáneamente a la misma máquina. Esta característica también disminuye los errores (colisiones de paquetes de datos, por ejemplo).



Switches

2.1.3. ¿Cómo funciona un switch de Capa 2?

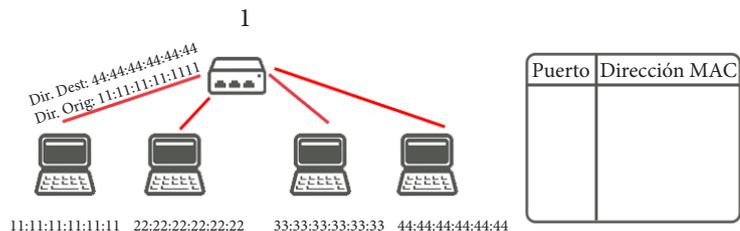
La función básica que realiza un switch se conoce como conmutación y consiste en conmutar datos entre los diferentes puertos del dispositivo de la red.

Cuando un switch opera en capa 2, toma las decisiones en base a la MAC destino de la trama. Lo que significa que:

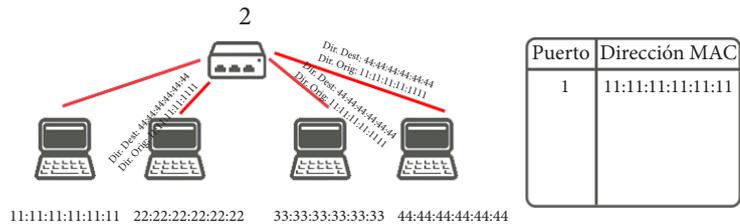
- El alcance del dominio de colisión consiste en el propio puerto del switch y los que esté conectado a él.
- Los hosts conectados pueden operar en full-duplex, pueden enviar y recibir a la vez.
- El ancho de banda no es compartido, cada puerto ofrece ancho de banda dedicado.
- Los errores en la trama no son propagados.

Funcionamiento:

1) El switch inspecciona la trama y busca la MAC destino en su tabla de direcciones (CAM) para ver si tiene el puerto a donde tiene que enviar la trama a la cual envía si la encuentra.



2) Si no tiene la información en la CAM, el switch envía la trama a todos los puertos asignados a la VLAN por donde recibió la trama, esto es llamado "unknown unicast flooding" con la dirección de destino unicast desconocida.



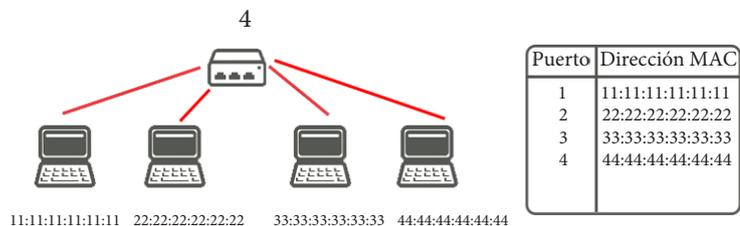
Además aprende en su tabla la dirección MAC y el puerto.

3) Cuando el equipo al que se le ha enviado la trama responde, la MAC junto al puerto se añaden otra vez a la tabla CAM.



También se almacena la hora (time stamp). Si la MAC aprendida en un puerto aparece en otro puerto, la entrada vieja es eliminada y se almacenan los datos más actuales (time stamp más reciente). Si una MAC ya está en la tabla CAM en el mismo puerto y VLAN solo se actualiza la hora (time stamp).

4) Repitiendo este proceso se consigue conocer todos los puertos junto a las MAC que componen el switch.



2.1.4. ¿Cómo funciona un switch de Capa 3?

Los switches de capa 3 se caracterizan por que, además de las funciones tradicionales de la capa 2, incorporan algunas funciones de routing, como por ejemplo la determinación de un camino basado en informaciones de capa de red y soporte a los protocolos de routing tradicionales (RIP, OSPF, etc)

Los switches de capa 3 soportan también la definición de redes virtuales (VLAN), y según modelos posibilitan la comunicación entre las diversas VLAN sin la necesidad de utilizar un router externo.

Por permitir la unión de segmentos de diferentes dominios de difusión o broadcast, los switches de capa 3 son particularmente recomendados para la segmentación de redes LAN muy grandes, donde la simple utilización de switches de capa 2 provocaría una pérdida de rendimiento y eficiencia de la LAN, debido a la cantidad excesiva de usuarios.

2.2. VLANs

2.2.1. ¿Qué es una VLAN?

Una VLAN, acrónimo de virtual LAN (red de área local virtual), es un método para crear redes lógicas independientes dentro de una misma red física y su protocolo de utilización es el IEE 802. IQ.

Es decir, consiste en una red de ordenadores que se comportan como si estuviesen conectados al mismo conmutador o switch, aunque pueden estar en realidad conectados físicamente a diferentes segmentos de una red de área local. Los administradores de red configuran las VLANs mediante software en lugar de hardware, lo que las hace extremadamente flexibles.

Aunque las más habituales son las VLANs basadas en puertos (nivel 1), las redes de área local virtuales se pueden clasificar en cuatro tipos según el nivel de la jerarquía OSI en el que operen:

- VLAN de nivel 1 (por puerto). También conocida como “port switching”. Se especifica qué puertos del switch pertenecen a la VLAN, los miembros de dicha VLAN son los que se conecten a esos puertos. No permite la movilidad de los usuarios, habría que reconfigurar las VLANs si el usuario se mueve físicamente.

- VLAN de nivel 2 por direcciones MAC. Se asignan hosts a una VLAN en función de su dirección MAC. Tiene la ventaja de que no hay que reconfigurar el dispositivo de conmutación si el usuario cambia su localización, es decir, se conecta a otro puerto de ese u otro dispositivo. El principal inconveniente es que si hay cientos de usuarios habría que asignar los miembros uno a uno.

- VLAN de nivel 2 por tipo de protocolo. La VLAN queda determinada por el contenido del campo tipo de protocolo de la trama MAC. Por ejemplo, se asociaría VLAN 1 al protocolo IPv4, VLAN 2 al protocolo IPv6, VLAN 3 a AppleTalk, VLAN 4 a IPX, etc.

- VLAN de nivel 3 por direcciones de subred (subred virtual). La cabecera de nivel 3 se utiliza para mapear la VLAN a la que pertenece. En este tipo de VLAN son los paquetes, y no las estaciones, quienes pertenecen a la VLAN. Estaciones con múltiples protocolos de red (nivel 3) estarán en múltiples VLANs.

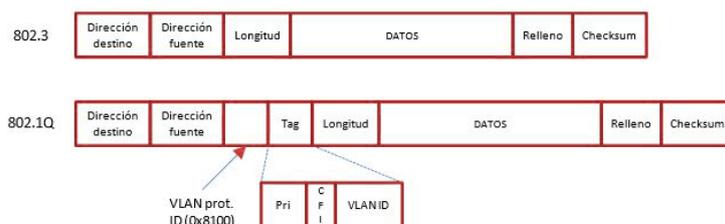
- VLAN de niveles superiores. Se crea una VLAN para cada aplicación: FTP, flujos multimedia, correo electrónico, etc.

2.2.2. Protocolo IEEE 802.1 Q y Tramas Ethernet

El protocolo IEEE 802.1.Q es un protocolo cuyo objetivo es el de permitir a múltiples redes compartir de forma transparente el mismo medio físico, sin problemas de interferencia entre ellas (Trunking).

Por lo tanto, permite identificar a una trama como proveniente de un equipo conectado a una red determinada. Una trama perteneciente a una VLAN sólo se va a distribuir a los equipos que pertenezcan a su misma VLAN, de forma que se separan dominios de broadcast.

El formato de la trama es el siguiente:



Trama Ethernet IEEE 802.1 Q

El protocolo 802.1Q propone añadir 4 bytes al encabezado Ethernet original en lugar de encapsular la trama original. El valor del campo EtherType se cambia a 0x8100 para señalar el cambio en el formato de la trama.

-Los primeros 3 bits son el campo User Priority Field que pueden ser usados para asignar un nivel de prioridad.

-El próximo bit es el campo Canonical Format Indicator (CFI) usado para indicar la presencia de un campo Routing Information Field(RIF).

-Los restantes 12 bits son el VLAN Identifier(VID) que identifica de forma única a la VLAN a la cual pertenece la trama Ethernet.

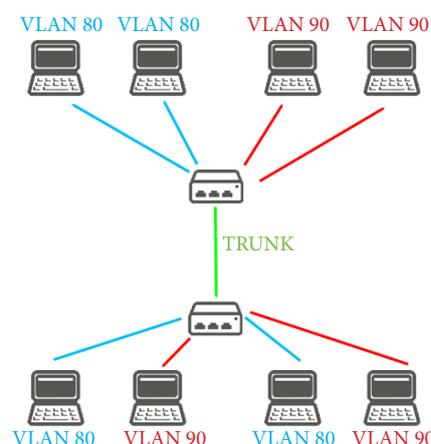
2.2.3 Tipos de puertos

Este proyecto se centra principalmente en las VLANs de nivel 1 (por puerto), siendo esta categoría la más usada comúnmente.

Existen dos tipos de puertos principalmente:

-Puertos modo acceso o untagged: La trama puede ser etiquetada (Tagged) mientras viaja de un switch a otro. Cuando entra una trama Ethernet se le añade el TAG de 802.1Q. Cuando sale una trama 802.1Q se le quita el TAG, para que llegue a la estación correspondiente con el formato IEEE 802.3 original. Se utiliza para separar los puertos en las correspondientes VLANs.

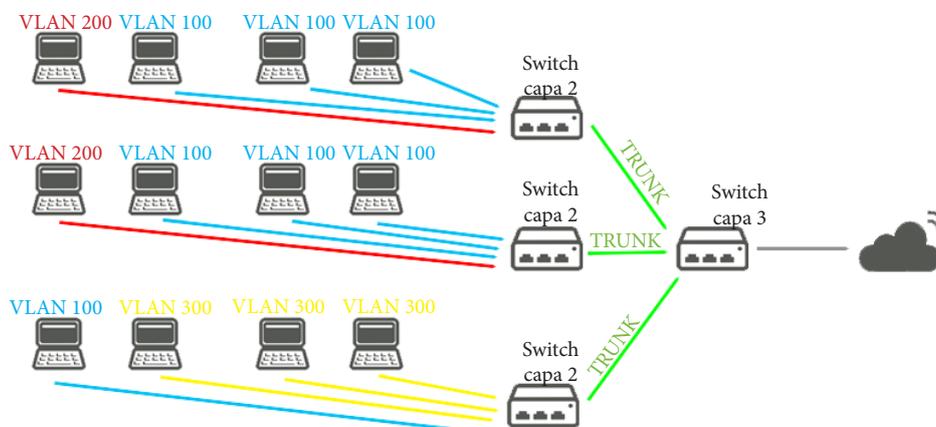
-Puertos modo trunk o tagged: La trama puede ser etiquetada según la VLAN a la que pertenece. Las tramas que le llegan y que salen llevan el Tag 802.1Q. Se utilizan para conectar Switches entre si y que pase el tráfico de diferentes VLANs a través de ellos.



Diferencia Access y Trunk

2.2.4. ¿Por qué usar VLANS?

Supongase que un centro docente tiene claramente diferenciada las partes de la red que quiere separar en divisiones lógicas (VLANs): administración (VLAN 90), estudiantes (VLAN 100), cuerpo docente (VLAN 200), invitados (VLAN 300), divididas en tres edificios con 3 switches capa 2 y un switch capa 3.



Se recomienda el uso de VLANs para:

-Crear una división lógica de grupos de trabajo.

Por ejemplo, suponiendo que todos los equipos de la planta de un edificio están conectados mediante una red de área local con nodos. Se puede crear una VLAN para cada grupo de trabajo de la planta, es decir, cuerpo docente, estudiantes, administración e invitados.

-Designar diferentes directivas de seguridad para los grupos de trabajo.

Por ejemplo, las necesidades de seguridad del departamento de administración y del departamento de estudiantes son muy diferentes. Si los sistemas de ambos departamentos comparten la misma red local, se puede crear una red VLAN independiente para cada departamento. Después, se puede asignar la directiva de seguridad apropiada para cada VLAN.

-Dividir los grupos de trabajo en dominios de emisión administrables.

El uso de redes VLAN reduce el tamaño de los dominios de emisión y mejora la efectividad de la red.

-Reducción de costo.

El ahorro en el costo resulta de la poca necesidad de actualizaciones de red caras y más usos eficientes de enlaces y ancho de banda existente.

-Mejor rendimiento.

La división de las redes planas de Capa 2 en múltiples grupos lógicos de trabajo (dominios de broadcast) reduce el tráfico innecesario en la red y potencia el rendimiento.

-Mitigación de la tormenta de broadcast:

La división de una red en las VLAN reduce la cantidad de dispositivos que pueden participar en una tormenta de broadcast.

-Administración de aplicación o de proyectos más simples.

Las VLAN agregan dispositivos de red y usuarios para admitir los requerimientos geográficos o comerciales. Tener funciones separadas hace que gestionar un proyecto o trabajar con una aplicación especializada sea más fácil, por ejemplo una plataforma de desarrollo de e-learning para el cuerpo docente. También es fácil determinar el alcance de los efectos de la actualización de los servicios de red.

3. MARCO PRÁCTICO

3.1. Implementación de VLANs a nivel genérico

La implementación de VLANs por parte de otros fabricantes de equipos diferentes a MikroTik siguen una pauta muy clara. Por lo que se comprende para un usuario con una base en creación y configuración de VLANs, una forma estándar de establecer una solución:

1. Declaración de VLANs

Las VLANs son declaradas añadiendo:

- Un VLAN-ID (identificador), que corresponde con un numero entero entre el 0-4095. Algunos valores son reservados como el 0 (Contiene datos de prioridad de usuarios 802.1Q), 1 (Puerto por defecto de VID 802.1Q) y 4095 (Reservada 802.1Q).
- Un nombre (optativo), que sirve como ayuda al usuario para la identificación.

2. Asignación de puertos

Los puertos se asignan según sean modo access (untagged) o trunk (tagged). Para ello se selecciona la interfaz y se pone en el modo de puerto deseado. Si es acceso se indica a que VLAN pertenecen y si es en modo trunk, que VLANs va a permitir tráfico.

3. Enrutamiento

Si las VLANs desean ser enrutadas se deberá:

- Ponerle una dirección IP junto a su máscara de subred a la interfaz VLAN.
- Establecer rutas

Ejemplo declaración VLANs para CISCO y HP.

```
CISCO

1- Declaración de VLANs
switch# configure terminal
switch(config)# VLAN 100
switch(config-VLAN)# name Servidores
switch(config-VLAN)# state active
switch(config-VLAN)# no shutdown

2- Asignación de puertos
Access:
switch# configure terminal
switch(config)# interface ethernet 0/13
switch(config-if)# switchport access VLAN 100

Trunk:
switch# configure terminal
switch(config)# interface ethernet 0/1
switch(config-if)# switchport mode trunk
switch(config-if)# switchport trunk native VLAN 105
switch(config-if)# switchport trunk allow VLAN 100

3- Enrutamiento
switch# configure terminal
switch(config)# interface VLAN 100
switch(config-VLAN)# ip address 10.1.100.1 255.255.255.0
switch(config-VLAN)# no shutdown
switch(config-VLAN)# exit

switch# ip route 192.168.100.0 255.255.255.0 192.168.100.1
```

```
HP

1- Declaración de VLANs
Switch# conf t
Switch(config)# VLAN 100 name Servidores
Switch(config)# end

2- Asignación de puertos
Access:
Switch# conf t
Switch(config)# VLAN 100
Switch(VLAN-100)# untagged 1-11
Switch(VLAN-100)# exit

Trunk:
Switch# conf t
Switch(config)# VLAN 100
Switch(VLAN-100)# tagged 24
Switch(VLAN-100)# exit

3- Enrutamiento
Switch# conf t
Switch(config)# VLAN 100
Switch(VLAN-100)# ip address 10.1.100.1 255.255.255.0
Switch(VLAN-100)# exit

Switch(config)# ip route 192.168.100.0 255.255.255.0 192.168.100.1
```

3.2. Implementación de VLANs en MikroTik (Sin usar la API)

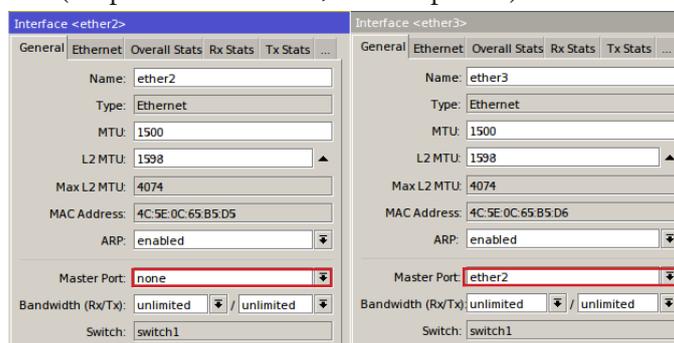
Como se ha comprobado en el apartado anterior, hay una proximidad a la hora de establecer soluciones por parte de otros fabricantes. Se observará ahora la disparidad entre MikroTik con el resto de fabricantes e incluso entre los propios dispositivos MikroTik, haciendo que la tarea sea compleja y no intuitiva. Los pasos a seguir se dividirán según ambos tipos de dispositivos:

ROUTERBOARD

1) Creación del switch

Primero se deberá comprobar que todos los puertos que queremos conmutar pertenezcan al mismo switch, es decir, estén conectados al mismo chipset. Para incluir a todos los puertos en el mismo Switch se deberá establecer un Master-Port (puerto maestro) y cualquier puerto cuyo master-port sea el mismo, formará parte del mismo Switch junto al master-port original.

Configuración visual (En pestaña Interfaces, en cada puerto):



Terminal:

Creación de Master-Port:

```
interface ethernet set master-port=none ether1
```

Añadir puertos al Switch que tiene como master-port ether1

```
interface ethernet set master-port=ether2 ether1
```

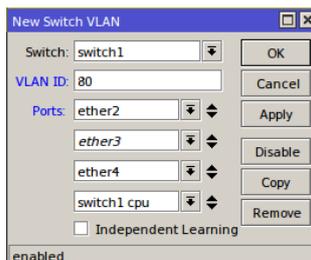
```
interface ethernet set master-port=ether3 ether1
```

2) Declaración de VLANs

Las VLANs son declaradas añadiendo:

- Un VLAN-ID (identificador), que corresponde con un numero entero entre el 0-4095.
- Un nombre (optativo), que sirve como ayuda al usuario para la identificación.
- Los puertos que van a formar parte de la VLAN. Además si se requieren funcionalidades via software como inter-VLAN routing o firewalling, se deberá incluir el puerto (switchX-cpu) donde X corresponde al número del switch.
- Además, para RouterBoard se deberá incluir también a qué switch pertenecen los puertos ya que dependiendo de la arquitectura hardware, pueden haber varios chips de switch en el mismo equipo.

Configuración visual (En pestaña Switch- VLAN, añadir):



Terminal:

Creación de VLAN:

```
interface ethernet switch VLAN add VLAN-id=80 ports=ether2,ether3,ether4,switch1-cpu
```

```
switch=switch1
```

3)Asignación de puertos

Los puertos se asignan según sean modo access (untagged) o trunk (tagged). Para ello se selecciona la interfaz y se pone en el modo de puerto deseado.

Access:

- "VLAN Mode: secure", se rechazarán los paquetes con un tag VLAN que no esté presente en la tabla VLAN. Los paquetes con tags VLANs que estén presentes en la tabla VLAN, pero cuyo puerto de entrada no coincide con cualquier puerto de entrada de la tabla de VLAN serán descartados también.
- "VLAN Header: always-strip", si la cabecera VLAN está presente, la cabecera se elimina del paquete.
- "Default VLAN ID", VLAN de acceso.

Configuración visual (En pestaña Switch- Ports, click en puerto deseado):



Terminal:

Puerto modo Access:

```
interface ethernet switch port set ether3 VLAN-mode=secure VLAN-header=always-strip default-VLAN-id=80
```

Trunk:

-No se establece que VLANs desea permitir el Trunk. El Trunk dejará pasar las VLANs en las que ese puerto está incluido.

- "VLAN Mode: secure", se rechazarán los paquetes con un tag VLAN que no esté presente en la tabla VLAN. Los paquetes con tags VLANs que estén presentes en la tabla VLAN, pero cuyo puerto de entrada no coincide con cualquier puerto de entrada de la tabla de VLAN serán descartados también.

- "VLAN Header: add-if-missing", si la cabecera VLAN no está presente, la cabecera se añadirá al paquete.

- "Default VLAN ID", VLAN nativa.

Configuración visual (En pestaña Switch- Ports, click en puerto deseado):



Terminal:

Puerto modo Trunk:

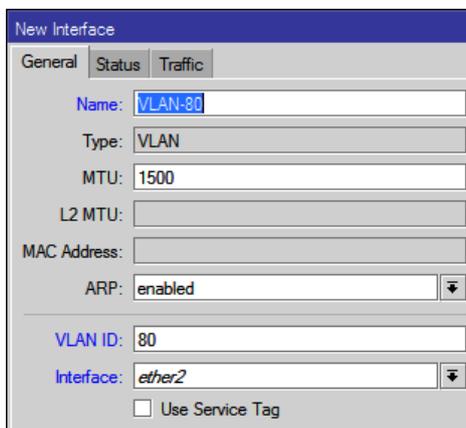
```
interface ethernet switch port set ether2 VLAN-mode=secure VLAN-header=add-if-missing
```

4) Enrutamiento

Para que las VLANs sean enrutadas se deberá:

- Añadir el switch-cpu en las VLAN que se quieran enrutar (Consultar paso 2)
- En los interfaces físicos, añadir las VLAN que vengan por el trunk (Es necesario establecer de qué interfaz cuelgan las VLANs, en el ejemplo cuelgan de ether2 que es master-port):

Configuración visual (En pestaña Interfaces-VLAN, añadir):

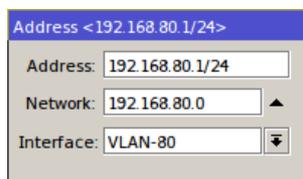


Terminal:

```
interface VLAN add interface=ether2 VLAN-id=80 name=VLAN-80
```

- Establecer dirección IP y máscara de subred por cada interfaz VLAN.

Configuración visual (En pestaña IP-Addresses, añadir):

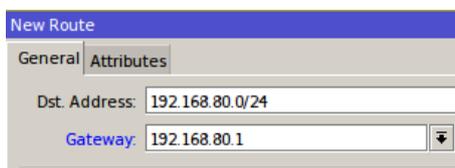


Terminal:

```
ip address add address=192.168.80.1/24 interface=VLAN-80
```

- Establecer Rutas

Configuración visual (En pestaña IP-Routes, añadir):



Terminal:

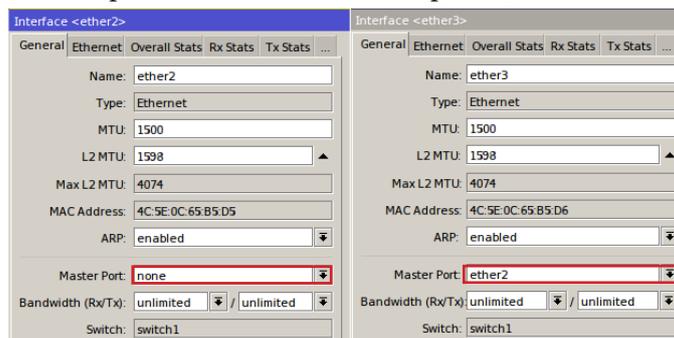
```
ip route add dst-address=192.168.80.0 gateway=192.168.80.1
```

CLOUD ROUTER SWITCH

1) Creación del switch

Primero se deberá comprobar que todos los puertos que queremos conmutar pertenezcan al mismo switch, es decir, estén conectados al mismo chipset. Para incluir a todos los puertos en el mismo Switch se deberá establecer un Master-Port (puerto maestro) y cualquier puerto cuyo master-port sea el mismo, formará parte del mismo Switch junto al master-port original.

Configuración visual (En pestaña Interfaces, en cada puerto):



Terminal:

Creación de Master-Port:

```
interface ethernet set master-port=none ether1
```

Añadir puertos al Switch que tiene como master-port ether1

```
interface ethernet set master-port=ether2 ether1
```

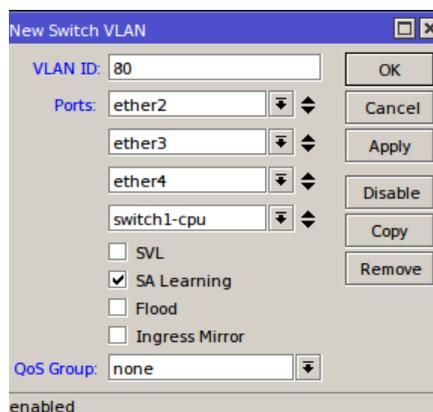
```
interface ethernet set master-port=ether3 ether1
```

2) Declaración de VLANs

Las VLANs son declaradas añadiendo:

- Un VLAN-ID (identificador), que corresponde con un numero entero entre el 0-4095.
- Un nombre (optativo), que sirve como ayuda al usuario para la identificación.
- Añadir SA Learning (Aprender la MAC de origen después de la traducción VLAN).
- Los puertos que van a formar parte de la VLAN. Además si se requieren funcionalidades via software como inter-VLAN routing o firewalling, se deberá incluir el puerto (switchX-cpu) donde X corresponde al número del switch.

Configuración visual (En pestaña Switch- VLAN- VLAN, añadir):



Terminal:

Creación de VLAN:

```
interface ethernet switch VLAN add VLAN-id=80 ports=ether2,ether3,ether4,switch1-cpu
```

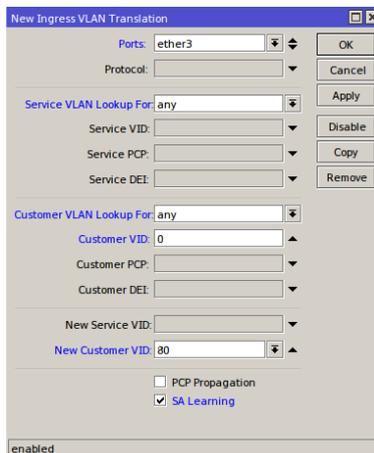
learn=yes

3)Asignación de puertos

Los puertos se asignan según sean modo access (untagged) o trunk (tagged). Para ello se selecciona la interfaz y se pone en el modo de puerto deseado.

Access (Añadir SA Learning (Aprender la MAC de origen después de la traducción VLAN)):

Configuración visual (En pestaña Switch- VLAN- In.VLAN Tran., añadir):

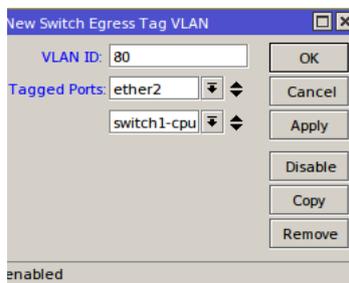


Terminal:

```
interface ethernet switch ingress-VLAN-translation add ports=ether3 sa-learning=yes  
new-customer-vid=80
```

Trunk (En pestaña Switch- VLAN- VLAN tagging, añadir):

Configuración visual:



Terminal:

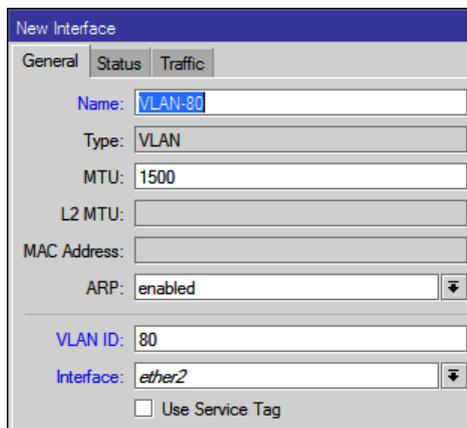
```
interface ethernet switch egress-VLAN-tag add VLAN-id=80 tagged-ports=ether2, switch1-  
cpu
```

4) Enrutamiento

Para que las VLANs sean enrutadas se deberá:

- Añadir el switch-cpu en las VLAN que se quieran enrutar (Consultar paso 2)
- En los interfaces físicos, añadir las VLAN que vengan por el trunk (Es necesario establecer de qué interfaz cuelgan las VLANs, en el ejemplo cuelgan de ether2 que es master-port):

Configuración visual (En pestaña Interfaces-VLAN, añadir):

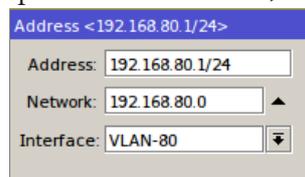


Terminal:

```
interface VLAN add interface=ether2 VLAN-id=80 name=VLAN-80
```

- Establecer dirección IP y máscara de subred por cada interfaz VLAN.

Configuración visual (En pestaña IP-Addresses, añadir):

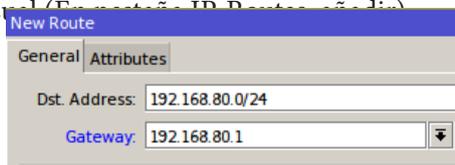


Terminal:

```
ip address add address=192.168.80.1/24 interface=VLAN-80
```

- Establecer Rutas

Configuración visual (En pestaña IP-Routes, añadir):



Terminal:

```
ip route add dst-address=192.168.80.0 gateway=192.168.80.1
```

3.3. Implementación de VLANs en MikroTik (Usando la API)

A diferencia del apartado anterior, no existe disparidad entre MikroTik con el resto de fabricantes e incluso entre los propios dispositivos MikroTik, haciendo que la tarea sea más sencilla e intuitiva.

Los pasos a seguir son:

1) Creación del switch

Primero se deberá comprobar que todos los puertos pertenecen al mismo Switch. Para incluir a todos los puertos en el mismo Switch se deberá establecer un Master-Port (puerto maestro) y cualquier puerto cuyo master-port sea el mismo, formará parte del mismo Switch junto al master-port original.

Configuración:

Switch	Port	Master-Port
Switch 1	ether1	none
ether1 (Master-port)	ether2	none
Switch 2	ether3	none
ether2 (Master-port)	ether4	ether1
ether3	ether5	ether2
ether4	ether6	ether3
ether5	ether7	ether4
sfp1	ether8	ether5
Switch 3	ether9	ether6
ether6 (Master-port)	ether10	ether7
ether7	sfp1	ether8
ether8	ether6	ether9
ether9	sfp1	ether10
	ether2	ether6

Eligiendo cada master-port de cada puerto según el menú desplegable individual, se irán formando los diferentes switches.

2) Declaración de VLANs

Las VLANs se crearán automáticamente junto a los puertos que pertenezcan según se vayan estableciendo los puertos en modo Access y Trunk. De este modo se agiliza la tarea de creación e implementación de VLANs y puertos, uniendo ambos pasos automáticamente.

3) Asignación de puertos

Los puertos se asignan según sean modo access (untagged) o trunk (tagged). Para ello se selecciona la interfaz y se pone en el modo de puerto deseado.

Access:

Configuración (Se elige la VLAN de acceso):

Select interfaces:
ether3

Access Trunk No Switchport

Access Vlan: 80

Submit

Trunk:

Configuración (Se elige las VLANs que pasan por el Trunk):

Select interfaces:
ether2

Access Trunk No Switchport

Allowed VLANs: 80, 90, 100

Submit

4) Enrutamiento

Para que las VLANs sean enrutadas se deberá:

- Se establece el nombre de la interfaz VLAN, la dirección IP y de que interfaz cuelgan.

Configuración:

VLAN Interface

Master interface:
ether2 (Master Port)

VLAN ID:
80

VLAN Name:
VLAN 80

IP Address:
192.168.80.1/24

Submit

- Se establecen rutas si no se añaden dinámicamente

Configuración:

Routes

Destination Address:
192.168.80.0/24

Gateway:
192.168.80.1

Submit

3.4. Establecimiento servidor LAMP

Antes de empezar a establecer conexiones de la API con los equipos MikroTik, se necesitó configurar y poner a punto un servidor Apache capaz de ejecutar código PHP. Para ello se decidió instalar un servidor LAMP (Linux, Apache, MySQL y PHP) en el sistema operativo Linux, debido a su rendimiento.

El procedimiento se divide en tres partes: Instalar y probar Apache e instalar y probar PHP (MySQL no se explica ya que no se utilizó).

APACHE

Instalación:

En un terminal, ingresar el siguiente comando:

```
sudo apt-get install apache2
```

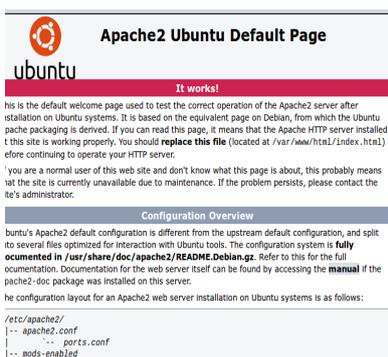
El directorio donde se tiene que almacenar los sitios web es: /var/www

Para poder hacerlo, es necesario otorgar a tu usuario los privilegios necesarios. El siguiente comando suele ser el adecuado en la mayoría de los casos, aunque eso puede variar dependiendo del grado de seguridad que se necesite en el servidor web:

```
sudo chmod -R 775 /var/www
```

Prueba:

Ingresar <http://localhost> en el explorador web. Se debe ver una página con la información de Apache.



PHP

Instalación:

En un terminal, ingresar el siguiente comando:

```
sudo apt-get install php5 libapache2-mod-php5 php5-cli php5-mysql
```

Prueba:

Para probar que se haya instalado correctamente, se creó un script PHP muy simple y se guardó en /var/www/ con el nombre de prueba.php:

```
sudo gedit /var/www/prueba.php
```

Ingresar el siguiente contenido y guardá el archivo:

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

Para ejecutar el script, se abrirá el explorador web y se accederá a la siguiente URL: <http://localhost/prueba.php>. Se verá una página con información sobre tu instalación de PHP.

PHP Version 5.5.9-1ubuntu4.9	
System	Linux Jorge-PC 3.16.0-38-generic #52-14.04.1-Ubuntu SMP Fri May 8 09:43:57 UTC 2015 x86_64
Build Date	Apr 17 2015 11:41:17
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php5/apache2
Loaded Configuration File	/etc/php5/apache2/php.ini
Scan this dir for additional .ini files	/etc/php5/apache2/conf.d
Additional .ini files parsed	/etc/php5/apache2/conf.d/05-opcache.ini, /etc/php5/apache2/conf.d/05-pdo.ini, /etc/php5/apache2/conf.d/20-gd.ini, /etc/php5/apache2/conf.d/20-ldap.ini, /etc/php5/apache2/conf.d/20-mysql.ini, /etc/php5/apache2/conf.d/20-pdo_mysql.ini, /etc/php5/apache2/conf.d/20-readline.ini

3.5. Conexiones con la API MikroTik

Una vez establecido el servidor Apache capaz de ejecutar scripts PHP y antes de empezar a realizar la API Web de Switching, se comprobó y estableció conexiones de PHP con los equipos MikroTik.

Para ello se utilizó la API PHP Class proporcionada por MikroTik (http://wiki.MikroTik.com/wiki/API_PHP_class) y se comprobó algunos de sus ejemplos de conexión:

Ejemplo 1:

Se creó un script php llamado ejemplo1.php y se guardó en /var/www, utilizando el siguiente comando:
`sudo gedit /var/www/ejemplo1.php.`

Dentro de este fichero se incluyó el siguiente contenido (Obsérvese que las letras en rojo serán rellenadas con la información de cada equipo):

```
<?php
require('routeros_api.class.php');

$API = new routeros_api();

$API->debug = true;

if ($API->connect('IP EQUIPO MikroTik', 'LOGIN', 'PASSWORD')) {

    $API->write('/interface/getall');

    $READ = $API->read(false);
    $ARRAY = $API->parse_response($READ);

    print_r($ARRAY);

    $API->disconnect();

}

?>
```

Posteriormente, se accedió a la página web ingresando en un navegador la url: `localhost/ejemplo1.php`, observando la siguiente salida correspondiente al listado de todos los interfaces. Comprobando así que la API conecta correctamente y nos devuelve información real del equipo:

```
Connection attempt #1 to 192.168.88.1:8728... <<< [6] /login >>> [5/5] bytes read. >>> [5, 39]ldone >>> [37/37] bytes read. >>> [37, 1]=ret=e87929b19fe21117f35be83396e7376b <<< [6] /login <<< [11]
=name=admin <<< [44] =response=0078cf29c2d827a95b1a5835eaf3fd4c4 >>> [5/5] bytes read. >>> [5, 1]ldone Connected... <<< [17] /interface/getall >>> [3/3] bytes read. >>> [3, 4340]re >>> [7/7] bytes
read. >>> [7, 4332]=id=*2 >>> [12/12] bytes read. >>> [12, 4319]=name=ether1 >>> [20/20] bytes read. >>> [20, 4298]=default-name=ether1 >>> [11/11] bytes read. >>> [11, 4286]=type=ether >>> [9/9] bytes
read. >>> [9, 4276]=mtu=1500 >>> [16/16] bytes read. >>> [16, 4259]=actual-mtu=1500 >>> [11/11] bytes read. >>> [11, 4247]=l2mtu=1598 >>> [15/15] bytes read. >>> [15, 4231]=max-l2mtu=4074 >>>
[30/30] bytes read. >>> [30, 4200]=mac-address=4C:5E:0C:65:B5:D4 >>> [15/15] bytes read. >>> [15, 4184]=fast-path=true >>> [41/41] bytes read. >>> [41, 4142]=last-link-down-time=jun/29/2015 09:37:24
>>> [39/39] bytes read. >>> [39, 4102]=last-link-up-time=jun/29/2015 09:37:27 >>> [13/13] bytes read. >>> [13, 4088]=link-downs=2 >>> [20/20] bytes read. >>> [20, 4067]=rx-byte=28832056163 >>>
[19/19] bytes read. >>> [19, 4047]=tx-byte=2534839149 >>> [19/19] bytes read. >>> [19, 4027]=rx-packet=22885737 >>> [19/19] bytes read. >>> [19, 4007]=tx-packet=12048819 >>> [10/10] bytes read. >>>
[10, 3996]=rx-drop=0 >>> [10/10] bytes read. >>> [10, 3985]=tx-drop=0 >>> [11/11] bytes read. >>> [11, 3973]=rx-error=0 >>> [11/11] bytes read. >>> [11, 3961]=tx-error=0 >>> [13/13] bytes read. >>> [13,
3947]=running=true >>> [15/15] bytes read. >>> [15, 3931]=disabled=false >>> [3/3] bytes read. >>> [3, 3926]re >>> [7/7] bytes read. >>> [7, 3918]=id=*3 >>> [12/12] bytes read. >>> [12,
3905]=name=ether2 >>> [20/20] bytes read. >>> [20, 3884]=default-name=ether2 >>> [11/11] bytes read. >>> [11, 3872]=type=ether >>> [9/9] bytes read. >>> [9, 3862]=mtu=1500 >>> [16/16] bytes read. >>>
[16, 3845]=actual-mtu=1500 >>> [11/11] bytes read. >>> [11, 3833]=l2mtu=1598 >>> [15/15] bytes read. >>> [15, 3817]=max-l2mtu=4074 >>> [30/30] bytes read. >>> [30, 3786]=mac-
address=4C:5E:0C:65:B5:D5 >>> [15/15] bytes read. >>> [15, 3770]=fast-path=true >>> [39/39] bytes read. >>> [39, 3730]=last-link-up-time=jul/01/2015 11:27:09 >>> [13/13] bytes read. >>> [13, 3716]=link-
downs=0 >>> [10/10] bytes read. >>> [10, 3705]=rx-byte=0 >>> [16/16] bytes read. >>> [16, 3688]=tx-byte=3904966 >>> [12/12] bytes read. >>> [12, 3675]=rx-packet=0 >>> [15/15] bytes read. >>> [15,
```

Información de los puertos mostrada por la API MikroTik

3.6. Código

Comprobado que tenemos acceso a los equipos mediante PHP, se empezó a programar la API Web para la creación de VLANs. A continuación se explicarán algunos de los fragmentos de código que se consideran más importantes, dejando de lado parte de código que, siendo también importante, extendería el documento y restaría importancia a partes más importantes de código.

Si se desea consultar el código, se encuentra disponible en github.com/jorlarui/Mikrotik-Cloud-Switching.

El código principal se divide en 6 archivos PHP correspondiente a las vistas de la API principalmente que serán analizados posteriormente:

- Index
- Status
- Switch
- VLANs
- Routing
- ACLs

Aunque el proyecto final tiene la siguiente estructura de árbol:

```
+bootstrap
+css
+highchart
+images
+jquery
-ACLs.php
-Help.html
-README.md
-Routing.php
-Status.php
-Switch.php
-VLANs.php
-datosACLs.php
-datosCPU.php
-datosGraficas.php
-datosRouting.php
-datosStatus.php
-datosStatusImage.php
-datosSwitch.php
-datosSwitchImage.php
-datosVLANs.php
-datosVLANsImage.php
-index.php
-routeros_api.class.php
```

3.6.1. Index

Este código PHP muestra un formulario para el establecimiento de conexión, junto con un footer con la información de la API.

El usuario ingresará la IP del equipo que desea conectar, el login y el password y una vez hecho click en 'Submit', estos valores pasarán a la sesión, haciendo que el usuario ya no tenga que volver a introducir sus datos mientras la sesión siga activa.



Vista de index.php

Código:

Importación de librerías e inicio de sesión para almacenamiento de variables:

```
<?php ob_start(); session_start(); require('routeros_api.class.php'); ?>
```

Establecimiento de conexión con la IP y eliminación de sesión en caso de Log Out:

```
<?php
    $API = new routeros_api();           //Se inicia una nueva conexión API MikroTik
    $IP = $_POST['ip'];                  //Se obtiene la IP, user y password del formulario MikroTik
    $user = $_POST['user'];
    $password = $_POST['password'];
    $notLogin = $_GET["logOut"];        //Se obtiene si el usuario ha accedido a Index después de click en Log
Out

    $_SESSION[ 'ip' ]=$IP;              //Se envian para la sesión la IP, el usuario y el password.
    $_SESSION[ 'user' ]=$user;
    $_SESSION[ 'password' ]=$password;

    $buttonClick = isset($_POST['send']);

    if( $buttonClick){
        if ($API->connect($IP, $user, $password)) {
            //Si el Login es correcto, se redirige a Status.php
            header( 'Location:Status.php' );
        }
        else {
            echo "<div class='error'>Error: Login incorrecto.</div>";
        }
        //Sino se imprime 'Login incorrecto'
    }

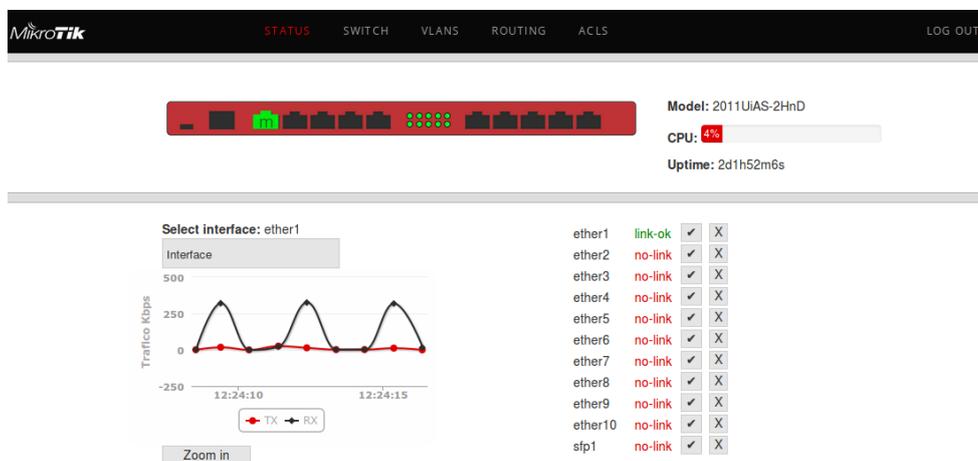
    if ($notLogin){
        //Si el usuario viene de hacer click en Log Out, la sesión se elimina.
        session_unset();
    }
?>
```

3.6.2. Status

Este código PHP muestra dos ventanas principalmente:

La ventana superior muestra información del equipo según el estado (imagen del equipo junto a puertos activos, carga de CPU, tiempo de actividad y nombre del modelo).

La ventana inferior permite seguir el tráfico en una gráfica y activar o desactivar los puertos.



Vista de Status.php

Código:

Comprobación de conexión y obtención de variables:

```
<?php
    $API = new routeros_api();
    $IP = $_SESSION[ 'ip' ];
    $user = $_SESSION[ 'user' ];
    $password = $_SESSION[ 'password' ];
    //Comprobamos conexion API
    if ($API->connect($IP, $user, $password)) {

        //Obtenemos información interfaces ethernet
        $Ports = $API->comm("/interface/ethernet/print");
        $numPorts = count($Ports);

        //Obtenemos información del modelo del equipo
        $modeloCom = $API->comm("/system/routerboard/print");
        $modelo=$modeloCom[0][ 'model' ];

        //Obtenemos información de la CPU
        $cpuInfo = $API->comm("/system/resource/print");

        //Obtenemos estado del enlace por cada puerto
        $valoresPar= json_encode(range(0, $numPorts-1));
        $valores = substr($valoresPar, 1, -1);
        $API->write("/interface/ethernet/monitor",false);
        $API->write("=numbers=".$valores,false);
        $API->write("=once=",true);
        $READ = $API->read(false);
        $statusPorts = $API->parse_response($READ);
        $API->disconnect();
    }

    //Si la API no conectara, volvería a index.php
    else {
        header( 'Location:index.php?notLogin=true' );
    }
?>
```

Muestra de información de puertos y su estado:

```
<?php
    for ($cont = 0; $cont < $numPorts; $cont++){ //Recorre todos los puertos
        echo "<tr>";
        echo "<td>".$Ports[$cont]['name'].</td>";
        if($statusPorts[$cont]['status']=='link-ok'){ //Si el puerto está el estado pertenecerá a la
class 'link-ok'
            echo "<td class='link-ok'>";
        }
        else if($statusPorts[$cont]['status']=='no-link'){ //Si el puerto está inactivo el estado
pertenecerá a la clase 'no-link'
            echo "<td class='no-link'>";
        }
        echo $statusPorts[$cont]['status'].</td>"; //muestra el estado del puerto}

    echo "</tr>";
?>
```

Activación/ Desactivación de puertos:

```
<?php
    for ($cont = 0; $cont < $numPorts; $cont++){//Recorre todos los puertos
        if(isset($_POST['enablePort.$cont'])){ //Si algún botón correspondiente al puerto ha sido click-
ado entrará a la condición
            $API = new routeros_api();
            $IP = $_SESSION[ 'ip' ];
            $user = $_SESSION[ 'user' ];
            $password = $_SESSION[ 'password' ];
            if ($API->connect($IP, $user, $password)) {
                $API->write("/interface/ethernet/set",false); //Comando para editar la inter-
faz en MikroTik
                $API->write("=disabled=no",false); //Activar el puerto
                $API->write("=.id=".$Ports[$cont]['name']); //Que puerto debe ser activado
                $Ports = $API->read();
                $API->disconnect();
            }
        }

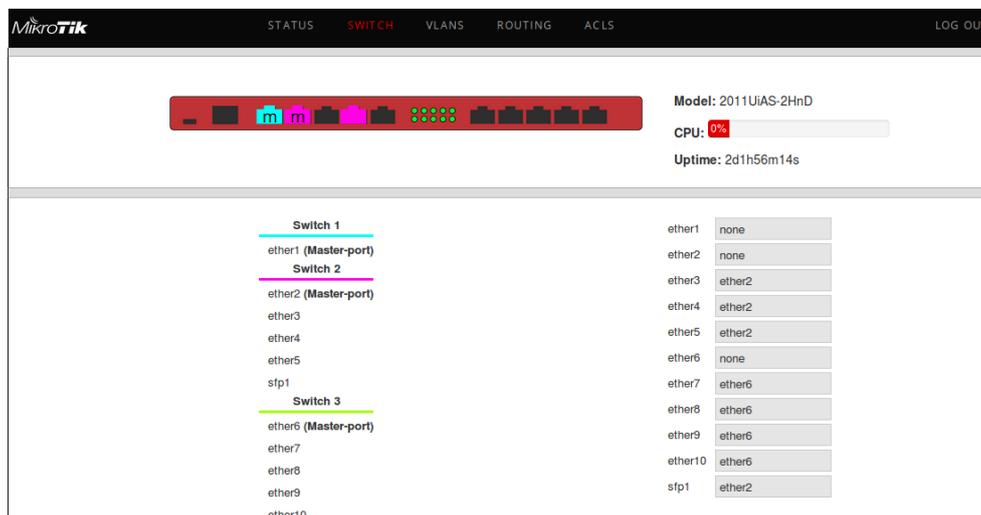
    for ($cont = 0; $cont < $numPorts; $cont++){ //Recorre todos los puertos
        if(isset($_POST['disablePort.$cont'])){ //Si algún botón correspondiente al puerto ha sido click-
ado pasará a desactivarlo
            $API = new routeros_api();
            $IP = $_SESSION[ 'ip' ];
            $user = $_SESSION[ 'user' ];
            $password = $_SESSION[ 'password' ];
            if ($API->connect($IP, $user, $password)) {
                $API->write("/interface/ethernet/set",false); //Comando para editar la inter-
faz en MikroTik
                $API->write("=disabled=yes",false); //Desactivar el puerto
                $API->write("=.id=".$Ports[$cont]['name']); //Que puerto debe ser desactiva-
do
                $Ports = $API->read();
                $API->disconnect();
            }
        }
    }
?>
```

3.6.3. Switch

Este código PHP muestra dos ventanas principalmente:

La ventana superior muestra información del equipo según el switch (imagen del equipo junto a puertos activos coloreados según el switch que pertenecen, carga de CPU, tiempo de actividad y nombre del modelo).

La ventana inferior permite comprobar los puertos que forman los switches y cambiar la configuración de éstos.



Vista de Switch.php

Código:

Configuración de switches eligiendo master-port según cada puerto:

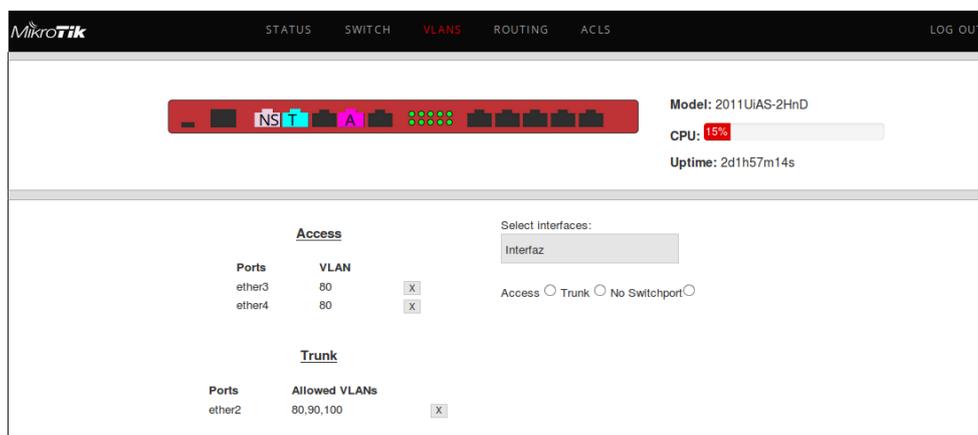
```
<?php
//Recorre todos los puertos
for ($cont = 0; $cont < $numPorts; $cont++){
//Si se ha elegido el cambio de algún puerto
if(isset($_POST['formMaster'.$cont])){
//Obtener del formulario el master-port que desea ser cambiado en el puerto
$seleccion = $_POST['formMaster'.$cont];
//Crear conexión API con el equipo
$API = new routeros_api();
$IP = $_SESSION['ip'];
$user = $_SESSION['user'];
$password = $_SESSION['password'];
//Si se conecta con el equipo
if ($API->connect($IP, $user, $password)) {
//Editar el puerto
$API->write("/interface/ethernet/set",false);
//Establecer el master-port del puerto por el seleccionado en el formulario
$API->write("=master-port=".$seleccion,false);
$API->write("=.id=".$Ports[$cont]['name']);
$ARRAY = $API->read();
$API->disconnect();
}
}
}
?>
```

3.6.4. VLANs

Este código PHP muestra dos ventanas principalmente:

La ventana superior muestra información del equipo según las VLANs (imagen de los puertos del equipo según si el puerto está en modo Access, Trunk o No Switchport, carga de CPU, tiempo de actividad y nombre del modelo).

La ventana inferior permite observar la configuración de cada puerto (Access o Trunk) y que VLANs pertenecen.



Vista de VLANs.php

Código:

Al ser la parte de código más extensa, se explicará como se establecen los puertos en modo Access, Trunk o No Switchport para RouterBoard y Cloud Router Switch sin la parte de código, sólo su metodología:

ACCESS:

Comprobar si el puerto en cuestión existe en otra VLAN. Si existe, eliminar el puerto de la VLAN, dejando los otros puertos.

Si el puerto era el único puerto de la VLAN, eliminar la VLAN.

Si la VLAN no existía, crear una nueva VLAN con el puerto.

Si es RouterBoard se añadirá el switch automáticamente, junto a “switch-cpu” y se establecerá “switch-cpu” en modo “fallback” para permitir el tráfico de paquetes marcados con VLANs.

Si es Cloud Router Switch, se añadirá el “switch-cpu” automáticamente para permitir el tráfico entre el propio switch.

Si el puerto estaba previamente en modo Trunk, eliminar cualquier configuración previa perteneciente a Trunk.

TRUNK:

Comprobar si el puerto en cuestión existe en otra VLAN. Si existe, eliminar el puerto de la VLAN, dejando los otros puertos.

Si el puerto era el único puerto de la VLAN, eliminar la VLAN.

Si la VLAN no existía, crear nuevas VLANs por cada VLAN perteneciente al Trunk.

Si es RouterBoard se añadirá el switch automáticamente, junto a “switch-cpu” y se establecerá “switch-cpu” en modo “fallback” para permitir el tráfico de paquetes marcados con VLANs.

Si es Cloud Router Switch, se añadirá el “switch-cpu” automáticamente para permitir el tráfico entre el propio switch.

Si el puerto estaba previamente en modo Access, eliminar cualquier configuración previa perteneciente a Trunk.

NO SWITCHPORT

Comprobar si el puerto en cuestión existe en otra VLAN. Si existe, eliminar el puerto de la VLAN, dejando los otros puertos.

Si el puerto era el único puerto de la VLAN, eliminar la VLAN.

Si es RouterBoard, editar el puerto configurarlo como:

VLAN-mode: disabled

VLAN-header: leave-as-is

default-VLAN-id: 0

Si es Cloud Router Switch:

Comprobar si el puerto era Trunk y eliminarlo de Egress-VLAN-Tag

Comprobar si el puerto era Access y eliminarlo de Ingress-VLAN-Translation

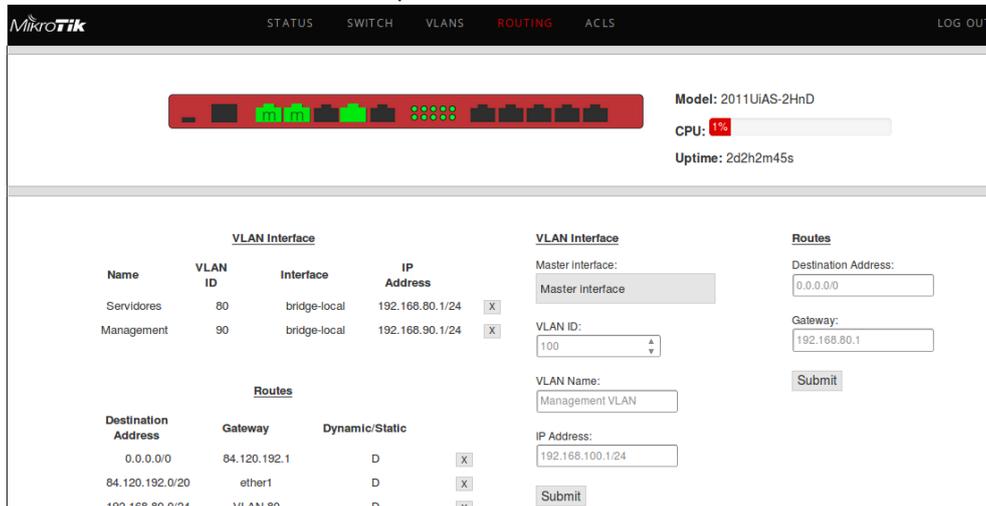
Poner IP al puerto en caso de que se haya seleccionado.

3.6.5. Routing

Este código PHP muestra dos ventanas principalmente:

La ventana superior muestra información del equipo según el estado (imagen del equipo junto a puertos activos, carga de CPU, tiempo de actividad y nombre del modelo).

La ventana inferior permite observar las interfaces VLAN creadas para el enrutamiento junto a las rutas. Además permite la creación de nuevas interfaces y de nuevas rutas.



Vista de Routing.php

Código:

Creación de interfaces y de rutas:

```
<?php
//Obtención de variables según el formulario para
creación de interfaces o rutas
$interfaz = $_POST['interfaz'];
$VLANID = $_POST['VLANID'];
$VLANName = $_POST['VLANName'];
$VLANAddress = $_POST['VLANAddress'];
$dstAddress = $_POST['dstAddress'];
$gateway = $_POST['gateway'];
//Si se ha pulsado el botón para creación de interfaces
if(isset($_POST['submitButton'])) {
    $API = new routeros_api();
    if ($API->connect($IP, $user, $password)) {
//Crear una nueva interfaz VLAN
        $API->comm("/interface/VLAN/
add", array(
//Interfaz maestra
            "interface" => $interfaz,
//Identificador de la VLAN
            "VLAN-id" => $VLANID,
//Nombre en formato "VLAN- <Identificador>"
            "name" => "VLAN-".$VLANID,
//Comentario con el nombre de la VLAN deseado
            "comment" => $VLAN-
Name,
        ));
    }
```

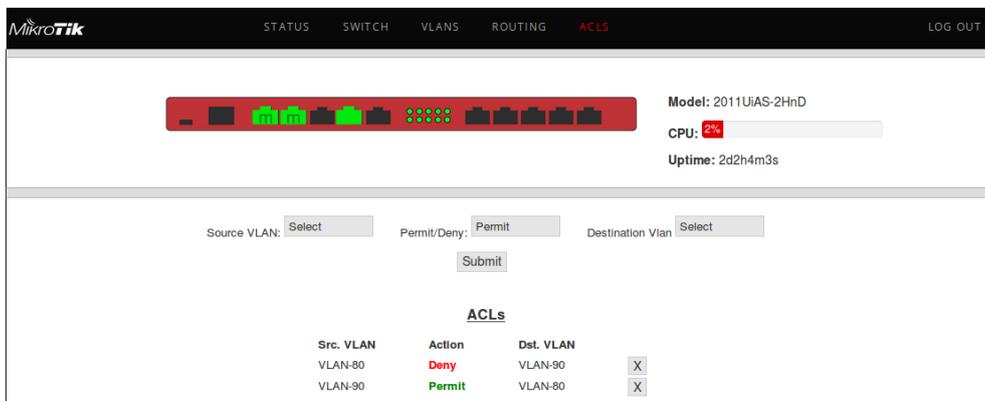
```
// Establecer dirección IP
$API->comm("/ip/address/add", array(
// Dirección IP junto a máscara de subred
    "address" => $VLANAddress,
// Interfaz VLAN a la que se asigna la IP
    "interface" => "VLAN-".$VLANID,
));
$API->disconnect();
}
//Si se ha pulsado el botón para creación de rutas
if(isset($_POST['submitButton2'])) {
    $API = new routeros_api();
    if ($API->connect($IP, $user, $password)) {
// Crear una nueva ruta
        $API->comm("/ip/route/add", array(
// Dirección de destino
            "dst-address" => $dstAddress,
// Puerta de enlace
            "gateway" => $gateway,));
$API->disconnect();}
?>
```

3.6.6. ACLs

Este código PHP muestra dos ventanas principalmente:

La ventana superior muestra información del equipo según el estado (imagen del equipo junto a puertos activos, carga de CPU, tiempo de actividad y nombre del modelo).

La ventana inferior permite observar y crear las listas de acceso. Es decir, permitir tráfico o denegar entre VLANs.



Vista de ACLs.php

Código:

Creación y eliminación de reglas de acceso:

```
<?php
//Obtención de variables
$permitDeny = $_POST['permitDeny'];
$sourceVLAN= $_POST['sourceVLAN'];
$destinationVLAN= $_POST['destinationVLAN'];
//Si se ha pulsado el botón para creación de reglas

if(isset($_POST['submitButton'])){
    $API = new routeros_api();
    if ($API->connect($IP, $user, $password)) {
        // Añadir regla de acceso
        $API->comm("/ip/firewall/filter/
add", array(
            //Cadena 'Forward' por defecto
            "chain" => "forward",
            // Accion dependerá de Permitir/
            // Denegar
            "action" => $permitDeny,
            // VLAN de origen
            "in-interface" => $sourceV-
LAN,
            //VLAN de destino
            "out-interface" =>
$destinationVLAN,
        ));
        $API->disconnect();
    }
}
```

```
//Recorremos todas las reglas
for ($cont = 0; $cont < $numFirewall; $cont++){
//Si se ha pulsado el botón para eliminación de reglas
if(isset($_POST['disableRule'.$cont])){
    $API = new routeros_api();
    $IP = $_SESSION[ 'ip' ];
    $user = $_SESSION[ 'user' ];
    $password = $_SESSION[ 'password' ];
    if ($API->connect($IP, $user, $pass-
word)) {
        //Eliminar regla de acceso
        $API->write("/ip/firewall/filter/re-
move",false);
        $API->write("=.id=".$cont);
        $API->disconnect();
    }
}
?>
```

4. CONCLUSIONES

4.1. Conclusiones

Una vez finalizado el proyecto, se puede concluir que el resultado ha sido satisfactorio, ya de forma personal como objeto de utilidad.

A nivel personal se ha ampliado y aplicado el conocimiento en apartado de redes de interconexión, concretamente en Switching, creación y configuración de VLANs a nivel general y del protocolo utilizado por éstas. Además de una gran mejora y aprendizaje en programación (HTML, CSS, PHP, JQUERY, Bootstrapping...), junto a maneras y organización para desarrollar un proyecto de gran envergadura.

A nivel de objeto de utilidad, se ha conseguido que el proyecto tenga un fin y sirva para establecer una herramienta de ayuda a un usuario final. Para ello se ha estudiado y profundizado la forma que tienen MikroTik para crear VLANs y se ha establecido una solución cercana al resto de fabricantes, simplificando así la configuración a un usuario final.

En cuanto al desarrollo del proyecto y las dificultades encontradas, han sido principalmente la búsqueda de simplicidad continua, teniendo que prescindir de partes que se consideraban imprescindibles y estableciendo nuevas soluciones y algoritmos para que el usuario final tuviera una mejor experiencia. También, la falta de documentación ha complicado el desarrollo, ya que la gran diversidad de opciones que ofrece MikroTik dificulta la comprensión en su manera de operar. Por lo que se tuvo que comprobar muchas de las opciones de configuración posibles para ir estableciendo las soluciones adecuadas.

Para concluir este apartado, observamos que el proyecto puede ser puesto en práctica y utilizado por una comunidad global, teniendo la posibilidad de ser mejorado y editado por la mayoría de usuarios ya que es de código libre.

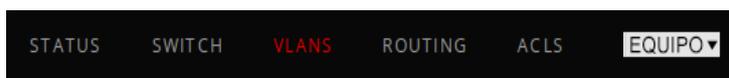
Teniendo esta meta y resultados, concluimos que el proyecto ha sido altamente satisfactorio y los objetivos cumplidos.

4.2. Futuras mejoras

Como futuras mejoras para este proyecto se ha pesado, entre otras opciones:

Elección de dispositivos de red dentro de una cuenta de usuario:

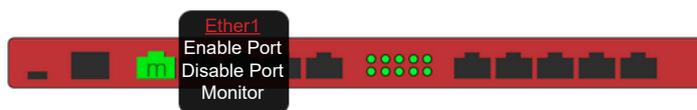
Cada usuario se dará de alta con un nombre de usuario y una contraseña, y será capaz de añadir los equipos de red por su dirección IP. Por lo tanto, si desea cambiar de equipo, no tendrá que desloguearse y elegir una nueva dirección IP, simplemente seleccionando dentro de su perfil el equipo de un menú desplegable bastará. Concluyendo que todos los equipos serán alcanzables dentro de un mismo perfil, agilizando la tarea de configuración.



Posibilidad de elección de equipo a configurar

Menús independientes por click en la imagen del puerto:

El usuario será capaz de haciendo click en la imagen del puerto, de configurar ese puerto, independientemente de los menús inferiores. Permitiendo una mayor movilidad y capacidad de configuración y haciendo a la API más estética e intuitiva.



Menú independiente por cada puerto clicando en la imagen

Analizar la posibilidad de ampliar la gama de configuraciones: routing, wireless, firewalling derivando a una solución cloud de configuración de dispositivos MikroTik.

Se pretende en un futuro aumentar la disponibilidad de equipos al mayor número posible de éstos junto a la mayor potencia de configuración posible ligado siempre a la mayor simplicidad.



Variedad de switches MikroTik

5. BIBLIOGRAFÍA

MikroTik Página Oficial

<<http://www.MikroTik.com>> [Consulta: 2014-2015]

Wiki MikroTik

<http://wiki.MikroTik.com/wiki/Main_Page> [Consulta: 2014-2015]

Formación MikroTik

<<http://www.formacionMikroTik.com/sobre-MikroTik/>> [Consulta: Junio 2015]

Transparencias Curso MTCNA

Raul Llinares y David Vaello. (2014). *MikroTik MTCNA (MikroTik Certified Network Associate)*.

¿Switching vs Routing?

<<http://neutron.ing.ucv.ve/revista-e/No4/articulo.htm>> [Consulta: Junio 2015]

El switch: cómo funciona y sus principales características

<<http://redestematicas.com/el-switch-como-funciona-y-sus-principales-caracteristicas/>> [Consulta: Junio 2015]

CCNP Switch 2: Funcionamiento de un Switch

<http://www.imd.guru/redes/certificaciones/ccnp_route-switch/switch/ccnp-switch-02-funcionamiento-de-un-switch.html> [Consulta: Junio 2015]

VLAN: Definición y montaje de una red virtual

<<https://fonquernie.wordpress.com/2013/01/23/VLAN-definicion-y-montaje-de-una-red-virtual/>> [Consulta: Junio 2015]

Wikipedia (Consultas varias)

<<https://es.wikipedia.org/>> [Consulta: Junio 2015]

Ubiquiti

<<https://www.ubnt.com/>> [Consulta: Junio 2015]

Características y configuración básica de VLANs

<<https://riunet.upv.es/bitstream/handle/10251/16310/Art%C3%ADculo%20docente%20configuraci%C3%B3n%20b%C3%A1sica%20VLANs.pdf?sequence=1>> [Consulta: Junio 2015]

Ventajas de las VLAN

<<https://sites.google.com/site/isaacivantorresgonzalezVLAN/home/ventajas-de-las-VLAN>> [Consulta: Junio 2015]

Instalar LAMP en Linux

<<http://blog.desdelinux.net/como-instalar-lamp-en-ubuntu/>> [Consulta: 2014-2015]

API PHP CLASS

<<http://blog.desdelinux.net/como-instalar-lamp-en-ubuntu/>> [Consulta: 2014-2015]

Documentación PHP

<<http://php.net/>> [Consulta: 2014-2015]

Documentación JQuery

< <http://jquery.com/>> [Consulta: 2014-2015]

Documentación Bootstrap

< <http://getbootstrap.com/>> [Consulta: 2014-2015]

W3Schools

< <http://www.w3schools.com/>> [Consulta: 2014-2015]

CSS-Tricks

< <https://css-tricks.com/>> [Consulta: 2014-2015]

Creating Ethernet VLANs on Catalyst Switches

< <http://www.cisco.com/c/en/us/support/docs/lan-switching/VLAN/10023-3.html>> [Consulta: 2014-2015]

HOWTO - VLAN configuration on HP ProCurve 2810 Switch

< http://www.skullbox.net/hp_procurve_VLAN.php> [Consulta: 2014-2015]

UniFi and switch VLAN configuration

< http://wiki.ubnt.com/UniFi_and_switch_VLAN_configuration> [Consulta: 2014-2015]

